DETECTING AND REVOKING COMPROMISED KEYS

Tsutomu Matsumoto
Graduate School of Environment and Information Sciences
Yokohama National University
79-7, Tokiwadai, Hodogaya, Yokohama, 240-8501, Japan
tsutomu@mlab.jks.ynu.ac.jp

This note describes two correlated topics in cryptography. The first topic is the entity exclusion, or how to distribute a cryptographic key over a broadcasting shared by n entities so that all but d excluded entities can get a group key. In a system such as Pay-TV, Internet multicasting and group mobile telecommunication, a dishonest user or an unauthorized terminal should be excluded as quickly as possible. We discuss the points of evaluation and history of the field followed by concrete schemes smartly enabling the entity exclusion. The second topic is how to discover the existence of a "clone," that is, another entity with the same ID and the same secret key as the original. This problem is rather hard to solve in general. However, depending on environmental conditions there are approaches for solving the problem. We suggest some effective ways for the clone discovery.

Contents

1. Introduction

Imagine a system consisting of a lot of entities. An entity is what creates, sends, receives, processes, modifies, or uses data. The system employs cryptography to maintain the confidentiality, integrity, or authenticity of data exchanged among the entities.

A typical scenario is that each entity is assigned a unique identifier (ID) and an individualized private key as well as other parameters. Security attained by cryptography is based on the assumption that every private key is kept properly. Therefore two natural questions arise.

How can the system know that the assumption is satisfied? How can the system revoke a particular key if there is a need to do so? Revoking a key may be restated as excluding an entity possessing the corresponding key.

2. Entity Exclusion in Group Communication Over a Broadcast Channel to All Users

2.1. *Theme*

A *broadcast encryption* allows a distributor to send the same message simultaneously over a broadcast channel to all authorized users with confidentiality. Pay-TV via cable and satellite networks, Internet multicasts, and mobile group telecommunication such as a private mobile radio or a taxi radio, are typical examples. A secure and fast method to distribute a shared key (which is called a *group key* in this note) to all the proper users is required.

The main part of this note focuses on the *entity exclusion*, or how to transmit a group key over a broadcast channel shared by n entities so that all but d excluded entities can get the group key. For example, entity exclusion can prevent a lost or stolen mobile terminal to be used to eavesdrop the secret broadcasting. Entity exclusion can also prevent unauthorized access to Pay-TV and Internet.

2.2. *Development*

A simple method of entity exclusion is that a distributor distributes a new group key to each entity except the excluded users, in encrypted form by a secret key of each user. This method requires each entity to keep only one secret key, while the distributor should transmit $n-d$ encrypted new group keys.

Another simple method is that each entity has common keys for all subsets of n users. This method does not require the distributor to transmit any message, while each entity should keep a lot of keys.

To improve this trade-off between the amount of transmission and the key storage of each user, many ideas [?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?] have been proposed. Criteria for evaluation may be listed as

flexibility a fixed and privileged distributor is required or not.
reusability a secret key of each entity can be reused or not.

scalability in complexity the amount of transmission and key storage of each entity is independent of the group scale n or not.

Berkovits [?] proposed a scheme using secret sharing. Mambo, Nishikawa, Tsujii and Okamoto [?] proposed a broadcast communication scheme, which is efficient in terms of computation of each entity and the amount of transmission. These two works can be applied to key distribution for a pre-determined privileged subset, but not to a dynamically changing subset.

A major step in key distribution with entity exclusion was marked when Fiat and Naor proposed a scheme [?]. The scheme is resilient to any coalition of d users, by extending an original scheme, which excludes a single user, using multi-layered hashing techniques. In the scheme, each entity stores $O(d \log d \log n)$ keys and the distributor broadcasts $O(d^2(\log d)^2 \log n)$ messages. Blundo, Mattos and Stinson extend this basic work in the papers [?, ?], and Luby and Staddon studied the trade-off in the paper [?].

