Zero-Knowledge Authentication Scheme with Secret Key Exchange (extended abstract)

Jørgen Brandt, Ivan Damgård, Peter Landrock, Torben Pedersen

University of Aarhus,
Department of Mathematics and Computer Science.

Abstract

In this note we first develop a new computationally zero-knowledge interactive proof system of knowledge, which then is modified into an authentication scheme with secret key exchange for subsequent conventional encryption. Implemented on a standard 32-bit chip or similar, the whole protocol, which involves mutual identification of two users, exchange of a random common secret key and verification of certificates for the public keys (RSA, 512 bits) takes less than 0.7 seconds.

1 Introduction

Recently, some very effective zero-knowledge identification-schemes have been constructed, such as Fiat-Shamir, [FS], Micali-Shamir, [MS], and Guillou-Quisquater, [GQ]. However, they only provide authentication, not confidentiality. Consequently, if one aims at implementing a hybrid system based on public keys to provide user authentication and secret key exchange, these fast schemes are of no help.

The protocol, we are going to describe, is designed for software implementation on a standard chip such as Intel 80386 or Motorola 68030 or perhaps on a DSP-chip with the extra requirement that the whole communication setup is based on 512 bits RSA-modulus and takes less than 2 seconds.

2 The Basic Authentication Scheme

We first describe an example of a more general construction of a computationally zero-knowledge interactive proof system of knowledge based on a public key system.

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO '88, LNCS 403, pp. 583-588, 1990.

In the example we use RSA as the public key system, but the method would work in general under some reasonably weak assumptions.

For proof technical reasons we need two independent general public key pairs (one-way functions with trapdoor) $\tilde{n} = \tilde{p}\tilde{q}$, $\tilde{e}$ a public exponent prime to $\phi(\tilde{n})$, and $\bar{n} = \bar{p}\bar{q}$, $\bar{e}$ a public exponent prime to $\phi(\bar{n})$. $(\tilde{n}, \tilde{e})$ and $(\bar{n}, \bar{e})$ can be generated when the system is set up. Then $\tilde{p}, \tilde{q}, \bar{p}$ and $\bar{q}$ can be deleted and are therefore assumed unknown to all users. Thus the prover, P and the verifier, V, receive as auxiliary input $\tilde{n}$ and $\bar{n}$, which they must trust to be generated correctly. This is similar to the non-interactive zero-knowledge model where P and V share a bit-string which they must trust to be random [BFM].

In the final version of this paper, we remove the need for $\tilde{n}$. This requires a more complicated argument to prove the zero-knowledge property, but simplifies the protocol (see Section 4).

Moreover each prover computes his own public-key pair $n = pq$, e public exponent, d secret ($ed \equiv 1 \bmod \phi(n)$). The "knowledge" he is going to prove to possess is that he can provide arbitrary signatures $m^d \bmod n$. For convenience, we assume that $k = \log n = \log \tilde{n} = \log \bar{n}$.

Protocol:

1. V chooses arbitrary bit-strings Q, R of length $\frac{1}{2}\log n$, subject to $(Q \parallel R) < n$, where $(Q \parallel R)$ is the number represented by the concatenation, R in the least significant half.
   He also chooses S, S' of length $\frac{1}{2}\log \tilde{n}$, subject to $(S \parallel R), (S' \parallel Q) < \tilde{n}$. He computes $(Q \parallel R)^e \bmod n$, $(S \parallel R)^{\tilde{e}} \bmod \tilde{n}$ and $(S' \parallel Q)^{\tilde{e}} \bmod \tilde{n}$ and sends these numbers to P.

2. P recovers $(Q \parallel R)$, chooses T, T', arbitrary bit strings of length $\frac{1}{2}\log n$, subject to $(T \parallel R), (T' \parallel Q) < \bar{n}$, computes $(T \parallel R)^{\bar{e}} \bmod \bar{n}$ and $(T' \parallel Q)^{\bar{e}} \bmod \bar{n}$ and sends these numbers to V.

3. V sends S, S' to P.

4. Using what he received in step 2) and 3), P checks the values for $(S \parallel R)^{\tilde{e}} \bmod \tilde{n}$ and $(S' \parallel Q)^{\tilde{e}} \bmod \tilde{n}$ that he received in step 1).
   If OK, he sends T, T' to V. Otherwise, he halts.

