The GCHQ Protocol and its Problems

Ross Anderson, Michael Roe

Cambridge University Computer Laboratory
Pembroke Street, Cambridge CB2 3QG
Email: (rja14,mrr)@cl.cam.ac.uk

Abstract. The UK government is fielding an architecture for secure electronic mail based on the NSA's Message Security Protocol, with a key escrow scheme inspired by Diffie-Hellman. Attempts have been made to have this protocol adopted by other governments and in various domestic applications. The declared policy goal is to entrench commercial key escrow while simultaneously creating a large enough market that software houses will support the protocol as a standard feature rather than charging extra for it.

We describe this protocol and show that, like the 'Clipper' proposal of a few years ago, it has a number of problems. It provides the worst of both secret and public key systems, without delivering the advantages of either; it does not support nonrepudiation; and there are serious problems with the replacement of compromised keys, the protection of security labels, and the support of complex or dynamic administrative structures.

1 Introduction

Over the last two years, the British government's crypto policy has changed completely. Whereas in 1994 the Prime Minister assured the House of Commons that no further restrictions on encryption were envisaged, we now find the government proposing to introduce a licensing scheme for 'trusted third parties', and licenses will only be granted to operators that escrow their customers' confidentiality keys to the government's satisfaction [11, 21].

In March 1996, a document describing the cryptographic protocols to be used in government electronic mail systems was issued by CESG, the department of GCHQ concerned with the protection of government information. This document was initially shared with contractors involved in building government systems but was leaked to one of us in June. After we distributed copies at Crypto 96, a version was published by the government on the world wide web [4].

According to this document, policy goals include '*attempting to facilitate future inter-operability with commercial users, maximising the use of commercial technology in a controlled manner, while allowing access to keys for data recovery or law enforcement purposes if required*'[1].

[1] A UK official who chairs the EU's Senior Officials' Group — Information Security (SOGIS) has since admitted that 'law enforcement' in this context actually refers to national intelligence [10].

A document on encryption in the National Health Service, issued in April, had already recommended that medical traffic should be encrypted, and keys should be managed, using mechanisms compatible with the future 'National Public Key Infrastructure' [26]; part of the claimed advantages for the health service were that the same mechanisms would be used to protect electronically filed tax returns and applications from industry for government grants. Furthermore, attempts are being made to persuade other European countries to standardise on this protocol suite.

So the soundness and efficiency of the GCHQ protocol proposals could be extremely important. If an unsound protocol were to be adopted across Europe, then this could adversely affect not just the secrecy of national classified data, the safety and privacy of medical systems, and the confidentiality of tax returns and government grant applications. It could also affect a wide range of commercial systems too, and make Europe significantly more vulnerable to information warfare. If the protocols were sound but inefficient, then they might not be widely adopted; or if they were, the costs imposed on the economy could place European products and services at a competitive disadvantage.

In this paper, we present an initial analysis of the security and efficiency of the GCHQ protocol.

2 The GCHQ Protocol

The precursor of the government protocol was first published by Jefferies, Mitchell and Walker at a conference in July 1995 [13]. A flaw was pointed out there[2] and a revised version was published in the final proceedings of that conference; this version also appeared at the Public Key Infrastructure Invitational Workshop at MITRE, Virginia, USA, in September 1995 and at PKS '96 in Zürich on 1st October 1996 [14]. The final, government approved, version of the protocol fixes some minor problems and adds some new features.

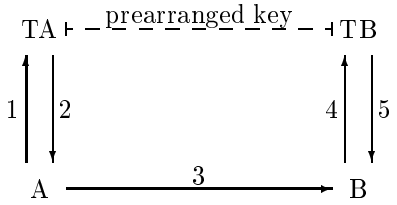
The document [4] is not complete in itself, as the protocol is presented as a series of extensions to the NSA's Message Security Protocol [18]. In the next section we will attempt for the first time to present the whole system in a complete and concise way, suitable for analysis by the cryptologic and computer security communities. We will then discuss some of its more obvious faults.

The GCHQ architecture assumes administrative domains 'corresponding approximately to individual departments', although there may be smaller domains where a department is scattered over a large geographical area. Each will have a 'Certificate Management Authority', under the control of the departmental security officer, which will be responsible for registering users and supplying them with keys. Key management will initially be under the control of GCHQ but might, in time, be devolved.

