Technical Report

UCAM-CL-TR-709
ISSN 1476-2986

Number 709

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

Protocols and technologies for security in pervasive computing and communications

Ford Long Wong

January 2008

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted August 2007 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Girton College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

As the state-of-the-art edges towards Mark Weiser's vision of ubiquitous computing (ubicomp), we found that we have to revise some previous assumptions about security engineering for this domain. Ubicomp devices have to be networked together to be able to realize their promise. To communicate securely amongst themselves, they have to establish secret session keys, but this is a difficult problem when this is done primarily over radio in an ad-hoc scenario, i.e. without the aid of an infrastructure (such as a PKI), and when it is assumed that the devices are resource-constrained and cannot perform complex calculations. Secondly, when ubicomp devices are carried by users as personal items, their permanent identifiers inadvertently allow the users to be tracked, to the detriment of user privacy. Unless there are deliberate improvements in designing for location privacy, ubicomp devices can be trivially detected, and linked to individual users, with discomfiting echoes of a surveillance society. Our findings and contributions are thus as follow. In considering session key establishment, we learnt that asymmetric cryptography is not axiomatically infeasible, and may in fact be essential, to counter possible attackers, for some of the more computationally capable (and important) devices. We next found existing attacker models to be inadequate, along with existing models of bootstrapping security associations, in ubicomp. We address the inadequacies with a contribution which we call: 'multi-channel security protocols', by leveraging on multiple channels, with different properties, existing in the said environment. We gained an appreciation of the fact that location privacy is really a multi-layer problem, particularly so in ubicomp, where an attacker often may have access to different layers. Our contributions in this area are to advance the design for location privacy by introducing a MAC-layer proposal with stronger unlinkability, and a physical-layer proposal with stronger unobservability.

Acknowledgements

There are many people to whom I owe thanks for the completion of this PhD dissertation.

First and foremost, I thank my supervisor, Frank Stajano, for his supervision, guidance, encouragement, feedback, editing, ideas, suggestions and other unnamed contributions.

Secondly, I thank my advisor, Prof. Andy Hopper, for his encouragement, support, advice and for providing an innovation-driven and practical-oriented environment in the Digital Technology Group (DTG, previously named Laboratory of Communication Engineering or LCE).

Next, I record my thanks to my colleagues in both the DTG and the Security Group, who provided collegiality, entertainment, and a spur and a sounding board for my research; and including Prof. Ross Anderson for his organising of the stimulating atmosphere in the Security Group, and his feedback.

In particular, I thank my collaborators: Jolyon Clulow, Hoon Wei Lim, Min Lin, Shishir Nagaraja and Ian Wassell. The following colleagues/friends: Øistein Andersen, Demetres Christofides, Saar Drimer, Feng Hao, Joon Woong Kim, Francisco Monteiro, Tyler Moore, Steven Murdoch and Piotr Zieliński, are also much appreciated for various fruitful discussions and help, and some of them for their proofreading of the whole or parts of this dissertation (though the responsibility for any errors is mine).

I am grateful to the Department of Engineering, and the Computer Laboratory, for their financial support of part of my conference travel costs.

I thank my parents for raising me and their forbearance of an absent son, and my sister and brother-in-law for family warmth.

I gratefully acknowledge sponsorship from DSO Singapore, especially support from the management, which enabled me to undertake my studies.

My deep appreciation also goes to my thesis examiners: Professor Bruce Christianson and Dr David Greaves, for their patient reading of the dissertation, and their invaluable suggestions for improvements/corrections.

Contents

List of Figures

Chapter 1

Introduction

Over a decade ago, Mark Weiser [1952-1999] coined the phrase ‘ubiquitous computing’ [1] (henceforth ubicomp) and outlined a futuristic vision of human-computer interaction [196], in which information processing is deeply integrated into everyday objects and activities. Today, reality has moved closer towards Weiser’s vision. We see deployments of Bluetooth devices, Radio Frequency Identification (RFID) tags, Wireless LAN hotspots; widespread adoption of wireless-equipped Personal Digital Assistants (PDAs), BlackBerry, and of course, the already true ubiquity of the cellphone, which is often feature-packed. We hardly bat an eyelid when a new class of networked device is rolled out in the market (– consider WLAN-equipped iPods, GPS-equipped cellphones, etc).

A common thread running through these objects is that they are often networked wirelessly, besides being interwoven into people’s everyday lives. In years to come, more networked applications will transition out of university and industrial labs, and enter mainstream product life. Appliances which are currently stand-alone are also expected to become networked to other devices. Also, with increasing numbers of sensors embedded into the environment, the user can look forward to a environment that is more intelligent and tuned to his individual needs. There is still some gap to be closed before devices ‘disappear’ and blend into the environment such that we are unaware that they are even there, to become a ’disappearing computer’ (or at least a disappearing piece of information-processing silicon). But we are getting there, in terms of ‘invisibility’ and networking.

However, things can also go seriously wrong. Security can break down. We are not referring to issues such as software bugs or obsolescence; we are referring to malicious adversaries, who would attack ubicomp systems having poor security design. Some of the previous security assumptions carried over from the homogeneous wireline network domain, to the ubicomp domain, as well as a certain conservatism regarding likely capabilities of ubicomp devices and attackers, are proving to be erroneous. In fact, certain ubicomp systems are clearly insecure, as we would show, over the course of this dissertation. So we need to address security at many levels: such as at the physical, link (i.e. Medium Access

[1] Pervasive computing is sometimes treated as subtly different from ubiquitous computing, but in this dissertation we will consider them as the same, as according to Satyanarayanan [166].

Control or MAC), network, transport, and application (such as in terms of usability and human-computer interaction); and in many different dimensions: authentication, confidentiality, authorization and privacy (including location privacy). Thinking in terms of these dimensions also spurs us to differentiate the channels available in ubicomp according to their properties, and hence optimize their use. The thesis which we put forward, is that ubicomp security on the whole should be re-evaluated on the basis of **multiple levels** and **multiple channels**, since these are often all at once exposed in ubicomp, more so than in a more traditional (wireline) network environment.

We ought to be a little paranoid when designing for security in such a ubicomp environment. For a start, with so many devices in the mix, there is even greater potential for confusion. Confidentiality in communications between devices, if required, can be assured by encryption, which is a well-known construct. With a lack of an infrastructure (such as a Public Key Infrastructure (PKI)) in ubicomp, the problem of initial secret session key establishment may be solved by sharing some short string and then performing some cryptography to produce a strong random key, to be used to encrypt communications, secure against an eavesdropper.

When the work for this PhD began, the prevailing wisdom regarding the use of asymmetric cryptography in pervasive devices was that the computational costs that this would have entailed were simply too heavy for ostensibly lightweight processors in pervasive devices to bear. The problem is that this viewpoint had fallen well behind the continuing improvements in hardware, as well as attacker sophistication. One of our contributions in this area is to demonstrate, via an implementation, that the symmetric cryptography-based password-authenticated key agreement of a particular wireless pervasive technology, namely Bluetooth, can be practically attacked. While other researchers have considered that this is possible, ours is the first such attack implemented in the open literature. Secondly, we showed, through prototype implementation, that processors in some pervasive devices, certainly in handphones at least, could perform asymmetric cryptography without unbearable latencies. Thus, we believe that our research is helping to change the mindset that asymmetric cryptography was irreconcilable with pervasive devices. These are described in Chapter 2. Recent standardization work to update the security specifications of Bluetooth, among other wireless technologies, is suggesting that our viewpoint is essentially correct, and is in step with (if not leading) the march of technology.

Despite their limitations, passwords are still a well-accepted means of authentication by users, are convenient, and their use may not be a lost cause. If we are to retain passwords for bootstrapping, we have explored and developed a scheme to do inter-domain multi-party key agreement between two clients, each belonging to different domains, (mediated by their respective domain servers, hence ‘multi-party’,) by leveraging on identity-based cryptography. Identity-based cryptography is one of the more recent advances in cryptography, and we consider that our application is a novel use of this, by combining it with passwords. Inter-domain authentication is an under-researched area, even though we perceive that in the future, clients registered under multiple authentication domains may very well need to carry out transactions with one another. We have proposed a scheme which is lightweight in terms of communication complexity. One of the key issues involved in the use of an identity-based cryptographic scheme is the requirement to distribute domain parameters, and hence a PKI is usually required at the client side, but in our scheme we are able to design away these requirements, but yet not compromise security. We achieve the aim of reducing excessive communication complexity from certificate checking by the clients. The domain servers involved can still do certificate checking. We believe that our solution offers some very useful insights for inter-domain identity-based cryptographic protocols. We cover all these in Chapter 3.

Assuring that the party which one's device is forming the security association with is the correct party in the first place, was not a fully addressed problem. Even though some initial work had begun quite some years before, there have been significant shortcomings in previous solutions, in the lack of appreciation of the potential of multiple channels and their properties. There were also inadequacies in the previous attacker models, when the problem is specified to the pervasive computing environment. We are some of the first to develop new ways of describing and solving the problem, suited to the said operating environment. We call this: 'multi-channel security protocols'. As indicated by the name, we make use of the different types of channels available. In particular, the property of authenticity in some of the auxiliary channels available in ubicomp allows a strong security association to be bootstrapped, one which is resistant to passive and attackers, including active attackers who can carry out a collision attack to brute-force hash codes or key fingerprints. Our protocols allow us to do away with password use, and the attendant problem of human memorability. We propose that the explicit consideration of channel properties, hitherto often considered as an afterthought, offers more clarity in the design of security protocols. Our contribution is also in the proposal and analysis of specific secure multi-channel two-party (in Chapter 4) and multi-party key agreement (in Chapter 5) protocols themselves, in addition to the multi-channel protocol design viewpoint (Chapter 4), whose utility may extend beyond ubicomp. We feel that the 'multi-channel protocols' would influence the field of protocol research, and be a fertile area. It can be seen that the research community is just beginning to analyze and tackle the set of problems.

So far, we have covered mainly issues of secret key establishment. Another area which we have considered in the dissertation, is privacy, in particular, location privacy. The ubicomp vision has this one other angle to it: the preponderance of sensing devices, networked and embedded in the environment, unfortunately, is beginning to reek of a surveillance society. The danger is that an adversary who controls a portion of these sensing devices (or the database of movement information which these devices write back to) can track an unsuspecting human subject as he goes about his business. This is an important technical problem to address, the outcomes of which can help keep society open and free. The use of temporary pseudonyms has been proposed by other researchers, and has even seen deployment in today's GSM cellular communications. For the pervasive wireless technology of Bluetooth, whose devices are currently highly track-able due to each device's use of a permanent MAC layer identifier, several researchers have proposed location privacy solutions. Our contribution is to propose a more refined pseudonym-based solution, with stronger unlinkability, which yet retains pairwise statefulness. We make use of simple primitives — hash functions, which are certainly lightweight enough. We have enumerated all the location privacy vulnerabilities in Bluetooth, and also sketched other technical and policy mechanisms to secure these. This work is covered in Chapter 6. Many researchers in pervasive computing privacy have concentrated on RFID devices, which are processing- and memory-constrained dumb tags for supply chain and inventorizing purposes, while we have chosen to address the rather neglected space of ad-hoc short-range wireless connectivity technology typified by Bluetooth. It is worth noting that Bluetooth devices have proliferated in their millions and have entered mainstream life, and as mentioned earlier, new versions of the Bluetooth specifications are being prepared to get things right the next time round. We believe our location privacy work would contribute to this effort (as our work [201] has been cited by a number of publications).

It is sometimes easy to be lulled into believing that the location privacy problem of a wireless technology can be completely solved once the conspicuous problems at, say, the MAC layer, are solved. However, it has been suggested by others, and we concur here, that it is important to appreciate that location privacy is actually a multi-layer problem. Solving the problem at higher layers does not automatically address the problems at the other layers; each layer must be secured individually and also de-linked with respect to one another. We highlight in particular the vulnerability at the lowest layer. In our thread of work, we consider the leakage of location information via the lowest (i.e. physical) layer itself, to a passive adversary, who can localize the position of an emanating source by means of direction-finding. For the wireless technology of wireless LAN, in the outdoor environment, we show from simulations that it is possible to reduce the observability of signals from your mobile device to unintended parties. We parameterized the problem, and proposed a solution based on adaptive beamforming, leveraging on the advent of multiple antenna found in new generations of radio hardware. We believe that ours is one of the earliest proposals of using multiple antenna to improve location privacy in the civilian domain. One of our contributions is to construct an evaluation framework to analyze the security offered by our proposed solution, and actually compare it side-by-side with the status quo solution. This framework is flexible, and could be extended to consider other location-privacy-enhancing schemes. The work is described in Chapter 7. We believe our work will encourage other researchers to look more closely at the problem of observability and thence location privacy at the physical layer.

In terms of joint work, I have learnt much from working with other researchers. The papers which I have written during the course of the PhD programme are often the result of joint work with collaborators. A list of these refereed papers is provided below.

[205] Ford-Long Wong, Frank Stajano and Jolyon Clulow. "Repairing the Bluetooth Pairing Protocol". In "Proceedings of the 13th International Workshop in Security Protocols", Cambridge, UK, Apr 2005.

[199] Ford Long Wong and Hoon Wei Lim. "Identity-Based and Inter-Domain Password Authenticated Key Exchange for Lightweight Clients". In "Proceedings of the 3rd IEEE International Symposium on Security in Networks and Distributed Systems", Niagara Falls, Canada, May 2007.

[202] Ford-Long Wong and Frank Stajano. "Multi-channel Protocols". In "Proceedings of the 13th International Workshop in Security Protocols", Cambridge, UK, Apr 2005.

[203] Ford-Long Wong and Frank Stajano. “Multi-channel Protocols for Group Key Agreement in Arbitrary Topologies”. In “Proceedings of 3rd IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2006)”, Pisa, Italy, March 2006.

[204] Ford Long Wong and Frank Stajano. “Multichannel Security Protocols”. To appear in *IEEE Pervasive Computing, Special Issue on Security & Privacy*, Oct-Dec 2007.

[201] Ford-Long Wong and Frank Stajano. “Location Privacy in Bluetooth”. In Refik Molva, Gene Tsudik and Dirk Westhoff (eds.), “Proceedings of 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks”, vol. 3813 of *Lecture Notes in Computer Science*, pp. 176–188. Springer-Verlag, Visegrad, Hungary, July 2005.

[200] Ford Long Wong, Min Lin, Shishir Nagaraja, Ian Wassell and Frank Stajano. “Evaluation Framework of Location Privacy of Wireless Mobile Systems with Arbitrary Beam Pattern”. In “Proceedings of 5th Conference on Communications Networks and Services Research”, IEEE, Fredericton, New Brunswick, May 2007.

[205] is described in Chapter 2. Frank Stajano provided much instructions and guidance to shepherd this paper into its final form. Jolyon Clulow made an important contribution by spotting an error in our earlier version of the protocol fix, which was corrected for the final version.

[199] is covered in Chapter 3. Hoon Wei Lim, my collaborator, provided the initial idea of combining secret public keys with identity-based cryptography in an earlier work. I suggested applying this to the multi-party inter-domain setting. For the development, refinement, security analysis and performance analysis, both authors contributed equally.

[202] is covered in Chapter 4. The analyses, example and protocol proposals are my contributions. Frank made important contributions in generalizing the consideration into the issue of different channels having different properties (hence the term ‘Multi-channel’), highlighting the shortcomings in current protocol design. Frank also revised the original draft, so that progressive improvements over existing protocols could be more easily understood by the readers.

[203] is described in Chapter 5. The analyses and protocol proposals are my work. Frank helped proof-read it and made editorial recommendations to present the problem statement in a more intuitive and smoother way.

Likewise, for [204], which would be published in the fourth quarter of 2007, and covers subject matter described in Chapters 4 and 5, Frank has made significant contributions in refining the presentation so as to make it suit better the broad readership of the publication in question. The protocol improvements in [204], over those in [202] and [203] were my work, as were the implementations.

[201] is described in Chapter 6. This is mostly my work. Frank pointed out flaws in an early version of the stateful pseudonym protocol, and also contributed in the organization and writing of the paper.

[200] is the article for me which involves the most co-authors. Its contents are covered in Chapter 7. Each collaborator brings something unique to the joint effort. While I conceived of the work, I am indebted to Min Lin and Shishir Nagaraja for their contributions to this investigation; Min for providing the

direction-finding simulation code, and doing the early part of the radio simulations; Shishir for ideas on improving joint radio-mobility simulation and evaluating mobile location privacy. I also owe Frank and Ian Wassell credit for critiques of arguments and assumptions in earlier drafts, providing proofreading, sanity-checking and editorial improvements.

The dissertation's title 'Protocols and Technologies for Security in Pervasive Computing and Communications' is finally chosen, on the basis of its inclusiveness of the entire contents we intend to cover. Had we omitted the work on physical layer location privacy in Chapter 7 (which is salient to our argument on multi-layer location privacy), the title may have well been shortened to just 'Protocols for Security in Pervasive Computing'. As the reader would discover, the chapter covers beamforming, which is more of a technology for communications, rather than a protocol for computing, thus we retain the current title for this dissertation.

Chapter 2

Password-based Device Pairing

2.1 Outline

This chapter discusses the use of password channels and asymmetric cryptography in key agreements between ubiquitous computing devices.

One of the premises in this chapter is that passwords are used over a secure side-channel in ubicomp. Channels must be used with their specific security properties in mind; the specific property of the password channel is its confidentiality against likely attackers. It is important to ensure that the channels used are fit for purpose. An argument we want to advance is that passwords may and *should* now be used with asymmetric key agreement methods in ubicomp devices, to resist trivial guessing attacks.

We implemented an attack against the otherwise weak symmetric key agreement technique used in a wireless technology which typifies ubicomp, i.e. Bluetooth. We show that such an attack can be effectively mounted, and given that it is such a clear threat, asymmetric key agreement methods should be used, and these are not necessarily prohibitive in terms of resource requirements. We implemented a stronger key agreement protocol which makes use of asymmetric techniques, on a laptop as well as on a mobile phone, to ascertain whether the processing delay of computationally complex asymmetric operations is unbearably long; in fact it is not; latencies and usability seem to remain reasonably good. Thus, another consideration of this chapter is the impact of latencies in key agreement on the quality of the user experience. This should be considered up-front, because pervasive computing devices often have constrained processors.

2.2 Related Work

Passwords are a well-known construct in use in security. Here, we are not concerned with the problem of the cracking of a password file, in say, a desktop system, in the case when an attacker manages to gain unauthorized access to the system and retrieve a copy of the password file. An example that comes to mind is the weak Microsoft Windows LAN Manager hash (also known as LANMAN or LM hash) format, which is used to encrypt passwords in terminals [138]; this is highly vulnerable to brute-forcing even if the passwords used are long because the algorithm operates on passwords in 7-character chunks.

We are concerned with a variant of the problem, where passwords are used by two parties in a password-authenticated key agreement protocol, and where messages are transmitted over an insecure network prior to the establishment of the strong key. The messages have freshness. Intuitively, it had seemed impossible to be able to use a short secret to do key establishment to produce a strong key, such that brute-force search to find the short secret is not possible [32]. It was said that this intuition may explain why the first strong password-authenticated key agreement protocols did not appear until 1989, such as the one by Lomas et al [125] which was set in a client-server scenario, where it used the additional assumption that the client knows the server's public key beforehand.

Bellovin and Merritt proposed a class of protocols called Encrypted Key Exchange (EKE) in 1992, which did not require the additional assumption as above. In EKE, the initiator generates an ephemeral public key, but uses the shared password to encrypt the public key. The responder decrypts and obtains the public key, and uses it to send a session key which he generates back to the initiator. The idea is that an attacker who brute-forces the passwords would not be able to distinguish what ephemeral public key was used. And even when the public key was found by the attacker, it cannot be used to learn the session key since the private key could not be obtained from the public key. Halevi and Krawczyk [86] presented a viewpoint, with a formal proof, that such strength (and also, in fact the property of forward secrecy) may only be provided by the use of asymmetric cryptography.

The original EKE had not specified the encryption to be used, nor how the password would be exactly transformed into a key. But the EKE spawned many password-authenticated key agreement protocols. Many, though not all, in their asymmetric component, are based on Diffie-Hellman key agreement. Various protocols, such SPEKE [98], SRP [207], PAK [33], AMP [114], are being proposed to and adopted by public-key cryptography standardization committees, such as in the IEEE P1363.2 draft standard.

There were various other password-based protocol proposals, such as by Roe et al. [159] and Christianson et al. [45], which use the RSA algorithm, 'confounders', and neither hashes nor symmetric cryptography, and a novel one by Anderson and Lomas [7], which uses 'collisionful' [75] hashes.

However, the way in which passwords are used in Bluetooth today, may be said to be of the same generation as in the Windows LANMAN.

2.3 Ubicomp Environment - Lack of an Infrastructure

In ubiquitous computing, devices usually do not have access to a heavyweight infrastructure, such as a Public Key Infrastructure (PKI). If such an infrastructure is available, then authenticity between ubiquitous devices meeting even for the first time would be straightforward. Using certificates issued by a mutually trusted Certificate Authority, a device can authenticate another device. Establishment of a secret session for secure communications between the devices can be subsequently carried out with the aid of these digital certificates [16], using asymmetric key establishment techniques if necessary.

In the absence of an infrastructure, there needs to be other mechanisms to bootstrap the security association. A landmark work in this area is Stajano and Anderson's Resurrecting Duckling [180], which proposed a policy model and a contact-based mechanism to share a key. One of the contributions in the work by Stajano and Anderson is the recognition that there needs to be an alternative and usable way of bootstrapping security association between ubiquitous devices without using infrastructural support. The suggested way is through physical contact.

Apart from physical contact, which is a secure channel, much the same effect is achieved through using passwords on the two clients which are attempting to carry out an authenticated key agreement.

2.4 The Private Channel - Passwords

A password would have been pre-shared through a confidential out-of-band channel. The advantages of using passwords are clear. Passwords are convenient and they are intuitive to users. Unlike physical tokens, they are not required to be carried around, except in one's head. However, the advantage of passwords in being human-memorably short is also its disadvantage.

Definition 2.1 : *Password entropy* is the uncertainty of an attacker's knowledge of the password, or how hard it is to guess it. It can be expressed in bits.

$$\text{Entropy of } X = -\sum_{i}^{n} Pr[X = x_i] \log_2 Pr[X = x_i]$$

where X is the password and can be thought of as a discrete random variable that can take on possible values $\{x_1, x_2, \cdots, x_n\}$. If the passwords are randomly chosen, then the password entropy is the size of the password space, for example, a random 16-bit number has 2^{16} possible values (i.e. $n = 2^{16}$), and its entropy would be 16 bits. We use 'space' inter-changeably with the term *cardinality*, the latter which is sometimes encountered in more mathematically oriented literature, to denote 'the number of elements in the set'.

Unfortunately, due to reasons of human memorability, not only is the password space not large, [139; 209], but also, the chosen passwords are often not random (because certain passwords have a higher probability than others of being chosen, such as '0000' or '1234' for 4-digit passwords). Thus, passwords can be described as low-entropy secrets because an adversary can easily carry out a brute-force attack or a dictionary attack against a password, resulting in the secret being compromised. Passwords may also described as *weak*, or *low-grade* secrets, which carry the same meaning.

Passwords have a long history in security protocol design. Passwords comprise one of the three pillars of authentication, being 'what you know'. The other two are 'biometrics: what you are', and 'tokens: what you have'.

In many information technology applications having access control, password authentication often occupies the first line of defence. To access one's email account, to log in to a work terminal, to access one's internet banking account, or to withdraw money from an Automated Teller Machine (ATM), the use of passwords is well-established.

2.4.1 Channel Properties

Before we begin to enumerate properties of the password channel, it would be worthwhile to define our concept of a *channel*. The intended meaning of the word 'channel' usually becomes clear in its particular context of use. The term is used in many closely related fields, such as communications, and even radio spectrum allocation, etc.

Definition 2.2 : A *channel* refers to the medium through which information is transmitted from a sender (or transmitter) to a recipient (or receiver). It can also be thought of functionally as a signal path over which a message passes between two endpoints.

Roughly speaking, we consider each channel as one type of physical medium. In our context, a 1 Mbps capacity channel at 2.4 GHz, and another 1 Mbps capacity channel at 5.6 GHz, are the same type of channels in our viewpoint, in the sense that they are both radio frequency channels.

Thus, in our usage, one of the central features of the password channel is the property that the information is transmitted confidentially between the source and destination. There is also high integrity and authenticity with the information transfer. Often, but not always, this password channel operates with a human in the loop, who is manually pressing buttons on a keypad with his/her fingers, and guided by visual feedback. This may sound patently obvious, but stating with explicitness, as a later chapter would show, is necessary to take into account the implications of different channels and their properties, in a pervasive computing environment.

2.4.2 Limitations of Passwords

The particular shortcomings of passwords must be factored into their usage in security protocols. A misunderstanding of the actual scenario in which security protocol operates would open up weaknesses.

Thus, besides the low entropy, which predisposes passwords to dictionary attacks if a security protocol is badly designed, password use is predicated on the existence of a *confidential* channel. If the assumption of confidentiality is violated, i.e. an attacker observes the PIN (essentially a numeric password) which you key into your keypad, then the security protocol will be defeated.

Also, despite the ability of most people to reliably remember a small number of passwords in everyday life, this ability actually degrades if the number of passwords increases beyond manageable sizes. Thus, password use is often not scalable. Users may use the same passwords for different accounts – that is an obvious problem. Or they may use passwords which are only very slight variations of their other passwords.

As indicated in a previous section, passwords are often not chosen with total randomness by users. For numeric PINs, '1234' for example seems to have a high propensity to be chosen, such that the password probability is not uniformly distributed, hence the phrase 'dictionary attack'.

When a certain password is in use unchanged for a long time, its vulnerability increases because a series of infrequent password guesses may be occurring in the background and have escaped notice. This threat requires passwords to be regularly changed. Similarly, an attacker may attempt to mount repeated in-band guessing attacks within a short space of time. Against this, possible defences would include throttling guessing attempts exceeding a threshold within a specified length time, logging unsuccessful attempts for subsequent audit, or locking out the account/service when a threshold is reached. All these do have a negative impact on usability.

Having described these various constraints, we have not even touched on the technical design of protocols where passwords are an essential part of. For this, the protocol trace needs to be invulnerable to a offline dictionary/brute-force password search. We will come across such a vulnerability clearly in the next section.

2.5 A Vulnerable Symmetric-Key Two-Party Password-Based Key Agreement Protocol

Menezes, van Oorschot and Vanstone [137] gave the following definitions:

Definition 2.3 : *Key establishment* is a process or protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use.

Key establishment may be in turn be broadly subdivided into *key transport* and *key agreement*.

Definition 2.4 : A *key agreement* protocol or mechanism is a key establishment technique in which a shared secret is derived by two (or more) parties as a function of information contributed by, or associated with, each of these, (ideally) such that no party can predetermine the resulting value.

We overview a simple two-party password-based key agreement protocol, used by the existing ubicomp technology of Bluetooth. The Bluetooth specifications [79; 80; 81] define a short-range cable replacement wireless technology for ubiquitous devices, such as laptops, personal digital assistants, printers, headsets and other peripherals.

This example illustrates that generating a session key using passwords and symmetric key agreement techniques is highly insecure. This fact was known [86]. However, what we show through our research and implementation which we conducted, is that the attack is highly practical. We also show that fixing the flaw is not beyond the capability of certain new hardware with less resource constraints than that envisaged by the original Bluetooth design parameters.

Bluetooth natively provides authentication and encryption. Authentication is provided by a 128-bit link key, which is a shared secret known to a pair of Bluetooth devices which have previously formed a pair. The cipher algorithm is E0—a stream cipher whose key can be up to 128 bits long. The algorithms (i.e. E_1, E_{21}, E_{22}, E_3) for authentication and for the generation of link and encryption keys are all based on SAFER+ [131], here used basically as a hash.

The hash function is a very commonly used primitive in security. It will be worthwhile to summarize its properties here.

Definition 2.5 : Minimally, a *hash function* H maps an input x of arbitrary length into a hash output $H(x)$ of fixed length, thus *compression* takes place; and $H(x)$ is easy to compute when x is given. In cryptographic usage, there are other desirable properties (which are often taken for granted) for hash functions. The computation in the forward direction to calculate $H(x)$ when x is given is easy, while the reverse direction is difficult – this property is termed *one-wayness* and it also implies *pre-image resistance*. In the context of security, calculating x when $H(x)$ is given, is called the *pre-image attack*. A hash function also needs to have the property of *second pre-image resistance*: given x, to find a second pre-image x' such that $x' \neq x$ while $H(x') = H(x)$, is computationally difficult. For the hash function to possess the property of *collision-resistance*, this means that it is difficult to find two distinct inputs (where there is free choice of these inputs) which hash to the same output – the attack corresponding to this is called the collision attack or *birthday attack*.

Returning to Bluetooth, its security architecture defines three possible security modes for a device [29; 79]. Mode 1 is non-secure, Mode 2 enforces security at the fine-grained service level and Mode 3 enforces security at the link level.

In the case of Modes 2 and 3, if pairing is turned on, when two Bluetooth devices meet for the first time, they pair using the following key:

$$K_{init} \quad = \quad E_{22}\{PIN, BD\_ADDR_A, RAND_B\}$$

where BD_ADDR_A is the 48-bit device address of device A, $RAND_B$ is a 128-bit random number contributed by device B and PIN is a shared password that the user must generate and then manually enter into both devices. So, BD_ADDR_A and $RAND_B$ can be read in cleartext by an eavesdropper, while PIN cannot.

Once two devices share a link key, the following protocol allows them to renew it and derive a new one, known as a combination key K_{AB}, which becomes the new link key used from that point onwards.

The devices each produce a random number (LK_RAND_A or LK_RAND_B), mask it by XORing it with K_{init} and send it to the other party. Both parties individually hash each of the two random numbers with the Bluetooth address of the device that generated it, using the algorithm E_{21}. The two hashes are then XORed to generate the combination key:

$$\begin{aligned} K_{AB} &= E_{21}(LK\_RAND_A, BD\_ADDR_A) \\ &\quad \oplus E_{21}(LK\_RAND_B, BD\_ADDR_B). \end{aligned}$$

The combination link keys calculated by each device after the key agreement should of course be the same if the procedure is successful. The old link key (either K_{init} or a previous K_{AB}) is then discarded.

Another, less secure kind of link key is the unit key K_{unit}, used by devices that don't have the memory to store a link key for each pairing. The unit key has been generated once, at installation, of the Bluetooth device, and thereafter seldom changed. The restricted memory device will negotiate to use its unit key as the pairwise link key. It will then mask the unit key by XORing it to the K_{init} formed earlier and send it over to the other device.

For authentication, a challenge-response scheme is used. Device A sends B a challenge $RAND_A$, from which Device B must produce the authentication token $SRES$ and transmit it to A, based on the following:

$$\{SRES, ACO\} = E_1\{K_{AB}, RAND_A, BD\_ADDR_B\}$$

The challenge-response is run bilaterally. If encryption is needed, the encryption key is derived from the link key.

2.5.1 Bluetooth Pairing - Our Implemented Attack

We used a Bluetooth protocol analyzer [141] to obtain a trace of Bluetooth transmission from the air and to decode the packets. A picture of the device is shown in Fig. 2.1. We captured traces of transmissions from commercial Bluetooth devices such as handphones, PDAs, laptops, etc. The packets bearing the required pairing, key formation and authentication processes, as shown in Fig. 2.2, were analyzed.

We wrote a program that parsed the captured traces and extracted the relevant parameters as they appeared in the transmissions. The trace contains the device addresses, the random numbers exchanged, the challenge and response tokens and all other relevant parameters of one protocol run. Using these, we carried out a kind of dictionary attack (trying first the "easy" PINs such as those made of repeated or sequential digits) and then, where necessary, a full brute force search of the small PIN space. The user interfaces of many Bluetooth devices further restrict the set of characters that may be used in the PIN. For handphones, this set is often just 0 to 9. For each PIN we first computed the K_{init}. Then, using the observed intermediate parameters, we computed the combination key and authentication token $SRES$. Both the key agreement using the combination key and the key transport using the unit key can be successfully attacked. If the trace showed that a unit key had been used instead, the number of intermediate parameters is even fewer. Guessing a correct PIN results in a $SRES$ value identical to the one observed in the trace. Thus, we can infer from matching $SRES$ values that a correct PIN has been found, with high probability.

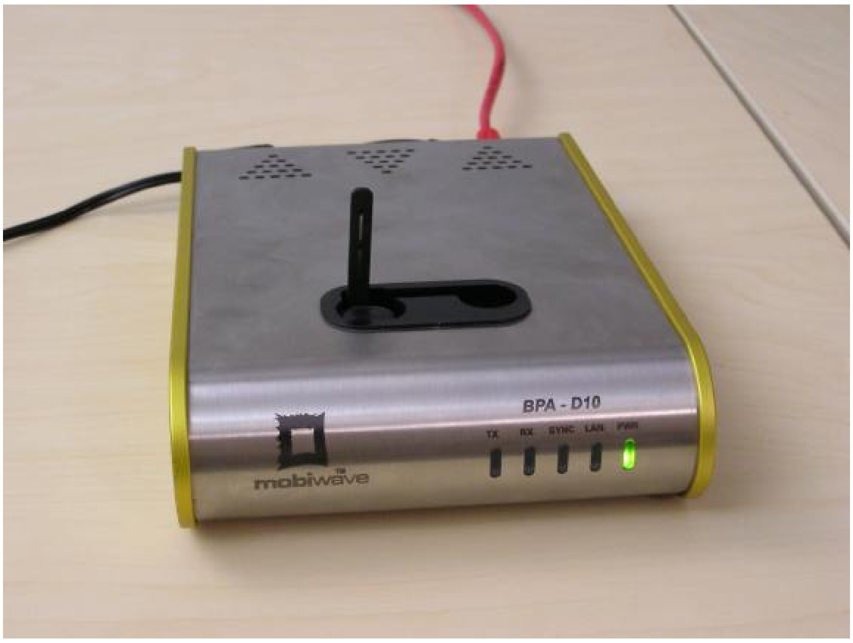
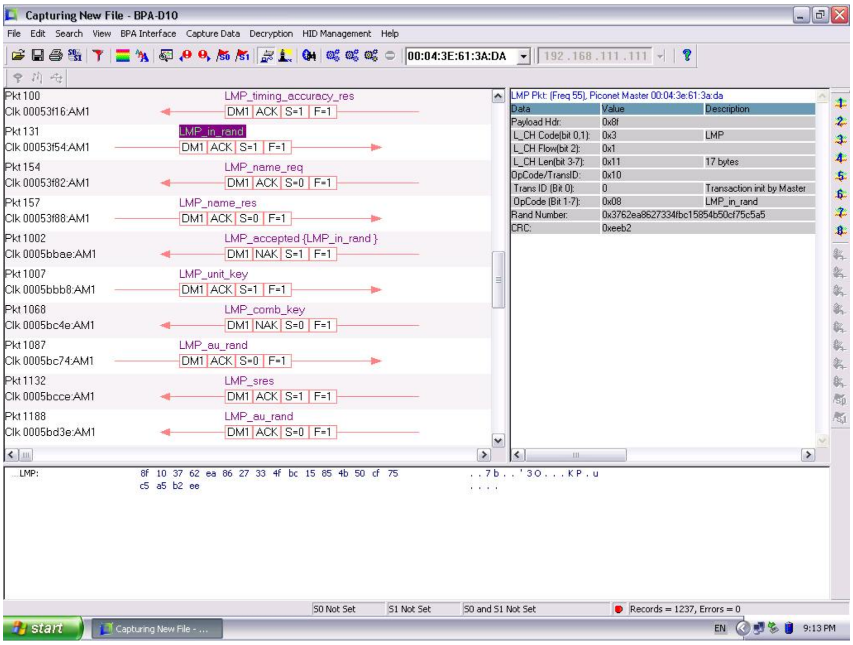


Figure 2.1: Bluetooth Protocol Analyzer used



```
Pkt 100             LMP_timing_accuracy_res
Clk 00053f16:AM1    DM1 ACK S=1 F=1
Pkt 131             LMP_in_rand
Clk 00053f54:AM1    DM1 ACK S=1 F=1
Pkt 154             LMP_name_req
Clk 00053f82:AM1    DM1 ACK S=0 F=1
Pkt 157             LMP_name_res
Clk 00053f88:AM1    DM1 ACK S=0 F=1
Pkt 1002            LMP_accepted {LMP_in_rand }
Clk 0005bbae:AM1    DM1 NAK S=1 F=1
Pkt 1007            LMP_unit_key
Clk 0005bbb8:AM1    DM1 ACK S=1 F=1
Pkt 1068            LMP_comb_key
Clk 0005bc4e:AM1    DM1 NAK S=0 F=1
Pkt 1087            LMP_au_rand
Clk 0005bc74:AM1    DM1 ACK S=0 F=1
Pkt 1132            LMP_sres
Clk 0005bcce:AM1    DM1 ACK S=1 F=1
Pkt 1188            LMP_au_rand
Clk 0005bd3e:AM1    DM1 ACK S=0 F=1
```

LMP Pkt: (Freq 55), Piconet Master 00:04:3e:61:3a:da

Data	Value	Description
Payload Hdr:	0x8f	
L_CH Code(bit 0,1):	0x3	LMP
L_CH Flow(bit 2):	0x1	
L_CH Len(bit 3-7):	0x11	17 bytes
OpCode/TransID:	0x10	
Trans ID (Bit 0):	0	Transaction init by Master
OpCode (Bit 1-7):	0x08	LMP_in_rand
Rand Number:	0x3762ea8627334fbc15854b50cf75c5a5	
CRC:	0xeeb2	

```
....LMP:        8f 10 37 62 ea 86 27 33 4f bc 15 85 4b 50 cf 75      . . 7 b . . ' 3 O . . . K P . u
                c5 a5 b2 ee                                          . . . .
```

S0 Not Set | S1 Not Set | S0 and S1 Not Set | Records = 1237, Errors = 0

Figure 2.2: Protocol Analyzer Capture of Bluetooth Key Agreement

2.5.2 Performance of Attack

The attack program was a prototype and had not been optimised for speed. Running it on a 1.2 GHz Pentium III laptop gave the following timing results (for randomly chosen PINs, of course—the ones subject to pseudo-dictionary attack could be cracked instantaneously). Note that cracking combination keys require more calls to E_{21} compared to cracking unit keys. Even with our non-optimised program, 4-digit PINs can be broken instantaneously, 6-digit PINs take less than a minute and 8-digit PINs can be cracked in less than an hour (Fig. 2.3). This is due to the very small size of the PIN space.

Type of key	**No. of digits**	**Time taken**
Unit key	4	0.15 s
	6	15 s
	8	25 mins
Combination key	4	0.27 s
	6	25 s
	8	42 mins

Figure 2.3: Time to Crack Bluetooth PIN

We have found the unit key used in some models of Bluetooth-equipped PDAs. As mentioned in [100] and elsewhere, once the unit key is discovered, the attacker can thereafter freely impersonate the device in its interactions with any other device. Fortunately, we found that at least the unit key was not set to the same string from device to device for the manufacturer's particular line of products. The unit key is deprecated from version 1.2 onwards of the Bluetooth specification [80].

The only current practical obstacle to widespread replication of the attack described here is that not every would-be eavesdropper will have easy access to a Bluetooth protocol analyzer ($\approx$ 3000 GBP), which we have used. We predicted [205] in 2005 that enterprising hackers could in time figure out ways to use cheap standard Bluetooth modules to access the baseband layer directly, without requiring expensive debugging and diagnostic tools. In 2007, some progress in this direction has been made by one researcher, whereby he was able to flash the firmware, and change the Bluetooth device address of a Bluetooth dongle of a particular model by a particular vendor [142]. It was suggested that further development could indeed turn a cheaply available commercial Bluetooth dongle into a sniffer, without needing to obtain an expensive specialized Bluetooth protocol analyzer, making the entry barrier for an eavesdropping Bluetooth hacker very low indeed.

2.5.3 Re-Keying as a Short-Term Remedy

Although the protocol is broken by an attacker who eavesdrops on the key establishment phase, within its constraints of avoiding public key cryptography it is not an overly weak design. If the attacker misses the initial key establishment, this attack cannot work. Moreover, as discussed in Part C, Section 3.4

of Version 1.1 [79] and Vol 2, Section 4.2.3 of Version 1.2 [80] of the specification, Bluetooth already supports renewal of the link key. An attacker who knows the previous link key and eavesdrops on the link key renewal exchange can trivially discover the new key, but if he misses that exchange then the two devices are safe from further eavesdropping, unless they recourse to bootstrapping a new link key from the weak PIN. This is a good security feature but unfortunately it is rarely exploited in current commercial Bluetooth devices. Change of link key is cheap for the user because it does not involve typing a PIN; yet most devices do not offer the option.

As a concrete example, we can propose that the initiator, Device A send a random number R_A to Device B, and both devices then calculate a new link key K'_{AB} as follows (conveniently re-using the existing E_{21} algorithm in Bluetooth):

$$\begin{aligned} K'_{AB} &= E_{21}(R_A, BD\_ADDR_A) \\ &\quad \oplus E_{21}(K_{AB}, BD\_ADDR_B). \end{aligned}$$

After this new link key K'_{AB} is calculated, a bilateral challenge-response (generating an authentication token, $SRES$ each time) would be carried out to confirm to both devices that the other party has successfully updated to the same link key.

We do propose that manufacturers provide users with the ability to initiate link key change on their paired devices whenever they wish and we further recommend that users exercise this option often. In fact, this is conceivably very simple to implement, and can be set by policy so that when two previously paired devices launch any successive communication session with each other, they would simply run the link key change protocol transparently from the users, at the end of which the copies of the pairwise link key on the two devices would be updated accordingly. Frequent change of link key forces an attacker to be continually present when the two target devices are communicating.

For resistance against an attacker who could be continually present when the link key is changed, then there would be a compelling case to calculate the initial key via asymmetric techniques, and perhaps to even further re-key via asymmetric techniques; these are not yet supported in the current specification though. Frequent change of link key would also mitigate the risks of Barkan-Biham-Keller-style encryption key replay attacks raised by Ritvanen and Nyberg [158]. Fixes to the Bluetooth cipher algorithm and encryption key derivation are however, beyond our scope.

After our work [205] was presented at a workshop, another paper [172] appeared, which performed a similar attack to the one described in Sections 2.1 and 2.2. The computation speed of their further optimized attack seemed to be of the same order as ours. An attack suggested in that paper is to force repeated Bluetooth pairing of devices which have already been paired, so as to exploit the existing vulnerability to offline guessing. However, there was no actual demonstration of this using actual commercial devices, so we are unable to conclude if the attack is indeed feasible.

Vaudenay also suggested re-keying as a fix [189], which post-dated our paper. He provided a security proof of the security of re-keying.

We note that the authors of [172; 189; 205] all use the term ‘repairing’, but to mean slightly different things! We [205] used it to refer to rectifying and strengthening a broken protocol, Shaked et al [172] used it to refer to an attack to fool already-paired devices into initiating a new password-authenticated protocol run, and Vaudenay [189] used it to refer to renewal of the link key – the last of which of course is the very topic of this sub-section.

2.6 Secure Asymmetric-Key Two-Party Password-Based Key Agreement Protocols

2.6.1 Required Security Properties

After ascertaining the practical vulnerability of the Bluetooth pairing protocol, we attempted to repair it. The first goal was to establish a strong, eavesdropper-resistant shared key between the two paired devices. We found it difficult to do this within the thrifty constraints chosen by the Bluetooth designers, so we had to resort to asymmetric cryptography, in particular to the Diffie-Hellman key agreement. We sought to make this exchange resistant against active man-in-the-middle attacks. For this we turned to the vast literature on Encrypted Key Exchange [25] and derivatives.

Let us outline the basic security properties [137] required of the replacement protocol. It is worthwhile to define these before using, because different authors have sometimes subtly different meanings for their usage of the terms, which they could not agree upon [32, Chapter 2].

Definition 2.6 : An authenticated key establishment protocol is a key establishment protocol which provides *key authentication* (defined below).

Definition 2.7 : *Key authentication* is the property whereby one party is assured that no other party aside from a specifically identified second party (and possibly additional identified trusted parties) may gain access to a particular secret key.

Definition 2.8 : A protocol is said to have *perfect forward secrecy* if the compromise of long-term keys (i.e. passwords in this case) does not compromise past session keys.

Definition 2.9 : A protocol is said to be vulnerable to a *known-key* attack if the compromise of past (resp. future) session keys allows an adversary to compromise future (resp. past) session keys, or impersonation. We require a protocol to be invulnerable to this. The property is also sometimes referred to as *key independence*.

Definition 2.10 : *Resistance to offline guessing attack* is the property possessed by a password-based protocol if a guessing attack against the password by an adversary cannot succeed with more than negligible probability.

The script of a successful protocol run is usually assumed to be made available to such an adversary for an offline guessing attack. Resistance to offline guessing attack is a very fundamental security requirement for password-based security protocols; the attack is *passive* and cannot be detected online.

Definition 2.11 : *Resistance to online guessing attack* is the property possessed by a password-based protocol if a guessing attack against the password by an adversary who is an active participant in a protocol run cannot succeed with more than negligible probability.

It is normally the responsibility of higher layers to set policy and put in place mechanisms to trigger an alert if an online guessing attack by such an *active* adversary is detected. For example, the number of failed key establishment attempts can be tracked and a threshold set, before an alarm is set off, to indicate that an in-progress online guessing attack is suspected.

The success probabilities of the offline and online guessing attacks mean subtly different things. For the offline attack, success means being able to conduct a large enough search spanning the whole range of possible guesses, and being able to verify the correct guess against the recorded transcript of the protocol run. The attack is usually defeated because the space is too large to be computationally tractable for a probabilistic polynomial-time adversary, and/or the guess cannot be verified uniquely. For the online attack, success for the adversary means being able to participate in a protocol run and fool the victim into believing that the latter is communicating with a legitimate party. The attack is usually defeated if the attacker can be forced into making a one-shot random guess.

