Failsafe Key Escrow Systems
(Extended Abstract)

Tom Leighton

Mathematics Department
Laboratory for Computer Science
MIT
Cambridge, MA 02139

August 6, 1994

Abstract

This paper describes a method for escrowing cryptographic keys, which we call **Failsafe Key Escrow (FKE).** The method is substantially more secure than alternatives such as the Fair Public Key Cryptosystem approach advocated by Micali, and it is particularly well suited for use in escrowing DSS keys.

Keywords: Cryptosystems, Secret Sharing, Verifiable Secret Sharing, Key Escrow, Fair Cryptosystems, Digital Signatures, Encryption

1 Introduction

In this paper, we describe a method for escrowing cryptographic keys (which we call **Failsafe Key Escrow (FKE)**) in which the authorities interact with the users to select the cryptographic keys that are to be escrowed. The system has the following five properties:

Property 1: Each user in the system should have sufficient control over his or her secret key to be sure that the key is chosen securely.

Property 2: The central authority will also be guaranteed that the secret key for each user is chosen securely even if the user doesn't have access to a good random number generator or if the user fails to use the random number generator properly.

Property 3: Each user will be guaranteed that his or her secret key will remain secret unless a sufficient number of trustees release their shares of the key to the central authority.

Property 4: The central authority needs to be assured that it can obtain the secret key for a user who is suspected of using his or her key for encryption in the context of illegal activities by retrieving shares of the key from a certain number of trustees.

Property 5: The central authority needs to be assured that the escrow system will not be abused by criminals in a way that allows them to communicate without fear of court-authorized wiretapping. More precisely, if two criminals abuse the FKE by using their public keys to communicate using any published public-key encryption algorithm, and the central authority is provided knowledge of the criminals' secret keys by the trustees, then it should as easy (at least on a probabilistic basis) for the central authority to decrypt the message traffic between the criminals as it is for the criminals themselves to decrypt that traffic.

The new method is substantially more secure than the Fair Public-Key Cryptosystem (FPKC) approach advocated by Micali [5]. This is because the FPKC approach does not satisfy Properties 2 and 5.

In particular, Killian [3] has recently shown how the public keys stored in Micali's FPKC escrow scheme can be used by criminals to communicate (using

a published public key cryptosystem (PKC)) in a way that the Government will not be able to decipher, even if the secret keys for the users are provided to the Government by the trustees. This means that it is fairly easy for criminals to subvert the Micali FPKC so as to prevent the Government from deciphering their communications. Such abusive use of the key escrow system is not possible in the Failsafe Key Escrow approach described here.

The Failsafe Key Escrow method described here also has the advantage of insuring that legitimate but technically unsophisticated users will be prevented (with overwhelmingly high probability) from choosing keys which are not cryptographically secure. Hence, the Government or a company can be sure that its employees are getting secure keys even if they fail to properly access a secure random number generator. Such assurances are not possible in the Micali FPKC.

The FKE method described here is no more expensive to use than (and, in some cases, it is much less costly than) Micali's FPKC technology. In addition, by Properties 1 and 3, it provides the same basic assurances of fairness to legitimate users as does the Micali FPKC. Hence, the Failsafe Key Escrow approach offers all of the benefits of FPKC while providing the substantial advantage of security for the Government as well as the unsophisticated user.

2 Background

In a Public Key Cryptosystem, each user is assigned or chooses a matching pair of keys (P_X, S_X), where P_X is the public key corresponding to the pair and S_X is the secret key. For authentication purposes, the public key for each user is catalogued and/or certified by a central authority (or authorities) so that other users in the system can retrieve the authentic public key for any individual. Public Key Cryptosystems can be used for many purposes, including encryption and/or digital signatures.

One problem with a PKC (and Cryptosystems in general) is that they may be abused by non-law-abiding users. For example, two criminals could communicate using a PKC established by the Government and an authority would have no way to decrypt their message traffic, even if the authority had received a court authorization to wiretap the communication. Such activity might take place even if the PKC were established solely for the purposes of

digital signatures since the criminals might use the PKC for other purposes such as encryption.

This problem has been addressed in a series of papers. Blakley [1] and Shamir [6] describe methods wherein the secret cryptographic key of each user is shared among one or more trustees. (Trustees are presumably few in number and are highly trusted entities.) In particular, each trustee is given a secret piece of the secret key for each user. The sharing of a key needs to satisfy 2 properties. First, no subset of k trustees should be able to pool their knowledge in order to figure out the secret key of a user. Second, any set of $h > k$ trustees should be able to recover the secret key of a user by pooling their shares of that key. Many such "secret sharing" schemes are known in the literature (e.g., see the survey paper by Simmons [7]). In such a scheme, the user is assured that the authorities cannot learn his or her secret key without the approval of at least $k + 1$ trustees, and the authorities are assured that they can obtain the secret key of any individual with the approval of any h trustees. Variations of these schemes are known which can also handle trustees who work in cooperation with the criminals, provided that the number of such malicious trustees is not too large.

