Journal of Cryptology, Volume 13, Number 3.
Pages 361–396, Springer-Verlag, 2000.

Journal of CRYPTOLOGY
The Journal of the International Association for Cryptologic Research

Security Arguments for Digital Signatures and Blind Signatures*

David Pointcheval and Jacques Stern

Laboratoire d'Informatique, École Normale Supérieure,
75230 Paris Cedex 05, France.
{David.Pointcheval, Jacques.Stern}@ens.fr
http://www.di.ens.fr/~{pointche,stern}

Communicated by Don Coppersmith

Received 24 October 1997 and revised 22 May 1998

Abstract. Since the appearance of public-key cryptography in the seminal Diffie-Hellman paper, many new schemes have been proposed and many have been broken. Thus, the simple fact that a cryptographic algorithm withstands cryptanalytic attacks for several years is often considered as a kind of validation procedure. A much more convincing line of research has tried to provide "provable" security for cryptographic protocols. Unfortunately, in many cases, provable security is at the cost of a considerable loss in terms of efficiency. Another way to achieve some kind of provable security is to identify concrete cryptographic objects such as hash functions with ideal random objects and to use arguments from relativized complexity theory. The model underlying this approach is often called the "random oracle model." We use the word "arguments" for security results proved in this model. As usual, these arguments are relative to well-established hard algorithmic problems such as factorization or the discrete logarithm.

In this paper we offer security arguments for a large class of known signature schemes. Moreover, we give for the first time an argument for a very slight variation of the well-known El Gamal signature scheme. In spite of the existential forgery of the original scheme, we prove that our variant resists existential forgeries even against an adaptively chosen-message attack. This is provided that the discrete logarithm problem is hard to solve.

Next, we study the security of blind signatures which are the most important ingredient for anonymity in off-line electronic cash systems. We first define an appropriate notion of security related to the setting of electronic cash. We then propose new schemes for which one can provide security arguments.

Keywords: Cryptography, Digital signatures, Blind signatures, Security arguments, Existential forgery, One-more forgery, Forking lemma.

* This paper is the full version of "Security Proofs for Signature Schemes" [43] presented at Eurocrypt '96 and "Provably Secure Blind Signature Schemes" [42] presented at Asiacrypt '96.

Introduction

Since the beginning of public-key cryptography with the Diffie-Hellman paper [16], many new schemes have been proposed and many have been broken. Thus, the simple fact that a cryptographic algorithm withstands cryptanalytic attacks for several years is often considered as a kind of validation procedure. In this approach, cryptanalysis is viewed as a heuristic measure of the strength of a new proposal.

A completely different paradigm is provided by the concept of "provable" security. A significant line of research has tried to provide proofs in the asymptotic framework of complexity theory. Stated in a more accurate way, this approach proposes computational reductions to well established problems, such as factorization, RSA [49], the discrete logarithm problem or any $\mathcal{NP}$-complete problem [24]. Of course, these are not absolute proofs since cryptography ultimately relies on the existence of one-way functions and the $\mathcal{P}$ versus $\mathcal{NP}$ question. Moreover, in many cases, provable security is at the cost of an important loss in terms of efficiency [29],[27],[28],[1].

Recently, the scope of these methods has been considerably widened by using a model where concrete cryptographic objects, such as hash functions, are identified with ideal random objects, the so-called "random oracle model" formalized by Bellare and Rogaway [2]. In this model, DES [34] is viewed as a random permutation and SHA [36] as a random function with the appropriate range.

Using this model, we offer security arguments for a large class of digital signatures. Moreover, we give, for the first time, an argument for a very slight variation of the well-known El Gamal signature scheme [17]. In spite of the existential forgery of the original scheme, we prove that our variant resists existential forgeries even against an adaptively chosen-message attack. This is provided that the discrete logarithm problem is hard to solve. Furthermore, we study the security of blind signatures, especially for their application in electronic cash systems: we first define adequate security notions for blind signatures, then we propose the first schemes for which security arguments can be given.

We now briefly describe the organization of our paper. We first define the so-called "random oracle model" and explain why such a theoretical model can help in proving the validity of the design of a cryptographic scheme. We then recall the definition of a signature scheme together with the various attacks and forgeries that we consider. Also, we present the notion of blind signatures and its use for anonymity (and even revokable anonymity) in electronic cash schemes. Next, we consider the attacks that are relevant in the context of digital payments.

In Section 2, we propose schemes for which one can provide security arguments. In order to simplify the proofs, we first explain our generic technique, the "oracle replay attack" and we present a simple probabilistic lemma, the "splitting lemma." In Section 3, we prove two fundamental "forking lemma's" for digital signatures and blind signatures. They are our main ingredient for providing security arguments for many schemes.

1. Definitions

1.1. *The Random Oracle Model*

Many cryptographic schemes use a hash function f (such as the Message Digest family MD4 [47], MD5 [48], and derived functions SHA-1 [36], HAVAL [40], RIPEMD [46], or RIPEMD-160 [5]). This use of hash functions was originally motivated by the wish to sign long messages with a single short signature. In order to achieve *nonrepudiation*, a minimal requirement on the hash function is to ask that it is impossible for the signer to find two different messages providing the same hash value, this property is called *collision freeness*.

It was later realized that hash functions were an essential ingredient for the security of signature schemes. In order actually to obtain security arguments, while keeping the efficiency of the designs that use hash functions, several authors (e.g. [21], [2],[3], [43], [42] and [44]) have suggested using the 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 truly random value for each new query. Of course, if the same query is asked twice, identical answers are obtained. This is precisely the context of relativized complexity theory with "oracles," hence the name. It is argued that proofs in this model ensure security of the overall design of a signature scheme provided that the hash function has no weakness.