Other preferences for the protocol include efficiency measures such as low computational complexity and low message complexity.

We believe that asymmetric cryptography may be used to strengthen the pairing process, without incurring prohibitive performance degradation. To validate the feasibility of the approach, we implemented a single-machine simulation of the whole algorithm (with a single thread performing all the calculations that the two parties would in turn perform during a run of the protocol) and we ported it to a Bluetooth handphone. In our implementation, we are not performing the protocol over radio, because the API of the phone does not allow us to modify the Bluetooth stack, but we demonstrate that the processor in a modern handphone can perform the protocol with no prohibitive penalty in terms of time or energy.

In the Bluetooth protocol, the eavesdropper may brute-force the PIN offline and learn the session key. To defend against that type of attack, we use a variant of Encrypted Key Exchange (EKE). Regardless of the actual PIN, the eavesdropper cannot discover the session key, since it is established via Diffie-Hellman [62]. It is generally accepted that the Diffie-Hellman problem is a hard problem, because it is related to the hard discrete logarithm problem (DLP). The PIN instead is used to defeat active middleperson attacks and it cannot be brute-forced because the middleperson is detected at the first wrong guess.

We had initially also considered the threat of denial-of-service (DoS) attacks, which fool victim devices into running expensive asymmetric cryptography operations needlessly. Such attacks may perhaps be defended against with the use of client cryptographic puzzles [9; 102]. However, it was felt that the human operating the device would only activate the key agreement process if there is a partner device in view, and not otherwise, hence the DoS threat is not significant.

2.6.2 Use of Diffie-Hellman

EKE was introduced in 1992 by Bellovin and Meritt [25]. Thereafter, there have been a number of suggestions for password based authenticated key-exchange protocols. These include the schemes described in [33; 98; 99; 114; 207]. Many protocols use Diffie-Hellman as a basic building block.

We do have a choice of the domain over which to construct the Diffie-Hellman Problem (DHP). We will now briefly overview the number-theoretic aspects involved.

2.6.2.1 Diffie-Hellman over Multiplicative Groups

Typically, the Diffie-Hellman problem is set in a (multiplicatively written) finite cyclic group $\mathbb{G}$ of order n with a generator g. As a more concrete approach, it is possible to think of $\mathbb{G}$ as the multiplicative group $\mathbb{Z}_p^*$ of order $p-1$, in which p is prime, where the group operation is performed as multiplication modulo p [1]. We use the following definitions from Menezes et al [137].

Definition 2.12: The Diffie-Hellman Problem (DHP) [2] is the following: given a prime p, a generator g of $\mathbb{Z}_p^*$, and elements $g^a \pmod{p}$ and $g^b \pmod{p}$, find $g^{ab} \pmod{p}$.

The DHP is closely related to the discrete logarithm problem, though not proven exactly equivalent [132; 175]. The security of the Diffie-Hellman key exchange is based on the intractability of the discrete logarithm problem when p is sufficiently large.

Definition 2.13: The Discrete Logarithm Problem (DLP) is the following: given a prime p, a generator g of $\mathbb{Z}_p^*$, and an element $\beta \in \mathbb{Z}_p^*$, find the integer x, $0 \leq x \leq p-2$, such that $g^x \equiv \beta \pmod{p}$.

[1] The set of integers $\{\cdots,-2,-1,0,1,2,\cdots\}$ is denoted by the symbol $\mathbb{Z}$. The integers modulo n, denoted by $\mathbb{Z}_n$ is the set of (equivalence classes of) integers $\{0,1,2,\cdots,n-1\}$. The multiplicative group of $\mathbb{Z}_n$ is denoted by $\mathbb{Z}_n^* = \{i \in \mathbb{Z}_n \,|\, gcd(i,n) = 1\}$. In particular, if n is prime, then $\mathbb{Z}_n^* = \{i \,|\, 1 \leq i \leq n-1\}$.

[2] In fact, there is the Computational DHP and there is the Decisional DHP. The Decisional Diffie-Hellman assumption states that it is difficult to distinguish between the two probability distributions (g^a, g^b, g^{ab}) and (g^a, g^b, g^c), where g^c is also an element of the group, and this is a stronger assumption on computational hardness than the Decisional Diffie-Hellman assumption. But we would not need to get that precise in this dissertation, because we would be mainly referring to the Computational variant.

The discrete logarithm problem in *subgroups* of $\mathbb{Z}_p^*$ has attracted attention, due to its perceived difficulty. It is used in, for example, the NIST Digital Signature Algorithm, among others.

2.6.2.2 Diffie-Hellman over Elliptic Curve Groups

In order to mitigate the overall cost of the algorithm, instead of basing the asymmetric key agreement on the Discrete Logarithm Problem in a multiplicative group of a prime field (whether it is a traditional discrete logarithm system or a subgroup discrete logarithm system), we may consider it advantageous to base it on the Elliptic Curve Discrete Logarithm Problem (ECDLP). The use of elliptic curves in cryptography had been proposed independently by Neal Koblitz [108] and Victor S. Miller [140] in the mid-1980s.

The algorithms for solving the Discrete Logarithm Problem are classified into index-calculus methods and collision search methods. The latter have exponential running time. Index-calculus methods run at subexponential time, but these require certain arithmetic properties to be present in a group to be successful. The absence of such properties has led to the lack of any known index-calculus attack on elliptic curve discrete logarithms. The best general-purpose algorithm to solve the ECDLP has exponential running time. The current lack of subexponential-time algorithms to solve the ECDLP, as well as the development of efficient implementations of elliptic curve arithmetic, are two main reasons why ECDLP has become attractive on which to base cryptosystems.

We consider a finite field of q elements, written as $\mathbb{F}_q$. An elliptic curve E over $\mathbb{F}_q$ is defined in terms of the solutions to an equation in $\mathbb{F}_q$. We have used elliptic curves over a characteristic 2 field, $\mathbb{F}_{2^m}$ for some positive integer m. Further details of the choice of the elliptic curve group and the domain parameters used are given in Appendix A. G is a base point (x_G, y_G) of prime order n on the curve. $f(x)$ is the reduction polynomial, and $a, b \in \mathbb{F}_{2^m}$. The integer h is the cofactor. Elliptic curve domain parameters over a characteristic 2 field are a septuple [39]:

$$T = (m, f(x), a, b, G, n, h)$$

Informally speaking, the relation between the Diffie-Hellman Problem and the Discrete Logarithm problem for the elliptic curve setting is analogous to that for the multiplicative setting. For completeness, we outline these below.

Definition 2.14: The Elliptic Curve Discrete Logarithm Problem (ECDLP) is the following: Let E be an elliptic curve over a finite field $\mathbb{F}_q$; given P, a point on the curve, and a point $k.P$, which is in the group generated using P, find the integer k.

It is generally well-believed that the ECDLP is computationally hard to solve if G has large prime order.

Definition 2.15: The Elliptic Curve Diffie-Hellman Problem (ECDHP) is the following: Let E be an elliptic curve over a finite field $\mathbb{F}_q$; given P, a point on the curve, and the points $j.P$ and $k.P$, which are in the group generated using P, find the point $j.k.P$, where j and k are integers.

Based on the recommendations of Lenstra and Verheul [122], we may choose an elliptic key size which is equivalent to a symmetric key length of 86 bits (assuming no cryptanalytic progress), or equivalent to a symmetric key length of 79 bits (assuming cryptanalytic effectiveness doubles every 18 months, ie. $c = 18$ in Definition VI of [122]). Such a length is suggested to be 163 bits. This is roughly equivalent to 1024 bit length of RSA and DH. Some suggested domain parameters are given in SECG SEC 2 [40], and they are compliant with or recommended under ANSI X9.62, ANSI X9.63, IEEE P1363, IPSec and NIST FIPS 186.2, SP 800-56A and SP 800-57.

To get a more practical sense of the security offered, we note that the most recent ‘challenges’ to be solved under the ‘Certicom ECC Challenge’ [47] issued by Certicom Corporation, were the 109-bit challenges of ECCp-109 (over a prime field, solved in November 2002) and ECC2-109 (over a characteristic 2 field, solved in April 2004). The ECC2-109 effort required 2600 computers and took 17 months, which in terms of gross CPU time would be comparable to that of an Athlon XP 3200+ working nonstop for about 1200 years [46]; it was said that to solve 163-bit ECC, that would be ‘one hundred million times harder’.

2.6.2.3 Implementation

We thus implement a solution which is based on the AMP suite developed by Kwon [113; 114; 115], in particular AMP+. The password is entangled in such a way that an attacker who has no knowledge of the password is not able to mount such an active guessing attack, and also at the same time not able to compute the shared key formed between two genuine participating parties. The other advantages of this scheme include the following: its relatively good efficiency, its simple and understandable structure, it can be easily generalised, i.e. to elliptic curve (EC) groups, and a security proof is provided.

The protocol may be sketched as follows. The participants share a weak password P. They hash the password with the parties' identifiers (ie. their device addresses) to produce the password hash s, which is then multiplied with G, the common base point of the elliptic curve, to produce V. Alice sends her ephemeral public key $G.r_A$ [1]. Both Bob and Alice hash their identifiers with Alice's public key to obtain e_1. Bob multiplies Alice's public key by e_1, adds it to V, and multiplies this with Bob's private key r_B. This result is Bob's password-entangled public key, Q_B. Bob sends Q_B to Alice. Both parties would hash both of their identifiers and both parties' public keys together to obtain e_2. Alice computes ω, using her secret key r_A, the password hash s, and the values of e_1 and e_2 computed earlier. After obtaining ω, and using Bob's password-entangled public key Q_B, Alice is able to calculate $(r_A + e_2).r_B.G$ and derive the shared key. Over at Bob's end, he knows r_B, and using Alice's public key Q_A and the value of e_2 he

[1] Analogous to the case of multiplicative groups in finite fields, where r_A and g^{r_A} would be the private and public keys respectively, likewise for the EC case, given r_A finding $G.r_A$ is easy, while given $G.r_A$ finding r_A is hard.

has computed, Bob would likewise be able to calculate $(r_A + e_2).r_B.G$ and derive the shared key. The resulting protocol is shown in Fig. 2.4.

#	Alice		Bob
	$s = H_0(I_A, I_B, P)$		$s = H_0(I_A, I_B, P)$
	$V = G.s$		$V = G.s$
	Chooses random r_A		
	$Q_A = G.r_A$		
1		$\xrightarrow{Q_A}$	
	$e_1 = H_1(I_A, I_B, Q_A)$		$e_1 = H_1(I_A, I_B, Q_A)$
			Chooses random r_B
			$Q_B = (Q_A.e_1 + V).r_B$
2		$\xleftarrow{Q_B}$	
	$e_2 = H_2(I_A, I_B, Q_A, Q_B)$		$e_2 = H_2(I_A, I_B, Q_A, Q_B)$
	$\omega = (r_A e_1 + s)^{-1}(r_A + e_2)$		
	$K = H_3(h.Q_B.\omega)$		$K' = H_3(h.(Q_A + G.e_2).r_B)$
	$M_1 = H_4(I_A, I_B, Q_A, Q_B, K)$		
3		$\xrightarrow{M_1}$	
			$M_1' = H_4(I_A, I_B, Q_A, Q_B, K')$
			Verifies $M_1 = M_1'$
			$M_2 = H_5(I_A, I_B, Q_A, Q_B, K')$
4		$\xleftarrow{M_2}$	
	$M_2' = H_5(I_A, I_B, Q_A, Q_B, K)$		
	Verifies $M_2 = M_2'$		

Figure 2.4: Password-based Key Agreement (based on AMP+)

The security of the AMP family as secure authenticated key exchange protocols has been asserted by a formal proof [113] (with the caveats of security proofs [110]).

By inspecting the structure of the protocol, it can be seen that a passive eavesdropper would not able to compute the shared session key, unless he knows either r_A or r_B. This is the Diffie-Hellman number-theoretic problem considered hard. An active adversary, Eve, may attempt to carry out a protocol run (either full or partial) so as to obtain a trace to conduct a password guess (either online or offline). She does not have a high chance of success. If Eve attempts to masquerade as Alice, she has one chance at correctly guessing the password hash s, so as to calculate the correct K and subsequently send the correct M_1 to Bob. If she fails at one try, Bob will abort the protocol run without revealing more than one wrong guess. If Eve attempts to masquerade as Bob, and she contributes a password-entangled public key while not knowing the password hash s, even if she manages to collect an M_1 sent by Alice, Eve would need to solve the Diffie-Hellman problem to recover the corresponding r_B that would produce the same K solution which Alice has calculated.

Perfect forward secrecy in the protocol is provided through Diffie-Hellman, so an adversary is not able to calculate past session keys after knowing P or s. By the same token, there is resistance to the known-key attack, in which an adversary may attempt to attack the password or a previous session key, after knowing a session key (and also, there is resistance to the related Denning-Sacco attack[1] [60]).

The protocol resists the two-for-one guessing attack [129] by an adversary which masquerades as Bob. The idea behind this attack is that an attacker can validate two password guesses on one connection attempt. Earlier versions of AMP and SRP [207] were vulnerable to this slight deficiency. The improved AMP version resists this by doing a EC multiply (or exponentiation in discrete logarithm notation) of Q_A by e_1 to obtain Q_B, which breaks the symmetry which would otherwise exist between Q_A and V. Many-for-many guessing attacks, first raised by Kwon [115], particularly affect three-pass protocols. We do not think that many-for-many guessing attacks are too risky for ad-hoc devices though, since it is not expected at present that the devices would participate in more than a single password-based key agreement run at any one time. We however choose to use a four-pass protocol, and not a three-pass one, even if the latter is more efficient, in case of future feature creep where ad-hoc devices become more powerful and get called upon to behave more like server-type machines supporting multiple connection instances.

2.6.3 Key Derivation and Key Confirmation

2.6.3.1 Key Derivation

As recommended by IEEE 1363 Standard Specifications for Public Key Cryptography [95], a *key derivation function* is used because the output from an elliptic curve secret value derivation step may not always be appropriate to use directly as a session key, due to possible bias in some of its bits. A key derivation function based on some hash function will be able to utilize more effectively the entire entropy of the secret value. The hash also helps resist a known-key attack. The co-factor h is used to provide resistance to attacks like small subgroup attacks.

$$K = H_3(h.(r_A + e_2).r_B.G)$$

[1] The Denning-Sacco attack is an attack where the adversary compromises a session key from an eavesdropped session and uses the key to impersonate a user directly, as described in the paper [60], against the original Needham-Schroeder symmetric key protocol [145]. Timestamps were proposed as a defence against the attack, by preventing replays of a compromised session. The Denning-Sacco attack is sometimes also used to encompass the subsequent mounting of a dictionary attack against the password after a session key is compromised this way.

2.6.3.2 Key Confirmation

Definition 2.16 : *Key confirmation* is the property (or process) whereby one party is assured that a second party actually has possession of a particular secret key [137].

Key confirmation usually involves one party receiving a message from another party, containing evidence that the latter possesses the secret key. In practice, the possession of a secret key may be demonstrated by either showing a hash of the key, the use of the key in a keyed hash function, or encryption using the key.

In our case, we carry out a key confirmation procedure to ascertain that the parties know the password S and have established the same shared key K. It is bilateral. Including the identifiers of both parties adds explicitness and more defence-in-depth. The key confirmation functions H_4 and H_5 are hash functions. They may be differentiated by having different short strings pre-concatenated with their other hash inputs.

For subsequent mutual authentication between the pair of devices after they have established and confirmed the shared key, they may use the bilateral challenge-response steps similar to what is in Bluetooth's pre-existing specification. We use challenge-response with random nonces then instead of running the earlier key confirmation function again because freshness and resistance to replay attacks would then be necessary.

2.6.4 Implementation Results

We developed a demonstration program which implemented the entire key agreement protocol run described above. A 163-bit binary curve is used for the elliptic-curve cryptosystem. The hash function used is SHA-1[1]. Due to the difficulties of integrating this functionality described by the protocol into commercial Bluetooth devices, as changes are required at the baseband level of Bluetooth's protocol stack, we have not proceeded to implement this in a pair of Bluetooth demonstrator devices.

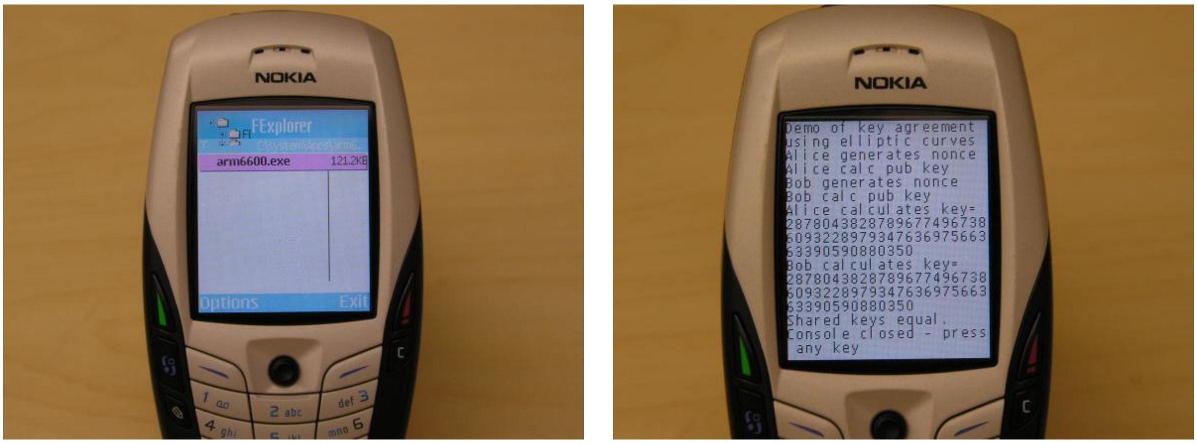
2.6.4.1 Performance - Laptop

The unoptimized simulation runs over a 1.2 GHz Pentium M laptop. Our program used routines from the Shamus MIRACL [126] cryptographic library to do the elliptic curve operations. On average, it took 3 milliseconds to perform an elliptic curve multiply operation (the most expensive single operation). As an upper bound, we consider that each of the computations of Q_B and K' requires 2 EC multiplies. The other operations take negligible time with respect to the public-key operations. The entire protocol run is completed in the order of time taken for 6 EC multiplies, which is 18 milliseconds on our platform.

[1] This choice may be re-evaluated in the light of recent attacks [193], though the usages of the hash functions in the protocol do not require random collision resistance [121].

2.6.4.2 Performance - Handphone

The software prototype was ported to a commercial handphone, a Nokia 6600 with a 104 MHz ARM processor and running the Symbian operating system [87]. The timing results proved that this phone can run the protocol without any speed problems. An EC multiply of the said order took merely 80 milliseconds. Thus, ignoring communication delays, all the computations of the entire protocol run may be completed in around half a second. If *V* can be assumed to have been pre-computed, there is a saving of 80 milliseconds. Note that these public key operations, while intensive, are only required for key agreement, which is usually done once-off between a pair of devices.



```
Demo of key agreement
using elliptic curves
Alice generates nonce
Alice calc pub key
Bob generates nonce
Bob calc pub key
Alice calculates key=
287804382878967749673
860932289793476369756
633390590880350
Bob calculates key=
287804382878967749673
860932289793476369756
633390590880350
Shared keys equal.
Console closed - press
 any key
```

(a) Console App on Phone

(b) Public-Key Protocol Run on Phone

Figure 2.5: Application on Symbian OS Nokia Phone

Having validated the protocol with a prototype implementation on actual handphone hardware, we suggest that asymmetric cryptography should no longer be axiomatically considered taboo for Bluetooth-class devices.

2.7 Comments on Provable Security

During these few years, the role of provable security in cryptography has come under particular scrutiny by the research community, including that by Koblitz and Menezes [109; 110], among others. Viewpoints are sometimes quite polarized in this area. Some of the criticisms against security proofs (more appropriately called security reductionist arguments) include: the proofs are difficult to read by non-specialists in cryptography who are nonetheless interested in applying cryptography, and that the security proofs are not as rigorously assessed as they ought to be and thus may possibly be incorrect.

To give a very summarized tour of the field, the Dolev and Yao paper [63], besides describing an all-powerful adversary, also first presented a 'formal model' of the adversary, where the cryptographic properties are often treated as a black-box (favoured by the formal methods community), but the model sometimes results in the loss of partial information. The computational complexity approach was developed soon after, from the work of Goldwasser and Micali [74], from the 1980s. In this model of security, the adversary is modelled as having polynomial computational power. One perspective regarding their differences is that the formal view enables higher-level logical reasoning, allowing even automated proofs, and is undoubtedly useful for increasingly complex systems, while the computational view makes explicit lower-level requirements on the concrete instantiations of protocols and primitives, helping give indications of the exact security parameter (for instance, key size) ranges needed [2]; both views are useful in their own right of course.

One of the milestones in the latter area is the work by Bellare and Rogaway [24] in the 1990s, which proposed computational complexity considerations for entity authentication, and provided a definition of adversary capabilities and an associated definition of security. Importantly, they advocated the influential concepts of ‘matching conversations’ and ‘(random) oracles’ to help determine whether a mutual authentication protocol is secure or not.

Bellovin and Merritt’s EKE protocols [25; 26] were later proven secure by Bellare et al. [22], who made a *random oracle* assumption. A protocol proven secure in such a proof is described as being secure in the random oracle model [23]. A random oracle is an oracle that responds to every query with a random response chosen uniformly from its output domain. However, it is only a mathematical abstraction, often used to model hash functions (with strong assumptions about these functions’ randomness). No real-world function can implement a random oracle. A protocol can alternatively be proven secure in the *standard model* (i.e. if no such strong randomness assumptions are used). Random oracle assumptions are sometimes problematic; there are schemes which are proven secure in the random oracle model but which are known to be insecure when a real-world function is used in place of the random oracle. Beginning from several years ago, researchers would tend to try to prove protocols secure in the standard model.

In addition to the discussions of Koblitz and Menezes [109; 110] Choo et al. [44] have described flaws in several protocols and their claimed security proofs. They in fact also described subtle differences in some well-accepted proof models for key agreement protocols, including some of the models that we have mentioned above (such as [22; 24]), which shows that the security proofs in the models are not exactly comparable.

It would seem that asserting and believing a ‘security proof’ for a protocol, ought to be positions reached only after exercising considerable care. We would also highlight that one should not place too much and unfair expectations on a ‘security proof’. A security proof (whether formal or computational or others) may, at best, confirm that the specific checked-for weaknesses are not present in a protocol or mechanism; in other words, the proof can only demonstrate that the result of finding particular weaknesses is negative. A security proof is unable to absolutely prove that a system is totally secure, especially against unchecked-for (or, as yet unknown) weaknesses, even though the security claims may be completely valid against the associated attacker models considered in the proof (and not to mention errors in the development of the mathematical or logical steps themselves); hence, the existence of a security proof should not be over-interpreted to imply the positive result that a system is totally secure.

(It should be noted that formal proof techniques have seen quite some success in the field of hardware verification.)

Taking these into account, while we implemented a protocol on a handphone after analyzing it heuristically, as described earlier in this chapter, we have not closely evaluated the security proof for correctness, though Kwon [113] had provided some security proofs for the AMP family of protocols;

2.8 Future Directions

Having established that asymmetric cryptography is affordable for a modern handphone, there is still the problem of simpler peripherals, such as Bluetooth headsets. It would be useful to develop an "asymmetric protocol", in which the more powerful device performs most of the computation. Another area that would need attention, assuming that the old protocol would be supported alongside the new for compatibility, is that of safeguards against the classic attack of persuading the devices to use the old less secure protocol even when they could both use the new one.

The Bluetooth SIG has initiated steps to rectify the security vulnerabilities of the current generation of Bluetooth devices, and has recently released a consultative whitepaper [30], in late 2006. This "Simple Pairing Whitepaper" proposes solutions which are user-friendly and intuitive, and not least of all, secure. The whitepaper embraces the use of asymmetric cryptography for key agreement in Bluetooth; this does validate the research work that we have done in 2004/2005 and have described in this chapter.

2.9 Summary

Our contributions:

- We implemented an actual attack against the Bluetooth PIN-based (or password-based) symmetric-key device pairing, showing that the attack is quite feasible. Our attack, though known to be possible, is the first implemented full attack against commercial Bluetooth devices in open literature.

- We proposed a stop-gap measure, namely, for paired devices to re-key often so as to update the link key, which would provide resistance against an attacker who is not continuously eavesdropping on them.

- We proposed that eventually password-based asymmetric key agreement would be quite essential to protect against the attack which we have mounted – we then implemented such a protocol, based on elliptic curve groups, on ubicomp devices such as smart phones. The latency/performance results are fairly good, showing that ubicomp device computational capabilities are rapidly making asymmetric cryptography feasible.

It was not unexpected the Bluetooth symmetric-key PIN-based key agreement protocol is vulnerable to a brute-force attack once the protocol trace could be sniffed wirelessly. Results indicate that 4-digit

PINs (i.e. the ones in the format to which people are most used from their experience with bank cards, car radios and indeed handphones) can be cracked under a second. Longer PINs cannot be considered secure either, since our non-optimised attack program cracks 8-digit ones in less than an hour. Even longer PINs, and especially PINs in which the characters were more varied than 0–9, would offer somewhat better protection; but it is only a matter of time before such incremental improvement is superseded by faster hardware on the attacker side.

Using asymmetric cryptography may somewhat exceed the design parameters, but it may be said that the original constraints are simply becoming outdated. There could still, though, be a legion of smaller Bluetooth-capable devices with much lesser computational capabilities and energy reserves than a smart phone. For the new generation of powerful devices (phones, PDAs, laptops) that are the most likely custodians of digital data that is worth more protection, stronger authentication via means of asymmetric cryptography, would be beneficial for use with not just Bluetooth, but also other pervasive wireless technologies, and this is becoming practical.

Chapter 3

Inter-Domain Password-Authenticated Identity-Based Key Agreement

3.1 Outline

Further on considering password-authenticated device *pairing* (i.e. the two-party case) which we have looked at in the previous chapter, we will consider inter-domain password-authenticated key agreement in this chapter. For this, we consider an approach slightly different from traditional ones; we make use of recent advances in identity-based cryptography (IBC) [31; 173]. Essentially, we will consider the case of inter-domain key exchange between pervasive computing devices having user input interfaces, so that passwords can be keyed in by human users, on demand.

Succinctly, the goal of an inter-domain authenticated key exchange protocol is to address cross-domain authentication and key establishment between two users registered under two distinct authentication servers. The computations involved is *multi-party* in character, unlike in the previous chapter.

As an example, let's suppose that each hospital has its own authentication domain, under which all its staff are registered. A medical consultant (i.e. entity A), working in Hospital X, visits Hospital Y carrying a PDA. He speaks to a surgeon (i.e. entity B) in Y, on his way from an operating theatre, carrying a PDA, and they decide they need to exchange some clinical information quickly. We assume a path exists for A to access his own authentication server S_A through Y's wireless network. The entities have not met *a priori*. They do not know whether the other is registered with an authentication server which their own authentication server recognizes. A needs to initiate a protocol, which when completed successfully, would indicate to A that B is properly registered by a password to his authentication server S_B, and that S_B is in fact a server that is recognized and trusted by S_A, and would also result in both parties sharing a fresh and authenticated session key.

In this chapter, we investigate the potential roles of identity-based cryptography (IBC) which can be exploited to overcome some security and usability issues. In particular, we [199] extend the recent proposal of identity-based secret public keys[1] (ID-SPK) by Lim and Paterson [123] to devise an identity-based four-party password authenticated key exchange (ID-4-PAKE) protocol. The concept of identity-based secret public keys, which was descended from Gong *et al.*'s work [76] on secret public keys, combines the use of passwords and identifiers in the IBC setting. Hence, an identity-based secret public key can only be constructed by a party who knows the associated password. We propose a solution which pays attention to both efficiency and security.

3.2 Related Work

Recent work, such as that by Yeh and Sun [210], reminds us of the relevance of inter-domain authentication protocols. They proposed two four-party password-based authenticated key establishment protocols, which are based on key transport and key agreement techniques, respectively. While the proposals attempt to address issues of inter-domain authentication, they suffer from some limitations. Firstly, their proposals were based on the assumption that the users have access to their respective authentication servers' public keys. This implies the need for a public key infrastructure (PKI) to distribute and verify the servers' public keys for the clients. This is a significant requirement for standard password-based authentication protocols which may be acceptable for certain networked applications, but less desirable for lightweight computing environments. Secondly, Yeh and Sun claimed that their protocols satisfy the property of forward secrecy. However, they have not taken the authentication servers' long-term private keys into consideration. The exposure of an authentication server's long-term private key could trivially reveal its users' passwords, and for their KTAP protocol (derived from the key transport [2] technique), even past session keys.

Kerberos [146] is another solution to inter-domain password-based authentication. It is known for its efficiency since it employs symmetric cryptographic techniques. However, purely symmetric key management for inter-domain secure communications is non-trivial and not scalable. In [212], a PKI-supported initial authentication in Kerberos was proposed to improve the scalability of Kerberos. However, deployment of PKI at the client side within lightweight environments is, again, not desirable.

Our proposal of an identity-based password authenticated inter-domain key agreement protocol for lightweight clients pays attention to efficiency as well as security. Ours is a somewhat novel application of identity-based cryptography. It requires only minimal communication bandwidth, because IBC is certificate-free, and small key sizes can be used. Our protocol requires users to remember only their respective passwords. It is PKI-free at the client side. It is convenient and user-friendly, as the clients do not have to obtain and verify public key certificates of their respective authentication servers, nor check for revocation of certificates of other clients. The domain servers can continue to use digital certificates. While the deployment of an identity-based cryptographic scheme generally requires distribution

[1] A secret public key is no different from a conventional public key except that it is only known among the intended parties.

[2] In contrast with key agreement (Definition 2.4), a *key transport* protocol or mechanism is a key establishment technique in which one party (or a subset of the participating parties) creates or otherwise obtains a secret value, and securely transfers it to the others(s).

of system parameters, and thus an infrastructure such as a PKI at the client side, is required for the users to authenticate these parameters, we will show that our protocol overcomes this requirement, i.e. a client-side PKI is not required in our protocol.

Unlike other proposals, which do not provide the property of forward secrecy, we show that it is possible to retain such a property in an inter-domain authenticated key exchange protocol. Thus, the compromise of a server's long-term secret does not reveal the user password nor past session keys.

3.3 Security Requirements

Our protocol proposal aims at fulfilling the following security objectives:

- **Mutual Authentication**: At successful conclusion of a protocol run, S_A and A have mutually authenticated each other; S_B and B have mutually authenticated each other; S_A and S_B have mutually authenticated each other. This is sometimes achieved by means of key confirmation (–see Section 2.6.3.2).

- **Resistance to Offline Guessing**: After a protocol run is made, a passive adversary who observed the traffic and collected the associated transcript is unable to guess the password(s) and verify the guess(es) with non-negligible probability of success.

- **Resistance to Online Guessing and Active Attacks**: An active adversary is one who can listen to, modify, delay, re-order, replay and insert traffic. The protocol is required to resist this adversary, such that this adversary cannot succeed in online password guesses, and he is unable to participate in a protocol run successfully and fool the participants into believing he is a legitimate party, nor succeed in other man-in-the-middle attacks by decrypting and reading traffic, with non-negligible probability.

- **Forward Secrecy**: If a domain server's master secret has been revealed to an adversary, the adversary should have only negligible probability of compromising domain users' passwords or past session keys.

- We also do want to defend against an attack whereby a domain server, say S_B, attempts either to impersonate a user A from *another* domain, to that user's domain server S_A, or to guess A's password. We define this malicious insider as a **weakly honest server** [1]. We will show later that our protocol addresses this subtle threat. This threat is related to mutual authentication, and active

[1] Some authors, such as Phan and Goi [155], use the term 'malicious server' to denote this. However, we feel that 'malicious server' would be less precise, and over-maligns the narrower type of attacker we have in mind. This is because if an insider server is totally malicious, such that it would even attempt to impersonate a user in this insider server's *own* domain to any entity outside that domain, then in fact we do not have protection against it – protecting against this is a 'non-goal'. Our 'weakly honest server' does not go quite so far in its attacks.

attacks. (Conversely, a strongly honest server is a completely well-behaved non-malicious server.)

The following is a non-goal:

- We do not address the unusual but not entirely inconceivable scenario whereby a domain server, say S_B, impersonates a user, say B, from S_B's *own* domain to another domain server (S_A) or user (A) from another domain. Such an attack seems impossible to defend against, if: (i) a server and a user share a password; and (ii) the server is an intermediary to the key agreement protocol (and (iii) there is no prior security association between the two users). We will make the reasonable assumption that a server is honest in this particular respect. (Thus, a server can be honest in this respect, and at the same time be 'weakly honest', as described earlier.)

3.4 Identity-Based Cryptography

Identity-based cryptography (IBC) was first introduced by Shamir [173]. It is a type of asymmetric key cryptography in which the public key can be any arbitrary string. Thus, the public key of a user can be obtained based on a unique and publicly available identifier – which may be the user's identity or email address. A trusted authority, called the private key generator (PKG), would generate the corresponding private key. This private key would be issued to the user U after the user proves his or her identity to the trusted authority. The attraction of IBC is that it removes the need for certificates, and allows an identifier to be associated with the public key to be used for encrypting a message. However, for many years after the proposal by Shamir [173], there did not appear to be easy and straightforward ways of developing an identity-based scheme.

In recent years, after the seminal discovery of a practical and secure identity-based encryption (IBE) scheme by Boneh and Franklin [31], there has been an increased intensity in research on IBC. Their scheme uses pairings over elliptic curves.

In what follows, we provide more details of pairings. We also sketch the Boneh and Franklin IBE scheme of [31], which will be used in our proposal.

3.4.1 Pairings

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of order q for some large prime q, where $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ denotes a related multiplicative group. A *pairing* in the context of IBC is a function $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties.

- *Bilinear*: Given $P, Q, R \in \mathbb{G}_1$, we have

$$\hat{e}(P, Q+R) = \hat{e}(P,Q) \cdot \hat{e}(P,R) \text{ and}$$

$$\hat{e}(P+Q, R) = \hat{e}(P,R) \cdot \hat{e}(Q,R).$$

  Hence, for any $a, b \in \mathbb{Z}$, $\hat{e}(aP, bQ) = \hat{e}(abP, Q) = \hat{e}(P, abQ) = \hat{e}(aP, Q)^b = \hat{e}(P,Q)^{ab}$.

- *Non-degenerate*: There exists a $P \in \mathbb{G}_1$ such that $\hat{e}(P,P) \neq 1$ and $\hat{e}(P,P) \in \mathbb{G}_2$.
- *Computable*: If $P,Q \in \mathbb{G}_1$, $\hat{e}(P,Q)$ can be efficiently computed.

For any $a \in \mathbb{Z}$ and $P \in \mathbb{G}_1$, we write aP as the scalar multiplication (or point multiplication) of group element P by integer a. Typically, $\mathbb{G}_1$ is obtained as a subgroup of the group of points on a suitable elliptic curve over a finite field, $\mathbb{G}_2$ is obtained from a related finite field, and $\hat{e}$ obtained from the Weil or Tate pairing on the curve. Note that a scalar multiplication aP can be computed very efficiently. However, the problem of finding abP when given P, aP and bP, is believed to be intractable, when the curve is appropriately chosen. This problem is known as the Elliptic Curve Diffie-Hellman (ECDH) Problem, as indicated in the previous chapter.

3.4.2 Boneh-Franklin IBE Scheme

The following four algorithms underpin Boneh and Franklin's IBE scheme [31].

SETUP**:** Given a security parameter $k \in \mathbb{Z}^+$, the algorithm:

1. specifies two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and a pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$;
2. chooses an arbitrary generator $P \in \mathbb{G}_1$;
3. defines four cryptographic hash functions, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$, $H_2 : \mathbb{G}_2^* \rightarrow \{0,1\}^n$ for some n, $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q^*$, and $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$; and
4. picks a master secret $s \in \mathbb{Z}_q^*$ at random and computes the matching public component as sP.

The system or public parameters are $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, sP, H_1, H_2, H_3, H_4 \rangle$.

EXTRACT**:** This algorithm extracts a private key $sH_1(\mathrm{ID})$ when given an arbitrary identifier string $\mathrm{ID} \in \{0,1\}^*$.

ENCRYPT**:** To encrypt a message $m \in \{0,1\}^n$ under an identifier ID, the public key used is $Q_{\mathrm{ID}} = H_1(\mathrm{ID})$. The algorithm selects a random $z \in \{0,1\}^n$ and sets $r = H_3(z,m)$. The resulting ciphertext is then set to be $c = \langle U,V,W \rangle = \langle rP, z \oplus H_2(g^r), m \oplus H_4(z) \rangle$, where $g = \hat{e}(Q_{\mathrm{ID}}, sP) \in \mathbb{G}_2$.

DECRYPT**:** To decrypt a ciphertext $c = \langle U,V,W \rangle$ encrypted using the identifier ID, the private key used is $sQ_{\mathrm{ID}} \in \mathbb{G}_1^*$. If $U \notin \mathbb{G}_1^*$, reject the ciphertext. The plaintext m is then recovered by performing the following steps:

1. compute $V \oplus H_2(\hat{e}(sQ_{\mathrm{ID}}, U)) = z$;
2. compute $W \oplus H_4(z) = m$; and
3. set $r = H_3(z,m)$, if $U \neq rP$, reject the ciphertext, otherwise accept m as the decryption of c.

The SETUP and EXTRACT algorithms are run by a PKG within a domain. The IBE scheme is secure against adaptive chosen ciphertext attacks (IND-ID-CCA) [31], provided the Bilinear Diffie-Hellman Problem is hard.

Definition 3.1: The Bilinear Diffie-Hellman Problem (BDHP) is the following: Given P, kP, lP and mP in $(\mathbb{G}_1, \mathbb{G}_2, \hat{e})$ for some $k, l, m \in \mathbb{Z}_q^*$, find $\hat{e}(P, P)^{klm}$.

As in all identity-based schemes and not just in the Boneh-Franklin IBE scheme, all the users within a domain are assumed to share the same system parameters, i.e. $\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, sP, H_1, H_2, H_3, H_4 \rangle$. In the identity-based setting, each PKG must distribute its parameter set to its users *a priori*. While most of the components of these parameters can be fixed and made public, and thus require no further authenticity verification, there exists a component, sP, which is mathematically tied to the PKG's master secret s. The failure of authenticating a PKG's parameter set generally could allow a trivial man-in-the-middle attack; but we will show that in our protocol, the server's public component does not need to be authenticated for resisting man-in-the-middle attack.

3.5 Architecture

Here, we describe the architecture and trust hierarchy that we employ in our proposal. We assume that all the system parameters used in the Boneh-Franklin IBE scheme ($\langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, H_1, H_2, H_3, H_4 \rangle$) except sP are fixed and bootstrapped in the system. All new users/devices are assumed to be initialized with these fixed parameters. This allows each authentication server to transmit only a server-specific value, i.e. sP, across the network (henceforth, we refer to a public component as a server-specific sP value). This represents a trade-off between savings in communication costs and lack of flexibility in supporting groups derived from different elliptic curves. The use of different curves and groups to achieve different levels of security is implementation-dependent, and thus will not be further discussed here.

Our protocol is based on an architecture of which the trust hierarchy consists of three tiers, as shown in Figure 3.1. We now briefly describe the key management aspect of our identity-based architecture.

3.5.1 Tier 1

At this tier, there exists a root PKG which owns a public component s_0P, of which s_0 is the corresponding master secret. The root PKG issues daily private keys to authentication servers at tier 2 using the EXTRACT algorithm of the Boneh-Franklin IBE scheme. These private keys correspond to public keys of the form $H_1(S_A \| \text{date})$ and $H_1(S_B \| \text{date})$ for authentication servers S_A and S_B, respectively.

3.5.2 Tier 2

As with a typical identity-based system, an authenticated copy of the root PKG public component, s_0P, is made available to the authentication servers, S_A and S_B, beforehand. If authenticity verification of the

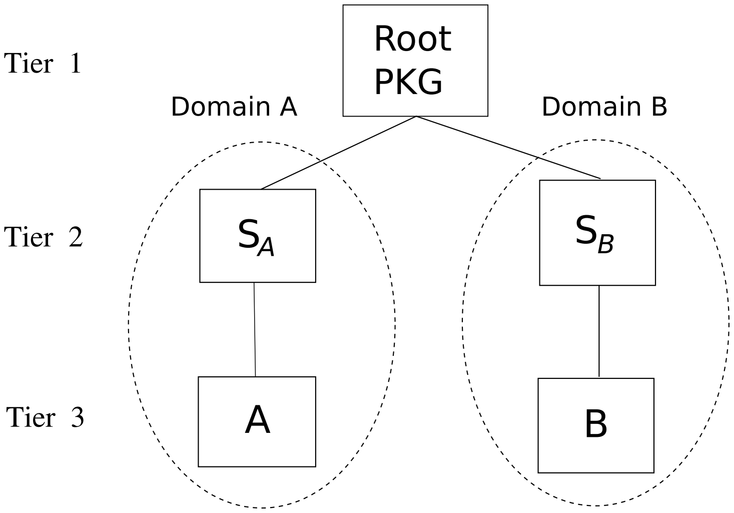


Figure 3.1: Architecture and trust hierarchy.

root PKG public component, and fine-grained (i.e. within the span of a day) revocation of the public keys of S_A and S_B are required, then an infrastructure, such as a PKI[1], would be clearly necessary at the domain server tier.

Each domain server[2] holds a list of the passwords of the users in its respective domain. The domain servers, i.e. S_A and S_B, also have another role: they each act as the domain PKG [43; 177] in their own domain, in that they own a master secret (s_A and s_B, respectively) which is used to extract certain decryption keys during a protocol run. Note that the associated server public components are s_AP and s_BP, respectively.

3.5.3 Tier 3

At the bottom tier, each user holds a password which he shares with his domain server. The way a password is defined and derived will be explained in Section 3.6.

3.6 Proposed Protocol

In our identity-based setting, a user A holds a low-entropy secret, the password PW_A and her authentication server S_A holds the matching image $PW_{S_A}[A]$, as defined in [22]. In our protocol, we assume $PW_{S_A}[A] = PW_A$, although they may be different in actual protocol implementations. We then set the transformed password as $\pi_A = H_1(A\|S_A\|PW_A)$, where H_1 is a full-domain hash function from $\{0,1\}^*$

[1] Standard revocation techniques such as Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) can be adopted in the identity-based setting.

[2] We will refer to 'domain servers' and 'authentication servers' interchangeably.

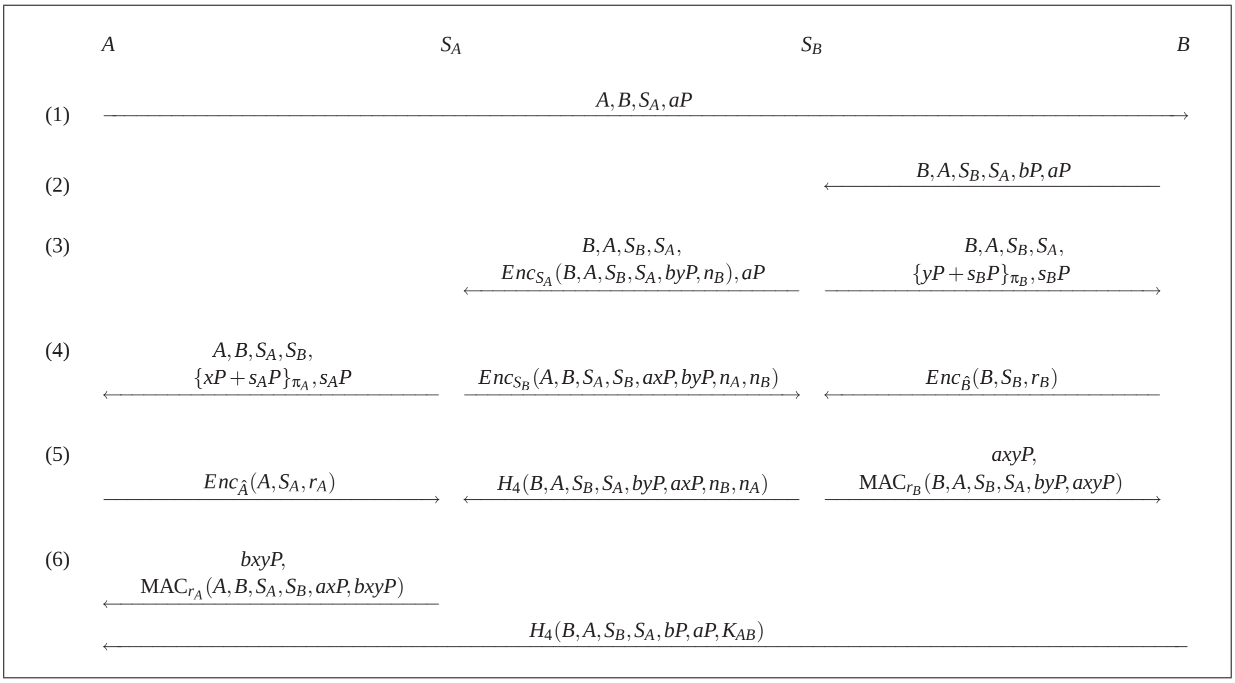

A
S_A
S_B
B
(1) A, B, S_A, aP
(2) B, A, S_B, S_A, bP, aP
(3) B, A, S_B, S_A, Enc_{S_A}(B, A, S_B, S_A, byP, n_B), aP
B, A, S_B, S_A, {yP + s_B P}_{π_B}, s_B P
(4) A, B, S_A, S_B, {xP + s_A P}_{π_A}, s_A P
Enc_{S_B}(A, B, S_A, S_B, axP, byP, n_A, n_B)
Enc_{B̂}(B, S_B, r_B)
(5) Enc_{Â}(A, S_A, r_A)
H_4(B, A, S_B, S_A, byP, axP, n_B, n_A)
axyP, MAC_{r_B}(B, A, S_B, S_A, byP, axyP)
(6) bxyP, MAC_{r_A}(A, B, S_A, S_B, axP, bxyP)
H_4(B, A, S_B, S_A, bP, aP, K_{AB})

Figure 3.2: ID-4-PAKE Protocol

into $\mathbb{G}_1^*$ (as defined in Section 3.4). We use $\{\cdot\}_{\pi_A}$ as a password-based mask generation function [22] under a password π_A (henceforth, we refer to a password as a transformed password using a full-domain hash of the password). For instance, $\{aP\}_{\pi_A}$ denotes encrypting a Diffie-Hellman value aP with a password π_A, which in turn, implies calculating the addition of aP and π_A. To decrypt and recover aP, one can simply subtract π_A from $\{aP\}_{\pi_A}$.