A second major step is a hierarchical key distribution scheme (called HKDS in the current note) using a balanced binary tree. This was done by two research groups of Wallner, Harder and Agee [?] and Wong, Gouda and Lam [?]. In this scheme, the amount of transmission is $O((degree{-}1) \times \log n)$ and the number of keys for each entity is $O(\log n)$, where n is the number of users on the broadcast channel and *degree* is the number of users in the bottom subgroup of the binary tree. Canetti, Garay, Itkis, Micciancio, Naor and Pinkas proposed an extended method that reduces the amount of transmission in the paper [?]. Canetti, Malkin and Nissim studied the trade-off in the paper [?].

Recently, two works on the entity exclusion problem have been presented: one by Kumar, Rajagopalan and Sahai [?] and one by Matsuzaki and Anzai [?]. In the Kumar-Rajagopalan-Sahai scheme [?] using algebraic geometric codes, each entity has an individual subset of a key-set. Redundant pieces of message using an error-correcting code are encrypted by keys belonging to users who are not excluded and are broadcast. The amount of transmission is $O(d^2)$ regardless of n and the key storage of each entity is $O(d \times \log n)$. Consequently, the scheme enables an efficient entity exclusion of which the amount of transmission does not depend on the group scale. However, this scheme still requires the key storage that depends on the group scale n, and a fixed and privileged distributor. Matsuzaki and Anzai proposed a scheme [?] using mathematical techniques which are well-known as *RSA common modulus attack* and *RSA low exponent attack*. The

Matsuzaki-Anzai scheme can simultaneously exclude up to d users. The amount of transmission is $O(d)$ regardless of n and each entity has only one key. Therefore, the scheme enables an efficient entity exclusion when the group scale n becomes large, while the distributor should pre-send a secret key of each entity for every key distribution, since the secret key cannot be reused, and the scheme requires a fixed and privileged distributor who knows all secret keys of users.

Anzai, Matsuzaki and Matsumoto [?] proposed a scheme, called **MaSK**, which is the abbreviation of **Ma**sked **S**haring of Group **K**eys. They applied the *threshold cryptosystems* given in [?] to achieve good reusability and flexibility and scalability in complexity. Therefore the trick they used may be interesting. The following two sections precisely introduce them.

3. MaSK: An Entity Exclusion Scheme

We describe MaSK, the Anzai-Matsuzaki-Matsumoto scheme [?]. MaSK can be based on an appropriate *Diffie-Hellman Problem* [?] defined over a finite cyclic group, including a subgroup of Jacobian of an elliptic curve and so on. We describe it over a prime field $\mathbf{Z}_p$. MaSK contains two phases: system setup phase and key distribution phase. Before explaining them we should clarify the target system and assumptions.

3.1. *Target system and assumptions*

The target system consists of the following:

System manager: A trusted party which decides system parameters and sets each user's secret key. It manages a public bulletin board.

Public bulletin board: A public bulletin board keeps system parameters and public keys for all users.

User i: An entity labeled i as its ID number is a member of the group. We assume the number of total users is n. Let $\Phi = \{1, 2, \ldots, n\}$ be the set of all users.

Coordinator x: A coordinator decides one or more excluded user(s) and coordinates a group key distribution with entity exclusion. We use the term "coordinator" to distinguish it from the fixed and privileged distributor discussed before. In the scheme, any entity can become the coordinator.

Excluded user j: A user to be excluded by the coordinator. Let $\Lambda (\subset \Phi)$ be a set of excluded users, having d users.

Valid user v: A user who is not an excluded user. The set of all valid users forms the *group*.

In the target system, we make the following system assumptions:

(1) All users trust the system manager. The system manager does not do anything illegal.
(2) All users have simultaneous access to the data that the coordinator broadcasts.
(3) All users can get any data from the public bulletin board at any time.
(4) The broadcast channel is not secure, i.e., anyone can see the data on the broadcast channel.

and the following security assumptions:

(1) The *Diffie-Hellman Problem* is computationally hard to solve.
(2) In $(k, n+k-1)$ threshold cryptosystems, anyone with less than k shadows cannot get any information about the secret S.
(3) Excluded users may conspire to get a group key.
(4) Excluded users may publish their secret information to damage the system security.
(5) Valid users do not conspire with excluded users. If this assumption is not satisfied, excluded users can get the group key from valid users.
(6) Valid users do not publish their secret keys.
(7) The system manager manages the public bulletin board strictly so as not to change it. Or, the public bulletin board has each parameter with the certificate checked before using the public parameters.