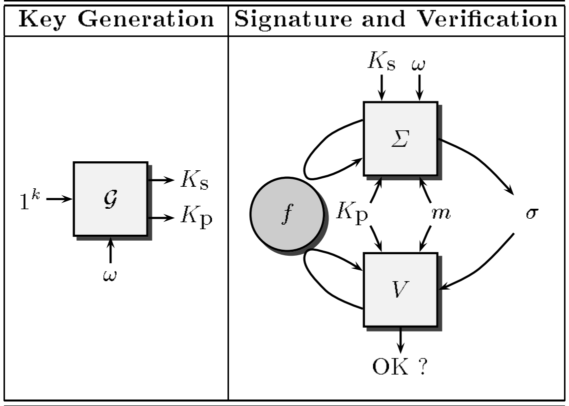
In the following we replace any hash function by a random oracle which outputs k-bit long elements, where k is a security parameter of the cryptographic scheme. In other words, k denotes both the security parameter of the cryptographic (signature) scheme and the length of the output of the random oracle. Roughly speaking, the security level is 2^k.

1.2. *Digital Signature Schemes*

We now turn to digital signature schemes, the electronic version of handwritten signatures for digital documents: a user's signature on a message m is a string which depends on m, on public and secret data specific to the user and–possibly–on randomly chosen data, in such a way that anyone can check the validity of the signature by using public data only. The user's public data are called the *public key*, whereas his secret data are called the *secret key*. Obviously we would like to prevent the forgery of a user's signature without knowledge of his secret key. In this section, we give a more precise definition of signature schemes and of the possible attacks against them. These definitions are based on [28].

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

- The *key generation algorithm* $\mathcal{G}$. On input 1^k, where k is the security parameter, the algorithm $\mathcal{G}$ produces a pair $(K_{\mathrm{p}}, K_{\mathrm{s}})$ of matching public and secret keys. We denote by n the length of the public key. Algorithm $\mathcal{G}$ is probabilistic (with random tape ω).
- The *signing algorithm* Σ. Given a message m and a pair of matching public and secret keys $(K_{\mathrm{p}}, K_{\mathrm{s}})$, Σ produces a signature σ. The signing algorithm



Sizes of objects

$\left.\begin{array}{l} k \text{ is the security parameter} \\ \text{Outputs of } f \text{ are of size } k \\ K_P \text{ is of length } n \end{array}\right\}$ such that $k \gg \log n$

Fig. 1. Signature schemes.

might be probabilistic (with random tape ω), and in some schemes it might receive other inputs as well.

– The *verification algorithm* V. Given a signature σ, a message m and a public key K_P, V tests whether σ is a valid signature of m with respect to K_P. In general, the verification algorithm need not be probabilistic.

1.2.1. *Examples*

As shown in the Diffie-Hellman paper [16], the trapdoor permutation paradigm allows us to create signatures in the public key setting. Two years later, Rivest et al. [49] proposed the first signature scheme based on the RSA trapdoor function:

The RSA Signature. In the RSA context the generation algorithm produces a large composite number $N = pq$, a public key e, and a secret key d such that $e \cdot d = 1 \bmod \varphi(N)$. The signature of a message m is the eth root of m, $\sigma = m^{1/e} = m^d \bmod N$.

The RSA scheme is not secure by itself since it is subject to existential forgery. In other words, it is easy to create a valid message-signature pair, without any help of the signer, using the public verification relation $m = \sigma^e \bmod N$. In many cases, this is not really dangerous because the resulting message is not intelligible or does not have the proper redundancy. Still such an RSA signature does not prove by itself the identity of the sender.

In 1986 a new paradigm for signature schemes was introduced. It is derived from fair zero-knowledge identification protocols involving a prover and a verifier [26], and uses hash functions in order to create a kind of virtual verifier. In [21], 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 security results for other signature schemes like Schnorr's [50], [51] are considered folklore results but have never appeared in print. We refer the reader to the literature for the precise description of those schemes, and we only recall the Schnorr signature:

The Schnorr Signature. The generation algorithm produces two large primes p and q, such that $q \geq 2^k$, where k is the security parameter, and $q \mid p-1$, as well as an element g of $\mathbb{Z}_p^\star$ of order q. It also creates a pair of keys, $x \in \mathbb{Z}_q^\star$ and $y = g^{-x} \bmod p$. The signer publishes y and keeps x secret. The signature of a message m is a triple (r, e, s), where $r = g^K \bmod p$, with a random $K \in \mathbb{Z}_q^\star$, the "challenge" $e = H(m, r) \bmod q$ and $s = K + ex \bmod q$. It satisfies $r = g^s y^e \bmod p$ with $e = H(m, r)$, or simply $e = H(m, g^s y^e \bmod p)$, which is checked by the verifying algorithm.

1.2.2. *Generic Digital Signature Schemes*

In this paper we consider signature schemes which, given the input message m, produce triples (σ_1, h, σ_2) where σ_1 randomly takes its values in a large set, h is the hash value of (m, σ_1) and σ_2 only depends on σ_1, the message m, and h. In particular, we can remark that each signature is independent of the previous ones. More precisely, in the proof of resistance against the strongest attacks, we assume that no σ_1 can appear with probability greater than $2/2^k$, where k is the security parameter. This assumption is satisfied in the Schnorr signature scheme: $\sigma_1 = g^K \bmod p$ for a randomly chosen K in $\mathbb{Z}_q^\star$; since g is of order q, and $k \leq \log q$, the probability for σ_1 to get a specific value is less than $1/(q-1) \leq 2/2^k$. In the same way, the Fiat-Shamir [21] scheme and many others also satisfy this assumption.

In some cases, in order to optimize the size of signatures, σ_1 or h can be omitted, since they can be correctly recovered during the verification process. For notational purposes we ignore these possible optimizations and keep σ_1, h as parts of the signature.

