An Improvement of the Fiat-Shamir Identification and Signature Scheme

by

Silvio Micali

Lab. for Computer Science
MIT
USA

Adi Shamir

Applied Mathematics Dept.
The Weizmann Institute
Israel

Abstract

In 1986 Fiat and Shamir exhibited zero-knowledge based identification and digital signature schemes which require only 10 to 30 modular multiplications per party. In this paper we describe an improvement of this scheme which reduces the verifier's complexity to less than 2 modular multiplications and leaves the prover's complexity unchanged.

The new variant is particularly useful when a central computer has to verify in real time signed messages from thousands of remote terminals, or when the same signature has to be repeatedly verified.

1. Introduction.

Informally speaking, a digital signature is a value associated with a message which is easy to verify but difficult to forge. After having generated and verified it, the signature can be later presented to a judge since the signer cannot disown his messages. An identification scheme is a simplified signature scheme in which there are no messages disputes or judges: the proof of identity is interactive, and the verifier can either accept or reject the prover's claimed identity, with no legal or long-term consequences. To be useful and secure, the identification scheme should satisfy the following three conditions:

1) A real verifier should accept a real prover's proof of identity with overwhelming probability.
2) A real verifier should accept a cheating prover's proof of identity with negligible probability.
3) A cheating verifier should not learn anything from polynomially many interactions with a real prover that will enable him to misrepresent himself as the prover to someone else with non-negligible probability.

The best known example of a signature scheme is the RSA (Rivest, Shamir and Adleman[1978]). To use it as an identification scheme, the verifier can simply ask the prover to sign a random test message. The original scheme requires about 750 modular multiplications per party, but the verifier's complexity can be reduced to a few modular multiplications by using a low-exponent variant. A 512 bit implementation of the RSA scheme requires 10-15 seconds on IBM PC's, and several minutes on smart cards.

A faster and provably secure identification and signature scheme was proposed in Fiat and Shamir[1986]. It is based on the zero knowledge paradigm introduced in Goldwasser Micali and Rackoff[1985], and more particularly on the quadratic residuosity protocol presented by Fischer Micali and Rackoff at Eurocrypt 84. The Fiat-Shamir protocol reduces the time and communication complexities of the Fischer-Micali-Rackoff protocol by simultaneously proving the quadratic residuosity of many numbers, but by doing so it destroys the zero knowledge nature of the protocol. (The formal proof of security of the Fiat-Shamir protocol is thus based on the fact

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO '88, LNCS 403, pp. 244-247, 1990.

that it reveals no "transferable knowledge", which is a new measure of cryptographic strength introduced and studied in Feige, Fiat and Shamir[1987].)

In this paper we show how to substantially speed up the Fiat and Shamir scheme. There are many variants of this scheme. Our ideas speed up each single one of them. Thus below we confine ourselves to recall and speed up its simplest version.

2. The original Fiat-Shamir Scheme

Let s be a security parameter and let n be the product of two random prime numbers whose size is s. (Unlike the RSA scheme, it is not necessary to know the factorization of n in order to execute the protocol, and thus each prover can pick his own public modulus n, or use a universal modulus n published by a trusted center.)

Each prover picks a secret key consisting of k random numbers $s_1, \ldots, s_k$ in Z_n^* (the multiplicative group mod n), computes $v_j = 1/s_j^2$ (mod n) for j= 1,...,k, and publishes $v_1,\ldots,v_k$ (along with n, if it was chosen by him) in a public key directory.

The identification scheme is based on the following protocol:

1) The prover picks a random r in Z_n^*, and sends $x = r^2$ (mod n) to the verifier.

2) The verifier sends k random bits $e_1,\ldots,e_k$ to the prover.

3) The prover sends $y = r\prod_j s_j^{e_j}$ (mod n) to the verifier.

4) The verifier accepts the proof iff $x = y^2 \prod_j v_j^{e_j}$ (mod n).

In practice, we would accept the probability of successful misrepresentation to be at most 1 in a million per each attempt and thus a choice of k= 20 suffices for most applications. The key size (either public or private) in 512-bit implementations is about 1.3 kilobytes, and the average number of modular multiplications per party is about 10. The communication complexity is about 1000 bits per proof, but this can be almost halfed by sending a hashed version of x to the verifier. Other optimizations and tradeoffs can be found in Fiat and Shamir[1986].

To turn this interactive identification scheme into a non interactive signature scheme, it suffices to make $e = e_1, \ldots, e_k$ to be the value of a pseudo-random function f, easy to evaluate, but hard to invert, at input (x,m), where m is the message to be signed. This pseudo random function f is universal, and its values are accessible to all the parties. The resultant signature generation protocol is:

1) Choose at random r in [0,...,n).

2) Compute $e = f(r^2(mod\ n)\ ,\ m)$ and $y = r \cdot \prod_j s_j^{e_i}$.

3) Send e and y as the signature of m.

The corresponding signature verification scheme is:

Accept the signature if *syntax error file -, between lines 234 and 234* e = f ($y^2\Pi$.

The scheme is provably secure when f is a truly random function (computed by a trusted call-up center) or when f is a strong pseudo-random function in the sense of Goldreich, Goldwasser and Micali [GGM] given to the parties in tamper-proof devices: unless factoring is easy, a cheater cannot forge the signature of a new message with non-negligible probability even after he was given polynomially many signatures of other messages and polynomially many values of f at arguments of his choice. This sketched (but formalizable) proof breaks down for technical reasons when the parties are given access to the algorithm of f (and not just to its values). However, we strongly believe that the scheme remains secure even in this case, provided that f does not interact badly with the modular multiplication operations.