5. V uses T, T' to check the value for $(T \parallel R)^{\bar{e}} \bmod \bar{n}$ and $(T' \parallel Q)^{\bar{e}} \bmod \bar{n}$ he received in step 2).
   If OK, he accepts. Otherwise he rejects.

We now informally sketch how to prove the correctness and security of this protocol. We will say that $f(x)$ is pseudorandom given $g(x)$ (where $x \in \{0,1\}^k$) if no polynomial time probabilistic algorithm can distinguish between pairs of the form $(g(x), f(x))$ and $(g(x), r)$, where r is chosen uniformly from $Im(f)$.

It has been shown earlier, [ACGS], that the least significant $\log k$ bits of x are pseudorandom given $x^e \bmod n$. Following Micali and Schnorr [MS] we stretch this a little to get

Assumption*:
The least significant $\frac{1}{2}k$ bits of x are pseudorandom given $x^e \bmod n$ if the factors of n are unknown.

When we speak of a proof system of knowledge, we mean that the following must hold for an arbitrary prover P^*: if P^* completes the protocol successfully with non-negligible probability, then there exists a probabilistic polynomial time algorithm, which with the help of P^* finds x from $x^e \bmod n$ with non-negligible probability.

This is a little twist of the notion defined by Fiat, Feige and Shamir in [FFS]: a prover does not prove his possession of a certain bit-string, but his ability to do something.

Theorem 1.
Under * the protocol is a zero-knowledge proof system of knowledge.

Proof:
Zero-knowledge:
Given an arbitrary verifier, V^*, we describe a simulator, M_{V^*}. We may assume that the algorithm used to generate $\tilde{n}$ is public. Therefore M_{V^*} can generate $\tilde{n}$ with known factorization and correct distribution and give this number to V^* as input. We may now proceed as follows:

1. Receive $(Q \parallel R)^e \bmod n$, $(S \parallel R)^{\tilde{e}} \bmod \tilde{n}$ and $(S' \parallel Q)^{\tilde{e}} \bmod \tilde{n}$ ¿from V^*.
2. Recover $(Q \parallel R)$, and check against $(Q \parallel R)^e \bmod n$. If OK, compute and send $(T \parallel R)^{\tilde{e}} \bmod \tilde{n}$ and $(T' \parallel Q)^{\tilde{e}} \bmod \tilde{n}$ as P would have done. Otherwise send $X^{\tilde{e}} \bmod \tilde{n}$, $Y^{\tilde{e}} \bmod \tilde{n}$ for random X, Y.

Now observe:

- If V^* sends correct messages in step 1), we can complete the protocol with exactly the right distribution of messages.
- If this is not the case, P would always stop in step 3), and the random $X^{\tilde{e}} \bmod \tilde{n}$, $Y^{\tilde{e}} \bmod \tilde{n}$ cannot be distinguished from the "real" $(T \parallel R)^{\tilde{e}} \bmod \tilde{n}$, $(T' \parallel Q)^{\tilde{e}} \bmod \tilde{n}$ - by *.

¿From this, it is clear that the simulation works.

Proof-system:
Completeness is obvious. As for soundness assume some P^* completes the protocol

with non-negligible probability. We now describe an algorithm M_{P^*} which will find x from $x^e \bmod n$ with non-negligible probability.
Choose $\bar{n}$ with known factorization and give this as input to P^*. Suppose we are given $x^e \bmod n$, x unknown. Then choose $y, z < \tilde{n}$ and send $x^e \bmod n$, $y^{\bar{e}} \bmod \tilde{n}$ and $z^{\bar{e}} \bmod \tilde{n}$ to P^*.
Receive $(T \parallel R)^{\bar{e}} \bmod \bar{n}$ and $(T' \parallel Q)^{\bar{e}} \bmod \bar{n}$ ¿from P^*, recover $(Q \parallel R) = x$.
By *, P^* cannot distinguish the random $y^{\bar{e}} \bmod \tilde{n}$, $z^{\bar{e}} \bmod \tilde{n}$ ¿from the real $(S \parallel R)^{\bar{e}} \bmod \tilde{n}$ and $(S' \parallel Q)^{\bar{e}} \bmod \tilde{n}$, hence the recovered $(Q \parallel R)$ will be correct with essentially the same (i.e. non-negligible) probability as in an actual conversation with V. □

In a formal proof of this fact, what we get is a result saying that P^* finds x with the same probability when talking to V as when talking to M_{P^*} or the combined system (P^*, M_{P^*}) can invert $y^{\bar{e}} \bmod \tilde{n}$. But this is an empty statement if M_{P^*} already knows $(\tilde{p}, \tilde{q})$. This is why we cannot use $\bar{n} = \tilde{n}$.