1.2.3. *Attacks*

We focus on two specific kinds of attacks against signature schemes: the *no-message attack* and the *known-message attack*. In the first scenario, the attacker only knows the public key of the signer. In the second one, the attacker has access to a list of message-signature pairs. According to the way this list was created, we distinguish four subclasses of known-message attacks:

- The *plain known-message attack:* the attacker has access to a list of signed messages, but he has not chosen them.
- The *generic chosen-message attack:* the attacker can choose the list of messages to be signed. However this choice must be made before accessing the public key of the signer. We call this attack "generic" because the choice is independent of the signer.
- The *oriented chosen-message attack:* as above, the attacker chooses the list of messages to be signed, but the choice is made once the public key of the signer has been obtained. This attack is oriented against a specific signer.

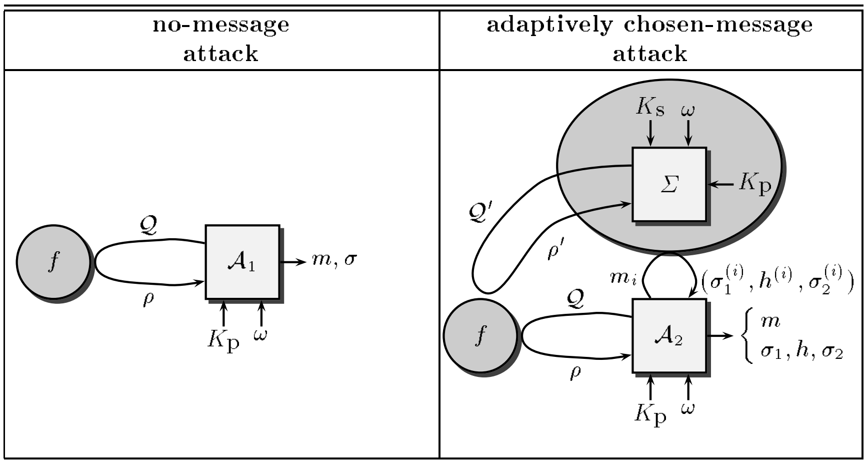


Fig. 2. Attacks.

– The *adaptively chosen-message attack:* having knowledge of the public key of the signer, the attacker can ask the signer to sign any message that he wants. He can then adapt his queries according to previous message-signature pairs.

In the following we only consider the two extreme scenarios, the no-message attack and the adaptively chosen-message attack (see Fig. 2).

1.2.4. *Forgeries*

We now classify the expected results of an attack:

– Disclosing the secret key of the signer. It is the most serious attack. This attack is termed *total break.*
– Constructing an efficient algorithm which is able to sign any message. This is called *universal forgery.*
– Providing a new message-signature pair. This is called *existential forgery.* In many cases this attack is not dangerous, because the output message is likely to be meaningless. Nevertheless, a signature scheme which is not existentially unforgeable does not guarantee by itself the identity of the signer. For example, it cannot be used to certify randomly looking elements, such as keys.

Definition 2. (Secure Signature Scheme). A signature scheme is *secure* if an existential forgery is computationally impossible, even under an adaptively chosen-message attack.

The first secure signature scheme was proposed by Goldwasser et al. [27] in 1984. It uses the notion of claw-free permutations pairs: informally, these are permutations f_0 and f_1 over a common domain for which it is computationally infeasible to find a triple (x, y, z) such that $f_0(x) = f_1(y) = z$. Furthermore, Goldwasser et al. proved that such "claw-free" permutations pairs exist if factoring is hard (see [27] and [28] for details).

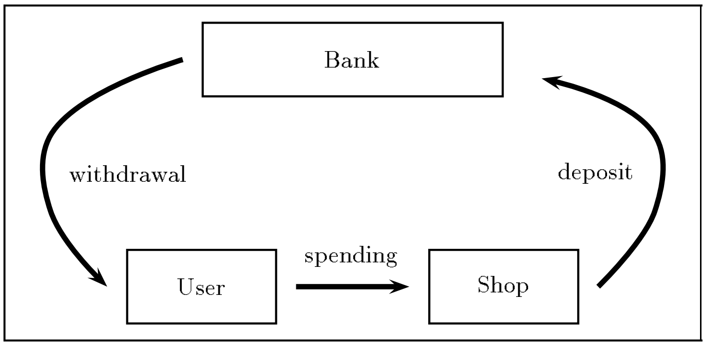


Fig. 3. Coin life.

1.3. *Blind Signatures*

After this brief outline of signature schemes, we review another cryptographic primitive: blind signatures. We first motivate their use and give some well-known examples. We then define specific security properties of blind signatures related to the setting of electronic cash.

1.3.1. *Motivation: Electronic Cash*

As early as 1982, Chaum's [13] pioneering work aimed at creating an electronic version of money. To achieve this goal, he introduced the notions of "coins" and "randomized blind signatures" (or simply "blind signatures"). He claimed that this was the only way to ensure the required *anonymity*: in real life, a coin cannot be easily traced from the bank to the shop, furthermore, two spendings of a same user cannot be linked together. These are two main properties of real coins that Chaum wanted to mimic: *untraceability* and *unlinkability*.

He proposed to define an *electronic coin* as a number with a certificate (a signature) produced by the bank; it is withdrawn from the bank, spent by the user, and deposited by the shop (see Fig. 3).

On-line electronic cash. In his first scheme, Chaum used blind signatures for the production of coins. The user makes the bank blindly sign a coin. Then the user is in possession of a valid coin that the bank itself cannot recognize nor link with the user. When the user spends the coin, the shop immediately returns it to the bank. If the coin has already been spent, the bank detects the fact and informs the shop so that it refuses payment. It is an "on-line" context: there is a continuous communication between the shop and the bank in order to verify the validity of coins. In order to define the scheme, Chaum introduced the first blind signature scheme, based on the RSA hypothesis. It is a by now classical transformation of the original RSA signature scheme [49]:

The Blind RSA Signature. The bank has a large composite number $N = pq$, a public key e, and a related secret key d. It also uses a public hash function H. In order to get the signature of a random number ρ, the user "blinds" it with a random value $r^e \bmod N$, and sends $m = H(\rho)r^e \bmod N$ to the signer. The latter returns a signature σ' of m such that $\sigma'^e = m = r^e H(\rho) \bmod N$. Then the user can "unblind" this signature computing $\sigma = \sigma' r^{-1} \bmod N$. A coin is any pair (ρ, σ) which satisfies $\sigma^e = H(\rho) \bmod N$.