We use $\hat{PK}$ and PK to represent a secret public key [123] and a standard public key, respectively. A secret public key is no different from a conventional public key except that it is only known among the intended parties. We use the notation $Enc_A(\cdot)$ to indicate asymmetric encryption with A's public key and based on the Boneh-Franklin IBE scheme.

We are now ready to present our identity-based four-party password authenticated key exchange (ID-4-PAKE) protocol, as depicted in Fig. 3.2.

Our ID-4-PAKE protocol can be described as follows:

1. $A \rightarrow B : A, B, S_A, aP$
   To begin, A sends an initiating message to B. The message contains the identities of: (i) initiator, (ii) recipient, and (iii) initiator's authentication server. A also includes an ephemeral Diffie-Hellman value aP, where $a \in \mathbb{Z}_q^*$ is a randomly selected secret value.

2. $B \rightarrow S_B : B, A, S_B, S_A, bP, aP$
   In step (2), upon receiving the initiating message from A, B randomly selects a secret value $b \in \mathbb{Z}_q^*$

and computes his Diffie-Hellman value bP. B then forwards this value and the original message that he received from A to his authentication server S_B.

3. $S_B \rightarrow S_A : B, A, S_B, S_A, Enc_{S_A}(B, A, S_B, S_A, byP, n_B), aP$
When S_B receives the message in step 2 from B, it identifies the intended communicating target (A) and the corresponding authentication server (S_A). Subsequently, S_B randomly chooses a secret value $y \in \mathbb{Z}_q^*$ and computes byP. S_B also chooses a nonce n_B. The values of byP and n_B, and the identities of A, B, S_A and S_B are then encrypted using a public key computed from the date and S_A's identifier. The resulting ciphertext and other information, such as S_B's identity and A's chosen Diffie-Hellman value aP, are sent to S_A.

   $S_B \rightarrow B : B, A, S_B, S_A, \{yP + s_BP\}_{\pi_B}, s_BP$
In parallel with[1] the previous message from S_B to S_A, S_B computes its Diffie-Hellman value yP which is then sent to B along with S_B's public component s_BP. Note that yP is added to s_BP, and encrypted under B's password π_B because the Diffie-Hellman value will be used later for both S_B and B to authenticate each other. The rationale for adding yP and s_BP before their sum is encrypted using π_B is to resist active insider attackers; this will become clearer in Section 3.7.

4. $S_A \rightarrow S_B : Enc_{S_B}(A, B, S_A, S_B, axP, byP, n_A, n_B)$
As with what S_B did in the previous step, S_A randomly selects a secret value $x \in \mathbb{Z}_q^*$, and then computes a composite Diffie-Hellman value axP. S_A selects a nonce n_A. The message $(A, B, S_A, S_B, axP, byP, n_A, n_B)$, encrypted under S_B's daily public key, is forwarded to S_B. Note that S_A includes the Diffie-Hellman value byP and the nonce n_B, in the message to authenticate itself to S_B.

   $S_A \rightarrow A : A, B, S_A, S_B, \{xP + s_AP\}_{\pi_A}, s_AP$
At the same time, S_A computes its Diffie-Hellman value xP. The value xP is added to s_AP, and transmitted to A encrypted with A's password π_A. Other information such as S_B's identity and S_A's public component s_AP is also included in the transmission.

   $B \rightarrow S_B : Enc_{\hat{B}}(B, S_B, r_B)$
On the other hand, B recovers yP using his password and subtracting s_BP, and computes the composite Diffie-Hellman value byP, which in turn is used to calculate a secret public key $\hat{PK}_B = H_1(B\|A\|\pi_B\|S_B\|S_A\|byP)$. This secret public key is then used to encrypt the identities of B and S_B, and the chosen random nonce r_B, and produce a ciphertext which could only be decrypted by a party who can extract the matching private key of $\hat{PK}_B$.

5. $A \rightarrow S_A : Enc_{\hat{A}}(A, S_A, r_A)$
In this step, A encrypts a message that contains the identities of A and S_A, and a fresh random number r_A, with a secret public key $\hat{PK}_A = H_1(A\|B\|\pi_A\|S_A\|S_B\|axP)$. Note that $\hat{PK}_A$ can be computed by A only after she has successfully recovered xP coming from S_A.

[1] It makes sense that once y has been chosen, S_B can produce and send the relevant messages to S_A and B simultaneously.

$S_B \rightarrow S_A : H_4(S_B, S_A, byP, axP, n_B, n_A)$
This hash value is generated by S_B to authenticate itself to S_A by proving to S_A that it has recovered the Diffie-Hellman value axP and the nonce n_A successfully. See Section 3.4.2 for the definition of H_4.

$S_B \rightarrow B : axyP, \mathrm{MAC}_{r_B}(B, A, S_B, S_A, byP, axyP)$
Here, S_B decrypts the ciphertext from S_A in step (4) and recovers axP. It then calculates a composite Diffie-Hellman value $axyP$. Additionally, S_B also generates a MAC value by taking as input r_B and the message $(B, A, S_B, S_A, byP, axyP)$. The $axyP$ value and the MAC value would be sent to B.

6. $S_A \rightarrow A : bxyP, \mathrm{MAC}_{r_A}(A, B, S_A, S_B, axP, bxyP)$
In the final step, analogous to the message from S_B to B in the previous step, S_A computes the relevant composite Diffie-Hellman value $bxyP$. The value of $bxyP$ and a MAC value derived from the relevant information, as specified above, are transmitted to A. The session key shared between A and B is $K_{AB} = F(A, B, S_A, S_B, abxyP)$, where F is a key derivation function.

   $B \rightarrow A : H_4(B, A, S_B, S_A, bP, aP, K_{AB})$
   The above hash value is computed by B and sent to A to provide key confirmation. This signifies the completion of a successful run of the protocol.

3.7 Security Analysis

We present heuristic security analyses of the ID-4-PAKE protocol.

Mutual Authentication. In the protocol, each party contributes a Diffie-Hellman component for the generation of a session key K_{AB}. The Diffie-Hellman values chosen by the servers, xP and yP, are added to the respective servers' public components, $s_A P$ and $s_B P$, and encrypted under the users' passwords, π_A and π_B, respectively. If S_A can successfully decrypt the ciphertext $Enc_{\hat{A}}(A, S_A, r_A)$ such that the identities of A and S_A are revealed in the resulting plaintext, A is authenticated to S_A. This is because A can only construct the correct $\hat{PK}_A = H_1(A\|B\|\pi_A\|S_A\|S_B\|axP)$ if she could recover the right xP from S_A using her password π_A, and thus generate the proper ciphertext for S_A.

On the other hand, S_A is authenticated to A if A can derive the same MAC value as what she received from S_A. This indicates that S_A has successfully extracted the matching private key of $\hat{PK}_A$ using its master secret s_A and subsequently recovered r_A chosen by A.

In a similar fashion between A and S_A, B and S_B authenticate each other using similar techniques.

What happens if an adversary E attempts to claim to be the legitimate server S_A to A, by generating any random fake s'_A and then sending $s'_A P$ to A? This attack would fail against our protocol, because E does not know the password and is unable to guess it correctly with high probability. More discussion is given in the later part of this section on 'Active Attacks and Online Guessing'.

The mutual authentication between S_A and S_B is straightforward. In step (3), S_B sends $Enc_{S_A}(B, A, S_B, S_A, byP, n_B)$ to S_A, encrypted under $PK_{S_A} = H_1(S_A\|date)$. The corresponding decryption private key has been obtained by S_A from the Root PKG at the start of each day. Then S_A decrypts the contents and recovers byP and n_B, which it would then encrypt together with axP and n_A and send to S_B in step (4). If S_B recovers byP and n_B successfully by decrypting the message, S_A is authenticated to S_B. In a similar way, when S_A receives the hash from S_B in step (5) and is able to compute the same hash, it proves that S_B has decrypted the preceding message from S_A, and S_B is authenticated to S_A.

We next show that each client participant, even if he is not completely honest, has negligible probability of successfully compromising the mutual authentication between the two servers. A (resp. B) cannot impersonate successfully as S_A (resp. S_B) to S_B (resp. S_A), because A (resp. B) does not have S_A's (resp. S_B's) decryption key, so does not know know the value of n_B (resp. n_A), and does not receive the value of byP (resp. axP) at any stage. B (resp. A) cannot impersonate successfully as S_A (resp. S_B) to S_B (resp. S_A) either, because B (resp. A) does not have S_A's (resp. S_B's) decryption key, so does not know the value of n_B (resp. n_A), even though he can calculate the value of byP (resp. axP).

We remark that the last message in step (6) from B to A is essential to confirm that B has authenticated himself to S_B and that he has calculated the same session key as A. This is because B would only receive the value of $axyP$ from S_B after he is authenticated to S_B, which will enable him to calculate the session key. As for A, she would receive the value of $bxyP$ from S_A after she has been authenticated to S_A, which will allow her to calculate the same session key and verify B's key confirmation message.

Offline Guessing. An adversary E cannot deduce any useful information by attempting to decrypt $\{xP + s_AP\}_{\pi_A}$ (resp. $\{yP + s_BP\}_{\pi_B}$) with a guessed password π'_A (resp. π'_B) and then subtract by s_AP (resp. s_BP). This is because the use of any candidate password will result in a random point in $\mathbb{G}_1$. Similarly, since the Boneh-Franklin IBE scheme is probabilistic and secure against adaptive chosen ciphertext attacks (IND-ID-CCA) [1] [31], E cannot learn any useful information from the ciphertext produced in the protocol.

Active Attacks and Online Guessing. We observe that even though the servers' public components s_AP and s_BP are sent in the clear and unauthenticated, E cannot mount man-in-the-middle attacks by impersonating S_A or S_B. Suppose E tries to impersonate S_A by replacing the message $(A, B, S_A, S_B, \{xP + s_AP\}_{\pi_A}, s_AP)$ with $(A, B, S_A, S_B, \{x'P + s'_AP\}_{\pi'_A}, s'_AP)$, of which the master secret s'_A and the value $x'P$ are known to E, and π'_A is a guessed password from E's password dictionary. However, E cannot predict, in polynomial time, $\hat{PK}_A$ that A computes and thus extract the corresponding private key. The reason for this is that, assuming A recovers a Diffie-Hellman value $x''P$ with the correct password π_A, the only way for E to correctly predict the value x'' (in order to compute $ax''P$) is equivalent to solving the ECDL problem.

[1] Some well-accepted notions of security exist for asymmetric encryption schemes, and three such notions, in order of increasing security strength, are: indistinguishability under chosen plaintext attack (IND-CPA), indistinguishability under non-adaptive chosen ciphertext attack (IND-CCA1), and indistinguishability under adaptive chosen ciphertext attack (IND-CCA2). Security under a latter definition also implies security under the previous one(s) [19]. The concept of 'indistinguishability' formalizes the adversary's (in)ability to learn any information about the plaintext underlying a ciphertext.

Forward Secrecy. Based on similar reasoning as the previous, even if S_A's master secret (s_A) is exposed, the probability of guessing the correct password (π_A) or recovering a past session key appears to be negligible. The adversary is unable to verify a password guess because decrypting by any guess will result in a random point in $\mathbb{G}_1$. In trying to calculate a past session key, the adversary is hindered by his lack of knowledge of any past ephemeral Diffie-Hellman private values, which contributed to that session key. Thus we conjecture that the ID-4-PAKE Protocol in Fig. 3.2 has the property of forward secrecy.

Insider Attacks by Weakly Honest Servers. Note that in the case of a weakly honest server (as defined in Section 4), say, if S_B is weakly honest, three related attacks are conceivable. In the first attack, S_B attempts to guess the password π_A. S_B swaps the s_AP value which S_A sends to A in cleartext in step (4) with his own chosen s'_AP (where he knows s'_A). A will be now manipulated to calculate a secret public key of the form $\hat{PK}'_A = H_1(A\|B\|\pi_A\|S_A\|S_B\|a(xP + s_AP - s'_AP))$, and to encrypt (A, S_A, r_A) under this key. Can S_B extract the corresponding decryption key with high probability by brute-forcing the password, since he holds the master secret s'_A? S_B has received axP from S_A in step (4), and he also knows $(s_AP - s'_AP)$. Yet he remains unable to construct the secret public key $\hat{PK}'_A$ because he is unable to obtain the value of $a(s_AP - s'_AP)$ to calculate $(axP + a(s_AP - s'_AP))$ [1]. The difficulty is equivalent to solving the ECDL problem. This attack can only succeed with negligible probability.

In the second attack, S_B attempts to impersonate A to S_A. Differing slightly from the first attack, here S_B allows the message which S_A sends to A in step (4) to proceed unmodified. Again, from the message which S_A sends simultaneously to S_B, S_B can recover axP. S_B intercepts the message from A to S_A in step (5), and attempts to substitute it with his own message. Can S_B construct the secret public key $\hat{PK}_A = H_1(A\|B\|\pi_A\|S_A\|S_B\|axP)$ for impersonating A? As S_B does not know π_A, he has a negligible probability of finding the correct password in one online guessing attempt. (In this attack, S_B does not hold the master secret s_A.) Thus, this second attack can only succeed with negligible probability.

In the third attack, S_B attempts to perform an offline dictionary attack of the password π_A after having obtained the transcript of a successful protocol run. This attack is similar to the second attack, except that S_B does not attempt to impersonate A with an online guess in this case. Again, S_B knows the value of axP. Can S_B find the correct password π_A and calculate the correct secret public key $\hat{PK}_A$? As S_B does not hold the master secret s_A, he cannot extract a corresponding decryption key to verify a guess. Without holding s_A, it can be argued that S_B may attempt to exhaustively construct different secret public keys derived from different password guesses, to use for encrypting (A, S_A, r'_A), where r'_A is a guess of r_A, and try to match that with the ciphertext which had been captured, but this attack, again, has a negligible probability of success, because the entropy of r_A is assumed to be high.

Likewise, S_A instead of S_B may assume the weakly honest server role, and similar attacks mounted by S_A are also resisted by the protocol.

[1] No opportunity exists in the protocol run for a weakly honest S_B to fool A into becoming an oracle to exponentiate some value by a and then return it (to S_B).

3.8 Comparisons

In this section, we compare our protocol to several related protocols in terms of protocol message round complexity, computational costs, and infrastructure requirements. We give an overview of these protocols.

3.8.1 PKI-Kerberos

Kerberos can be used to achieve cross-realm authentication (PKCROSS) by using public key cryptographic techniques. The messages exchanged between two Key Distribution Centres (KDCs) closely follow the PKINIT specification [212]. The PKCROSS approach utilises a PKI to simplify the administrative burden of maintaining cross-realm keys.

The basic operations of PKCROSS are as follows:

1. The client submits a signed request to the local KDC for credentials for the remote realm.

2. The local KDC submits a signed PKINIT request to the remote KDC to obtain a special PKCROSS ticket. This is a standard PKINIT request.

3. The remote KDC responds as per PKINIT, except that the ticket contains policy information, such as lifetime of cross realm tickets, issued by a remote KDC to a client. The local KDC must reflect this policy information in the credentials it forwards to the client.

4. The local KDC passes a cross-realm Ticket Granting Ticket (TGT), between the client and remote KDC, to the client. This ticket contains the cross-realm key.

5. The client submits the request directly to the remote KDC and proceeds with the standard symmetric crytographic techniques for Kerberos [111].

In short, PKCROSS extends the public key cryptographic techniques to cross-realm KDC-to-KDC authentication: this is analogous to our ID-4-PAKE protocol.

We remark that if a KDC's private key is compromised, then past keying material is exposed; thus PKCROSS does not achieve our definition of perfect forward secrecy.

3.8.2 Yeh-Sun KAAP/KTAP

In [210], two protocols were proposed – a key transport version (KTAP) and a key agreement version (KAAP); we are primarily concerned with the latter. Like the PKI-Kerberos, the Yeh-Sun proposals require the clients to obtain the servers' static public keys, and hence a PKI which which interacts directly with the clients is again implied.

We reproduce here a client-to-client message from YS-KAAP, where P_{S_A} refers to a traditional public key of the server S_A:

$$A \rightarrow B : S_A, \{A, B, r_A, \pi_A\}_{P_{S_A}}$$

Referring to the message, clearly, if the private key of S_A is compromised, then the password π_A can be found easily. Thus, the protocol does not actually fully satisfy the property of perfect forward secrecy. This is because the password is just encrypted with the public key. This is contrasted to our ID-4-PAKE scheme.

3.8.3 Three 2-party Key Agreements

We consider a straightforward protocol made up of three 2-party key agreements. It contains two 2-party password-authenticated key agreements using servers' static public keys, and one server-to-server 2-party key agreement. Surveying the literature on 2-party password-authenticated key agreement protocol [96], we see that the most efficient ones have the minimum of three message rounds. As we want to benchmark our ID-4-PAKE scheme against the most round-efficient scheme, we will use as a building block a protocol having three message rounds, such as the Halevi-Krawczyk scheme [86] (HK-PAKE), which is provably secure. Proceeding in a straightforward and naive way, we arrive at a scheme with the flow shown in the Fig. 3.3 (– message details omitted for brevity).

Round 1 includes one message which the initiator client uses to notify the other client. After Round 3, the first 2-party password-authenticated key establishment (AKE1) is completed. The second 2-party key establishment (AKE2) starts one round later than the first, and ends after Round 4. The key agreement and authentication between the two servers (AKE3), is completed at the end of Round 6. HK-PAKE makes use of a concept which the authors termed 'public password', which is a pre-distributed hash of the server's static public key for quick verification, but a client-end PKI is nevertheless required in their architecture. For AKE3, both servers possess static public keys – thus two encryptions and two decryptions would conceivably be used. The establishment of the shared key between A and B and key confirmation (KC) is completed at Round 8. The message round complexity figure of 8 which we arrived at with the flow in the protocol shown in Fig. 3.3 is also in correspondence with that arrived at by a suggested 3-party HK-PAKE (3-HK-PAKE) composition hinted at by Yeh and Sun [210].

Like the Yeh-Sun protocols, the HK-PAKE [86] protocol suffers from a similar security weakness. Reproducing the relevant flows below, where P is the group generator, a and x are Diffie-Hellman private values generated by A and S_A respectively, PK_{S_A} is S_A's static public key, r is a random nonce generated by S_A, k is MAC key randomly chosen by A, π_A is the password, PRF is a pseudorandom function, and using elliptic-curve notation:

$$\begin{array}{rcl} S_A \rightarrow A & : & r, xP, PK_{S_A} \\ A \ \ calculates \ \ T & = & PRF(\pi_A, r, xP, aP, k, A, S_A) \\ A \rightarrow S_A & : & aP, Enc_{PK_{S_A}}(k, T) \end{array}$$

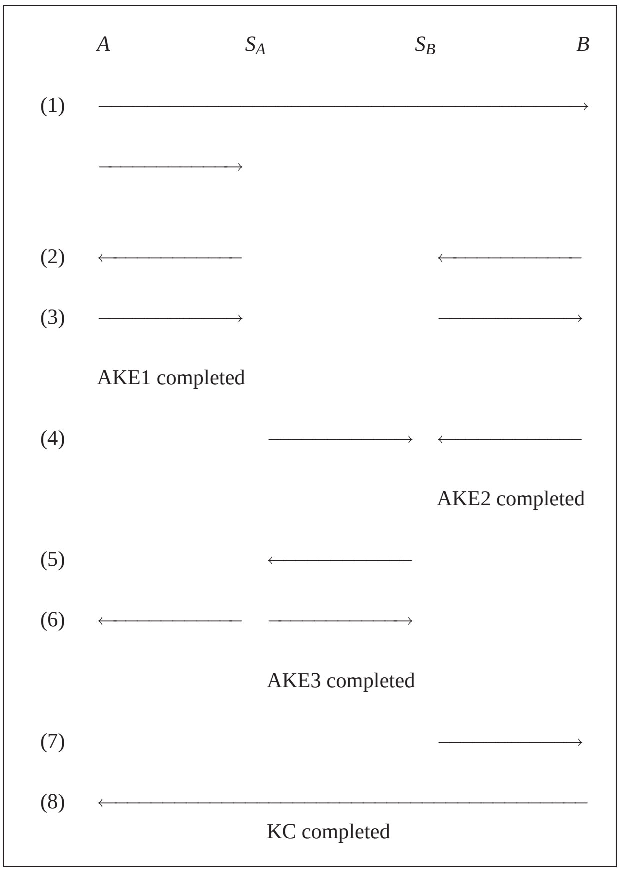


Figure 3.3: Three 2-Party AKE

If the server's long-term private key becomes compromised, then the user's password π_A is exposed to dictionary attack: the protocol does not actually have perfect forward secrecy, according to our definition. We remark that the security proofs provided for the protocol do not seem to have taken this into account.

Independent of the 3-HK-PAKE composition, we remark that it is possible to imagine a composed three 2-party key agreement protocol in which the two client-to-server key agreements are mediated by ephemeral public keys, and not static (authenticated) public keys. We have not encountered such a protocol proposal before, which suggests that the research community may not have paid much attention to this area. While such a composition would confer the benefit of certificate-free operation at the client end, however, we note that its server-to-server key agreement would still need to rely on servers' authenticated public keys — implying an infrastructure would nevertheless be required at the server level. We conjecture that such a composition would require at least the same number of message rounds as the composed 3-HK-PAKE protocol, when derived in a similarly straightforward way.

3.8.4 Comparison Metrics

We use various metrics of performance, including metrics of complexity. We elaborate on a few comparison metrics below.

Number of message rounds. This is the number of message rounds for a protocol to run successfully. Independent messages may be interleaved; dependent messages in a protocol are sequenced accordingly. It is a measure of the latency of a protocol.

Number of asymmetric encryptions/decryptions/signings/verifications. Such asymmetric operations, using public or private keys, consume significant computational resources. Within our ID-4-PAKE protocol, these operations refer to both the operations using standard public keys and secret public keys. The dominant cost for an identity-based cryptographic scheme is the evaluation of a pairing. Although, generally, computing a pairing is slower than performing a modular exponentiation, it is perceived to be acceptably fast to implement in practice. Implementations of the Boneh-Franklin IBE scheme on smart cards by Gemplus is a good example [135]. Recent results (see for example [17; 170]) have shown improvements in computing pairings with the use of various optimisation techniques and this should give hope to faster IBE schemes in the near future. Here, to a first approximation, it seems reasonable to consider identity-based asymmetric cryptographic operations to be comparable to traditional asymmetric operations.

Communication complexity at client end. We do not quantify this metric precisely, because it is dependent on the message format, as well as group/order sizes and the required security parameters. We assume that the transmission of a digital certificate makes the dominant contribution to communication complexity during the protocol run; in comparison, transmission of an unauthenticated public key is less complex.

PKI-free at client end. This is an important comparison. If a protocol uses a PKI at the client end, this entails the obtaining of certificates, and the tedious certificate verifications, in the background. These processes are not accounted for by the message round complexity metric for the protocol, nor the metric for communication complexity at the client end per protocol run.

A comparison of the relative performance of the protocols is presented in Table 3.4. Our protocol is comparable with (and sometimes superior to) the existing protocols in efficiency terms. In terms of computational complexity, our protocol requires, for example, only one asymmetric encryption computation at each client, which is comparable to YS-KAAP and 3-HK-PAKE. In terms of communication complexity at the client end, our ID-4-PAKE protocol is a good performer, since no certificates are required.

Based on these comparisons, we believe we have demonstrated that our protocol proposal is viable in terms of overall performance (i.e. security and efficiency).

Performance/Protocol	**PKI-Kerberos**	**3-HK-PAKE**	**YS-KAAP**	**ID-4-PAKE**
# message rounds	8	8	6	6
PKI-free at client end	No	No	No	Yes
# asymmetric enc/dec/sig/ver	12	8	6	8
# asymmetric enc/dec/sig/ver per client	3	1	1	1
Communication complexity at client	High	Low	Low	Low
Perfect forward secrecy	No	No	No	Yes

Table 3.4: Performance Comparison

3.9 Future Directions

For future work, we would consider how to reduce the protocol's message complexity. Beyond this dissertation, we believe there is scope to continue work to now consider a more formal security analysis of the protocol; if the conceivable attacks can be modelled well in such a more rigorous framework, that will improve confidence in the security of the protocol, even if that admittedly does not constitute an ultimate proof of security.

Without a doubt, there is much on-going work in the research community to apply identity-based cryptography to many interesting problems. We believe that identity-based cryptography could merit further study in password-based key agreements.

3.10 Summary

Our contributions:

- We have proposed a password-authenticated protocol for inter-domain key agreement using identity-based cryptography and the concept of secret public keys.
- In our proposed design, there is no need for a client-side PKI for the purpose of authenticating domain parameters (even though authenticating domain parameters is usually required in identity-based cryptography). Our design cleverly allows the clients to make use of the unauthenticated domain parameters to do identity-based encryption.
- We have carried out heuristic security analysis of the protocol, and it is shown that the protocol is secure against various types of passive as well as active attacks. Our performance comparisons show that our proposal is reasonably efficient.

The area of inter-domain authentication protocols is under-researched. We envisage that pervasive devices which are registered under different authentication domains may need to participate in transactions, and our proposal, leveraging on password use and recent advances in identity-based cryptography, is a response to this need. We outlined a likely architecture for this inter-domain scenario. It is noteworthy that we are able to do away with client-side PKI, hence saving on the inconveniences of obtaining a certificate and the associated key management issues (such as protection and revocation of keys) in our protocol proposal. Our comparisons with related protocols have revealed that the proposed protocol is reasonably efficient and quite viable, and it certainly appears that there is more scope for research in this area.

Chapter 4

Multi-Channel Security Protocols

4.1 Outline

Ubiquitous computing environments provide new means and models for different devices to interact with one another. Compared to a traditional wireline networking model, in which signals are likened to travel down a bunch of wires, or else a completely wireless model, in which signals are completely of a broadcast nature, in a ubiquitous environment many different types of channels are brought into play. Such channels include visible light, infra-red, audio, laser, metal contact, ultrasound channels, and even including mere physical touch. (We will give more precise examples of these channels in later sub-sections.) Often, these *auxiliary* channels are *human-mediated*, whereby some human operator is directly operating and supervising the channel in question.

The main theses of this chapter include: the existence of such *auxiliary* channels may be utilized in key establishment security protocols which explicitly take into account the channels' different properties, with benefits gained in terms of both security and usability. Secondly, being explicit about the properties of the channel over which each message of a protocol is transmitted is critical for understanding the protocol in greater depth and addressing subtle vulnerabilities early on. Actually, *multi-channel security protocols* have existed for a long time before we recognized them as such, for example the PGP fingerprint is a component in a multi-channel protocol, in which the fingerprint is attested over a more authentic channel than the channel used to transmit the PGP public key itself. There is much to gain by adopting the multi-channel viewpoint: it forces us to be more precise about the security requirements, the channel properties, and the attacker model. This helps avoid design flaws.

Multi-channel protocols are particularly relevant for ubicomp because it is precisely in this scenario that multiple heterogeneous communication channels are naturally found—as opposed to, for example, the comparatively more uniform scenario offered by the packet-switched Internet. One of Weiser's seminal papers [197] about ubicomp asked the question: "Can the device communicate simultaneously along multiple channels?".

Further, a particular property of any auxiliary channel which concerns us is *data-origin authenticity*. This property is a weaker security property than confidentiality, which is possessed by a password channel in the preceding chapter.

4.2 Related Work

An *auxiliary* channel is of course, *auxiliary* to some other main channel (which may be wired or, in ubicomp, wireless radio), the latter of which is used to carry the bulk of the information in a session. Compared to the main channel, an auxiliary channel usually possesses stronger properties which are difficult to assure for the main channel.

Different researchers have used several kinds of auxiliary channels, using a variety of terms that emphasize specific properties of interest such as “location-limited”, “out-of-band”, “empirical” and so forth. Balfanz et al. [15] use a “location-limited” channel to commit to a hash of a public key, similar in function to PGP’s concept of key fingerprints. This kind of protocol becomes vulnerable, however, if for example, the hash truncated to fit the capacity of the auxiliary channel is too short, as would be shown in Protocol Trace 4.2 in Section 4.3.3.1. Hoepman [92] distinguishes between “private” (i.e. confidential) and “authentic” channels; he proves, with his φKE protocol, that authenticity is sufficient for a secure bootstrap of ephemeral Diffie-Hellman key agreement. McCune et al. [136] implement a “visual channel” based on 2D visual codes printed on labels or shown on displays and then acquired by camera phones. Vaudenay [190] describes extractable and equivocable commitment schemes and also provides informal notions of stronger authenticity in channels, such as stall-freeness and listener-readiness. Cagalj et al. [191] propose three protocols based on visual and verbal interaction between the participants. Laur et al. [118; 119; 120], building on earlier proposals of Gehrmann et al. [70], describe a round-efficient two-party protocol, MANA IV, and provide suggestions for practical constructions of the commitment schemes. The researchers used different notations to signify different types of channels.

In terms of security proofs, Maurer and Schmid [133] developed early on a calculus of channel security properties and transformations between them. Hoepman [92], Vaudenay [190], and Laur and Nyberg [118; 120] provide security proofs for their proposals. Naor et al. [143] argue that the difference in user effort required to manually authenticate longer strings and shorter strings necessitates the calculation of tighter security bounds, and they provided proof techniques to calculate tight bounds. They provided a protocol proposal which is proven to be optimal according to the chosen criterion of reducing the length of the authenticated bits for a given security level, but this is at the cost of an increased number of message rounds. Creese et al. [49] argue that, in the formalization of the attacker model in a ubicomp environment, it is not necessary to assume a Dolev-Yao [63] attacker across all channels; and they consider how to model this in formal tools such as FDR, CSP and Casper.

In terms of instantiation of specific channels, McCune et al. [136] propose the above-mentioned visual channel of “Seeing-is-Believing”, with protocols derived from the Balfanz et al. proposal. Goodrich et al. [77] introduce a system called “Loud and Clear” that translates a hash value into a human-verifiable vocalized sentence; their auxiliary channel is thus audio. Kindberg and Zhang [107] use ultrasound.

Usability is an essential facet of security, even though it has received insufficient attention. Uzun et al. [187] have conducted a usability study of different set-ups for the visual channel.

Some standardization bodies are also working on secure and usable device pairing. We will list these at the end of this chapter.

4.3 Multiple Channels: open insecure channel and auxiliary channel

We will now review the previous protocols proposed by other researchers, which had made use of different types of channels available in a ubicomp environment.

Common to all of these protocols is at least an open insecure channel – a (wireless) radio frequency channel, which is necessarily broadcast in nature. Additionally, the protocols make use of auxiliary channels. These may be wired contact, visual, audio etc, possessing stronger security properties than the open channel.

We will leave a fuller discussion of the properties possessed by channels until later. Our purpose here is to present our argument that a lack of explicitness in considering the security properties of channels has hindered the very design of secure protocols.

4.3.1 Inadequacy of Open Insecure Channel

The radio medium used by most wireless devices is of a broadcast nature. Any receiver within audible radio range (i.e. within some distance from the radio transmitter, depending on the propagation loss characteristics of the environment and frequency) can overhear the transmission. Thus, any plaintext transmission over radio may be overheard by an adversary within close proximity, or even at long range if this adversary is equipped with a range-extending directional antenna.

More serious than merely overhearing (i.e. reading all traffic), the standard adversary is modelled as being able to modify, delete and create messages over this open channel. The original definition is given in Dolev and Yao's paper [63], hence the usual terminology of a *Dolev-Yao* attacker.

Thus the radio channel is commonly accepted as being totally insecure and on its own inadequate to bootstrap a security association. We have ruled out a public-key infrastructure with a certification authority (which would imply revocation checking), hence a message received over the radio channel needs to be authenticated by other means. We will next overview simple ad-hoc key agreement over a radio channel between two parties, and step through why it is insecure.

In the Resurrecting Duckling model [180], trust is bootstrapped from a secret transferred via a secure channel between the two parties. The recommended secure channel is physical contact: it gives a strong guarantee that the shared key has been established between the two chosen devices and no others, with high confidentiality and integrity. It makes cryptography redundant for key establishment. Wired contacts on personal devices, however, are surprisingly expensive in the eyes of manufacturing engineers once we take into account not just the cost of the connectors but the additional board area and the geometrical and ergonomic constraints on industrial design.

#	Alice		Bob
		Basic DH	
1	Chooses random a		Chooses random b
2		$- g^a \rightarrow$	
3		$\leftarrow g^b -$	
4	$K = (g^b)^a$		$K' = (g^a)^b$
		B challenges A	
5			Chooses random C_b
6		$\leftarrow C_b -$	
7	$M_1' = H(K, C_b)$		$M_1 = H(K', C_b)$
8		$- M_1' \rightarrow$	
9			Verify $M_1' = M_1$
		A challenges B	
10	Chooses random C_a		
11		$- C_a \rightarrow$	
12	$M_2 = H(K, C_a)$		$M_2' = H(K', C_a)$
13		$\leftarrow M_2' -$	
14	Verify $M_2' = M_2$		

Protocol Trace 4.1: Diffie-Hellman with key confirmation.

It should be clear, however, that carrying out a Diffie-Hellman exchange over RF gives no guarantees about the party with which one is establishing a key. The process is therefore vulnerable to a middleperson attack. Even if each of the two parties successfully challenges the other to prove ownership of the established key, as in steps 5–9 and 10–14 of Protocol Trace 4.1, the confirmation phase can never prove that the key was established with the desired party.

The obvious remedy is to have steps 8 and 13 take place over an extra channel, such as manual transfer [71] or visual transfer [136], that guarantees data origin authenticity. Manual transfer may be implemented by displaying on the first device a string that encodes the Message Authentication Code (MAC) [1] and by having the user type this string into the other device. This channel has limited capacity because it is unpleasantly laborious for human users to transfer long strings manually without making

[1] MACs are keyed hash functions used for message authentication purposes. In the past, the more common implementations of a MAC use a block cipher in cipher-block-chaining (CBC) mode. While the key of a MAC function can be thought of simply as part of the input into a hash function, strictly speaking these are not quite the same in terms of security, since a hash function does not accommodate naturally the notion of a secret key. From the mid-1990s, there has been interest to obtain a MAC function by using any sufficiently strong cryptographic hash function as a 'drop-in' [20]. For most sections of this chapter though, we will use (keyed) hash and MAC interchangeably, until we need to distinguish a hash and a MAC more clearly in Section 4.5.

mistakes. One may then have to transmit a truncated MAC.

4.3.2 Auxiliary Channel with (Data-Origin) Authenticity

In ubiquitous computing, manual transfer of MAC values or check codes by hand is but one type of channel.

Various researchers have coined the auxiliary channels as ‘location-limited channels’ [15] or ‘manual channel’ [118]. A more dated and better-known parlance would be ‘out-of-band channel’, which distinguishes it from the principal insecure broadcast channel and was used more often to refer to a totally secure (i.e. *confidential* in addition to data-origin authentic) channel.

Our preference is to designate such channels as *auxiliary channels*, having *data-origin authenticity*, with the said property being the key security property which we want to emphasize. During the operation of the channel, it is easy to ascertain the origin of the signal.

Another important security-related property regarding these channels is their limited data capacity or bandwidth, affecting usability. The size of this bandwidth comes into play in affecting the efficacies of two (broad) types of attacks: firstly, the probability of an adversary’s success in guessing correctly with one pass for an online attack, secondly, the search space of a brute-force enumeration offline attack, which we will cover in a later sub-section.

As with any open channels, there is a non-zero chance of noise and error coming in, in the case of these human-mediated auxiliary channels, the error may be due to faults in the transmitting and receiving equipment, or it may be due to human error in transcribing the information from one device to another. This will become clear in the following subsection.

4.3.3 Some Attacks on Protocols under Different Attacker Models

Let us consider attacks on several ad-hoc two-party key establishment schemes. The ways the attacks succeed are variously due to the continuing improvements in computational speeds of the adversary, and improvements in surveillance capabilities of an adversary which would negate assumptions about the security properties of the auxiliary channels.

4.3.3.1 Attack against Short Check Codes exchanged over Auxiliary Channel

As pointed out by Gehrmann et al. [70, section 2.3], if the manually transferred authentication code is too short then a middleperson attack is still possible.

The attack is shown in the Protocol Trace 4.2. The column ‘Ch’ describes whether the step consists of a transfer over the radio (RF) channel or over the manual (M) channel. In the context of this protocol trace, when we say RF we mean a channel subject to eavesdropping and substitution attacks and with no data origin authenticity, but no practical limits on capacity. When we say M we mean a channel offering data origin authenticity but with very limited capacity, of the order of 10–20 bits per message.

#	Ch	Alice		Carol		Bob
1		Chooses random a				
2	RF		$- g^{a} \rightarrow$			
3				Chooses random a'		
4	RF				$- g^{a'} \rightarrow$	
5						Chooses random b
6	RF				$\leftarrow g^{b} -$	
7				$K_{bc} = (g^{b})^{a'}$		$K_{bc} = (g^{a'})^{b}$
8						Chooses random C_b
9	RF				$\leftarrow C_b -$	
10				$M_1 = H(K_{bc}, C_b)$		$M_1 = H(K_{bc}, C_b)$
11				Chooses random b'		
12	RF		$\leftarrow g^{b'} -$			
13		$K_{ac} = (g^{b'})^{a}$		$K_{ac} = (g^{a})^{b'}$		
14				Finds C'_b s.t. $H(K_{ac}, C'_b) = M_1$		
15	RF		$\leftarrow C'_b -$			
16		$M'_1 = H(K_{ac}, C'_b)$				
17	M		$- M'_1 \rightarrow$	$\rightarrow$	$\rightarrow$	
18						Verify $M'_1 = M_1$
19		Chooses random C_a				
20	RF		$- C_a \rightarrow$			
21		$M_2 = H(K_{ac}, C_a)$		$M_2 = H(K_{ac}, C_a)$		
22				Finds C'_a s.t. $H(K_{bc}, C'_a) = M_2$		
23	RF				$- C'_a \rightarrow$	
24						$M'_2 = H(K_{bc}, C'_a)$
25	M		$\leftarrow$	$\leftarrow$	$\leftarrow M'_2 -$	
26		Verify $M'_2 = M_2$				

Protocol Trace 4.2: Middleperson Attack on Short MACs.

As the MACs must be short in order to be transmitted over the M channel, it is computationally feasible for middleperson Carol to search for second pre-images. After intercepting Alice's key contribution (step 2), Carol pretends to Bob that she is Alice and establishes a key with him (steps 3–7).

At this point Bob wishes to challenge his RF correspondent, whom he hopes to be Alice; the verification code will be received over the extra channel (step 17) and will therefore undeniably come from Alice. What does Carol do to fool Bob?

After receiving Bob's challenge C_b in step 9 and computing the keyed hash value M_1 in step 10 from the session key shared with Bob, Carol forms a session key with Alice (steps 11–13) and performs a brute force search (step 14) to find a challenge C'_b such that the keyed hash value M'_1 derived from it equals M_1. She sends the forged challenge C'_b to Alice (step 15). Alice computes M'_1 (step 16) and shows this result over the manual transfer channel (step 17) to Bob, who verifies it (step 18) against his computed result M_1 and finds that they match. Bob has been fooled.

Carol then performs the same forgery in the symmetrical situation of Alice challenging Bob (steps 19–26), fooling Alice as well.

The effort required by Carol to attack each challenge-response is of the order of 2^r trials [1] , where r is the bit length of each short MAC. Assuming an adversary with powerful computing resources who is able to perform 1 billion trials a second, and a device time-out of 10 seconds, Gehrmann et al. [70] calculate that a 48-bit code is needed to defeat this attack. But manually transferring 48 bits (which correspond to 12 hexadecimal digits) is tedious and prone to error. One alternative is to use an extra channel of greater capacity.

4.3.3.2 Attack against Key Fingerprints exchanged over Auxiliary Channel

It is possible to acquire the code from the screen with a camera instead of typing it on the keypad [160; 161; 169]. This would be an instance of a machine visual channel (vis-a-vis a manual visual channel). The number of bits that can be reliably transferred is slightly greater and usability improves significantly [186].

By 2005, the camera phones commercially available in Europe have reached resolutions of 1280 × 1024 = 1.3 megapixels (3.2 megapixels in Asia). Depending on the particular circumstances of the usage, the limiting factors for data throughput would include the camera resolution, the screen resolution and the focusing distance of the camera. Screen resolutions have reached 66 kilopixels, though 36 kilopixels are still more common. Based on these figures, with a suitable 2D encoding the screen-to-camera channel can reliably provide about 40 to 100 bits per message. This is still not enough for a full hash but it is sufficient for a longer code that would solve the problem described in the previous section. In actuality, current camera-phones are usable but are not optimal for this task: there are certainly problems of focusing distance, resolution and illumination. We will describe more about experiences and issues with the machine visual auxiliary channel in Section 4.6 on Implementations.

[1] While the *worst-case* complexity (from the attacker's point of view) for a pre-image attack or a second pre-image attack (see Definition 2.5) is 2^r, the *average-case* complexity would be 2^{r-1}. Since $r \gg 1$, for this dissertation we will simply approximate $r-1 \approx r$. Note that a collision attack would have a rather different complexity figure, of around $2^{r/2}$.

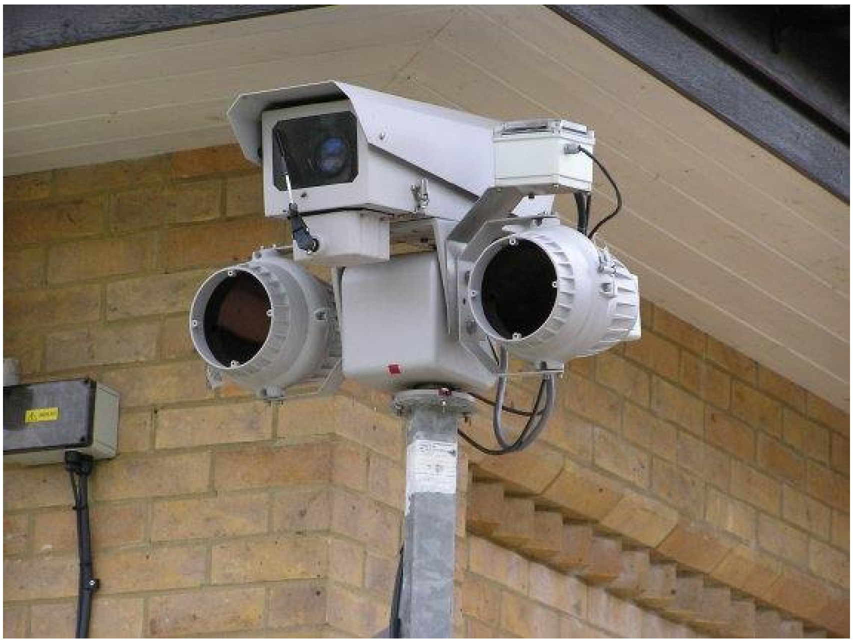


Figure 4.3: Outdoor CCTV Camera in University of Cambridge

The idea of transferring short cryptographic codes visually between camera phones was originally proposed by McCune et al. [136]. Their Seeing-is-Believing (SiB) protocol, derived from earlier work by Balfanz et al. [15], closes the vulnerability pointed out in section 4.3.3.1 by transferring a longer code over the extra channel. The protocol would still fail if the attacker could find a preimage but, because the camera phone channel used by the authors allows 68 bits per transfer (well over twice the capacity of a manually typed PIN), the brute force search is no longer feasible in real time. In other words, SiB requires at least a "medium length" code, not a "short" code[1].

4.3.3.3 Eavesdropping Attack on Non-Confidential Auxiliary Channel

There is however another threat. In protocols that implicitly or explicitly use an additional channel, as happens in many EKE variants in which a strong session key is formed from a weak PIN, there is often the assumption that the additional channel is somehow "local" and safe from eavesdropping.

We argue that, for the manual visual (screen and keypad or keypad and keypad) channel and for the machine visual (screen and camera) channel, this assumption is no longer realistic with the current

[1] Our very informal semantics for "short", "medium" and "long" codes in this context are as follows. "Short" is a code that can be brute-forced in a few seconds, during a run of the protocol, for example 10 bits. "Medium"is a code that can be brute-forced in an hour, a day or a month but not in real time during a protocol run, for example 50 bits. "Long" is a full length code that, assuming the hash function is not otherwise broken, cannot be brute-forced in hundreds of years, for example 250 bits.

#	Ch	Alice		Bob
1		Chooses random a		Chooses random b
2	RF		$- g^a \rightarrow$	
3	RF		$\leftarrow g^b -$	
4		$D = g^a \,\vert\, g^b$		$D = g^a \,\vert\, g^b$
5		Chooses random R		
6	V		$- R \rightarrow$	
7		Chooses random K_1		Chooses random K_2
8		$M_1 = m_{K_1}(I_A \,\vert\, D \,\vert\, R)$		$M_2 = m_{K_2}(I_B \,\vert\, D \,\vert\, R)$
9	RF		$- M_1 \rightarrow$	
10	RF		$\leftarrow M_2 -$	
11	RF		$- K_1 \rightarrow$	
12	RF		$\leftarrow K_2 -$	
13		$M_2' = m_{K_2}(I_B \,\vert\, D \,\vert\, R)$		$M_1' = m_{K_1}(I_A \,\vert\, D \,\vert\, R)$
14		Verifies $M_2 = M_2'$		Verifies $M_1 = M_1'$
15	L		– outcome →	
16	L		← outcome –	

Protocol Trace 4.4: MANA III.

proliferation of CCTV cameras, both indoors and outdoors (such as that shown in Fig. 4.3). In many cases, such cameras even operate covertly, hidden behind opaque domes that allow them to pan and zoom without the victims knowing where the cameras are pointing. In this section we will discuss the consequences of this change in the attacker model.

