An alternative to the Fiat-Shamir protocol *

Jacques Stern
Équipe de Logique
Université Paris 7
et
Département de mathématiques et informatique
École Normale Supérieure

Abstract

In 1986, Fiat and Shamir [2] exhibited zero-knowledge based identification and digital signature schemes. In these schemes, as well as in subsequent variants, both the prover and the verifier have to perform modular multiplications. This paper is an attempt to build identification protocols that use only very basic operations such as multiplication by a fixed matrix over the two-element field. Such a matrix can be viewed as the parity-check matrix of a linear binary error-correcting code. The idea of using error-correcting codes in this area is due to Harari [3] but the method that is described here is both simpler and more secure than his original design.

1 The signature scheme

The proposed scheme uses a fixed $(n\text{-}k)$-matrix G over the two-element field. This matrix is common to all users and is originally built randomly. Thus, considered as a parity-check matrix, it should provide a linear binary code with a good correcting power. Also common to all users is a family $w_1, \ldots, w_q$ of words with n bits.

Any user chooses a secret key s which is an n-bit word with a prescribed number p of 1's. This prescribed number p is also part of the system. Then he computes his public identification as

$$i = G(s)$$

*Research supported by the PRC mathématiques et informatique

J.J. Quisquater and J. Vandewalle (Eds.): Advances in Cryptology - EUROCRYPT '89, LNCS 434, pp. 173-180, 1990.

The sequence w_j will be essentially used as a pseudo-random number generator. Whenever an n-bit word z is given, together with two random integers a, c, with a prime to q, a sequence

$$z_j = z \oplus w_{aj+c} \quad 1 \leq j \leq q$$

is produced.

The identification scheme will rely heavily on the technical notion of a *commitment*. If u is an n-bit word, a *commitment* for u is a sequence

$$G(u_1), \ldots, G(u_l)$$

built from

$$u_1, \ldots, u_l$$

such that

$$\begin{cases} u_i \odot u_j = 0 & 1 \leq i < j \leq l \\ \bigoplus_{i=1}^{l} u_i = u \end{cases}$$

where $\odot$ and $\oplus$ respectively denote bitwise multiplication and bitwise addition modulo 2. Usually, a commitment will be built by choosing randomly a partition of $\{1, \ldots, n\}$ into l pieces. The notion of commitment can be extended to words of length $>$ n. By padding randomly, we may restrict ourselves to words whose length is a multiple of n and break these into a sequence of words of length exactly n. Especially, a sequence of integers $n_1, \ldots, n_q$ coded on a fixed number h of bits can be written as a single word of length hq and be given a commitment.

A commitment will be used as a one-way function: in order to disclose it, one announces the sequence

$$u_1, \ldots, u_l$$

from which it was built. Once this is done, anyone can check the correctness of the commitment by applying G to the sequence u_j, recover the original word u and use the information it encodes.

We now describe the interactive protocol that enables any user (which we will call the prover) to identify himself to another one (which we call the verifier). The protocol includes r rounds, each of these being performed as follows.

1. The prover picks a random n-bit word y and sends commitments for y and $y \oplus s$ to the verifier.

2. The verifier computes $x = G(y)$ and $x' = G(y \oplus s)$ and checks that $x = x' \oplus i$. Then, he sends a random n-bit word z to the prover.

3. The prover computes the sequences

$$n_j = \mid y \oplus z_j \mid$$
$$m_j = \mid y \oplus z_j \oplus s \mid$$

where $\mid v \mid$ is the weight of an n-bit word v, that is the number of its ones and where the sequence z_j is produced, as explained above, from integers a and c randomly chosen by the prover. He sends commitments of the sequences $n_j, 1 \leq j \leq q$ and $m_j, 1 \leq j \leq q$, to the verifier.

4. The verifier sends a random element b of $\{0, 1, 2\}$.

5. If b is 0 or 1, the prover announces his commitment for y' where $y' = y \oplus b \cdot s$. He also discloses another of his commitments: the one corresponding to the sequence n_j if b equals 0 and the one for m_j if b is 1. Finally, if b equals 2, the prover reveals both commitments for sequences but no other information.

6. If b equals 0 or 1, the verifier checks that the commitment was correct and that the integers n_j or m_j disclosed from the commitment have been computed honestly. Now, if b is 2, the verifier checks the commitments and computes the average value

$$\mu = \frac{1}{q} \sum_{j=1}^{q} \mid n_j - m_j \mid$$

In the last case, the verifier accepts the round if

$$\mu < 1.07\sqrt{p}$$

The number r of consecutive rounds depends on the required level of security and will be discussed further on as well as the values of the parameters n, k, l, p, q.

