Advances in Cryptology – Proceedings of EUROCRYPT '96 (may 12–16, 1996, Zaragoza, Spain)
U. Maurer, Ed. Springer-Verlag, LNCS 1070, pages 387–398.

Security Proofs for Signature Schemes

David Pointcheval and Jacques Stern

Laboratoire d'Informatique, cole Normale Suprieure,
45, rue d'Ulm, F – 75230 PARIS Cedex 05.
E-mail: {David.Pointcheval, Jacques.Stern}@ens.fr

Abstract. In this paper, we address the question of providing security proofs for signature schemes in the so-called random oracle model [1]. In particular, we establish the generality of this technique against adaptively chosen message attacks. Our main application achieves such a security proof for a slight variant of the El Gamal signature scheme [3] where committed values are hashed together with the message. This is a rather surprising result since the original El Gamal is, as RSA [11], subject to existential forgery.

1 Introduction

Since the appearance of the public key cryptography, in the famous Diffie-Hellman paper [2], a significant line of research has tried to provide "provable" security for cryptographic protocols. In the area of computational security, proofs have been given in the asymptotic framework of complexity theory. Still, these are not absolute proofs since cryptography ultimately relies on the existence of one-way functions and the $\mathcal{P}$ vs. $\mathcal{NP}$ question. Rather, they are computational reductions to and from well established problems from number theory such as factoring, the discrete logarithm problem or the root extraction problem, on which RSA relies [11].

In the present paper we will exclusively focus on signatures. As shown in the Diffie-Hellman paper [2], the trapdoor function paradigm allows to create signatures in the public key setting. Nevertheless, both the RSA scheme and the El Gamal scheme are not provably secure since they are subject to existential forgery. In other words, it is easy to create a new valid message-signature pair. In many cases, this is not really dangerous because the message is not intelligible or does not have the proper redundancy. Still an RSA signature does not prove by itself the identity of the sender.

The first signature scheme proven secure against a very general attack, the so-called adaptively chosen-message attack which will be defined later in this paper, has been proposed by Goldwasser-Micali-Rivest [6] in 1984. It uses the notion of claw-free permutations. We refer to [6] for details.

In 1986, a new paradigm for signature schemes was introduced. It is derived from zero-knowledge identification protocols involving a prover and a verifier [5], and uses hash functions in order to create a kind of virtual verifier. In [4], Fiat and Shamir proposed a zero-knowledge identification protocol based on the hardness of extracting square roots. They also described the corresponding signature scheme and outlined its security. Similar results for other signature schemes like Schnorr's [12] are considered as folklore results but have never appeared in the literature.

In this paper, we review the basic method for proving security of signature schemes in the random oracle model [1] and surprisingly, we prove the security of a very close variant of the El Gamal signature scheme.

2 Framework

2.1 Generic Signature Schemes

In a signature scheme, each user publishes a public key while keeping for himself a secret key. A user's signature on a message m is a value which depends on m and on the user's public and secret keys in such a way that anyone can check validity just by using the public key. However, it is hard to forge a user's signature without knowing his secret key. In this section, we will give a more precise definition of a signature scheme and of the possible attacks against such schemes. These definitions are based on [6].

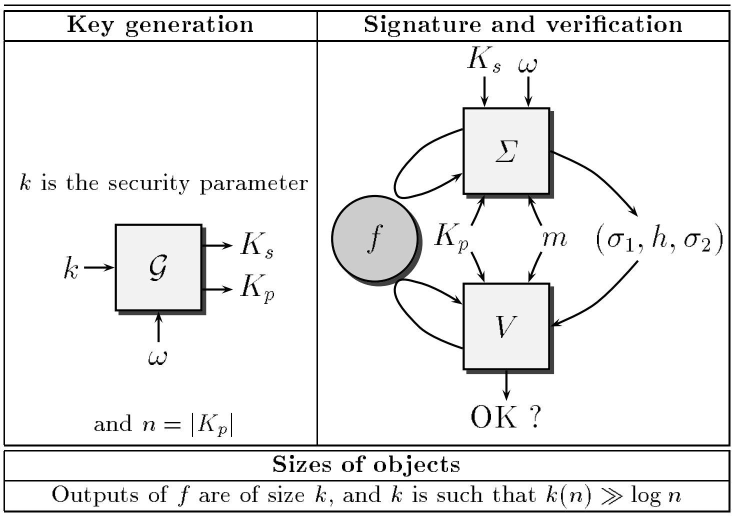

Fig. 1. Signature schemes

Definition 1. A signature scheme is defined by the following (see figure 1):