In this scheme all coins have the same value, but in a real system different denominations might be encoded by different exponents e.

Off-line electronic cash and the "cut-and-choose" methodology. In an "off-line" context we cannot prevent a user from spending a coin twice or even more, since the detection is made too late to refuse payment. This fraud is called "double-spending." We only can hope that the double-spender will be discovered later and punished. Chaum et al. [14] were able to build such schemes by introducing the identity of the user in the coin in such a way that it remains concealed, unless double-spending happens. Once, blind signatures were a critical point for anonymity, and, as before, the authors used the blind RSA signature, together with the "cut-and-choose" technique: in their proposition, a coin is a kind of list of k blind signatures, each having an embedded copy of the identity of the user. To be sure that double-spending will reveal the real identity of the user, the bank would like to verify that the signatures actually have the requested format, which would revoke anonymity. Then the bank helps the user to get $2k$ signatures, randomly chooses k of them, and verifies the inner structure of the selected signatures. Since these signatures are no longer anonymous, the user throws them away and constructs the coin with the k other ones. The probability for a cheater to be finally in possession of a fraudulous coin is about 2^{-2k}.

The main drawback of the "cut-and-choose" technique is that the coins are very large, as well as the amount of computations. In 1993 Ferguson [20] and Brands [7] proposed new schemes without "cut-and-choose." The first one uses once again the blind RSA signature, whereas Brands' scheme uses a new blind signature derived from the Schnorr signature scheme [50], [51]:

The Blind Schnorr Signature. The generation algorithm produces two large prime integers p and q such that $q \,|\, p-1$ as well as an element g of $\mathbb{Z}_p^\star$ of order q. It also creates a pair of keys, (x, y), where $x \in \mathbb{Z}_q^\star$ is the secret one, and $y = g^{-x} \bmod p$ is the public one. The signer publishes y. In order to get the signature of a secret message m, the user asks the signer to initiate a communication. He chooses a random $K \in \mathbb{Z}_q^\star$, computes and sends the "commitment" $r = g^K \bmod p$. The user then blinds this value with two random elements $\alpha, \beta \in \mathbb{Z}_q$, into $r' = r g^{-\alpha} y^{-\beta} \bmod p$, computes the value $e' = H(m, r') \bmod q$ and sends the "challenge" $e = e' + \beta \bmod q$ to the signer who returns the value s such that $g^s y^e = r \bmod p$. Finally, the user computes $s' = s - \alpha \bmod q$. This way, the pair (e', s') is a valid Schnorr signature of m since it satisfies $e' = H(m, g^{s'} y^{e'} \bmod p)$.

In both schemes Ferguson and Brands managed to hide the identity of the user in a much more efficient way than the "cut-and-choose" methodology. Again, the identity is revealed after double-spending. Those blind signatures which hide a specific structure, such as the identity, are called "restrictive blind signatures" [11], [9], [8], [45]. Many extensions [19], [6], [10] have been proposed, followed by some attacks [8], [11] and repairs [9], [52]. All of them use blind signatures, and the security of the proposed schemes is totally dependent on the security of the blind signatures they use. Surprisingly, no security proofs have been proposed so far for blind signatures.

Revokable Anonymity A few years ago [57], an undesirable feature of total anonymity in transactions was considered: perfect crimes (anonymous crimes without leaving any traces and consequently without any risk of being suspected later). Accordingly, a new line of research in electronic cash has investigated "revokable anonymity" [12], [22], [31] which proposes anonymity unless a Trusted Third Party (TTP) partially revokes it for some established reasons or in view of an obvious fraud (e.g. in case of double-spending). Again, those new schemes rely on the security of blind signature schemes.

1.3.2. *Security*

As far as we know, no formal notion of security has ever been studied, or proved, in the context of blind signatures. However, it is a critical point in electronic cash systems. In the context of blind signatures, the previous definitions of security are no longer significant. In fact, existential forgery is somehow the basis for blind signatures. Nevertheless, a fundamental property for electronic cash systems is the guarantee that a user cannot forge more coins than the bank gives him. In other words, with ℓ blind signatures of the Bank, the user must not be able to create more than ℓ coins. This form of security was more or less informally assumed in connection with several schemes, for example in [10], or under the "unexpandability" property of [23].

Definition 3 (The $(\ell,\ \ell+1)$-Forgery). For any integer ℓ, an $(\ell,\ \ell+1)$-forgery comes from an attacker that produces $\ell+1$ signatures after ℓ interactions with the signer Σ.

Definition 4 (The "One-More" Forgery). For some integer ℓ, polynomial in the security parameter k, an attacker can obtain $\ell+1$ valid signatures after fewer than ℓ interactions with the signer. In other words, a *"one-more forgery"* is an $(\ell,\ \ell+1)$-forgery for some polynomially bounded integer ℓ.

Definition 5 (The Strong "One-More" Forgery). An $(\ell,\ell+1)$-forgery for a polylogarithmically bounded integer ℓ (*i.e.*, for some constant α, $\ell \leq (\log k)^\alpha$, where k is the security parameter) is called a *strong "one-more" forgery.*

As usual, several scenarios can be envisioned. We focus on two kinds of attacks which naturally come from the use of blind signatures in electronic cash :

- The *sequential attack* (see Fig. 4): the attacker interacts sequentially with the signer. This attack can be performed by a user who withdraws coins, one after the other.
  It is clear that, in practical situations, many users might be allowed to withdraw money at the same time. The following attack must then be considered.
- The *parallel attack* (see Fig. 5): the attacker interacts ℓ times in parallel with the signer. This attack is stronger. Indeed, the attacker can initiate new interactions with the signer before previous ones have ended. This attack can be performed by a group of users who withdraw many coins at the same time.