2 Soundness of the scheme

We first prove that a fair user will not be rejected. This is not obvious, at least when b sends a 2 and the probabilistic analysis that we need will also be used to establish the security of the scheme. We consider a random n-bit word. Such a random word t can be viewed as a sequence of n independant Bernoùilli trials. We let

$$m = \mid t \mid, m' = \mid t \oplus s \mid.$$

and we consider the distribution of $\mid m - m' \mid$.If we let

$$T = \sum_{s(i)=1} t(i)$$

then, it is easily seen that $\mid m - m' \mid$ is exactly $\mid 2T - p \mid$. Thus, we have to study the random variable $\mid 2T - p \mid$, where T is the sum of p independant Bernouilli trials. The expectation ν of this variable can be estimated by

$$\nu = \sqrt{\frac{2p}{\pi}} \approx 0.798\sqrt{p}$$

and its standard deviation by

$$\sigma = \sqrt{p - 2p/\pi} \approx 0.603\sqrt{p}$$

Now, the values $\mid n_j - m_j \mid$, which are computed in step 6, are precisely values of $\mid 2T - p \mid$ corresponding to the random choice $t = y \oplus z_j$. We will use the central limit theorem in order to estimate the probability that the computed average value μ does not differ too much from the expectation ν. Of course, this is not quite correct as the w_j's are fixed so that we don't have independant variables. Still, it is heuristically justified as the w_j's have been chosen randomly. Furthermore, no contradiction arises from extensive numerical simulations.

Following these lines, we get

$$\begin{aligned} \mathbf{P}\{\mid \mu - \nu \mid \geq \tau\sqrt{p}\} &\leq \int_{\frac{\tau\sqrt{pq}}{\sigma}}^{\infty} e^{-x^2/2} dx \\ &\leq \frac{2\sigma}{\tau}\sqrt{\frac{2}{\pi pq}} e^{-\tau^2 pq/2\sigma^2} \\ &\approx \frac{0.962}{\tau\sqrt{q}}(0.253)^{\tau^2 q} \end{aligned}$$

Setting $\tau = 0.272$, one finds that the probability of having μ above $1.07\sqrt{p}$ is at most $6.96\ 10^{-7}$ for $q = 128$. Even if the number of rounds is 60, in which case average values will be computed about 20 times, this makes an overall probability that a fair user is rejected as small as $1.4\ 10^{-5}$. This can be easily handled, e.g. by giving the prover another chance to identify himself.

3 Security of the scheme

Before we discuss the security of the proposed scheme, let us consider possible collisions between public keys. If s ans s' are secret keys which yield the same public identification, then, $s \oplus s'$ is a codeword of lenth at most $2p$. Now, recall that G was chosen randomly, so that the corresponding code should have a good correcting power. More precisely, it is known, that random codes almost surely satisfy the Gilbert-Varshamov bound ([4]). Granted this fact, we get, for $k = n/2$, that any non-zero codeword has weight at least $0.11\ n$. Thus taking $p < 0.055\ n$ should prevent any collision to happen. Even if we don't fulfill this condition, collisions remain very unliquely. The same argument shows that a commitment essentially bounds the prover to his original choice, provided that the pieces have small weight.

Of course, the security of the scheme relies on the difficulty of inverting the function

$$s \longrightarrow G(s)$$

when its arguments are restricted to valid secret keys. In order to give evidence of this difficulty, let us recall from [1] that it is NP-complete to determine whether a code has a word s of weight $\leq p$ whose image is a given k-bit word i. Let us also observe that, in case no collision of secret keys can happen, finding s is exactly equivalent to finding the codeword w minimizing the weight of $t \oplus w$, when an element t of $G^{-1}(i)$ is chosen. But this is the problem of decoding unstructured codes which is currently believed to be ûnsolvable.

In order to counterfeit a given signature without knowing the secret key, various strategies can be used.

- Having only y ready for the verifier's query and annoucing something very close to $\mid y \oplus z_j \mid$ in place of m_j. In this case, the false prover hopes that b is 0 or 2 and the probability of success is $(2/3)^{-r}$, where r is the number of rounds. A similar strategy can be defined with $y \oplus s$ in place of y and shifting beetween the two yields the same probability of success.

- Having both y and $y \oplus t$ ready where t is some element such that $G(t) = i$, presumably distinct from s but whose weight is reasonably small. If the cheater realizes he will fail the round that way, he can still go back to the previous strategy for this round.

