The Global Trust Register 1998

Ross J. Anderson

Bruno Crispo

Jong-Hyeon Lee

Charalampos Manifavas

Václav Matyáš Jr.

Fabien A.P. Petitcolas

Publisher: Northgate Consultants Ltd., 10 Water End, Wrestlingworth,
Sandy, Bedfordshire SG19 2HA, United Kingdom

The Global Trust Register for 1998

Editors:

Ross J. Anderson
Bruno Crispo
Jong-Hyeon Lee
Charalampos Manifavas
Václav Matyáš Jr.
Fabien A.P. Petitcolas

ISBN: 0-9532397-0-5
1. Computer Security 2. Cryptography

In no event will the publisher be liable for any consequential, incidental or indirect damages (including damages for loss of business profits, business interruption, loss of business information and the like) arising out of the use or inability to use the information in this book.

We accept no liability at all for any errors in the entries in this book. It is a fact of publishing life that typos occur, as do mistakes in compilation and bugs in software. We believe that we have taken reasonable care, but we have not had the time or other resources to take extreme precautions against errors and attacks.

The purpose of this book is not due diligence but risk reduction. We aim to help a prudent person to check the validity of public key certificates that are presented to her electronically; we do not aim to provide the only such means, still less to provide protection against the very large number of things that can go wrong with computer systems, or be utilised maliciously by an attacker.

Where it is necessary to be able to place legal reliance on a digital signature, the reader should obtain confirmation directly from the counterparty of the value or fingerprint of the public key or certificate, as well as of any contractual conditions, restrictions, jurisdiction and other conditions.

Cover illustration: Jirí Matyáš

This book contains 99 numbered pages.

Printed 1998-02-09.

Table of Contents

Foreword

The growth and commercial development of the Internet will depend on solving the big open question of trust. A significant part of this problem depends on having some way of identifying and authenticating users — essentially a robust way of answering the question '*To whom am I speaking?*'

The probable solution to this problem involves public key cryptography. This technology lets each user generate two related keys: one key is kept private and used to create 'digital signatures' on messages, while the other is made public and used to verify them.

The question that now arises is how to find out which user is associated with which public key.

When public key cryptography came along in the 1970's, its inventors suggested that the names of computer users, and their public keys, should be published in a public directory — a kind of phone book. By the 1980's, the idea had shifted to certification authorities — bodies that would sign and issue electronic documents to the effect that 'the key with value X belongs to company Y'.

However people still have to get authentic copies of the public keys of the certification authorities themselves, and this is the principal problem which this book addresses.

The hope is sometimes expressed that all the world's numerous certifiers would sign each others' keys. However, as they are often competitors, this is unlikely to happen soon; and even if they did, it would be limited value as they often use incompatible certification policies and mechanisms.

Meantime, we believe that the kind of trust that an old-fashioned paper directory can provide will be a useful complement to the trust that can be placed in electronic certification mechanisms.

1 Introduction

This book is a register of the fingerprints of the world's most important public keys; it implements a top-level certification authority (CA) using paper and ink rather than in an electronic system.

Its purpose is fourfold:

1. to provide the currently missing top level in the global key certification hierarchy, thus enabling users to verify the authenticity of root certificates that they acquire online;
2. to ground the trust required for electronic commerce and other online applications in the trust that has been built up over the years in the world of print publishing, and thereby to build confidence in electronic trust mechanisms;
3. to broaden our understanding of the scientific, engineering and business issues associated with top-level certification, and in the process to discover as many bugs and other problems as possible in existing public key standards and their implementations;
4. to use the privilege of a print publisher — to print and distribute anything that is not pornographic, seditious or a breach of copyright — to forestall plans by some governments to license CAs and to impose oppressive licensing conditions, such as the escrow of private confidentiality keys.

We will now explore each of these purposes in turn.