- the *key generation algorithm* $\mathcal{G}$ which, on input 1^k, where k is the security parameter, produces a pair (K_p, K_s) of matching public and secret keys. It is clear that $\mathcal{G}$ must be a probabilistic algorithm.
- the *signing algorithm* Σ which, given a message m and a pair of matching public and secret keys (K_p, K_s), produces a signature. The signing algorithm might be probabilistic, and in some schemes it might receive other inputs as well.
- the *verification algorithm* V which, given a signature σ, a message m and a public key K_p, tests whether σ is a valid signature of m with respect to K_p. In general, the verification algorithm need not be probabilistic.

Signature schemes often use a hash function f. In this paper, we will only consider signature schemes which, on the input message m, produce triplets (σ_1, h, σ_2) independent of previous signature. In those triplets (σ_1, h, σ_2), h is the hash value of (m, σ_1) and σ_2 just depends on σ_1, the message m, and h. This covers the case of Fiat-Shamir [4], Schnorr [12] and many others. In some cases, σ_1 or h can be omitted, but we will keep them for more generality.

2.2 Attacks

We will only consider two different scenarios involving probabilistic polynomial time Turing machines, the no-message attack and the adaptively chosen message attack (see figure 2).

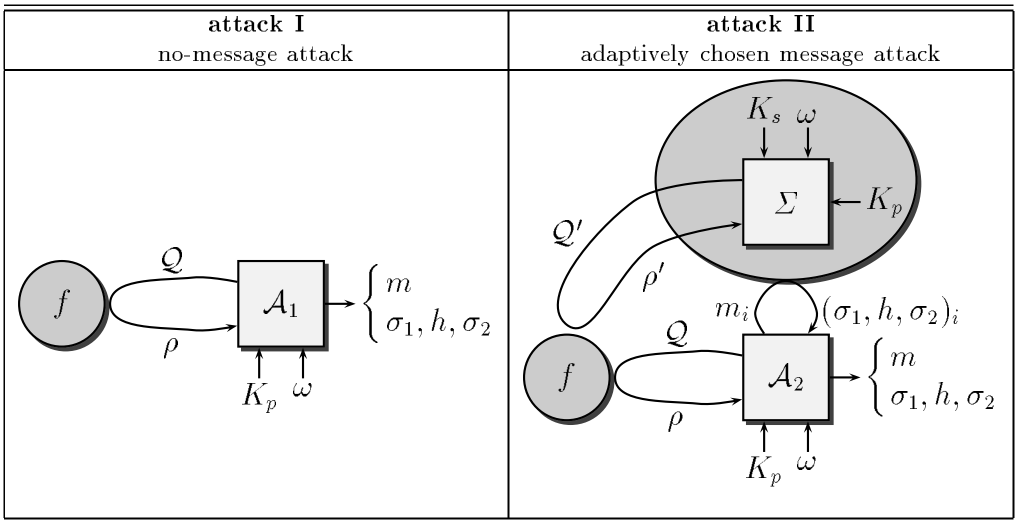

Fig. 2. Attacks

In the former, the attacker only knows the signer's public key. In the latter, he can dynamically ask the legitimate user to sign any message, using him as a kind of oracle. For the resistance against adaptively chosen message attacks, which is a stronger requirement, we will use the possible simulation of the legitimate signer, which relies on the honest verifier zero-knowledge property of the identification scheme.

2.3 The Random Oracle Model

As we already pointed out, signature schemes often use a hash function f (e.g. MD5 [10] or SHS [8]). This use of hash functions may have been motivated by the wish to sign long messages with a single signature. Accordingly, the requirement of the function was collision freeness. It was later realized that hash functions were an essential ingredient for the security of the signature schemes. Still, in order to actually provide such a security proof, stronger assumptions seem to be needed and several authors (e.g. [4] and [1]) have suggested to use to hypothesis that f is actually a random function. We follow this suggestion by using the corresponding model, called the "random oracle model". In this model, the hash function can be seen as an oracle which produces a random value for each new query. Of course, if the same query is asked twice, identical answers are obtained. Proofs in this model ensure security of the overall design of a signature scheme provided the hash function has no weakness.