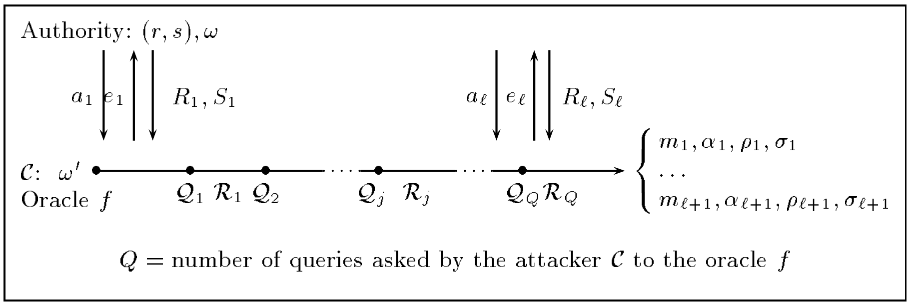


Fig. 4. The sequential attack.

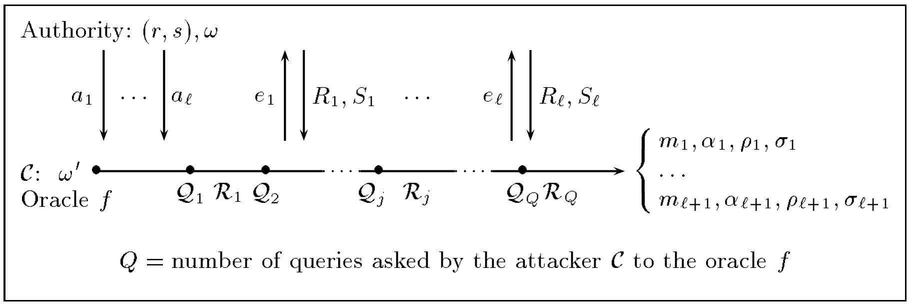


Fig. 5. The parallel attack.

2. Preliminaries

2.1. *Complexity Theory and "Oracle Replay Attack"*

In this paper we offer several security arguments for digital signatures and blind signatures. All our results are given in the context of complexity theory. Hence, any participant is modeled by a probabilistic polynomial time Turing machine. Our paradigm is to use a supposedly efficient attacker in order to solve a difficult algorithmic problem. This goes through a generic reduction technique (see Fig. 6) which we call the *oracle replay attack* (see Fig. 7): by a polynomial replay of the attack with different random oracles (the $\mathcal{Q}_i$'s are the queries and the ρ_i's are the answers), we make the attacker successfully forge signatures which are suitably related. More precisely, we want to obtain two signatures (σ_1, h, σ_2) and $(\sigma_1', h', \sigma_2')$ of an identical message m such that $\sigma_1 = \sigma_1'$, but $h \neq h'$. We then extract the solution of a difficult problem from the ability to forge such pairs. In the reductions, an important problem is to simulate properly the interactions that the attacker should have with other entities (with the random oracle f and particularly with the signer Σ). Those simulations should be indistinguishable from real interactions from the point of view of the attacker despite the obvious fact that no secret key is available.

2.2. *Distinguishability of Distributions of Probability*

As explained above, in our reductions, we have to provide indistinguishable simulations: the communication tapes between the attacker and the simulator and

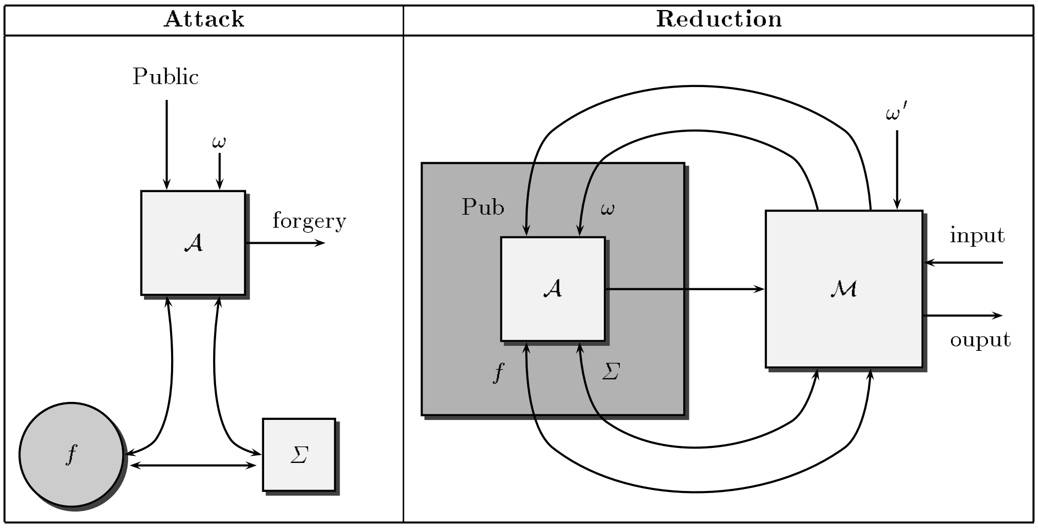


Fig. 6. Reduction of a difficult problem.

between the attacker and the signer (for example) will have to follow indistinguishable distributions of probability. In this subsection we define two notions of indistinguishability.

Recall that a function $f(k)$ is *negligible* in k if, for every polynomial p, $f(k)$ is smaller than $1/|p(k)|$, for k large enough; otherwise, it is *nonnegligible*.

