The GCHQ Protocol and its Problems

Authors' names removed

for refereeing

Abstract. The UK government is adopting an architecture for secure electronic mail that was designed by CESG, a department of GCHQ. It is based on the NSA's Message Security Protocol with a key escrow scheme adapted from a paper by Jefferies, Mitchell and Walker. The government is attempting to foist it on the commercial sector too, in the hope of not just entrenching commercial key escrow but also creating a enough large market that software developers will support the protocol as a standard feature rather than charging extra for it.

However, like the 'Clipper' proposal from the US government a few years ago, the architecture is poorly engineered. It provides the worst of both secret and public key systems, without delivering the advantages of either. It also has a number of other technical security problems, some of which we describe here.

1 Introduction

Cryptography was once concerned almost exclusively with the secrecy of military and diplomatic message traffic, and was the domain of specialist national agencies; but over the last 15–20 years, encryption technologies have been adopted in embedded commercial applications such as automatic teller machines, burglar alarms, pay-TV and prepayment utility meters. These applications now dwarf military cryptosystems by at least an order of magnitude, in both gross investment and in the number of fielded terminals [3].

There have so far been few civilian applications where cryptography is used to provide message confidentiality, but national intelligence agencies are worried that this will change, to the detriment of their eavesdropping capability. This concern led in the United States to the launch of the Clipper chip. Clipper was supposed to replace DES with a classified encryption algorithm that would be made available only in tamper-proof hardware, packaged with a protocol that would give law enforcement access: session keys would be transmitted along with the ciphertext, encrypted under a key of which government agencies would have a copy [14].

As the world now knows, Clipper was a failure. Not only did the chip arouse intense public hostility — dooming the AT&T secure telephone that was the first product to incorporate it — but technical flaws emerged too. The key escrow protocol was first broken by Blaze [7], while further attacks were found by Lomas and Roe [19] and by Frankel and Yung [15]. We also understand that the Clipper chip's tamper protection was broken by US chip makers — and presumably by foreign laboratories too.

Freedom of Information Act requests by US activists have dug out documents which show that the policy, which was initiated under the Bush administration, was aimed at ultimately outlawing all cryptography that the NSA could not break (the interested reader can consult [13] for details). What is less well publicised is that the US government has been working assiduously with the signals intelligence agencies of friendly countries to sell a successor to the Clipper chip to governments worldwide. This is known as CKE-TTP — 'Commercial Key Escrow by Trusted Third Parties" — or just TTP for short.

At the urging of the US intelligence community, the UK government has changed its policy on encryption completely over the last two years:

- in 1994, John Major stated in a parliamentary written reply to David Shaw MP that no further restrictions on encryption were envisaged in the UK;
- by early 1995, our defence industry contacts were telling of an escrow debate raging in Whitehall, in which the public was not invited to participate;
- by mid 1995, a proposed key escrow architecture developed by Royal Holloway appeared at an Australian conference [17], and a senior GCHQ man was on hand to support a proposed Australian escrow policy [23];
- following press reports of imminent European introduction of key escrow [25] [30], the EU's crypto policy body, SOGIS (the senior officials' group, information security) asked in November 1995 for a Council decision to spend 25 million ECUs to establish a European network of TTPs [29];
- by February 1996 the trade press was reporting that the US — together with France and Britain — was pushing the other countries in the OECD to adopt a common line on the use of TTPs for key escrow [11];
- in April 1996, a report on encryption in the National Health Service implied that key escrow would be a requirement: '*NHS should consider … whether it wishes to implement the key recovery capability within it or not*' [31]. It also stated that '*a number of necessary National Policy decisions were made by CESG in the last months of 1995'*.
- on the 10th June the government went public with an announcement that it would set up a licensing scheme for trusted third party services [24] [26];
- when this policy was presented at a public meeting on the 26th June, a government spokesman said that escrowing of confidentiality keys would be mandatory. He also said that an OECD expert group was working on 'global crypto guidelines', and made clear that the controls would focus on small-to medium enterprises and individuals rather than on large companies. Large companies would be exempt; the rationale given for this was that organisations with enough assets would be responsive to warrants.

So after months of careful news management, the UK government has changed course 180^0 from its former liberal crypto control regime.

In this paper, we look at the national key management system and associated electronic mail security architecture that the government is adopting for its own use and trying to sell to commerce and industry. According to [10], 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*'. So the policy will stand or fall according to whether the proposed system is perceived to bring any value to commercial users.