The MANA III scheme by Gehrmann et al. [70], developed as a variant of Larsson's SHAKE, is shown in Protocol Trace 4.4. It aims to establish that both parties have correctly received each other's public key. It complements the radio transmissions with an exchange of short codes using manual transfer. As the authors themselves say, "Informally, the security of the scheme relies on the fact that R remains secret to the attacker (it is never sent over the air)…". In the presence of a passive attacker in the optical domain, which we believe can no longer be dismissed, this protocol can be cracked.

I_A and I_B are identifiers of Alice and Bob respectively and are publicly known. D is a data string formed from the concatenation of Alice's and Bob's public keys g^a and g^b. K_1 and K_2 are long keys. M_1 and M_2 are long MAC values formed using the function m. R is a short randomly selected string shared between Alice and Bob over the manual channel. Alice will only send K_1 after she has received M_2, and Bob will only send K_2 after he has received M_1.

Crucially, after the verification (step 14), each device must signal whether the verification succeeded (steps 15 and 16), over a channel (e.g. red/green LED) guaranteeing integrity and data origin authenticity. Although the M channel could be reused here, we indicate this channel as L (LED), rather than M, to point out that its requirements are less demanding than those of the channel used in step 6. In particular, its required capacity is only one bit per message. Note the additional subtlety that the LED is signalling to the *operator* of the device(s), not to the other device directly.

As noted in the original paper [70], if Bob were not told that Alice's verification failed, middleperson Carol could send Alice a random M_2 in step 10, grab K_1 from Alice in step 11, ignore the rest of the protocol run with Alice, find R by brute force and successfully impersonate Alice to Bob, who would not notice the forgery.

The reason why the attack of Section 4.3.3.1 no longer suffices is essentially because the short code exchanged over the extra channel is not the MAC but the challenge: the MAC, now transmitted over radio, is full length and not vulnerable to second preimage attacks. The protocol's security, however, relies on the challenge R being kept secret from the middleperson attacker.

If this assumption is violated, the attack is as follows. Assume that middleperson Carol is able to observe the string R being keyed into the devices. She may then send modified public keys to both Alice and Bob, such that Alice's and Bob's copies of D are different from each other's, but match the two copies held by Carol. Thereafter, Carol can individually choose different K's and generate the M's so as to authenticate successfully with both Alice and Bob. Note that this attack is independent of the length of the MACs.

4.4 Types and Properties of Channels

In earlier sections, we have briefly mentioned specific security properties such as *data-origin authenticity*, *confidentiality*, etc. In this section, we will now discuss in greater detail all the channel properties that are relevant to the design of security protocols. Basic parameters of these channels such as bandwidth, degree of *human-mediatedness*, and so on are also of interest.

We will also discuss briefly the informal notion of a *Multi-Channel Attacker*.

4.4.1 Differences of Channels

4.4.1.1 Authenticity

Degrees of *authenticity* have been highlighted by Vaudenay [190]. He described ‘integrity-protection’ for his ‘authentication channels’, where attackers can stall, replay or remove a message, but cannot modify it. For ‘stronger authentication channels’, he mentioned stall-freeness, listen-ready situations, and transmission acknowledgements. Another level of differentiation is possible: namely, weaker authentication channels, where it is conceivable that a very low number of bits may be added to a message sent over the channel without detection. We suppose the impact of selective bit-flipping of several bits on protocols which utilize values exchanged over such channels would be amenable to further study.

Closely related to authenticity is the property of integrity: Cagalj et al [191] have proposed an integrity-protecting scheme for what would normally be classified as the totally insecure channel of the radio.

It is to be noted that for the subsequent protocols in our discussion, we make the straightforward assumption that our auxiliary channels possess the following sub-properties of authenticity: data-origin authenticity and high integrity. We are not too concerned at all if the adversary can read, delay or cause denial-of-service on the messages over the auxiliary channels (in fact we take for granted that he can read them). But it is clear that in a ubicomp environment, the salient property of data-origin authenticity is derived from the close human-machine interaction.

Remark : Auxiliary channels with data-origin authenticity are human-mediated channels whereby the source of the messages carried over them can and are corroborated by the human operator (who is usually the recipient of the message or the owner of the receiving device).

We will elaborate on human-mediation in Section 4.4.1.4.

4.4.1.2 Bandwidth

We describe informally a *high-bandwidth channel* as one suitable for transferring a long public-key value or full hash value. The radio channel is such a channel.

An auxiliary channel is almost always *low-bandwidth channel*. When these channels are used in security protocols, the governing guidance is that the values they transfer must be applied such that the attacker has only a one-shot probability of success related to the entropy of the code, instead of merely brute-forcing the code to find a match.

We use another type of channel in our protocols. This is, namely, the authentic *1-bit per message channel*, used because of its property of authenticity. This is one of our original contributions to the study of multi-channel security protocols. The 1-bit value sent over this channel is not used to introduce randomness and to force an active attacker to predict some value and commit prematurely, unlike in the case of the authentic auxiliary channel. Instead, this 1-bit channel is typically used to authentically acknowledge receipt of an intended message, or to signal that a computed value has verified successfully.

4.4.1.3 Input or Output or Input/Output

For channels of the same type, for example a visual channel, at one level we can consider whether two devices have an input (I) and output (O). Visual channel inputs would encompass keypads (susceptible to human transcription errors) and cameras (able to capture 2D images and thence more information). Visual channel outputs would encompass anything from a primitive LED text screen to a colour high-resolution display monitor. If an auxiliary channel comprises of an Input at one end, and an Output at the other end, then we define this as a unidirectional channel. If the channel has both Inputs and Outputs at both ends, the channel is clearly bidirectional.

The slight difficulties occur when the auxiliary channel is O/O or I/I. If it consists of an Output at both ends (which can be used to compare the check values computed by both devices), then we may perhaps define this channel as unidirectional; and a separate one-bit channel seems necessary to notify the devices on whether their check values agree. In the case of a channel being I/I, strictly speaking, there is no means of transferring a short random value produced in one device to the other, to allow comparison. However, the protocol can be designed to allow a human to key in the same short value into both devices, as a passkey, though this usage often assumes confidentiality. Or, if the channel input is audio, a human can speak the same phrase into both devices' microphones. It is clear that there often is a strong though variable element of human-mediation on these auxiliary channels, and sometimes directionality does not cover everything. Often, such a setup suggests that the value is used as a sort of short *confidential* PIN (i.e. password).

4.4.1.4 Human-mediation

We bring attention to the informal notion of the degree of *human-mediation* on the auxiliary channels. It is not merely a matter of human error in perceiving and transcribing values in an error-free way over, say, a visual channel. For certain auxiliary channels, such as infra-red and NFC [1], a human operator may bring two participating devices close so that their transceivers can exchange messages, yet, the human sometimes has no direct knowledge of whether the protocol trace actually takes place, or if a transmitted message is received as intended with high fidelity. (Some IR-equipped devices do emit a cute sound as feedback when they detect another device in range.) An attacker using diffuse infra-red, and a long-range radio antenna, may be able to violate assumptions of usage regarding these particular auxiliary channels. On the other hand, if the auxiliary channel in question is human-mediated audio, it may sometimes be difficult for the human user to pinpoint which device a melody is emanating from; this channel may be said to have integrity but less data-origin authenticity. Clearly, there is much scope for usability studies on diverse channels.

As a note for protocol designers, the operation of an auxiliary channel must not make excessive demands on the able-bodied human user, for example, in terms of concentration, memory, coordination, and sensory acuity. Instead it should be simple, intuitive, and pleasant. Tying in with our emphasis on explicitness in the channel modelling, the human-mediated action required on the auxiliary channel used should be well-defined, so that oversights are not inadvertently introduced that will result in insecurity.

To accommodate users with some handicaps or less sensitive faculties (such as the blind, the deaf, the long-sighted, etc), designers and system developers would need to be more inclusive, and that is perhaps where a diversity of channels would offer alternative ways of bootstrapping security.

4.4.1.5 Comparison of Channels

We present some types of channels found in a pervasive computing environment below; the list is illustrative and not exhaustive.

[1] Near Field Communication or NFC is a short-range radio frequency communication technology, operating at the unlicensed band of 13.56 MHz, up to a range of nominally 20 centimetres. It is standardized in ECMA-340 and ISO/IEC 18092, and is marketed at, among others, 'contactless' mobile payment and usage on handphones.

#	Channel	Output	Input	Range	Confid.	Authent.	Bandwidth	Convenience
1	Radio	RF Txr	RF Rxr	Long	Low	Low	High	High
2	Visual	Screen	Keypad	Medium	Medium	High	Low	Medium
3	Visual	Screen	Camera	Medium	Medium	High	Medium	Medium
4	1-bit Pushbutton	Screen	Button	Medium	Low	High	V Low	Medium
5	Infra-Red	Diode	IR Rxr	Short	Medium	Medium	Medium	Medium
6	Contacts	Pin	Pin	V Short	High	High	High	High
7	Cable	Port	Port	Medium	Medium	Medium	High	Low
8	NFC	Txr	Rxr	Short	Medium	Medium	High	High
9	Audible Sound	Speaker	Mic	Medium	Medium	Medium	Medium	Medium
10	Ultrasound	Speaker	Mic	Medium	Medium	Low	Medium	Medium
11	Voice	Mouth	Mic	Medium	Medium	Medium	Medium	Medium

Figure 4.5: Table of Channel Types and Properties

The levels: High, Medium, Short and Very Short are general, since we do not give precise figures; it is difficult to do so, being particularly implementation-specific.

Channel number 1, the Radio Channel, is an example of the totally non-secure channel. The column for ‘Authenticity’ (Authent.) is the most pertinent for the selection of an auxiliary channel; and it is related to degrees of human-mediatedness, since the human operator in the loop should take corrective or abortive action in the protocol run if he senses something amiss. For conciseness, we will not break down this column further into its possible sub-components. For usability reasons, the ‘Convenience’ column is also very important.

4.4.2 Multi-Channel Attacker Model

Creese et al. [49] have proposed Variant and Twin-Channel Threat Models in the ubiquitous computing environment, to aid formal analysis of the security of protocols. This is an important advance.

It seems natural that this can be extended. We can suggest that in a diverse ubicomp environment, there will be a diversity of channels at play (not limited to just two types), hence there will be different attacker capabilities across the range of channels, such that it will be worthwhile to consider a *multi-channel attacker model*.

It is recognized that the Dolev-Yao attacker is of course the most powerful theoretical attacker in formal modelling, and it is justifiable that many previous security protocols are predicated on resisting this attacker of nearly unlimited (except for cryptographic) capability. Attacker capabilities on other auxiliary channels are usually much more limited.

Specific properties of a channel, and thence the corresponding attacker capability on it, can be defined and analyzed. For example, in addition to the informal notions proposed by Vaudenay [190], other properties were also suggested by Creese et al [49], such as 'Atomicity of Transaction', which they define as the situation where the attacker cannot *both* block *and* hear a message at the same time on the channel under consideration. Further combinations and refinements in properties are conceivable.

The degree of human-mediation required for operation of the auxiliary channel does affect its usability, and is intertwined with security, hence impacting on the multi-channel attacker model. Thus, designers have to be careful not to be over-demanding in the degree of user engagement during the protocol run. For example, where the auxiliary channel used in a protocol is audio, it may be useful to ask, would a human user be reasonably able to detect when an attacking device is inserting another sound clip during a run of the protocol?

In the multi-channel attacker model, it is worth noting that the attacker combines and coordinates his actions over a variety of channels, to achieve his purpose. We have sketched the multi-channel attacker, but we have not managed to formalize the model. A more formal model of a multi-channel attacker with different capabilities across different channels would be beneficial for theoreticians, who often want to construct formal proofs and attack games or complexity-theoretic bounds for security protocols.

4.5 Multiple Channels for Two-Party Key Agreement

4.5.1 Security Requirements

We require an *authenticated two-party key agreement protocol* which makes use of the channel properties commonly available in a ubicomp environment. The existing protocols were unsatisfactory, because their security breaks down under different assumptions about attacker capability on the auxiliary channels.

We want to combine resistance to passive attacks offered by Diffie-Hellman, with resistance to active attacks offered by some short-length random value which is authentically exchanged but which gives the attacker a low one-shot probability of success at predicting it.

Succinctly, we require the following security properties:

Resistance to passive attacker : This property would be provided by entangling Diffie-Hellman in the protocol. The work factor of a passive attacker to calculate the shared DH key would be of around the order of the security parameter (i.e. the size of the finite group) of the DH key agreement used.

Resistance to active attacker[1] : This property would be provided by giving the active attacker a low probability of predicting correctly some randomly generated authentic value. This probability is based on the security parameter of the authentic value, namely the entropy with which the value is randomly

[1] Sometimes researchers make the distinction between impersonation attacks and substitution attacks. In an impersonation attack, adversary Eve tries to inject a message when the potential victim Alice has not sent an message. In a substitution attack, Eve tries to modify in transit a message that has been sent by Alice. We would not make the distinction here because it would not be particularly informative in our discussion.

selected.

4.5.2 Multi-Channel Security Protocol Proposal

4.5.2.1 Protocol with Bidirectional Auxiliary Channels

Sections 4.3.3.1 and 4.3.3.2 have shown protocols that can be cracked if the attacker can brute-force in real time the short code sent over the extra channel; they therefore require at least a “medium length” code. Section 4.3.3.3 has shown a protocol that resists brute force even with a short code, but which is vulnerable if the attacker can eavesdrop on the extra channel. Is it possible to come up with a protocol that transmits only short codes (rather than “medium” ones) on the extra channel but, despite that, is not broken by eavesdropping?

We developed such a protocol [202] and presented it at a workshop (the 13th International Workshop on Security Protocols, in Cambridge in April 2005), and later[1] found it to be equivalent (in terms of properties of channels present, and protocol objectives) to one of Hoepman’s φKE proposals. It is apparent that our independently discovered protocol has similar functionality to the φKE protocol with the bidirectional authentic channel [92, Protocol 3.3 and Fig. 3]. In both our protocol and Hoepman’s protocol, the extra channel (which could, for example, be screen-to-camera or screen-to-keypad) is utilized for its integrity and data origin authenticity, so confidentiality is not assumed for this channel.

Our protocol proposal was inspired by and is an improvement upon MANA III. Our proposal is shown in Protocol Trace 4.6, We aim to assure that a session key is established with the correct party. The pre-conditions are an insecure channel having low confidentiality and low integrity, and a bandwidth-limited auxiliary channel having low confidentiality but high integrity and high data origin authenticity.

Remark : In the specific instances which we would use throughout the rest of this chapter in our protocol proposals, the insecure channel is a radio frequency channel and is denoted by ‘RF’, while the auxiliary channel, unless otherwise stated, would be a visual channel be denoted by ‘V’.

R_a and R_b are short random nonces. K_a and K_b are long nonces. M_1, M_1', M_2 and M_2' are long MAC values.

We depart slightly from the notation in our earlier paper [202] where we had used H to signify a (keyed) hash. Equivalently, instead of describing this as a (keyed) hash function H, it is also desirable and more explicit to describe this as a MAC function m, in which case the K values would be used as the corresponding MAC keys instead of being part of the input into the pseudorandom function con-

[1] A fellow attendee (namely Alf Zugenmaier) at the workshop subsequently brought to our attention Hoepman’s φKE paper [92], which was published in the preceding year.

cerned. That means that, $H_1 = H(I_A \,|\, g^a \,|\, g^b \,|\, R_a \,|\, K_a)$ [1] , for example, is more precisely re-written as $M_1 = m_{K_a}(I_A \,|\, g^a \,|\, g^b \,|\, R_a)$. The MAC function m is used as a non-malleable commitment [51; 52] (see below).

Definition 4.1 : Informally, the property of *non-malleability* [64] in the context of encryption states that it is impossible to derive from a given ciphertext a new ciphertext such that the underlying plaintexts are meaningfully related. Conversely, in a *malleable* cryptosystem, anybody can compute an encryption of plaintext message x into a valid encryption of $f(x)$, for some restricted class of functions f, without necessarily knowing x (and without knowing the key). For example, stream ciphers and the one-time pad, when used without integrity protection, are highly malleable.

Definition 4.2 : A *commitment* scheme is a method that allows a party to commit to a value (or message) while keeping it hidden, and while preserving the party's ability to reveal the committed value later. The interactions of the scheme typically take place in two phases: the *commit* phase in which a value is chosen, and the *reveal* phase in which the value is revealed and checked. In other words, the committer party, at the commit phase, with the inclusion of some public parameters, some random value r, transforms a value x into a commitment string c and a decommitment value d (d is usually (x,r)); he then sends the commitment c to the receiver party. At an appropriately later time, at the reveal phase, he sends the decommitment value d to the receiver; the receiver will use c and d and check whether he can derive the value x from these (i.e. whether he can *open* the commitment). Two basic cryptographic properties of a commitment schemes are the following: its *hiding* property (the extent to which the value chosen during the commit phase cannot be discovered), and its *binding* property (the extent to which the value thence chosen is the only one that can be revealed during the reveal phase).

Definition 4.3 : Intuitively, such a scheme is a *non-malleable commitment* scheme if given a valid commitment c, it is difficult for an adversary to calculate a valid commitment whose underlying message is related, which can be successfully opened when the decommitment value d (corresponding to c) is given subsequently.

In practice, a viably efficient commitment construct can be obtained from the HMAC [2] [20; 21] family, such HMAC-SHA-256. Roughly speaking, if the MAC used is not non-malleable, then it is conceivable that an active attacker may be able to swap the actual MAC output for a different MAC output, for which the two corresponding MAC inputs can comprise a common nonce, even though the attacker does not know in advance the value of the nonce. Thus, in our context, the notion of non-malleability of

[1] As mentioned, a hash function is not by design intended to use a secret key. Consider an attempt to use the key as a *secret prefix*, as in $H(K|x)$, where H is an iterative hash function; an adversary can extend the message by a single block y, and can derive $H(K|x|y)$, without even knowing K [137, Section 9.5.2]. A different attack succeeds if K is used as a *secret suffix*.

[2] The structure is $HMAC_K(x) = H((K \oplus opad)|H((K \oplus ipad)|x))$ where $ipad$ and $opad$ are the inner and outer paddings respectively, each of which is one block in length.

#	Ch	Alice		Bob
1		Chooses random a		Chooses random b
2	RF		$- g^a \rightarrow$	
3	RF		$\leftarrow g^b -$	
4		Chooses random R_a		Chooses random R_b
5		Chooses random K_a		Chooses random K_b
6		$M_1 = m_{K_a}(I_A \mid g^a \mid g^b \mid R_a)$		$M_2 = m_{K_b}(I_B \mid g^b \mid g^a \mid R_b)$
7	RF		$- M_1 \rightarrow$	
8	RF		$\leftarrow M_2 -$	
9	V		$- R_a \rightarrow$	
10	V		$\leftarrow R_b -$	
11	RF		$- K_a \rightarrow$	
12	RF		$\leftarrow K_b -$	
13		$M_2' = m_{K_b}(I_B \mid g^b \mid g^a \mid R_b)$		$M_1' = m_{K_a}(I_A \mid g^a \mid g^b \mid R_a)$
14		Verifies $M_2 = M_2'$		Verifies $M_1 = M_1'$
15	L		– outcome →	
16	L		← outcome –	

Protocol Trace 4.6: Our MANA III variant.

the commitment requires that:

Given a commitment $m_K(x|x_0)$, it is difficult for an adversary to generate a valid $m_{K'}(x'|x_0)$.

Remark: Note that $(x|x_0)$ and $(x'|x_0)$ are *related*; the adversary has essentially arbitrary choice of K' and x', but he has no control over x_0.

In the protocol, each party generates an ephemeral Diffie-Hellman private value, computes the corresponding public key and sends it to the other party. Alice chooses a short random R_a (step 4), a long random K_a (step 5), and computes the concatenation of R_a with her identifier I_A and the public keys, into a long MAC output M_1, using K_a as the MAC key (step 6). Bob does likewise and produces a long MAC output M_2. Next, Alice and Bob both send over the RF channel their computed MAC outputs to each other, which represent their commitments (steps 7 and 8). Bob must indicate that he has received a MAC, and only then, and not before, Alice may release R_a over the visual channel and K_a over the radio channel. Similarly, Alice must indicate that she has received a MAC and only then, and not before, is Bob allowed to release R_b and K_b. After all the R and K have been received, both sides proceed to compute the MACs and verify that they match the copies they had received earlier in steps 7 and 8.

The length of the long MAC outputs determines the size of the complexity theoretic problem a potential middleperson attacker would face for finding their second pre-images. The length of the visually exchanged R values determines the probability or “luck” the attacker would have in choosing coincidentally the same R values for the commitments as Alice and Bob might later choose.

The protocol is symmetric: steps 7, 9 and 11 prove to Bob that he is communicating with Alice; conversely, steps 8, 10 and 12 prove to Alice that she is talking to Bob. If one set of steps is absent, the authentication is only unilateral. We explore this case in section 4.5.2.2.

Compared to MANA III, this protocol relies on the strong data origin authenticity property of the extra channel rather than on its confidentiality: when an R value is exchanged, we have high confidence that it originated from the observed party. The difference is that, here, both parties must issue their commitments M values before the release of any of the R and K values. Therefore an attacker Carol who manages to observe the R values will be too late to compromise the key agreement, because she must have already committed to a fake M for which she will not be able to generate a matching K.

One may wonder why we need the K values, if the unforgeable R values are there (ie. why use a MAC with a secret key K, if we can simply use a hash). This is because, if there were no K_a, middleperson Carol could otherwise intercept Alice’s M_1 in step 7 and try all possible values for R_a until she found the one that produced the correct hash. At that point Carol would be able to substitute her own key $g^{\tilde{a}}$, compute the hash $M_{\tilde{1}} = H(I_A \,|\, g^{\tilde{a}} \,|\, g^b \,|\, R_a)$ and send it to Bob. Since the R_a is the genuine one that Alice will later disclose, Bob will find that the M_1' he computes in step 13 will match this one he received from Carol in step 7 (all the inputs are the same). So the K values are there to prevent Carol from brute-forcing the R values out of the M values.

If step 14 completes with successful mutual verification of the MACs, both parties will have high confidence that the party from whom each has visually obtained the R value is the same party from whom each has received a public key. As in the original MANA III protocol, both devices must finally indicate (steps 15 and 16) whether the verification succeeded or not: each device should only consider the protocol run successful after receiving an indication that the other participating device also succeeded during step 14.

As suggested above, the middleperson attacker Carol has basically two options (apart from attacking the Diffie-Hellman component). In the first option, she guesses an R value, inserts a modified public key and a MAC computed from a random K value, and then hopes that the spoofed party will coincidentally choose the same R value. The probability of this attack succeeding is 2^{-r} where r is the bit length of the R value. Carol has less than 1% chance of success for an R as short as 7 bits. In the second attack option, Carol inserts a modified public key and a random hash. After the R value is disclosed by the spoofed party, the attacker embarks on a search for a K value which can yield the MAC she has already committed to. The complexity of such a search is of the order of 2^h where h is the bit length of the MAC. Since we said M was “long”, this is by definition infeasible.

Thus the protocol is strong even under the model of a powerful (i.e. dual-channel) attacker who is able to eavesdrop on the extra channel and rewrite messages on the RF channel, and in a situation in which the extra channel can only carry a “short” (not even “medium”) payload.

#	Ch	**Alice** (mother duck)		**Bob** (duckling)
0	PW			Start imprinting
1		Chooses random a		Chooses random b
2	RF		$- g^a \rightarrow$	
3	RF		$\leftarrow g^b -$	
4				Chooses random R_b
5				Chooses random K_b
6				$M_2 = m_{K_b}(I_B \mid g^b \mid g^a \mid R_b)$
7	RF		$\leftarrow M_2 -$	
8	PB		$-$ ack $\rightarrow$	
9	V		$\leftarrow R_b -$	
10	RF		$\leftarrow K_b -$	
11		$M_2' = m_{K_b}(I_B \mid g^b \mid g^a \mid R_b)$		
12		Verifies $M_2 = M_2'$		
13	PB		$-$ outcome $\rightarrow$	

Protocol Trace 4.7: Asymmetric pairing.

4.5.2.2 Protocol with Unidirectional Auxiliary Channel Restrictions

Now imagine the case in which the devices are not peers and the visual channel can only be established in one direction. For example, one device is a large stand-alone screen with some local processing power; it sits in a shop window and displays a pre-programmed sequence of text and graphics. The other device is a PDA that, every week or two, uploads a new sequence into the screen over radio.

The screen needs to be imprinted to the PDA of the shopkeeper so as to prevent anyone else from uploading messages to the screen. We assume that the PDA has a camera but the screen doesn't; and that, owing to industrial design constraints, it is not possible to use a wired connection between the two. Our goal is to devise a sufficiently secure method to perform the Resurrecting Duckling's imprinting operation in the absence of a wired contact.

Taking Alice as the mother duck PDA and Bob as the duckling screen, we cannot perform all the exchanges in Protocol Trace 4.6 because the visual channel only works from B to A; the message in step 9 cannot be sent and this cancels out the whole subprotocol in which A acts as prover and B as verifier (steps 7, 9, 11, 16 and Bob's half of steps 13 and 14).

The bits we can still do are in Protocol Trace 4.7. After successful completion, Alice the PDA is assured that she has established a key with Bob the screen, but Bob receives no proof that he is being imprinted to the correct PDA. This seems incomplete, which is what makes this protocol interesting.

What we wish to avoid is for Bob to be persuaded to imprint itself to another device Carol. How can this be stopped if Bob knows nothing about the device with which it is pairing? In the Resurrecting Duckling policy, introduced in [180] and formalized in [178, section 4.2.5], Bob the duckling imprints itself to the first mother duck he sees, whoever she is. What we want here is to prevent Carol from appearing in front of Bob for imprinting before he has a chance to see Alice. A crucial element of the solution is the presence of a human operator who wishes to imprint Bob to Alice.

Although manufacturers would love to get away with a Bob that had no other inputs than a wireless interface, we believe we also need at least the following:

1. a way to ask Bob to start imprinting;
2. a way to tell Bob whether to proceed or not, before committing to a proposed imprinting.

These two input mechanisms must be available only to a human operator Hermione having physical control of device Bob. The intention is to construct a protocol that cannot be subverted by hidden middleperson device Carol so long as human operator Hermione has physical control of duckling device Bob during the imprinting phase. Once imprinting is over, duckling device Bob may be left unattended: the Duckling policy will ensure that it can't be taken over by Carol or anyone else unless the mother duck device Alice first voluntarily relinquishes control.

Mechanism 1 could be implemented as nothing more than the act of switching on device Bob when he is still in his imprintable state. This is indicated (very poorly) as step 0 in the trace, with PW indicating the "power" channel. Mechanism 2, on the other hand, could be implemented as two mutually exclusive pushbuttons (yes/no, ok/cancel, proceed/abort…) with timout, indicated as channel PB in step 13, or even as a single pushbutton.

The exchange presented in Protocol Trace 4.7, obtained from Protocol Trace 4.6 by removing the steps in which Alice authenticates to Bob[1], proves to Alice that she and Bob are using the same two public keys g^a and g^b. Once human operator Hermione is satisfied that device Alice completed her verification succesfully in step 12, Hermione presses the "yes" pushbutton (step 13) on duckling device Bob, thereby ordering Bob to compute and commit to the imprinting key g^{ab}. If Hermione observes that Alice's verification failed, she presses pushbutton "no" (or lets Bob abandon the protocol by timeout, which could also be utilised as a way to allow just one pushbutton rather than two) and Bob forgets the previous exchange and remains imprintable.

An unattended attacking device Carol, with ability to eavesdrop on the V channel and with ability to rewrite messages on the RF channel, cannot imprint Bob to herself unless she can also *press* the "yes" pushbutton used in step 13 to commit the imprinting. Even if Carol had a mechanical finger that allowed her to press Bob's button, it is expected that Hermione would notice this and disallow it—that's the point of Hermione "having physical control" of Bob.

This protocol is interesting because it seems incomplete. Alice never proves herself to Bob. Bob doesn't actually know with whom he paired. Something appears to be missing. And yet, it works: Bob

[1] Note that we had to introduce a "one-bit-payload" step 8 to maintain synchronization. Bob should only display R_b after being sure that Alice received a hash. In Protocol Trace 4.6, this was achieved implicitly by Alice having to send something useful in step 9. Here, even though she has nothing useful to send at that stage, she must still signal to Bob, over the unforgeable extra channel, that she received the MAC and that he can proceed.

can only pair with the correct Alice (even if he can't recognize her) because Hermione won't let him proceed otherwise.

In the protocol of Section 4.5.2.1 we achieved mutual authentication with a bidirectional "short" extra channel. In this protocol we show that we can achieve the equivalent result even with a unidirectional "short" extra channel channel coupled with an even shorter "one bit only" extra channel in the opposite direction. In a sense, Bob is delegating his trust to Alice and Hermione.

Note that this core idea (tricky delegation-based strong security equivalent to mutual authentication despite asymmetric extra channel) could have been demonstrated with a much simpler protocol if the unidirectional extra channel had been allowed to be "long" rather than "short"; but this is true of most of the other protocols we discussed, which would have all basically reduced to a Diffie-Hellman augmented with unforgeable transmission of the MACs of the keys.

4.5.3 Security Analysis

We summarize here for completeness the security arguments presented earlier, and further elaborate on the third attack (which is also foiled by our protocol proposals).

We limit ourselves to a heuristic analysis in this dissertation. Security proofs, while raising confidence in a protocol's security if the proofs are well-constructed, are not watertight, as mentioned in Section 2.7. Regarding the particular area in this chapter, for his φKE protocols, Hoepman [92] had provided security proofs for his protocols in the Bellare-Pointcheval-Rogaway [22] model. Vaudenay [190, page 2] described Hoepman's security proof as being incomplete because no hash functions with the properties required by Hoepman exist. (We decided to use a MAC function, and not a hash function, in our proposed protocols, whereby the former's utility as a commitment function is better understood.) In his proposal, Vaudenay [190] used different types of commitment schemes, and provided security proofs associated with these. However, Laur et al. [118, section 4.2] have suggested that Vaudenay's proofs may be incorrect, and that some of the properties of his proposed commitment constructions are insufficient to assure the protocol's security.

It would appear that a rigorous and definitive security proof in this area remains an active area of research.

4.5.3.1 Attack I - Solving Diffie-Hellman Problem by Passive Attacker

As described in previous sections, a passive attacker who eavesdrop on the Diffie-Hellman public keys (i.e. g^a and g^b) may attempt to calculate the shared secret (i.e. g^{ab}), and has essentially to solve the DHP. The difficulty faced by such a passive attacker against in the two protocols is dependent on the security parameter of the finite group used.

4.5.3.2 Attack II - One-Shot Guess by Active Attacker

The active attacker in this case substitutes a DH public key and makes a commitment derived from a guess of the short nonce which is yet to be released. This attack, to predict the value of the authentically exchanged nonce, has a higher chance of success than the Attack I mounted by a passive attacker. However, this is statistical, rather than complexity-theoretic. The probability of success is around a one-shot probability of 1 in 2^r of predicting correctly the nonce, where r is the bit length of the nonce, R.

4.5.3.3 Attack III - Brute-Forcing Commitments (and Nonces)

In the third attack option, the attacker does not have to solve the Diffie-Hellman Problem (DHP) as in Attack I, nor have to rely on luck as in Attack II. Let us consider the case of this attack against the protocol with bidirectional low-bandwidth auxiliary channels, i.e. Protocol Trace 4.6. Assume that the attacker Carol is attempting to masquerade as Bob to Alice. She substitutes $g^{\tilde{b}}$ and commits to some $M_{\tilde{2}}$ before the real Bob releases the actual R_b. When the R_b is released, Carol then needs to provide a $K_{\tilde{b}}$, which will satisfy $M_{\tilde{2}} = m_{K_{\tilde{b}}}(I_B \,|\, g^{\tilde{b}} \,|\, g^a \,|\, R_b)$. Assuming that Carol has already used an incorrect R_b to calculate her commitment $M_{\tilde{2}}$, she is now required to perform a brute-force second pre-image attack to search around 2^h values (where h is the bit length of M_2) to recover a $K_{\tilde{b}}$ value which would yield the $M_{\tilde{2}}$ collision. Calculating 2^h possibilities in real-time (online) would be infeasible for the large h (i.e. over 200 bits) of full-length MAC values, such as HMAC-SHA-256.

Let us now consider the offline pre-computation case, where as before, Carol substitutes $g^{\tilde{b}}$ and commits to some $M_{\tilde{2}}$ before Bob releases the actual R_b. Then, for Carol to be always able to subsequently present a $K_{\tilde{b}}$, which would verify successfully, Carol would need to on average pre-compute $2^r \times 2^h$ tuples offline, so as to prepare a database with 2^r lines having the same $g^{\tilde{b}}$ value and the same $M_{\tilde{2}}$ value, and all the different possible values of R_b, with the corresponding $K_{\tilde{b}}$ values. r is the bit length of R_b. While Carol may be allowed to compute the tuples offline, we remark that the complexity involved is very large, considering the length of the full-length MAC values, further strengthened by that of R_a. This computationally complex offline attack, can be mounted in the said manner, against the protocol we presented as Figure 2, in the paper [204], where g^a and g^b were *not* jointly authenticated (for round efficiency reasons), but we believe it is computationally infeasible.

This attack is even harder to mount against our Protocol Trace 4.6, since g^a (contributed by Alice) is part of the input for calculating M_2, and Carol cannot predict g^a accurately prior to the protocol run being initiated, hence Carol is even unable to compute and populate the database offline; she has do it in real-time. (If she wants to pre-compute offline, she needs to further brute-force the key space of g^a, which is large.) Similarly, in our Protocol Trace 4.7, it is infeasible for Carol to pre-compute the database offline, because g^a is jointly authenticated with g^b.

Thus, overall, Attack III is infeasible (for the computationally bounded adversary which we have assumed throughout this dissertation).

4.6 Implementations

To validate the feasibility of our multi-channel security protocols, we have implemented proof-of-concept implementations of the key building blocks (though not full implementations) and we have experimented with two auxiliary channels.

4.6.1 Computing time

We ported the relevant C routines from the Shamus MIRACL [126] cryptographic library to the Symbian OS 7.0s used in several Nokia camera phones, we compiled them with the development environment of Symbian's Series 60 Developer Platform 2.0 SDK, and obtained very usable per-exponentiation timings. The timings for elliptic-curve group key sizes of 160, 192 and 224 bits for the two camera-phones which we have readily available in our laboratory are tabulated in Fig. 4.8. Each measurement was obtained by timing 1,000 exponentiations. Compared to exponentiations, MAC computations clearly take negligible time. On PDAs, laptops and PCs, computations are of course even less of an issue. The cryptographic code was around 17,000 lines of C and, once compiled, occupied just 122 kB, insignificant compared to the phones' 6 and 8 MB of shared memory (further expandable with MMC cards).

The radio messaging (i.e. the insecure open channel) is straightforward to implement using the Symbian Platform Bluetooth API.

Phone/ECC key size	**160 bits**	**192 bits**	**224 bits**
Nokia 6600 (104 MHz CPU)	81 ms	118 ms	160 ms
Nokia 6670 (123 MHz CPU)	68 ms	98 ms	137 ms

Figure 4.8: Computational time per exponentiation

4.6.2 Visual Channel

2D visual code software is now widely available. We experimented with TRIP [57] (which after much further development has become ShotCode [148]), a system from Cambridge University, and also with the open-source SDK offered by Semacode [206], both on PCs and on our two camera phones. TRIP was designed to allow automatic recognition (and estimation of location and orientation) of its circular targets in a video frame of a cluttered scene, whereas the square Data Matrix (ISO/IEC 16022) targets used by Semacode were meant to be acquired by explicitly photographing the tag straight on; for a given pixel size of the acquired target, therefore, Semacode tags, as shown in Fig. 4.9, carry many more bits and are better suited to our security application in which we would want the user to perform an explicit acquisition operation instead of letting the software grab any code it locks onto.

Measurements were taken of the maximum capacity of the phone-to-phone visual channel by encoding progressively longer strings as Semacode tags of increasing pixel count, displaying them on the

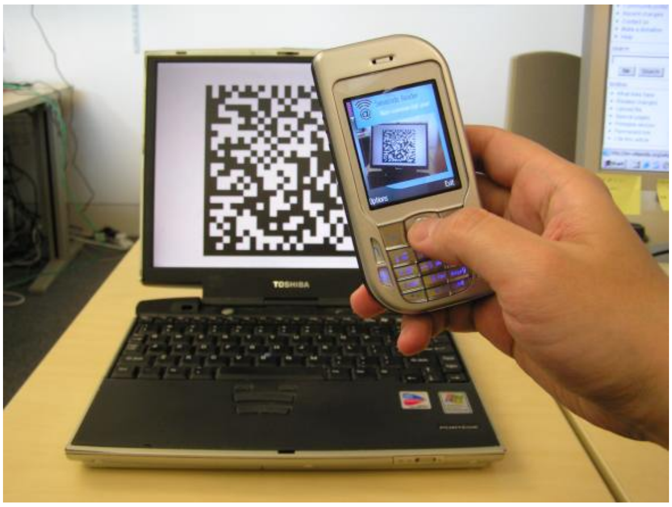


Figure 4.9: Machine visual channel: Semacode transfer from laptop to mobile phone

Nokia 6600, acquiring them with the Nokia 6670 (the device with the better camera) and recording the largest size at which they could be transferred reliably. It took on average 5 seconds to acquire each frame, and this was repeated 10 times for each frame, and we recorded the number of successes for each size. We did not perform extensive usability tests, but it may be expected that a casual user would have a somewhat lower success rate than that which we ourselves had registered. Our results are tabulated in Fig. 4.10. The first column gives the total number of rows and columns in a given Semacode tag. Using the Data Matrix specifications, the data payload supported for each tag size is then calculated by subtracting away the Reed-Solomon error-correction codewords.

The largest frame could be transferred reliably, i.e. with at most one failure in 10 trials, was 14×14 pixels ($12 \times 12 = 144$ non-border pixels, of which 80 were used for Reed-Solomon error correction), carrying 8 codewords and taking 2 further seconds to decode on the 6670 after successful acquisition. With the 6-bits/codeword ASCII-based encoding imposed by the API, which was the default, this meant 48 bits of payload, though theoretically 8 codewords could carry up to 64 bits of payload (at 8 bits/codeword). With today's hardware capability, 48 bits (assuming only a single camera snapshot frame is used, not multiple frames) can be considered to be just about feasible to be brute-forced by an adversary if that were used as a public-key fingerprint, as in the protocol described in Section 4.3.3.2, but are quite safe if used according to our Protocol Trace 4.6. At 8 bits per codeword, 64 bits may not even give very much more margin. The largest frame for which at least one transfer out of 10 still worked was 22×22, carrying 180 bits of payload at 6 bits/codeword.

Under the circumstances, it would seem that the particular limiting factor was not the resolution of the acquiring phone's camera (1152×864), nor the much lower resolution of the source display (176×208), but the absence of a macro facility: at least 8 cm was needed for the 6670 to focus and at that distance the 6600's screen was too small to allow the high density Semacodes to be captured and decoded.

Total row x col	**Success**	**Failure**	**Data Payload (if 6 bits/CW)**	**Data Payload (if 8 bits/CW)**
12x12	100%	0%	30	40
14x14	90%	10%	48	64
16x16	80%	20%	72	96
18x18	70%	30%	108	144
20x20	70%	30%	132	176
22x22	50%	50%	180	240
24x24	0%	100%	216	288

Figure 4.10: Semacode Acquisition: Nokia 6600 Screen to Nokia 6670 Camera

Total row x col	**Success**	**Failure**	**Data Payload (if 6 bits/CW)**	**Data Payload (if 8 bits/CW)**
36x36	100%	0%	516	688
40x40	90%	10%	684	912
44x44	80%	20%	864	1152
48x48	80%	20%	1044	1392
52x52	0%	100%	1224	1632

Figure 4.11: Semacode Acquisition: Computer Screen to Nokia 6670 Camera

It is possible to successfully acquire much larger codes with the 6670, at distances of 15 to 30 cm, from the LCD display of a laptop. This is tabulated in Fig. 4.11. Results from a laptop LCD screen and a 21-inch desktop LCD screen are, for all intents and purposes, very similar. The largest frame transferred here with at most 1/10 failures was 40×40 pixels, carrying 912 bits at 8 bits/codeword, but it took 10 seconds to decode. Such message capacity is sufficient to defeat brute force attacks.

Most Japanese phones are equipped with macro lenses and automatically decode QR Code (ISO/IEC 18004). It can be surmised that if visual codes become widespread, manufacturers would make phones everywhere macro-capable. This would allow the phone-to-phone transfer of visual codes long enough to act as full public key fingerprints, defeating the Man-in-the-Middle attacker without requiring multiple camera frames.

4.6.3 Melodic Audio

We also experimented with transferring a random nonce from one device to another by playing a short monophonic tune. The audio channel provides some integrity, as it is hard for the attacker to interfere without being detected by the human operator. Data origin authenticity is not as strongly guaranteed as by the visual channel (sometimes it is difficult to detect where a tune came from) but it is still better than with radio. There is clearly no confidentiality, since any attacker in range can hear. Speaker and microphone are cheaper than LCD and camera and, especially on the source side, smaller; this may make them more suitable for certain ubicomp gadgets. It is also harder for the operator to miss the fact that a transmission is taking place, which may be good for security but sometimes bad for usability. We attempted to address the latter point by making the tunes more pleasant.

Our prototype algorithms generate 3.5-second monophonies: values are encoded in the pitch of the notes. We did not explore the limits of the transmission range but with commodity hardware (external PC speakers and mobile phone microphones) we repeatably achieved zero symbol errors over a room-scale distance of two metres (no failures over 10 trials, without error correction).

We sought musical guidance and developed three monophony generation algorithms, outlined as follows:

1. Randomly choose notes from an octave.
2. Randomly choose notes, but only from the C major scale.
3. Randomly choose notes as in (2), but additionally, restrict large changes in pitch between consecutive notes.

We also paid attention to reducing the amplitude of the signal ('shaping' the signal) when a preceding note ends and the following note begins in a tune, so as to smoothen the transitions and avoid excess harmonics. In informal usability experiments with 14 human listeners aged 20 to 35, we let the subjects listen to a set of monophonies which were generated by the three algorithms and randomly ordered. The subjects would then give each melody a "pleasantness" score on a scale of 1 (worse) to 10 (best), and they were allowed to replay each melody to listen again if required. In ascending order, the algorithms received "pleasantness" scores averaging 3.1, 4.7 and 5.6 respectively. This revealed a significant preference for the third algorithm.

It also illustrates a not unexpected security-vs-usability trade-off: for the same melody length (i.e. 7 symbols transmitted over 3.5 seconds), our most listener-friendly scheme (i.e. Algorithm 3) gives an entropy of around 15 bits per tune, while the least listener-friendly scheme (i.e. Algorithm 1) provides 25 bits per tune. Human perception of aural pleasantness is quite subjective, and subject to cultural influences. Some listeners may like listening to Western classical music, others may prefer more trendy pop tunes, and there could very well be biases along age-group and gender lines. There would be scope for further research in determining algorithms for generating *high-entropy* but pleasant melodies; this would be an art as well as a science.

4.7 Future Directions

The work which we have done in the past three years in this area has turned out to be well-aligned with practical industry concerns. Various industry consortia have recently released proposals which now make use of various channels, in addition to legacy confidential password channels.

Among them, in 2006 the Bluetooth Special Interest Group issued a “Simple Pairing Whitepaper” [30] for the Lisbon release of Bluetooth: it introduces “Out-of-Band” and “Numeric Comparison” methods that use commitments and visual checks of short codes (in addition to embracing asymmetric cryptography). The “Certified Wireless USB” specification includes an “Association Models Supplement” [82] featuring the “Cable Association” method (contact-based) and the “Numeric Comparison” method (visual comparison, after commitment). The Wi-Fi Alliance has also developed the “Wi-Fi Protected Setup” [4], featuring a network-based “Push Button Configuration” method based on visual comparison as well as optional “Near Field Communication” (confidential short-range) and USB (secure contact-based) methods.

Clearly, advances in hardware have meant that public-key-based key agreements are no longer out of the question for pervasive devices, and would be increasingly required to resist trivial eavesdropping attacks by more sophisticated attackers. Auxiliary channels would be increasingly used to assure authenticity and defeat active attackers.

We believe that multi-channel protocols would be a worthwhile new field of research. We need more precise notions of channel characteristics and the attendant attacker capability. The practice of protocol design would certainly improve in step with these. Further formalizations and complexity-theoretic analyses would help bound the security offered. To help researchers prototype these schemes, software and hardware components [134] which standardize interfaces, would certainly be most useful.

On the usability front, there remains much work to be done to find intuitive and user-friendly schemes to bootstrap security in pervasive environments, from among the diversity of channels and human-mediated action possible. No matter what type of auxiliary channel or human mediation action is chosen, it is worth emphasizing that excessive demands should not be made on the human user; the required user action should be as simple, intuitive and pleasant as possible.