3 The Oracle Replay Attack

In this section, we will prove a key lemma, which we call the forking lemma and which will be repeatedly used in the sequel. This lemma uses the "oracle replay attack": by a polynomial replay of the attack with the same random tape and a different oracle, we obtain two signatures of a specific form which open a way to solve the underlying hard problem.

Lemma 1 (the forking lemma). *Let $\mathcal{A}$ be a Probabilistic Polynomial Time Turing machine, given only the public data as input. If $\mathcal{A}$ can find, with non-negligible probability, a valid signature $(m, \sigma_1, h, \sigma_2)$, then, with non-negligible probability, a replay of this machine, with the same random tape and a different oracle, outputs two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$.*

Remark 1. Probabilities are taken over random tapes, random oracles, and in some cases, over messages and keys.

Before we prove this result, we state a well-known probabilistic lemma:

Lemma 2. *Let $A \subset X \times Y$, such that $\Pr[A(x, y)] \geq \epsilon$, then there exists $\Omega \subset X$ such that*
i) $\Pr[x \in \Omega] \geq \epsilon/2$
ii) whenever $a \in \Omega$, $\Pr[A(a, y)] \geq \epsilon/2$

With this lemma, we can split X in two subsets, a subset Ω consisting of "good" x's which provide a non-negligible probability of success over y, and its complement. We now return to the forking lemma.

Proof. We assume that we have a no-message attacker $\mathcal{A}$, which is a probabilistic polynomial time Turing machine with a random tape ω. During the attack, this machine asks a polynomial number of questions to the random oracle f. We may assume that these questions are distinct, for instance, $\mathcal{A}$ can store questions and answers in a table. Let $\mathcal{Q}_1, \ldots, \mathcal{Q}_Q$ be the Q distinct questions, where Q is a polynomial, and let $\rho_1, \ldots, \rho_Q$ be the Q answers of f. It is clear that a random choice of f exactly corresponds to a random choice of $\rho_1, \ldots, \rho_Q$. For a random choice of $\omega, \rho_1, \ldots, \rho_Q$, with non-negligible probability, $\mathcal{A}$ outputs a valid signature $(m, \sigma_1, h, \sigma_2)$. It is easy to see that the probability for the precise query (m, σ_1) not to be asked is negligible, because of the randomness of $f(m, \sigma_1)$. So, the probability that the query (m, σ_1) is one of the $\mathcal{Q}_i$'s, e.g. $\mathcal{Q}_\beta$, is non-negligible. Since β is between 1 and $Q(n)$, there exist a β and a polynomial P such that

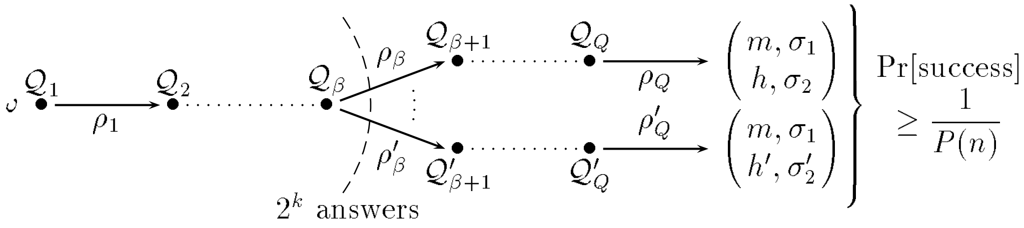

Fig. 3. The forking lemma

the probability of success, over $\omega, \rho_1, \ldots, \rho_Q$, with $\mathcal{Q}_\beta = (m, \sigma_1)$ is greater than $1/P(n)$ (see figure 3).