1.1 Global CA

When Whit Diffie and Marty Hellman invented public key cryptography in 1975, their vision was that people would look up public keys in the phone book. But a paper listing of billions of people is large and including keys would make it worse, so the phone companies elaborated the idea into a standard for a distributed electronic database (X.500). Although the key certificate part of this (X.509) is becoming widely used, the central database was never built. Now it probably will not be. The business model of a single, state-owned phone company in each country has been superseded; there are now many competing phone companies, and many more firms providing Internet and other data services. People use a wide variety of mechanisms, from specialist directories to web search engines, to find other people; and the X.400 email mechanism that underlies X.500 has been defeated in the marketplace by SMTP.

This has left X.509 adrift. The original concept had called for a global top-level CA, perhaps administered by the United Nations, which would sign the keys of national CAs; these in turn would certify the keys of lower level CAs, and below this there would be both corporate key hierarchies and encryption services provided to private individuals.

An alternative concept, promoted by companies like Microsoft and Netscape, cuts governments out of the loop and focusses instead on industry sectors. The top-level key in each application is provided by the software vendor, with its public component embedded in the application. Thus, for example, Microsoft's top level ('root') key would certify that VISA was a 'brand bindery', VISA in turn would certify its member banks, and they would then issue certificates to their customers.

This second model is being implemented in the specific context of the SET protocol for credit card transactions over the Internet, but it is turning out to be less than generally applicable. The reason is that the software vendors are unable or unwilling to certify the top level keys of all the organisations that wish to become or to establish CAs. Thus one finds a small number of X.509 CAs whose root certificates are shipped with products such as Netscape Communicator and Microsoft Internet Explorer; but a large number of CAs find themselves excluded. This is especially a problem for CAs outside the USA.

It is also a problem for professional practice. Lawyers, doctors and other professionals tend to be organised either locally or nationally (and in the latter case often by speciality), with no effective supranational organisations that could take over the registration function of a 'brand bindery'. We are beginning to find numbers of small CAs springing up to certify keys in specialised networks; we include a number of them in this book, ranging from an EU project to replace bills of lading throughout Europe to a scheme to share radiology images between five London hospitals.

One systematic solution that has been suggested is that CAs should cross-certify each others' keys. However, the larger CAs are commercial competitors and so do not want to promote each others' business, while the smaller CAs have little reason to interact; for example, a London radiology CA has no obvious motive to establish relations with a CA serving patent attorneys in New Zealand. In addition, the certification policies of these different entities are often incompatible. So cross-certification appears to be of limited value, at least for the time being.

An alternative approach to the hierarchical model of CAs is given by Pretty Good Privacy (PGP), in which users certify each others' keys in a 'web of trust'. The problem here is that the web of trust is very patchy and uneven; while there are well-connected components (especially of computer security professionals) there are groups of PGP users who form isolated components and many individuals who are not connected at all. As with small CAs, there is little reason to expect that different specialities will be connected (indeed, small CAs are often implemented using PGP technology rather than X.509). Furthermore, it is not clear what trust can be placed on a chain of introductions, given that trust is not transitive.

The upshot of this historical legacy is that there is no cheap and effective way for Internet users to check the validity of public keys on which they may wish to rely. A US user, for example, cannot easily check the validity of a certificate from a server in the Czech Republic, if this is issued by a Czech CA whose root certificate is not included in the user's browser. She may just accept the certificate into her registry but it might have been issued by anyone!

We are trying to solve this problem by making available in this book the fingerprints or other information by which the root certificates and keys of the X.509 and EDI CAs known to us can be verified, as well as a number of the more important PGP keys used in the web of trust and elsewhere. When downloading such a certificate or key, you can have your browser or PGP software compute the relevant fingerprint and compare it with the value in this book. This should enable you to get a higher level of assurance of the key's authenticity that would otherwise be the case.

1.2 Grounding Trust in a Paper Book

One reason to favour a printed book for the global root CA is that the security issues for a book are much clearer and more tractable than for an electronic service based on a root certificate embedded in common browsers.

By far the simplest attack on a book is to manufacture a forged copy of it with one or more critical entries changed, and supply it surreptitiously to the target of the attack. Users can defend themselves against such an attack by purchasing their copy of the book either directly from the publishers (see details below) or from a randomly chosen bookstore. With particularly critical trust decisions, they may double-check against another copy of the book held in a local library, or against a copy on the shelf of a bookstore.