2 The Background to the GCHQ Design

The protocol that underlies the government's email system was first published at the '*Cryptography — Policy and Algorithms*' Conference in Brisbane in July 1995 [17]. It has been elaborated somewhat since, yet it is worthwhile to study the original protocol first.

2.1 The Royal Holloway scheme — 1995 version

The scheme was proposed as a means of facilitating key escrow over multiple domains, such as the nation states which make up the European Union. The basic idea is that when Alice and Bob communicate, they set up their session key in such a way that both Alice's government's trusted third party *TA*, and Bob's government's trusted third party *TB*, can calculate it.

The mechanics of the original proposal are a slight variant of Diffie-Hellman key exchange [12]: let p be a large prime number and let g generate a subgroup of Z_p^* in which the Diffie-Hellman problem is hard. Let S be a digital signature algorithm, and let $S_{TA}(X)$ denote the message X signed by *TA*.

Each user now receives from her government some private keys xA_i. There are separate send and receive keys, and there may be separate keypairs for each foreign government in the scheme. We will ignore this multiplicity for the time being, and consider that our user Alice has the private key xA and public key $yA = g^{xA} \pmod p$. Her government provides her with a certificate S_{TA}(Alice, date, yA) for the latter. Now when Alice wishes to communicate with Bob, they can form a shared key g^{xAxB}; they can use this as a key encrypting key to transmit a session key. Both their governments can calculate this value from information available to them.

2.2 An attack on the 1995 version

The problem with this version of the scheme is that other governments may also gain access, using a variant of a well known attack invented by Burmester [8]. The underlying flaw is that if multiple keys are derived from a single instance of the Diffie Hellman problem, and one of these keys is later revealed, then others may be compromised too.

Suppose a government other than TA and TB observes the conversation between Alice and Bob, and wishes to decrypt it. This government can generate a certificate S_{TC}(Chirac, date, yB) which falsely binds Bob's public key to another name, and initiate a conversation. Alice will generate the same key g^{xAxB} as she did with Bob, and if this key is now disclosed her previous session can be read.

Attacks of this kind are by no means fanciful. System designers often decouple bulk encryption from key management; the former may be done in custom hardware (such as in a satellite TV decoder) or in software, while the latter may be done in a smartcard that customises the system to individual subscribers by storing their long term secrets. Alice, in our scenario, might expose herself to a hostile country later by going there on business; its government's agents could then implant a virus in her laptop — or even beat her into decrypting her earlier message from Bob.

Once the attack has been spotted, there are various possible countermeasures. For example, one could make it mandatory to use different keys for communicating between different government domains. This is offered as an option in the 1995 Royal Holloway proposal; but implementers tend to avoid options that will cause extra complexity.

2.3 The Royal Holloway scheme — 1996 version

After Jefferies presented the above scheme at Brisbane, one of us (Anderson) pointed out the above attack to him. In the paper that appeared in the final conference proceedings, the attack is prevented by making it mandatory for each pair of communicating TTPs to use a shared secret key to diversify the private keys that they issue to their users. (It would have been nice if the final paper had an acknowledgement.)

There are still a number of problems. For example, the paper specifies that g should be a primitive root modulo p; this means that if TA generates xA, he could expose Alice's traffic by choosing xA to be the generator of a smooth order subgroup of Z_p^*. Similarly, if she generates xA, she could choose weak keys, perhaps to deliberately leak classified information to an outside collaborator.

However, the serious problems are of an engineering nature, and they are better discussed in the context of the concrete implementation proposed by GCHQ for securing British electronic mail.

3 The GCHQ Protocol

In the document '*Securing Electronic Mail within HMG — part 1: Infrastructure and Protocol*' [10], the Royal Holloway 1996 proposal is elaborated into a design for an actual system and presented as a series of extensions to the US Department of Defense's Message Security Protocol [20]. As cited above, this document states that the system's uptake in other sectors is to be encouraged, in order entrench key escrow and make it worth software supliers' while to support it in their products.

This is confirmed elsewhere. For example, the Zergo report on encryption in the National Health Service [31] states that the government is planning a national public key infrastructure based on Diffie-Hellman rather than RSA technology, and for this reason the NHS is urged to use the former rather than the latter for compatibility. In this application, the proposal is to introduce encryption first, and bootstrap integrity and non-repudiation mechanisms such as digital signatures from it afterwards.

