Trusted third party based key management allowing warranted interception

Nigel Jefferies, Chris Mitchell, Michael Walker

Information Security Group
Royal Holloway, University of London

ABSTRACT

In this paper we propose a novel solution to the problem of managing cryptographic keys for end-to-end encryption, in a way that meets legal requirements for warranted interception. Also included are a discussion of what might constitute a reasonable set of requirements for international provision of such services, an analysis of the cryptographic properties of the scheme, and consideration of how it might operate in practice.

I. INTRODUCTION

There has been much recent discussion on the question of how to meet users' requirements for security services, such as confidentiality and authentication. This has been largely prompted by the US government's Clipper proposals [**1**], as well as the increasing use of electronic means for transferring commercially sensitive data. On the one hand, users want the ability to communicate securely with other users, wherever they may be, and on the other hand, governments have requirements to intercept traffic in order to combat crime and protect national security. Clearly, for any scheme to be acceptable on a wide basis, it must provide the service that users want, as well as meeting the legal requirements in the territories it serves.

To create a platform that can be used to provide user services, it is anticipated that solutions will be based on the use of trusted third parties (TTPs) from which users can obtain the necessary cryptographic keys with which to encrypt their data or make use of other security services. Law enforcement agencies' requirements will be focused on the need to obtain the relevant keys from a TTP within their jurisdiction, so that they can decrypt precisely those communications that they are authorised to intercept.

In this paper we propose a novel mechanism that will enable TTPs to perform the dual rôle of providing users with key management services and providing law enforcement agencies with warranted access to a particular user's communications. Unlike other proposals, the mechanism allows users to update their keys according

to their own internal security policies. Moreover, it provides a framework for Diffie-Hellman key establishment which obviates the need for directories.

We then list typical requirements for such a scheme, and consider how well the proposed mechanism meets these requirements. It is important to note that the scheme described here has been designed to establish keys for providing end-to-end confidentiality services, and not for integrity, origin authentication or non-repudiation services; the appropriateness of the mechanism for providing these services is a matter for further study. We conclude by considering possible variants of the basic method, and also how other proposed schemes for using TTPs in this way relate to the described method.

This paper was produced as part of the UK DTI/EPSRC-funded LINK PCP project 'Third-Generation Systems Security Studies'. Participants in this project are Vodafone Ltd, GPT Ltd and Royal Holloway, University of London.

II. THE MECHANISM

The proposed mechanism is based upon the Diffie-Hellman algorithm for key exchange [**2**]. In order to simplify our description, we consider the mechanism only in relation to one-way communication (such as e-mail). The adaptation of the scheme for two-way communication is very straightforward.

More specifically we present the mechanism in the context of a pair of users *A* and *B*, where *A* wishes to send *B* a confidential message and needs to be provided with a session key to protect it. We suppose that *A* and *B* have associated TTPs *TA* and *TB* respectively, where *TA* and *TB* are distinct.

A. Initial requirements

Prior to use of the mechanism, *TA* and *TB* need to agree a number of parameters, and exchange certain information.

- Every pair of TTPs whose users wish to communicate securely must agree between them values *g* and *p.* These values may be different for each pair of communicating TTPs, and must have the usual properties required for operation of the Diffie-Hellman key exchange mechanism, namely that *g* must be a primitive element modulo *p,* where *p* is a large integer (satisfying certain properties). These values will need to be passed to any client users of *TA* and *TB* who wish to communicate securely with a client of the other TTP.
- Every pair of TTPs whose users wish to communicate securely must agree on a digital signature algorithm. They must also each choose their own signature key/verification key pair, and exchange verification keys in a reliable way. Any user *B* wishing to receive a message from a user *A*, with associated TTP *TA*, must be equipped with a copy of *TA*'s verification key in a reliable way (typically this would done by their own TTP *TB*).
- Every pair of TTPs whose users wish to communicate securely must agree a secret key $K(TA,TB)$ and a Diffie-Hellman key generating function *f.* This function *f* shall take as input the shared secret key and the name of any user, and generate for that user a private integer *b* satisfying $1 < b < p-1$ (which will be

a ‘private receive key’ assigned to that user–see immediately below). The secret key $K(TA,TB)$ might itself be generated by a higher-level Diffie-Hellman exchange between the TTPs, or by any other bilaterally agreed method.

