A Key Distribution System Based On Any One-Way Function

(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.

1 Introduction

Assuming the existence of one-way functions, we describe a simple protocol to exchange secret keys through an insecure (but authenticated) channel. If no pre-computation is allowed, our scheme uses $O(n)$ time for agreement on a number in the range $1..n^2$. An intruder takes time $O(n^2)$ to obtain the secret key. Thus, the number of steps necessary to cryptanalyze is the square of the number of steps in the protocol. If pre-computation is allowed to one of the parties in the key-exchange and also to the enemy, then this performance can be improved significantly. The assumptions necessary about the one-way-function are weaker than the assumptions in [Mer78] and in [DH76]. [1] The potential applications of our protocol also are more general than those of Merkle's protocol.

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:

- $|\alpha| < \sqrt{K}$.

[1] The set of functions usable in Merkle's protocol is a subset of the set of one-way trap-door functions. The set of functions usable in our protocol is a superset of the set of one-way functions.

J.J. Quisquater and J. Vandewalle (Eds.): Advances in Cryptology - EUROCRYPT '89, LNCS 434, pp. 75-79, 1990.

- 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).

Note that the first and second assumptions do not appear to imply that one-way functions exist.

3 The protocol

In the following protocol, players A and B will agree on a common secret key $k \in \{1...K\}$.

step 1 Player A chooses α at random.

step 2 Player A computes and stores $(r_i, F_\alpha(r_i))$ for n distinct randomly chosen $r_i \in \{1...K\}$.

step 3 Player A sends α to player B.

step 4 Player B chooses a random $k \in \{1...K\}$ and sends $F_\alpha(k)$ to player A.

step 5 Player A checks whether $k = r_j$ for some j by checking $F_\alpha(k)$ against the values computed at step 2. If this is the case, then A sends B a 1 (meaning k is the agreed-upon key). Otherwise A sends B a 0.

step 6 Steps 4-5 are repeated until an agreement is achieved.

Alternatively, B may send, at step 4, sufficiently many random $F_\alpha(k)$'s so that the probability of at least one $F_\alpha(k)$ being in A's table is high. In this case, A would tell B, at step 5, which $F_\alpha(k)$ was in its table.

4 Analysis

The running time of the protocol depends on n, the size of A's precomputed table. Assuming no memory or communication constraints for either player, the choice of n which minimizes the running time is $\sqrt{K}$. With this choice of n, the expected number of iterations of steps 4-5 is $\sqrt{K}$, since the experiment is a sequence of independent Bernoulli trials with $p = 1/\sqrt{K}$. The probability of no agreement after cn iterations of steps 4-5 is $(1 - 1/n)^{cn} \approx e^{-c}$. The eavesdropper will find the key, using exhaustive search, in (expected) $K/2$ steps. Therefore the resources needed to break the protocol are proportional to the square of the resources invested in the protocol.[2]

Note that the open channel is used *after* player A has computed a large table with known encryptions. If we assume that player A has more resources (e.g. time, memory, security) than player B, then we may choose n differently. For example, player A could compute a table of size $K^{3/4}$. Then, following the protocol, the open channel is used to agree on a key in expected number of messages equal to $O(K^{1/4})$. But the eavesdropper's time is still $O(K)$. Thus, the number of steps necessary to cryptanalyze would be the fourth power of the number of steps in the protocol (not counting precomputation by A).

In the next section we discuss the performance of our protocol under different constraints regarding channel capacity, time, and memory.

5 Choosing parameters under constraints

The optimal choice of parameters K and n depends on the rate R at which $F_\alpha(t)$ can be computed (in mappings per second). Since R can be varied (see, for example, [QDD86]), we will treat it as a parameter. Note that the expected number of messages in our protocol is K/n.

One constraint on the choice of parameters K, R and n is the maximum number C of $F_\alpha(r_i)$ messages which can be communicated. A second constraint is the maximum size M of A's table, in number of $(r_i, F_\alpha(r_i))$ pairs stored. A third constraint is given by a maximum time T, in seconds, allowed for the precomputation of A's table. The expected time necessary to obtain the secret key by exhaustive search is $K/(2 \cdot R)$. Thus, we must choose K, n, and R such that K/R is maximized subject to the constraining inequalities

$$\frac{K}{n} \leq C \quad ; \quad n \leq M \quad ; \quad \frac{n}{R} \leq T.$$

It is clear that one should choose $n = M$ and $K = C \cdot M$. This implies $K/R = C \cdot M/R$, and therefore R should be minimized subject to $n/R = M/R \leq T$. Thus, the optimal value of R is M/T. The time to find the secret key by exhaustive search is $K/(2 \cdot R) = C \cdot T/2$ seconds.