One difficulty with the secret sharing schemes is that there is no provision for insuring that the trustees have received valid shares for each user's secret key. Indeed, when the trustees reveal their shares under a court order (say), the shares may be found to be useless because the criminal user did not provide proper shares of his or her secret key. This problem is resolved in [2], where it is shown how shares can be provided in a way so that each trustee can be assured that he or she has received a valid share of the secret key. A user who does not provide valid shares for their secret key can then be identified and excluded from the system.

A secret sharing scheme in which each trustee can be assured that he or she has a valid share of a secret is known as a Verifiable Secret Sharing (VSS) scheme. Many such schemes are known in the literature. In [5], Micali claims that a VSS scheme used in this fashion forms what he calls a Fair Public-Key Cryptosystem. Although the precise definition of a Fair PKC is not provided, Micali states that a key property of a Fair PKC is that it "cannot be misused by criminal organizations" [5]. As demonstrated by the Killian attack, however, it is clear that the Micali method for Fair PKCs can be seriously misused by

criminals.

The flaw in the Micali method is derived from the fact that it is possible for a user X to choose a pair of keys (S_X, P_X) with the special properties that:

1) the trustees can be provided with valid shares of the secret key S_X, and

2) the public key P_X can be easily converted into a second public key P'_X (using a published algorithm) for a second cryptosystem for which the user has also precomputed a second secret key S'_X.

The criminal user can then communicate using the second cryptosystem and the second pair of keys. The central authority (with the aide of the trustees) can retrieve S_X but this will not be useful in deciphering traffic encrypted with S'_X. Moreover, the central authority may have no hope of discovering S'_X.

This problem can be resolved by having the trustees themselves select the pair of keys for each user, as suggested in [4]. But schemes in which the trustees select the secret key for each user may leave the user with no assurance that his key has been properly generated (so as to be secure). Such a scheme would not satisfy Property 1.

It would be desirable to have a method for the selection of key pairs for individuals that protects the privacy and security concerns of law-abiding users as well as the security concerns of the central authority. That is the subject of this paper.

3 The Failsafe Key Escrow Approach

In what follows, we describe one embodiment of the Failsafe Key Escrow approach. This embodiment is based on a Discrete-Log PKC such as Diffie-Hellman or DSS. Here we assume that a prime modulus Q and a generator G for Z_Q are publicly known. In this case the public key P_X that is escrowed for user X is $G^{S_X} \bmod Q$, where S_X is the secret key for user X. The escrow system that will be used in conjunction with the US Digital Signature Standard has this form.

The keys for a user X are selected as follows:

Step 1: The user picks a random secret value A from $[0, Q-2]$ and announces

the value of $G^A \bmod Q$ to the trustees and/or the central authority.

Step 2: The user "shares" A with the trustees using a VSS scheme. (The precise VSS scheme that is used depends on the degree to which the trustees can be trusted to behave properly and the degree to which the users distrust the trustees.) This requires X to send the shares of A to the trustees and it requires the trustees to verify that they received valid shares of A.

Step 3: The trustees and/or the central authority select a random value B from the interval $[0, Q-2]$ and they set the user's public key to be $P_X = (G^A)G^B \bmod Q$. The value of B is returned to the user and is escrowed with the public key for X. The value of B is not released to the public.

Step 4: The user then sets his secret key to be $S_X = A + B \bmod (Q-1)$.

In what follows, we show that Properties 1–5 hold for this system. For simplicity, we will argue informally.

Verification of Property 1: Every user who follows the protocol can be sure that he or she has a randomly chosen secret key. This is because the user chooses A at random in $[0, Q-2]$. The authority chooses B, but does so with no knowledge of A (depending on the VSS scheme that is used). Hence, from the user's point of view, A might as well have been selected after B. This means that if A was selected at random by the user, then the user can be assured that $S_X = A + B \bmod (Q-1)$ is a random integer in $[0, Q-2]$.

Verification of Property 2: Even a user who fails to select the value of A correctly (e.g., by using a birthday instead of a random number generator) will get a random secret key. This is because the value of B is selected randomly by the authorities after the user commits to the value of A. Hence, the authorities can be assured that $S_X = A + B \bmod (Q-1)$ is a random integer in $[0, Q-2]$.

Verification of Property 3: Each user can be assured that his or her secret key stays secret unless a sufficient number of trustees release their shares. This is because knowledge of A can be revealed only with the assent of a sufficient number of trustees by the properties of the VSS scheme. Even if B were to be public, this means that $A + B \bmod (Q-1)$ will remain secret unless a sufficient number of trustees cooperate to reveal A.

