ELSEVIER

Information Processing Letters 53 (1995) 49–54

Information Processing Letters

On key agreement protocols based on tamper-proof hardware

Yuliang Zheng [1]

The Centre for Computer Security Research, Department of Computer Science, The University of Wollongong, Wollongong, NSW 2522, Australia

Communicated by K. Ikeda; received 15 November 1993; revised 11 July 1994

Abstract

A key agreement (or distribution) protocol is a set of communication rules whereby two users can establish a shared common key. The shared key can be used by the users in future secure communications. We analyze a key agreement protocol presented by Leighton and Micali at the CRYPTO'93 conference, which is based on tamper-proof hardware, and show that the protocol fails in that a common key shared between two users can always be easily obtained by a number of legitimate users in a system where the proposed protocol is employed. An interesting point is that the legitimate users can derive the key without opening a single tamper-proof chip. We also propose a very simple identity based conference key agreement protocol that frees of the flaw possessed by Leighton and Micali's protocol. Furthermore, we employ ideas behind our protocol to successfully repair Leighton and Micali's failed protocol.

Keywords: Cryptography; Distributed systems; Key distribution protocols; Security in digital systems

1. Leighton and Micali's protocol

At the CRYPTO'93 conference, Leighton and Micali proposed two key agreement protocols [2], which were aimed at such communications scenarios as the one based on the Clipper Chip. The paper was further extended and appeared as [3]. The first protocol presented in [3] is new and does not appear in [2]. The second protocol in [3] is essentially the same as the first protocol in [2], while the third protocol in [3] represents an improvement to the second protocol in [2]. Hereafter the three protocols in [3] will be referred to as LM-1, LM-2 and LM-3 respectively.

While the focus of this paper is mainly on LM-2, it is worthwhile to make a few remarks on LM-1 and LM-3 as well. LM-1 is conceptually very simple. However, from this author's point of view, the protocol is not practical in terms of the number of secret keys that have to be kept by an individual user. We justify our view in the following. In LM-1, the number of secret keys, each k bits, for each individual user is between $O(B^2 \log N)$ and $O(B^3 \log N)$, where N is the total number of users and B is the maximum number of dishonest users in a system. Typically $k \geqslant 64$. Now suppose that LM-1 is employed in a country with ten million ($N \approx 2^{23}$) users among which a thousand ($B \approx 2^{10}$) are dishonest. Then the number of secret keys each user has to keep is at least 2^{24}, which is even worse than the naive solution in which each user keeps $N - 1$ secret keys.

[1] Email: yuliang@cs.uow.edu.au.

0020-0190/95/$09.50 © 1995 Elsevier Science B.V. All rights reserved
SSDI 0020-0190(94)00174-X

LM-3 is primarily a memoryless version of an authentication server based key agreement protocol, such as the (modified) Needham-Schroeder protocol. The secret key database of the authentication server is removed by a technique which has nowadays become a classic method for reducing memory, namely, the use of a cryptographically strong pseudo-random function. In practice, a cryptographically strong pseudo-random function is usually implemented by a secret key encryption algorithm, such as DES.

Now we turn our attention back to LM-2. This protocol relies on a tamper-proof VLSI chip that contains a CPU together with internal memory. It also assumes the existence of a trusted agent (or a group of agents at least one of which is trusted). The following is a brief description of the tamper-proof hardware based protocol [2].

The trusted agent has M secret keys $(X_1, \dots, X_M)$, each of which is k bits long and chosen uniformly at random by the agent, where k is a sufficiently large integer. When user i enrolls in the system, the agent selects M random integers $(\alpha_1, \dots, \alpha_M)$ from the interval $[1, L]$, where L is an integer. Leighton and Micali recommended the size of M be $O(B^3 \log N)$, where N is the total number of users and B is the maximum number of dishonest users in a system Next the agent calculates $Y_m = h^{\alpha_m}(X_m)$ for all $m = 1, \dots, M$. Here h is a cryptographically strong public one-way hash function, and $h^s(X)$ indicates applying consecutively the function h on an input X for s times, namely,

$$h^s(X) = \overbrace{h(\cdots h(h(}^{s \text{ times}} X)) \cdots).$$

Then the agent puts $(\alpha_1, \dots, \alpha_M)$ into the public key file, and with the absence of user i, injects $(Y_1, \dots, Y_M)$ into the tamper-proof chip of the user. Note that $(\alpha_1, \dots, \alpha_M)$ act as the public key of user i, while $(Y_1, \dots, Y_M)$ as the corresponding secret key. As the M numbers representing the secret key are stored in the tamper-proof chip, they are kept secret even from user i, the owner of the chip [3].

After the enrollment, user i can obtain the common key shared with another user j in the following way:

(1) retrieve user j's public key $(\beta_1, \dots, \beta_M)$ from the public key file.