With such a β, using lemma 2, we get the existence of a non-negligible subset Ω_β of "good" ω's. For such a "good" ω, the probability of success, over $\rho_1, \ldots, \rho_Q$, with $\mathcal{Q}_\beta = (m, \sigma_1)$, is greater than $1/2P(n)$. With such β and ω, using lemma 2 again, we obtain the existence of a non-negligible subset $R_{\beta,\omega}$ of "good" $(\rho_1, \ldots, \rho_{\beta-1})$'s. For such a "good" $(\rho_1, \ldots, \rho_{\beta-1})$, the probability of success of the attacker, over $\rho_\beta, \ldots, \rho_Q$, with $\mathcal{Q}_\beta = (m, \sigma_1)$, is greater than $1/4P(n)$. Then, with such β, ω and $(\rho_1, \ldots, \rho_{\beta-1})$, if we randomly chose $\rho_\beta, \ldots, \rho_Q$ and $\rho'_\beta, \ldots, \rho'_Q$, with a non-negligible probability, we obtain two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma'_2)$ such that $h \neq h'$; this uses the fact that $k(n) \gg \log n$.

Finally, with a random choice of $\beta, \omega, \rho_1, \ldots, \rho_{\beta-1}, \rho_\beta, \ldots, \rho_Q$ and $\rho'_\beta, \ldots, \rho'_Q$, we obtain, with a non-negligible probability, two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma'_2)$ such that $h \neq h'$.

4 The Fiat-Shamir Signature Scheme

We will now apply the lemma to the Fiat-Shamir signature scheme in order to prove its security against no-message attacks. This result is outlined in [4] and we include it for the reader's convenience.

4.1 Description

Firstly, we describe the single key Fiat-Shamir signature scheme [4]:

- the key generation algorithm: for a security parameter k, it chooses two large primes p and q which are kept secret and computes their product N and defines a random hash function f with a k-bit output. Then, it chooses a random $s \in \mathbb{Z}/N\mathbb{Z}$ and publishes its square $v = s^2 \bmod N$. N and f are public and s is the secret key.
- the signature algorithm: in order to sign a message m, one generates k random numbers, $r_i \in \mathbb{Z}/N\mathbb{Z}$ for $i = 1, \ldots, k$, computes their respective squares $x_i = r_i^2 \bmod N$ as well as the challenge $h = (e_1 \ldots e_k) = f(m, x_1, \ldots, x_k)$. From these data, one sets $y_i = r_i s^{e_i} \bmod N$ and outputs $\sigma_1 = (x_1, \ldots, x_k)$ and $\sigma_2 = (y_1, \ldots, y_k)$. The signature is (σ_1, h, σ_2).
- the verification algorithm is as follows: for a given message m and a given signature $\sigma_1 = (x_1, \ldots, x_k)$, $h = (e_1 \ldots e_k)$ and $\sigma_2 = (y_1, \ldots, y_k)$, it checks whether $h = f(m, \sigma_1)$ and $y_i = r_i s^{e_i} \bmod N$ for all i.

4.2 Proof of Security

From the forking lemma we easily get a proof in the random oracle model.

Theorem 1. *Consider a no-message attack in the random oracle model. If an existential forgery of the Fiat-Shamir signature scheme is possible with non-negligible probability, then factorization of RSA moduli can be performed in polynomial time.*

Proof. Let $N \in \mathbb{N}$ be the integer to factor. Let us choose $s \in_R \mathbb{Z}/N\mathbb{Z}$, and let $v = s^2 \bmod N$. If an attacker $\mathcal{A}_1$ can break the Fiat-Shamir signature scheme, then by using the forking lemma, he can obtain two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$, such that $h \neq h'$. From this, we get i such that $h_i \neq h_i'$, say $h_i = 0$ and $h_i' = 1$. We get $y_i^2 = x_i \bmod N$ and $y_i'^2 = x_i v \bmod N$. If we let $z = y_i' y_i^{-1} \bmod N$, then $z^2 = v = s^2 \bmod N$.

Since the algorithm cannot distinguish s from other roots, we conclude that, with a probability $1/2$, $\gcd(z - s, N)$ provides a factor of N.

Remark 2. Because of the easy simulation of the communication with an honest verifier, even in the context of the parallel version of Fiat-Shamir, the proof of security against adaptively chosen message attacks is straightforward.