The Zergo report also speaks of encouraging uptake of the technology by allowing small businesses to use it to file tax returns electronically, and government suppliers to submit bids with it. These proposed solutions will not just use the proposed key management technology, but also a block cipher called 'Red Pike', which we will mention later.

The system will be based on 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 can, in time, be devolved.

Each user or other entity will have private and public send and receive keys which are generated as follows. Firstly, each domain TX has a top level interoperability key, which we will call K_{TX}, for internal use, and a set of top level interoperability keys K_{TXY} for communicating with other domains.

As in the Royal Holloway protocol, Alice communicates with Bob by obtaining his public receive key from TA and exponentiating it with her own secret send key. TA generates his public key by deriving a 'seed key' from his name using the top level interoperability key K_{TAB}. We will simplify the notation of [10] and write:

$$KS_{TA,B} = \{B\}_{K_{TAB}} \tag{1}$$

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

$$xB_{TA,d} = \{d\}_{KS_{TA,B}} \tag{2}$$

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

$$yB_{TA,d} = g_A^{xB_{TA,d}} \pmod{N_A} \tag{3}$$

where the 'base' g_A and the modulus N_A are those of *TA*'s domain.

Finally, Bob's public key of the day is incorporated into a certificate S_{TA}(Bob, d, $yB_{TA,d}$).

Alice's send key is generated by *TA* using her name and its internal top level interoperability key K_A to produce a seed key $KS_{TA,A} = \{A\}_{K_A}$ from which her secret send key of the day is derived as $xA_{TA,d} = \{d\}_{KS_{TA,A}}$ and her public send key as $yA_{TA,d} = g^{xA_{TA,d}}$. *TA*sends her the secret send key, plus a certificate S_{TA}(Alice, d, $yA_{TA,d}$) on her public receive key.

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

$$k_{A,B,d} = (yB_{TA,d})^{xA_{TA,d}} \pmod{N_A} \tag{4}$$

Unlike in the Royal Holloway proposal, 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 is actually

$$\{M\}_{k_{sess}}, \{k_{sess}\}_{k_{A,B,d}}, S_{TA}(Bob, d, yB_{TA,d}), S_{TA}(Alice, d, yA_{TA,d}) \tag{5}$$

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

3.1 The first big problem with the GCHQ protocol

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 [21] and since deployed in products like Kerberos [22]. 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. 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.

3.2 The second big problem with the GCHQ protocol

Key management in medicine is a highly topical application to consider, given the debate on clinical confidentiality in general [5, 6] and on the security of a new network for Britain's National Health Service in particular [4]. The latter prompted the Zergo report and appears to have hastened the publication of the GCHQ protocol.

The Zergo report proposed that a single trusted third party should manage keys for the whole of the NHS, and went into detail about the budgets and timescales involved in setting it up (£250,000 to set up and £400,000 a year to run, with eight full time equivalent staff).

The clinical professions perceived this as an attempt to transfer control of professional registration to the NHS Executive from the current independent bodies — the General Medical Council, the General Dental Council, and so on. When confronted with these concerns, the Department of Health backed down and conceded that trust structures in the electronic world should mirror those in existing practice — a point already been made by Roßnagel [27]. The proposal was then made that the GCHQ protocol should be adopted but with the professional registration bodies acting as the TTP [32]. But this is unlikely to work.

The first problem is that the natural place to administer any kind of access right — whether a crypto key, logon password or old fashioned metal key — is where the rest of employee administration is done. In the health service, for example, that would mean at some 12,000 general practices, hospitals and community care facilities. Any attempt to centralise will be expensive, and will scale badly.

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 the solution could not possibly scale to a health service of a million people. Having a TTP in each of the approximately 200 health districts would not help: in New Zealand, a proposal to do just this and 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 locally [16].

The second problem is that trust in medicine is not a simple hierarchy. For example, professional registration bodies do not publish, or even possess, enough information to provide a directory service. The GMC knows and publishes doctors' names, registration numbers, dates of birth, degrees and other professional qualifications. Nurses are registered with the UK Central Council, which keeps its register secret and will not disclose members' names, let alone addresses, to third parties. They will merely confirm whether a given person is currently registered as a nurse, midwife or health visitor.

The problem is not just one of discretion; the registration bodies simply cannot keep track of their members' employment, for the reasons discussed above. Many nurses in particular work through agencies and might do shifts at a dozen different hospitals or clinics in the course of a month; having all their keys managed by the UKCC would entail a prohibitive volume of traffic and introduce many unnecessary failure modes.

