KEY ESCROW IN MULTIPLE DOMAINS[1]

Liqun CHEN

ROYAL HOLLOWAY, UNIVERSITY OF LONDON
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK

Christopher J. MITCHELL
ROYAL HOLLOWAY, UNIVERSITY OF LONDON

[1] This work has been funded by the European Commission under ACTS project AC095 (ASPeCT).

Abstract

This paper presents a key escrow system which meets possible requirements for international use, where escrow agencies do not span more than one country. We first propose a special conference key distribution scheme, where the key consists of contributions from all members. Each member can independently choose how often to access the conference key and update his contribution to the key. We then make use of this scheme to design a multi-agency escrow mechanism. Each escrow agency can independently provide warranted access to the user's communications, although they must have access to certain public information from other agencies via a globally accessible file.

1 Introduction

1.1 Motivation

Modern mobile telecommunications systems are breaking down the regulatory boundaries between different countries. In particular, third generation mobile systems are intended to provide communications networks covering the whole world. To meet growing legitimate needs for confidentiality, an end-to-end encryption service will need to be provided in these future mobile networks. In contrast to this user requirement, governments may need to intercept user traffic in this international communications environment to combat crime and protect their national security.

> *Consider the following scenario: two mobile users A and B, who communicate with each other using end-to-end encryption, are citizens of countries C and D, respectively, work in countries E and F, are registered with two mobile companies based in countries G and H, and are roaming in two countries I and J. Their traffic might conceivably need to be intercepted by law enforcement agencies in any of countries $C - J$.*

Note also that a user might wish to make use of an international telecommunications system with key escrow for one or more of the following reasons.

1. The user has to get assistance to establish secure communications with other users.
2. The user wants to have a recovery or back-up function for a session key to enable recovery if the key is lost (e.g., see [12] and [9]).
3. The user wants to use a third party as a notary to prevent the other user from denying the communications later.
4. The user wants to get protection from his government if something unpredictable occurs associated with the communications.
5. The user is legally obliged to escrow keys used for end-to-end encryption.

Hence an international key escrow system may be required which provides relevant agencies of all governments involved with warranted access to user communications. To make matters more complicated, the agencies involved may not trust one other. For example, a law enforcement agency in one country might not wish to let its counterpart in another country know that a particular user's communications are being intercepted.

1.2 Design requirements

Following the US government's Clipper proposals [1], a number of key escrow systems have recently been proposed, and for an overview of the field, the reader is referred to [4]. Certain key escrow systems have been designed specifically for international use, in that two domains are involved. For examples, Jefferies, Mitchell and Walker [7] make use of a pair of trusted third parties as escrow agencies, one in each domain, who separately compute an escrowed key using a secret key shared between them. Frankel and Yung [6] consider a key escrow agency (or agencies) working for mutually distrusting domains. Chen, Gollmann and Mitchell [3] adopt multiple third parties, who are trusted collectively but not individually, to perform key escrow in mutually mistrusting domains.

In this paper, we describe an international key escrow scheme for the case where more than two countries are involved, and where the countries involved do not trust one another; for the maximum generality we refer to *domains* instead of countries throughout. We refer to *interception authorities* where we mean bodies such as law enforcement agencies who may be given the right to access communications within a single domain. We also refer to *escrow agencies* who will be responsible for maintaining all the information necessary to provide access to interception agencies, when presented with the appropriate legal authorisation.

We now state our requirements for key escrow in multiple domains.

1. Escrow agencies will not span more than one domain, although they must have access to public information from other domains. This public information, for which retrieving agencies must be provided with origin authentication and integrity services by some means, is either placed in a globally accessible file or sent with the messages which can be intercepted for warranted interception.

2. No domain can individually control the generation of an escrowed key, and the escrowed key consists of contributions from all domains.

3. Escrow agencies in different domains can separately gain access to the escrowed key.

4. Escrow agencies in different domains can separately choose how often to update their contribution to an escrowed key.

5. Apart from the escrow agency in the domain that is responsible for providing the user authentication and key distribution service, escrow agencies do not have to be on-line during user authentication and key distribution processing.

