On the Importance of Memory Resources in the Security of Key Exchange Protocols

(Extended Abstract)

George Davida Yvo Desmedt René Peralta*

Dept. EE & CS,
Univ. of Wisconsin – Milwaukee
P.O. Box 784
WI 53201 Milwaukee
U.S.A.

Abstract

We present a protocol for key exchange which relies on the existence of permutations which are not necessarily trap-door, and which are one-way in a weaker sense than that usually assumed in the literature. Our main result is that, under this assumption, two players can exchange a secret key over an open channel in such a way that an eavesdropper must spend time proportional to $TIME \cdot SPACE$, where $TIME$ is the time spent by the two players and $SPACE$ is the amount of information which can be stored and transmitted by the two players. Hence the importance of storage technology for security.

1 Introduction

It is not known whether or not one-way trap-door functions exist. Moreover, proving (from a complexity theory point of view) that these functions do exist implies proving $P \neq NP$, and therefore such a proof is not likely to be found in the near future. In fact, every year a number of researchers claim they have proven $P = NP$ (even though their proofs are invariably incorrect or incomprehensible). Given this state of affairs, it is reasonable to explore the possibility of solving the main cryptographic problems under weaker assumptions.

*Supported in part by NSF Grant Number CCR-8909657.

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 11-15, 1991.

In this paper we present a protocol for key exchange which relies on the existence of permutations (bijections) which are not necessarily trap-door, and which are one-way in a weaker sense than that usually assumed in the literature. Our main result is that, under this assumption, two players can exchange a secret key over an open channel in such a way that an eavesdropper must spend time proportional to $TIME \cdot SPACE$, where $TIME$ is the time spent by the two players and $SPACE$ is the amount of information which can be stored and transmitted by the two players. Hence the importance of storage technology for security. Using current optical-disk technology both for storage and transfer of information, we can think of $SPACE$ as being in the gigabytes range. Therefore, if the players are willing to invest one week in computation time each, then an eavesdropper will have to spend gigaweeks to obtain the secret. This scenario is reasonable, for example, in the case of embassies exchanging keys with their governments on a weekly basis.

Our protocol combines techniques appearing in [Mer78, DDP90] for key exchange without trap-door functions and uses Carter-Wegman universal hashing [CW79] to implement ideas similar to Hellman's time-memory tradeoff [Hel80]. The security achieved is similar to that of the protocols in [Mer78, DDP90] but our assumptions are weaker. In particular we do not assume, as is done in [DDP90], that (weakly) one-way functions exist which have arbitrarily low rates of encryption.

2 The assumptions

Let F_α be a family of bijections parametrized by α and with domain $\{1...K\}$. We suppose F_α is implemented by a specific circuit. In our protocol, players A and B will use F_α to exchange a secret key k over an open channel. Player E (the eavesdropper) will have access to the whole communication. Player E's goal is to compute k given A and B's communication. We make the following assumptions:

- $|\alpha| < \sqrt{K}$.
- the fastest algorithm to compute k given α and $F_\alpha(k)$ uses exhaustive search on a set of expected size $O(K)$.
- We assume the existence of an authenticated channel.
- We assume that E's technology is comparable to A and B's technology (E, however, may spend much more resources computing k than A and B do).

Note that the first and second assumptions do not imply that calculating k from $\alpha, F_\alpha(k)$ takes exponential time, since $|\alpha|$ itself is allowed to be exponential in $|k|$.

3 The protocol

In the following protocol, players A and B will agree on a common secret key $k \in \{1...K\}$. A set T of size h, is defined as follows:

- a random hashing function $H : \{1...K\} \rightarrow \{1...K/h\}$ is chosen from a *universal*$_2$ family of hashing functions (see [CW79] for the definition of universal hashing functions).
- we let $T = \{x \mid H(x) = 1\}$

The use of universal hashing is for the purpose of making T behave like a randomly chosen subset of the key space (a truly random subset cannot be described in polynomial time in $|h|$).

Let $F_\alpha^1(x) = F_\alpha(x)$ and $F_\alpha^i(x) = F_\alpha^{i-1}(F_\alpha(x))$ for $i > 1$. Given T, we define $G_{\alpha,T}(x) = F_\alpha^{u_x}(x)$ where u_x is the minimum positive integer such that $F_\alpha^{u_x}(x) \in T$. If no such integer exists, then $G_{\alpha,T}(x)$ is undefined. Note that if u_x is defined, then it has expected value $\leq K/h$, under the assumption that T is a truly random subset of K.

The protocol is as follows:

precomputation:

Step 1 Player A chooses α and H at random.

Step 2 Player A computes and stores $(x, G_{\alpha,T}(x), u_x)$ for n distinct randomly chosen $x \in \{1...K\}$.

Step 3 Player A sends α and a description of H to player B.

communication: Steps 4-5 are repeated until an agreement is achieved.

Step 4 Player B chooses h/n distinct random $z \in \{1...K\}$ and sends $(G_{\alpha,T}(z), u_z)$ to player A.

Step 5 Player A checks whether $G_{\alpha,T}(z) = G_{\alpha,T}(x)$ for some x in the table computed at Step 2 and some z sent at Step 4. If this is the case, then A sends u_x and $G_{\alpha,T}(z)$ to player B. The secret key is x if $u_x < u_z$ and z otherwise.