Further afield, the research has raised for us other interesting questions, such as: how far into a device may the endpoint of a channel be considered to really extend – this is relevant for knowing how widely the channel properties hold, and for understanding the assumptions made about the hardware, software and component partitioning; another example: for simplicity we have more or less considered the device and the human owner-operator as as single entity, say Alice, but would there be benefits in making a greater distinction between the two; and yet another question: what if a device supports multiple users (like the shared multi-user workstation in today’s office environment)?

4.8 Summary

Our contributions:

- We highlighted the shortcomings of previous approaches in designing protocols for key agreement, and proposed greater explicitness in describing and using the properties, especially since channel types are often diverse in a ubicomp environment. We call our developments: ‘multi-channel protocols’.

- We showed how existing protocols may be attacked, especially when assumptions regarding the channel properties, and linked to that, the attacker model, are flawed.

- Using channels with authenticity (but without confidentiality), we independently developed a secure two-party key agreement protocol for the ad-hoc scenario.

- For the case where the medium-bandwidth authentic channel available to the protocol designer is merely uni-directional (not bidirectional), we have developed a two-party key agreement protocol with strong security, that is equivalent to having mutual authentication via bidirectional authentic channels. We may well be the first group to propose such a protocol.

- In particular, we have highlighted and showed the powerful use of the authentic single-bit-per-message authentic channel (-which seems to have escaped the notice of other researchers in this field). This advance supports our thesis that explicitness in considering channel properties is of substantial benefit for protocol design.

- We carried out proof-of-concept implementations to validate some of our proposals, and obtained results regarding computational speeds on handphones, operation of the machine visual channel, performance of the melodic audio channel, and some usability issues relevant to these.

We believe that our ‘multi-channel’ way of thinking is suited to (and perhaps overdue for) improving the design of protocols, especially in the ubicomp environment, where there is diversity in the types of channels available, such as infra-red, visual, audio, etc, instead of the channel type being uniform such as in the wireline network case.

Chapter 5

Group Key Agreement using Multi-Channel Security Protocols

5.1 Outline

In very similar circumstances to the two-party case, several participants each in possession of a pervasive device may wish to establish a group-wise secret session key for communications.

For example, a group of people in a face-to-face business meeting wish to protect multicast transmissions among their mobile phones or laptops. Since they are all there together, the problem appears at first trivial. Can't one of them just broadcast a random key to everyone around? No, because people outside the room might overhear it. Can't he write the key on the whiteboard for all to see? No, because firstly we assume that a good key will be too long to transcribe, and secondly the spy with binoculars, or the cleaner, will also learn the key. And also because, in both cases, the key is generated by only one participant.

Our aim is to build a *contributory* (see Section 5.3) protocol that will produce a strong shared key, known only to the people at the meeting, even in the presence of active attackers on the radio channel and passive attackers on the other channels. How can the protocol recognize who is at the meeting (for the purpose of excluding others)? Some previous group key agreement protocols have assumed that all legitimate participants share pairwise keys. Some proved to be vulnerable [58; 151; 152] .

We developed two protocols [203; 204]. Our protocols have no need to recognize pre-established shared keys: they recognize the participants by the fact that they can influence button presses on each other's devices during the protocol's run. They are therefore instances of multi-channel protocols that exploit physical presence, suited to the pervasive computing scenario of an ad-hoc group of human participants equipped with personal devices.

5.2 Related Work

For two participants, the 2-party Diffie-Hellman key exchange is well-studied. Over the years, researchers have made multi-party extensions to DH for group communications [18; 37; 181; 182]. Some use passwords [8] or public keys [103] to bootstrap; others make no assumptions about the topology [91]. But straightforward multi-party extensions to the 2-party DH can turn up subtle vulnerabilities [151; 152].

The use of auxiliary channels in key agreement has been also studied. Balfanz et al [15] assume a high-bandwidth auxiliary channel, and Gehrmann et al [70] assume that the channel is low-bandwidth but confidential. Hoepman [92], and our earlier work [202] refute the common implicit assumption that the auxiliary channel is confidential. These work on auxiliary channels have covered mainly the 2-party case, and only very briefly the multi-party case [15].

Asokan and Ginzboorg [8] gave a good overview of different topologies for ad-hoc multi-party key agreement, and provided password-based solutions. Their first protocol is a multi-party extension of the 2-party EKE [25], and the group has a 'star' topology. They modified it to make it 'contributory'. The second of their protocols uses a Diffie-Hellman type of key agreement, where the group topology is essentially a linear chain. Their third protocol uses DH multi-party key exchange on a 'Hypercube' [18].

The drawback of using passwords lies not principally in the limits of human memorability, since the password would probably be disclosed to all participants immediately before the protocol is run. The main problem is the presence of an eavesdropper on the channel on which the password is shared, be it a visual, audio or other channel which has no privacy. As mentioned in the preceding chapter, this problem had been raised by Hoepman [92]. This weakness extends likewise to multi-party computations. If the password is compromised, then the password-based protocols are all vulnerable to active attacks. Apart from that, Asokan and Ginzboorg's second protocol is related to *Cliques*, which is susceptible to an interesting generic insecurity, to be considered in a subsequent section.

The work by Creese et al. [50], Nguyen and Roscoe [147] and Valkonen et al. [188] are some which have similarities with our work in multi-party key agreement using auxiliary channels possessing authenticity. For these, and for most of the multi-party protocols in current literature, they assume homogeneity (all devices using the same types of channels). We remark that it may actually be also worth considering heterogeneous channels in a group of diverse devices.

5.3 Security Requirements

We present the definitions of several security properties that are relevant to our discussion on vulnerabilities of group key agreement protocols.

Definition 5.1 : A key agreement protocol is *contributory* if each party contributes equally to the key and guarantees its freshness.

Definition 5.2 : A key agreement protocol is *centralized* if key generation is performed by a single entity; typically the group leader.

Definition 5.3 : A key agreement protocol is *partially contributory* if some operations result in contributory, and others in centralized, key agreement.

(Based on these definitions, the Cliques protocol suite [182] is partially contributory.)

Definition 5.4 : *Implicit key authentication* is the property that one party is assured that no other party aside from a specifically identified party may gain access to a particular secret key.

Definition 5.5 : *Perfect forward secrecy* is the property that the compromise of long-term keys does not compromise past session keys.

Definition 5.6 : *Resistance to known-key attack* is the property that compromise of some session keys does not allow compromise of other session keys, nor allow impersonation by an adversary.

5.4 A Straightforward Multi-Party Extension of Two-Party Diffie-Hellman Protocol: Cliques

The Cliques Group Key Agreement protocol suite was developed by Steiner et al [182; 183] in the late-1990s, and underwent successive refinements. The protocol suite defines an entire set of key agreement for dynamic groups, with an Initial Key Agreement (IKA) stage, and an Auxiliary Key Agreement (AKA) stage to provide for subsequent subgroup and single member operations. It is a Diffie-Hellman 'multi-party extension', as it was reasoned that the 2-party DH problem was well-studied, and the key establishment is contributory.

There are many variants in the Cliques family, some are basic group key agreement (GKA) protocols which are secure against a passive adversary, others are authenticated group key agreement (AGKA) protocols which are secure against an active adversary. In the latter group of variants, the group members are assumed either to initially share strong secure pair-wise secrets with the group leader, or else they initially share pair-wise secrets with all other members.

The topology is basically a linear chain structure, and during the IKA stage, the $n-1$ parties exponentiate (also known as providing the required 'services') certain inputs and send the outputs along to the next group member down the ring. The nth member, who is the group leader/controller, would provide his required 'service' and carry out either a broadcast or $n-1$ unicasts to the same effect. Cliques is actually a *partially* contributory protocol, because not all group operations require new contributions from each member.

Fig. 5.1 shows the message flow topology for a group size of three for the AGKA case at initial key agreement. (Note that the bottom horizontal line is part of a two-headed arrow.)

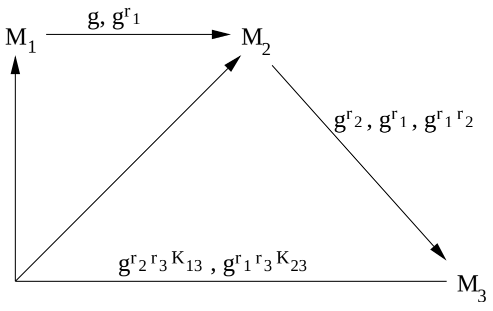

M_1
g, g^{r_1}
M_2
g^{r_2}, g^{r_1}, $g^{r_1 r_2}$
$g^{r_2 r_3 K_{13}}$, $g^{r_1 r_3 K_{23}}$
M_3

Figure 5.1: Cliques A-GDH.2 with 3 Members

We next overview the Cliques IKA computations. Each group member M_i selects a random key contribution r_i. For the AGKA variants, it is also assumed that the group leader, M_n shares with each of the other members, M_i, a pre-established secret key K_{ni}.

Round i $(1 \leq i < n)$:

$$\begin{aligned} M_i \rightarrow M_{i+1} \quad &: \quad \{\alpha^{\frac{r_1 \cdots r_i}{r_j}} | j \in [1,i]\},\ \alpha^{r_1 \cdots r_n} \\ &: \quad C_{i,1}, \cdots, C_{i,j}, C_{i,j+1} \\ &: \quad C_i \end{aligned}$$

Either GKA - Round n :

$$\begin{aligned} M_n \rightarrow All\, M_i \quad &: \quad \{\alpha^{\frac{r_1 \cdots r_n}{r_i}} | i \in [1, n-1]\} \\ &: \quad C_{n,1}, \cdots, C_{n,n-1} \\ &: \quad C_n \end{aligned}$$

Or AGKA - Round n :

$$\begin{aligned} M_n \rightarrow All\, M_i \quad &: \quad \{\alpha^{\frac{r_1 \cdots r_n K_{in}}{r_i}} | i \in [1, n-1]\} \\ &: \quad C_{n,1}, \cdots, C_{n,n-1} \\ &: \quad C_n \end{aligned}$$

Key Computation :

$$\mathbf{GKA} : \mathrm{S_n(M_i)} = \{\alpha^{\frac{\mathrm{r_1 \cdots r_n r_i}}{\mathrm{r_i}}}\}$$

$$\mathbf{AGKA} : \mathrm{S_n(M_i)} = \{\alpha^{\frac{\mathrm{r_1 \cdots r_n K_{in} r_i K_{in}^{-1}}}{\mathrm{r_i}}}\}$$

In round i, M_i would generate and submit to M_{i+1} a set of exponentials — we write these as $C_{i,1}, \cdots, C_{i,j}, C_{i,j+1}$. If these exponentials are concatenated, they can be represented by C_i. In round n, M_n would add its contribution and broadcast the set of partial keys to all M_i's. For the AGKA case, the pairwise keys would be used to exponentiate as well. On receipt of the broadcast message, the $n-1$ members can calculate the group key, S_n as shown above.

5.4.1 Some Generic Attacks on Group Key Agreement Protocol Cliques

Pereira and Quisquater [151; 152] have discovered and proved generic insecurities of Cliques AGKA protocols, whenever the group size is at least 3, using formal methods. They found that the implicit key authentication (IKA) security property, for instance, is not achieved.

Consider a group size of 3. Say, the intruder M_I wants to fool member M_2. M_1, M_2 and M_3 are legitimate participants in the first protocol run, while M_I, M_2 and M_3 are participants in the second run. In the *second* run, M_I replaces the input values of M_3's r_3K_{I3}-service and r_3K_{23}-service with a random value he knows, say α^y. M_3 then broadcasts $\alpha^{yr'_3K_{I3}}$ and $\alpha^{yr'_3K_{23}}$. M_I replaces the input of M_2's r_2-service with $\alpha^{yr'_3K_{I3}}$, then M_2 would send $\alpha^{yr'_3K_{I3}r_2}$. Intruder M_I hears this, and can exponentiate it by K_{I3}^{-1} to obtain $\alpha^{yr'_3r_2}$. He now has possession of a pair $(\alpha^{yr'_3K_{23}}, \alpha^{yr'_3r_2})$. Finally, M_I replaces $\alpha^{r_1r_3K_{23}}$ with $\alpha^{yr'_3K_{23}}$ in M_3's broadcast message in the *first* protocol run. M_2 would be fooled into computing $\alpha^{yr'_3r_2}$ as the group key, which M_I knows, for the first protocol session. M_2 ends up sharing a key with the attacker, hence the IKA property is violated.

Attacks on other security properties were also described by Pereira and Quisquater [151; 152], namely attacks against the perfect forward secrecy and resistance to known key attack properties.

Briefly, in the first attack, say a long-term pairwise key K_{13} is compromised by intruder M_I, and he can replace the input of the r_3K_{13}-service with $\alpha^{r_1r_2}$. When M_3 adds r_3 to the sub-key and broadcasts it, M_I can hear the message sent to M_1, and he can compute the key $\alpha^{r_1r_2r_3}$ established between M_2 and M_3.

In the known-key attack, two protocol runs are required. In the first run, M_I modifies the input of the r_3K_{13}-service to $\alpha^{r_1r_2}$. M_2 and M_3 share the key $k = \alpha^{r_1r_2r_3}$, while M_1 computes the key $k_1 = \alpha^{r_1r_1r_2r_3}$. We assume k is compromised by M_I. In the second run, each member generates new contributions. M_I modifies the input of the r_3K_{13}-service to $\alpha^{r_1r_2r_3}$ (known from earlier), and also alters the cardinal value $\alpha^{r'_1r'_2}$ to $\alpha^{r_1r_2r_3K_{13}}$ (overheard earlier). M_3 then computes the group key as $k_2 = \alpha^{r_1r_2r_3K_{13}r'_3}$, and at the same time also sends M_1 the sub-key $\alpha^{r_1r_2r_3K_{13}r'_3}$, which are equal. M_I hears this, and now can impersonate M_1 to M_3.

5.5 Multi-Channel Security Protocol for Group Key Agreement I

We re-visit the assumptions underlying the Cliques design. It is observed in the AGKA variants that strong pairwise keys are assumed to have been pre-established between the group controller and the $n-1$ members, or even among all members. Applying these keys in Round n is meant to achieve authentication. One may guess that these keys must have been established via authenticated 2-party DH between pairs of members, before the AGKA process.

Despite the presence of these keys, the designers decided not to use conventional cryptography. In retrospect, we consider this is an unnecessary barrier for achieving authentication. Encryption is today not a prohibitively costly operation, and some MACs use cipher algorithms at their core. We highlight the curious situation of not leveraging these keys in conventional cryptography to guarantee confidentiality and authenticity.

Our contribution [203] is to address the vulnerabilities mentioned earlier, with recourse to auxiliary channels. We argue that the auxiliary channels often exist in a pervasive computing environment, though they have often been not well-recognized or well-modelled, but may now be leveraged profitably to bootstrap authenticated group key agreement.

Our approach is to augment both Round i and Round n with auxiliary channels.

Protocol Objective : To assure implicit key authentication, perfect forward secrecy and resistance to known-key attack for contributory group key agreement under conditions of an active adversary operating on the open channel.

The MACs used in our solution are keyed from randomly generated keys. The basic building block is derived from the surprising result of the asymmetric pairing situation given in Protocol Trace 4.5 in the previous chapter. We adapt that to Round i of the original protocol, as shown in Fig. 5.2.

I_i and I_{i+1} are M_i's and M_{i+1}'s identifiers respectively. M_i chooses a short random nonce R_i, a long one-time key K_i, and produces MAC_i based on

$$MAC_i \quad = \quad MAC_{K_i}(I_i \,|\, I_{i+1} \,|\, C_i \,|\, R_i)$$

It is worth re-emphasizing that our protocol is predicated on the following assumptions:

1. Auxiliary channels (such as 'visual' and 'pushbutton' channels), possessing the property of *data-origin authenticity*, exist between group members.

2. The adversary acting on these auxiliary channels is limited to be a *passive* adversary, who can eavesdrop on messages but cannot modify them.

The protocol does not rely on long-term passwords (as in Cliques AGKA) nor the confidentiality of the auxiliary channels (as in MANA III [70]). Values visually exchanged this way today run the risk of being eavesdropped by pervasive CCTVs, as highlighted by Fig. 4.3 in the previous chapter.

#	Ch	M_i	msg	M_{i+1}
1	RF		$- C_i \,\vert\, MAC_i \rightarrow$	
2	PB		$\leftarrow$ ack $-$	
3	V		$- R_i \rightarrow$	
4	RF		$- K_i \rightarrow$	
				Verify MAC_i
5	PB		$\leftarrow$ outcome $-$	

Figure 5.2: Augmented Round ***i***

Re-using the terminology presented in Chapter 3, the column 'Ch' refers to the type of channel utilised. The 'RF' channel has high bandwidth, but is vulnerable to an active attacker, who can eavesdrop on as well as modify messages. The 'V' refers to a low-bandwidth *unidirectional* visual channel of limited bandwidth, commonly found in devices as a screen and keypad, and including two human operators. The 'PB' channel is a 'push-button' unidirectional channel that is allowed to have bandwidth as low as 1 bit, and whose operation is also mediated by human operators. It can use the same 'V' channel too if providing an additional channel is expensive. Under the assumptions, we believe:

Proposition 1 : The advantage of an active adversary who modifies $\{C_i|MAC_i\}$ and attempts to fool M_{i+1} into believing it is from M_i, is of the order of the probability of M_I correctly guessing R_i, i.e. the inverse of R_i's length.

Proposition 2 : The advantage of a passive adversary who attempts to compute the session key from C_i is of the order of the Computational Diffie-Hellman problem on the group.

Thus, without requiring a confidential channel, nor pre-established pairwise keys between members M_i and M_{i+i}, the augmented Round i guarantees the data-origin authenticity of the exponentials. An active adversary cannot re-write a chosen C_i at will, required for an attack.

Proposition 3 : On successful completion of a Round i, M_{i+1} has assurance that the received C_i originates from a human-verifiable member M_i with high probability.

In a similar vein with Fig. 5.2, we can augment the Cliques IKA's Round n with the trace in Fig. 5.3.

Successfully verifying the authenticity of M_n's multicast message requires $n-1$ 'ack' and $n-1$ 'outcome' messages to be properly transmitted and registered via human-verifiable 'V' and 'PB' channels. M_n must wait for all the 'acks' to be received before releasing R_n. This series of protocol steps assure that C_n cannot be modified.

Having data-origin authenticity enforced on the point-to-point Round i messages, and the multicast Round n message, renders these messages unforgeable by Pereira et al's active adversary, and completely

#	Ch	M_n	msg	All M_i
1	RF		$- C_n \mid MAC_n \rightarrow$	
2	PB		$\leftarrow \mathrm{ack}_1 -$	
	$\ldots$		$\ldots$	
	PB		$\leftarrow \mathrm{ack}_{\mathrm{n}-1} -$	
3	V		$- R_n \rightarrow$	
4	RF		$- K_n \rightarrow$	
				Verify MAC_n
5	PB		$\leftarrow \mathrm{outcome}_1 -$	
	$\ldots$		$\ldots$	
	PB		$\leftarrow \mathrm{outcome}_{\mathrm{n}-1} -$	

Figure 5.3: Augmented Round ***n***

foils the attacks.

Proposition 4 : If C_i's *and* C_n cannot be modified by an intruder without detection, then the attacks against the Implicit Key Authentication, Perfect Forward Secrecy and Resistance to Known Key Attack properties cannot succeed with non-negligible probability.

The augmentation of Round n is in fact recommended more for the GKA scheme (i.e. no pairwise keys) than for the AGKA scheme. Doing so yields the twin benefits of saving the computation and latency of at least $n-1$ pairwise key establishment rounds, and transforming an otherwise *unauthenticated* scheme into an *authenticated* scheme.

5.5.1 Costs

Enhancement to security notwithstanding, the disadvantages of the technique include the increased latency per round and increased user intervention. The increased latency is mitigated by the fact that the scaling per round is by a constant factor. The attendant message complexity has been necessarily increased, though this is not usually a significant performance metric.

Topology-wise, the proximity requirements of the auxiliary channels entail that the group members be arranged in a form of a physical linear chain, so it is not just that the flow of group key contributions is in a linear chain. In other words, each successive member M_{i+1} needs to be positioned to be within a human visual range of M_i that allows M_{i+1}'s human-owner to distinguish the visual message transmitted in message 3 of Round i by M_i.

Hardware requirement-wise, as auxiliary channels (such as screen and keypad) are often already present in devices, provisioning this should not be a major barrier.

Strong security can also be achieved via an alternative method which uses private/public key pairs for signature and verification, as described, for example, in the scheme of Katz and Yung [103]. However, this is achieved through higher computational complexity, compared to MAC calculations which are cheaper.

5.5.2 Augmented Group Operations

Group membership is often dynamic. Members can leave or join, sub-groups may leave or fuse. Augmenting with multiple channels allow all the group operations defined in the original Cliques suite, such as member addition, mass join, group fusion, member exclusion and subgroup exclusion, to be essentially retained.

5.6 Arbitrary Topologies

Though the first group key agreement protocols are designed for a homogeneous broadcast medium (such as the Internet, or radio), topology has evolved into a subject of interest because of round efficiency reasons. In multi-channel protocols, we are interested in topology not for efficiency reasons, but for reasons of convenience, since different channels may have different topologies; data-origin authentic channels are often limited in reachability.

5.6.1 Star

Asokan et al's first password-based protocol in [8] uses a star topology in the radio channel. Using multiple channels by overlaying a star topology of bidirectional data-origin authentic visual channels [202], we can do away with passwords and turn this protocol into one which is resistant to an attacker who eavesdrops on the visual channel.

5.6.2 Hypercube

The Hypercube Protocol [18] reduces the number of rounds to approximately $log_2(n)$ where n is the number of members, through having members perform 2-party DH exchanges and successive 2-party exchanges with the resulting key residues. Applying multiple channels to this, each member needs to have $log_2(n)$ bidirectional data-origin authentic channels with his required DH partners. (The Tree Group Diffie Hellman (TGDH) Protocol [106] has somewhat similar topology to Hypercube.)

5.6.3 Octopus

The Octopus topology [18] comprises a d-dimensional hypercube of 2^d members in the centre, with the rest of the members in the 'tentacles'. Applying multiple channels to this, each member in the centre requires a number of bidirectional data-origin authentic channels equal to d plus the number of its own tentacles.

5.6.4 Tree

Burmester and Desmedt's tree-based system [37] builds a group key using just broadcasts, secure against passive adversaries. By applying the Augmented Round n (from Figure 4) to this scheme, we can transform this into an authenticated scheme secure against active adversaries.

5.6.5 Topology Constraints

For the proposed group key agreement protocol run purely on an open radio medium, as long as a roll-call is performed and all members are within easy radio range, it can be seen that the relative spatial arrangement of members is unimportant, and for an given topology, each member's position on it is largely unimportant. But for multi-channel protocols, relative proximity and line-of-sight impact convenience and usability. Thus, it may be surmised that for our first proposed multi-channel group key agreement protocol, it ought to be used together with algorithms that can decide the best topology and how to populate participants as nodes on the topology.

5.7 Efficiency and Usability Considerations

In terms of usability however, there needs to be $n-1$ unidirectional authentic channels (from member M_i to member M_{i+1} where i goes from 1 to $n-1$), and $n-1$ single-bit pushbutton (authentic) channels going in the opposite direction between the same members. Furthermore, there needs to be $n-1$ unidirectional authentic channels, or a single broadcast authentic channel, from the group leader M_n to all the rest of the members, and a single-bit pushbutton (authentic) channel in the opposite direction (essentially meaning a pushbutton on the group leader device).

While the security of the scheme is strong, because the origin of each key contribution is authenticated and verified, the total number of channels required is high. Operating each of these auxiliary channels exacts extra latency, and penalties in terms of convenience and usability.

We consider that it would be possible to make the whole group key agreement more modular, by separating the distribution of key contributions into one phase, and moving the key authentication into a second phase. In other words, we could let participants distribute key contributions and compute the shared key through a non-secure scheme in the first phase, but in the second phase we would verify each member's computed key. We will explore this in the following section.

5.8 Multi-Channel Security Protocol for Group Key Agreement II

Our second proposed scheme [204] runs after the completion of a GKA to allow all participants to check whether they have computed the same group key $g^{r_1 r_2 \cdots r_n}$. In other words, it can be thought of as a group key confirmation round *after* an unauthenticated group key is calculated. Our proposal is shown as Fig. 5.4. All the attacks of Pereira and Quisquater are countered by this scheme.

Topologically, after having used a linear chain for the GKA, for the verification we now have a "star" structure in which the various M_i (for $i \in \{1, 2, \ldots, n-1\}$) only talk back to M_n and not to each other, while the last member M_n broadcasts to all others. For the low-capacity authentic channel (e.g. when M_i takes a picture of M_n's screen), it may be possible to use a broadcast (e.g. by connecting M_n to a projector) but, if not, iterating a point-to-point transmission to each M_i would also work. For the one-bit-per-message authentic channel in the opposite direction (e.g. when the human operator presses the "OK" or "Cancel" button on M_n based on M_i's result), it will be necessary to repeat the point-to-point procedure for each member, as well as keeping track of which M_i is causing this button press.

#	Ch	M_n	msg	All M_i
1	RF		$- MAC_K(I_1, I_2 \cdots I_n \vert F(g^{r_1 r_2 \cdots r_n}) \vert R) \rightarrow$	
2	PB		$\leftarrow \mathrm{ack}_1 -$	
	$\cdots$		$\cdots$	
	PB		$\leftarrow \mathrm{ack}_{\mathrm{n}-1} -$	
3	Auth		$- R \rightarrow$	
	$\cdots$		$\cdots$	
	Auth		$- R \rightarrow$	
4	RF		$- K \rightarrow$	
				Verify MAC
5	PB		$\leftarrow \mathrm{outcome}_1 -$	
	$\cdots$		$\cdots$	
	PB		$\leftarrow \mathrm{outcome}_{\mathrm{n}-1} -$	
6	PB		$- \mathrm{outcome}_{\mathrm{n}} \rightarrow$	

Figure 5.4: Multi-Channel Group Key Agreement II - Validation of Computed Key

First, M_n broadcasts a keyed commitment $\mathrm{MAC}_K(I_1, I_2 \cdots I_n | F(g^{r_1 r_2 \cdots r_n}) | R)$ to all M_i (where $i \neq n$), using the high-capacity channel vulnerable to the Dolev-Yao attacker. Here $F()$ is a pseudorandom key derivation function, R is a short random nonce and K is a long random MAC key. Then, after all the M_i devices acknowledge reception of the broadcast (using the authentic one-bit-per-message channel), M_n releases the nonce R and the key K to all M_i, over the visual and radio channel respectively. Each M_i then recomputes the commitment, verifies whether it matches the one received from M_n and reports the answer to M_n over the one-bit authentic channel. Finally, after receiving everyone's reports, M_n tells everyone, over an authentic channel (which needs only carry one bit but could be the visual channel, because it can do broadcast, instead of the pushbutton one), whether they all reported success or not. If even a single verification fails, the protocol will abort.

5.8.1 Advantages - Efficiency and Usability

Our re-use of our earlier construction make this protocol slightly more efficient than any other comparable one we found in the literature, including that of Valkonen et al. [188]. Compared to their proposal, our proposal uses fewer hash computations, and fewer radio messages. Comparing our second proposal (this Section 5.8) with our earlier proposal (Section 5.5), the second proposal does not require point-to-point authentic links *between* members, other than authentic group-leader-to-group-member links. Our protocol proposal illustrates the usefulness of making channel modelling explicit in protocol design.

Note that, in message 3, if $n-1$ unicasts are used instead of one broadcast, these unicasts don't necessarily all have to take place over the same type of authentic channel. If the M_i devices are heterogeneous, they may each use their preferred auxiliary channel (camera, keypad, audio, near-field, contact etc.), so long as it meets three requirements: it offers data origin authenticity, it offers sufficient capacity to transmit R, and M_n can act as a source for it.

5.8.2 Security Analysis

In this second proposed protocol, we assume that group members are motivated to indicate an error if their calculated MAC output does not verify correctly. In other words, as a group member, if you found that your MAC output calculated before executing Step 5 does not match the copy you have received in Step 1, then you would raise an alarm. You would not keep mum and indicate success, otherwise you would be essentially shut out of the encrypted communications which would take place later.

For the active adversary who substitutes the MAC value destined to a particular member M_i in Step 1, his probability of success in getting the MAC to verify correctly later is around $O(2^{-r})$, where r is the bit length of R. In other words, it is a one-shot guess at the value of R. (This is related to that in Protocol Traces 4.6 and 4.7 in the previous chapter.)

Suppose the active adversary modifies copies of the MAC value sent to x number of members. He does not improve his chance of success by varying his guess of the R value across these x members. This is because *all* of the $(n-1)$ MAC verifications have to succeed for the whole protocol run to indicate 'success' to the legitimate participants; if even a single verification fails, the protocol run will indicate 'failure' and simply abort.

Therefore, the performance of the attack against the protocol is certainly not equal to:

$$\sum_{i}^{n-1} Pr(verification\ success\ with\ M_i)$$

Instead, the performance of the attack is, roughly speaking, equivalent to:

$$\prod_{i}^{n-1} Pr(verification\ success\ with\ M_i)$$

It is to be noted that we make the assumption that the group leader is honest, which is reasonable in a real-world situation. If the group leader is actually dishonest, he may send different MAC values and K values to one or several group members, such that they could verify correctly their MAC calculated outputs when using the correct revealed R as input, even though their copies of the computed group key may be incorrect. This can happen if the group leader is malicious, he controls the insecure (radio) channel, and he wants to partition the group of members into sub-groups with different computed group keys. We assume the group leader is typically chosen with more care, so this potential vulnerability is of little concern.

5.9 Future Directions

Work remains to develop provable security arguments for the group setting of multi-channel security protocols (i.e. security proofs). Since more than two participants are involved in such multi-party protocols, we would reasonably expect that the proof strategies would be more complex than for the two-party case. We have already previously seen how the original Cliques protocol suite (which was designed for just a homogeneous network environment) was finally found to have subtle security flaws, years after it was proposed – secure multi-party protocols are not straightforward extensions of two-party protocols. Thus, we think that there might yet be surprises ahead when researchers investigate more carefully the security of multi-channel protocols for groups.

In the meantime, we believe that with our proposal(s), it would be feasible and usable to bootstrap a group secret key among multiple devices utilizing multiple auxiliary channel types, which is at least secure on the basis of heuristic security analysis.

Just as in the two-party case, and perhaps even more so for the group case, much usability studies remain. It would be necessary to determine how groups of individuals would interact and what would be the most user-friendly way to bootstrap a group key in a ubicomp multi-channel environment.

5.10 Summary

Our contributions:

- Using ideas first advanced in the previous chapter for two-party multi-channel security protocols, in this chapter we present two protocols to bootstrap key agreement in a group of ubicomp devices.

- Our proposals help address recently found security flaws in an existing group key agreement protocol suite, namely Cliques. The protocol suite can be strengthened by retro-fitting with the suggestions in our first protocol proposal, which authenticate the origins of key contributions at each and every round.

- We have shown how our first protocol proposal directly helps strengthen group key agreement protocols with linear-and-star topologies such as Cliques. Our protocol proposal can also be easily adapted to other group key agreement protocols having topologies *different* from Cliques, thereby strengthening them against active attacks.

- Separately, instead of authenticating the origins of key contributions at each round, it is in fact possible to obtain security by confirming that all members have computed the same group key in only a final round; we present this in our second proposed protocol. We further suggest that diverse, heterogeneous auxiliary channels (eg. visual, audio, NFC, etc) can be simultaneously used in one protocol run for a group of ubicomp devices to bootstrap such a group key.

Thus, we have shown that, by using low-bandwidth authentic (not confidential) channels, it is possible to form a group key securely, overcoming subtle flaws in a earlier group key agreement protocol. The clarity and insights provided by our multi-channel viewpoint allowed us to develop a second group key agreement protocol that is more efficient than the other protocol proposals that we have seen.

Chapter 6

Location Privacy at the MAC Layer

6.1 Outline

As gadgets become more networked and pervasive, they are able to collect, store, aggregate and disclose information about human subjects. The preceding chapters have discussed authenticity in key agreements, so as to produce secret keys to guarantee confidentiality during communications. Confidentiality protects the contents of the messages. However, to an adversary, it is often not necessary to learn these contents. The mere fact that communications is taking place between two identifiable parties and has become known constitutes a loss of security. The adversary can perform traffic analysis and thence *infer* some information from the link data and other meta-data.

Privacy is the security property concerned with protecting the identities of the communication parties from disclosure. In traditional notions of privacy, the problem is often specified in terms of message sender (or source) anonymity and recipient (or destination) anonymity of the communicating parties. In a wireline network, the locations of the sender and recipient tend to remain quite constant, and the network topology is often quite static, such that the routes of the messages would be rather long-lived. This is not the case for mobile nodes. In this chapter, we are more interested in a particular kind of privacy, namely location privacy. The addition of location considerations to the notion of privacy comes into play for subjects who are mobile. Location privacy is an issue in a world of mobile cellular coverage, i.e. where every human being carries a mobile phone. Location privacy will only become a bigger concern with the advent of pervasive computing.

Pervasive computing envisages an era where the average human user owns tens, if not hundreds of computing devices, which would be inter-connected. For these devices to make maximum impact in improving the human quality of life, they would be required to collect and analyze contextual information on-the-fly, to adapt the devices' behaviour. Location information is one type of such contextual information which is taken on board. As devices with improved information processing capabilities and improved storage capabilities become increasingly built and deployed, they can well be instead misused for surveillance purposes of human subjects. As security engineers, the onus is on us to develop deliberately ubicomp systems with inbuilt privacy-enhancing characteristics, because firstly, without a deliberate focus on security, new systems that are produced would tend to re-use flawed assumptions from earlier generations of less-networked systems, and secondly, it could be harder to integrate privacy protections as an afterthought than if it has been designed in from the outset.

In this chapter, we describe our contributions for making new mobile and pervasive technologies more location-private. Conceptually, we outline the idea that location privacy is a multi-layer problem. At the data link layers [1] and above, pseudonyms do help in assuring privacy for the communicating entities. Specifically, we propose a better way of managing pseudonyms for stateful device pairing and a protocol [201] associated with this, for a pervasive wireless technology (namely Bluetooth), which improve upon earlier proposals by other researchers. We will overview some of the existing location privacy problems in the technology.

In a subsequent chapter, we will proceed along the same theme of multi-layer location privacy, and consider location privacy in terms of the *physical layer*.

We begin here by overviewing related work, the current legal environments of privacy and wireless location privacy, and defining some properties related to location privacy.

6.2 Related Work

Ubiquitous gadgets have been steadily proliferating, posing an increasing threat to personal information privacy. The types of ubiquitous devices may be simplistically arranged on a spectrum according to their intended pervasiveness (which is often negatively correlated to the device hardware capability).

On one end we would have the personal cellphone, which we may find one apiece for each individual person, where each cellphone is identifiable and traceable by its network. At the other end, we may find by the hundreds RFIDs—passive radio tags returning unique IDs (i.e. Electronic Product Code or EPC), for which the length of the most prevailing version is 96 bits. Privacy solutions for cellphones include the use of network-issued temporary pseudonyms - the ‘Temporary Mobile Subscriber Identity’ (TMSI), and ways to manage these [105; 192]. RFIDs are already found in many current applications, such as automobile immobilizers, animal tracking, payment systems, automatic toll collection and inventory management. Garfinkel et al. [69] have categorized the corporate data security threats, personal privacy threats, and cloning threats – the last of which are more to do more with payment fraud than privacy though. The first two of these are closely related to, but still different from, our targeted problem domain for wireless ad-hoc technology. Solutions for RFID privacy [69; 101] include ‘killing’ the tag upon purchase of the attached item, or enclosing it in a mesh, or changing its ID by ‘re-encrypting’ with an external agent, etc. Most of these RFID privacy solutions, especially for RFIDs used in inventory control,

[1] According to the OSI Network Model, the Data Link layer provides the functional and procedural means to transfer data between network entities and to detect errors in the Physical layer. The layer arranges bits from the Physical layer into frames. It also does flow control and frame synchronization. The layer is sub-divided into the Media Access Control (MAC) sub-layer and the Logical Link Control (LLC) sub-layer. The MAC sub-layer manages protocol access to the physical network medium. The IEEE MAC specification defines MAC addresses, which enable multiple devices to identify one another at the Link layer.

are less suited for the Bluetooth type of usage. In RFIDs, the hardware constraints are more restrictive, and also, it is usually assumed that only the RFID tag has a pseudonymous ID (such as in [90]), and not the RFID reader.

We propose that short-range ad-hoc wireless technologies, such as Bluetooth, which lie in the middle of the spectrum (of pervasiveness) among ubiquitous devices, lend themselves to a different solution framework. Bluetooth devices have finally appeared in large numbers in recent years after initial problems, and have gained market acceptance and general user familiarity. We feel that this generally well-tested and well-received pervasive technology can serve as a platform to design for location privacy, instead of considering any hypothetical short-range wireless technology still on the drawing board whose niche is not clear yet. Location privacy problems in Bluetooth have been pointed out by Jakobsson and Wetzel [100], and an initial set of MAC layer solutions using pseudonyms had been suggested by Gehrmann and Nyberg [71]. This chapter will cover our proposal to work towards a more refined privacy solution framework for Bluetooth at the MAC layer. Singelee and Preneel [176] also have a set of proposals, for a more general case of wireless personal area networks, in circumstances where the two devices pre-share different types of information.

Gruteser and Grunwald [84] have proposed the use of pseudonymous MAC addresses for Wireless LAN. Beresford and Stajano [27; 28] suggested the use of mix zones with pseudonyms, but this is for resisting adversarial application providers, which is a different context from what we are primarily concerned with in our own MAC layer solution. Kohno et al. [112] demonstrated the very significant result of being able to fingerprint devices remotely; this relies on TCP timestamps, and operates at the transport layer.

6.3 Viewpoints on (Location) Privacy

6.3.1 Legal

The United Nations Universal Declaration of Human Rights [144] states in its Article 12 that ‘No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.’

The compliance of the government of the country in question, to this Declaration, in its spirit and letter, of course varies widely.

It is not unknown, of course, for this ‘right’ to be selectively revoked for an individual who is under suspicion of questionable activities. For example, the compromise of the location privacy of GSM handsets is used [78] to help track down suspects by law enforcement forces.

6.3.2 Technical Community

The Internet Society has issued a ‘Geopriv Requirements’ memo [53]. It is an attempt to inform the technical community of the need to assure privacy when handling location information, needed by location-based services and applications. It does not specify a standard of any kind, but is a creditable step towards formalizing a set of requirements and common terminology.

6.3.3 Individual

Recent studies [54; 56] of moderate scope have revealed the monetary price that individuals are willing to pay for location privacy. Using auction techniques, Danezis et al. [56] found that experimental subjects have submitted a median bid of 10 GBP for allowing their locations to be queried every few minutes 24 hours a day for 28 days through their mobile phones, to the resolution of a cell. This bid value was considered to be neither too low, nor too high. At this moment, it appears that location privacy is still a relatively unappreciated concept in the eyes of the general public. Perhaps, when major compromises in location privacy do take place, which make big news, that will raise the profile of the issue of location privacy, and cause the thence better-informed general populace to begin to price location privacy more highly.

6.3.4 Adversary

Depending on how fine-grained one prefers to get, there are several sub-categories of personal location privacy threats [69], such as association threat (i.e. a user's identity becomes associated with an item which leaks an identifier), constellation threat (i.e. a group of devices signifies some individual), and breadcrumb threat (i.e. a discarded device becomes unfairly linked to an individual when it is used at a crime scene): these particular threats are applicable to mobile devices besides RFIDs.

Often, the adversary is modelled as a *Big Brother*, such as an over-intrusive government, or corporation. Or the adversary can be modelled as what can be colloquially called *Little Sisters* - distributed adversaries each of whom does not have a global coverage, but shares information with each other. In terms of motivation, it is conceivable that criminal elements, purely profit-driven commercial entities (e.g. insurance companies, paparazzi) and private interests (e.g. a jealous spouse) can also have an interest in one's location information.

6.4 Definitions

There are different components to the notion of *privacy*. The Common Criteria [1] is an international standard (ISO/IEC 15408) for computer security. The Common Criteria analyzes privacy into anonymity, pseudonymity, unlinkability and unobservability.

Definition 6.1 : *Anonymity* deals with whether a subject may use a resource without disclosing the user identity.

Definition 6.2 : *Pseudonymity* makes a user accountable for the use, without disclosing his identity, by providing an alias.

Definition 6.3 : *Unlinkability* ensures that a user may make multiple uses of resources or services without others being able to link these uses together. It attempts to obscure the relations between actions by the same user.

Definition 6.4 : *Unobservability* ensures that a user may use a resource without third parties being able to observe that it is being used. For example, a broadcast obscures from third parties who actually received and used that information.

The word *traceability* is sometimes used by some authors. However, we will stay with ‘privacy’, for consistency. Pfitzmann et al.[153; 154] have given finer-grained definitions of the above terms and their inter-relationships.

Here, we are more precisely concerned with *location* privacy of mobile devices/principals, and not privacy per se, though these are closely related. We will use the succinct definition from Beresford and Stajano [27] below.

Definition 6.5 : *Location privacy* is the ability to prevent other parties from learning one’s current or past location.

Sub-components of privacy, such as anonymity, pseudonymity, unlinkability and unobservability, from the Common Criteria, can be thought of as being thence inherited by the concept of location privacy. In other words, it would be sensible to speak of ‘location unlinkability’.

6.5 Privacy Within and Across Multiple Layers

A broad thread which emerged from our research is the observation that not only is it important to design for privacy *within* each layer of the communication stack, it is also important to design for privacy *across* different layers. These two points emerged when we considered the question of location privacy in Bluetooth, a current technology which typifies pervasive communications. We found that Bluetooth device identifiers at the Medium Access Layer are intertwined with the lower layers, i.e. the baseband and the physical layers. In attempting to propose an improvement to Bluetooth anonymity at the MAC layer, we had no choice but to also consider the impact any changes may have at the lower layers. That in fact, also illustrates that security, in this case privacy, ought to be designed in at the outset, because it can be hard to add security improvements as an afterthought to an existing communication stack, especially one which does not sufficiently isolate identifiers from different layers.

Several observations have emerged from our research:

(1) Location privacy is a multi-layer problem. Identifying information can leak at various layers, because other entities in the network (and adversaries) may interact with our device at different layers of the protocol stack. This has the following obvious, and non-obvious, implications.

(2) To solve the location privacy problem well, it is necessary to address it by solving it *separately* at each layer. In other words, each layer would require a different solution, suited for that layer. (Avoine and Oechslin [11] have also suggested this point, in the context of RFIDs.)

(3) Next, to solve the problem well, it is essential to *de-link/de-correlate* identifiers from different layers, from one another. This is important so as to prevent an adversary from mounting successful *correlation attacks across layers*.

(4) Lastly, it is necessary to solve the location privacy at *all* the layers of the protocol stack, which a device may conceivably expose to a potential adversary.

The rest of the material in this chapter, (i.e. Chapter 6) and that in Chapter 7, support Point 1 above. Later in this chapter, by describing our improved solution to a MAC layer privacy-enhancing system, we will show that we are tackling Point 2. Point 3 is an issue which we state as a concern, but which we do not actually cover in detail, and for which we do not propose a specific solution – it is a non-goal in our research; we will assume it has been addressed within the device or application concerned. The extension of work in this chapter into the thread of work in Chapter 7, indicates that we are moving in the direction commended by Point 4; we do not, however, claim to have completely covered the problems in all the layers by merely covering the MAC and (parts of) the physical layers.

6.6 MAC Layer Vulnerabilities of a Pervasive Technology

Current well-known authentication weaknesses in Bluetooth [79; 80; 81] could be relatively easily resolved by recourse to asymmetric key establishment techniques [71; 205] at the cost of slightly increased computation. These enhancements would defeat even a strong adversary, by which we mean one which is omnipresent, has significant computational resources, and is able to mount active attacks.

In comparison, it is generally difficult to secure privacy, including location privacy. Awareness to Bluetooth's vulnerabilities in this area was first raised by Jakobsson and Wetzel [100]. Each Bluetooth device is identified by a unique permanent 48-bit Bluetooth Device Address (BD_ADDR). As Bluetooth is usually attached onto personal devices, the detection of a particular BD_ADDR in the neighbourhood would suggest that a particular human operator is nearby (i.e. association threat, in Section 6.3.4). That individual may even be carrying multiple Bluetooth devices and, if such a cluster of BD_ADDRs is detected, it is highly probable that the individual is nearby (i.e. constellation threat, in Section 6.3.4).

Furthermore, the device's BD_ADDR is used as an input into many procedures in Bluetooth. It is deeply entangled into certain parts of the protocol stack, and it is difficult to engineer it away easily, showing the general difficulty of providing security as an afterthought. This incidentally relates us back to our Point 3 in Section 6.5.

We will provide an overview of aspects of the Bluetooth radio and baseband layers, which are cause for privacy concerns. They can be summarized into:

1. problems of discoverability
2. problems of the non-discoverable mode
3. disclosure of the identity in certain packets
4. derivation of the access code from the identity
5. derivation of the frequency hop set from the identity

A *mode*, as described in the Bluetooth Generic Access Profile (GAP) [79; 80; 81] , refers to any particular set of directives that defines how a device will respond to certain events. Three types of modes are defined in the GAP; they are discoverability, connectability and pairing related, but only the first two are more immediately related to location privacy. They are shown in Fig. 6.1 below.

Mode	**Setting**	**Actions**
Discoverability	Non-discoverable mode	Unresponsive to inquiry
	Limited discoverable mode	Can enter Inquiry Scan state and respond to limited inquiry
	General discoverable mode	Can enter Inquiry Scan state and respond to inquiry
Connectability	Non-connectable mode	Unresponsive to paging
	Connectable mode	Can enter Page Scan state and respond to paging

Figure 6.1: Modes with direct effect on Bluetooth location privacy

Note that a device enters the Inquiry state for the purpose of discovering remote devices (which in turn must be in the Inquiry Scan state to be able to respond). A device enters the Page state if it needs page for some already discovered remote device for the purpose of establishing a connection.