In this paper we use a conference key scheme to construct a multiple domain key escrow system meeting the above requirements. The proposed mechanism is based upon the Diffie-Hellman algorithm for key exchange [5]. An escrowed key is an n-iterated Diffie-Hellman exponentiation, where n is the total number of users and escrow agencies involved.

2 Key escrow mechanism

2.1 Background

The proposed mechanism operates in the context of a single user making use of a key escrow service involving multiple domains, each domain having an escrow agency selected by the user. The key escrow service does the following.

- It provides support for key establishment for the user, when communicating with other users (examples of how this operates are provided in Section 2.5).
- The key is established for the user in such a way that an escrow agency in any domain can provide warranted access to that user's communications, without direct support from any of the agencies in other domains.

2.2 A conference key scheme

We next describe a conference key distribution scheme, on which our key escrow system is based. As was mentioned earlier, an escrow agency in one domain might not wish to let the agencies in any other domains know that a particular user's communications are being intercepted. For this reason we have a special requirement on our conference key scheme, namely that not all the members need actively participate in computing the conference key, provided that an authenticated copy of each member's public key agreement key is accessible to all other members.

A number of conference key schemes have recently been proposed. Burmester and Desmedt, [2], surveyed various basic configurations for conference key distribution, e.g. 'star', 'tree', 'broadcast' and 'cycle', which are designed to exploit the particular configuration of the network used. The star based configuration with a conference chair meets our requirement. However, in a typical star based protocol, the chair totally controls key generation, which could be a major problem in the context of our key escrow scheme.

We now present a conference key distribution scheme in which the distributed key is an n-iterated Diffie-Hellman exponentiation when n members are involved; this scheme meets our requirement for use in an escrow scheme. The basic idea of the scheme was previously described by Steer et al. in 1988 [11] using a cycle based protocol. For the purposes of our key escrow mechanism, we make use of a star configuration, and let the key be a function of all the member's public key agreement keys, such that no member including the chair can individually control key generation.

The proposed scheme makes use of a globally accessible file.

Definition 1 *We say that GAF is a Globally Accessible File if each authorised entity can read all messages in GAF, and can also write messages to the GAF. Origin au-*

thentication and data integrity for data retrieved from the GAF are provided by using a globally agreed digital signature algorithm. All the entities each have their own signature key/verification key pair and exchange verification keys in a reliable way.

Algorithm 2 *Let $U_1, U_2, \ldots, U_n$ be the members of a conference where U_1 is the chair. $U_1, U_2, \ldots, U_n$ agree values g and p with the properties required for secure operation of the Diffie-Hellman key agreement mechanism, namely that p is a large prime, $(p-1)/2$ is also a large prime, and g is a primitive element modulo p. $U_1, U_2, \ldots, U_n$ must also all have access to the GAF.*

1. *Each member U_i $(i = 1, ..., n)$ generates a private key agreement key S_{U_i} which is known only to U_i and a public key agreement key $P_{U_i} = F(S_{U_i}, g)$, and places P_{U_i} signed by U_i in GAF, where here, as throughout, $F(x, y) = y^x \bmod p$.*

2. *U_1 computes the following private and public key agreement value pairs:*

$$S_{U_1U_2} = F(S_{U_1}, P_{U_2}),\ P_{U_1U_2} = F(S_{U_1U_2}, g),$$
$$S_{U_1U_2U_3} = F(S_{U_1U_2}, P_{U_3}),\ P_{U_1U_2U_3} = F(S_{U_1U_2U_3}, g),$$
$$\vdots$$
$$S_{U_1U_2\ldots U_{n-1}} = F(S_{U_1U_2\ldots U_{n-2}}, P_{U_{n-1}}),\ P_{U_1U_2\ldots U_{n-1}} = F(S_{U_1U_2\ldots U_{n-1}}, g),\ \text{and}$$
$$S_{U_1U_2\ldots U_n} = F(S_{U_1U_2\ldots U_{n-1}}, P_{U_n})$$

   U_1 then places all the public key agreement values ($P_{U_1U_2\ldots U_i}$, $2 \le i \le n-1$) in GAF, in each case signed by U_1.