More pernicious attacks on a book include hacking into the editors' computer system, or the publishers' computer system, and introducing errors either to cause a false key to be accepted, or to undermine public confidence in the whole operation. We have taken reasonable precautions to prevent such attacks, although we cannot exclude them entirely; and it is in the nature of things that typos and other errors will occur. The nature of the checks we have carried out, the residual risks, our disclaimers, and the way in which trust in this product will be developed over time, are all set out below.

However, trust issues are straightforward in book publishing compared with the problems of running an electronic CA, and especially one at the root of a global system. An enormous variety of attacks may be mounted on computer systems — ranging from high-tech attacks such as cryptanalysis and Tempest through the exploitation of chance vulnerabilities in operating system and network software through to subversion of personnel and 'legislative' attacks such as requiring that CAs hand over keys to the government on demand.

In consequence, an online CA that were to serve mutually distrusting governments would be exceptionally difficult to construct; indeed we really have no idea how to do this technically. The repeated attempts to secure government access to CA and other keys (some of which involve dubious deals between equipment manufacturers, software vendors and governments) suggest that the problem of creating global trust in systems is probably a political impossibility as well.

By contrast, trust in a printed directory is relatively unproblematic because of the very long period over which trade directories, telephone directories, public registers of doctors and lawyers, and books of bankers' specimen signatures have been used in everyday life.

There are also legal considerations. A number of jurisdictions from Germany to Utah have adopted digital signature laws, while lawyers in some other jurisdictions aver that digital signatures are already valid under laws which define the essence of a signature as intent. Whatever the relative merits of these new laws and reinterpretations of old laws, the point is that they differ. So if a global top-level CA were to be instantiated in an electronic system, there would be enormous scope for confusion; it might be acceptable in some jurisdictions but not others, or have different force in different places. Thus, for a long time, there will be value in having a public register by which root CA certificates may be authenticated, and which escapes much of this legal morass by virtue of being implemented in ink and paper rather than software and hardware.

From the point of view of resilience, the fact that systems fail (and in particular the chaos forecast with the year 2000 date rollover problem) makes it prudent to have fallback paper mechanisms for critical procedures. Thus, even if a global electronic CA were to exist, something like this book would also have to be created as an emergency backup.

Finally, there are much less tangible cultural aspects to trust. Even among computer scientists (and security experts) there are still many people who trust paper documents over electronic ones; and in the lay population, who are exposed almost daily to scare stories about the insecurity of the Internet, paper is far more trusted. It is our hope that the publication of this book can start the process of transferring trust from the world of paper to cyberspace.

1.3 Scientific and Engineering Aspects

The exercise of collecting a large number of public keys highlighted several problems in currently implemented trust services, including both X.509 and PGP.

1.3.1 X.509

The most common software used to handle certificates nowadays is the web browser. There are several problems related to the common browser implementations and we give here an overview of the most important shortcomings.

Lack of transparency: when users download a copy of a browser, they get numerous different certificates that are supposed to be trusted and which are already embedded in the software. When they get a chain of certificates from somewhere on the net and need to verify it in order to authenticate a server (for example), the browser will perform the verification and will report at the end of the check if the verification has been successful or not. From the security point of view, this is not enough since users need to know also which, among all the CAs in the browser, they had to trust in the process. Users might not trust all the hard coded CAs, or they might associate different levels of trust with them.

Lack of information: the common browsers do not display all the information used for computing the fingerprint. The fingerprint of the X.509 certificates listed in sections 2 and 3 is the MD5 hash of the DER encoding of the whole certificate. With some browsers, it is not possible to find an official copy of the hard coded root certificates, whether in the browser itself or on the relevant CA's web site. It is easy to find a fingerprint of the certificate, but requires considerable skill and effort for users to verify its correctness with their own implementation of the hash algorithm. This forces users to place a quite unnecessary level of trust in the browser, and increases vulnerability to attacks in which malicious code or other technical tricks are used to corrupt the browser's operation. (Our web page provides the missing information.)