We provide a survey of the first three of the above problems, which are well-known. The last two problems have been raised before partially, but we analyze and quantify more fully the risks involved.

6.6.1 Problems of Discoverability

The purpose of discovery is to allow one to find devices which one has not encountered before. The inquiry scan mode is also known as discoverable mode. The discovering (or inquirying) device sends ID packets, which contain just an access code — either the General Inquiry Access Code (GIAC) or a

Dedicated Inquiry Access Code (DIAC), and according to the requisite inquiry hop sequence. A device in the inquiry scan mode will respond to inquiries with a Frequency Hop Synchronisation (FHS) packet, disclosing its own BD_ADDR and CLKN (native clock). The response is not immediate, but is on receipt of the next packet, so as to avoid collision with other slaves.

Essentially, the discovery process enables a hitherto stranger device to be found after at most tens of seconds, and from the privacy perspective the real identity is unfortunately disclosed when a device is discoverable. Keeping devices constantly discoverable is clearly a privacy risk. It is advisable to turn off discoverability whenever it is not needed.

6.6.2 Problems of Non-discoverable Mode

Devices which are set to 'non-discoverable' are nevertheless responsive to some degree. If they are set to 'connectable', they can still be detected, due to privacy weaknesses in the page and page scan states. During page, the master device will page for another device using an ID packet containing a Device Access Code (DAC) derived from the Lower Address Part (LAP) of the latter's BD_ADDR. The hopping at the physical layer during page is similar to the case for inquiry. The hop sequence is derived from the DAC, instead of the GIAC or some DIAC, together with the estimated clock (CLKE) of the paged device. When the slave detects the page message containing its own DAC, it will reply with an ID packet containing the same DAC. After that, the master will transmit an FHS packet.

Thus, a slave device set to non-discoverable and connectable will not respond to inquiry messages, but it will respond to page messages containing its permanent DAC. Devices which have previously encountered this device and have a record of its BD_ADDR and/or DAC can still page for it successfully if the device is within radio range. If its BD_ADDR is not known, the discovering devices can conduct a brute-force search of the BD_ADDR range, or more precisely, the 24-bit LAP range. The only means of protection against being tracked this way possible under the current specification are to either turn off Bluetooth, or to switch to non-connectable mode if such fine-grained control is supported on the particular device, and lastly, to reduce occurrences of pairing to a minimum so as avoid over-exposing the device's BD_ADDR.

6.6.3 Disclosure of Identity in FHS packets

The FHS is a special control packet. The entire BD_ADDR of the sender, comprising the Lower Address Part (LAP), Upper Address Part (UAP), and Non-significant Address Part (NAP), are disclosed in the FHS packet, together with the highest 26 bits of its 28-bit native clock CLKN. The FHS packet is sent on two occasions: by a slave device in inquiry scan mode responding to an inquiry; and by a master device in page mode responding in turn after a slave in page scan mode has responded to the page. The device's identity is hence revealed to the opposite party and to any eavesdropper who is monitoring the spectrum.

6.6.4 Baseband Access Code Derived from Identity

The derivation of the Channel Access Code (CAC) from the master device's LAP had been recognized by Jakobsson et al [100] as a privacy risk, because the LAP can be reverse-engineered. We generalize further that the derivation of not just the CAC but also the DAC from the LAP carry privacy risks.

The access code of a Bluetooth packet is either one of three types. The CAC is used during the Connected state, the DAC is used during page and page scan, and the Inquiry Access Code (IAC) is used during inquiry and inquiry scan. The sync word is a 64-bit code derived from the LAP of a BD_ADDR. In the CAC, the sync word is derived from the master device's LAP; in the DAC, the LAP of the paged slave unit is used; and in the IAC, either the single reserved LAP is used or else certain dedicated IACs are used. The inquiry state is of less interest for privacy because the IACs being correlated for are not too device-specific.

The attacker only needs to compute once a dictionary of 2^{24} (ie. 16.7 million) LAP entries and their corresponding 64-bit sync words. As raised in [100], when the attacker detects a CAC, he can perform a table lookup and learn the master device's LAP. For completeness, we further raise that when this attacker detects a DAC sent by a paging master and a responding slave, he can perform a table lookup using the same pre-computed dictionary, and learn the slave device's LAP. As such, the slave device, non-discoverable but connectable, also faces location privacy risks. Note that a particular LAP is not unique, though collisions would be rare. The remaining address bits — the 24 bits of the UAP and NAP which constitute the 'company_id', do not span the entire 24 bits of entropy — the allocated numbers are published by IEEE Standards, and as of Jan 2005, there were only around 2^{13} issued numbers.

6.6.5 Hop-Set Derived from Identity

Jakobsson et al [100] observed that since the hop sequence in a connected piconet is a function of the master device's BD_ADDR and CLKN, and thus if one can capture an FHS packet sent by the master, the hop sequence can be trivially calculated. We investigated the reverse attack—the more difficult one of how to recover the master device's address by tracking the frequency hopping pattern if we failed to capture the master's FHS packet, and we found that collecting just 6 packets, along with other information, is adequate. This attack produces 28 bits of address — 4 more bits than attacking the access code.

Bluetooth uses frequency-hopping mainly to mitigate environmental interference, and to reduce collisions among different piconets. There are five types of hopping sequences for the 79-hop system, one type each for the inquiry, inquiry response, page, page response and connected states. Each of these sequences is determined by the 24-bit LAP and the lower 4 bits of the UAP, of the relevant device's BD_ADDR, and its clock. The choice of device address used here is identical to that used to compute the access codes for the different states.

Thus 28 bits of LAP/UAP and 27 bits of the clock go into the hop selection box at any one time to choose one frequency. This function is fully documented in the specification and is strictly surjective.

In the connected state, the output selects one of 79 frequencies, corresponding to an approximate 6-bit range. Based on a reasonable assumption of a uniform distribution, thus for the same clock offset, roughly 2^{22} LAP/UAP values would result in the same frequency.

We can carry out the following attack. Capture a first packet and form a tuple of the clock and frequency. Do a brute-force search and narrow the set of 2^{28} LAP/UAP values into a set of about 2^{22} possibilities. Collect another packet and obtain the tuple. Assuming uniform distribution, we can narrow further to a set of 2^{16} possibilities. Continuing in a similar way, just 6 packets in total are required to determine a unique 28-bit LAP/UAP with a probability calculated at 99+%. (This is described more fully at the end of this section, Section 6.6.5.) The overall work factor is on the order of 2^{28}. With so few packets required for successful attack, the attacker may simply listen at a fixed frequency for it to be re-visited, instead of scanning the entire band. We have to add a caveat that, since the clock setting at each packet is required, determining the master device's clock setting initially without recourse to capturing its FHS packet would entail an indirect route of obtaining a LMP (Link Manager Protocol) packet containing the slave's clock offset relative to the master's, and inquiring the slave (which needs to be discoverable) to learn the slave's clock.

This novel attack shows that even if a master device is non-discoverable and non-connectable, its hop pattern in a connected state and a discoverable slave could betray its identity.

Bluetooth was not expressly designed to be resistant to interception and deliberate narrowband jamming, unlike, for example, military tactical communications. Our interest with the frequency hop in Bluetooth is on the anonymity issues rather than availability. By resource-sharing the radio access via different clock offsets and public long-term identifiers, frequency hopping achieves equitable allocation of the spectrum and reduces collisions, but it hurts privacy. To improve privacy, the options are: either to disentangle the identifier from the time-frequency allocation, thereby requiring a re-design of the radio layer; or else to just de-link the identifier from the long-term identity, which is simpler.

Recovery of Address Bits from Frequency Hop

The mathematics of the method can be formulated as a binomial distribution. We assume that each of the 79 outputs is equi-probable. We want to find the probability that after k rounds, only one input is left, ie. all of the other $2^{28}-1$ inputs are discarded at some round. Each one of these remains with probability $(1/79)^k$. Assuming independence of clock values, and independence between the outcomes of different inputs, the probability we seek is

$$(1+x)^n = (1-(\frac{1}{79})^k)^{2^{28}-1}$$

As the exponent is large, the numerical result is difficult to compute. Since x is small with respect to 1, we can do a binomial expansion.

$$(1+x)^n = 1+\frac{nx}{1!}+\frac{n(n-1)x^2}{2!}+\cdots$$

For $k = 6$, the first two terms sum to 0.9989. If we approximate $1/79$ to $1/2^6$, the result is 0.9961.

6.7 MAC Layer Privacy-Enhancing Solutions

6.7.1 Adversary Types

We identify two classes of adversaries, in ascending order of capability to compromise the privacy of the Bluetooth device.

Attacker Type I. The first class of attackers use commercial Bluetooth devices that can inquire and page as usual, and can therefore find any discoverable Bluetooth devices, as described in Section 6.6.1. The attacking range may be extended by directional antennae, a concept well-known to EM/RF engineers. For example, a 18 dBi yagi antenna can boost the 100 m Class I Bluetooth range to around 900 m, and a 24 dBi antenna to 1.6 km, assuming low RF losses at the joints, though such antennae are large and obtrusive. Within this class of attackers, we can distinguish a slightly more sophisticated sub-class, who can conduct brute-force searches of the BD_ADDR space, or rather, the LAP space, so as to find connectable victim devices, as described in Section 6.6.2. Such proof-of-concept code has been released [198], though it is estimated to require around 11 hours to conduct a complete search of the space, using 127 devices working in parallel. We have developed our own version of this attack using a shell script, the open-source BlueZ stack, and an ordinary Bluetooth USB dongle.

Attacker Type II. A second class of attackers uses radio receivers, or modified Bluetooth devices, which are not constrained to frequency hop.
(II.a.) The first sub-class can listen on one selected channel continuously for all types of messages in the inquiry, inquiry scan, page, page scan, and connected state hops. If this attacker sees a CAC or DAC, he can carry out his table lookup privacy attack, as described in Section 6.6.4. If he sees an FHS packet, then he has learnt the full BD_ADDR, as described in Section 6.6.3. He can also derive the master's identity by knowing at which clock offsets a particular hop frequency is re-visited, and by probing a discoverable slave, as described in Section 6.6.5.
(II.b.) Another more powerful sub-class is capable of listening on the entire 2.4 GHz band simultaneously. This attacker is less likely to miss any packets, and is more effective than the first sub-class in determining the CLKN of the target master device for the attack in Section 6.6.5. Attacking the access code is less costly than attacking the frequency hop pattern though. The first sub-class of attacks can be readily demonstrated with today's Bluetooth protocol analyzers, such as the Frontline-Tektronix BPA-100 and 105.

We distinguish between hardware, and do not distinguish between the cryptographic capability among the classes, because programs which do such computations can be commoditised easily and can run on generic PCs. The first category of adversaries are able to successfully compromise the privacy of today's Bluetooth devices easily, unless tight discipline is maintained over the use of the discoverable mode and connectable mode. The second category of attackers is able to compromise the privacy of Bluetooth devices even when their victims maintain tighter discipline over discoverability and connectability, and whenever devices are transmitting in a connected state. The overall efficacy of location privacy attacks also depends on the pervasiveness (and investment) of the attackers, and how effectively they can correlate and fuse information obtained by their various spatially distributed sensors to continuously track the location of their victims.

6.7.2 Location Privacy Goals

The current specification of Bluetooth does not support strong location privacy. Before we go into the detailed technical mechanisms, we need to define the usage scenarios for this short-range wireless connectivity technology. Then we will articulate the privacy goals which take into account the usage.

Little location awareness required. Bluetooth-equipped devices tend to talk to other personal devices, and less with fixed immobile network infrastructure. The interaction is mostly peer-to-peer. Users of Bluetooth do not seem to require it to have substantial location-awareness for it to work well for cable-replacement. This is contrasted with applications such as the Active Badge [194], which was designed precisely for tracking purposes. A higher application layer may require location-awareness, but Bluetooth, as a connectivity layer, does not require location-awareness built-in, and can very well lean towards the location-private part of the continuum. These differences make its location privacy requirement different from other technologies which have been analyzed elsewhere, which had assumed a network backbone [27]. We admit that the security interaction of Bluetooth with the location-aware parts, where present, of the host device may merit further study.

Lightweight cryptography allowed. On the other hand, devices hosting Bluetooth are rather much smarter than dumb tags such as RFIDs. Bluetooth chips are relatively less constrained, not having to meet the demanding (low) price and (small) form-factor requirements of RFID.

Stateful pairings. In Bluetooth, a 'pairing' is well-accepted to refer to the security association between two devices. Privacy solutions for Bluetooth ought to be stateful, since session keys have been established. Identifiers are required for this and cannot be eliminated. This is true for the piconet configuration and also for the scatternet configuration.

Temporary pseudonyms for unlinkability. Temporary throwaway pseudonyms [41; 71; 84] can be of help. However, these must not be completely stateless, otherwise prior pair-wise relationships and piconet configurations would be quickly lost, and require frequent re-initialization. From the point of view of privacy, the need for a permanent identifier is debatable. Apart from helping manufacturers tell their product lines apart, having hierarchically arranged BD_ADDRs does not appear to do privacy much good.

Disentangle identifiers from other layers. Spectrum allocation and collision avoidance at the physical layer have been mentioned to have privacy implications. A good solution must resolve these.

Out-of-scope: De-correlating between different layers. While we have discussed exclusively about Bluetooth, in practice some other protocol is sometimes tunneled over Bluetooth. One important issue for anonymity is that the different protocols must carry out proper de-identification between them and be stateless. For example, if TCP/IP is tunneled over Bluetooth, the BD_ADDR should be de-linked from the IP address. However, we will consider this as outside the scope of this dissertation.

Unobservability: policy. The discoverable mode in Bluetooth should be turned off where possible. This is probably the best user policy, and should be enforced.

Thus we require a privacy framework which provides sender and destination anonymity in a mostly peer-to-peer ad-hoc wireless environment. Pseudonyms may be used, and unlinkability between pseudonyms should be provided. The solution should account for cases in which the wireless personal area network stays in a static configuration, and for cases where state needs to be kept between two paired devices over different sessions, due to the inconvenience of establishing a new session key (via a fresh run of a PIN-authenticated key establishment procedure). *Unobservability* should also be provided. If the premises underlying the usage scenario evolve, the privacy framework needs to change too. The means to establish strong pair-wise keys is assumed to exist [71; 205]: this is a non-goal.

6.7.3 An Existing Proposal with Stateful Pairing

As the identifier BD_ADDR is tightly integrated in the protocol and is used in many computations, it cannot be easily discarded. Gehrmann and Nyberg [71] proposed throwaway pseudonymous 'BD_addr_actives', to be used within an anonymity mode in Bluetooth, as a response to the location privacy flaws pointed out by Jakobsson and Wetzel [100]. It is a good first proposal, because using frequently changing pseudonyms would improve the unlinkability between actions by the same actual principal, and also protect the permanent BD_ADDR, which the device still retains, from disclosure to a casual observer. Using pseudonymous BD_addr_actives this way also allow the original design of the access code and frequency hop to be essentially retained.

We will briefly discuss their proposal here, but the details are in their paper [71]. In their proposal, a Bluetooth device can operate in anonymity or non-anonymity mode. A anonymity-enabled device uses two addresses: its real identity BD_ADDR (as in the current specification), and an 'active' address BD_addr_active. (The rest of the discussion covers only the anonymity mode.) The BD_addr_active acts as a randomly generated pseudonym, and is updated at regular intervals. A device in inquiry scan mode, replying to an inquiry, will send its current BD_addr_active, and not its BD_ADDR. There are two page scan modes; in the first mode, a device will listen to DACs based on its BD_addr_active as well as its BD_ADDR; in the second mode, a device will listen only to DACs based on its BD_addr_active. For the page mode, there are two situations: in the first situation, a master device will page for a previously paired slave device via the slave's BD_ADDR; in the second situation, the master device will page for a device via the latter's BD_addr_active. Further to the first situation, in fact the two paired devices had previously agreed on a parameter called BD_addr_alias; on connection, the master will send the slave a packet containing BD_addr_alias, which the slave will now use to look up the master's BD_ADDR in its (the slave's) database.

However, that proposal has two privacy weakness. The first is that the real identity, the BD_ADDR, is being used and may be disclosed to any device with which one has paired previously, though the identity is protected against other casual observers. Thus, adversaries can link different actions to the same actual principal if they can pair with this device, no matter what its particular BD_addr_active is at the instant. This is not an ideal privacy quality to possess, as policy-wise it should not automatically be assumed that all devices which have paired with one's own are not adversarial with respect to one's privacy.

A second weakness is with regards to the usage of BD_addr_alias, which is another 'BD_ADDR-like' identifier, established by two devices after they have paired, to signify the pairing in their respective database. For example, a BD_addr_alias would in Alice's database serve as an alias signifying Bob to Alice, and in Bob's database as an alias signifying Alice to Bob. In Alice's database there would be a tuple containing this BD_addr_alias and Bob's real BD_ADDR. In one mode, after Alice pages Bob, and before authentication takes place, Alice would send a packet containing this BD_addr_alias to Bob in an attempt to find out if they have paired before. Bob will now look up this alias in his database to find Alice's BD_addr, and respond accordingly. The problem with this usage is as follows: if Alice pages for Bob, but this is intercepted by an adversary Eve, and Eve receives the BD_addr_alias sent by Alice, while Eve will fail the test, Eve would be able to page Bob later using the alias, and thence be able to probe whether Bob has previously paired with Alice. The observability of transactions between Alice and Bob could thus be compromised offline.

As a new BD_addr_alias is randomly generated and securely distributed whenever a new connection is successfully established between the two previously paired devices, hence, the same BD_addr_alias would not be broadcasted each time that would allow an observing adversary to deduce that the same two devices are communicating again [1].

There are other caveats concerned with the use of temporary pseudonyms, which we would discuss. One of the most germane ones is that if a device could continually be tracked, even as it changes its pseudonym, that could still be linked to the previous one.

We propose an enhanced anonymity mode, also using pseudonyms, which would attempt to address the said problems, while recognizing that pairings may be stateful. We emphasize that this mode by itself will not resolve all privacy risks; a policy which requires discoverability and connectability to be turned off most of the time must be applied.

[1] At the time of writing of our paper [201], we thought the Gehrmann and Nyberg solution had this third weakness, but afterwards we noticed that this potential weakness was already addressed in the last line of page 14 of their paper [71] though overlooked by us previously. Nevertheless, this does not invalidate our protocol proposal.

#	Alice		Bob
1	Chooses random R_1		
2	$H_1 = H(I_B \mid R_1 \mid K_{AB})$		
3		$-ID1 : (R_1 \mid H_1) \rightarrow$	
4			Verifies H_1
5			Chooses random R_2
6			$H_2 = H(I_A \mid R_1 \mid R_2 \mid K_{AB})$
7		$\leftarrow ID2 : (R_2 \mid H_2) -$	
8	Verifies H_2		
9	Chooses random R_3		
10	$H_3 = H(I_B \mid I_A \mid R_1 \mid R_2 \mid R_3 \mid K_{AB})$		
11		$-ID3 : (R_3 \mid H_3) \rightarrow$	
12			Verifies H_3

Figure 6.2: Protected Stateful Pseudonyms

6.7.4 Our Enhanced Protocol Proposal

This protocol (Fig. 6.2), designed for our second situation described above, attempts to protect past pseudonyms from all third parties.

Modified ID packets. We modify the ID packet from the original Bluetooth specification. We now use three ID packets, denoted by ID1, ID2 and ID3. The contents of these packets are indicated in the Fig. 6.2.

Lightweight crypto: retain hash function. The only cryptographic primitive used is the hash function. Hashings are inexpensive operations, thus the parties can do these easily.

Temporary pseudonyms. The relevant past pseudonyms of Alice and Bob are denoted by I_A and I_B. H is a hash function, R_1, R_2 and R_3 are random nonces, and K_{AB} is the shared link key formed by Alice and Bob previously.

Three-way handshake. The three-way handshake is essential. Say, Alice intends to page for Bob. On verifying correctly the ID2 packet, Alice will have the assurance that Bob knows his previous pseudonym, her previous pseudonym, and their shared key. On verifying correctly the ID3 packet, Bob has the assurance that Alice knows these same three things.

Database containing unexpired paired individual pseudonyms. Alice keeps a database of tuples each containing her temporary pseudonym, the pseudonym of the other party, and the shared link key. Bob keeps a similar database.

Unlinkability: past pseudonym of responder protected. Alice wants to page for Bob. She selects a random nonce R_1, computes the hash H_1, and sends an ID1 packet. The hash in the ID1 packet hides the past pseudonym of Bob. Bob would compute and verify the expected hash in the ID1 packet using his list of the paired devices' pseudonyms and their associated link keys with the nonce.

Unlinkability: past pseudonym of initiator protected. When Bob successfully finds a match, he chooses a random nonce R_2, computes H_2, and responds with the ID2 packet. As Bob generates nonce R_2 randomly, he can be sure that his challenge to Alice is fresh. Alice, on receiving the ID2 packet, will verify the hash. If there is a match, Alice will generate a nonce R_3, compute the hash H_3, and reply with the ID3 packet. Bob will verify the hash on receipt of the $ID3$ packet. After the protocol runs successfully, both parties can proceed to carry out mutual authentication as usual. The security of the protocol depends on the randomness of the nonces, the irreversibility of the hash function, and the secrecy of the shared link key.

Replay resistance. A naive replay attack — the second weakness mentioned in Section 6.7.3 — incarnated here as an adversary capturing an ID1 packet previously sent by Alice and received by Bob, and replaying it, would be defeated, because Bob checks for freshness of R_1 and R_3, and Alice checks for freshness of R_2.

Resistance to delayed relay attack against responder. In another conceivable and more sophisticated attack, an adversary Eve intercepts an ID1 packet and prevents it from reaching Bob, but replays it later to Bob. Such an ID1 packet will pass Bob's R_1 freshness test. However, Bob now sends an ID2 packet with a fresh R_2. We can set a policy whereby uncompleted handshakes would raise an alarm at Bob's end, to alert Bob of the possibility of an intruder, so unless Eve next responds with a correctly formed ID3 packet, Bob would receive an alert. The 3-way handshake is essential for mitigating such an attack. Over at Eve's end, on her receipt of Bob's ID2 packet, Eve may suspect that Alice and Bob had paired previously, but she retains some doubt, because of possible collisions among the hashes.

But no resistance to online relay 'attack' - no distance-bounding. The protocol is not resistant to an online relay attack — in which Eve would position herself between two widely geographically separated victims — because the protocol does not incorporate any distance-bounding algorithm. While we raise this, this property was not intended to be one of the goals.

We leave it open whether the length of the ID1, ID2 and ID3 packets need to be equivalent to the DAC length of 68 bits. If they are also 68-bit, especially the ID1 packet, then it helps to obscure the fact from simplistic traffic analysis that Alice is paging for a old pseudonym of Bob, in which case the random nonce would take up, say, 34 bits, and the hash the other 34 bits. Or else these packets can extend up to the length of 160 bits plus a suitable length of a nonce, to better resist brute-forcing.

Re-generate past pseudonyms on databases after successful connection. This is especially useful for privacy if two previously paired devices need to re-connect often. After successful re-connection between the two devices, Alice and Bob will each generate randomly a new pseudonym (I'_A and I'_B respectively), and then distribute the pseudonym to the other party securely (i.e. encrypted). The tuple on their respective databases will be updated accordingly. This re-generation of pseudonyms after successful re-connection will further strengthen unlinkability. (This step is a further enhancement over the earlier proposals in our paper [201].) Precisely, how this step strengthens privacy can be considered as follows: let us assume that the length of the random nonce is set to be only 34-bit in length, and the ID1, ID2 and ID3 packets set to 68-bit in length (as suggested above), then if the random nonce R_1 happens to be the same in two different transactions (i.e. a birthday collision) and is observed by an adversary while the two paired past pseudonyms remain constant, then the adversary may be able to deduce that it might be the same device communicating to a paired device since the ID1 packet would be the same both times. The suggested step of updating past pseudonyms on the database after each successful re-connection removes this (slight) vulnerability. But more importantly, this step would make the checking for freshness of the nonces unnecessary.

Thus, overall, the proposed protocol provides good scalability in remembering and responding to past pairings, while not leaking the permanent BD_ADDR nor previous pseudonyms unnecessarily. Bob keeps changing pseudonyms, yet remains able to respond to his previous pairing arrangements, as he has kept a history of these pairings, whose ages are set by policy.

6.7.5 Additional MAC Layer Recommendations

6.7.5.1 Inquiry and Inquiry Scan

For device discovery, we keep to the Gehrmann and Nyberg proposal [71], where the inquiry and inquiry scan states are left as according to the original specification, with the change that the identifier returned at inquiry scan is the slave's BD_addr_active instead of its BD_ADDR.

As a matter of strong privacy policy to counter tracking, we recommend that a device's discoverability should be turned off whenever it is not required.

6.7.5.2 Page and Page Scan

As mentioned, the Gehrmann and Nyberg proposal featured two paging situations. One situation is where a master pages a slave based on the latter's current BD_ADDR_active. The second situation is where a master pages a slave based on the latter's long-term BD_ADDR, which is useful for previously paired units. The second situation allows pairings to be remembered, but has the unfortunate weakness of leaking the BD_ADDR of the slave being paged, hence compromising linkability as well.

We prefer that the long-term BD_ADDR never be leaked. We hence propose a somewhat different second situation, in which a master would attempt to page a slave using modified ID packets derived from the previous BD_ADDR_actives which the master and the slave had used to pair. These packets cryptographically protect the addresses from casual sniffing. The formats of the packets and the required protocol are as described in the following section. We believe that it is more private to have done pairing with the pseudonyms than with the long-term identifiers. This is not too difficult to support, as Bluetooth pairing is already based on a shared password rather than on permanent identifiers. It can be decided by policy settings how soon to expire pairings, as well as how soon a device expects a paired device to have changed pseudonyms.

As a policy setting, we recommend that a device's connectability be turned off whenever the owner does not expect connection requests to be received.

6.7.5.3 Physical Layer

We have described in Section 6.6 that parts of the device address can be recovered from the access code and the frequency hop pattern. This privacy risk can be resolved by using changing pseudonyms. A more involved solution would be to totally de-link the resource allocation at the physical layer from the device identifiers (both permanent and pseudonymous ones). So, the frequency hop pattern could be initialized from other parameters instead of a device's identifier and its clock. The choice of the physical layer is quite orthogonal to privacy; we may equally choose to use direct-sequence (DS) spread spectrum (though DS is generally more costly) instead of frequency hopping, such that different DS sequences either use different pseudonyms or are even de-linked from these.

6.7.5.4 Triggers for Pseudonym Change

We propose several triggering mechanisms to change pseudonyms. It is well-known that if a device can be continuously tracked, such as when it is discoverable and is the only device in a locality, then even a change of identifier would not prevent linkability. Discoverability ought to be turned off during pseudonym change. We suggest a sub-state in the anonymity mode in which the device is ready to change pseudonyms. A change may be triggered by any of several events. Firstly, it may be brought about by the owner's manual action. Secondly, it can be automatically changed at random time intervals. Thirdly, the pseudonym should be changed when a certain threshold large number of discoverable devices are detected in an inquiry sweep. The rationale is that it would be easy to 'blend in with the crowd' and anonymise oneself. This method should be carefully applied because an attacker can spoof the presence of a large number of devices. [1] It uses the concept of a mix zone [27], the difference being that here, pseudonym change is handled by the devices themselves instead of a network infrastructure.

[1] However, the attacker would not reduce the anonymity of the victim by forcing a pseudonym change — the only effect would be to make the victim *believe* he is more anonymous that he actually is, which might perhaps lead him to lower his guard.

6.7.5.5 Further Issues

New pseudonyms must be randomly generated, and one solution is by hashing some counter. Also, the 28-bit 3.2 kHz Bluetooth device clock, which has a cycle of 23.3 hours, of which the highest 26 bits are disclosed (or a 1.25 ms resolution), must be randomly re-adjusted on a pseudonym change, to prevent an adversary from linking pseudonyms to a clock, even accounting for the clock drift. Bluetooth uses a ‘friendly name’, which is a human-readable name to tag devices during device discovery and to help manage the list of paired devices locally. To reconcile privacy with usability, we propose the following: the field should be left empty or not transmitted during device discovery, but the user could be allowed to locally tag his list of paired devices with ‘friendly names’ of his choice to help him better distinguish the devices than through hexadecimals.

Certain RF attacks attempt to pinpoint the location of devices by measurements of irradiated power, and more sophisticated attacks distinguish RF signatures of individual devices, but these are outside our scope. In our privacy framework, we have not made use of digital certificates, because though these allow strong authentication, they seem to be inimical to anonymity.

6.8 Future Directions

It is clear that location privacy considerations are becoming important in emerging pervasive wireless technologies. While the RFID space is being well-covered by researchers, the niche occupied by ‘less dumb’ technologies such as Bluetooth (and similarly, Zigbee) is less covered. Besides ourselves, other researchers have found and demonstrated less-than-desirable privacy characteristics of new ubiquitous devices [165].

We foresee several threads of work leading off from this area. Firstly, there needs to be a better appraisal of the exact privacy versus performance trade-off required. If the intended devices are less constrained by hardware (i.e. processor capability, form-factor, battery power, latency, etc), then stronger privacy can be considered, such as using ‘secret handshakes’ [14] (which make use of computationally intensive identity-based pairings), or else public-key operations. Secondly, we also envisage that the development of privacy-enhancing solutions for Bluetooth-type of technologies would benefit from a more rigorous type of security analysis and adversary modelling, for example, like the kind presented for RFIDs by Avoine [10]; even if it is allowed that these in itself are not an ultimate proof of security, the analyses thence obtained would be invaluable for comparing different proposals side-by-side against a common benchmark set of well-modelled privacy attacks, and for giving indications of the exact security level offered.

6.9 Summary

Our contributions:

- We identified and quantified the various location privacy problems of Bluetooth, a well-received pervasive computing wireless technology which has been widely deployed.

- We have presented our proposal for a MAC layer location-privacy-enhancing solution for Bluetooth. We use simple hash functions as the basic primitives. Our proposal has improved unlinkability over an earlier solution.

- We have also provided some policy recommendations governing the use of Bluetooth devices (and other future wireless ad-hoc devices in the same niche) so as to improve unobservability and strengthen the overall location privacy.

- At the broader level, we articulated the idea that location privacy is a multi-layer problem, even while we ourselves attempt to solve the problem in terms of unobservability and unlinkability at Bluetooth's MAC layer.

Location privacy as a security problem is becoming increasingly recognized, with legal, personal, social, economic, and of course technical dimensions. Technically, it is not confined to merely one layer of the protocol stack; it is a multi-layer problem. We solidified the idea that location privacy is a multi-layer problem: the problem must be addressed at every layer that is exposed, identifiers must be de-linked and de-correlated across the layers, on the other hand, these do not absolve the designer from the solving the location privacy problem *very well* at the particular layer that he wishes to take responsibility for. In this chapter, we have described our proposal to solve the location privacy problems at the MAC layer for Bluetooth.

Chapter 7

Location Privacy at the Physical Layer

7.1 Outline

We have referred to location privacy as a multi-layer problem in the previous chapter. It is shown that for a technology such as Bluetooth, securing its location privacy MAC layer is an important step towards addressing the problem of location privacy, because Bluetooth devices are primarily identifiable by their permanent factory-set Bluetooth MAC addresses. Thus, we had proposed unlinkable pseudonyms, which are flexible enough to be memorable by previously paired devices.

However, the step, while important, is not sufficient to assure overall location privacy. In this chapter, we further elaborate on the theme of multi-layer privacy. We will show how a low-level passive adversary monitoring physical layer signals may pinpoint and track the positions of a mobile node, hence compromising its location privacy. Using pseudonymous higher-layer (such as MAC layer) identifiers are insufficient to protect against such an adversary. We describe our proposal to defend against this, and perform some simulations to show that our proposal performs better than current systems.

The essence of our contribution is to consider covert beampattern, in particular using multiple antennas to do adaptive beamforming, for reducing observability. We outline the components required to engineer such a privacy-enhancing system, and develop an evaluation framework to analyze and quantify the location privacy offered [200]. The parameters we used are wireless LAN operating parameters. We find that signal-to-noise ratios to carry out successful direction-finding is more stringent than that for mere successful communications, which confers an additional security margin. We have made extensions to performance metrics for measuring location privacy from the static into the mobile situation, and using an end-to-end integrated radio and mobility simulation, we compare location privacy performance of omnidirectional versus adaptive beamforming beams. Under this evaluation, our proposed privacy-enhancing system is shown to perform better. The framework we develop is flexible and extensible to other wireless technologies and beam patterns as well.

We put forward that *unobservability* (vis-a-vis unlinkability) at this lowest, physical layer, is a particularly important property to assure overall location privacy.

7.2 Related Work

Many researchers have discussed location privacy for upper layers and have proposed countermeasures such as degrading the location information or changing pseudonyms at appropriate intervals, by Beresford and Stajano [27; 28] Grutuser and Grunwald [83; 84], Gruteser and Hoh [85; 93], Wong and Stajano [201], and so on. These are, however, insufficient in the presence of a passive observer at the lowest (physical) layer. Such an adversary could detect the location of the victim by bypassing the degradation and could observe the changeover from one pseudonym to the other, thereby re-linking the two pseudonyms of the victim.

We distinguish our problem with the related, but subtly different *distance-bounding* problem, which often uses physical layer characteristics, and the problem is addressed, for example, by Brands and Chaum [34] through a protocol, and by Fishkin and Roy [67] through measuring Signal-to-Noise Ratios in the antenna. These solutions are meant to help a verifier calculate the upper bound of the claimant's range.

Before we discuss how existing solutions would break down against physical location privacy attacks, we will outline the application scenario and the adversary model, so as to make the discussion more concrete.

7.2.1 Application Scenario

The expected application scenario is as follows. Suppose you as a user, are carrying a wireless-enabled personal laptop. Your laptop maintains connectivity to the Internet through base stations connected to a network backbone, and they may operate, by way of example, the 802.11 wireless LAN type of technology. In an attempt to maintain your location privacy, the wireless interface of your laptop periodically changes disposable pseudonyms [84]. Through the course of your movement through an urban centre, you may cross over between footprints of different base stations.

7.2.2 Adversary

We model the adversary as the following. He does not control the public network of base stations, but he controls a network of receivers whose function is to carry out position-location of mobile nodes. This is shown pictorially in Fig. 7.1. His objective is to track your movement over pseudonym changes. As long as he can maintain a high enough resolution location fix on your emitting mobile node, even if the anonymity set of the perceived mix zone [27; 28] is large, he can compromise your linkability significantly.

Specific algorithms this attacker may use to locate your position encompass those performing Time-Difference-Of-Arrival (TDOA), Time-Of-Arrival (TOA), Angle-Of-Arrival (AOA), and Receive-Signal-Strength-Indicator (RSSI), which will be sketched in Section 7.3. The parameters of the link between the mobile node (MN) and the base station (BS) will be described in Section 7.7.

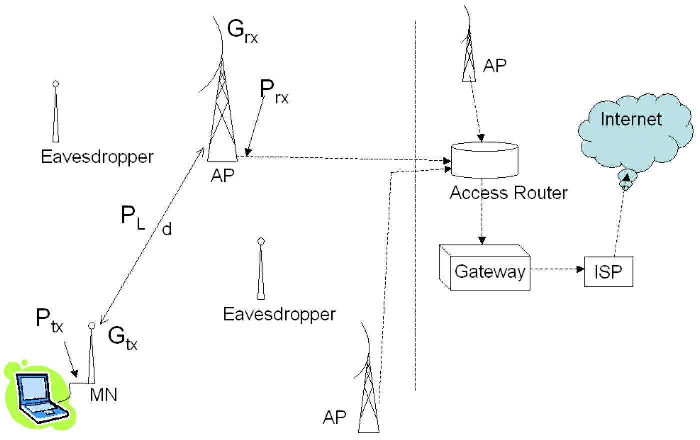


Figure 7.1: Adversary Model

7.2.3 Why High-Level Countermeasures Fail

High-level anonymizing mechanisms, such as pseudonyms, would fail under the above threat model. The effect of using changing pseudonyms is: firstly, to obscure the real identity of the user, and secondly, to confer unlinkability between different pseudonymous identities of the same user. Pseudonyms fail if the attacker can continue to pinpoint the nodes even as these change pseudonyms within mix zones. A mix zone, shown in Fig. 7.2, is a spatial construct within which pseudonyms are 'mixed' by the middleware or network provider, and hence protected against adversarial application providers, in the original model proposed by Beresford and Stajano [27; 28]. The middleware or the network provider are assumed to be honest. Conversely, pseudonym changes which violate the model by occurring outside mix zones would open the pseudonyms to be linkable by application providers and allow compromise.

In fact, the mix zone result of Beresford and Stajano is more generalized than what was explicitly stated in their papers [27; 28]. A signal-level mix zone can be conceived. This signal-level mix zone can be defined as the area (or more accurately the volume) of space within which the attackers cannot distinguish between the positions of two similar signal sources with high probability. We may assume that the signal-level mix zone is usually smaller in size than the application-provider-level mix zone.

Introducing radio silence periods [94] helps increase the uncertainty experienced by the low-level attacker, and using the mix zone analogy, is equivalent to expanding the mix zone radius. But this requires close coordination of the radio silence overlaps of mobile different nodes as they change pseudonyms, so as to guarantee anonymity. These countermeasures are clearly inadequate under our attacker model.

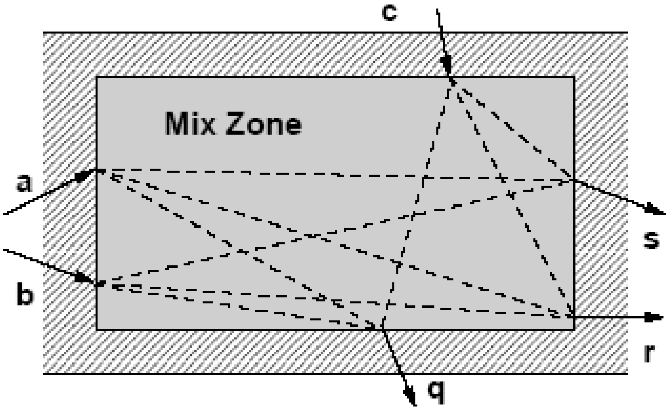


Figure 7.2: Plan View of Mix Zone

7.3 Localization Schemes

A passive eavesdropper who attempts to silently detect the location and movement of a mobile node would need to use some localization scheme. Several different generic localization schemes would be covered in this section. Different localization schemes produce error geometries of different shapes.

Received Signal Strength Indicator (RSSI) technology uses measurements of signal strengths to calculate distances from the transmitters. It depends on knowledge of the transmitter power and the propagation loss.

Time Difference of Arrival (TDOA) technology measures the arrival times of a signal at different receiving stations, and using the difference in time it arrives at different stations, it is able to trilaterate the signal source and locate it if at least three receiving stations are used. The stations must be well-synchronized.

Time Of Arrival (TOA) technology also measures the arrival time of a signal, and is similar to TDOA, except it uses the absolute time, to carry out ranging and triangulation of the transmitter. (The Global Positioning System (GPS) [72] utilizes such a concept. The satellites have very accurate atomic clocks; the GPS receiver uses an internal crystal oscillator-based clock that is continually updated using the signals from the satellites. The receiver identifies each satellite by its distinct Code/Acquisition (C/A) code pattern and generates an identical C/A sequence itself internally. By lining up the two sequences, the receiver can measure the time delay for each satellite, and calculate the distance (known as pseudorange). By calculating its distances from several hovering satellites, whose positions are known, a GPS receiver is able to calculate its own coordinate position on Earth.)

Angle Of Arrival (AOA) technology measures the incident angle of the signal at each receiving station. With data from two or more stations, the location of a transmitter may be found. For this technology, the receiving stations have to deal with multi-path signals well.

In our proposal, our primary interest is in the AOA scheme.

7.4 Physical Layer Solution: Adaptive Beamforming

To address such a threat, we analyzed the requirements and proposed an architecture. Advances in RF frontends and digital electronics have made it possible to produce highly directional antenna patterns using adaptive beamforming. Smart antennas are increasingly being used in wireless LANs and cellular communications in their base station antennas. Smart antennas direct radio energy towards the intended users by beamforming, and reduce energy away from unintended users. The use of multiple antennas at both ends capitalizes on spatial multiplexing and increases data throughput, a concept epitomized by the MIMO (Multiple Input Multiple Output) model. Smart antennas also mitigate any negative effects of multipath fading. For location privacy, we are more concerned with multiple antennas' beamforming property – we propose for mobile nodes to use smart antenna to produce beam patterns with narrow main lobes and low sidelobes. It is worth noting that what we are proposing is actually a different use of multiple antenna elements, compared to MIMO. In MIMO, one or multiple beams are created by an antenna array, and can be used to take advantage of transmit diversity, whereas in our proposal, we advocate a single beam, to achieve a *covert* mode.

The majority of the building blocks necessary to engineer a system hardened against low-level wireless location privacy attacks exist. We outline the overall system strategy.

7.4.1 Assumptions

We make the following assumptions:

(1) Mobile nodes have a robust method of estimating their own location. Increasingly, satellite radionavigation receivers (namely, GPS, and in the near-future, Galileo) are being integrated into mobile handsets. New generation GPS receivers themselves are able to use technologies such as adaptive beamforming [36] or analog electronics, to suppress the effect of ground-based jammers and spoofers, and thus could derive own geo-position with good reliability against even a moderately strong active attacker. Mobile nodes also assumed to have a means of estimating their azimuthal bearing, such as by means of a compass.

(2) The intended physical environment ranges from (a) flat unobstructed terrain to (b) moderately urban environment where the base station antennas are sited at high locations such that there is typically good line-of-sight to mobile nodes. If the environment is instead composed of many RF scatterers and reflectors, then the attacker's job of finding the direct and dominant signal path is already made very difficult.

7.4.2 System

The main features of the system are the following:

(1) Mobile nodes have a secure way of learning the true coordinates of the base stations. These coordinates could be pre-distributed by a trusted authority, appropriately digitally signed and attested.

(2) A mobile node will shape a radio beam in the direction of the base station it is associated and communicating with, based on knowledge of both nodes' positions.

(3) Mobile nodes will carry out mutual authentication with base stations, as a further protection against an attacker which attempts to trivially spoof a base station.

(4) The mobile node will not emit much power outside its main lobe. The main lobe is required to be narrow, nulls are required to be deep and the sidelobes need to be low; these are, however, in turn dependent on the antenna array geometry.

7.5 Security Considerations

We analyze security issues raised by our proposal:

(1) Base Station Spoofing: In current technologies, a mobile node associate and connect with a base station which emits the beacon signal with the highest power. In our proposal, the base stations' coordinates are pre-distributed via a trusted mechanism. The mobile node would select a base station from the list and shape its antenna pattern to point towards it and thereby associate with this station. Our mobile nodes will ignore beacon frames as an indicator of position. This design choice would limit the effectiveness of an attacker who tries to carry out signal synthesis attacks or signal relay attacks [1] to spoof base station positions in an attempt to steer the mobile node's beam towards itself.

(2) Self-Positioning Inaccuracies: Our proposal is dependent on the mobile node having a reasonably accurate knowledge of its own position and bearing. We can make a reasonable assumption that the node's position estimate can be accurately derived using radio-navigation signals such as from GPS, and in the future, Galileo. Radio-navigation receivers with shaped antenna having deep nulls [36], resistant to spoofing as well as jamming attacks, are even now becoming available. Further, we may build tolerances into the mobile node's beam-pointing, to take into account slight self-positioning imprecision in the geo-positioning which may propagate into azimuth calculations. (When the node moves indoors, an inertial navigation system can be used, and it is in turn cued from radio-navigation signals.)

(3) Inter-Layer Linkability: De-linking between different layers of the protocol stack is important to guarantee overall anonymity. Fig. 7.3 shows the network stack, and the identifiers at each layer. As has been mentioned earlier in the chapter and in the sections on the Bluetooth MAC layer pseudonym management, the anonymizing mechanisms at each layer need to de-link the identifiers across different layers. These mechanisms ought to be well-coordinated to counter attacks which occur across layers.

[1] In a *signal relay attack*, the adversary receives the signal from a spoofed base station, relays it, and broadcasts it from a different location, pretending to be the base station. In a *signal synthesis attack,* the adversary himself synthesizes the signal that is expected were the spoofed base station positioned where the adversary is and is broadcasting from there.

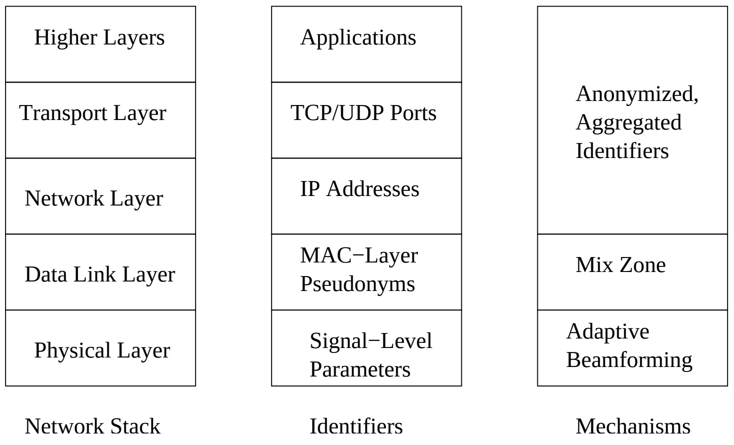


Figure 7.3: Layers

The problem would be worsened by feature interaction and optimizations within and across layers. In this section, we will not cover layers other than the physical layer; we will just make the assumption that the layers have been properly de-linked (i.e. isolated).

(4) Samples for Tracking Missing: We can relate our work to previous work in multi-target tracking (MTT) [85], which had used tracking and target labelling algorithms on discretized samples of sightings of the mobile nodes' pseudonyms. The effectiveness of our covert adaptive beamforming proposal can be equated to *missing samples* available to a distributed passive adversary for carrying out MTT. With samples gone missing, the adversary would be harder pressed to locate and de-anonymise nodes precisely. This is elaborated upon in Section 7.9.