3. *Each member U_i ($2 \le i \le n$) then reads certain of the other members' signed public key agreement keys ($P_{U_{i+1}}, P_{U_{i+2}}, \ldots, P_{U_n}$) and one of the chair's signed public key agreement values ($P_{U_1U_2\ldots U_{i-1}}$) from GAF, verifies them using the appropriate members' public verification keys, and computes the conference key $K = S_{U_1U_2\ldots U_n}$ in the following way:*

$$\begin{aligned} y_{i1} &= F(S_{U_i}, P_{U_1U_2\ldots U_{i-1}}), \\ y_{i2} &= F(y_{i1}, P_{U_{i+1}}), \\ &\vdots \\ y_{i,n-i+1} &= F(y_{i,n-i}, P_{U_n}) = K. \end{aligned}$$

Theorem 3 *By following the above algorithm, all members of the conference can independently compute the same conference key.*

Proof. We simply need to show that the key $y_{i,n-i+1}$ computed by U_i $(2 \le i \le n)$ is the same as the key $S_{U_1U_2\ldots U_n}$ computed by U_1.

By definition:

$$\begin{aligned} y_{i1} &= F(S_{U_i}, P_{U_1U_2\ldots U_{i-1}}) \\ &= F(S_{U_i}, F(S_{U_1U_2\ldots U_{i-1}}, g)) \\ &= F(S_{U_1U_2\ldots U_{i-1}}, F(S_{U_i}, g)) \quad \text{(by the commutativity property of exponentiation)} \\ &= F(S_{U_1U_2\ldots U_{i-1}}, P_{U_i}) \\ &= S_{U_1U_2\ldots U_i} \end{aligned}$$

Thus, by induction, we can show that $y_{ij} = S_{U_1U_2\ldots U_{i+j-1}}$ $(1 \leq j \leq n-i+1)$ and the result follows. $\square$

2.3 The escrow mechanism

Suppose that user A has m escrow agencies $T_1, T_2, \ldots, T_m$, one per domain. The identifiers of the m escrow agencies associated with A will need to be bound in some way to the identifier of A, typically by means of a signature verifiable by all parties. One of the agencies must be capable of real-time communications with the user — we call this the *on-line agency*; the others are referred to as *off-line agencies*. Note that which of the m agencies is on-line may vary. Without loss of generality suppose the on-line agency is T_1.

For example, in future mobile telecommunications networks, A's identity, concatenated with the identities of A's m escrow agencies, might be signed by a telecommunications service provider. In such a case, the public signature verification key of the service provider would need to be known to all relevant agencies in a reliable way. Further, the on-line agency could be the network operator for the network which A is currently using to access the telecommunications service.

Prior to use of the mechanism, $T_1, T_2, \ldots, T_m$ and A need to agree a number of parameters.

- ◇ They must agree among them values g and p. These values may be different for each group of escrow agencies associated with a user, and must the properties required for secure operation of Diffie-Hellman key agreement (see Section 2.2).
- ◇ They must agree on a digital signature algorithm, each have their own signature key/verification key pair, and exchange verification keys in a reliable way (e.g. using signed public key certificates). In fact the user A only needs to know the on-line agency's public verification key, and hence A need not store the public keys for those escrow agencies which are never required to be 'on-line agencies'.
- ◇ They must agree a conference key distribution scheme, which can be used to establish a secret key $S_{T_1T_2\ldots T_m}$ satisfying $1 \leq S_{T_1T_2\ldots T_m} \leq p-1$. In the previous section, we described a conference key scheme with appropriate properties, and we use this scheme to construct our escrow mechanism below. However, this could be replaced with other suitable conference key schemes.

Before use of the mechanism the following preparatory work also needs to be performed.

- An authority trusted by A (e.g. A's Service Provider in a telecommunications environment) must sign a string consisting of A's name concatenated with a set of escrow agencies. This signed string shall be provided to A. There may be many such signed strings, for use by A in differing circumstances (e.g. depending on which country A is working in).
- Each escrow agency T_i shall insert a public key agreement key for A in the *GAF* (and shall retain the corresponding private key agreement key). The public key can be stored in the GAF tagged with A's identity.