The complexity is increased still further by the fact that many healthcare professionals have multiple roles in parallel as well as in series. The same person might simultaneously be a GP, a hospital doctor, a trainer, a trainee, a patient and a police doctor. The same person may have both parallel and serial employment complexity; it is not uncommon for someone with a part-time job as a hospital pharmacist to also work shifts as a locum for local chemists' shops. In these different roles, she sould use a range of mainframe, mini and PC systems.

So each clinical professional will generally need a different signature key for each role she undertakes. The requirements are discussed in more detail in [5]; the simplest workable key management scheme appears to be for each doctor to have a long term signing key, signed by the GMC, with which she would authenticate a signature key for each role. These role keys would also be signed by the CA for that role (general practice manager, hospital personnel officer, etc). Thus the compromise of a single CA would have relatively little effect: an attacker would have to compromise a doctor's long term key as well in order to sign a bogus role key.

The cost of this resilience appears to be the absence of the central point of failure that is the purpose of escrow protocols, but escrow is hardly relevant in medicine: clinical message traffic between doctors and hospitals is stored in the clear, at both ends, and kept for at least six years. Procedures for law enforcement access to medical records are well established and unproblematic.

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 a real world application such as medicine, and 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.

3.3 The third big problem with the GCHQ protocol

The third problem with the GCHQ scheme is the plan to set up an escrowed trust structure of confidentiality keys first, and then bootstrap signature keys from this in applications where this is required. This emerges from both [10] and [31], although both documents are slightly vague on the point.

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 in government departments might overlook this distinction and produce a flawed implementation. 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 in court of law (or elsewhere), then it is vital that the private key used to create that signature must not be escrowed. The possibility that the signature could have been forged by anyone with access to the escrow database undermines the value of the signature as evidence.

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 them 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. Returing to the medical context, for example, it is unlikely that either doctors or patients would be happy with a system that allowed the police or the intelligence services to forge prescriptions tracelessly; indeed, we doubt that a wise chief constable would expose his officers to the resulting temptations.

In all such applications, we need an infrastructure of signature keys that is as trustworthy as we can make it, and its design must be able to withstand scrutiny by hostile expert witnesses if it is to be relied on in evidence [2]. Bootstrapping the trust structure from a system of escrowed confidentiality keys, many aspects of whose design and implementation are likely to be obscured behind a cloud of ‘national security’, is unacceptable.

3.4 Further problems with the GCHQ protocol

A number of further problems arise with the British government’s proposal.

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 users’ seed keys will be compromised (e.g. if the machines holding them are stolen; 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 re-issue the same (compromised) key.

To recover from this situation, either the user has to change their name, or the CA has to change the interoperability key and re-issue new seed keys for every user in the domain. Both of these alternatives are unacceptable. This is a serious flaw in the GCHQ protocol. It could be remedied by making the seed key a function of the domain interoperability key, the user name and an initial time stamp. This fix would necessitate adding this new timestamp at several places in the protocol.

Certificate dates 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 is unchangeable policy, then a fix (that is consistent with the point above) might be to include an initial timestamp in each user’s distinguished name. However, we would not consider this to be prudent engineering practice!

Effects 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 and we know of cases where organisations have had to reissue millions of customer cards as a result of a key compromise that they had considered impossible and which they therefore had no plan to cope with. Introducing such a vulnerability on purpose is highly imprudent.

It might be argued that this is the price that has to paid for escrow; that is, that any system which permits law enforcement access is necessarily vulnerable to catastrophic failures caused by a single rogue individual. However, comparison with other escrow systems such as Clipper shows that this is not the case; it is possible to provide some degree of protection against insider attacks even in escrow systems. For example, Clipper uses two escrow agents so that a single escrow agent acting alone cannot compromise the system. Clipper is not perfect in this regard, but it at least shows that it is possible to do better. (Of course, if corrupt law enforcement officers are allowed to abuse the system indefinitely, then no cryptographic protocol can put things right. Any discussion of insider attacks must assume that there exist prcoedures for dealing with misbehaving insiders, and indeed for detecting misbehaviour in the first place.)

Clear security labels In the US MSP, 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. This is felt to be needed because of differences in 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.

As a result of this, in the UK 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 and unprotected lines for others. The easiest way to achieve this is to mark the ciphertext with the classification of the plaintext.

However, there is a problem with doing this. If an attacker can get past the physical protection (which is often quite easy to do), then they can derive valuable information from the clear text security label, taken together with the identity of the sender and recipient. Indeed, with some security 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 clear security labels, and it would clearly be desirable for UK to follow suit. However, the rules may be so entrenched that this is infeasible.