5 The Modified El Gamal Signature Scheme

The original El Gamal signature scheme [3] was proposed in 1985 but its security was never proved equivalent to the discrete logarithm problem nor to the Diffie-Hellman problem. Using the forking lemma, we will prove the security of a slight variant of this scheme.

5.1 Description of the Original Scheme

Let us begin with a description of the original scheme [3]:

- the key generation algorithm: it chooses a random large prime p of size n and a generator g of $(\mathbb{Z}/p\mathbb{Z})^\star$, both public. Then, for a random secret key $x \in \mathbb{Z}/(p-1)\mathbb{Z}$, it computes the public key $y = g^x \bmod p$.
- the signature algorithm: in order to sign a signature of a message m, one generates a pair (r, s) such that $g^m = y^r r^s \bmod p$. To achieve this aim, one has to choose a random $K \in (\mathbb{Z}/(p-1)\mathbb{Z})^\star$, compute the exponentiation $r = g^K \bmod p$ and solve the linear equation $m = xr + Ks \bmod (p-1)$. The algorithm finally outputs (r, s).
- the verification algorithm checks the equation $g^m = y^r r^s \bmod p$.

5.2 Security

As already seen in the original paper, one cannot show that the scheme is fully secure because it is subject to existential forgery.

Theorem 2. *The original El Gamal signature scheme is existentially forgeable.*

Proof. This is a well-known result, but we describe two level of forgeries:

1. the one parameter forgery: let $e \in_R \mathbb{Z}/(p-1)\mathbb{Z}$, if we let $r = g^e y \bmod p$ and $s = -r \bmod p-1$, it is easy to see that (r, s) is a valid signature for the message $m = es \bmod p-1$.
2. the two parameter forgery: let $e \in_R \mathbb{Z}/(p-1)\mathbb{Z}$ and $v \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^\star$, if we let $r = g^e y^v \bmod p$ and $s = -rv^{-1} \bmod p-1$, then (r, s) is a valid signature for the message $m = es \bmod p-1$.

We now modify this scheme by using a hash function.

5.3 Description of the modified El Gamal scheme

In this variant, we replace m by the hash value of the entire part of the computation bound not to change, namely $f(m, r)$.

- the key generation algorithm: unchanged.
- the signature algorithm: in order to sign a message m, one generates a pair (r, s) such that $g^{f(m,r)} = y^r r^s \bmod p$. In order to achieve this aim, one generates K and r the same way as before and solves the linear equation $f(m, r) = xr + Ks \bmod (p-1)$. The algorithm outputs $(r, f(m, r), s)$.
- the verification algorithm checks the signature equation with the obvious changes due to the hash function.

5.4 Proofs of security

In this section, we will see that the modification allows to prove the security of the scheme even against an adaptively chosen message attack, at least for a large variety of moduli. We let $|p|$ denote the length of an integer p.

Definition 2. Let α be a fixed real. An α-hard prime number p is such that the factorization of $p-1$ yields $p-1 = QR$ with Q prime and $R \leq |p|^\alpha$.

Remark 3. Those prime moduli are precisely those used for cryptographic applications of the discrete logarithm problem.

Security against a no-message attack. Firstly, we study the resistance of the modified El Gamal Signature scheme against no-message attacks.

Theorem 3. *Consider a no-message attack in the random oracle model against schemes using α-hard prime moduli. Probabilities are taken over random tapes, random oracles and public keys. If an existential forgery of this scheme has non-negligible probability of success, then the discrete logarithm problem with α-hard prime moduli can be solved in polynomial time.*

Proof. Using the forking lemma, we get two valid signatures (m, r, h, s) and (m, r, h', s') such that $g^h = r^s y^r \bmod p$ and $g^{h'} = r^{s'} y^r \bmod p$. Hence, we get $g^{hs'-h's} = y^{r(s'-s)} \bmod p$ and $g^{h'-h} = r^{s-s'} \bmod p$. Since g is a generator of

$(\mathbb{Z}/p\mathbb{Z})^\star$, there exist t and x such that $g^t = r \bmod p$ and $g^x = y \bmod p$. Therefore,

$$hs' - h's = xr(s' - s) \bmod p - 1 \qquad (1)$$
$$h' - h = t(s - s') \bmod p - 1 \qquad (2)$$