3.2. *System setup phase*

At the beginning, a system setup phase is carried out only once.

(1) A system manager decides a parameter k satisfying

$$0 \leq d \leq k-2 < n,$$

where n is the number of users in the group and d is the upper bound of the number of excluded users.

(2) The system manager decides the following system parameters and publishes them to the public bulletin board:

- p: a large prime number
- q: a large prime number such that $q \mid p-1$ and $n+k-1 < q$
- g: a q^{th} root of unity over $\mathbf{Z}_p$

- *sign*(s, m): a secure signature generation function, which outputs a signature Z of the message m using a secret key s. We use the signature scheme based on DLP here, such as DSA [?] or Nyberg-Rueppel message recovery signature scheme [?].
- *verify*(y, Z): a secure signature verification function, which checks the validity of the signature Z using a public key y. The function outputs the original message m if the signature Z is "valid".

(3) The system manager generates a system secret key $S \in \mathbf{Z}_q$ and stores it secretly. And, the system manager divides the system secret key S into $n + k - 1$ shadows with the threshold k, using the well-known Shamir's secret sharing scheme [?] as follows:

(a) The system manager puts $a_0 = S$.

(b) The system manager defines the following equation over $\mathbf{Z}_q$:

$$f(x) = \sum_{f=0}^{k-1} a_f x^f \bmod q \tag{3.1}$$

where $a_1, a_2, \ldots, a_{k-1}$ are random integers that satisfy the following conditions:

$$0 \le a_i \le q - 1 \quad \text{for all } 1 \le i \le k-1 \quad \text{and } a_{k-1} \ne 0\,.$$

(c) The system manager generates $n + k - 1$ shadows as follows:

$$s_i = f(i) \quad (1 \le i \le n + k - 1) \tag{3.2}$$

(4) The system manager distributes the shadows $s_1, \ldots, s_n$ to each user $1, \ldots, n$, respectively, in a secure manner. Each user keeps its own shadow as its secret key.

(5) The system manager calculates public keys $y_1, \ldots, y_{n+k-1}$ by the following equation:

$$y_i = g^{s_i} \bmod p \quad (1 \le i \le n + k - 1) \tag{3.3}$$

Then, the system manager publishes $y_1, \ldots, y_n$ on the public bulletin board with the corresponding user's ID numbers. The remaining $y_{n+1}, \cdots, y_{n+k-1}$ and the corresponding ID numbers are also published to the public bulletin board as spare public keys. The system manager may remove the secret keys $s_1, \ldots, s_{n+k-1}$ after the system setup phase. The remaining tasks of the system manager are to maintain the public bulletin board and to generate a secret key and a public key of a new user.

3.3. *Key distribution phase*

3.3.1. *Broadcasting by a coordinator*

First, a coordinator x generates a broadcast data $B(\Lambda, r)$ as follows:

(1) The coordinator x decides excluded users. Let Λ be the set of excluded users and d is the number of the excluded users.

(2) The coordinator x chooses $r \in Z_q$ at random and picks $k-d-1$ integers from the set $\{n+1, \ldots, n+k-1\}$ and let Θ be the set of chosen integers. Then, the coordinator calculates $k-1$ exclusion data as follows:

$$M_j = y_j^r \bmod p \quad (j \in \Lambda \cup \Theta) \tag{3.4}$$

using the public keys of excluded users and the spare public keys on the public bulletin board.

(3) The coordinator x calculates the following preparation data:

$$X = g^r \bmod p \tag{3.5}$$

(4) Using its own secret key s_x, the coordinator x generates the signature for the data consisting of the preparation data, own ID number, $k-1$ exclusion data, and the corresponding ID numbers:

$$Z = \mathit{sign}(s_x, X \| x \| \{[j, M_j] \mid j \in \Lambda \cup \Theta\}) \tag{3.6}$$

where $\|$ indicates *concatenation* of data.