Definition 6. Let δ^0 and δ^1 be two distributions of probability. A *distinguisher* $\mathcal{D}$ is a probabilistic polynomial time Turing machine, with random tape ω, which, on input ρ, answers 0 or 1.
The *advantage* of $\mathcal{D}$ with respect to two distributions δ^0 and δ^1 is defined as

$$Adv(\mathcal{D}, \delta^0, \delta^1) = \frac{1}{2} \times \left| \underset{\rho \in \delta^0}{E} [\mathcal{D}(\omega, \rho)] - \underset{\rho \in \delta^1}{E} [\mathcal{D}(\omega, \rho)] \right|.$$

It is easy to derive the following equality:

$$\Pr_{\substack{\omega \\ c \in \{0,1\} \\ \rho \in \delta^c}} [\mathcal{D}(\omega, \rho) = c] = \frac{1}{2} \pm Adv(\mathcal{D}, \delta^0, \delta^1).$$

So, if this advantage is negligible, the answer of the distinguisher looks like the result of flipping a coin.

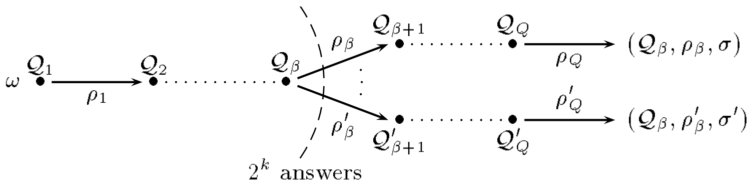


Fig. 7. The oracle replay attack.

Two distributions δ^0 and δ^1 are *polynomially indistinguishable* if there does not exist any distinguisher $\mathcal{D}$ with a nonnegligible advantage.

Two distributions δ^0 and δ^1 are *statistically indistinguishable* if

$$\sum_y \left| \Pr_{x\in\delta^0}[x=y] - \Pr_{x\in\delta^1}[x=y] \right| \text{ is negligible.}$$

Remark 1. It is clear that if two distributions are *statistically indistinguishable*, they are *polynomially indistinguishable.*

2.3. *The Splitting Lemma*

Throughout this paper we repeatedly use the "Splitting Lemma" below. It translates the fact that when a subset A is "large" in a product space $X \times Y$, it has many "large" sections.

Lemma 1. *(The Splitting Lemma). Let* $A \subset X \times Y$ *such that* $\Pr[(x,y)\in A] \geq \varepsilon$. *For any* $\alpha < \varepsilon$, *define*

$$B = \left\{ (x,y) \in X \times Y \;\middle|\; \Pr_{y'\in Y}[(x,y')\in A] \geq \varepsilon - \alpha \right\} \quad \textit{and} \quad \bar{B} = (X\times Y)\backslash B,$$

then the following statements hold:

(i) $\Pr[B] \geq \alpha$
(ii) $\forall (x,y)\in B, \Pr_{y'\in Y}[(x,y')\in A] \geq \varepsilon - \alpha$.
(iii) $\Pr[B \mid A] \geq \alpha/\varepsilon$.

Proof. In order to prove statement i), we argue by contradiction.
Assume that $\Pr[B] < \alpha$. Then

$$\varepsilon \;\leq\; \Pr[B]\cdot\Pr[A\mid B] + \Pr[\bar{B}]\cdot\Pr[A\mid\bar{B}] \;<\; \alpha\cdot 1 + 1\cdot(\varepsilon-\alpha) \;=\; \varepsilon.$$

This implies a contradiction, hence the result.

Statement (ii) is a straightforward consequence of the definition.

We finally turn to the last assertion, using Bayes' law:

$$\begin{aligned}\Pr[B\mid A] &= 1 - \Pr[\bar{B}\mid A]\\ &= 1 - \Pr[A\mid\bar{B}]\cdot\Pr[\bar{B}]/\Pr[A] \geq 1 - (\varepsilon-\alpha)/\varepsilon = \alpha/\varepsilon.\end{aligned}$$

□

3. Security Arguments for Digital Signatures

This section is devoted to digital signatures and extends our previous results on their security [43]. Recall that an identification scheme [21] is an interactive protocol which involves a prover and a verifier. The prover tries to convince the verifier of his knowledge of a secret related to his identity. More specifically, a three-pass honest-verifier zero-knowledge identification protocol is an identification scheme with three interactions between the prover and the verifier, which leaks no information about the secret provided the verifier plays honestly, namely randomly choosing his queries. The three interactions correspond to three messages: the "commitment" a sent by the prover, the "challenge" e randomly chosen by the verifier, and the "answer" r of the prover. The verifier finally accepts the proof if and only if this triple satisfies a test $V(a, e, r) = 1$. As described by Fiat and Shamir [21], any three-pass honest-verifier zero-knowledge identification protocol can be turned into a generic digital signature scheme: let (a, e, r) be a round of the identification protocol, we get a digital signature scheme by replacing the query of the verifier, which is a random value e, by the hash value of the message m to be signed together with the commitment a which is bound not to change, namely, $e = f(m, a)$, where f is the hash function. If the identification protocol needs several sequential iterations in order to reach an adequate level of security, then, in the signature setting, one parallelizes the protocol. Accordingly, a signature of a message m is a triple (σ_1, h, σ_2), where σ_1 represents all successive "commitments" of the parallelized protocol, $h = f(m, \sigma_1)$ and σ_2 represents all successive "answers" of the parallelized protocol. It satisfies a test $V(\sigma_1, h, \sigma_2) = 1$ as described above in the generic digital signature schemes section (see Section 1.2.2.). For example, the Schnorr signature scheme is precisely the result of the above transformation applied to the Schnorr identification protocol.

In what follows, we assume that f outputs k-bit long elements, where k is the security parameter of the signature scheme, as described above.