A technical alternative is to make the cleartext security label an indication purely of the the handling requirements. In this way, routers have the information they need in order to do routing, and attackers get no more information than this (it could be argued that attackers can derive the handling requirements of a message in any case by observing the route it takes). To make this work, implementations need to be aware that the encrypted security label may differ from the plaintext label, and they should label the decrypted plaintext based on the enciphered label not the plaintext one.

In the GCHQ protocol, the cleartext security label has no protection at all — whether of confidentiality or integrity. This gives rise to an attack 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 lower classification that it really had, which might lead to its compromise. If care is taken in implementations to distinguish the enciphered label from the plaintext one, this dangerous situation can be avoided.

Tokens The GCHQ protocol defines a structure called a token, which may be used to transfer private keys in an encrypted form. The token is defined in the section of the specification dealing with authentication. However, as was explained above, the private keys used for authentication should not be escrowed. If public key techniques are used, the private signature key should be generated by the user and there is no need to transport it between entities. There is thus no need for the token in the context of authentication. If message authentication codes are used, then the keys will indeed need to be transported between entities, but it is unclear how this can be achieved if all confidentiality keys are escrowed.

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 extension 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, very nearly as quick to compute, and it protects against some attacks which the UK version is vulnerable to. (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 mis-interpreting 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.

Weak message authentication The original US MSP has a mode of operation which provides confidentiality and integrity but not non-repudiation. In this mode, the message isn't 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 may also be a part of UK MSP. However, a new problem arises when this mode is used in conjunction with the GCHQ key management. When the two are combined, the effect is that an escrowed confidentiality key is used to authenticate the message. In other words, if this mode of operation is permitted, then persons with access to the escrowed confidentiality keys can forge messages.

For this reason, we recommend that the normal policy if using the GCHQ protocol should be to never send MAC-only messages, and to ask the purported sender to resend a properly signed version if a MAC-only message is received. However, there are some special purpose uses in which this mode is useful; we won't describe these here.

Choice of encryption algorithm GCHQ wants everyone to adopt an unpulished block cipher with 64 bit block and key size called Red Pike. This 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. It is even implied 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.

However, despite this fog of 'fear, uncertainty and doubt', it is clear that 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 best efforts of the cryptanalytic community for a number of years. The management of this algorithm's introduction is quite bizarre.

4 Why?

This whole affair is reminiscent of a banking project in the 1980's. A large clearing bank had got worried that it was falling behind the curve; it had a diversity of systems, some home grown and others from a number of different vendors, running on an assortment of hardware. Managers decided to replace all this with a single IBM mainframe; the retail systems would be replaced with an integrated software suite based on the leading US retail banking product. The internal justification for buying this product was that 'we just don't have the expertise to specify a next generation retail system ourselves'.

Software contracts were signed, mainframes were ordered, and a team was hired to do the implementation. A senior manager, with long experience of the branch network, was appointed as the system's 'user' with a team to help him specify the changes that would be needed to make it work in the different environment of another country. Some 30,000 changes were made; and eventually it was noticed that the new system was doing almost exactly the same as the old system did. To get from A to B (or rather from A to A) had cost a nine figure sum, the careers of many experienced IT managers and staff, and four years' chaos in which the bank's competitors improved their market share.

Something similar may have happened here. An organisation with long experience of a technology based on secret keys, aimed at secrecy rather than integrity and with key distribution organised in rigid hierarchies, is suddenly faced with a paradigm shift. Instead of encryption being a 'bump-in-the-wire' technology that was largely independent of broader system considerations, it now needs to be closely integrated with distributed computing systems. How could an organisation that traditionally hired mathematicians rather than computer scientists possibly cope?

Perhaps, as in the banking case, a bid was mounted to buy in the missing technology. A contractor was hired, a new system duly arrived and was customised according to the client's traditional view of the world. But the world had changed, and the world view had not. Doctrinal rigidity ensured that, once all the amendments had been made, the new system did just what the old one did, but at ten times the price.

5 Conclusion

The GCHQ protocol is very poorly engineered.

- Firstly, 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 bound to lead to the subtle and unexpected implementation bugs which are the cause of most real world security failures [1].