Verification of Property 4: The central authority is guaranteed to be able to retrieve the secret key of any user provided that a sufficient number of trustees reveal their shares. This is because the properties of the VSS scheme assure that a sufficient number of trustees can collaborate to reveal A. Since B is escrowed, it is then a simple matter to compute $S_X = A + B \bmod (Q-1)$.

Verification of Property 5: If two criminals attempt to abuse the FKE by using their public keys to communicate using any published public-key encryption algorithm, and the central authority is provided knowledge of the criminals' secret keys by the trustees, then it should be as easy (at least on a probabilistic basis) for the central authority to decrypt the message traffic between the criminals as it is for the criminals themselves to decrypt that traffic.

Proving this fact is somewhat more difficult. Suppose that two criminals X and Y attempt to abuse the FKE by using their public keys to communicate using any published PKC. Let P'_X be the public key for X in the PKC. Without loss of generality, we will assume that P'_X is computable as a published function F of P_X. (I.e., $P'_X = F(P_X)$.) Otherwise, the criminals would be using secret information to communicate (in which case, they wouldn't need to abuse the FKE in the first place).

Let S'_X be the matching secret key for P'_X in the PKC, and define H to be the (published but presumably hard to compute) function that maps a public key of the PKC to its corresponding secret key. (I.e., $S'_X = H(P'_X)$.) Then $S'_X = HF(P_X) = HFE(S_X)$, where $E(S_X) = G^{S_X} \bmod Q$. Since $S_X = A + B \bmod (Q-1)$, we know that $S'_X = R(A + B \bmod (Q-1))$, where $R = HFE$ is a published (but possibly hard to compute) function.

The user picks A and so he or she may know a great deal of information about A that is unknown to the central authority (such as the discrete-log of A). This means that it might be much easier for the user to compute $R(A)$ than it would be for the central authority to compute $R(A)$. The user has no control over the distribution of $A + B \bmod (Q-1)$, however, since this distribution is uniform for all A. This means that before B is selected, the user can generate no more information (probabilistically speaking) about $A + B \bmod (Q-1)$ than can the central authority in an equivalent amount of time. (To be precise, we need to assume that the central authority has the

same initial knowledge and computational power as the user for this statement to be true.) Once B is selected, both the user and the central authority know $A + B \bmod (Q-1)$ (assuming that the trustees have cooperated to reveal A, of course), and the central authority will be equally capable of generating S'_X as the user.

This completes the sketch of the proof that Property 5 holds for the FKE protocol.

Similar protocols can be developed for use with other PKCs such as RSA, but the details become more complicated since the authorities need to interact with the user to choose a "random" number with some special structure. For example, the public keys used with RSA need to be the product of a small number of primes.

The proof method just described can also be extended to show that the FKE system provides security against collections of criminals that band together to produce public keys which can be combined to form a single public key in another cryptosystem.

4 Applications

Failsafe Key Escrow systems can be used in conjunction with any PKC to protect the interests of both law enforcement and the users. FKE may prove to be particularly valuable in the context of the new US Digital Signature Standard (DSS). In particular, it will be important to insure that criminals are not able to use DSS keys for the purposes of encrypting communications in a way that is indecipherable to the Government. This issue is of particular concern in the context of DSS since DSS keys can be easily adapted for encryption. The FKE approach described in Section 3 prevents precisely this sort of abuse.

5 Limitations

It is also worth pointing out the limitations of the Failsafe Key Escrow Approach. Most importantly, the FKE approach does not prevent a pair of criminals from communicating securely using secret information or an alternative escrow system, or from using other protocols for secret key agreement. The

main point of the FKE is to prevent criminals from abusing the public keys in the key escrow system. In other words, by designing the key escrow system in a failsafe fashion, the Government can be assured that the escrow system will not make it any *easier* for criminals to communicate securely.

6 Acknowledgments

I would like to thank Bonnie Berger for her help with this work.

References

[1] G. Blakley. Safeguarding cryptographic keys. *In AFIPS – Conference Proceedings*, 48:313–317, June 1979.

[2] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. *Proceedings of the 26th IEEE Symposium of Foundations of Computer Science*, pages 383–395, 1985.

[3] J. Killian. Fair public-key cryptosystems aren't. Unpublished manuscript, 1994.

[4] T. Leighton and S. Micali. Secret key distribution without public-key cryptography. *Crypto 93*, August 1993.

[5] S. Micali. Fair public-key cryptosystems. Technical Report 579, MIT Lab. for Computer Science, September 1993.

[6] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[7] G. Simmons. How to really share a secret. *Crypto 90*, pages 390–448, August 1990.