[2] The original protocol allowed the same base and modulus to be used in different domains and was thus vulnerable to Burmester's attack [2]

The basic idea is that if Alice wants to send email to Bob, she must go to her certificate management authority, whom we will call *TA*, and obtain from him secret information that enables her to calculate a key for communicating with Bob. She also receives a certificate of this secret information, and sends this to Bob along with the encrypted message. On receipt of the message Bob contacts his certificate management authority *TB* and obtains the secret information that he needs to decrypt the message. Thus two individuals can communicate only if both their departmental security officers decide to permit this.

The communication flow can be visualised in the following diagram:



We will now describe the content of these messages. The protocol is a derivative of Diffie Hellman [5] and the basic idea is that, in order to communicate with Bob, Alice must obtain a 'public receive key' for him from *TA* and operate on this with a 'secret send key' that *TA* also issues her, along with a certificate on the corresponding 'public send key'. At the other end, Bob will obtain a 'secret receive key' for her from *TB* and will use this to operate on her 'public send key' whose certificate he will check.

The secret receive keys are known to both users' authorities, and are calculated from their names using a shared secret master key. Each pair of domains *TX*, *TY* has a 'top level interoperability key', which we will call K_{TXY} for managing communication. The relevant key here is K_{TAB} which is shared between *TA* and *TB*. The mechanisms used to establish these keys are not described.

We will simplify the GCHQ notation by following [3] and writing $\{X\}_Y$ for the block X encrypted under the key Y using a conventional block cipher. Then the long term seed key that governs Bob's reception of traffic from all users in the domain of *TA* is:

$$\text{rseed}_{B,A} = \{B\}_{K_{TAB}} \tag{1}$$

A secret receive key of the day is then derived by using this seed key to encrypt a datestamp:

$$\mathrm{SRK}_{B,A,D} = \{D\}_{\mathrm{rseed}_{B,A}} \tag{2}$$

and Bob's public key of the day, for receiving messages from users in the domain *TA*, is

$$\mathrm{PRK}_{B,A,D} = g_A^{\mathrm{SRK}_{B,A,D}} \pmod{N_A} \tag{3}$$

where the 'base' g_A and the modulus N_A are those of *TA*'s domain (the document does not specify whether N_A should be prime or composite, or the properties that the group generated by g_A should possess).

Finally, *TA* certifies Bob's public key of the day as $S_{TA}(B, D, \mathrm{PRK}_{B,A,D})$.

Only receive keys are generated using secrets shared between authorities. Send keys are unilaterally generated by the sender's authority from an internal master key, which we will call K_{TA} for *TA*, and the user's name. Thus Alice's seed key for sending messages is $\mathrm{sseed}_A = \{A\}_{K_{TA}}$; her secret send key of the day is derived as $\mathrm{SSK}_{A,D} = \{D\}_{\mathrm{sseed}_A}$ and her public send key is $\mathrm{PSK}_{A,D} = g_A^{\mathrm{SSK}_{A,D}} \pmod{N_A}$. *TA* sends her the secret send key, plus a certificate $\mathit{Cert}(A, D, \mathrm{PSK}_{A,D})$ on her public send key. Send seed keys may be refreshed on demand.

Now Alice can finally generate a shared key of the day with Bob as

$$k_{A,B,D} = (\mathrm{PRK}_{B,A,D})^{\mathrm{SSK}_{A,D}} \pmod{N_A} \tag{4}$$

This key is not used directly to encipher data, but as a 'token key' to encipher a token containing a session key. Thus, when sending the same message to more than one person, it need only be encrypted once, and its session key can be sent in a number of tokens to its authorised recipients.

Anyway, Alice can now send Bob an encrypted version of the message M. According to the GCHQ protocol specification, certificates are sent with the object 'to simplify processing', so the packet that she sends to Bob (in message 3 of the diagram overleaf) is actually

$$\{M\}_{k_{sess}}, \{k_{sess}\}_{k_{A,B,D}}, \mathit{Cert}(B, D, \mathrm{PRK}_{B,A,D}), \mathit{Cert}(A, D, \mathrm{PSK}_{A,D}) \tag{5}$$

This protocol is rather complex. But what does it actually achieve?