– Secondly, it is designed for tightly hierarchical organisations, and it 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.
– Thirdly, it goes about establishing trust in completely the wrong way. To plan to bootstrap signature keys from a 'national public key infrastructure' of escrowed confidentiality keys shows either a profound ignorance, or a cavalier disregard, of the realities of evidence and judicial process. It makes clear that the driving force behind the programme is the national intelligence interest rather than, as ministers and officials claim, law enforcement.
– Fourthly, there are a number of serious technical problems with the modifications that have been made to the US Message Security Protocol, which underlies GCHQ's effort. Quite independently of the key management scheme and trust hierarchy that are eventually adopted, these modifications are unsound and should not be used.

Indeed, it should be a matter of national concern that the organisation responsible for the security of Britain's classified defence and diplomatic traffic should have fielded such a substandard offering.

References

1. RJ Anderson, "Why Cryptosystems Fail", in *Communications of the ACM* v 37 no 11 (Nov 94) pp 32–40
2. RJ Anderson, "Liability and Computer Security: Nine Principles", in *Computer Security — ESORICS 94*, Springer LNCS v 875 pp 231–245
3. RJ Anderson, "Crypto in Europe — Markets, Law and Policy" in *Cryptography: Policy and Algorithms*, Springer LNCS v 1029 pp 75–89
4. RJ Anderson, "NHS wide networking and patient confidentiality", in *British Medical Journal* v 310 no 6996 (1 July 1996) pp 5–6
5. RJ Anderson, *'Security in Clinical Information Systems'*, published by the British Medical Association, January 1996; also available from `http://www.cl.cam.ac.uk/users/rja14/#Med`
6. RJ Anderson, "Clinical system security: interim guidelines", in *British Medical Journal* v 312 no 7023 (13 Jan 1996) pp 109–111
7. MA Blaze, "Protocol Failure in the Escrowed Encryption Standard", in *Proceedings of the Second ACM Conference on Computer and Communications Security* (2–4 November 1994) pp 59–67
8. M Burmester, "On the Risk of Opening Distributed Keys", in *Advances in Cryptology — CRYPTO '94*, Springer LNCS v 839 pp 308–317
9. UK ITSEC Scheme Certification Body, *'UK IT Security Evaluation and Certification Scheme'*, UKSP01 issue 2.0 (April 94)
10. "Securing Electronic Mail within HMG — part 1: Infrastructure and Protocol" 21 March 1996, document T/3113TL/2776/11
11. "United States pushes key escrow, security in OECD", in *Computer Fraud and Security Bulletin* (Mar 96) pp 7–9

12. W Diffie, ME Hellman, "New Directions in Cryptography", in *IEEE Transactions on Information Theory*, IT-22 no 6 (November 1976) p 644–654
13. *1996 EPIC Cryptography and Privacy Sourcebook*, Electronic Privacy Information Center, Washington, DC
14. *'Escrowed Encryption Standard'*, FIPS PUB 185, US Department of Commerce, February 1994
15. 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
16. P Gutman, *personal communication*, July 96
17. 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
18. 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
19. TMA Lomas, MR Roe, "Forging a Clipper Message", in *Communications of the ACM* v 37 no 12 (Dec 94) p 12
20. U.S. National Security Agency, *'Secure Data Network System : Message Security Protocol (MSP)'*, SDN.701, revision 4.0 (January 1996)
21. 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
22. 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
23. S Orlowski, "Encryption and the Global Information Infrastructure: An Australian Perspective", in proceedings of *Cryptography Policy and Algorithms Conference*, op. cit., pp 31–39
24. Press Association, "Move to Strengthen Information Security", 06/10 1808
25. D Peachey, "Euro-Clipper chip scheme proposed", in *Communications Week* no 151 (18 September 1995) pp 1, 81
26. Reuter, "UK to license information encryption services", RTf 06/10 1355, London, June 10th
27. Roßnagel A, "Institutionell-organisatorische Gestaltung informationstechnischer Sicherungsinfrostrukturen", in *Datenschutz und Datensicherung (5/95) pp 259–269*
28. B Schneier, *'Applied Cryptography – Protocols, Algorithms, and Source Code in C'* (second edition), John Wiley & Sons, New York, 1996
29. SOGIS, "Proposal for a Council Decision — in the field of security of information systems, concerning the establishment of a Europe-wide network of Trusted Third Party Services (ETS)", 23/11/95
30. J Thorel, "EC plans encryption rules in bid to police information superhighway", in *Nature* v 377 no 6547 (28/9/95) p 275
31. 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
32. Zergo Ltd., '*Zergo's response to "The Zergo Report — Initial Comments" by Dr Ross Anderson*', 17/5/96