Since h and h' come from "oracle replay", we may further assume $h - h'$ is prime to Q, so that $\gcd(s - s', Q) = 1$. Nevertheless, we cannot make any further assumption for r, and accordingly, two cases appear:

case 1: r is prime to Q. In this case, equation (1) provides the Q modular part of x, $x = (hs' - h's)(r(s - s'))^{-1} \bmod Q$. With an exhaustive search over the R modular part of x, we can find an x which satisfies $y = g^x \bmod p$.

case 2 otherwise, $r = bQ$ with b small. In this case, equation (2) provides the Q modular part of t, $t = (h - h')(s - s')^{-1} \bmod Q$. With an exhaustive search over the R modular part of t, we can find a t which satisfies $bQ = g^t \bmod p$. We note that t is prime to Q.

At this point, we have a probabilistic polynomial time Turing machine $\mathcal{M}$ which, on input (g, y), outputs, with non-negligible probability, $x \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that $y = g^x \bmod p$ (case 1) or $b \in \mathbb{Z}/R\mathbb{Z}$ and $t \in \mathbb{Z}/(p-1)\mathbb{Z}$ such that $bQ = g^t \bmod p$ (case 2). Probabilities are taken over g, y, and the random tapes of $\mathcal{M}$. Using lemma 2, let $\mathcal{G}$ be a non-negligible set of g's such that whenever $g \in \mathcal{G}$, the set of y's which provides the above witnesses is non-negligible. To make things precise, we consider both probabilities to be greater than ε, where ε is the inverse of some polynomial. Let G_{good} be the set of $g \in \mathcal{G}$ which lead to the first case with probability greater than $\varepsilon/2$. Let G_{bad} be the set of $g \in \mathcal{G}$ which lead to the second case with probability greater than $\varepsilon/2$. We know that $\mathcal{G}$ is the union $G_{good} \cup G_{bad}$.

If G_{good} has probability greater than $\varepsilon/2$, then we have a probabilistic polynomial time Turing machine which can compute, for a non-negligible part of (g, y), the discrete logarithm of y relatively to g.

Otherwise, bad g's are in proportion greater than $\varepsilon/2$. Since the set of possible b's is polynomial, we get a fixed b and a non-negligible subset $G_{bad}(b)$ of bad g's such that, with non-negligible probability, $\mathcal{M}(g, y)$ outputs integers b and t such that $bQ = g^t \bmod p$. Let $g \in G_{bad}(b)$ and y be any number. Running $\mathcal{M}(g, z)$, for random z, we get, with non-negligible probability, some t such that $g^t = bQ \bmod p$. Running $\mathcal{M}(yg^\ell, z')$, for random ℓ and z', we get, with non-negligible probability, $yg^\ell \in G_{bad}(b)$ and some t' such that $(yg^\ell)^{t'} = bQ = g^t \bmod p$. Hence, $xt' = t - \ell t' \bmod (p-1)$. Since t' is prime to Q, we get $x \bmod Q$. After polynomially many trials over the R modular part of x, we find the logarithm of y. Then we have another probabilistic polynomial time Turing machine $\mathcal{M}'$ which can compute for a non-negligible part of (g, y), the discrete logarithm of y relatively to g.

Now, let fix g and y. Running the machine on (g^u, yg^v) with random u and v, we obtain, with non-negligible probability, an x such that $yg^v = g^{ux} \bmod p$, hence we get $y = g^{ux-v} \bmod p$. This finally contradicts the intractability assumption.

Security against an adaptively chosen message attack. We now prove a more surprising theorem about the security against adaptively chosen message attacks. In the adaptively chosen message scenario, the attacker uses the signer as a kind of oracle. If it is possible to simulate the signer Σ by a simulator $\mathcal{S}$ who does not know the secret key (see figure 4), then we can make the attacker and the simulator collude in order to break the signature scheme, and, the same way as before, we can solve the discrete logarithm.