2.1 Problem 1 — why not just use Kerberos?

The obvious question to ask about the GCHQ protocol is why public key techniques are used at all. After all, if TA and TB share a secret key, and Alice and Bob have to interact with them to obtain a session key, then one might just as well use the kind of protocol invented by Needham and Schroder [19] and since deployed in products like Kerberos [20]. Where Alice shares the key K_A with TA and Bob shares K_B with TB, a suitable protocol might look like

$$\begin{array}{lll} A & \rightarrow TA & : A, B \\ TA & \rightarrow A & : \{A, B, K_{AB}, d, \{K_{AB}, A, B, d\}_{K_{TAB}}\}_{K_A} \\ A & \rightarrow B & : A, \{K_{AB}, A, B, d\}_{K_{TAB}}, \{k\}_{K_{AB}}, \{M\}_k \\ B & \rightarrow TB & : A, B, \{K_{AB}, A, B, d\}_{K_{TAB}} \\ TB & \rightarrow B & : \{K_{AB}, d, A, B\}_{K_B} \end{array}$$

This protocol uses significantly less computing than the GCHQ offering, and no more messages. It can be implemented in cheap commercial off-the-shelf tokens such as smartcards, and with only minor modification of the widely available code for Kerberos. This would bring the further advantage that the implications of 'Kerberising' existing applications have been widely studied and are fairly well understood in a number of sectors (see, e.g. [12]). On the other hand, the integration of a completely new suite of authentication and encryption software would mean redoing this work. Given that the great majority of actual attacks on cryptosystems exploit blunders at the level of implementation detail [1], this will mean less secure systems.

The GCHQ response to this criticism is [15]:

> This is not so much an attack on the recommendations as an objection to the Trusted Third Party concept and the need for key recovery. The recommendations offer a realistic architectural solution to a complex problem and, as with any system, will require professional implementation.

This completely misses the point. Given that the UK government has decided (or been persuaded by the US government) to adopt key escrow in its own internal systems, exactly the same functionality could have been provided by a simple adaptation of Kerberos at much less cost and risk.

The only extra feature that appears to be provided by the GCHQ protocol is that users who receive mail from only a small number of other departments, and who operate under security rules that permit seed keys to persist for substantial periods of time, may save some communications with their TTPs by storing receive seed keys locally. This leads us to consider the issue of scalability.

2.2 Problem 2 — where are the keys administered?

How well the GCHQ protocol (or for that matter Kerberos) will scale will depend on how many key management authorities there are. With a large number of

them — say, one per business enterprise — the problem of inter-enterprise key management would dominate and the above protocol would have solved nothing.

The British government may be aware of this problem, as they propose to minimise the number of authorities. Under the legislation currently proposed, large companies would be permitted to manage their own keys — the rationale being that having significant assets they would be responsive to warrants — while small to medium enterprises and individuals would have to use the services of licensed TTPs — organisations such as banks that would undertake the dual role of certificate management authority and escrow agent.

We do not believe that this will work. One of us has experience of a bank with 25,000 employees, managed through seven regional personnel offices, trying to administer mainframe passwords at a central site. With thirty staff and much message passing to and from the regions, the task was just about feasible, but compelling a million small businesses to conduct a transaction with the 'Trusted Third Party' every time a staff member was hired, fired or moved, would do little for national economic competitiveness.

Medicine is another application to consider, as the issue of encryption and signature of medical records is the subject of debate in a number of European and other countries. There is relevant experience from New Zealand, where a proposal to have doctors' keys managed by officials in the local district hospitals turned out to be impractical. It is now proposed that keys there should be managed at the practice level [9]. In the UK, with some 12,000 general practices, hospitals and community care facilities, centralised key management is even less likely to be workable.

The GCHQ response to this criticism is [15]:

> It has also been suggested that a TTP network could become large and that some users would have to keep a large number of public keys. This problem is overcome in the Royal Holloway architecture since any user can obtain all the necessary key material from its local TTP. This is inherently more scalable than other approaches.

This again misses the point. If the UK health service, with 12,000 providers, has 12,000 TTPs, then the inter-TTP communications would be the bottleneck.