Since a cheater can know in advance whether a proposed signature is valid, the value of k in practical implementations should be at least 64. This increases the key size to about 4 kilobytes, and increases the number of modular multiplications to about 32 per party. The size of a signature is 576 bits, about the same as in the RSA scheme.

3. The New Improvement

Our improvement comes about from choosing the v_i's to be the first k prime numbers ($v_1=2, v_2=3, v_3=5$, etc). The s_i's will then be set to be a random square root of the corresponding v_i mod n. Each prover should choose his own modulus n and use its factorization in order to extract these roots. (The factorization is now no longer needed and it can be erased.) The actual proofs of identity and signatures are generated and verified in the standard way described in the previous section.

Newly arising difficulties

Before analyzing the efficiency of this scheme, it should be noticed that we have to overcome some technical difficulties. In fact, not all of the v_i's will be quadratic residues mod n. We overcome this technical difficulty with an appropriate perturbation technique which will be described in the full version of the paper.

Gain in efficiency

The above additional difficulties are worth dealing with. Since our choice of the v_j's is universal, provers should only publish n as their public key. This reduces the size of the public key directory to 64 bytes per user, and makes it possible to use the same directory in order to verify our new signatures, as well as other signatures based on factoring, like the previous Fiat-Shamir, the RSA and the Rabin's scheme. The size of the secret key remains about 4 kilobytes, but this size is less critical since the information is stored LOCALLY rather than TRANSMITTED, and each user keeps only one such file.

The main benefit of our improvement, though, is the GREATLY reduced complexity of verification: since most of the v_j's are single-byte numbers, their product is particularly easy to compute as does not even require modular reductions! The only expensive operation left is the modular squaring of y, and thus the total complexity of verification is somewhere between 1 and 2 modular multiplications.

Security

The security of the original Fiat-Shamir scheme is based on the fact that the extraction of square roots of random vj values is as difficult as the factorization of the modulus. This proof technique is not directly applicable to the new version, since the extraction of square roots of small primes may concievably be easier than the extraction of square roots of random numbers.

For simplicitly sake, let us discuss only the security of the identification scheme (the signature scheme only needs a more complex notation). The identification scheme in question is based on on zero-knowledge proofs. Very roughly (see Feige, Fiat and Shamir for a detailed discussion) this means that the proof of identity is constituted by a proof of "knowledge of something." n our case this "something" is not a proof of quadratic residuosity (either for a particular prime or for all the primes) in the original language-theoretic sense of Goldwasser Micali and Rackoff: Since the parties execute only one round of the protocol, the prover can succeed with probability 1/2 even if all the primes are quadratic non-residues! Similarly, the protocol is not a proof of knowledge of square roots (either for a particular prime or for all the primes) as in Feige Fiat and Shamir: The knowledge tape could contain the square roots of all the 400 pairwise products of the primes, and thus a cheating prover could convince the verifier with probability 1/2 without actually knowing even one of the original roots.

A CAREFUL analysis, carefully omitted in this abstract!, shows that, in our scheme, this "something" is the square root of the product of a subset times the inverse of another subset of the first 20 primes. We thus need to argue that this piece of knowledge is not easily available to everyone, and thus distinguishes the prover from everyone else. We already know that extracting square roots modulo composite numbers is as hard as integer factorization. This implies the following fact:

> Assume there exists an algorithm A that, on input m (a s-bit long modulus) and S (a random set of quadratic residues mod m whose cardinality is k), finds a square root of the product of a subset of S and the inverse of another subset of S in time T(s). Then, there exists a factoring algorithm A' that runs essentially in time $2^k \cdot T(s)$.

The proof of this fact, though not hard, is also postponed to the final paper. One would be tempted to conclude that if the "piece of knowledge" underlying our new scheme were computable in time T(s), then one could factor in one million T(s) steps an s-bit modulus, which would imply, as we need, that T(s) is large. This is, however, a too hasty conclusion. In fact, we can assume without loss of generality that the first 20 primes are quadratic residues (since our true scheme cops with those which are not to squares mod n), but they are NOT a random subset of size 20. Thus a natural question arises: is the computational difficulty of extracting square roots of small primes any lower than for random (quadratic) residues? The answer apperas to be negative. In fact, Morrison-Brillhart type methods would be substantially sped up if square root of small primes were easier to compute! This and other details (including a formal intractability assumption) needed to transform this discussion into a proof will be given in the final paper.

Let us mention that there is a more direct way to prove the security of our scheme if one is willing to make an intractability assumption that is stronger than the one derivable by formalizing the above argument. Informally, this stronger assumption states that factoring remains difficult even when one is given the square root of a small number of small primes.

It is worth mentioning that while such an assumption is sufficient to prove the security of the new scheme, its being false DOES NOT imply that our scheme is insecure! In fact, even if a cheating verifier knows how to factor n by using the square roots of a small number of small primes, he is unlikely to get hold of these square roots since the schemes are the parallel versions of zero knowledge protocols. In other words, only the real prover is likely to benefit from such a number-theoretic breakthrough, but he already knows this factorization!

Remark

This improvement of the Fiat-Shamir scheme was discovered independently by the two authors. Additional optimization ideas will be described in the full version of this paper.