(2) provide his tamper-proof chip with $(\beta_1, \dots, \beta_M)$. The chip outputs the following number as the common key between user i and user j:

$$K_{i,j} = h(h^{s_1}(Y_1) || \cdots || h^{s_M}(Y_M)), \tag{1}$$

where $s_m = 0$, if $\alpha_m \geqslant \beta_m$ and $s_m = \beta_m - \alpha_m$, otherwise $(m = 1, \dots, M)$, and $||$ denotes concatenation.

Note that

$$\begin{aligned} K_{i,j} &= h(h^{s_1}(Y_1) || \cdots || h^{s_M}(Y_M)) \\ &= h(h^{\delta_1}(X_1) || \cdots || h^{\delta_M}(X_M)) \end{aligned}$$

where $\delta_m = \max(\alpha_m, \beta_m)$, $m = 1, \dots, M$. This indicates that the common key calculation procedure is symmetric with respect to user i and user j. Hence we have $K_{i,j} = K_{j,i}$.

Using an asymptotic argument, the authors proved that if an adversary tries to obtain a common key between two users *by opening tamper-proof chips, completely or partially*, then the chance for him to succeed was so slim that it could be ignored in practical applications. This led them to conclude that the protocol was secure.

While the asymptotic argument might be appropriate for the situation where a persistent but narrow-minded adversary tries to crack the protocol *by compromising tamper-proof chips*, it does not exclude the possibility that the protocol might be vulnerable to other types of adversaries. That is, the asymptotic argument is not

[2] To be precise, LM-2 in fact has two versions. The first version does not use a one-way hash function while the second version does. Due to the fact that the number of secret keys for each user in the first version is larger than that in the second version, Leighton and Micali are clearly in favor of the second version described in this section.

[3] Clearly, like LM-1, LM-2 is impractical in terms of the large number $O(B^3 \log N)$ of secret keys each user has to keep, even if it were secure.

sufficient to conclude that the protocol is secure. Indeed, we will show in the following that the hardware based protocol LM-2 is easily breakable by far less sophisticated adversaries. In particular, we will show that the protocol fails in that a common key shared between two users is always clear to a number of legitimate users in a system that employs the protocol. In doing this the legitimate users need *not* to open a single tamper-proof chip!

2. Failure of the protocol

Note that the common key between user i and user j is largely determined by $(\delta_1,\dots,\delta_M)$ where $\delta_m = \max(\alpha_m,\beta_m)$, $m = 1,\dots,M$. To examine how the protocol fails, first we consider the case when $\alpha_m \leqslant \beta_m$ for all $1 \leqslant m \leqslant M$. Let $(\gamma_1,\dots,\gamma_M)$ be the public key of a third user k. Suppose that user k's public key satisfies $\gamma_m \leqslant \beta_m$ for all $1 \leqslant m \leqslant M$. Then we have $\max(\gamma_m,\beta_m) = \max(\alpha_m,\beta_m) = \beta_m$ for all $1 \leqslant m \leqslant M$. This implies that $K_{k,j} = K_{i,j}$ and that communications between user i and user j are clear to user k. A similar situation occurs when $\beta_m \leqslant \alpha_m$ and $\gamma_m \leqslant \alpha_m$ for all $1 \leqslant m \leqslant M$.

Example 1. As a small example, suppose that $L = M = 5$ and that users i, j and k have the following public keys:

User i: $(\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5) = (2,4,1,3,2)$
User j: $(\beta_1,\beta_2,\beta_3,\beta_4,\beta_5) = (5,4,2,3,4)$
User k: $(\gamma_1,\gamma_2,\gamma_3,\gamma_4,\gamma_5) = (3,4,1,2,3)$

Let (Y_1,Y_2,Y_3,Y_4,Y_5) be user i's secret key, where $Y_1 = h^2(X_1)$, $Y_2 = h^4(X_2)$, $Y_3 = h^1(X_3)$, $Y_4 = h^3(X_4)$ and $Y_5 = h^2(X_5)$. Then on input $(\beta_1,\beta_2,\beta_3,\beta_4,\beta_5)$, user i's tamper-proof chip outputs the following number as the common key between user i and user j:

$$\begin{aligned} K_{i,j} &= h(h^3(Y_1)||h^0(Y_2)||h^1(Y_3)||h^0(Y_4)||h^2(Y_5)) \\ &= h(h^5(X_1)||h^4(X_2)||h^2(X_3)||h^3(X_4)||h^4(X_5)). \end{aligned}$$

Now let (Z_1,Z_2,Z_3,Z_4,Z_5) be user k's secret key. Recall that $Z_1 = h^3(X_1)$, $Z_2 = h^4(X_2)$, $Z_3 = h^1(X_3)$, $Z_4 = h^2(X_4)$ and $Z_5 = h^3(X_5)$. Then the common key between user k and user j is

$$\begin{aligned} K_{k,j} &= h(h^2(Z_1)||h^0(Z_2)||h^1(Z_3)||h^1(Z_4)||h^1(Z_5)) \\ &= h(h^5(X_1)||h^4(X_2)||h^2(X_3)||h^3(X_4)||h^4(X_5)). \end{aligned}$$

Hence we have $K_{i,j} = K_{k,j}$, and all communications between user i and user j are clear to k. Symmetrically, all communications between user k and user j are also clear to user i.