There is also the issue of trust. In the UK, the medical profession perceived the recommendation in [26] that key management should be centralised in a government body as an attempt to undermine the independence of the institutions currently responsible for professional registration — the General Medical Council (for doctors), the UK Central Council (for nurses), and so on. Retaining these organisations as top level CAs is essential for creating professional trust without which a security system would deliver little value.

But with the GCHQ protocol, this would appear to mean that a doctor who wished to send an encrypted email to a nurse working in the same practice would have to send a message to the GMC to get a key to encrypt the message, and

the nurse would have to contact the UKCC to get a key to decrypt it. This is clearly ludicrous.

In short, the GCHQ protocol may work for a strictly hierarchical organisation like government may be thought to be (though if that were the case, a Kerberos like system would almost certainly work better). But it is not flexible enough to accommodate real world applications such as small business and professional practice. This raises the question of whether it will even work in government. We suspect it would work at best badly — and impose a structural rigidity which could frustrate attempts to make government more efficient and accountable.

The GCHQ response to this criticism is [15]:

> The frameworks for confidentiality and authentication have been designed to cater for a wide range of environments. A hierarchy is defined only for the authentication framework and this is necessary because good security requires tight control.

This claim is inconsistent with the protocol document according to which *'As the Certificate Management Authority is responsible for generating the confidentiality keys, it should also take on the role of a certification authority in order to authenticate them'*. Thus the confidentiality and authentication hierarchies are clearly intended to be identical.

Rossnagel made the point that trust structures in the electronic world should mirror those in existing practice [23]; a point which all security engineers should consider carefully.

2.3 Problem 3 - should signing keys be escrowed?

The next problem is the plan to set up an escrowed trust structure of confidentiality keys first, and then bootstrap authentication keys from this [4] [26].

The GCHQ protocol defines a structure called a token to transfer private keys in an encrypted form. What is also required is a mechanism to convey public signature verification keys to the authority for certification, as well as a means to revoke signature keys (which should be independent of the 'key of the day' system that provides implicit revocation of encryption keys). Such mechanisms are not provided.

Similar considerations apply to MACs. The original US MSP has a mode of operation which provides confidentiality and integrity but not non-repudiation. In this mode, the message is not signed, and instead the confidentiality key (or a key derived from it) is used to generate a MAC on the message. As the GCHQ protocol is specified by citing the US MSP specification and explaining the differences, it would appear that this mode will also be a part of it; but when combined with the GCHQ key management, the effect is that an escrowed confidentiality key is used to authenticate the message.

Even if confidentiality keys are eventually required by law to be escrowed, the keys used for authentication must be treated differently, and there is a risk that programmers and managers responsible for implementing the GCHQ protocol might overlook this distinction and produce a flawed system. So it is worth explaining explicitly.

The stated purpose of key escrow is to enable law enforcement and other government employees to monitor the contents of encrypted traffic (and, in some escrow schemes, to facilitate data recovery if users lose or forget their keys). Its stated purpose does not include allowing government employees to create forged legal documents (such as contracts or purchase orders). It would be highly undesirable if people with access to the escrow system were able to use this access to forge other people's digital signatures. The scope for insider fraud and conspiracy to pervert the course of justice would be immense.

Any police officer will appreciate that if he can get copies of my bank statements, then perhaps he can use them in evidence against me; but if he can tracelessly forge my cheques, then there is no evidence at all any more. So if there is any possibility that a digital signature might be needed as evidence, then the private key used to create it must not be escrowed.

In fact, we would go further than this: keys which are used only for authentication (and not non-repudiation) should not be escrowed either. For example, suppose that some piece of equipment (e.g. a power station, or a telephone exchange) is controlled remotely, and digital signatures or MACs are used to authenticate the control messages. Even if these messages are not retained for the purposes of evidence, it is clearly important to distinguish between authorising a law enforcement officer to monitor what is going on and authorising him to operate the equipment. If authentication keys are escrowed, then the ability to monitor and the ability to create seemingly authentic control messages become inseparable: this is almost certainly a bad thing. Returning to the medical context, it is unlikely that either doctors or patients would be happy with a system that allowed the police to forge prescriptions, or the intelligence services to assume control of life support equipment. We doubt that a well informed minister would wish to expose himself and his officers in such a way.