In order to see how the protocol performs in practice, we substitute typical values for T, M, and C.

Let

$$M = 10^8 \quad ; \quad T = 10^5 \quad ; \quad C = 10^6.$$

Then the optimal R is $M/T = 10^3$. Thus, a function or chip should be chosen such that $F_\alpha(t)$ can be computed in one millisecond (*and no faster*). There will

[2] we have chosen to ignore the fact that it takes slightly longer than n trials to generate n distinct random elements from a set of size $K > n$.

be $K = C \cdot M = 10^{14}$ possible keys. Precomputation time will be $M/R = 10^5$ seconds. The time to obtain the secret key by exhaustive search is $K/(2 \cdot R) = 10^{11}/2$ seconds (approximately 1.5 thousand years). If we allow $C = 10^7$ messages communicated, then a secret key is obtained which takes 15 thousand years to find by exhaustive search.

Thus, we have shown that the protocol can be used in practice.

5.1 The effect on security of increased resources

It is useful to analyze the protocol's security when resources available to all parties are increased by a factor ω. Let

$$M' = \omega \cdot M \quad ; \quad T' = \omega \cdot T \quad ; \quad C' = \omega \cdot C$$

be the new memory, precomputation time, and communication constraints. Suppose also that the eavesdropper has ω chips for the computation of F_α. That is, the enemy can compute many $F_\alpha(t)$ at a time, even though he cannot compute a particular $F_\alpha(t)$ any faster than A and B can. The reader can verify that, under the new constraints, the rate R remains the same but the size of the key space in increased by ω^2. Thus, the eavesdropper's time to find the secret key is increased by ω, even though he can now search ω possible keys at a time.

6 Conclusions and future research

There is a generalized feeling in the cryptographic community that modern cryptography strongly depends on assumptions about the asymptotic complexity of certain functions and their inverses. In particular, the fact that there is an odd possibility that $P = NP$ seems to make cryptographers very nervous. The usual definition of protocol security goes somewhat like this

- a protocol is secure if, when the legitimate parties use resources in an amount N, then the resources necessary to break the protocol is an exponential function of N.

We have exhibited a protocol which can be broken by an amount of resources which is only the square of the resources used by the protocol itself. The fact that this protocol seems secure suggests that the above definition of security may be stronger than necessary.

We would like to point out the following problems suggested by our research:

- Our protocol takes n steps to agree on a secret key which can be found by the eavesdropper in $O(n^2)$ steps. Is there a protocol which achieves security

$O(n^3)$ or $O(n^4)$, under the same assumptions? [3]

- Can the assumptions be weakened? In particular, can we trade keys without one-way functions?

- Given a chip which computes a mapping F at a rate R of mappings per second, it is possible to define a function which (apparently) can only be computed at a much slower rate. For example, we could simply define the function $G(t) = F^{(i)}(t)$, where i is an integer and $F^{(i)}$ is the composition of F with itself i times. Can we guarantee that G is not computable at a faster rate than R/i? If not, is there a provably secure way to decrease the rate R?

REFERENCES

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT–22(6):644–654, November 1976.

[IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. *Proceedings of the 21th Annual ACM Symposium on the Theory of Computing*, pages 44–61, 1989.

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

[QDD86] J.-J. Quisquater, Y. Desmedt, and M. Davio. The importance of 'good' key scheduling schemes (how to make a secure DES scheme with ≤ 48 bit keys?). In Hugh C. Williams, editor, *Advances in Cryptology. Proc. of Crypto'85 (Lecture Notes in Computer Science 218)*, pages 537–542. Springer–Verlag, 1986. Santa Barbara, California, U.S.A., August 18–20.

[3] This, however, may prove to be a very hard problem, as suggested by recent research by Impagliazzo and Rudich [IR89]