Given that *B* is to be provided with the means to receive a secure message from *A,* prior to use of the mechanism *A* and *B* need to be provided with certain cryptographic parameters by their respective TTPs.

- Using the function *f,* the secret key $K(TA,TB)$ and the name of *B,* both *TA* and *TB* generate the private integer *b* satisfying $1 < b < p-1$ (as described above). This key is known as *B*’s *private receive key*. The corresponding *public receive key* for *B* is set equal to $g^b \bmod p$. The private receive key *b* for *B* needs to be securely transferred from *TB* to *B* (like the other transfers discussed here, this can be performed ‘off-line’). Note that *B* will be able to derive its public receive key from *b* simply by computing $g^b \bmod p$. Note also that this key can be used by *B* to receive secure messages from any user associated with *TA*; however, a different key pair will need to be generated if secure messages need to be received from users associated with another TTP.
- *A* must be equipped with a *send key pair*, for use when sending confidential messages to users associated with TTP *TB* (in fact this key pair could be used with many, perhaps all, other TTPs, as long as they share the values *g* and *p*). *A* randomly chooses a *private send key*, denoted *a* (where $1 < a < p-1$). *A*’s *public send key* is then set equal to $g^a \bmod p$.

  A then passes its public send key to *TA*, by some reliable means (which does not need to preserve secrecy), and *TA* then signs a copy of *A*'s public send key concatenated with the name of *A* using its private signature key; this yields a certificate for *A*’s public send key. The signed certificate is then passed to *A*.
- *A* must also be equipped with a copy of *B*’s public receive key. *B*’s private receive key *b* can be computed by *TA* using *f*, the name of *B*, and the key $K(TA,TB)$. *TA* can then compute *B*’s public receive key as g^b, which can then be transferred in a reliable way from *TA* to *A*.

B. The mechanism itself

As we have seen, prior to use of the mechanism, *A* possesses the following information:

- *A*’s own private send key *a*;
- a certificate for *A*’s own public send key ($g^a \bmod p$), signed by *A*’s TTP *TA*;
- the public receive key ($g^b \bmod p$) for user *B*, and;
- the parameters *g* and *p*.

This information can be employed to generate a shared key $g^{ab} \bmod p$ for protecting the confidentiality of a message to be sent from *A* to *B*. This key can be used as a session key, or, even better, as a key-encryption key (KEK). The KEK would then be used to encrypt a suitable session key. This latter approach has a number of advantages. For example:

- it would facilitate the sending of email to multiple recipients, since the message can be encrypted once under a random session key, and this session key can then be distributed to each recipient by encrypting it using the KEK, and
- it allows the use of a new key for each message.

User *A* then sends the following information to user *B*:

- the message encrypted using the session key (either g^{ab} mod p or a key encrypted using g^{ab} mod p),
- *A*'s public send key (g^a mod p) signed by *TA*, and
- the public receive key (g^b mod p) for user *B*

Once received, the public receive key g^b mod p allows user B to find its corresponding private receive key *b* (there will be a different receive key for each TTP with whose users *B* communicates). User *B* can then generate the (secret) session key g^{ab} mod p by raising *A*'s public receive key (g^a mod p) to the power of *B*'s own private receive key *b,* and thus can decrypt the received message.

C. Warranted interception

Should there be a warrant for legal interception of this communication, an intercepting authority can retrieve the private receive key of the 'receiving user' from the trusted third party within its jurisdiction, and use this in conjunction with the public send key of the 'sending user' to find the session key for the encryption. There is no requirement for the intercepting authority to deal with any TTPs outside of its jurisdiction, or for any TTPs outside of its jurisdiction to know what is going on.

More specifically, suppose user *A* has sent a message to user *B* (served by TTP *TB*). If *TA* is required to provide access to this message, it can do so by combining

- *B*'s private receive key *b* (generated from *K*(*TA*,*TB*) and the name of *B* using the key generating function *f*), and
- *A*'s public send key (g^a mod p),

to generate the shared key g^{ab} mod p. The TTP *TB* can gain access to the shared key using exactly the same calculation.

Hence, in this scheme, any TTP will know the *private receive keys* for all its own users, and also all such keys used to help protect messages sent by its own users. No TTP need have access to any *private send keys*.

D. Properties of the mechanism

We next observe a few significant properties of the proposed mechanism.