In such applications, we need an infrastructure of signature keys that is as trustworthy as we can make it. Bootstrapping the trust structure from a system of escrowed confidentiality keys is unacceptable.

The GCHQ response to this criticism is [15]:

> This confuses the authentication and confidentiality frameworks. There is no intention to bootstrap signature keys required for non-repudiation purposes within the authentication framework.

The protocol document states (2.2.1) that *'to provide a non-repudiation service users would generate their own secret and public authentication key pairs, then pass the public part to a certification authority'*. But no mechanism for this is provided; in the rest of the document, it is assumed that all secret keys are

generated by the certification authority, and both the secret and public parts passed to the user. The conclusion we are compelled to draw is that the GCHQ protocol is not intended to provide a non-repudiation service at all.

Furthermore, both authentication and confidentiality key material is under the control of the Departmental Security Officer. Thus if there is a failure of security — and an embarrassing message is leaked — then it is always possible to claim that the message was forged, perhaps by the very security officer whose negligence permitted the leak in the first place.

Such 'plausible deniability' may be a government requirement, but if any non-governmental use of the GCHQ protocol is contemplated — or compelled by legislation — then signing keys should be managed by some other means (and not escrowed). We also recommend that if using the GCHQ protocol, normal policy should prohibit the sending of MAC-only messages; if a MAC-only message is received, the purported sender should be asked to resend a properly signed version (there are some special purpose uses in which the MAC-only mode is useful, but we won't describe them here).

2.4 Problem 4 — clear security labels

In the original NSA Message Security Protocol, the label describing the security classification of the contents of an encrypted message is also encrypted. The GCHQ version adds an extension which contains the label in clear (we will refer to this as the 'cleartext' security label, while the actual classification is the 'plaintext' security label).

There is a problem with doing this. An attacker can often derive valuable information from the cleartext label, taken together with the identity of the sender and recipient and the message volume. Indeed, with some labels, the attacker learns all she wants to know from the label itself, and cryptanalysis of the message body is unnecessary. This is why the US does not use cleartext security labels.

The GCHQ response to this criticism is [15]:

> CESG's modifications have been made after careful consideration of government requirements and in consultation with departments; they are sensible responses to these requirements.

We understand that these 'requirements' concern the national rules concerning the forms of protection which are deemed appropriate for various types of information.

Under the UK rules, it is possible for a combination of physical and cryptographic mechanism taken together to be deemed adequate, whereas either mechanism on its own is deemed inadequate. For example, a message classified SECRET can be enciphered with RAMBUTAN and then transmitted over a link

which lies entirely within the UK. The protection provided by RAMBUTAN is deemed insufficient if the same message is being transmitted across the Atlantic.

So British enciphered messages need to divided into two or more types: those that require various forms of additional physical protection, and those that don't. The message transfer system needs to be able to tell which messages are which, so it can use physically protected communications lines for some messages but not for others. The easiest way to achieve this is to mark the ciphertext with the classification of the plaintext.

However, if an opponent can get past the physical protection (which is often quite easy), then she can carry out the attacks described above. It would clearly be desirable for UK to follow the American lead and encrypt all security labels.

It may be argued that the rules are so entrenched that this is infeasible. A technical alternative is to reduce the cleartext security label to a single bit indicating only the handling requirements. In this way, routers have the information they need, and attackers get no more information than this (which they could arguably derive in any case by observing the route that the message takes). Using a completely incompatible (and information-losing) syntax for cleartext labels would also prevent lazy or careless implementers using them as plaintext labels — an error that can be predicted to occur under the GCHQ proposal.

This is particularly critical because the GCHQ protocol gives the cleartext label no protection at all — whether of confidentiality or integrity. If the cleartext label is ever used to determine the sensitivity of the decrypted plaintext, then the attacker could trick the recipient into believing that the message had a different classification, which might lead to its compromise.

2.5 Problem 5 — identity based keys

The GCHQ protocol gives users seed keys which they hash with timestamps to get user keys. But it is quite likely that some seed keys will be compromised (e.g. by Trojan horses previously inserted by attackers; via theft of computers; if smart cards holding them are lost etc). In that case, the user's certificates can be revoked, but the user cannot be issued with a new seed key, as it is a deterministic function of her name. All the CA can do is reissue the same (compromised) key.

