Amending Leighton and Micali's Key Distribution Protocol *

Yuliang Zheng

The Centre for Computer Security Research
Department of Computer Science
The University of Wollongong
Wollongong, NSW 2522, AUSTRALIA
yuliang@cs.uow.edu.au

November 2, 1993

Abstract

A key distribution protocol is a set of rules by which two users can establish a shared common key between them and then use the key in future secure communications. We analyze a key distribution 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 obtained by a number of other 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 distribution 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.

1 Leighton and Micali's Protocol

At the CRYPTO'93 conference, Leighton and Micali proposed two key distribution protocols [1], which were aimed at such communications scenarios as the one based on the Clipper Chip. Both of the protocols assume the existence of a trusted agent (or a group of agents at least one of which is trusted). The first protocol also relies on a tamper-proof VLSI chip that contains a CPU together with internal memory. Using an asymtotic argument, the authors proved that if an adversary tried to obtain a common key between two users *by opening tamper-proof chips*, then the chance for him to succeed was slim that it could be ignored in practical applications. This led them to conclude that the protocol was secure.

While the asymtotic argument might be appropriate for the situation where an adversary armed with sophisticated machinery 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 asymtotic argument is not sufficient to conclude that the protocol is secure. Indeed, we will show in this paper that the hardware based protocol proposed by Leighton and Micali is easily breakable by much 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 other legitimate users in a system that employs the protocol. In doing this the legitimate users need *not* to open any tamper-proof chips !

The following is a brief description of the tamper-proof hardware based protocol. The trusted agent has M secret keys $(X_1, \ldots, X_M)$, each of which is of k-bits long and chosen uniformly at

*Preprint No. 93-17. This work was supported in part by the Australian Research Council under the reference number A49232172.

random by the agent. When user i enrolls in the system, the agent selects M random integers (α_1, ..., α_M) from the interval $[1, L]$, where L is an integer. Leighton and Micali recommended the size of M be between 10^2 and 10^5, and the size of L be about $\sqrt{M}$. Next the agent calculates $Y_m = h^{\alpha_m}(X_m)$ for all $m = 1, \ldots, M$. 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 (α_1, ..., α_M) into the public file, and with the absence of user i, injects (Y_1, ..., Y_M) into the tamper-proof chip of the user. Note that (α_1, ..., α_M) act as the public key of user i, while (Y_1, ..., 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.

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 (β_1, ..., β_M) from the public file;
2. provide his tamper-proof chip with (β_1, ..., β_M). The chip outputs the following number as the common key between user i and user j:

$$K_{i,j} = h(h^{\delta_1}(X_1)|| \cdots ||h^{\delta_M}(X_M)), \tag{1}$$

where $\delta_m = \max(\alpha_m, \beta_m)$, $m = 1, \ldots, M$, and $||$ denotes concatenation.

As the M numbers $Y_m = h^{\alpha_m}(X_m)$, $m = 1, \ldots, M$, are stored in the chip's internal memory, it is straightforward for the chip to compute $h^{\delta_m}(X_m) = h^{|\delta_m - \alpha_m|}(Y_m)$, $m = 1, \ldots, M$, and hence the key $K_{i,j}$. In addition, since the common key calculation procedure is symmetric with respect to user i and user j, we have $K_{i,j} = K_{j,i}$.

2 Failure of the Protocol

Note that the common key between user i and user j is largely determined by (δ_1, ..., δ_M) where $\delta_m = \max(\alpha_m, \beta_m)$, $m = 1, \ldots, M$. To examine how the protocol fails, first we consider the case when $\alpha_m \leqq \beta_m$ for all $1 \leqq m \leqq M$. Let (γ_1, ..., γ_M) be the public key of a third user k. Suppose that user k's public key satisfies $\gamma_m \leqq \beta_m$ for all $1 \leqq m \leqq M$. Then we have $\max(\gamma_m, \beta_m) = \max(\alpha_m, \beta_m) = \beta_m$ for all $1 \leqq m \leqq M$. 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 $\beta_m \leqq \alpha_m$ and $\gamma_m \leqq \alpha_m$ for all $1 \leqq m \leqq M$.