3 Combining with Key-Exchange

For this, we can use exactly the same protocol as before, except that V will choose first $K < n$ and compute $(Q \parallel R) \equiv K^e \bmod n$. The protocol runs as before, but P can recover K as $K \equiv ((Q \parallel R)^e)^{d^2} \bmod n$. We can now use the least significant $\frac{1}{2}k$ bits of K as secret key.

Theorem 2.
The secret key exchanged as above is pseudorandom given the conversation between P and V, assuming *.

Proof:
By *, the key is pseudorandom given $(Q \parallel R)$, and given $(Q \parallel R)$, the rest of the conversation can be simulated exactly. □

4 Practical Implementation

As mentioned earlier, it is possible to redesign the protocol so that it does not use $\tilde{n}$. The resulting version is given below. By a somewhat complicated argument, given in the final version of this paper, also this version can be proved to be a zero-knowledge proof system. It is an open problem, whether the protocol remains correct if we use the tempting simplification of putting $\bar{n} = n$.

Our protocol is intended for use in a public network, where users wish to be convinced about each others identities before communicating secretly. Part of this goal is often achieved by letting some center, C, provide signatures (certificates) on

public keys. It would be natural to let C provide $\bar{n}$ too. This does not constitute any serious problems: as mentioned, the factorization of $\bar{n}$ can simply be forgotten after setting up the system.

Thus the practical version of the protocol will be as follows:

1. V chooses $K < n$ randomly and computes Q and R as $(Q \parallel R) \equiv K^e \bmod n$. He then computes $(Q \parallel R)^e \bmod n$ and sends this number to P.
2. P recovers K, Q and R. As before P chooses T, T', computes $(T \parallel R)^{\bar{e}} \bmod \bar{n}$ and $(T' \parallel Q)^{\bar{e}} \bmod \bar{n}$ and sends these numbers to V.
3. V sends Q, R to P.
4. P checks the values for Q and R.
   If OK, he sends T, T' to V. Otherwise, he halts.
5. V checks the value for $(T \parallel R)^{\bar{e}} \bmod \bar{n}$ and $(T' \parallel Q)^{\bar{e}} \bmod \bar{n}$ he received in step 2).
   If OK, he accepts. Otherwise he rejects.

For efficiency, it is advisable to choose e and $\bar{e}$ small, i.e. ≈ 3. Then the only really time consuming part is finding $x^{d^2} \bmod n$ from x. Note that the version with key exchange does not require more time, if e is small. P can precompute $d^2 \bmod \phi(n)$, and the protocol will then only require 2 extra exponentiations to the e'th power, which is negligible. With a little care, it is possible to use $e = 2$, which will be even more efficient.

Finally note, that A can prove himself to B while B is proving himself to A. In particular we can ensure that "decryption" of $(Q \parallel R)^e$ takes place simultaneosly for A and B. Thus this will not take more time than the basic version of the protocol.

It will be natural to obtain the final key as the xor of the two produced keys, since this will ensure that the key is known to precisely A and B, and no one else.

The basic operations needed for this protocol have been implemented on a standard 16 MHz Intel 80386 processor. The results of this show that the whole protocol, including verification of public-key certificates can be completed in less than 0.7 seconds.

References

[ACGS] Alexi,W., Chor, B., Goldreich, O. and Schnorr, C.P.: "RSA and Rabin Functions: Certain Parts Are as Hard as the Whole". Proc. of the 25th FOCS, 1984, pp. 449-457.

[BFM] Blum, M., Feldman, P. and Micali, S.: "Proving Security Against Chosen Cyphertext Attack". These proceedings.

[FFS] Feige, U., Fiat, A. and Shamir, A.: "Zero Knowledge Proofs of Identity". Proc. of the 19th STOC, 1987, pp. 210-217.

[FS] Fiat, A. and Shamir, A.: "How to Prove Yourself: Practical Solution to Identification and Signature Problems". Advances in Cryptology - CRYPTO'86, Lecture Notes in Computer Science 263, 1987, pp. 186-199.

[GQ] Guillou, L. and Quisquater, J-J: "A "Paradoxical" Identity-Based Signature Scheme Resulting from Zero-Knowledge". These proceedings.

[MS] Micali, S. and Shamir, A.: "An Improvement of the Fiat-Shamir Identification and Signature Scheme". These proceedings.