To recover from this situation, either the user has to change her name, or the CA has to change the interoperability key and reissue new seed keys for every user in the domain. Both of these alternatives are unacceptable, and this is a serious flaw in the GCHQ protocol. It might be remedied by making the seed key also depend on an initial timestamp (which would also have to be added at several other places in the protocol).

2.6 Problem 6 — certificate formats

A related problem is the use of X.509 certificates, which have only a two digit date field. The threat posed to UK government and industry by the 'millenium

bug' is serious enough without introducing a further source of error. If the use of X.509 were unchangeable policy, then a fix that is consistent with the point above might be to include a four digit year as an extension field in the certificate. However, we would not consider this to be prudent engineering practice, because of the risk that implementers would get confused.

The UK government does not consider X.509 to be unchangeable, since (we understand) they are seeking to change it. The idea is to add an 'Alice-Bob bit' to distinguish the certificates for send and receive public keys. It is unclear whether the lack of this bit could give rise to practical attacks; it may be just to avoid the difficulty of parsing distinguished names when widely dispersed departments have multiple administrative domains. We hope that the date problem can also be fixed. In any case, a proper resolution of the problem of certificate structures may involve moving from X.509 to newer proposals such as SDSI [24]. This would enable developers to avoid the other drawbacks of X.509 (such as lack of support for dual control, for third party revocation services, for roles, and for signatures that must persist for a long period of time).

2.7 Problem 7 — scope of master key compromise

The compromise of the interoperability key between two domains would be catastrophic, as all traffic between users in those domains could now be read. In our experience, the likelihood of master key compromise is persistently underestimated. We know of cases in both the banking and satellite TV industries where organisations have had to reissue millions of customer cards as a result of a key compromise that they had considered impossible and for which they therefore had no disaster recovery plan. Introducing such a vulnerability on purpose is imprudent.

The GCHQ response to this criticism is [15]:

> CESG is fully aware of the need adequately to secure such high level exchanges and there are a number of ways this could be done.

Indeed, and comparison with other escrow systems such as Clipper shows that it is possible to provide some degree of protection against accidental disclosure and rogue insiders, by using two escrow agents in different departments rather than the single crypto custodians proposed by GCHQ. Clipper is not perfect in this regard, but it at least shows that it is possible to do better. At the very least, it would be prudent to change the interoperability keys frequently; this would remove the need for seed keys (and thus strengthen the argument for using Kerberos instead).

Of course, if corrupt law enforcement officers are allowed to abuse the system indefinitely, then no cryptographic or dual control protocol can put things right. Any discussion of insider attacks must assume that there exist procedures for dealing with misbehaving insiders, and indeed for detecting misbehaviour in the first place. This can be done with non-escrowed key management protocols (see for example [16]) but appears more difficult when escrow is a requirement.

2.8 Problem 8 — MOAC

The GCHQ protocol defines an extension which provides a "simple message origin authentication check". This is a digital signature computed on the contents of the message, and nothing else. By way of contrast, the original US MSP provided message origin authentication by computing a digital signature on a hash of the message and some additional control information. This additional control information can contain the data type of the message (e.g. whether it is an interpersonal text message or an EDI transaction).

The GCHQ proposal is an extension, rather than a replacement. That is, messages will contain two forms of digital signature: the old US form and the new UK form. As a result, this extension has not made the protocol simpler; it has made it more complex.

In nearly all circumstances, it would best to use the original US form of signature rather than the new UK one. The US form is very nearly as quick to compute, and it protects against some attacks to which the UK version is vulnerable. (It is possible for a bit string to have two different interpretations, depending on which data type the receiver believes it to be. The UK signature does not protect the content type, so an attacker could change this field and trick the receiver into misinterpreting the message.)

The only situation in which there might be a use for this UK extension is in implementing gateways between the GCHQ protocol and other security protocols. For example, it would be possible for a gateway to convert between Internet Privacy Enhanced Mail and UK MSP by replacing the PEM header with an MSP header and copying the PEM signature into the UK signature extension field. The important point to note about this is that such a gateway does not need access to any cryptographic keys, as it does not need to re-compute the signature. By way of contrast, a gateway between US MSP and PEM would need access to the sender's signature key: this is a very bad idea for obvious reasons.