Note that if $u_x < u_z$ then B can calculate x by computing $F_\alpha^{u_z - u_x}(z)$. If $u_z < u_x$ then A can calculate z by computing $F_\alpha^{u_x - u_z}(x)$.

It is not hard to show that each iteration of this protocol has a chance of about $1 - e^{-1}$ of reaching agreement on a secret key. Alternatively, B may send, at Step 4, sufficiently many random $G_{\alpha,T}(x)$'s so that the probability of at least one $G_{\alpha,T}(x)$ being in A's table is exponentially high. This has the desirable effect of reducing the number of rounds in the protocol to, essentially, one.

4 Analysis

The security of the protocol follows from the fact that the key agreed on is randomly chosen from the key space (in this version of the protocol a slight deviation from the uniform distribution is caused by the fact that, in Step 5, x is favored over z

if $u_x < u_z$). The information available to the eavesdropper is, essentially, a pair $(i, F_\alpha^i(x))$ with $i > 1$. Recovering x from this information can be no easier than recovering x from $F_\alpha(x)$. By assumption, the fastest way to recover x from $F_\alpha(x)$ is by exhaustive search.

The costs of the protocol depend on the parameters K, n,and h. Let $ATIME$ and $BTIME$ be the cost of the protocol, in number of computations of F, to A and B respectively. We assume $ATIME \geq \sqrt{K} > |\alpha|$ so that we may ignore the time incurred in transmitting α.

Let $AMEM$ be the memory costs of A, in terms of triples $(x, G_{\alpha,T}(x), u_x)$ stored at Step 2. Let C be the communication cost of the protocol in terms of pairs $(G_{\alpha,T}(z), u_z)$ sent by B in Step 4.

Under the heuristic assumption that, given a random x, the sequence $\{F_\alpha^i(x)\}_i$ behaves (until it loops) as a random walk in the key space, it is easy to derive the following:

- $ATIME \approx \frac{nK}{h}$.
- $BTIME \approx \frac{K}{n}$.
- $AMEM \approx n$.
- $C \approx \frac{h}{n}$.

Let $\overline{ATIME}$ be the maximum value of $ATIME$ acceptable to player A. Similarly define $\overline{BTIME}$, $\overline{AMEM}$, and $\overline{C}$.

Thus, ignoring logarithmic and constant factors, we have the following constraints:

- $K \geq h \geq n$.
- $\overline{ATIME} \geq \frac{nK}{h}$.
- $\overline{BTIME} \geq \frac{K}{n}$.
- $\overline{AMEM} \geq n$.
- $\overline{C} \geq \frac{h}{n}$.

Since the security of the protocol is proportional to K, we must maximize K subject to these constraints. Under the assumption that $\overline{C} \leq \overline{BTIME}$ and $\overline{AMEM} \leq \overline{ATIME}$, the solution to this optimization problem is

$$n = \overline{AMEM}; h = \overline{AMEM} \cdot \overline{C}$$

and

$$K = min(\overline{BTIME} \cdot \overline{AMEM}, \overline{ATIME} \cdot \overline{C}).$$

Thus, the security of our protocol is proportional to

$$min(\overline{BTIME} \cdot \overline{AMEM}, \overline{ATIME} \cdot \overline{C}).$$

From this we can derive the impact of future technology on the security of this protocol. It turns out that faster chips do not help, since the effect of this is to increase both K and the eavesdropper's speed by the same factor. On the other hand, if both $\overline{C}$ and $\overline{AMEM}$ increase, then security increases by a proportional amount. This would be the effect of technology which increases the capacity of storage devices.

5 An open problem

We have assumed the existence of families of bijections F_α of a space of size N which require exhaustive search to invert. This assumption implies the existence of one-way-functions as usually defined in the literature, unless the size of the key α is large (i.e. more than polylogarithmic in N). To our knowledge, all bijections which have been proposed in the literature and which remain one-way after the key is made public have a key-size which is $O(\log N)$. On the other hand, if we could truly choose random permutations of a space of size N, then it would take $O(N \log N)$ bits to describe these permutations. The problem we propose is finding a family of permutations F_α on a space of size N such that it seems plausible that exhaustive search is the fastest way to invert F_α and α has length more than $poly(\log N)$. Note that the difficulty in achieving this is because of the condition that α is public. Otherwise, DES-like functions with the required property can be easily constructed.

6 Acknowledgement

Several useful comments from referees are gratefully acknowledged.

References

[CW79] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[DDP90] G. Davida, Y. Desmedt, and R. Peralta. A key distribution system based on any one-way function. In *Advances in Cryptology - proceedings of EUROCRYPT '89*, Lecture Notes in Computer Science. Springer-Verlag, 1990. to appear.

[Hel80] M. E. Hellman. A cryptanalytic time-memory tradeoff. *IEEE Tr. Inform. Theory*, 26(4):401–406, July 1980.

[Mer78] Ralph Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294 - 299, 1978.

[QD88] J.-J. Quisquater and J.-P. Delescaille. Other cycling tests for DES. In C. Pomerance, editor, *Advances in Cryptology, Proc. of Crypto '87 (Lecture Notes in Computer Science 293)*, pages 255–256. Springer-Verlag, 1988. Santa Barbara, California, U.S.A., August 16–20.