Lack of standards compliance: browser manufacturers claim to display X.509 certificates but omit some mandatory fields. Microsoft Internet Explorer 4.0 does not include the serial number, despite this being mandatory and used to implement certificate revocation lists. Netscape Communicator 4.03 does not provide the right validity period: the hour and minute are missing while in the standard they are mandatory. Neither browser specifies the version or signature algorithm fields, and neither displays all the extension fields present, but only those fields defined by their vendors. This is a serious problem as extension fields are often used for access control decisions, and this misfeature prevents their being displayed and examined when the certificate is checked. This could allow unauthorised access to systems, or lead to owners of certificates being granted rights they have never asked for and thus assuming unwanted liability.

Dependency on browsers: some CAs have hard-coded different certificates for the same key and the same subject, but with different expiration dates, in different browsers. This practice could lead to some dangerous and dubious situations. What happens if a user successfully verifies a public key with one browser, and at the same time the same public key appears expired with another browser? Which copy of the certificate or of the browser should users believe and why? Having different certificates for the same subject name can cause confusion, because the fingerprints displayed by the browser for those two certificates are different while the key is the same.

This problem arises from the overlap of two different concepts: a certificate and a public key. In X.509, which all browsers claim to implement, the cryptographic credential used to identify a principal is the whole certificate rather than just the public key. Another disadvantage of having different certificates in different browsers is that users may need to request a different certificate from the same CA depending on their choice of browser. An unnecessary dependency between certificates and software is thus created.

Aggregation of trust at key generation: Using the browser to generate the key pair and submit a certification request to a CA may help in rapid deployment of the technology but is less than ideal. Suppose the CA's policy is that when it receives a certificate request it will have a staff member who knows the customer call her by telephone and verify the fingerprint of the public key (this might be the reasonable procedure where the CA is a professional practice such as a doctor or lawyer). However, the current browser implementations of key generation do not let the user see either the newly generated public key itself or its fingerprint. The fingerprint is also useful in other protocols that can be used to prevent middleperson attacks.

One solution to these problems is to have separate software for key generation. There is no engineering reason to create a strict dependency between the generation of users' certificates and the software they use it with. On the contrary, the general robustness of protection systems would be increased if users could choose from a range of vendors of key generation software, certificate management software, email encryption software, digital signature software and browser software. Such unbundling would tend to localise failures and to force interfaces to be transparent and well-documented.

Problem with root key revocation: some CAs have decided to hard code their root certificate in user software, even where the lifetimes of certificates and software are different. This can cause software to self-destruct at a given time in the future, a situation not unlike the 'millennium bug' of year 2000 rollover. If software persists past its design lifetime, or if the private part of the root key is compromised, then it becomes necessary to change the root public key in all the copies of the software. This could be a difficult and complex task, especially if the software is no longer maintained. Introducing such vulnerabilities without a very good reason is foolish, and 'planned obsolescence' for marketing purposes is not a good reason. Systems often persist for decades past their design lifetime, a good example being all the 1960's COBOL that still forms part of mission critical systems and whose maintenance is now a serious problem for many organisations.

Implementation defects: browsers, like other software, may contain bugs. For example, Netscape Communicator v 4.03 reports the expiry date of the MCI CA certificates as the 17th July 1998, rather than the 16th July 1998 as appears in these certificates. However the Belsign Class 1–3, which also expire on the 16th July, appear to be handled correctly.

1.3.2 PGP

For PGP keys, unlike X.509 certificates, there are already several sites that keep a public copy of the global `pubring.pgp`, so called a PGP key server.

Although these services are useful, they have lots of security holes. For example, one finds keys with names such as '`president@whitehouse.gov`' which bear no relation to the individual that one would naturally associate with that name. In fact, PGP key servers simply collect all keys sent to them, and verifying their authenticity is up to the user.

Users are supposed to verify keys either by means of a non-electronic channel (such as by meeting the keyholder in person and exchanging key fingerprints) or by means of a chain of introductions, in each of which one user signs another's key in order to certify his identity. These chains of introductions make up the so called 'web of trust'. However this web is at best patchy, and although it is easy to verify the keys of some people via multiple independent chains, there are many others for whom such authentication is problematic or impossible. By including a number of important PGP keys in this book, we hope to improve this situation somewhat.