2.9 Problem 9 — choice of encryption algorithm

GCHQ wants people to use an unpublished block cipher with 64 bit block and key size called Red Pike. According to a report on the algorithm prepared in an attempt to sell it to the Health Service, it is similar to RC5 [22] but with a different key schedule. It will apparently be the standard for government traffic marked up to 'Restricted', and it is claimed that systems containing it may be less subject to export controls: health service officials have claimed that US companies operating in the UK may be allowed by the US government to use Red Pike in products in which the use of DES would be discountenanced by the US State Department. US officials will not confirm this.

More significantly, Red Pike will shortly be fielded in mass market software, and will thus inevitably be reverse engineered and published, as RC2 and RC4 were. So it is hard to understand why the UK government refuses to publish it,

or why anyone should trust it, at least until it has been exposed to the attention of the cryptanalytic community for a number of years. If GCHQ scientists have found a weakness in RC5 and a fix for it — or even a change that speeds it up without weakening it — then surely the best way to gain acceptance for such an innovation would be to publish it.

The GCHQ response to this criticism is [15]:

> Another common misconception is that the CESG Red Pike algorithm is being recommended for use in the public arena. No confidentiality algorithm is mandated in the recommendations; for HMG use, however, approved algorithms will be required; Red Pike was designed for a broad range of HMG applications.

Vigorous efforts are still being made to promote the use of Red Pike in the health service, and as noted above, it is supposed to be used in a wide range of citizens' interactions with government such as filing tax returns and grant applications. Thus the accuracy of the above response is a matter of how one interprets the phrase 'public arena'.

3 Conclusion

The GCHQ protocol is very poorly engineered.

1. The key management scheme gives us all the disadvantages of public key crypto (high computational complexity, long key management messages, difficult to implement on cheap devices such as smartcards), and all the disadvantages of secret key crypto (single point of failure, little forward security, little evidential force, difficulty of 'plug and play' with shrink-wrapped software). It does not provide any of the advantages that one could get from either of these technologies; and its complexity is likely to lead to the subtle and unexpected implementation bugs which are the cause of most real world security failures.
2. It is designed for tightly hierarchical organisations, and cannot economically cope with the more complex trust structures in modern commerce, industry and professional practice. Its main effect in government may to perpetuate rigid hierarchies and frustrate the efficiency improvements that modern management techniques might make possible.
3. It goes about establishing trust in the wrong way. To plan to bootstrap signature keys from a 'national public key infrastructure' of escrowed confidentiality keys shows a cavalier disregard of the realities of evidence and of safety-critical systems.
4. There are a number of serious technical problems with the modifications that have been made to the US Message Security Protocol, which underlies the UK government's offering. Quite independently of the key management scheme and trust hierarchy that are eventually adopted, these modifications are unsound and should not be used.

The above four conclusions appeared in an earlier draft of this paper. The GCHQ response that that draft, which we have cited here, has not persuaded us to change a single word of their text.

We call on the cryptologic and computer security communities to subject this protocol to further study. If adopted as widely as the British government clearly hopes it to be, it would be a single point of failure of a large number of applications on which the security, health[3], privacy and economic wellbeing of Europe's citizens would come to depend.

Acknowledgement: We are grateful to the security group at Cambridge for discussions, and to Paul van Oorschot for pointing out that the second version of this protocol was presented at two other conferences as well as appearing in the Queensland conference proceedings [14].