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

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

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 (β_1, β_2, β_3, β_4, β_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. Given three users i, j and k's public keys (α_1, ..., α_M), (β_1, ..., β_M) and (γ_1, ..., γ_M) respectively, the common key between user i and user j can be obtained by user k with the help of his tamper-proof chip if

$$\gamma_m \leqq \max(\alpha_m, \beta_m), \text{ for all } 1 \leqq m \leqq M. \tag{2}$$

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

$$(\max(\alpha_1, \beta_1), \ldots, \max(\alpha_M, \beta_M)) \tag{3}$$

as the public key of an existing or non-existing user. The output of the chip is identical to the common key between user i and user j. It is also identical to the common key between user k and the (existing or non-existing) user who public is represented by the M numbers ($\max(\alpha_1, \beta_1)$, ..., $\max(\alpha_M, \beta_M)$).

It should be noted

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

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

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. Then the common key between user k and user j is

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

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

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, $1 \leqq q \leqq S$, is selected from the interval $[1 + (q-1)L, qL]$.

Note that the public key (α_1, ..., α_M) of user i can be viewed as the user's (extended) identity. Thus in a sense Leighton and Micali's protocol is an *identity based* key distribution 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 [2]). This technique, however, seems not applicable to Leighton and Micali's protocol. 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 Leighton and Micali's protocol so that the resulting protocol does not have the flaw explained in the previous section. Modification to the protocol is achieved by substituting the equation (1) with

$$K_{i,j} = \begin{cases} h(h^{\delta_1}(X_1)||\cdots||h^{\delta_M}(X_M)||i||j), & i \leqq j, \\ h(h^{\delta_1}(X_1)||\cdots||h^{\delta_M}(X_M)||j||i), & i > j. \end{cases} \tag{4}$$

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. The flaw possessed by the original protocol, namely $K_{i,j} = K_{k,j}$ for three different users i, j and k, is removed when users' names are properly encoded (see Section 4 for detailed discussions).

The modified protocol can be extended to generate a common key for three or more users. For instance, a shared key for three users i, j and k can be defined as

$$K_{i,j,k} = h(h^{\delta_1}(X_1)||\cdots||h^{\delta_M}(X_M)||n_1||n_2||n_3), \tag{5}$$

where

$$\delta_m = \max(\alpha_m, \beta_m, \gamma_m), m = 1, \ldots, M,$$

n_1 is the smallest, n_2 is the second smallest and n_3 is the largest among i, j and k.

4 A New Identity Based Protocol

In this section we propose a new identity based key distribution 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.

We also assume that each user i has a unique identity and that all users agree upon a uniform encoding scheme for identities. Denote by ID_i the uniformly encoded identity of user i. To simplify our discussions, we further assume that encoded identities are *prefix-free*, namely no idenity is the prefix of another idenity. A possible choice for such identities is international ISDN numbers.

The agent selects a k-bit random number X, where k should be sufficiently large, say $k \geqq 100$, 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:

$$K_{i,j} = \begin{cases} h(X||ID_i||ID_j), & ID_i \leqq ID_j, \\ h(X||ID_j||ID_i), & ID_i > ID_j. \end{cases} \tag{6}$$

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

$$K_{i,j,k} = h(X||ID'||ID''||ID'''). \tag{7}$$

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 distribution 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 (6) with (4), we can see that the amended Leighton-Micali protocol can be viewed as a variant of the new protocol, and vice versa. Advantages of the new protocol over the amended Leighton-Micali protocol 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 Leighton-Micali protocol. 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 Leighton-Micali protocol.

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

References

[1] Leighton, T., and Micali, S. New approaches to secret-key exchange. In *Advances in Cryptology - Proceedings of Crypto'93,* Lecture Notes in Computer Science (1993), Springer-Verlag. (to appear).

[2] Matsumoto, T., and Imai, H. On the key predistribution systems: A practical solution to the key distribution problem. In *Advances in Cryptology - Proceedings of Crypto'87,* Lecture Notes in Computer Science, Vol. 293 (1988), Springer-Verlag, pp. 185–193.