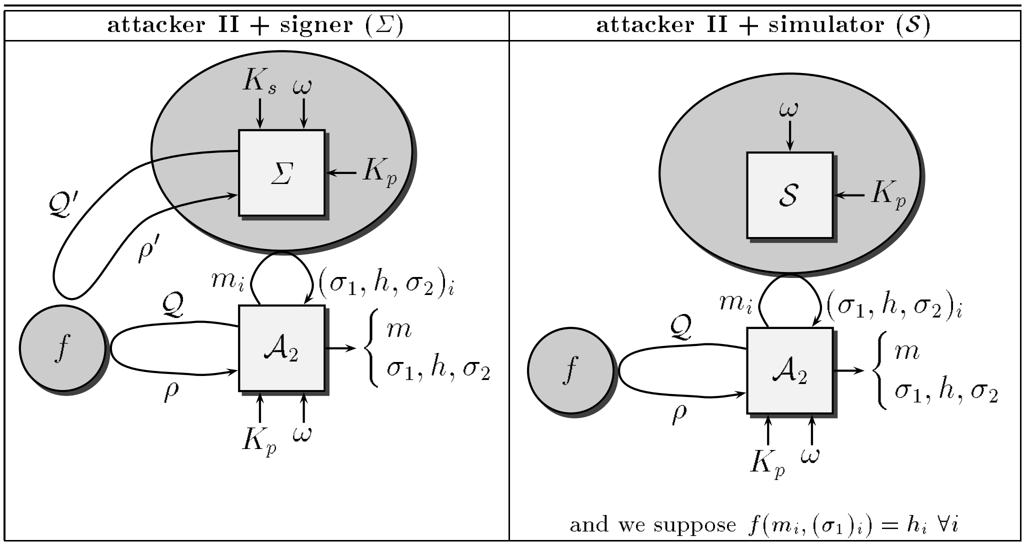

Fig. 4. Adaptively chosen message scenario

Lemma 3. *For α-hard prime numbers, the signer can be simulated with an indistinguishable distribution.*

Proof. A key ingredient of the proof is as follows: values returned by the random oracle can be freely computed and have no correlation with messages whose signature is requested.

In this proof, we identify the output set H of random oracles with the set $\{0, \ldots, 2^k - 1\}$ and we assume that $2^k \geq Q$.

Using the two parameter forgery for the Q modular part, and an exhaustive search for the other part, we can obtain an indistinguishable simulation: we first randomly choose $u \in \mathbb{Z}/Q\mathbb{Z}$, $t \in (\mathbb{Z}/Q\mathbb{Z})^\star$ and $\ell \in (\mathbb{Z}/R\mathbb{Z})^\star$. Then, we let $e = uR \bmod (p-1)$, $v = tR \bmod (p-1)$ and $r = (g^e y^v) g^{Q\ell} \bmod p$. We start the simulation again in the (unlikely) situation where r is not a generator of $(\mathbb{Z}/p\mathbb{Z})^\star$. This corresponds to separately dealing with the forgery in the two subgroups respectively generated by g^R and g^Q. Mimicking the two parameter forgery in the subgroup generated by g^R, we need to set $s = -rv^{-1} \bmod Q$ and $h = -erv^{-1} \bmod Q$. For the R modular part, we randomly choose $h \bmod R$ until $h \in H$, and we exhaustively search for an s which satisfies $g^h = y^r r^s \bmod p$: taking logarithms, this reads as $h = rx + Q\ell s \bmod R$, so that the number of trials is only polynomial. We can easily check that the triplet (r, h, s) is a valid signature of a message m as soon as $h = f(m, r)$.

Let $(r, h, s) \in (\mathbb{Z}/p\mathbb{Z})^\star \times H \times \mathbb{Z}/(p-1)\mathbb{Z}$ such that $g^h = r^s y^r \bmod p$ and r is a generator of $(\mathbb{Z}/p\mathbb{Z})^\star$. Trying to output this signature through our simulation yields the system of equations

$$\begin{cases} hv + re = 0 & \bmod Q \\ xv + e = \log_g r & \bmod Q \end{cases}$$

If $h \neq xr \bmod Q$, then there is exactly one solution and therefore one way for $\mathcal{S}$ to generate such a signature. If $h = xr \bmod Q$, then $\mathcal{S}$ can generate such a signature only if $r = h = s = 0 \bmod Q$, and $Q - 1$ different ways.