The mechanism below can be used by A and the escrow agencies T_1, T_2, ..., T_m to establish an escrowed key K_A.

Mechanism 4 *A initially chooses one of the escrow agencies from its set of m escrow agencies to be the on-line entity. The identity of the on-line identity will typically vary depending on the location of A. As previously described, we denote the on-line agency by T_1 and the $m-1$ off-line agencies by $T_2, T_3, \ldots, T_m$.*

1. *A sends a request for a new key to T_1, together with a copy of its identity concatenated with the identities of $T_1, T_2, \ldots, T_m$ all signed by a trusted authority.*

2. *T_1 checks the signature, and places the signed string in GAF to start chairing the conference among $T_1, T_2, \ldots, T_m$. T_1 also recovers the public keys for A for each of the other agencies $T_2, T_3, \ldots, T_m$ ($P_{T_2}, P_{T_3}, \ldots, P_{T_m}$ say). Following the conference key distribution scheme described in Algorithm 2, T_1 also computes $P_{T_1T_2\cdots T_i}$, for every i ($2 \leq i \leq m-1$), and places these values in the GAF.*

3. *Following the conference key scheme described in Algorithm 2, each member of the conference can now (if they wish) obtain the private conference key $S_{T_1T_2\ldots T_m}$.*

4. *T_1 signs the public conference key value $P_{T_1T_2\ldots T_m} = F(S_{T_1T_2\ldots T_m}, g)$, and sends this signed key to A.*

5. *A generates his private key agreement key S_A and the corresponding public key agreement key $P_A = F(S_A, g)$. A then computes the escrowed key as $K_A = F(S_A, P_{T_1T_2\ldots T_m})$.*

6. *A signs P_A and sends it with messages encrypted using K_A. These messages may be intercepted for warranted interception.*

7. *Any entity given the private conference key and P_A can compute the escrowed key as $K_A = F(S_{T_1T_2\ldots T_m}, P_A)$.*

2.4 Warranted interception

When a warrant for legal interception of a message is required in one domain, there are then two possible ways for the escrow agency to provide warranted access to the message.

- The agency could pass the private conference key to the interception authority in the same domain, and then take no further part in the interception process.

- The agency could compute the session key used to encrypt a message presented to it by the interception authority (using the private conference key), decipher the message, and then return it the authority, without revealing the conference key itself.

2.5 Application examples

We conclude the mechanism description by considering three examples of communications services where it could be employed. In these examples we assume that data transferred to and from the GAF has its origin and integrity protected (e.g. by the use of signatures and signed certificates). We also assume that all the verification key certificates and signed public key agreement keys are sent with the encrypted messages, so that these public keys will be accessible to any related users, escrow agencies and interception authorities.

Example 1 *Mobile telephony between users A and B, where A and B both make use of a key escrow service.*

Suppose that A has m escrow agencies T_1, T_2, ..., T_m, and B has n escrow agencies V_1, V_2, ..., V_n. A is roaming in the domain with an on-line escrow agency T_1, and B is roaming in the domain with an on-line escrow agency V_1. By following the procedure of Mechanism 4, A's asymmetric escrowed key pair is $(K_A = S_{AT_1T_2...T_m}, P_{K_A} = F(K_A, g))$, and B's asymmetric escrowed key pair is $(K_B = S_{BV_1V_2...V_n}, P_{K_B} = F(K_B, g))$. A and B then exchange their public escrowed keys in a reliable way, e.g. by sending the keys signed by T_1 and V_1 respectively, with their encrypted messages. A and B can establish a shared encryption key as $K_{AB} = F(K_A, P_{K_B}) = F(K_B, P_{K_A})$. This key can be separately computed by all the escrow agencies associated with A or B.