- First note that a user can change his/her *send key pair* at any time. A user simply generates a new key pair for him/herself, and passes the public send key to his/her TTP for signing.
- No directories are required to make the system work. An entity wishing to send a message only needs to obtain the public receive key for the intended recipient from his/her own TTP, who can generate this information merely from the name

of the recipient and the identity of the recipient's TTP. A recipient of an enciphered message will, given the information contained in the message, possess all the data necessary to obtain the session key, without further reference to any third parties.

- Whilst, given the description above, receive key pairs are apparently fixed, by including the year (or month and year) within the scope of the key generating function *f*, all receive key pairs can automatically be updated at regular intervals.

III. A TYPICAL SET OF REQUIREMENTS ON A TRUSTED THIRD PARTY SCHEME

Clearly, the definition and agreement of a set of requirements acceptable across a broad set of countries is largely a political process. However, we can give a set of typical or likely requirements on which to base an analysis of the suitability of the proposed mechanism.

1. *Use of the scheme should provide visible benefits for the user*. The design and operation of the scheme means that the TTPs are capable of offering their services to users on a commercial basis. By signing up to a licensed TTP, the user will be able to communicate securely with every user of every TTP with whom his TTP has an agreement. The user would potentially be able to choose from a number of TTPs in his home country, thus increasing his trust in the TTP.
2. *The scheme should allow national and international operation.* The proposed scheme achieves this by ensuring that the intercepting authority can obtain the required keys from a TTP within its jurisdiction.
3. *Details of the scheme should be public.* This is achieved for the proposed scheme by the publication of this paper!
4. *The scheme should be based on well known techniques*, and Diffie-Hellman certainly qualifies.
5. *All forms of electronic communication should be supported*. The proposed scheme can easily be adapted to include two-way communication such as voice telephony.
6. *The scheme should be compatible with laws and regulations on interception, as well as on the use, export and sale of cryptographic mechanisms*. This matter is the subject of further study, but no problems have yet been identified.
7. *Access must be provided to the subject's incoming and outgoing communication, where a warrant is held*. This is clearly achieved for the proposed scheme, as the subject's TTP will be able to provide the appropriate session keys.
8. *The scheme should support a variety of encryption algorithms, in hardware and software*. As the proposed scheme deals solely with key management, any suitable encryption algorithm can be used, as long as it is available to the recipient and legitimate interceptors. The best way to achieve this may be to use a standard list of algorithms, such as the ISO register.

9. *An entity with a warrant should not be able to fabricate false evidence.* This is particularly applicable in countries where intercepted communications are admissible as evidence in court. The proposed scheme as it stands does not meet this requirement, but the provision of digital signatures as an additional service by the TTP will allow it to be met.

10. *Where possible, users should be able to update keys according to their own internal policies*. The proposed scheme allows a user to generate new send key pairs as often as wished (provided that he deposits them with his TTP) or have them generated by his TTP. The receive keys, which are generated deterministically based on the TTPs' shared key and the user's identity, are more permanent, and change only if the TTPs' shared key or the user's identity changes. However, as we have already noted, if there is a requirement for receive keys to be changed at regular intervals, a date stamp could be included within the scope of the key generating function *f*. This would have the advantage that any private receive key provided to an intercepting authority would have only a limited period of validity, meaning that the warranted interception capability could only last for a certain time period before needing to be renewed.

11. *Abuse by either side should be detectable by the other*. We believe that this is the case for the proposed scheme, although abuse by collusion between the two sides may still be possible. The main disincentive to such abuse may be the 'shrink-wrapped' provision of the software, which we would expect to be bundled in with, say, an email system or other telecommunications software.

12. *Users should not have to communicate with TTPs other than their own.* The only communication required in the proposed scheme is with the user's own TTP.

13. *On-line communication between TTPs should not be required*. The independent generation of the receive keys in the proposed scheme means that no such communication is required for the proposed scheme.

IV. OPTIONS AND OTHER ISSUES

A. A modified scheme

The proposed scheme can almost certainly be modified in many ways. We briefly present one such modification here, and consider its associated advantages and disadvantages.

The modified scheme works exactly as the scheme described above, with the following exceptions:

- The secret key $K(TA,TB)$ and the key generating function f are not required. Instead, every TTP generates a public/private receive key pair for each of its users. The public receive key for each user, *A* say, is then sent, by some reliable means, to every other TTP whose users may wish to send messages to *A*.
- A user's send key pair needs to be generated by its TTP (at least the TTP needs to be provided with a copy of the user's private send key).