Both types of exceptions contribute to the overall distance by some term bounded by $\frac{2R}{(Q-1)\varphi(R)}$ which is less than $4\alpha n^{\alpha+1} \times 2^{-n}$, a negligible value, where $n = |p|$

Theorem 4. *Consider an adaptively chosen message attack in the random oracle model against schemes using α-hard prime moduli. Probabilities are taken over random tapes, random oracles and public keys. If an existential forgery of this scheme has non-negligible probability of success, then the discrete logarithm problem with α-hard prime moduli can be solved in polynomial time.*

Proof. For each simulated signature of m_i, (r_i, h_i, s_i), $\mathcal{S}$ is assumed to have asked $f(m_i, r_i)$ and obtained h_i, a new random value. We observe that collisions of queries happen with negligible probability, therefore, the attacker cannot distinguish the simulator from the legitimate signer. And then, like in theorem 3, a collusion of the attacker and the simulator enables to compute discrete logarithms.

6 Further Results

In this section, we mention several additional results: the first is the extension of theorem 1 to the adaptively chosen message attack. Furthermore, because of the possible simulation of the Schnorr scheme, we can prove the following theorem in the random oracle model:

Theorem 5. *If an existential forgery of the Schnorr signature scheme, under an adaptively chosen message attack, has non-negligible probability of success, then the discrete logarithm in subgroups can be solved in polynomial time.*

The same results are true for every signature scheme which comes from the transformation of a honest verifier zero-knowledge identification protocol (Guillou-Quisquater [7], the Permuted Kernel Problem [13], the Syndrome Decoding problem [14], the Constrained Linear Equations [15], the Permuted Perceptrons Problem [9], etc.). For each of them, existential forgery under an adaptively chosen-message attack in the random oracle model is equivalent to the problem on which the identification scheme relies .

References

1. M. Bellare and P. Rogaway. Random Oracles are Practical: a Paradigm for Designing Efficient Protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, U.S.A., 1993. ACM press.
2. W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Transactions on Information Theory*, volume IT–22, no. 6, pages 644–654, November 1976.
3. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *IEEE Transactions on Information Theory*, volume IT–31, no. 4, pages 469–472, July 1985.
4. A. Fiat and A. Shamir. How to Prove Yourself: practical solutions of identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology – Proceedings of CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa-Barbara, California, 1987. Springer-Verlag.
5. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proceedings of the 17th ACM Symposium on the Theory of Computing STOC*, pages 291–304, Providence, Rhode Island, U.S.A., 1985. ACM Press.
6. S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptative Chosen-Message Attacks. *SIAM Sournal of Computing*, 17(2):281–308, April 1988.
7. L. C. Guillou and J.-J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In C. G. Günter, editor, *Advances in Cryptology – Proceedings of EUROCRYPT '88*, volume 330 of *Lecture Notes in Computer Science*, pages 123–128, Davos, Switzerland, 1988. Springer-Verlag.
8. NIST. Secure Hash Standard (SHS). Federal Information Processing Standards PUBlication 180–1, April 1995.
9. D. Pointcheval. A New Identification Scheme Based on The Perceptrons Problem. In L.C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology – Proceedings of EUROCRYPT '95*, volume 921 of *Lecture Notes in Computer Science*, pages 319–328, Saint-Malo, France, 1995. Springer-Verlag.
10. R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, April 1992.
11. R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.
12. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In G. Brassard, editor, *Advances in Cryptology – Proceedings of CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 235–251, Santa-Barbara, California, 1990. Springer-Verlag.
13. A. Shamir. An Efficient Identification Scheme Based on Permuted Kernels. In G. Brassard, editor, *Advances in Cryptology – Proceedings of CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 606–609, Santa-Barbara, California, 1990. Springer-Verlag.
14. J. Stern. A New Identification Scheme Based on Syndrome Decoding. In D. R. Stinson, editor, *Advances in Cryptology – proceedings of CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21, Santa-Barbara, California, 1994. Springer-Verlag.
15. J. Stern. Designing Identification Schemes with Keys of Short Size. In Y. G. Desmedt, editor, *Advances in Cryptology – proceedings of CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 164–173, Santa-Barbara, California, 1994. Springer-Verlag.