(5) The coordinator x broadcasts the following broadcast data to all users:

$$B(\Lambda, r) = Z \| x \tag{3.7}$$

Next, the coordinator x calculates a group key U using its own secret key s_x and broadcast data $B(\Lambda, r)$:

$$\begin{aligned} U = {} & X^{s_x \times L(\Lambda \cup \Theta \cup \{x\}, x)} \\ & \times \prod_{j \in \Lambda \cup \Theta} M_j^{L(\Lambda \cup \Theta \cup \{x\}, j)} \bmod p \end{aligned} \tag{3.8}$$

where

$$L(\Psi, w) = \sum_{t \in \Psi \setminus \{w\}} \frac{t}{t-w} \bmod q \quad (\forall\, \Psi : \mathit{set}, \forall\, w : \mathit{integer}) \tag{3.9}$$

Since $M_j = g^{s_j \times r} \bmod p$ holds, the system secret key S is recovered in the exponent part of equation (**??**), gathering k sets of secret keys.

In summary, MaSK (Table **??**) is effective to implement quick group key distribution with entity exclusion function when the group scale n is very large compared to the number of excluded users d and it works well for devices with limited storage.

Table 1. Comparison of five schemes.

	Flexibility	Reusability	Scalability	
			Amount of transmission	Key storage
MaSK	OK	OK	OK	OK
HKDS	OK	OK	NG	NG
FN	NG	OK	NG	NG
MA	NG	NG	OK	OK
KRS	NG	OK	OK	NG

4. How to Discover the Existence of a Clone

4.1. *Security*

A type of attack likely to occur would be the "one-shot attack," in which an attacker reads out by some means the ID, the private key and other information from the target terminal and makes a clone. After getting the above information, the attacker returns the terminal without any change to the holder of the terminal. In other words, the attacker does not use the original terminal at the very moment when it tries to fool the center.

Figure **??** illustrates the shell model. Certificate $\langle\langle EE\rangle\rangle CA_3$ is valid at time t_1 as all three certificates are valid, but is invalid at time t_2 as certificate $\langle\langle CA_2\rangle\rangle CA_1$ has expired.

Acknowledgments

The content of this note is based on a lecture given at IMS of National University of Singapore in September 2001. The author thanks Professor Harald Niederreiter and IMS for providing such a nice opportunity.

This work was partially supported by MEXT Grant-in-Aid for Scientific Research 13224040.

References

1. R. Anderson, *Security Engineering*, John Wiley & Sons, pp. 352–353, 2001.
2. J. Anzai, N. Matsuzaki, and T. Matsumoto, "A method for masked sharing of group keys (3)," *Technical Report of IEICE*, ISEC99-38, pp. 1–8, 1999.
3. J. Anzai, N. Matsuzaki, and T. Matsumoto, "A flexible method for masked sharing of group keys," *IEICE Trans. Fundamentals*, vol. E84-A, no. 1, pp. 239–246, 2001. Preliminary version appeared as J. Anzai, N. Matsuzaki, and T. Matsumoto, "A quick group key distribution scheme with entity

revocation," *Advances in Cryptology – ASIACRYPT '99*, LNCS vol. 1716, pp. 333–347, Springer-Verlag, 1999.