7.6 Radio Environment

We consider a communication system similar to the wireless LAN IEEE 802.11b system.

7.6.1 Link Budget

The radio link budget equation is given as: $P_{tx} + G_{tx} - P_L + G_{rx} = P_{rx}$, where P_{tx} is the transmitter output power (of the mobile node), G_{tx} is the transmitter antenna gain, P_L is the path loss, P_{rx} is the power received at the receiver of the base station, and G_{rx} is the antenna gain of the receiver. It refers to the uplink. The values are given in dB, dBi or dBm as appropriate.

7.6.2 Loss Propagation Model

Selecting a suitable propagation model so as to compute a most realistic path loss P_L is one of the key aspects of this work. For a link without obstruction, the standard free space loss model is:

$$Loss = 32.45 + 20\log(D/\mathrm{km}) + 20\log(f/\mathrm{MHz})$$

But this oversimplified model is inadequate. Models that represent real scenarios more adequately exist, such as the COST-Hata Model [48]. It is not site-specific to the extent of requiring every physical structure to be modelled. It had been fitted to observed power degradation curves through exhaustive field studies of mobile environments similar to what we are considering.

$$\begin{aligned} Loss = 46.3 + 33.9\log(f/\mathrm{MHz}) - 13.82\log(h_{base}/\mathrm{m}) \\ -a(h_{mobile}/\mathrm{m}) + (44.9 - 6.55\log(h_{base}/\mathrm{m}))\log(D/\mathrm{km}) \\ +C_m \end{aligned}$$

The parameters are f for frequency, h_{base} for the base station height, h_{mobile} for the mobile node height, D for the distance, a is a function [48] and C_m is a constant.

7.6.3 Regulatory Requirements

The Federal Communications Commission (FCC) in the USA has limited the transmitter power to a maximum of 30 dBm or 1 watt, for 2.4 GHz [66]. Using the maximum power, the maximum transmitting antenna gain is 6 dBi. Point-to-multipoint systems are given a lower limit than point-to-point systems – we will use the more conservative limit. For this, the maximum EIRP is 36 dBm, and for every dB by which the transmitter power is reduced, the antenna gain may be increased by 1 dBi. We use Federal Communications Commission (FCC) limits [66] for our analysis. European Telecommunications Standards Institute (ETSI) limits may likewise be investigated. ETSI limits on maximum output power and EIRP, adopted by many European countries, are lower than those mandated by FCC.

7.6.4 Antenna Gain Pattern

In adaptive beamforming, the operations of phase shifting and amplitude scaling for each antenna element are carried out adaptively. A beampattern is basically dependent on the array geometry (i.e. the number of elements, the spacing and the aperture size), the transmitted carrier frequency, the gain pattern of each element, and the antenna weights. The array factor (AF) representing the gain for the antenna is described analytically [13] as follows, substituting $N = 4$ for a 4-element antenna.

$$AF(\theta) = \sum_{n=1}^{N} e^{+j(n-1)(kd\cos\theta+\beta)}$$

where k is the wave number, d is the inter-element spacing and β is the excitation phase. The expression for the radiation pattern is:

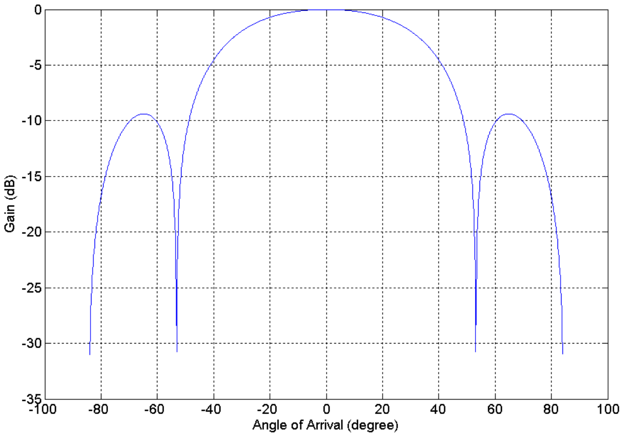


Figure 7.4: Array Factor of a 4-element Antenna Array

$$E(\theta, r) \quad = \quad a_{\phi} j\eta \frac{kI_o e^{-jkr}}{4\pi r} \times AF(\theta)$$

where the first term is the electric field description, which will be replaced with our normalized transmission power and an antenna gain of unity. The second term is the array factor describing of the radiation pattern in a vertical dipole. Choosing values so as to achieve appropriate trade-offs between minimizing sidelobe power and minimizing mainlobe beamwidth, setting the inter-element spacing to be half-wavelength (i.e. 6.25 cm for 2.4 GHz) [1] and assuming such antenna can be synthesized, we can derive the array factor in Fig. 7.4.

7.6.5 Attacker's Direction-Finding

Due to tolerances and known calibration issues in direction-finding systems, and environmental factors, localizing an emitter in the field is not as trivial as simulations suggest, but these set an upper-bound on an attacker's effectiveness.

Position localization via Received-Signal-Strength-Indicator is unreliable, because the transmitter power may be unknown and received power may fluctuate due to multipath propagation and shadowing. Time-Of-Arrival and Time-Difference-Of-Arrival methods require the signal to have a very short pulse-width to be effective, such as Ultra-WideBand signals. The reliable option left against WLAN-type systems is Angle-of-Arrival (AOA) estimation by two or more sensors, followed by triangulation, as indicated in Fig. 7.5. (MN represents Mobile Node.)

[1] An aperture size of 18.75cm for an 4-element linear array can conform comfortably onto the top rim of, for example, a 9-inch LCD screen of a laptop.

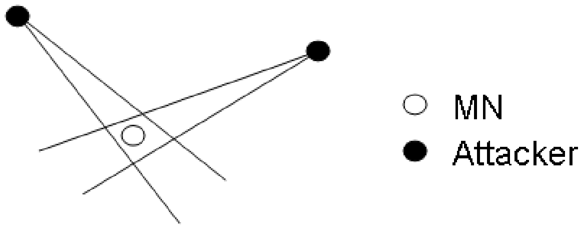


Figure 7.5: Geometrical error constraints using AOA

Signal Model

The signal model is as follows. There are $M \leq L$ narrowband sources (where L is the number of antenna elements) centred at frequency ω_0. We also assume the sources are deterministic. Additive noise is modelled as a stationary zero-mean random process. To mathematically describe the effect of the translational invariance of the antenna array, we apply the conventional practice of treating the array as being comprised of two sub-arrays, Q_x and Q_y, which are identical in every respect, except that they are physically separated by a known linear displacement, Δ. The received signal model at the ith antenna element of the two sub-arrays can be expressed as:

$$\begin{aligned} x_i(t) &= \sum_{k=1}^{M} s_k(t) a_i(\theta_k) + n_{xi}(t) \\ y_i(t) &= \sum_{k=1}^{M} s_k(t) e^{j\omega_0 \Delta \sin\theta_k / c} a_i(\theta_k) + n_{yi}(t) \end{aligned}$$

where θ_k is the direction-of-arrival of the kth source relative to the direction of the translational displacement vector Δ. Combining the outputs of elements in the two sub-arrays, the received data vector can be derived as:

$$\mathbf{x}_i(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}_x(t)$$
$$\mathbf{y}_i(t) = \mathbf{A}\Phi\mathbf{s}(t) + \mathbf{n}_y(t)$$

where $\mathbf{s}(t)$ is the $M \times 1$ vector of impinging signals (wavefronts) as observed at the reference sub-array (in our case, we choose this to be Q_x), and the matrix Φ is a diagonal $M \times M$ matrix of the phase delays, and is given by:

$$\Phi = \mathrm{diag}\{e^{j\beta_1}, \cdots, e^{j\beta_M}\}$$

where $\beta_k = \omega_0 \Delta \sin\theta_k / c$

Finding the Angle-Of-Arrival (AOA)

The MUSIC (MUltiple SIgnal Classification) algorithm [167] was one of the first significant algorithms for doing AOA estimation, then ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques) [150; 162; 163] emerged, which was found to be particularly cost-effective. The MUSIC algorithm is powerful, but rigorous analysis has found that MUSIC estimations are biased [208]. Its bias increases as the signal source moves away from boresight, and also increases as the number of elements increases while keeping the same aperture. Comparisons of MUSIC and ESPRIT have been made [73]. ESPRIT is compelling because it is much less computationally intensive, but requires a linear array - an undemanding requirement (otherwise, non-linear arrays are generally possible).

There are different versions of ESPRIT. Studies of these have been conducted and it had been shown that some versions produce the same mean-square-error [156]. In our studies, we used Total Least Squares-ESPRIT (TLS-ESPRIT) [163] version, which was introduced to make the estimation more practical when only a finite number of noisy measurements is available.

We apply the TLS-ESPRIT algorithm [185] as follows:

(1) We define L as the number of elements in the direction-finding array. Since we are assuming a 4-element array, so $L = 4$. We next define a L-dimensional random vector $\bar{x}$ corresponding to L consecutive data samples. We estimate the correlation matrix $\hat{R}_{\bar{x}}$ from the data.

(2) We compute the generalized eigenvectors and eigenvalues of $\hat{R}_{\bar{x}}$, where k ranges from 1 to L :

$$\hat{R}_{\bar{x}}\bar{e_k} = \bar{\lambda}_k \Sigma_{\bar{\eta}} \bar{e_k}$$

(3) Usually there is a need to first estimate the number of sources, M. (Several algorithms exist to compute this.) The maximum number of signal sources whose direction can be estimated is limited to be 1 less than the number of antenna elements. Thus, a 4-element array is unable to estimate for more than 3 signal sources simultaneously.

(4) We next generate a basis spanning the signal subspace and partition it as $\bar{Q}$, where $\bar{Q} \in \mathbb{C}^{L\times M}$.

$$\bar{Q} = \Sigma_{\bar{\eta}} \begin{bmatrix} | & & | \\ \bar{e}_1 & \cdots & \bar{e}_M \\ | & & | \end{bmatrix} = \begin{bmatrix} Q \\ \times\cdots\times \end{bmatrix} = \begin{bmatrix} \times\cdots\times \\ Q' \end{bmatrix}$$

(5) We perform singular value decomposition on: $\begin{bmatrix} Q & Q' \end{bmatrix}$ We then extract V as the right singular vector, and partition V into four $M \times M$ submatrices:

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$$

(6) Next we compute the eigenvalues λ_1, λ_2,... λ_M of the matrix Ψ_{TLS} which is equal to $-V_{12}V_{22}^{-1}$.

(7) We come to our last step, to compute the angles of arrival, where δ is the separation between adjacent antenna elements:

$$\theta_k = \sin^{-1}\frac{Arg(\lambda_k)}{2\pi\delta}$$

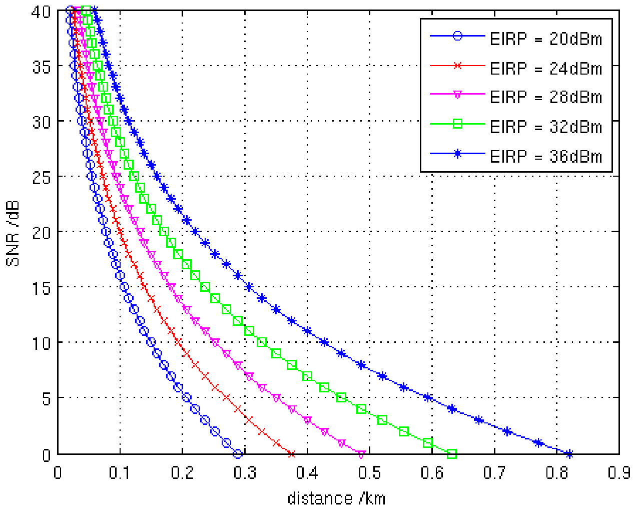


Figure 7.6: SNR versus distance for different EIRP

We are not considering an adversary who might use physically steered directional antennas (such as Yagi antennas), because these devices would firstly, be cumbersome, and secondly, allow only one signal source to be tracked at a time per receiver.

7.7 Radio Simulation

7.7.1 Path Loss

Using procedures outlined earlier, we calculate the Signal-to-Noise Ratios (SNRs) for a range of Effective Isotropic Radiated Power (EIRP) values for the uplink. The upper end is the FCC limit. The lower end can be considered as the mobile device exercising power control to limit its output power to just 20 dBm, and using an omnidirectional antenna. Many PC card wireless adapters today have omnidirectional antennas with a gain of 0 dBi (or equivalently, the mobile node could also be transmitting at less than 20 dBm, and using a high gain antenna). We assume a reasonable receiver noise figure (NF) of 10 dB for the base station. The receiver antenna gain, G_{rx} is set at 6 dBi, again a typical figure, for many current BS antennas. Fig. 7.6 shows the degradation of SNR with increasing distance, for different uplink EIRP, using the previously stated assumptions.

Fig. 7.7 shows the change in SNR for different receiver noise figures at a fixed EIRP of 20 dBm. A better noise figure of 6 dB adds a 4 dB increase of SNR over the case where the receiver noise figure is 10 dB. Thus, an attacker who invests in a more expensive low-noise receiver will improve the SNR of the received signals for a given distance.

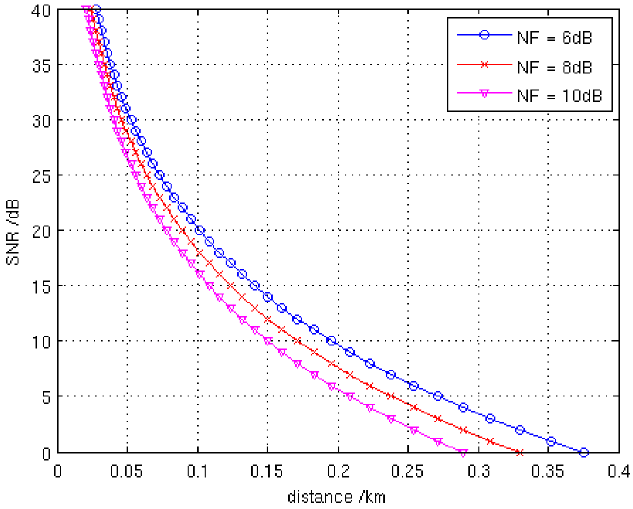


Figure 7.7: SNR vs distance for different Rx NF

7.7.2 Angle-Of-Arrival Estimation Errors

We investigate the performance attainable using the TLS-ESPRIT algorithm to intercept wireless LAN type of signals. We make the assumption that the attacker uses a 4-element linear array for his receiving station. (If the attacker invests in more antenna elements for his system and thence a correspondingly greater aperture size, he can obtain better performance and future simulations can elucidate this, provided the channel characteristics do not change while he takes a longer time to collect the signals.) Angle-of-Arrival estimation is carried out and the the mean-square-error (MSE) of the estimation is computed. The plot of the MSE of AOA estimates with different SNRs is shown in Fig. 7.8. The SNR refers to that experienced at the *attacker's receiver*. Again, we assume that the receiver NF is 10 dB.

Fig. 7.8 shows that the AOA estimates converge to an acceptable accuracy when the SNR is 10 dB or better. At lower SNR levels, the fluctuation is quite significant, often giving more than 1 deg MSE. At SNR levels of 15 dB or more, the MSE flattens out. Thus, the adversary who attempts to do Angle-of-Arrival estimation using TLS-ESPRIT needs to be placed in a position where he can receive the MN's uplink signal at a SNR value of 10 dB or better, otherwise his estimate will be very error-prone. To give an idea of the error, if the MSE of the AOA estimate is 1.4 deg^2 and the attacker is 150 metres away from the MN, then the error in distance is around ±5 metres.

The above simulations indicate the **upper bound** on the attacker's direction-finding ability with the said set-up. In real field conditions, there will be imperfections in equipment. For example, frequency stability of the clock used for demodulating and sampling the measured signals affects the AOA estima-

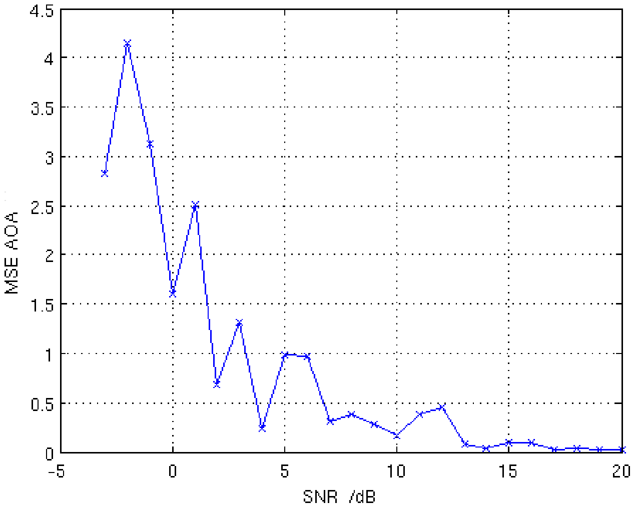


Figure 7.8: MSE of AOA estimation vs SNR

tion accuracy [124]. This is shown in Fig. 7.9.

Briefly, the frequency offset (i.e. phase rotation observed at the receiver) is caused by differences in the oscillator reference frequencies at the transmitter and observer. The frequency shift depends on the frequency stability of the reference clocks, specified in parts per million (ppm). If both the transmitter and receiver have different clock accuracies, then the maximum frequency accuracy in terms of ppm is

$$f_{ppm(max)} = f_{ppm(max)rx} + f_{ppm(max)tx}$$

The maximum phase rotation over a burst of N samples with a sample rate r_s is:

$$\theta_{max} = (2\pi \times f_c \times f_{ppm(max)rx}) \times \frac{N}{r_s}$$

where f_c is the carrier frequency. Consequently, the incremental phase rotation for each received sample is:

$$\theta_i = \frac{\theta_{max}}{N}$$

For example, if we have maximum frequency offset f_Δ of 300 Hz, the frequency stability will be:

$$f_{ppm} = \frac{f_\Delta}{f_c} = \frac{300\text{ Hz}}{2.4\text{ GHz}} = 0.125 \times 10^{-6}(\text{i.e. } 0.125\text{ ppm})$$

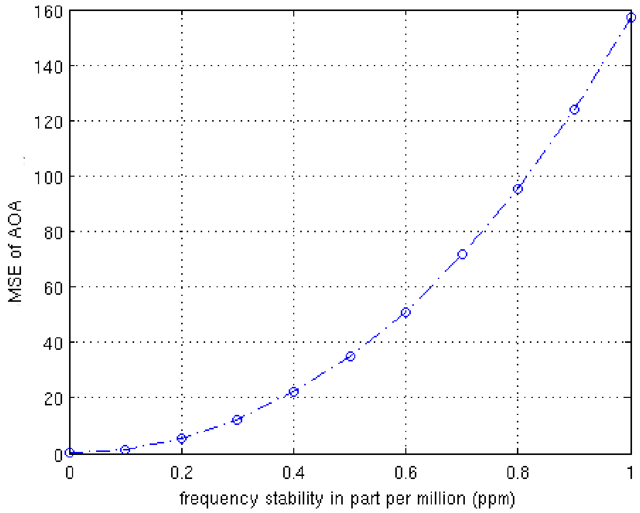


Figure 7.9: MSE of AOA estimation vs frequency stability

Fig. 7.9 shows the very significant deterioration of AOA estimation accuracy as the frequency stability worsens from 0.1 ppm to just 1 ppm. From 0.3 ppm upwards, errors make the AOA estimate rather unusable. Obtaining 0.1 ppm frequency stability is costly; the attacker is forced to make significant investment in equipment.

7.7.3 Security Margin

For a fixed uplink EIRP of 20 dBm, Fig. 7.10 shows the distance for successful communication between the MN and the BS, and the distance for successful AOA estimate of the MN by an attacker.

For transmission with a data rate of 1 Mbps with a bit-error-rate better than 10^{-5}, a SNR level of around 4 dB is required at the receiver [97]. As shown in Fig. 7.10, the BS is allowed to be 222.5 metres away from the MN. On the other hand, the attacker needs to be as close as 150 metres away from the MN to be able to perform good AOA. This difference in linear distance can be thought of as a sort of ***security margin***.

7.7.4 Trade-off of Data Rate with Security

We calculate the SNR levels around a MN transmitting with the said shaped beam (from Fig. 7.4). We assume it uses a fixed EIRP of 20 dBm. Fig. 7.11(a) shows the plan view of the BS position relative to the MN. The BS receives with a SNR of 4 dB at this range (outer lobe). The concentric grid lines show the distances (in km) away from the MN. The adversary is required to be located within the *inner* lobes,

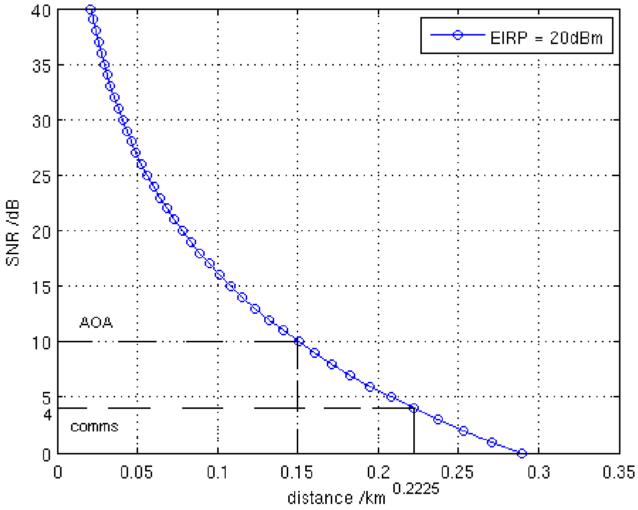


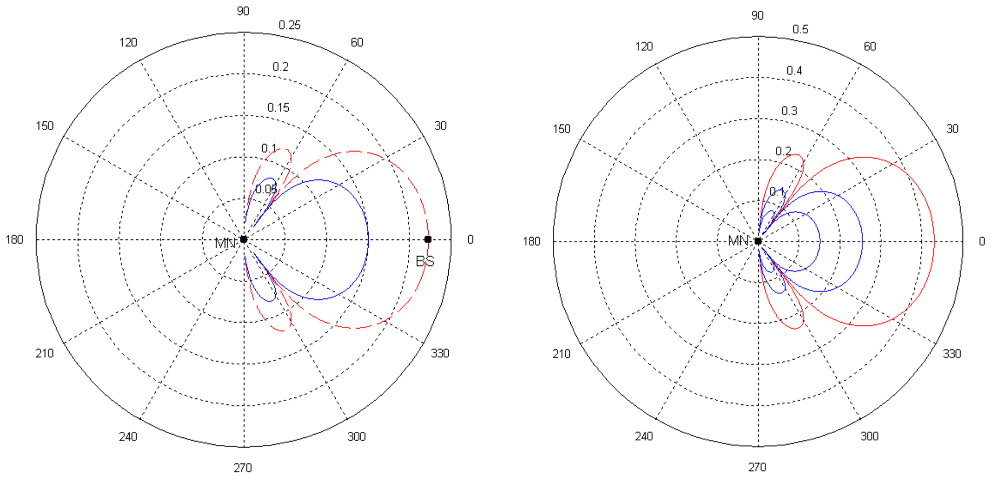
Figure 7.10: SNR required for AOA estimation and for successful comms

to be able to direction-fix the MN (it follows from Fig. 7.10).

In the earlier graphs, the MN is transmitting with a power level just sufficient to communicate at 1 Mbps with a BS 222.5 metres away. We now consider the security impact of increasing the data rate while retaining the same BS distance in Fig. 7.11(b). A SNR of 4 dB at the BS receiver is required for a 1 Mbps data rate for 802.11b, but a SNR of 8 dB is required for a 5.5 Mbps data rate [97]. A higher data rate necessitates a higher transmit power, invariably increasing the mobile node's radio signature even with a shaped beam, and enlarging the vulnerable area. The lobes in Fig. 7.11(b) correspond to the contours at which SNR = 10 dB for EIRP values of 36, 28 and 20 dBm. Transmitting with an EIRP of 20 dBm, the MN can be direction-fixed from 150 metres away, whereas if it transmits with an EIRP of 28 dBm (still well within FCC regulations), giving it a data rate better than 5.5 Mbps at the BS at the same position, it can be direction-fixed from 250 metres away, both distances referring to the main lobe direction. Clearly, data rate trades off with location privacy, and needs to be carefully managed.

7.8 Integrated Radio and Mobility Simulation

We used the IBM City Simulator [104] to generate node mobility output to drive the location anonymity analysis under adaptive beamforming (ABF) and omni-directional (OMNI) antenna radiation patterns. It simulates realistic motion of people moving in a city, carrying mobile wireless devices. We placed 100 mobile nodes on a grid, all communicating with one base station. The adversary places an increasing number of receivers at random points, whose locations are fixed for the length of the simulation. The job

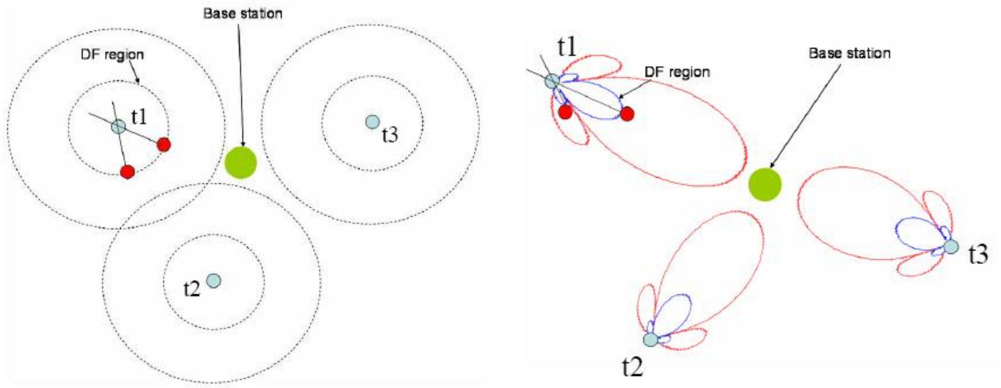


(a) Security Margin

(b) Data Rate/Security Trade-off

Fig. 7.11 Plan View of Radiation Coverage

of the adversary receiver equipment is to collect as much signal direction information as they can. The adversary aims to learn as much location information as possible. We wish to examine how the change of radiation pattern affects such information collection, and in turn location privacy.



(a) Omni Beams

(b) ABF Beams

Fig. 7.12 Integrated Radio and Mobility Simulation

Each mobile node exercises power control and optimizes transmission power so as to achieve an SNR of about 4 dB at the base station's receiver. The radio footprint that results is derived from procedures outlined earlier. An adversary receiver needs to be within the radiation zone of a SNR of 10 dB or better to be able to carry out direction-finding. These are simplistically represented in Fig. 7.12. t_1, t_2 and t_3 represent the different time snapshots of one moving node.

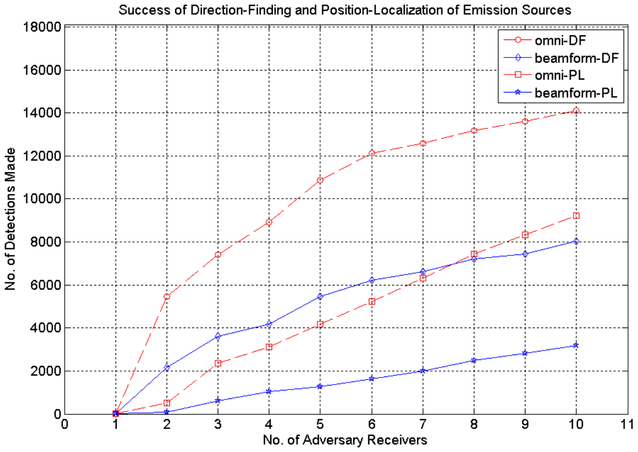


Fig. 7.13 Direction-Finding and Position Localization Success

7.9 Location Privacy Performance

Location privacy attacks depend on being able to track mobile nodes at a frequency high enough to reveal movement information. We consider the attacks in phases:

(1) Attacks that require the location information of as many nodes as possible at given points in time, without linking.

(2) *Location linking attacks* - that attempt to bind location information to unique nodes, hence linking as many positions and movements of a mobile node as it can.

7.9.1 Direction-Finding and Triangulation

In the first phase, the attacker attempts to gather as many direction estimates of the mobile nodes as possible, out of around 20×10^3 samples. We assume a detection if an attacker receiver falls within the beam coverage thresholded at SNR = 10 dB of a mobile node. If the attacker has two or more receivers successfully direction-fixing a victim node at a time instant, he can derive a triangulation of the node at that point in time. The results for both omnidirectional and adaptive beamforming beams are shown in Fig. 7.13. ABF is shown to be 6 to 7 times more covert (i.e. less detectable) than OMNI radiation pattern when the adversary uses a low number of receivers. Even when the adversary invests in a large number of receivers (eg. 10), the former still performs 3 times better. We also see that going from direction-finding (DF) to position-localization (PL), the attacker's success rate degrades more sharply against the ABF beams than for the OMNI beams.

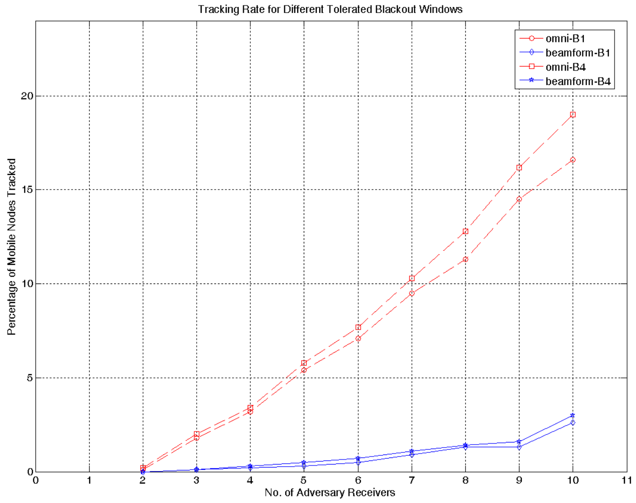


Fig. 7.14 Tracking Rate

7.9.2 Location Linkability Attacks and Blackout Window

Location linking is a powerful attack by which the attacker uses location information obtained at various points in time to de-anonymize an otherwise pseudonymous node and determine the path that it took. To carry this out, the attacker needs to know not only the location information of as many nodes as possible at various times, but to link them, he can only ‘lose’ them for a small window of time. We refer to this time as the ***blackout window B***, and use this as a metric of the robustness of different linking algorithms. The larger the value of ‘B’ that an algorithm can tolerate, the less its performance is impeded by factors such as a reduction in the radiation area. In the absence of precise simulations of linking algorithms (such as [85]), we approximate a linking attack’s robustness to the size of its tolerable blackout window. (In view of the different multi-target tracking algorithms already existing, this is not a fine-grained approximation, but it arguably serves our current purpose to just compare performance between different beam patterns.)

Conceptually, if S_t is the set of nodes whose movement the attacker has been able to track until time t, and let l_t be the corresponding location information, then at any time $t < t_k \leq t+B$, the attacker will be able to link position l_{t_k} with S_t with a probability close to 1. A linking attack algorithm loses tracks when it encounters a blackout larger than its B. Fig. 7.14 shows the tracking success for the two beam patterns for two sizes of B (specifically 1 and 4). ABF is shown to be 6$\sim$7 times better in resisting tracking. For the adversary, using more robust linking algorithms yields benefits equivalent to deploying more receivers.

7.9.3 Information-Theoretic Location Privacy

There are several ways one can express the quality of anonymity a system provides. Qualitative tags such as *absolute privacy, beyond suspicion, probable innocence, etc*, have been used in Crowds [157] and in other schemes. While this helps clarify, comparison across systems is difficult. In our analysis, we use entropy to measure the amount of information the attacker is missing for him to link node identity with location and movement. Our method is predicated on: "Anonymity of a system may be defined as the amount of information the attacker is missing to uniquely identify an actor's link to an action" [27; 171]. We remark that to possess anonymity, you require the existence of cover traffic for blending your messages with; similarly, to possess some location anonymity or location privacy, you require the presence of other mobile entities which have similar mobility patterns to yourself, for providing you with some cover. The more of such mobile entities there are around, and the more similar they are to you, the more beneficial it is for your anonymity. Conversely, if you are the only entity observed moving around, your anonymity is quite broken; thence your anonymity is heavily dependent on the presence and behavioural patterns of other potential victims.

In information-theoretic [174] terms, the anonymity of the entire system $\mathcal{A}$ is the entropy $\mathcal{E}$, of the probability distribution over all the actors α_i, that they committed a specific action. Hence,

$$\mathcal{A} = \mathcal{E}[\alpha_i] = -\sum_i Pr[\alpha_i] \log_2 Pr[\alpha_i] \qquad (7.1)$$

This expresses in bits the uncertainty experienced by the attacker; that is to say, the additional information bits he requires so as to identify a node.

For a number of nodes d_{t_k} that the attacker can track from t_0 to t_k, there are $N - d_{t_k}$ nodes whose whereabouts the attacker is uncertain about, and assumed non-trackable. To a coarse first approximation, the entropy of our privacy-enhancing system can be said to be:

$$\mathcal{A} = -\sum \frac{1}{N - d_{t_k}} \log_2\left(\frac{1}{N - d_{t_k}}\right) \qquad (7.2)$$

Thus, if a linking algorithm is very sensitive to blackouts, then as $B \longrightarrow 0$, $d_{t_k} \longrightarrow 0$. This is the case of maximal anonymity where the entropy of the privacy-enhancing system is $\mathcal{A} = -\log_2(\frac{1}{N})$, assuming uniform probability.

Accordingly, the performance of an attack using a robust linking algorithm ($B = 4$) is shown in Fig. 7.15. According to the metric, ABF outperforms OMNI, providing better information-theoretic location privacy. The different slope of the curves indicates that attacker investment in more receivers yields less steeply increasing benefits for the former than the latter.

It should be noted that because in our usage, we have assumed: (a) a uniform probability of the nodes which were not continuously tracked and hence considered un-trackable, (b) complete de-anonymisation of the continuously tracked nodes such that they each has a probability Pr of 1, and (c) restricting the

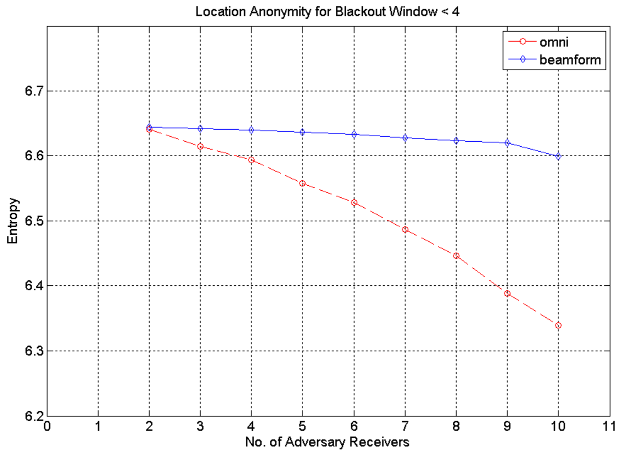


Fig. 7.15 Information-Theoretic Location Privacy of System

categories of trackability considered to just these two; hence the full granularity and capability of the information-theoretic approach in quantifying location privacy is not being realized here, although this advance can be taken as a reasonable first step. Thus, the disparity in information bits shown in Fig. 7.15 between the two schemes is not as large as one would have expected (they are derived at the moment only from the uniform probabilities of the untracked nodes). It seems intuitively likely that Equation 7.2 has somewhat underestimated the performance of ABF and overestimated the performance of OMNI, and that in reality, the ABF scheme would cause probability distributions of tracking to be more uniform than those in the OMNI scheme, thereby demonstrating better entropy. [1] We hope that when finer-grained data becomes available in future investigations, which allow individual probabilities to be more precisely captured, this would allow us to realize the potential of the information-theoretic approach more fully. In the meantime, we would suggest Fig. 7.14, over Fig. 7.15, as the more mature and fairer appraisal of the relative performance of the OMNI and ABF schemes.

7.10 Future Directions

Using our framework, other candidate wireless systems and beam patterns can be evaluated. Spread spectrum schemes, such as frequency hopping, which extend covertness into the time and frequency domains (requiring secret keying between the parties), may also be evaluated as another privacy-enhancing layer, on top of beamforming, as a plug-in to our framework. When we are able to consider more specific

[1] Considering a simple example of a binary random variable, a system with the associated probabilities of 0.8 and 0.2 has an entropy of 0.72 bits, whereas a system with both probabilities at 0.5 would have a higher (and the maximal) entropy of 1.0 bits.

tracking algorithms, our procedure for measuring location privacy and the metric could be refined.

In the longer term, we envisage that there needs to be a holistic look at the leakage of correlated identifying information across all layers of the protocol stack.

Looking into the future, it is possible to imagine that location privacy could be a credible value-added service (a sort of counter to ‘identification and authentication’, with elements of *deniability* and *repudiation*) which can be offered to consumers — something which can be turned on or turned off by the user according to personal preference, similar to today’s mobile cellular communications’ ‘caller ID non-display’ service.

7.11 Summary

Our contributions:

- We outlined how a passive adversary observing at the physical layer can bypass the countermeasures at the higher layers, and compromise a mobile node’s location privacy.

- We proposed a novel solution to improve unobservability at the physical layer by doing adaptive beamforming. We described the architecture of such a scheme. This will make the job of the attacker harder.

- We presented an evaluation framework and carried out simulations, showing that our proposal offers improved location privacy compared to the status quo solution of omnidirectional radio beams. The framework which we composed is flexible and extensible.

In this chapter, we have expanded on the notion of location privacy being a multi-layer problem, by considering issues at the lowest, i.e. physical, layer. It seems quite possible for a sufficiently motivated attacker to track your mobile node using direction-finding, even if your node has taken all the correct steps to anonymize itself at the higher layers. We have proposed using adaptive beamforming to obtain more unobservability. While our proposal sounds intuitively better, we decided to quantify the improvement gained through obtaining some hard numbers. We investigated using typical wireless LAN parameters, and compared between using the omnidirectional emitting beam, and adaptive beamforming. Using an evaluation framework which we composed, with radio simulation, and mobility simulation, we evaluated and verified that our proposal is certainly more location-privacy-enhancing. We offer our framework as an extensible first step for evaluating physical layer location privacy.

Chapter 8

Conclusion

After a decade on the wings, the field of pervasive computing has moved centre-stage. People in fields like context-aware computing, sensor networks, human-computer interaction, and middleware, together with those in the more traditional areas of wireless communications, applications, artificial intelligence and so on, are now working together to produce smart systems which when embedded into the environment will make services pervasively available. Some of the first products of the pervasive computing era are now deployed.

Along the way, over the course of PhD program, I have learnt some subtle lessons about transitioning security precepts from the traditional desktop and fixed tethered network models to the area of pervasive computing.

One set of the lessons which we have learnt includes: asymmetric cryptography (for key agreement, even if not for PKI) is now possible, and certainly desirable, to assure security in the face of increasingly sophisticated threats.

- I implemented an actual password guessing attack against a pervasive wireless technology which does not use asymmetric cryptography, and showed that the attack is feasible.
- I have implemented on pervasive devices (handphones) a password-authenticated key agreement protocol which uses asymmetric cryptography set on elliptic curve groups, and results show that the performance is good and latency is low.
- I proposed a password-authenticated identity-based (thereby using elliptic curves and pairing) inter-domain key agreement protocol, which does not require a PKI at the client side, hence avoiding the hassle of obtaining certificates and associated key management issues.

The other set of lessons I have learnt include: previous models of the adversary are outdated, and must be updated to reflect his different capabilities across the different channels present in a pervasive computing environment.

- I showed how key agreement protocols in a ubicomp scenario are insecure if the assumed attacker models are flawed.

- I proposed multi-channel protocols to carry out key agreement, for two-party as well as multi-party cases, making use of auxiliary channels in addition to a main insecure channel. I offered a multi-channel viewpoint to consider these.

- I carried out some implementations to validate our proposals, and to study performance and usability issues associated with different auxiliary channels, such as a machine visual channel and a melodic audio channel.

There remain important usability questions which must be answered in the next stage of work for other researchers in this field. What sort of channels would your average user find most comfortable and convenient, in addition to being amenable for combining into a secure protocol? In a related way, how would the operation of these channels be designed? In the longer-term, I believe the challenge is to standardize on these interaction models, so that the user becomes as accustomed to these as the desktop login screen to which they would instinctively enter their ‘username’ and ‘password’.

Assuring privacy is no less important than assuring authenticity, because a lack of care in designing privacy in the outset would make a surveillance society an increasing reality for all of us. I have learnt that certain deployed pervasive computing technologies have poor built-in location privacy features. Even an unsophisticated attacker can compromise their location privacy trivially. The next generation of devices ought to have these issues sorted out before deployment, and certainly before the capital outlay of the widespread deployments means that the unsolved problems would be there to stay with us for a long time.

- I sketched how location privacy is really a multi-layer problem,

- Concretely, I enumerated the location privacy problems in a pervasive wireless technology, namely Bluetooth, and proposed a MAC layer solution with stronger unlinkability for it, while also providing policy recommendations to improve unobservability.

- I showed how a passive attacker can compromise location privacy at the physical layer. I proposed using covert beam pattern to improve unobservability, and carried out extensive simulations, to verify the location privacy improvement our proposal offers over the status quo situation.

Evaluation tools, methodologies and perhaps testbeds need to be continually improved, so as to be able to better gauge the effectiveness of new proposals, and to this end we have provided an initial location privacy evaluation framework.

At the conclusion of this dissertation, on the whole I believe that there is still much cause for optimism for security in pervasive computing. Initial work has been done by different researchers. There remains much follow-up work to secure pervasive systems.

Scanning for broader lessons, it is possible to think of, say, the multi-channel work, as just a starting point (and not an ending point) for new research into ubicomp security, in that assumptions carried over from the homogeneous wireline paradigm break down dramatically when they encounter a new environment. It is worth noting that when the human is integrated into a protocol in meaningful ways, some problems can be easily solved; notice that it is so difficult (if not close to impossible) for a machine to authenticate the data origin of a short string of information, whereas it is so easy for a human to achieve the same. Perhaps it is time to remind ourselves to bring back the human user to the centre of things, instead of thinking of him as merely an obstacle to be overcome. This is fitting, because the ubicomp vision envisages the computer to become ‘invisible’, correspondingly that should imply that the human would become more ‘visible’. For that, we would need to figure out what a human does best, vis-a-vis a machine (even a very smart one).

We were well-acquainted with the following features of ubicomp: numerousness of devices, intermittent connectivity, resource constraints of processors, and dynamic ad-hoc networking. To these mix, we add our own observation of: multiplicity of channel types, and human mediation. We may yet ask: what are the other security expectations carried over from the wireline paradigm that are unrealistic for the types of transactions which we may want to carry out in ubicomp? Where, and how else, would broadbrush unconditionally high security assumptions be perhaps unsuitable (like the Dolev-Yao adversary when extended to ubicomp channels)? What imperfections can we truly live with, and what actions can we let the human user perform for himself? I admit I have no ready answers, but is this a place to re-introduce or re-visit multi-lateral security, if we want to, say, allow the flexibility of having multiple users use the same publicly owned devices for short periods of time? And how may we reliably manage the revocation of trust, assuming the bootstrapping of this in the first place has been solved? Many other interesting and exciting questions can be posed, and hopefully answered, in time. Ubicomp is set to be a major thrust area in computer science for a while, so the assuring of the security of ubicomp should continue to be a fertile area, with attendant security flaws, as well as unexpected solutions, revealing themselves every once a while.

Before we let over-optimism get the better of us, perhaps it was fortuitousness that the multi-channel thread of work, for one, has managed to turn up some viable solutions; not all problems may turn out to be tractable. However, it would be worth keeping in mind that generally the problems would be more soluble if (a) we pay attention to these before the unsolved problems become too tightly integrated and entangled into the ubicomp eco-system, (b) we are clear about the engineering trade-offs that would invariably need to be made, and (c) we are continually able to revise obsolescent assumptions and re-calibrate our solutions.

I believe that information security in this increasingly networked information society is simply too important to be left only to computer scientists and computer engineers. Stake-holders from the social, economic and political spheres should be engaged with the problems too. Certainly, there are outstanding issues in information security of alignment of incentives. However, once we all find common cause, our hard-gained technical proposals would be able to find true expression in new generations of ubicomp products, that are secure out-of-the-box for the end-user, and are usable at the same time. That would certainly be a ubicomp future worth looking forward to.

Appendix A

Appendix A

Appendix A describes the choice of the elliptic curve group used in the work in Chapter 1.

A.1 The Characteristic 2 Finite Field

We first consider a finite field of q elements, written as $\mathbb{F}_q$. In our implementation, we will use a construction based on elliptic curves over a characteristic 2 (i.e. binary) field. So $q = 2^m$, for some positive integer m, and the finite field used is written as $\mathbb{F}_{2^m}$. There is only one characteristic 2 finite field $\mathbb{F}_{2^m}$ for each power 2^m of 2 with $m \geq 1$; there are different ways to represent the elements of $\mathbb{F}_{2^m}$. One of the ways of representing these would be the set of binary polynomials of degree $m-1$ or less. The addition and multiplication of the field elements would be defined in terms of an irreducible binary polynomial $f(x)$ of degree m, known as the reduction polynomial.

A.2 Elliptic Curves over the Characteristic 2 Field

Let $a, b \in \mathbb{F}_{2^m}$ and where $b \neq 0$ in $\mathbb{F}_{2^m}$. An elliptic curve $E(\mathbb{F}_{2^m})$ over $\mathbb{F}_{2^m}$ consists of points $P = (x, y)$ for $x, y \in \mathbb{F}_{2^m}$ to the equation:

$$y^2 + x.y = x^3 + a.x^2 + b$$

The choice of a characteristic 2 field is made because this is easier to implement than others.