Example 2 *Email from A to B, i.e. A is the sender and B is the receiver, where only the receiver makes use of a key escrow service.*

Suppose that only B has associated escrow agencies $V_1, V_2, \ldots, V_n$ (where V_1 is on-line). By following the procedure of Mechanism 4, B's asymmetric escrowed key pair is $(K_B = S_{BV_1V_2...V_n}, P_{K_B} = F(K_B, g))$. A has his own asymmetric key agreement key pair $(S_A, P_A = F(S_A, g))$. Using these two key pairs, A and B establish a shared key as $K_{AB} = F(S_A, P_{K_B}) = F(K_B, P_A)$. This key is escrowed by all domains associated with B. In [7], the receiver's private key agreement key is escrowed by both the sender's domain and receiver's domain. This can be considered as a special case of this example.

Example 3 *Encrypted file storage for user A, where A makes use of a key escrow service.*

Suppose that A has associated escrow agencies T_1, T_2, ..., T_m (where T_1 is on-line). By following the procedure of Mechanism 4, the key used for the stored file encryption is $K_A = S_{AT_1T_2...T_m}$, which is escrowed by all domains associated with A.

3 Analysis of the mechanism

The mechanism has the following properties.

- First note that in the conference key distribution procedure, every member apart from the chair only places his public key agreement key in GAF, and makes no other contribution. If one member renews his contribution for a session, the conference key is changed. Any members who are interested in the current session must read the *GAF* and compute the renewed key. However, if one member does not want to update his own key agreement key, and/or is not interested in the current session, he takes no action, as long as his old public key agreement key is still available in *GAF*. This property allows all the members, apart from the chair, 'independently' to choose how often to update their contribution to the conference key and to gain access to the key without communicating with any other members.

Thus, for the key escrow mechanism, each agency can separately choose when to update its own contribution and when to gain access to the escrowed key. As long as the *GAF* does not reveal who accesses which data, no agency in any domain will know if the other agencies are interested in a particular user's communications. This property meets our basic requirement for an international key escrow model described in section 1.2.

- An escrowed key in the proposed mechanism is a function of contributions from the user and all relevant escrow agencies, and no individual can control key generation. One advantage is that if any entity updates its contribution using a fresh (i.e. not a replay of an old key) and random (i.e. not predictable by any party) number, the key will be fresh and random. Any entity which has updated his contribution to the key can verify the freshness of the key. Another benefit is that this property prevents two users, e.g. in Example 1 or 2, abusing the mechanism by using the 'shadow-public-key' attack proposed by Kilian and Leighton [8]. Chen, Gollmann and Mitchell, [3], discussed how the attack could work in a key escrow system based on Diffie-Hellman key exchange. However, other abuses by collusion among entities may still be possible. We must note that it is difficult for any key escrow system to force two users to use only the current escrow key for their end-to-end encryption if the users share a secret or can use their own security system.

- In the conference key scheme, any contribution of the chair, apart from his own public key agreement key, can be verified by at least one of the other members. It is simple to see that $P_{U_1U_2}$ can be separately computed by U_2, and each $P_{U_1U_2\ldots U_i}$ $(3 \leq i \leq n)$ can be separately computed by U_j $(2 \leq j \leq i)$, given $P_{U_1U_2\ldots U_{j-1}}$. This property to some extent prevent the chair from 'cheating'. Even if the chair colludes with one or more members, e.g. with U_2 to U_i $(3 \leq i \leq n)$, by contributing a forged $P_{U_1U_2\ldots U_i}$, their behaviour only results in changing their contributions to the key, and hence they gain nothing by such an 'attack'.

  Note that if none of the other members verifies the chair's contributions, the chair, by contributing a forged $P_{U_1U_2\ldots U_i}$, can force the other members to form two sub-conferences, so that U_2, U_3, ..., U_i accept one key and U_{i+1}, U_{i+2}, ..., U_n accept another key. Even worse, if the scheme is used in the proposed key escrow mechanism, the chair T_1 can blind all the other agencies and escrow the user's key alone. If this is considered as a serious weakness, the mechanism should explicitly include a verification function, so that each relevant entity checks the chair's contributions and broadcasts any contribution which does not match the protocol specifications.

- There is no need to transfer any secret messages in the mechanism. Only origin authentication and data integrity are required. All relevant public key agreement keys must be accessible for any related entities, e.g, by being placed in GAF or sent with messages interceptable on warrant.