The criteria we used to decide which PGP keys to include, and the various levels of care with which keys have been authenticated, are described below. Meanwhile, we will observe a technical weakness of PGP: that key fingerprints are computed on the concatenation of the RSA modulus and exponent. Thus the key with modulus `2E27...A1E733` and exponent `11` has the same fingerprint as the key with modulus `2E27...A1E7` and exponent `3311`. For this reason, a PGP key is not uniquely specified by its fingerprint, but by the combination of fingerprint and keylength.

Finally, there are a number of common problems with X.509 and PGP, whose scope we are only beginning to understand. One concerns the lack of global naming conventions. As an example, the AT&T X.509 root certificate has the distinguished name 'Certificate Services'. This is clearly not unique (neither is it an isolated example). We follow the Netscape interpretation and list this as 'AT&T Certificate Services'; one must however beware that MS Explorer lists it under the name 'ATT Certificate Services.' On the PGP side, the most common cause of failure was changing e-mail addresses.

A more serious problem with all the certification mechanisms is the widespread lack of operational robustness. Organisations trying to authenticate their keys to us have made just about every conceivable mistake: they have sent us the wrong keys, sworn to the wrong fingerprints, produced wrongly dated certificates, and have been unable to generate certificates with requested contents. Often the mechanisms just cannot be made to work at all. At very few sites have we found staff to be in control of their systems in the way that one expects, for example, in an office issuing passports or drivers' licences. Much of this clumsiness is no doubt due to the relative newness of the technology; but a very much higher level of robustness is required in implementations if the promise of public key technology is to be realised.

1.4 Politics

Some countries, such as the UK, have proposed government licensing of encryption services; a certification authority would be permitted to trade only if it escrowed its users' private confidentiality keys and made them available to the authorities on demand. Such a crippled certification authority is commonly known as a 'trusted third party' (the name chimes well with the NSA definition of trust, namely that a trusted system or component is one with the power to break one's security policy). The European Union, on the other hand, opposes national trusted third party regulations that would impede either the trade in encryption products and services, or electronic commerce in general; and various other international bodies have emphasised the need for electronic trust services that can be relied on.

Given the determination of some national intelligence agencies to obtain access to cryptographic keys used in industry and commerce, this debate is unlikely to be settled in the immediate future. One of the contributions of this book is to present an implementation of a certification authority that, on the one hand, is a paper book and thus protected by the constitutions and bills of rights of various nations, while being on the other hand an 'encryption service' in terms of the previous UK government's proposed legislation and thus liable to licensing. We undertake that under no circumstances whatsoever will we apply for such a licence; should this book be banned in the UK by future escrow legislation, then the entire publication process will be shifted to the USA or elsewhere.

We would point out that basing commercial and professional trust on printed directories has a long history. Examples that come to mind include the Medical Register, the yellow pages, the many specialised trade directories, and the books of managers' signatures that banks print and send to other banks with which they do business. A government that banned our trust register should logically ban these books as well.

1.5 How We Chose and Checked the Keys

We have included the X.509 and EDI keys of all certification authorities known to us who have made their public keys available, whether by giving them to us, by publishing them or otherwise, and all PGP public keys which have been used to sign a significant number of other keys or which are used in certain defined roles (as top-level certifier keys, by webmasters, software distributors or computer emergency response teams).

Where possible we have carried out independent checks on the authenticity of each key, and the keys are marked from **D** up to **A** depending on the level of verification that we were able to carry out.

The definition of these levels of trust is as follows:

D means that we have no reason to believe that the principal who owns the key is other than as stated. We also have no particularly strong reason to believe the principal is as stated, so the level of assurance given by this level is vestigial. (Its main function is that, if the key remains unchallenged for some time, then it might acquire a slightly higher rating in future editions of this book.)