We will now describe two choices of the parameters such that the corresponding code can resist attacks based on the ability of a cheater to produce elements of $G^{-1}(i)$ of moderate weight. We feel that our analysis gives evidence to the security of our scheme because the cheater's performances that we consider are not met by known algorithms, as far as we are aware.

- We let $n = 51\hat{2}, k = 256, l = 4, p = 30, q = 128$ and we assume that the cheater can produce elements of $G^{-1}(i)$ of weight approximately $0.2n$. With these assumptions, computations similar to the above show that, each time he tries some t, the cheater has a probability of producing an average deviation μ lying below the critic value $1.07\sqrt{30} \approx 5.86$ which is roughly $9\ 10^{-5}$ and this only increases by a minute fraction the probability of success $2/3$ of the basic cheating strategy, thus yielding a probability $(0.67)^r$ of going through r rounds.
- We let $n = 1024, k = 512, l = 8, p = 40, q = 128$ and we assume that the cheater can produce elements of $G^{-1}(i)$ of weight close to $0.12n$. Assuming that the Gilbert-Varshamov bound holds, this is almost the optimum, except of course if an algorithm can disclose s. With these assumption, any trial has a probability of producing μ below the critic value 6.76 which is roughly $1.03\ 10^{-3}$ and, once again, it does not increase drastically the probability of success of the basic strategy.

4 Discussion of the scheme

We close the paper by various remarks.

1. We first discuss the amount of information on the secret key s which is disclosed when the scheme is used. Our basic assumption is that it is not possible to break a commitment. Once again, this hypothesis relies on the supposed difficulty of finding a word of small weight whose image by G is given. In our examples, the average length of the words used to build a commitment is 64 and thus, these words are out of reach, given the ability that we have assigned to an opponent in each case. Granted this, the only information that comes out of a round is either a random word, y or $y \oplus s$, or else two distribution of numbers n_j and m_j. Now, if j is fixed, n_j and m_j can be considered as independant random variables following a binomial ditribution. Of course, the family obtained when j varies is not made of independant variables. Still, since the order of appearance is essentially unknown, it seems virtually impossible to undertake any statistical analysis that might reveal s.

2. One of the defects of the proposed scheme is the required amount of memory: in order to store G and the w_j's,one needs 150 kbits in the weaker case considered and 630 kbits in the stronger case. Still, because the operations to perform are very simple, they can be implemented in hardware in a quite efficient way. On the other hand, the communication complexity of the protocol is not much worse than in the Fiat-Shamir scheme: in our smaller example,the n_j's are numbers which almost surely do not differ from 256 by more than 128 and can therefore be coded on 8 bits, so that the sequence of 4 commitments sent by the prover is 4000 bits long.

3. The security of the scheme can be increased by taking q and r larger. Also, the z_j's can be interactively produced instead of being defined in a rather systematic way: all random choices are really independant and this makes the theoretical analysis of the protocol more reliable. On the other hand, the communication complexity becomes much larger and this does not seem to be desirable. Finally, the rounds can be performed in parallel, reducing the number of steps to 6.

4. In order to limit the communication complexity, it is possible to let the prover commit himself to y and $y \oplus s$ by simply announcing $G(y)$ and $G(y \oplus s)$. This actually opens a new way to cheat by trying a lot of members t of $G^{-1}(i)$ instead of one. Since it involves heavy on-line computations, the number of trials can be limited by a timing-out device. Still, it will be necessary to increase q. In order to resist against 10^4 trials, we propose $q = 150$. Accordingly, our statistical requirements can be made a bit more strict (e.g. $1.05\sqrt{p}$ instead of $1.07\sqrt{p}$).

5. It is tempting to lower the value of p. It would be rather dangerous: using the arguments of [5], one can see that secret keys of small weight (e.g. $p = 20$) will presumably be found.

6. **acknowledgement** We wish to thank S. Harari for sending us an early version of his work [3].

References

[1] E.R.Berlekamp, R.J.Mc Eliece and H.C.A. Van Tilborg, On the inherent intractability of certain coding problems, *IEEE Trans. Inform. Theory*, (1978) 384-386.

[2] A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, *Proceedings of Crypto 86*, Santa-Barbara (1986), 181-187.

[3] S. Harari, Un algorithme d'authentification sans transfert d'information *Proceedings*, Trois journées sur le codage,Toulon (1988), to appear.

[4] J.N. Pierce, Limit distributions of the minimum distance of random linear codes, *IEEE Trans. Inform. Theory*, (1967) 595-599.

[5] J.Stern, A method for finding codewords of small weight, *Proceedings*, Trois journées sur le codage, Toulon (1988), to appear.