We first prove the security of a generic digital signature scheme against no-message attacks. As an application, we directly obtain the security of the Schnorr signature scheme. Next, we extend our result to the adaptively chosen-message context. We close the section with a study of the El Gamal signature scheme [17]: in spite of the existential forgery of the original scheme, we present a slight variation which is existentially unforgeable under an adaptively chosen-message attack. This is provided that the discrete logarithm problem is hard to solve.

3.1. *No-Message Attacks*

In this part we consider the no-message scenario. We propose a generic result and we apply our technique to the Schnorr signature scheme.

3.1.1. *Generic Results*

Lemma 2. *Let $(\mathcal{G}, \Sigma, V)$ be a generic digital signature scheme with security parameter k. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data and which can ask Q queries to the random oracle, with $Q > 0$. We assume that, within the time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 7Q/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$. Then, within time $T' \leq 16QT/\varepsilon$, and with probability $\varepsilon' \geq \frac{1}{9}$, a replay of this machine outputs two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$.*

Proof. We start with a no-message attacker $\mathcal{A}$, which is a probabilistic polynomial time Turing machine with 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 and let $\rho = (\rho_1, \ldots, \rho_Q)$ be the list of the Q answers of f. It is clear that a random choice of f exactly corresponds to a random choice of ρ. Then, for a random choice of (ω, f), with probability ε, $\mathcal{A}$ outputs a valid signature $(m, \sigma_1, h, \sigma_2)$. Since f is a random oracle, it is easy to see that the probability for h to be equal to $f(m, \sigma_1)$ is less than $1/2^k$, unless it has been asked during the attack. So, it is likely that the question (m, σ_1) is actually asked during a successful attack. Accordingly, we define $\mathit{Ind}(\omega, f)$ to be the index of this question: $(m, \sigma_1) = \mathcal{Q}_{\mathit{Ind}(\omega,f)}$ (we let $\mathit{Ind}(\omega, f) = \infty$ if the question is never asked). We then define the sets

$$\begin{aligned} &\mathcal{S} = \{(\omega, f) \mid \mathcal{A}^f(\omega) \text{ succeeds } \& \ \mathit{Ind}(\omega, f) \neq \infty\}, \\ \text{and } &\mathcal{S}_i = \{(\omega, f) \mid \mathcal{A}^f(\omega) \text{ succeeds } \& \ \mathit{Ind}(\omega, f) = i\} \quad \text{for} \quad i \in \{1, \ldots, Q\}. \end{aligned}$$

We call $\mathcal{S}$ the set of the successful pairs (ω, f), and we note that the set $\{\mathcal{S}_i \mid i \in \{1, \ldots, Q\}\}$ is a partition of $\mathcal{S}$. With those definitions, we find a lower bound for the probability of success, $\nu = \Pr[\mathcal{S}] \geq \varepsilon - 1/2^k \geq 6\varepsilon/7$. Let I be the set consisting of the most likely indices i, $I = \{i \mid \Pr[\mathcal{S}_i \mid \mathcal{S}] \geq 1/2Q\}$. The following lemma claims that, in case of success, the index lies in I with probability at least $\frac{1}{2}$.

Lemma 3. $\Pr[\mathit{Ind}(\omega, f) \in I \,\big|\, \mathcal{S}] \geq \frac{1}{2}$.

Proof. By definition of the sets $\mathcal{S}_i$, $\Pr[\mathit{Ind}(\omega, f) \in I \mid \mathcal{S}] = \sum_{i \in I} \Pr[\mathcal{S}_i \mid \mathcal{S}]$. This probability is equal to $1 - \sum_{i \notin I} \Pr[\mathcal{S}_i \mid \mathcal{S}]$. Since the complement of I contains fewer than Q elements, this probability is at least $1 - Q \times 1/2Q \geq \frac{1}{2}$. □

We now run the attacker $2/\varepsilon$ times with random ω and random f. Since $\nu = \Pr[\mathcal{S}] \geq 6\varepsilon/7$, with probability greater than $1 - (1 - 6\varepsilon/7)^{2/\varepsilon}$, we get at least one pair (ω, f) in $\mathcal{S}$. It is easily seen that this probability is lower bounded by $1 - e^{-12/7} \geq \frac{4}{5}$.

We now apply the Splitting-lemma (lemma 1) for each integer $i \in I$: we denote by f_i the restriction of f to queries of index strictly less than i. Since $\Pr[\mathcal{S}_i] \geq \nu/2Q$, there exists a subset Ω_i of executions such that,

$$\begin{aligned} \text{for any } \ (\omega, f) \in \Omega_i, \ \ &\Pr_{f'}[(\omega, f') \in \mathcal{S}_i \mid f'_i = f_i] \geq \nu/4Q \\ &\Pr[\Omega_i \mid \mathcal{S}_i] \geq \tfrac{1}{2}. \end{aligned}$$

Since all the subsets $\mathcal{S}_i$ are disjoint,

$$\begin{aligned} &\Pr_{\omega,f}[(\exists i \in I) \ (\omega, f) \in \Omega_i \cap \mathcal{S}_i \mid \mathcal{S}] \\ &\qquad = \Pr\left[\bigcup_{i \in I} (\Omega_i \cap \mathcal{S}_i) \mid \mathcal{S}\right] = \sum_{i \in I} \Pr[\Omega_i \cap \mathcal{S}_i \mid \mathcal{S}] \\ &\qquad = \sum_{i \in I} \Pr[\Omega_i \mid \mathcal{S}_i] \cdot \Pr[\mathcal{S}_i \mid \mathcal{S}] \geq \left(\sum_{i \in I} \Pr[\mathcal{S}_i \mid \mathcal{S}]\right) / 2 \geq \tfrac{1}{4}. \end{aligned}$$