The above observation can be explored further. Let $(\alpha_1,\dots,\alpha_M)$, $(\beta_1,\dots,\beta_M)$ and $(\gamma_1,\dots,\gamma_M)$ be users i, j and k's public keys respectively. Then the common key between user i and user j can be obtained by user k with the help of his tamper-proof chip if the following condition is satisfied:

$$\gamma_m \leqslant \max(\alpha_m,\beta_m), \quad \text{for all } 1 \leqslant m \leqslant M.$$

There is no need for user k to know of his secret key. All the user has to do is to feed his tamper-proof chip with the M numbers

$$(\max(\alpha_1,\beta_1),\dots,\max(\alpha_M,\beta_M))$$

as the public key of an existing or non-existing user x. The output of the chip is the common key between user k and user x, and identical to the common key between user i and user j.

Example 2. Suppose that users i, j and k have the following public keys:

$$\begin{array}{ll} \text{User } i\text{:} & (\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5) = (2,4,1,3,5) \\ \text{User } j\text{:} & (\beta_1,\beta_2,\beta_3,\beta_4,\beta_5) = (5,2,2,1,4) \\ \text{User } k\text{:} & (\gamma_1,\gamma_2,\gamma_3,\gamma_4,\gamma_5) = (4,3,1,2,5) \end{array}$$

Let (Y_1, Y_2, Y_3, Y_4, Y_5) be user i's secret key. Then the common key between user i and user j is

$$\begin{aligned} K_{i,j} &= h(h^3(Y_1)||h^0(Y_2)||h^1(Y_3)||h^0(Y_4)||h^0(Y_5)) \\ &= h(h^5(X_1)||h^4(X_2)||h^2(X_3)||h^3(X_4)||h^5(X_5)). \end{aligned}$$

Let $(Z_1, Z_2, Z_3, Z_4, Z_5)$ be user k's secret key. User k provides his tamper-proof chip with the numbers shown below:

$$(\max(\alpha_1,\beta_1),\ldots,\max(\alpha_5,\beta_5)) = (5,4,2,3,5).$$

Suppose that $(5,4,2,3,5)$ is the public key of user x. The chip returns the following value as the common between user k and user x:

$$\begin{aligned} K_{k,x} &= h(h^1(Z_1)||h^1(Z_2)||h^1(Z_3)||h^1(Z_4)||h^0(Z_5)) \\ &= h(h^5(X_1)||h^4(X_2)||h^2(X_3)||h^3(X_4)||h^5(X_5)). \end{aligned}$$

Hence we have $K_{k,x} = K_{i,j}$, and all communications between user i and user j are clear to user k.

The above observations can be generalized to the case where a group of agents are involved. Similar observations apply to the multi-level security scenario where the public key of a user at a level q, $1 \leqslant q \leqslant S$, is selected from the interval $[1 + (q-1)L, qL]$.

Note that the public key $(\alpha_1,\ldots,\alpha_M)$ of user i can be viewed as the user's (extended) identity. Thus in a sense LM-2 is an *identity based* key agreement scheme. The main reason for the failure of their protocol is that users' public keys (namely identities in our terms) are involved in the derivation of common keys in their *plain, un-scrambled* form. This allows a malicious user to successfully tap communications among other users by searching through the public key file. The same fact was responsible for the failure of many other identity based protocols proposed in the past decade. In some cases, applying a one-way function to a user's identity before its participation in the computation of common keys is an effective way to thwart the attack (see for instance [4,5,1]). This technique, however, seems not applicable to LM-2. In the next section we present a modification to Leighton and Micali's protocol. The modification is simple and it repairs the flaw in the protocol.

3. How to remove the flaw

Using ideas to be described in Section 4, we can amend LM-2 so that the resulting protocol does not have the flaw explained in the previous section. A technical assumption with the modification is that each user i has a unique identity ID_i and that all users agree upon a uniform encoding scheme for identities. Another assumption is that encoded identities are *prefix-free*, namely no identity is the prefix of another identity. A possible choice for such identities is international ISDN subscriber numbers.