A.3 Elliptic Curve Domain Parameters

Elliptic curve domain parameters over a characteristic 2 field are a septuple:

$$T = (m, f(x), a, b, G, n, h)$$

G is a base point (x_G, y_G) of order n (which is prime) on the curve $E(\mathbb{F}_{2^m})$. We can denote the number of points on $E(\mathbb{F}_{2^m})$ as $\#E(\mathbb{F}_{2^m})$. The integer h is the co-factor, and $h = \#E(\mathbb{F}_{2^m})/n$

A.4 Selection of Parameters

Typically, the characteristic 2 finite fields $\mathbb{F}_{2^m}$ used would have the following values of m:

$$m \in \{113, 131, 163, 193, 233, 239, 283, 409, 571\}$$

These were designed to enable designers to deploy implementations capable of meeting common security requirements [39].

We chose $m = 163$. This is roughly equivalent to 1024 bit length of RSA and Diffie-Hellman. Hence, the reduction polynomial $f(x)$ would be:

$$f(x) = x^{163} + x^7 + x^6 + x^3 + 1$$

Elliptic curves over $\mathbb{F}_{2m}$ generally consist of two types of parameters: those associated with a Koblitz curve, and those chosen verifiably at random. A Koblitz curve is one with $a, b \in \{0, 1\}$. Koblitz curves allow particularly efficient implementation, but their extra structure also aids attack to some degree. In our trade-off, we prefer randomly chosen parameters, such as some suggested ones in [40].

References

[1] ISO/IEC-15408 (1999). *ISO/IEC-15408 Common Criteria for Information Technology Security Evaluation v2.1*. National Institute of Standards and Technology, Washington DC, 1999. `http://csrc.nist.gov/cc`. 104

[2] Martín Abadi and Phillip Rogaway. "Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption)". In "Proceedings of IFIP International Conference on Theoretical Computer Science (IFIP TCS2000)", vol. 1872 of *Lecture Notes in Computer Science*, pp. 3–22. Springer-Verlag, 2000. 38

[3] Gregory D. Abowd and Elizabeth D. Mynatt. "Charting Past, Present, and Future Research in Ubiquitous Computing". *ACM Transactions on Computer-Human Interaction*, **7**(1):29–58, March 2000.

[4] Wi-Fi Alliance. "Wi-Fi CERTIFIED for Wi-Fi Protected Setup: Easing the User Experience for Home and Small Office Wi-Fi Networks", 2007. `http://www.wi-fi.org/files/wp_18_20070108_Wi-Fi_Protected_Setup_WP_FINAL.pdf`. 85

[5] Ross Anderson. "Why Cryptosystems Fail". *Communications of the ACM*, **37**(11):32–40, November 1994.

[6] Ross Anderson. *Security Engineering – A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, 2001.

[7] Ross Anderson and Mark Lomas. "Fortifying Key Negotiation Schemes with Poorly Chosen Passwords". *Electronics Letters*, **30**(13):1040–1041, 1994. 20

[8] N. Asokan and P. Ginzboorg. "Key Agreement in Ad-hoc Networks". *Proceedings of Nordsec 1999*, Nov 1999. 88, 95

[9] Tuomas Aura, Pekka Nikander and Jussipekka Leiwo. “DOS-resistant Authentication with Client Puzzles”. In “Proceedings of 8th International Workshop on Security Protocols, 2000”, vol. 2133 of *Lecture Notes in Computer Science*. Springer-Verlag, 2001. 31

[10] Gildas Avoine. “Radio Frequency Identification: Adversary Model and Attacks on Existing Protocols”. Tech. Rep. LASEC-REPORT-2005-001, Swiss Federal Institute of Technology, Lausanne (EPFL), September 2005. 119

[11] Gildas Avoine and Philippe Oechslin. “RFID Traceability: A Multilayer Problem”. In “Proceedings of 9th International Conference on Financial Cryptography and Data Security - FC 05”, vol. 3570 of *Lecture Notes in Computer Science*, pp. 125–140. Springer, 2005. 106

[12] Paramvir Bahl and Venkata N. Padmanabhan. “RADAR: an in-building RF-based user location and tracking system”. In “Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)”, pp. 775–784. March 2000.

[13] Constantine Balanis. *Antenna Theory Analysis and Design*. John Wiley and Sons Inc., 1997. 128

[14] Dirk Balfanz, Glenn Durfee, Narendar Shankar, Diana Smetters, Jessica Staddon and Hao-Chi Wong. “Secret Handshakes from Pairing-Based Key Agreements”. In “Proceedings of the 2003 IEEE Symposium on Security and Privacy”, 2003. 119

[15] Dirk Balfanz, D. K. Smetters, Paul Stewart and H. Chi Wong. “Talking to strangers: authentication in ad-hoc wireless networks”. *Network and Distributed System Security Symposium*, Feb 2002. 60, 63, 66, 88

[16] Elaine Barker, Don Johnson and Miles Smid. “NIST Special Publication 800-56A Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)”. Tech. rep., National Institute of Standards and Technology, March 2007. `http://csrc.nist.gov/CryptoToolkit/kms/SP800-56Arev1_3-8-07.pdf`. 20

[17] Paulo S. L. M. Barreto, Steven Galbraith, Colm O hEigeartaigh and Michael Scott. “Efficient Pairing Computation on Supersingular Abelian Varieties”. Cryptology ePrint Archive, Report 2004/375, September 2005. `http://eprint.iacr.org/2004/375/`. 56

[18] Klaus Becker and Uta Wille. “Communication Complexity of Group Key Distribution”. *ACM Conference on Computer and Communications Security*, pp. 1–6, 1998. 88, 95

[19] D. Bellare, A. Desai, D. Pointcheval and P. Rogaway. “Relations Among Notions of Security for Public-Key Encryption Schemes”. In H. Krawczyk (ed.), “Advances in Cryptology - Proceedings of CRYPTO ’98”, vol. 1462 of *Lecture Notes in Computer Science*, pp. 26–45. Springer-Verlag, 1998. 51

[20] Mihir Bellare, Ran Canetti and Hugo Krawczyk. “Keying Hash Functions for Message Authentication”. In Koblitz (ed.), “Advances in Cryptology - Crypto 96 Proceedings”, vol. 1109 of *Lecture Notes in Computer Science*. 1996. 62, 74

[21] Mihir Bellare, Ran Canetti and Hugo Krawczyk. “Message Authentication using Hash Functions – The HMAC Construction”. *CryptoBytes*, **2**(1), Spring, 1996. 74

[22] Mihir Bellare, David Pointcheval and Phillip Rogaway. “Authenticated key exchange secure against dictionary attacks”. In Bart Preneel (ed.), “Advances in Cryptology - Proceedings of EUROCRYPT 2000”, vol. 1807 of *Lecture Notes in Computer Science*, pp. 139–155. Springer-Verlag, 2000. 38, 47, 48, 79

[23] Mihir Bellare and Phillip Rogaway. “Random Oracles are Practical: A Paradigm for Designing Efficient Protocols”. In “ACM Conference on Computer and Communications Security 1993”, p. 6273. 1993. 38

[24] Mihir Bellare and Phillip Rogaway. “Entity Authentication and Key Distribution”. In “Proceedings of Advances in Cryptology CRYPTO ’93”, vol. 773 of *Lecture Notes in Computer Science*, pp. 232–249. 1994. 38

[25] Steven M. Bellovin and Michael Meritt. “Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks”. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp. 72–74, May 1992. 29, 31, 38, 88

[26] Steven M. Bellovin and Michael Meritt. “Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise”. *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 244–250, 1993. 38

[27] Alastair R. Beresford and Frank Stajano. “Location Privacy in Pervasive Computing”. *IEEE Pervasive Computing*, **3**(1):46–55, 2003. 103, 105, 112, 118, 122, 123, 140

[28] Alastair R. Beresford and Frank Stajano. “Mix Zones: User Privacy in Location-aware Services”. *IEEE Workshop on Pervasive Computing and Communication Security*, 2004. 103, 122, 123

[29] Bluetooth SIG Security Experts Group. “Bluetooth Security White Paper”, April 2002. Revision 1.0. 24

[30] Bluetooth Special Interest Group. “Simple Pairing Whitepaper”. Version V10r00, August 2006. `http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf`. 39, 85

[31] Dan Boneh and Matthew K. Franklin. “Identity-based encryption from the Weil pairing”. In Joe Kilian (ed.), “Advances in Cryptology - Proceedings of CRYPTO 2001”, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229. Springer-Verlag, 2001. 41, 44, 45, 46, 51

[32] Colin Boyd and Anish Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, Berlin, 2003. 20, 29

[33] Victor Boyko, Philip Mackenzie and Sarvar Patel. “Provably Secure Password Authentication and Key Exchange Using Diffie-Hellman”. *EuroCrypt 2000*, pp. 156–171, 2000. 20, 31

[34] Stefan Brands and David Chaum. “Distance-bounding protocols”. In “Proceedings of Advances in Cryptology, EUROCRYPT 93”, vol. 765 of *Lecture Notes in Computer Science*, pp. 344–359. Springer-Verlag, May 1993. 122

[35] Emmanuel Bresson, Olivier Chevassut and David Pointcheval. “Dynamic Group Group Diffie-Hellman Key Exchange under standard assumptions”. In “Proceedings of Advances in Cryptology - EUROCRYPT 2002”, vol. 2332 of *Lecture Notes in Computer Science*, pp. 321–336. 2002.

[36] Alison Brown and Neil Gerein. “Test Results of a Digital Beamforming GPS Receiver in a Jamming Environment”. In “Proceedings of ION GPS 2001”, Salt Lake City, Utah, September 2001. 125, 126

[37] Mike Burmester and Yvo Desmedt. “A Secure and Efficient Conference Key Distribution System”. In “Advances in Cryptology - EUROCRYPT ’94”, vol. 950 of *Lecture Notes in Computer Science*, pp. 275–286. Springer, 1994. 88, 96

[38] James J. Caffery and Gordon L. Stuber. “Overview of Radiolocation CDMA Cellular Systems”. *IEEE Communications Magazine*, **36**(4):38–45, April 1998.

[39] Certicom Corp. “SEC 1: Elliptic Curve Cryptography”. *Standards for Efficient Cryptography*, September 2000. Version 1.0. 32, 148

[40] Certicom Corp. “SEC 2: Recommended Elliptic Curve Domain Parameters”. *Standards for Efficient Cryptography*, September 2000. Version 1.0. 33, 148

[41] David Chaum. “Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms”. *Communications of the ACM*, **24**(2):84–88, Feb 1981. 112

[42] David Chaum. “The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability”. *Journal of Cryptology*, **1**:65–75, 1988.

[43] L. Chen, Keith Harrison, A. Moss, David Soldera and Nigel P. Smart. "Certification of Public Keys within an Identity Based System". In Agnes Hui Chan and Virgil D. Gligor (eds.), "Proceedings of the 5th International Information Security Conference, ISC 2002", vol. 2433 of *Lecture Notes in Computer Science*, pp. 322–333. Springer-Verlag, 2002. 47

[44] Kim-Kwang Raymond Choo, Colin Boyd and Yvonne Hitchcock. "Errors in Computational Complexity Proofs for Protocols". In B. Roy (ed.), "Proceedings of Asiacrypt 2005", vol. 3788 of *Lecture Notes in Computer Science*, pp. 624–843. 2005. 38

[45] Bruce Christianson, Michael Roe and David Wheeler. "Secure Sessions from Weak Secrets". In Bruce Christianson, Bruno Crispo, James A. Malcolm and Michael Roe (eds.), "Proceedings of 11th International Workshop on Security Protocols", vol. 3364 of *Lecture Notes in Computer Science*. Springer-Verlag, 2003. 20

[46] Certicom Corp. "Certicom Announces Elliptic Curve Cryptography Challenge Winner". `http://www.certicom.com/index.php?action=company,press_archive&view=307`. 33

[47] Certicom Corp. "The Certicom ECC Challenge". `http://www.certicom.com/research/ecc_challenge.html`. 33

[48] COST. "COST Action 231: Digital mobile radio towards future generation systems: Final Report", 1999. 128

[49] Sadie Creese, Michael Goldsmith, Bill Roscoe and Irfan Zakiuddin. "The Attacker in Ubiquitous Computing Environments: Formalising the Threat Model". In "Proceedings of the Workshop on Formal Aspects in Security and Trust (FAST) 2003", Pisa, Italy, September 2003. 60, 71, 72

[50] S.J. Creese, M.H. Goldsmith, A.W. Roscoe and Ming Xiao. "Bootstrapping Multi-Party Ad-Hoc Security". In "Proceedings of ACM Symposium on Applied Computing (SAC 2006)", pp. 369–375. 2006. 88

[51] Giovanni Di Crescenzo, Yuval Ishai and Rafail Ostrovsky. "Non-interactive and Non-malleable Commitment". In "Proceedings of Symposium of Theory of Computing (STOC) '98", pp. 141–150. 1998. 74

[52] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky and Adam Smith. "Efficient and Non-interactive Non-malleable Commitment". In B. Pfitzmann (ed.), "Advances in Cryptology EUROCRYPT '01", vol. 2045 of *Lecture Notes in Computer Science*, pp. 40–59. Springer-Verlag, Innsbruck, Austria, 2001. 74

[53] J. Cuellar, J. Morris, D. Mulligan, J. Peterson and J. Polk. "Geopriv Requirements, RFC 3693", February 2004. `http://www.ietf.org/rfc/rfc3693.txt`. 103

[54] Dan Cvrcek, Marek Kumpost, Vashek Matyas and George Danezis. "A Study on The Value of Location Privacy". In "Proceedings of 5th ACM Workshop on Privacy in Electronic Society (WPES)", pp. 109–118. 2006. 104

[55] Joan Daemen and Vincent Rijmen. "AES Proposal: Rijndael", 3 September 1999.

[56] George Danezis, Stephen Lewis and Ross Anderson. "How Much is Location Privacy Worth". In "Fourth Workshop on the Economics of Information Security (WEIS)", 2005. 104

[57] Diego López de Ipiña, Paulo R. S. Mendonca and Andy Hopper. "TRIP: a low-cost vision-based location system for ubiquitous computing". *Personal and Ubiquitous Computing*, **6**(3):206–219, May 2002. 81

[58] Roberto Delicata. *A Security Analysis of the CLIQUES Protocol Suite*. Master's thesis, Oxford University, 2002. 87

[59] Dorothy E. Denning and Peter F. MacDoran. "Location-based authentication: Grounding cyberspace for better security". *Computer Fraud and Security*, pp. 12–16, February 1996. `http://www.cs.georgetown.edu/~denning/infosec/Grounding.txt`.

[60] Dorothy E. Denning and Giovanni Maria Sacco. "Timestamps in Key Distribution Protocols". *Communications of the ACM*, **24**(8):533–536, August 1981. 35

[61] Anind K. Dey and Gregory D. Abowd. "Towards a better understanding of context and context-awareness". In "Proceedings of the CHI 2000 Workshop on "The What, Who, Where, When, Why and How of Context-Awareness"", The Hague, Netherlands, April 2000.

[62] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography". *IEEE Transactions on Information Theory*, **22**(6):644–654, November 1976. 30

[63] D. Dolev and A. Yao. "On the security of public key protocols". In "Proceedings of the IEEE 22nd Annual Symposium of Computer Science", pp. 350–357. 1981. 37, 60, 61

[64] Danny Dolev, Cynthia Dwork and Moni Naor. "Non-Malleable Cryptography". In "Proceedings of 23rd Symposium on Theory of Computing (STOC)", pp. 542–552. ACM Press, 1991. 74

[65] Carl M. Ellison. "The nature of a useable PKI". *Computer Networks*, **31**(8):823–830, April 1999.

[66] FCC. "FCC Part 15.247 Operation within the bands 902-928 MHz, 2400-2483.5 MHz, and 5725-5850 MHz", 2002. 128

[67] Kenneth P. Fishkin and Sumit Roy. "Enhancing RFID Privacy via Antenna Energy Analysis". In "MIT RFID Privacy Workshop", November 2003. `http://www.rfidprivacy.us/2003/papers/fishkin.pdf`. 122

[68] Steven D. Galbraith. "Supersingular curves in cryptography". In Colin Boyd (ed.), "Advances in Cryptology - Proceedings of ASIACRYPT 2001", vol. 2248 of *Lecture Notes in Computer Science*, pp. 495–513. Springer-Verlag, 2001.

[69] Simson L. Garfinkel, Ari Juels and Ravi Pappu. "RFID Privacy: An Overview of Problems and Proposed Solutions". *IEEE Security & Privacy*, **3**(3), 2005. 102, 104

[70] Christian Gehrmann, Chris J. Mitchell and Kaisa Nyberg. "Manual authentication for wireless devices". *Cryptobytes*, **7**(1):29–37, 2004. 60, 63, 65, 67, 68, 88, 92

[71] Christian Gehrmann and Kaisa Nyberg. "Enhancements to Bluetooth Baseband Security". *Proceedings of Nordsec 2001*, Nov 2001. 62, 103, 106, 112, 113, 114, 117

[72] Ivan Getting. "The Global Positioning System". *IEEE Spectrum*, **30**(12):36–47, December 1993. 124

[73] Lal C. Godara. "Application of Antenna Array to Mobile Communications, Part II: Beam-forming and Directional-of-Arrival consideration". *Proc. of the IEEE*, **85**(8), Aug 1997. 131

[74] Shafi Goldwasser and Silvio Micali. "Probabilistic encryption & how to play mental poker keeping secret all partial information". In "Proceedings of 14th Annual ACM Symposium on Theory of Computing", pp. 365–377. San Francisco, 1982. 38

[75] Li Gong. "Collisionful keyed hash functions with selectable collisions". *Information Processing Letters*, **55**(3):167–170, Aug 1995. 20

[76] Li Gong, T. Mark A. Lomas, Roger M. Needham and Jerome H. Saltzer. "Protecting Poorly Chosen Secrets from Guessing Attacks". *IEEE Journal on Selected Areas in Communications*, **11**(5):648–656, June 1993. 42

[77] Michael T. Goodrich, Michael Sirivianos, John Solis, Gene Tsudik and Ersin Uzun. "Loud and Clear: Human-verifiable authentication based on audio". In "Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS)", 2006. 60

[78] Vanessa Gratzer, David Naccache and David Znaty. "Law Enforcement, Forensics and Mobile Communications". In "Keynote talk at 3rd IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2006)", Pisa, Italy, March 2006. 103

[79] Bluetooth Special Interest Group. “Specification of the Bluetooth System version 1.1”, February 2001. 23, 24, 28, 106, 107

[80] Bluetooth Special Interest Group. “Specification of the Bluetooth System version 1.2”, November 2003. 23, 27, 28, 106, 107

[81] Bluetooth Special Interest Group. “Specification of the Bluetooth System version 2.0”, November 2004. 23, 106, 107

[82] Wireless USB Promoter Group. “Association Models Supplement to the Certified Wireless Universal Serial Bus Specification, Revision 1.0”, March 2006. Available from `http://www.usb.org/developers/wusb/wusb_2007_0214.zip`. 85

[83] Marco Gruteser and Dirk Grunwald. “Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking”. *MobiSys*, 2003. 122

[84] Marco Gruteser and Dirk Grunwald. “Enhancing Location Privacy in Wireless LAN through Disposable Interface Identifiers: A Quantitative Analysis”. *First ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, 2003. 103, 112, 122

[85] Marco Gruteser and Baik Hoh. “On the Anonymity of Periodic Location Samples”. In “Proceedings of Security in Pervasive Computing (SPC) 2005”, Germany, 2005. 122, 127, 139

[86] Shai Halevi and Hugo Krawczyk. “Public-Key Cryptography and Password Protocols”. *ACM Transactions on Information and System Security*, **2**(3):25–60, 1999. 20, 23, 54

[87] Richard Harrison. *Symbian OS C++ for Mobile Phones: 1*. John Wiley and Sons Inc., 2003. 37

[88] Mike Hazas and Andy Ward. “A High Performance Privacy-Oriented Location System”. In “Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)”, pp. 216–223. Dallas-Fort Worth, USA, March 2003.

[89] Urs Hengartner and Peter Steenkiste. “Protecting access to people location information”. In “Proceedings of the First International Conference on Security in Pervasive Computing”, March 2003.

[90] Dirk Henrici and Paul Muller. “Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers”. In “Proceedings of Workshop on Security of Pervasive Computing and Communications (PerSec 2004)”, 2004. 103

[91] Maarit Hietalahti. *Efficient key agreement for ad-hoc networks*. Master’s thesis, Helsinki University of Technology, Department of Computer Science and Engineering, May 2001. 88

[92] Jaap-Henk Hoepman. “The Ephemeral Pairing Problem”. *8th International Conference on Financial Cryptography*, **3110**:212–226, 2004. 60, 73, 79, 88

[93] Baik Hoh and Marco Gruteser. “Protecting Location Privacy Through Path Confusion”. In “Proceedings of the IEEE/CreateNet International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm)”, Athens, Greece, 2005. 122

[94] L. Huang and H. Yamaneet. “Enhancing wireless location privacy using silent period”. In “5th Workshop on Privacy Enhancing Technologies (PET)”, Dubrovnik, Croatia, 2005. 123

[95] IEEE. “IEEE P1363 - Standard Specifications For Public-Key Cryptography”. `http://grouper.ieee.org/groups/1363`. 35

[96] IEEE. “IEEE P1363.2: Password-Based Public-Key Cryptography”. `http://grouper.ieee.org/groups/1363/passwdPK/index.html`. 54

[97] IEEE. “Supplements To IEEE Standard For Information Technology- Telecommunications And Information Exchange Between Systems-Local And Metropolitan Area Networks-Specific Requirements-Part 11: Wireless LAN Medium Access Control (MAC) And Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension In the 2.4GHz Band”. *IEEE Std 802.11b-1999*, 1999. 135, 136

[98] David Jablon. “Strong Password-Only Authenticated Key Exchange”. *Computer Communication Review*, **26**(5):5–26, Oct 1996. 20, 31

[99] David Jablon. “Extended Password Key Exchange Protocols Immune to Dictionary Attack”. *Proceedings of the Sixth Workshops on Enabling Technologies: Infrastructure for Collaborative Engineering*, **11**:248–255, June 1997. 31

[100] Markus Jakobsson and Susanne Wetzel. “Security Weaknesses in Bluetooth”. *Proceedings of RSA Conference 2001*, **2020**, 2001. 27, 103, 106, 109, 113

[101] Ari Juels. “RFID Security and Privacy: A Research Survey”. *IEEE Journal on Selected Areas in Communications*, **24**(2):381–394, February 2006. 102

[102] Ari Juels and John Brainard. “Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks”. In S. Kent (ed.), “Proceedings of 1999 ISOC Networks and Distributed System Security Symposium”, pp. 151–165. February 1999. 31

[103] Jonathan Katz and Moti Yung. “Scalable Protocols Authenticated Group Key Exchange”. In “Proceedings of CRYPTO 2003”, vol. 2729 of *Lecture Notes in Computer Science*, pp. 110–125. Springer-Verlag, Santa Barbara, USA, 2003. 88, 95

[104] J. Kaufman, J. Myllymaki and J. Jackson. "IBM City Simulator Spatial Data Generator 2.0", 2001. 136

[105] D. Kesdogan, H. Federrath, A. Jerichow and A. Pfitzmann. "Location Management Strategies increasing Privacy in Mobile Communication Systems". *Proceedings of the 12th IFIP SEC*, 1996. 102

[106] Yongdae Kim, Adrian Perrig and Gene Tsudik. "Simple and fault-tolerant key agreement for dynamic collaborative groups". In "7th ACM Conference on Computer and Communications Security", pp. 235–244. 2000. 95

[107] Tim Kindberg and Kan Zhang. "Validating and securing spontaneous associations between wireless devices". In "Proceedings of the 6th Information Security Conference (ISC '03)", vol. 2851 of *Lecture Notes in Computer Science*. Springer-Verlag, October 2003. 60

[108] Neal Koblitz. "Elliptic Curve Cryptosystems". *Mathematics of Computation*, **48**(177):203–209, 1987. 32

[109] Neal Koblitz and Alfred J. Menezes. "Another Look at "Provable Security". II". Cryptology ePrint Archive, Report 2006/229, 2006. 37, 38

[110] Neal Koblitz and Alfred J. Menezes. "Another Look at "Provable Security"". *Journal of Cryptology*, **20**(1):3–37, 2007. 34, 37, 38

[111] J. Kohl and C. Neuman. "The Kerberos Network Authentication Service (V5)", September 1993. `http://www.ietf.org/rfc/rfc1510.txt`. IETF RFC 1510. 53

[112] Tadayoshi Kohno, Andre Broido and K. C. Claffy. "Remote device fingerprinting". In "Proceedings of the 2005 IEEE Symposium on Security and Privacy", May 2005. 103

[113] Taekyoung Kwon. "Authentication and key agreement via memorable password". *Contribution to the IEEE P1363 study group for Future PKC Standards*, 2000. `http://eprint.iacr.org/2000/026`. 33, 34, 39

[114] Taekyoung Kwon. "Authentication and key agreement via memorable password". *ISOC Network and Distributed System Security Symposium*, Feb 2001. 20, 31, 33

[115] Taekyoung Kwon. "Summary of AMP (Authentication and key agreement via Memorable Passwords)", Aug 2003. `http://dasan.sejong.ac.kr/~tkwon/amp.html`. 33, 35

[116] Marc Langheinrich. "Privacy by design – principles of privacy-aware ubiquitous systems". In G.D. Abowd, B. Brumitt and S. Shafer (eds.), "Proceedings of the Third International Conference

on Ubiquitous Computing (UbiComp)", vol. 2201 of *Lecture Notes in Computer Science*, pp. 273–291. Springer-Verlag, 2001.

[117] Marc Langheinrich. "A privacy awareness system for ubiquitous computing environments". In "Proceedings of the Fourth International Conference on Ubiquitous Computing (UbiComp)", pp. 237–245. Springer-Verlag, 2002.

[118] Sven Laur, N. Asokan and Kaisa Nyberg. "Efficient Mutual Data Authentication Using Manually Authenticated Strings". Cryptology ePrint Archive, Report 2005/424, 2005. 60, 63, 79

[119] Sven Laur and Kaisa Nyberg. "Efficient Mutual Data Authentication Using Manually Authenticated Strings". In "Proceedings of 5th International Conference on Cryptology and Network Security (CANS 2006)", vol. 4301 of *Lecture Notes in Computer Science*, pp. 90–107. Springer, Suzhou, China, 2006. 60

[120] Sven Laur and Kaisa Nyberg. "Efficient Mutual Data Authentication Using Manually Authenticated Strings: Extended Version". Cryptology ePrint Archive, Report 2006/424, 2006. 60

[121] Arjen K. Lenstra. "Further progress in hashing cryptanalysis", February 2005. `http://cm.bell-labs.com/who/akl/hash.pdf`. 36

[122] Arjen K. Lenstra and Eric R. Verheul. "Selecting Cryptographic Key Sizes". *Journal of Cryptology*, **14**(4):255–293, 2001. 33

[123] Hoon Wei Lim and Kenny G. Paterson. "Secret public key protocols revisited". In "Proceedings of the 14th International Workshop on Security Protocols 2006", March 2006. 42, 48

[124] Min Lin and Ian Wassell. "Impact of Channel Sounder Frequency Offset On The Estimation of Channel Parameters". In "IEEE Vehicular Technology Conference 2006 Fall", September 2006. 134

[125] T. Mark A. Lomas, Li Gong, Jerome H. Saltzer and Roger Needham. "Reducing risks from poorly chosen keys". *ACM Operating Systems Review*, **23**(5):14–18, Dec 1989. 20

[126] Shamus Software Ltd. "Multiprecision Integer and Rational Arithmetic C/C++ Library". `http://www.shamus.ie/`. 36, 81

[127] David A. Lynch. *Introduction to RF Stealth*. SciTech Publishing Inc, 2004.

[128] Philip Mackenzie. "More Efficient Password-Authenticated Key Exchange". *RSA Conference, Cryptographer's Track*, pp. 361–377, 2001.

[129] Philip MacKenzie. “On the Security of the SPEKE Password-Authenticated Key Agreement Protocol”. Cryptology ePrint Archive, Report 2001/057, July 2001. 35

[130] Wenbo Mao. *Modern Cryptography Theory and Practice*. Prentice Hall, 2004.

[131] James Massey, Gurgen Khachatrian and Melsik Kuregian. “Nomination of SAFER+ as Candidate Algorithm for the Advanced Encryption Standard”. *Proceedings of the 1st AES Candidate Conference*, 1998. 24

[132] Ueli Maurer. “Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms”. In Y. Desmedt (ed.), “Advances in Cryptology - Crypto ’94”, No. 839 in Lecture Notes in Computer Science, pp. 271–281. Springer-Verlag, 1994. 31

[133] Ueli Maurer and Pierre Schmid. “A Calculus for Secure Channel Establishment in Open Networks”. In “Proceedings of ESORICS 94”, vol. 875 of *Lecture Notes in Computer Science*. Springer, 1994. 60

[134] Rene Mayrhofer. “Towards an open source toolkit for ubiquitous device authentication”. In “Proceedings of 4th IEEE International Workshop on Pervasive Computing and Communication Security”, pp. 247–252. March 2007. 85

[135] Noel McCullagh. “Securing e-mail with identity-based encryption”. *IT Professional*, **7**(3):61–64, May/June 2005. 56

[136] Jonathan M. McCune, Adrian Perrig and Michael K. Reiter. “Seeing is Believing: Using Camera-Phones for Human-Verifiable Authentication”. *Computer Science Technical Report CMU-CS-04-174*, 2004. 60, 62, 66

[137] Alfred J. Menezes, Paul van Oorschot and Scott Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. 23, 29, 31, 36, 74

[138] Microsoft. “How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases”. `http://support.microsoft.com/default.aspx?scid=KB;EN-US;q299656`. Q299656, version 9.3, updated 14 August 2007. 20

[139] George A. Miller. “The Magic Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information”. *Psychological Review*, **63**:81–97, 1956. 21

[140] Victor S. Miller. “Use of Elliptic Curves in Cryptography”. In “Proceedings of Advances in Cryptology, CRYPTO 85”, vol. 218 of *Lecture Notes in Computer Science*, pp. 417–426. Springer, 1986. 32

[141] Mobiwave. “Bluetooth Protocol Analyzer BPA-D10”. http://www.mobiwave.com. 25

[142] Max Moser. “Busting The Bluetooth Myth - Getting RAW Access”, 2007. http://www.remote-exploit.org/research/busting_bluetooth_myth.pdf. 27

[143] Moni Naor, Gil Segev and Adam Smith. “Tight Bounds for Unconditional Authentication Protocols in the Manual Channel and Shared Key Models”. In Cynthia Dwork (ed.), “Proceedings of CRYPTO 06”, vol. 4117 of *Lecture Notes in Computer Science*, pp. 214–231. 2006. 60

[144] United Nations. “Universal Declaration of Human Rights: General Assembly Resolution 217 A (III) of 10”, December 1948. http://www.un.org/Overview/rights.html. 103

[145] Roger M. Needham and Michael D. Schroeder. “Using Encryption for Authentication in Large Networks of Computers”. *Communications of the ACM*, **21**(12):993–999, December 1978. 35

[146] B.C. Neuman and T. Ts’o. “Kerberos: An authentication service for computer networks”. *IEEE Communications*, **32**(9):33–38, September 1994. 42

[147] L. H. Nguyen and A. W. Roscoe. “Efficient group authentication protocols based on human interaction”. In “Proceedings of Foundation of Computer Security and Automated Reasoning for Security Protocol Analysis (FCS-ARSPA)”, pp. 9–33. Seattle, August 2006. 88

[148] OP3. “ShotCode”. http://www.shotcode.com/. 81

[149] Sarvar Patel. “Number Theoretic Attacks on Secure Password Schemes”. In “Proceedings of IEEE Symposium on Security and Privacy 1997”, pp. 236–247. IEEE Computer Society, 1997.

[150] A. Paulraj, Richard H. Roy and Thomas Kailath. “Estimation of signal parameters via rotational invariance techniques - ESPRIT”. In “Proceedings of 19th Asilomar Conference on Circuits, Systems and Computers”, pp. 83–89. Pacific Grove, California, Nov 1985. 131

[151] Olivier Pereira and Jean-Jacques Quisquater. “Generic Insecurity of Cliques-Type Authenticated Group Key Agreement Protocols”. *Proceedings of 17th IEEE Computer Security Foundations Workshop (CSFW)*, Jun 2004. 87, 88, 91

[152] Olivier Pereira and Jean-Jacques Quisquater. “Some attacks upon authenticated group key agreement protocols”. *Journal of Computer Security*, **11**(4):555–580, Jan 2004. 87, 88, 91

[153] Andreas Pfitzmann and Marit Hansen. “Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology”, May 2006. http://dud.inf.tu-dresden.de/Anon_Terminology.shtml. Version 0.28. 105

[154] Andreas Pfitzmann and Marit Kohntopp. “Anonymity, Unobservability and Pseudonymity – A Proposal for Terminology”. In H. Federath (ed.), “Designing Privacy Enhancing Technologies, Proc. Int’l Workshop Design Issues in Anonymity and Observability”, vol. 2009 of *Lecture Notes in Computer Science*, pp. 1–9. Springer-Verlag, 2001. 105

[155] Raphael C.-W. Phan and B. M. Goi. “Cryptanalaysis of Two Provably Secure Cross-Realm C2C-PAKE Protocols”. In “Proceedings of Indocrypt 2006”, vol. 4329 of *Lecture Notes in Computer Science*, pp. 104–117. Springer-Verlag, December 2006. 43

[156] Bhaskar D. Rao and K. V. S. Hari. “Performance analysis of ESPRIT and TAM in determining the direction of arrival of plane waves in noise”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37**(7):1990–1995, July 1989. 131

[157] Michael K. Reiter and Aviel D. Rubin. “Crowds: anonymity for web transactions”. *ACM Transactions Information Systems Security*, **1**(1):66–92, 1998. 140

[158] Kaarle Ritvanen and Kaisa Nyberg. “Upgrade of Bluetooth Encryption and Key Replay Attack”. *9th Nordic Workshop on Secure-IT Systems*, Nov 2004. 28

[159] Michael Roe, Bruce Christianson and David Wheeler. “Secure Sessions from Weak Secrets”. Tech. rep., University of Cambridge and University of Hertfordshire, 1998. 20

[160] Michael Rohs. “Real-World Interaction with Camera-Phones”. *2nd International Symposium on Ubiquitous Computing Systems*, Nov 2004. 65

[161] Michael Rohs and Beat Gfeller. “Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects”. *Advances in Pervasive Computing*, pp. 265–271, Apr 2004. 65

[162] R. Roy, A. Paulraj and T. Kailath. “ESPRIT - A subspace rotation approach to estimation of parameters of cissoids in noise”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **34**:1340–1342, 1986 1986. 131

[163] Richard Roy and Thomas Kailath. “ESPRIT - Estimation of Signal Parameters Via Rotational Invariance Techniques”. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37**(7):984–995, July 1989. 131

[164] Peter Ryan, Steve Schneider, Michael Goldsmith, Gavin Lowe and Bill Roscoe. *Modelling and Analysis of Security Protocols*. Addison-Wesley, 2001.

[165] T. Scott Saponas, Jonathan Lester, Carl Hartung, Sameer Agarwal and Tadayoshi Kohno. “Devices That Tell On You: Privacy Trends in Consumer Ubiquitous Computing”. In “Proceedings of 16th Usenix Security Symposium”, 2007. 119

[166] Mahadev Satyanarayanan. “A Catalyst for Mobile and Ubiquitous Computing”. *IEEE Pervasive Computing*, **1**(1):2–5, January 2002. 13

[167] Ralph Schmidt. “Multiple emitter location and signal parameter estimation”. *IEEE Transactions on Antennas and Propagation*, **34**:276–290, March 1986. 131

[168] Bruce Schneier. *Applied Cryptography, 2nd edition, Protocols, Algorithm, and Source Code in C*. Wiley, 2nd ed., 1996.

[169] David Scott, Richard Sharp, Anil Madhavapeddy and Eben Upton. “Using Visual Tags to Bypass Bluetooth Device Discovery”. *Mobile Computing and Communications Review*, 2005. 65

[170] Michael Scott. “Computing the Tate pairing”. In Alfred Menezes (ed.), “Proceedings of the RSA Conference: Topics in Cryptology - the Cryptographers’ Track (CT-RSA 2005)”, vol. 3376 of *Lecture Notes in Computer Science*, pp. 293–304. Springer-Verlag, 2005. 56

[171] Andrei Serjantov and George Danezis. “Towards an Information Theoretic Metric for Anonymity”. In Paul Syverson and Roger Dingledine (eds.), “Proceedings of Workshop on Privacy Enhancing Technologies (PET)”, vol. 2482 of *Lecture Notes in Computer Science*. 2002. 140

[172] Yaniv Shaked and Avishai Wool. “Cracking the Bluetooth PIN”. In “Proceedings of 3rd USENIX/ACM Conf. Mobile Systems, Applications, and Services (MobiSys)”, Seattle, WA, June 2005. 28, 29

[173] Adi Shamir. “Identity-based cryptosystems and signature schemes”. In G. R. Blakley and D. Chaum (eds.), “Advances in Cryptology - Proceedings of CRYPTO ’84”, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53. Springer-Verlag, 1985. 41, 44

[174] Claude Elwood Shannon. “A Mathematical Theory of Communication”. *Bell System Technical Journal*, **27**:379–423, July 1948. 140

[175] Victor Shoup. “Lower Bounds for Discrete Logarithms and Related Problems”. In “Proceedings of Advances in Cryptology - EUROCRYPT 97”, vol. 1233 of *Lecture Notes in Computer Science*. 1997. 31

[176] Dave Singelée and Bart Preneel. “Location Privacy in Wireless Personal Area Networks”. In “Proceedings of the 5th ACM Workshop on Wireless Security (WiSe 2006)”, pp. 11–18. Los Angeles, California, USA, 2006. 103

[177] D. K. Smetters and Glenn Durfee. “Domain-based administration of identity-based cryptosystems for secure email and IPSEC”. In G. R. Blakley and D. Chaum (eds.), “Proceedings of 12th USENIX Security Symposium”, pp. 215–229. August 2003. 47

[178] Frank Stajano. *Security for Ubiquitous Computing*. John Wiley and Sons Ltd., 2002. 77

[179] Frank Stajano and Ross Anderson. “The Cocaine Auction Protocol — On The Power Of Anonymous Broadcast”. In “Proceedings of 3rd International Workshop on Information Hiding”, vol. 1768 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999.

[180] Frank Stajano and Ross Anderson. “The Resurrecting Duckling — Security issues for Ad-Hoc Wireless Networks”. In Bruce Christianson, Bruno Crispo and Michael Roe (eds.), “Proceedings of 7th International Workshop on Security Protocols”, vol. 1796 of *Lecture Notes in Computer Science*. Springer-Verlag, 1999. 21, 61, 77

[181] Michael Steiner, Gene Tsudik and Michael Waidner. “Diffie-Hellman key distribution extended to group communication”. *Proceedings of the 3rd ACM conference on Computer and communications security*, pp. 31–37, March 1996. 88

[182] Michael Steiner, Gene Tsudik and Michael Waidner. “Cliques: A New Approach to Group Key Agreement”. *Proceedings of IEEE International Conference on Distributed Computing Systems*, May 1998. 88, 89

[183] Michael Steiner, Gene Tsudik and Michael Waidner. “Key Agreement in Dynamic Peer Groups”. *IEEE Transactions on Parallel and Distributed Systems*, **11**(8):769–780, 2000. 89

[184] Douglas R. Stinson. *Cryptography Theory and Practice*. Chapman & Hall / CRC, 3rd ed., 2006.

[185] Charles W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall, 1992. 131

[186] Eleanor Toye, Anil Madhavapeddy, Richard Sharp, David Scott and Alan Blackwell. “Using camera-phones to interact with context-aware mobile services”. Tech. Rep. UCAM-CL-TR-609, University of Cambridge, December 2004. 65

[187] Ersin Uzun, Kristiina Karvonen and N. Asokan. “Usability Analysis of Secure Pairing Methods”. In “Proceedings of Usable Security (USEC ’07)”, February 2007. 61

[188] Jukka Valkonen, N. Asokan and Kaisa Nyberg. “Ad Hoc Security Associations for Groups”. In “Proceedings of ESAS 2006”, vol. 4357 of *Lecture Notes in Computer Science*. Hamburg, Germany, September 20-21 2006. 88, 98

[189] Serge Vaudenay. "On Bluetooth Repairing: Key Agreement Based on Symmetric-Key Cryptography". In Dengguo Feng, Dongdai Lin and Moti Yung (eds.), "Proceedings of First SKLOIS Conference on Information Security and Cryptology (CISC)", vol. 3822 of *Lecture Notes in Computer Science*, pp. 1–9. Springer, 2005. 28, 29

[190] Serge Vaudenay. "Secure Communications over Insecure Channels Based on Short Authenticated Strings". In "Advances in Cryptology - CRYPTO 2005", vol. 3621 of *Lecture Notes in Computer Science*, pp. 309–326. Springer-Verlag, 2005. 60, 68, 71, 79

[191] Mario Čagalj, Srdjan Čapkun and Jean-Pierre Hubaux. "Key Agreement in Peer-to-Peer Wireless Networks". *Proceedings of the IEEE (Special Issue on Security and Cryptography)*, **94**(2):467–478, February 2006. 60, 68

[192] Srdjan Čapkun, Jean-Pierre Hubaux and Markus Jakobsson. "Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks". *EPFL-IC Technical report IC/2004/10*, Jan 2004. 102

[193] Xiaoyun Wang, Yiqun Lisa Yin and Hongbo Yu. "Finding Collisions in the Full SHA-1". In "Proceedings of Crypto 2005", 2005. `http://www.infosec.sdu.edu.cn/paper/sha1-crypto-auth-new-2-yao.pdf`. 36

[194] Roy Want, Andy Hopper, Veronica Falcao and Jon Gibbons. "The Active Badge Location System". *ACM Transactions on Information Systems*, **10**(1):91–102, January 1992. 112

[195] Samuel Warren and Louis Brandeis. "The Right to Privacy". *Harvard Law Review*, **IV**(5):193–200, December 1890.

[196] Mark Weiser. "The Computer for the Twenty-First Century". *Scientific American*, **263**(3):94–104, September 1991. 13

[197] Mark Weiser. "Some Computer Science Issues in Ubiquitous Computing". *Communications of the ACM*, **36**(7):75–84, 1993. 59

[198] Ollie Whitehouse. "RedFang", 2003. `http://www.atstake.com/`. 111

[199] Ford Long Wong and Hoon Wei Lim. "Identity-Based and Inter-Domain Password Authenticated Key Exchange for Lightweight Clients". In "Proceedings of 3rd IEEE International Symposium on Security in Networks and Distributed Systems", Niagara Falls, Canada, May 2007. 16, 17, 41

[200] Ford Long Wong, Min Lin, Shishir Nagaraja, Ian Wassell and Frank Stajano. "Evaluation Framework of Location Privacy of Wireless Mobile Systems with Arbitrary Beam Pattern". In "Pro-

ceedings of 5th Conference on Communications Networks and Services Research", pp. 157–165. IEEE Press, Fredericton, New Brunswick, Canada, May 2007. 17, 121

[201] Ford-Long Wong and Frank Stajano. "Location Privacy in Bluetooth". In Refik Molva, Gene Tsudik and Dirk Westhoff (eds.), "Proceedings of 2nd European Workshop on Security and Privacy in Ad hoc and Sensor Networks", vol. 3813 of *Lecture Notes in Computer Science*, pp. 176–188. Springer-Verlag, Visegrad, Hungary, July 2005. 16, 17, 102, 114, 117, 122

[202] Ford-Long Wong and Frank Stajano. "Multi-channel Protocols". In "Proceedings of the 13th International Workshop in Security Protocols", Cambridge, UK, Apr 2005. 16, 17, 73, 88, 95

[203] Ford-Long Wong and Frank Stajano. "Multi-channel Protocols for Group Key Agreement in Arbitrary Topologies". In "Proceedings of 3rd IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2006)", Pisa, Italy, March 2006. 17, 87, 92

[204] Ford Long Wong and Frank Stajano. "Multichannel Security Protocols". *IEEE Pervasive Computing, Special Issue on Security & Privacy*, **6**(4), Oct-Dec 2007. To appear. 17, 80, 87, 96

[205] Ford-Long Wong, Frank Stajano and Jolyon Clulow. "Repairing the Bluetooth Pairing Protocol". In "Proceedings of the 13th International Workshop in Security Protocols", Cambridge, UK, Apr 2005. 16, 17, 27, 28, 29, 106, 113

[206] Simon Woodside. "Semacode", 2004. `http://semacode.org/`. 81

[207] Thomas Wu. "The Secure Remote Password Protocol". *Proceedings of 1998 Internet Society Symposium on Network and Distributed System Security*, pp. 97–111, 1998. 20, 31, 35

[208] Xiao Liang Xu and K. M. Buckley. "Bias analysis of the MUSIC location estimator". *IEEE Transactions on Signal Processing*, **40**:2559–2569, 1992. 131

[209] Jianxin Yan, Alan Blackwell, Ross Anderson and Alasdair Grant. "The memorability and security of passwords – some empirical results". Tech. Rep. UCAM-CL-TR-500, University of Cambridge, September 2000. 21

[210] Her-Tyan Yeh and Hung-Min Sun. "Password authenticated key exchange protocols among diverse network domains". *Computers and Electrical Engineering*, **31**(3):175–189, 2005. 42, 53, 54

[211] Moustafa A. Youssef, Ashok Agrawala and A. Udaya Shankar. "WLAN Location Determination via Clustering and Probability Distributions". In "Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PerCom)", Dallas-Fort Worth, USA, March 2003.

[212] L. Zhu and B. Tung. “Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)”, June 2006. http://www.ietf.org/rfc/rfc4556.txt. IETF RFC 4556. 42, 53