4. J. Anzai, N. Matsuzaki, and T. Matsumoto, "Clone discovery," to appear.
5. S. Berkovits, "How to broadcast a secret," *Advances in Cryptology – EUROCRYPT '91*, LNCS vol. 547, pp. 535–541, Springer-Verlag, 1992.
6. C. Blundo, L. Mattos, and D. Stinson, "Generalized Beimel-Chor schemes for broadcast encryption and interactive key distribution," *Theoretical Computer Science*, 200(1-2), pp. 313–334, 1998.
7. C. Blundo, L. Mattos, and D. Stinson, "Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution," *Advances in Cryptology — CRYPTO '96*, LNCS vol. 1109, pp. 387–400, Springer-Verlag, 1996.
8. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and efficient constructions," *Proceedings of INFOCOM '99*, vol. 2, pp. 708–716, 1999.
9. R. Canetti, T. Malkin, and K. Nissim, "Efficient communication-storage tradeoffs for multicast encryption," *Advances in Cryptology — EUROCRYPT '99*, LNCS vol. 1592, pp. 459–474, Springer-Verlag, 1999.
10. B. Chor, A. Fiat, and M. Naor, "Tracing traitors," *Advances in Cryptology — CRYPTO '94*, LNCS vol. 839, pp. 257–270, Springer-Verlag, 1994.
11. R. Cramer, V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," *Advances in Cryptology — CRYPTO '98*, LNCS vol. 1462, pp. 13–25, Springer-Verlag, 1998.
12. Y. Desmedt, Y. Frankel, "Threshold cryptosystems," *Advances in Cryptology — CRYPTO '89*, LNCS vol. 435, pp. 307–315, Springer-Verlag, 1989.
13. A. Fiat, M. Naor, "Broadcast encryption," *Advances in Cryptology — CRYPTO '93*, LNCS vol. 773, pp. 480–491, Springer-Verlag, 1993.
14. P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," *Advances in Cryptology — CRYPTO '99*, LNCS vol. 1666, pp. 388–397, Springer-Verlag, 1999.
15. R. Kumar, S. Rajagopalan, and A. Sahai, "Coding constructions for blacklisting problems without computational assumptions," *Advances in Cryptology — CRYPTO '99*, LNCS vol. 1666, pp. 609–623, Springer-Verlag, 1999.
16. K. Kurosawa, Y. Desmedt, "Optimum traitor tracing and asymmetric scheme," *Advances in Cryptology — EUROCRYPT '98*, LNCS vol. 1403, pp. 145–157, Springer-Verlag, 1998.
17. M. Luby, J. Staddon, "Combinatorial bounds for broadcast encryption," *Advances in Cryptology — EUROCRYPT '98*, LNCS vol. 1403, pp. 512–526, Springer-Verlag, 1998.
18. M. Mambo, A. Nishikawa, S. Tsujii, and E. Okamoto, "Efficient secure broadcast communication systems," *Technical Report of IEICE*, ISEC93-34, pp. 21–31, 1993.
19. T. Matsushita, Y. Watanabe, K. Kobara, and H. Imai, "A sufficient content distribution scheme for mobile subscribers," *Proceedings of International Symposium on Information Theory and Its Applications, ISITA 2000*, pp. 497–500, 2000.

20. N. Matsuzaki, J. Anzai, "Secure group key distribution schemes with terminal revocation," *Proceedings of JWIS'98* (*IEICE Technical Report*, ISEC98-52), pp. 37–44, 1998.
21. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, pp. 113–114, 1997.
22. K. Nyberg, R.A. Rueppel, "Message recovery for signature schemes based on the discrete logarithm problem," *Advances in Cryptology — EUROCRYPT '94*, LNCS vol. 950, pp. 182–193, Springer-Verlag, 1995.
23. J.R. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely, "Partitioning attacks: or how to rapidly clone some GSM cards," *IEEE Symposium on Security and Privacy*, 2002.

24. A. Shamir, "How to share a secret," *Communications of ACM*, vol. 22, no. 11, pp. 612–613, 1979.
25. U. S. Dept. of Commerce/National Institute of Standards and Technology, "Digital signature standard," *Federal Information Processing Standards Publication 186-1*, 1998.
26. D. Wallner, E. Harder, and R. Agee, "Key management for multicast: issues and architectures," RFC2627, IETF, June 1999.
27. C. Wong, M. Gouda, and S. Lam, "Secure group communications using Key graphs," *Proceedings of ACM SIGCOMM '98*, 1998. Also available as Technical Report TR 97-23, Department of Computer Science, The University of Texas at Austin, July 1997.