Modification to the protocol is achieved by substituting Eq. (1) with

$$K_{i,j} = \begin{cases} h(h^{s_1}(Y_1)||\cdots||h^{s_M}(Y_M)||ID_i||ID_j), & ID_i < ID_j, \\ h(h^{s_1}(Y_1)||\cdots||h^{s_M}(Y_M)||ID_j||ID_i), & ID_i > ID_j. \end{cases} \tag{2}$$

With this modification, users' names (identities) are more directly involved in the generation of a common key. Due to the pseudo-randomness of the one-way hash function h, the probability that two different pairs of users are assigned an identical key is negligible. Thus the flaw possessed by the original protocol, namely $K_{i,j} = K_{k,x}$ for different users i, j, k and x, is removed.

4. A new identity based protocol

In this section we propose a new identity based key agreement protocol that can generate a common (conference) key for a group of two or more users. The protocol is based on the same assumptions as those employed by Leighton and Micali, namely

- the existence of a trusted agent,
- the availability of tamper-proof VLSI chips, and
- the availability of a one-way hash (or cryptographically strong pseudo-random) function h.

The agent selects a k-bit random number X, where k should be sufficiently large, say $k \geqslant 100$, in order to prevent it from exhaustive search attack. The agent keeps X as a secret. At the enrollment stage, the agent personalizes user i's tamper-proof chip simply by injecting into the chip the random number X and the user's identity ID_i. Note that the random number X is common to all users in the system. Also note that while X should never be seen by a user, the only requirement for the ID_i part is that it can not be altered once it is embedded in the chip.

Now user i can obtain the common key shared with user j by presenting user j's identity ID_j to his tamper-proof chip. The chip outputs the following number as a common key between the two users:

$$K_{i,j} = \begin{cases} h(X||ID_i||ID_j), & ID_i < ID_j, \\ h(X||ID_j||ID_i), & ID_i > ID_j. \end{cases} \tag{3}$$

Clearly the key generation procedure is symmetric with respect to user i and user j. Hence we have $K_{i,j} = K_{j,i}$.

To generate a common key for three users i, j and k, user i provides his tamper-proof chip with the other two users' identities ID_j and ID_k. The chip sorts the three identities (ID_i, ID_j, ID_k) according to the ascending order. Let (ID', ID'', ID''') be the re-arranged identity list. Then the common key among the three users is computed by

$$K_{i,j,k} = h(X||ID'||ID''||ID'''). \tag{4}$$

A common key for a group of four or more users is computed in a similar way. A generalization to the case of multiple agents is straightforward.

The security of the key agreement protocol relies on the trustworthiness of the agent(s), the tamper-resistance of the chips and the randomness of the one-way hash function. To reduce the risk of abusing stolen chips, authentication of a chip's owner should be conducted by such means as user passwords.

Comparing (3) with (2), we can see that the amended LM-2 can be viewed as a variant of the new protocol. Advantages of the new protocol over the amended LM-2 include

(1) it is orders of magnitude faster,
(2) it uses orders of magnitude less tamper-proof memory, and
(3) it does not need a public key file.

Since the cost of a tamper-proof chip is proportional to the amount of memory built in the chip, a chip for the new protocol can be orders of magnitude cheaper than that for the amended LM-2. Equivalently we can say that with the same cost, a chip for the new protocol can be made much more secure than that for the amended LM-2.

In conclusion, the new protocol represents a promising solution to the key agreement problem in terms of its computational efficiency, much less requirement on tamper-proof memory, low cost of implementation and flexibility in conference key generation.

Acknowledgments

This work was supported in part by the Australian Research Council (ARC) under the reference number A49232172, and the Australian Telecommunications and Electronics Research Board (ATERB) under the project number N069/412. The author would like to thank members of the Centre for Computer Security Research for their comments on early versions of the paper.

References

[1] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva and M. Yung, Systematic design of a family of attack-resistant authentication protocols, *IEEE J. Selected Areas Comm.* **11** (5) (1993) 679–693.

[2] T. Leighton and S. Micali, New approaches to secret-key exchange, Presented at Crypto'93.

[3] T. Leighton and S. Micali, Secret-key agreement without public-key cryptography (extended abstract), in: *Advances in Cryptology – CRYPTO'93,* Lecture Notes in Computer Science **773** (Springer, Berlin, 1994), 456–479.

[4] T. Matsumoto and H. Imai, On the key predistribution systems: A practical solution to the key distribution problem, in: *Advances in Cryptology – CRYPTO'87,* Lecture Notes in Computer Science **239** (Springer, Berlin, 1987) 185–193.

[5] S. Tsujii and J. Chao, A new ID-based key sharing system, in: J. Feigenbaum, ed., *Advances in Cryptology – CRYPTO'91,* Lecture Notes in Computer Science **567** (Springer, Berlin, 1992) 287–299.