- As with any other key escrow system with escrow agencies, a user must trust the agency he appoints to act on his behalf. In the proposed mechanism, the user must trust each escrow agency to maintain the escrowed key properly, i.e. not reveal it or use it illegally. In environments where this requirement is too strong for an individual agency, it is possible that some users will want the extra reassurance offered by having their keys shared between a number of independent agencies in each domain. The proposed mechanism can be adapted to provide this feature by using a set of moderately trusted agencies, e.g. see [3].

- The security of the proposed mechanism is based on the intractability of the Diffie-Hellman problem, i.e. that given elements g, g^x and g^y mod p it is computationally infeasible to obtain g^{xy}, which has withstood detailed scrutiny over a period of time. Note that the Diffie-Hellman problem can certainly be solved using discrete logarithms, and Maurer [10] presents some evidence for the equivalence of the two problems.

  The use of modular exponentiation in the mechanism can be generalised by using any function F satisfying the Diffie-Hellman commutativity and computational intractability conditions.

4 Conclusions

The conference key scheme presented allows each member of the conference 'independently' to choose how often to access the conference key and update his contribution to the key, although his updated contribution will be used by all the other members to renew the key. Based on this scheme, we have proposed a key escrow mechanism suitable for international use. The security of the proposed mechanism is based on the intractability of the Diffie-Hellman scheme.

Acknowledgements

The authors would like to thank Dieter Gollmann and Wenbo Mao for their valuable comments on an earlier version of the paper, and Didier Mazingue for his French translation of the abstract for this paper.

References

[1] National Institute of Standards and Technology. FIPS Publication 185: Escrowed Encryption Standard. February 1994.

[2] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In A. D. Santis, editor, *Lecture Notes in Computer Science 950, Advances in Cryptology — EUROCRYPT '94*, pages 275–286. Springer-Verlag, 1994.

[3] L. Chen, D. Gollmann, and C. Mitchell. Key escrow in mutually mistrusting domains. In M. Lomas, editor, *Lecture Notes in Computer Science 1189, Security Protocols — International Workshop, Cambridge, United Kingdom, April 1996*, pages 139–153, Springer-Verlag, 1997.

[4] D.E. Denning and D.K. Branstad. A taxonomy for key escrow encryption systems. *Communications of the ACM*, 39(3):34–40, 1996.

[5] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, November 1976.

[6] Y. Frankel and M. Yung. Escrow encryption systems visited: attacks, analysis and designs. In D. Coppersmith, editor, *Lecture Notes in Computer Science 963, Advances in Cryptology — CRYPTO '95*, pages 222–235. Springer – Verlag, 1995.

[7] N. Jefferies, C. Mitchell, and M. Walker. A proposed architecture for trusted third party services. In E. Dawson and J. Golić, editors, *Lecture Notes in Computer Science 1029, Cryptography: Policy and Algorithms Conference*, pages 98–104. Springer-Verlag, 1996.

[8] J. Kilian and T. Leighton. Fair cryptosystems, revisited. In D. Coppersmith, editor, *Lecture Notes in Computer Science 963, Advances in Cryptology - CRYPTO '95*, pages 208–221. Springer – Verlag, 1995.

[9] D.P. Maher. Crypto back-up and key escrow. *Communications of the ACM*, 39(3):48–53, 1996.

[10] U.M. Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms. In Y.G. Desmedt, editor, *Lecture Notes in Computer Science 839, Advances in Cryptology — CRYPTO '94*, pages 271–281. Springer-Verlag, 1994.

[11] D.G. Steer, L. Strawczynski, W. Diffie, and M. Wiener. A secure audio teleconference system. In S. Goldwasser, editor, *Lecture Notes in Computer Science 403, Advances in Cryptology - CRYPTO '88*, pages 520–528. Springer-Verlag, 1988.

[12] S.T. Walker, S.B. Lipner, C.M. Ellison, and D.M. Balenson. Commercial key recovery. *Communications of the ACM*, 39(3):41–47, 1996.