- Each TTP will then possess the private send and receive keys for its own users but not for any other users. Hence a TTP can always deduce a session key employed by one of its own users to encrypt or decrypt a message, by using the private send key or the private receive key of the user, as appropriate.

The main advantages of this modified scheme are as follows.

- A single receive key pair and a single send key pair will suffice for users associated with a variety of different TTPs, as long as they all agree on the parameters *g* and *p*.
- Users can have both their receive key pairs and their send key pairs changed as often as they like.

The main disadvantages are as follows.

- There is a need for TTPs to interchange large numbers of public receive keys, and subsequently store them. This requirement could be obviated by storing signed versions of public receive keys in directories, at the cost of introducing the need for directory services.

B. Trusting TTPs

The receiving party must trust the sending party's TTP, in order to verify the sending party's public key, and also because the sender's TTP can generate the receiver's private key. However, this trust only concerns communications between the receiver and senders belonging to that TTP.

There may also be a need for a certification hierarchy to identify a common point of trust for different TTPs; alternatively, all TTPs could manage their inter-relationships by bilateral agreement.

C. The Choice of Values

There has been considerable discussion in the literature on the benefits of using a composite modulus for Diffie-Hellman. This, and other matters such as the length of the modulus *p* and the primitive element *g*, are beyond the scope of this paper.

D. Commercial Value

The proposed scheme relies entirely on its perceived value to users in order to be taken up. Service providers will want to recover the cost of setting up the service from their customers. Therefore the scheme must be able to provide value-added end-to-end services that users want. Further investigation is required to assess the level of demand for services such as:

- end-to-end encryption;
- end-to-end authentication;
- non-repudiation of sent messages;
- message integrity.

Given that users will be paying for these services, they will expect a sufficient level of security. In the event of security failure with financial impact on the user, he will expect to be able to recover this, either via his insurers or from the organisation running the TTP. This makes running a TTP a potentially expensive business, unless the financial risks run by the TTP can be adequately protected against. If TTPs are not commercially viable, then the scheme will not be viable.

E. Combined two-way Session Key

The two-way version of the proposed scheme provides two keys for communication: one for each direction. These could be combined to form a single session key, or just one of the keys could be used. The advantages and disadvantages of this are a matter for further study.

F. Sharing Keys Between Trusted Third Parties

In an environment where commercial TTPs will be looking to offer additional services to their users, it is possible that some users will want the extra reassurance offered by having their keys shared between a number of independent TTPs. The proposed protocol is easily adaptable to provide this feature. For instance, the ideas of Micali [**3**] for adding secret sharing on top of existing schemes could be adopted.

V. OTHER PUBLISHED SCHEMES

A. The Goss Scheme

A scheme designed by Goss has been patented in the US [**4**]. In this scheme, a shared secret key is established by combining two Diffie-Hellman exponentiations using fixed and varying (per session) parameters. At first sight, this appears to correspond to the receive and send keys in the proposed scheme. However, the Goss scheme uses a universal modulus and primitive element. If x and x' are A's fixed and variant keys, and y and y' are B's, then the shared key is calculated as

$$\alpha^{xy'} \oplus \alpha^{x'y}$$

This could be viewed as a variant of the proposed two-way protocol whereby a universal modulus and primitive element are used and the two keys are combined by XOR-ing them.

B. Yacobi Scheme

This scheme [**5**] is almost identical to the Goss one, but uses a composite modulus, and combines the session keys by modular multiplication rather than XOR-ing.

VI. REFERENCES

[**1**] National Institute of Standards and Technology, *FIPS Publication 185: Escrowed encryption standard*. February 1994.

[**2**] W. Diffie and M.E. Hellman, *New directions in cryptography*. IEEE Transactions in Information Theory **IT-22** (1976) 644-655.

[**3**] S. Micali, *Fair cryptosystems*. MIT Technical Report **MIT/LCS/TR-579.b**, November 1993.

[**4**] U.S. Patent 4956863, *Cryptographic method and apparatus for public key exchange with authentication*. Granted 11th September 1990.

[**5**] Y. Yacobi, *A key distribution paradox*. In *Advances in Cryptology - CRYPTO 90*, Springer-Verlag, Berlin (1991) pp. 268-273.