C means that the key has been certified by someone whose key we rate at B, or (for the majority of PGP keys in the book) that we verified the binding between the key and the email address by sending email, encrypted under the key, to the address and getting a signed response. In the case of X.509 certificates, we sent a cleartext email to an address found on the web site whence we downloaded the certificate, and asked for the postal address and/or the CA root public key signed by the CA root private key or by a key directly certified by the CA root key. Both of these methods of verification can of course be overcome by someone with the ability to forge email; but a similar procedure is used by commercial CAs for low-assurance consumer certificates. In our case, we sent out a large batch of encrypted emails with no prior publicity, so the likelihood of attacks involving a temporary takeover of a user's email address is low, and the level of assurance attained is reasonable (though not watertight).

B means that our knowledge of the binding between the key and the listed name rests on an introduction by someone we consider trustworthy and competent; if this introduction was effected using public key cryptography, then it used a key we rate at level A. (We apply a general rule that introduction cuts the trust level by one, so a key certified by a key at level B will have level C). We also set at level B those keys whose owners have authenticated themselves thoroughly by methods relying on formal government certification (such as by presenting passports, certificates of company incorporation etc) and by use of multiple conventional authentication mechanisms (such as when we telephone a switchboard number found in a telephone book, ask for a responsible person by role rather than name, and then confirm the key fingerprint by a protocol involving registered mail).

A means that at least one of us* has definite personal knowledge that the key belongs to the person or entity listed. Examples are the keys of colleagues and of well known companies whose root keys we have been able to verify by strong means (e.g., we dug out the key from the company's software and had it independently certified by at least one appropriately senior employee who has been known to us for a long time). Thus grade A keys are certified by more than formal processes; the position of the key owner in the social structure has been verified.

The arrangement of the keys into chapters has of necessity been rather ad hoc and depended on the numbers of keys of various kinds which we have received. In the medium to long term, we expect to see separate chapters for general CA services and for various business sectors such as banking, insurance, healthcare, education and so on; for the time being, there are only a handful of online players in each sector, and so we felt it not worthwhile to have many chapters with only two or three entries in them.

It would have been possible to have a separate chapter for universities but we decided against this for the time being as a significant number of university CAs (especially in the X.509 world) have been funded as joint ventures with industry. We have included a short separate chapter of medical keys, both as an experi-

* By 'us' in this context, we mean the six editors of the 'Global Trust Register' and two contributing editors: Johann Bezuidenhoudt and Markus G. Kuhn. We recruited Johann for his knowledge of South African CAs and Markus for his knowledge of German CAs.

ment and to support various pilots and projects in medical telematics. We trust that the number and size of such chapters will increase rapidly.

The largest single collection of keys we have is PGP keys. We thought it useful to partition these into three separate chapters — one for PGP keys used as institutional CA keys (including computer emergency response team keys), one for personal keys, and one for anonymous remailers (as the nature of trust here is somewhat different).

1.6 Disclaimer

We accept no liability at all for any errors in the entries in this book — including omissions and attacks. We do not believe that we have managed to publish the first book free of typos, or that the software we have written is the first program to be free of bugs. We believe that we have taken reasonable care, with all entries checked by a second individual; however we have not had the time or other resources to take extreme precautions against common-mode errors or attacks (for example, on our shared file system). We have also been limited by the data and resources available to us, and in particular by the responses from people and companies contacted. We believe that this situation will improve with time.

The purpose of this book is not due diligence but risk reduction. We aim to help a prudent person check the validity of certificates that are presented to him electronically; we do not aim to provide the only such means, still less to provide protection against the very large number of things that can go wrong with computer systems, or be utilised maliciously by an attacker.

Note that those real-world certificates which can be shown to an unlimited number of people, such as passports, birth certificates, driving licences and university degrees, are generally issued by organisations that will not accept liability for errors or forgeries. On the other hand, certificates which convey financial value (such as credit cards) are restricted in their use. A cardholder who emailed his credit card number to a hundred thousand online merchants in the space of an hour would incur the displeasure of his card issuer!

The kind of trust service provided by this book is solidly in the former category, and should be compared to printed registers of the members of restricted professions such as doctors or lawyers. Although such registers have mistakes in some entries, they still perform a useful service. Our book should be used in this light; failure to find an expected entry, or the discovery that an entry has an unexpected value, does not constitute proof of wrongdoing but simply indicates that closer attention is called for.