References

1. RJ Anderson, "Why Cryptosystems Fail", in *Communications of the ACM* v 37 no 11 (Nov 94) pp 32–40
2. M Burmester, "On the Risk of Opening Distributed Keys", in *Advances in Cryptology — CRYPTO '94*, Springer LNCS v **839** pp 308–317
3. M Burrows, M Abadi, RM Needham, "A Logic of Authentication", in *Proceedings of the Royal Society of London A* v **426** (1989) pp 233–271
4. CESG, "Securing Electronic Mail within HMG — part 1: Infrastructure and Protocol" 21 March 1996, document T/3113TL/2776/11; available at URL `http://www.rdg.opengroup.org/public/tech/security/pki/casm/casm.htm`
5. W Diffie, ME Hellman, "New Directions in Cryptography", in *IEEE Transactions on Information Theory*, IT-22 no 6 (November 1976) p 644–654
6. *1996 EPIC Cryptography and Privacy Sourcebook*, Electronic Privacy Information Center, Washington, DC
7. *'Escrowed Encryption Standard'*, FIPS PUB 185, US Department of Commerce, February 1994
8. Y Frankel, M Yung, Escrow Encryption Systems Visited: Attacks, Analysis and Designs", in *Advances in Cryptology — CRYPTO 95*, Springer LNCS v 963 pp 222–235
9. P Gutman, *personal communication*, July 96
10. D Herson, in *interview with Kurt Westh Nielsen and Jérôme Thorel*, 25 September 1996; Ingeniøren/Engineering Weekly 10/04/1996; available at `http://www.ingenioeren.dk/redaktion/herson.htm`

[3] GCHQ has since claimed that the NHS proposals and its are 'similar but distinct' [15]. They are indeed similar, with many of the undesirable features described below being incorporated into NHS crypto pilots. The only respect in which they are clearly distinct is that the DH/DSA mechanisms were replaced by RSA after RSA was adopted as a European standard for healthcare. However, many of the undesirable features which we discuss above, such as the central generation of signature keys, have been retained in the NHS pilots.

11. N Hickson, Department of Trade and Industry, speaking at *'Information Security — Is IT Safe?'*, IEE, Savoy Place, London, 27th June 1996
12. "Kerberos on Wall Street", I Hollander, P Rajaram, C Tanno, in *Usenix Security 96* pp 105–112
13. N Jefferies, C Mitchell, M Walker, "A Proposed Architecture for Trusted Third Party Services", in proceedings of *Cryptography Policy and Algorithms Conference*, 3–5 July 1995, pp 67–81; published by Queensland University of Technology
14. N Jefferies, C Mitchell, M Walker, "A Proposed Architecture for Trusted Third Party Services", in *Cryptography: Policy and Algorithms*, Springer LNCS v 1029 pp 98–104; also appeared at the Public Key Infrastructure Invitational Workshop at MITRE, Virginia, USA, in September 1995 and PKS '96 in Zürich on 1st October 1996
15. ID Jones, *letter to R Anderson on behalf of GCHQ's Communications Electronics Security Group*
16. TMA Lomas, B Crispo, "A New Certification Scheme", in *Proceedings of the Fourth Cambridge Workshop on Cryptographic Protocols* (1996), Springer LNCS series pp 19-32
17. TMA Lomas, MR Roe, "Forging a Clipper Message", in *Communications of the ACM* v 37 no 12 (Dec 94) p 12
18. U.S. National Security Agency, *'Secure Data Network System : Message Security Protocol (MSP)'*, SDN.701, revision 4.0 (January 1996)
19. RM Needham, MD Schroder, "Using Encryption for Authentication in Large Networks of Computers", in *Communications of the ACM* vol 21 no 12 (Dec 78) pp 993–999
20. BC Neuman, T Ts'o, "Kerberos: An Authentication Service for Computer Networks", in *IEEE Communications Magazine* v 32 no 9 (Sep 94) pp 33–38
21. Press Association, "Move to Strengthen Information Security", 06/10 1808
22. RL Rivest, "The RC5 Encryption Algorithm", in *Fast Software Encryption* (1994), Springer LNCS v 1008 pp 86–96
23. Roßnagel A, "Institutionell-organisatorische Gestaltung informationstechnischer Sicherungsinfrostrukturen", in *Datenschutz und Datensicherung (5/95) pp 259–269*
24. "A Simple Distributed Security Infrastructure", RL Rivest, B Lampson, `http://theory.lcs.mit.edu/ rivest/publications.html`
25. B Schneier, *'Applied Cryptography – Protocols, Algorithms, and Source Code in C'* (second edition), John Wiley & Sons, New York, 1996
26. Zergo Ltd., '*The use of encryption and related services with the NHSnet*', published by the NHS Executive Information Management Group 12/4/96, reference number E5254; available from the Department of Health, PO Box 410, Wetherby LS23 7LN; Fax +44 1937 845381

This article was processed using the LaTeX macro package with LLNCS style