It should also be noticed that the great majority of commercial CAs have lengthy disclaimers and policy statements to the effect that they will not accept any liability either. Thus even if an entry in this book is correct and verifies that a certificate was issued by a given CA, and that CA certifies a key as belonging to a certain merchant, no particularly strong conclusion can be drawn. Hopefully this may change over time.

Where it is necessary to be able to place legal reliance on a digital signature, this commonly entails application specific mechanisms such as those in the Bolero system for bills of lading and in the proposed SET system for credit card transactions on the net. It is prudent for users to make themselves familiar with the terms and conditions under which each particular application operates; it is these terms on which users must rely when something goes wrong.

Finally, we want to emphasise that this first edition of the Global Trust Register is still experimental and should be considered a 'beta' release. Compiling it has enabled us to understand more thoroughly the engineering problems with various public key mechanisms, to develop and debug the software used to manage and format our key database, and perhaps most importantly to understand the logistics and procedural issues involved in running a global certification service.

1.7 Next Version

We have an outline agreement with MIT Press that they will publish the next (1999) edition of this book, which will be available at the end of 1998. We expect that the book will be greatly expanded and that the mechanisms for assuring trust in keys will be refined. The deadline for entries is the first of July 1998.

Errata will be made available from time to time on:

http://www.cl.cam.ac.uk/Research/Security/Trust-Register/

and at other mirrors to be arranged; they will be signed by three or more of us except in an emergency.

1.8 How to Buy a Copy

This book is distributed free with 'Computer and Communications Security Reviews', for which a subscription form can be found on:

http://www.cl.cam.ac.uk/~rja14/#SR

It can also be ordered direct from the publishers for £15.00 including airmail postage. We can accept email credit card orders, but some card issuers insist that your card number and expiry date be encrypted. You can use PGP; a key with fingerprint

```
E5 C7 93 BE : 37 9D 28 42 : : 49 DC A8 09 : A1 47 05 F6        1024
```

can be fetched from:

http://www.cl.cam.ac.uk/~rja14/

You can also fax your order to us on +44 1223 334678, or mail it to us at Northgate Consultants Ltd., 10 Water End, Wrestlingworth, Bedfordshire SG19 2HA, England.

Ross Anderson
Ross.Anderson@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~rja14

```
E5 C7 93 BE : 37 9D 28 42 : : 49 DC A8 09 : A1 47 05 F6      1024
AF 5E F0 DF : 70 E3 E6 5B : : 66 D8 86 48 : 62 A1 E9 0A      2048
```

Bruno Crispo
Bruno.Crispo@cl.cam.ac.uk
crispo@di.unito.it
http://www.cl.cam.ac.uk/~bc201

```
1A D1 4A 1C : EE DD 32 D9 : : EC 5F 0C 93 : 17 C1 C6 5D      1024
```

Jong-Hyeon Lee
Jong-Hyeon.Lee@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~jhl21

```
51 7B 32 A5 : 08 FA 8D B2 : : 3F 7B DA 8C : 73 17 95 15      1024
43 A2 6D 53 : 15 7C 88 1E : : E2 C2 A8 4B : D5 DE 78 D0      2048
```

Charalampos Manifavas
Charalampos.Manifavas@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~cm213

```
D4 BF D5 6C : 86 83 EC 37 : : 88 B1 8E 92 : 9D 9A 97 FA      1024
```

Václav Matyáš Jr.
Vaclav.Matyas@cl.cam.ac.uk
matyas@informatics.muni.cz
http://www.cl.cam.ac.uk/~vm206

```
C3 CB 4E FD : 9E DE F0 CA : : 65 D5 BF 7C : 46 8F 7B 5C       1024
B9 B3 F9 AA : E5 1B 38 0E : : F9 DF 00 21 : CA F6 87 57       2048
```

Fabien Petitcolas
Fabien.Petitcolas@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~fapp2

```
27 09 EB 53 : 98 99 B2 8B : : CA DB AC 31 : CC FD 2B 1B       1024
24 61 2F 47 : DB 04 27 DD : : 8E 6F 76 60 : 5A 24 20 45       2048
```