We let β denote the index $\mathit{Ind}(\omega, f)$ corresponding to the successful pair. With probability at least $\frac{1}{4}$, $\beta \in I$ and $(\omega, f) \in \mathcal{S}_\beta \cap \Omega_\beta$. Consequently, with probability greater than $\frac{1}{5}$, the $2/\varepsilon$ attacks have provided a successful pair (ω, f), with $\beta = \mathit{Ind}(\omega, f) \in I$ and $(\omega, f) \in \mathcal{S}_\beta$. Furthermore, if we replay the attack, with fixed ω but randomly chosen oracle f' such that $f'_\beta = f_\beta$, we know that $\Pr_{f'}[(\omega, f') \in \mathcal{S}_\beta \mid f'_\beta = f_\beta] \geq \nu/4Q$. Then

$$\begin{aligned}\Pr_{f'}&[(\omega, f') \in \mathcal{S}_\beta \text{ and } \rho_\beta \neq \rho'_\beta \mid f'_\beta = f_\beta] \\ &\geq \Pr_{f'}[(\omega, f') \in \mathcal{S}_\beta \mid f'_\beta = f_\beta] - \Pr_{f'}[\rho'_\beta = \rho_\beta] \geq \nu/4Q - 1/2^k \geq \varepsilon/14Q,\end{aligned}$$

where $\rho_\beta = f(\mathcal{Q}_\beta)$ and $\rho'_\beta = f'(\mathcal{Q}_\beta)$. We us replay the attack $14Q/\varepsilon$ times with a new random oracle f' such that $f'_\beta = f_\beta$. With probability greater than $\frac{3}{5}$, we get another success.

Finally, after less than $2/\varepsilon + 14Q/\varepsilon$ repetitions of the attack, with probability greater than $\frac{1}{5} \times \frac{3}{5} \geq \frac{1}{9}$, we have obtained two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m', \sigma'_1, h', \sigma'_2)$ with $\mathcal{Q}_\beta = (m, \sigma_1) = (m', \sigma'_1)$ and distinct challenges $h = f(\mathcal{Q}_\beta) \neq f'(\mathcal{Q}_\beta) = h'$. □

The careful reader has noticed that the mechanics of our reduction depend on some parameters related to the attacker $\mathcal{A}$, namely, its probability of success ε and the number Q of queries to the random oracle. This induces a lack of uniformity. In order to overcome this problem, we can use the reduction technique presented in our previous Eurocrypt '96 paper [43]. Unfortunately, the probability of success of the resulting reduction is much smaller: the expected time of success is of the order of Q^4/ε^3 instead of Q/ε. Accordingly, we end up extremely far from any form of the "exact security" concept [3].

It is better to see the resulting machine $\mathcal{M}$ as an expected polynomial time Turing machine:

1. $\mathcal{M}$ initializes $j = 0$;
2. $\mathcal{M}$ runs $\mathcal{A}$ until it outputs a successful pair $(\omega, f) \in \mathcal{S}$ and denotes by N_j the number of calls to $\mathcal{A}$ to obtain this success, and by β the index $\mathit{Ind}(\omega, f)$;
3. $\mathcal{M}$ replays, at most $140 N_j \alpha^j$ times, $\mathcal{A}$ with fixed ω and random f' such that $f'_\beta = f_\beta$, where $\alpha = \frac{8}{7}$;
4. $\mathcal{M}$ increments j and returns to 2, until it gets a successful forking.

For any execution of $\mathcal{M}$, we denote by J the last value of j and by N the total number of calls to $\mathcal{A}$. We want to compute the expectation of N. Since $\nu = \Pr[\mathcal{S}]$, and $N_j \geq 1$, then $\Pr[N_j \geq 1/5\nu] \geq \frac{3}{4}$. We define $\ell = \lceil \log_\alpha Q \rceil$, so that, $140 N_j \alpha^j \geq 28Q/\varepsilon$ for any $j \geq \ell$, whenever $N_j \geq 1/5\nu$. Therefore, for any $j \geq \ell$, when we have a first success in $\mathcal{S}$, with probability greater than $\frac{1}{4}$, the index $\beta = \mathit{Ind}(\omega, f)$ is in the set I and $(\omega, f) \in \mathcal{S}_\beta \cap \Omega_\beta$. Furthermore, with probability greater than $\frac{3}{4}$, $N_j \geq 1/5\nu$. Therefore, with the same conditions as before, that is $\varepsilon \geq 7Q/2^k$, the probability of getting a successful fork after at most $28Q/\varepsilon$ iterations at step 3 is greater than $\frac{6}{7}$.

For any $t \geq \ell$, the probability for J to be greater or equal to t is less than $(1 - \frac{1}{4} \times \frac{3}{4} \times \frac{6}{7})^{t-\ell}$, which is less than $\gamma^{t-\ell}$, with $\gamma = \frac{6}{7}$. Furthermore,

$$E[N \mid J = t] \leq \sum_{j=0}^{j=t} \left(E[N_j] + 140 E[N_j] \alpha^j\right) \leq \frac{141}{\nu} \times \sum_{j=0}^{j=t} \alpha^j \leq \frac{141}{\nu} \times \frac{\alpha^{t+1}}{\alpha - 1}.$$

– Initialization (security parameter k)	– Signature
p, q, two large primes such that $q \mid (p-1)$, $2^{k-1} \leq q < 2^k$	– $K \in \mathbb{Z}_q^\star$
g, element of $\mathbb{Z}_p^\star$ of order q	– $r = g^K \bmod p$
f, hash function	– $e = f(m,r)$
secret key $x \in \mathbb{Z}_q^\star$	– $s = K + xe \bmod q$
public key $y = g^{-x} \bmod p$	– $\sigma_1 = r$ and $\sigma_2 = s$
	– Verification
	– $e \stackrel{?}{=} f(m,r)$
	– $r \stackrel{?}{=} g^s y^e \bmod p$

Fig. 8. The Schnorr signature scheme.

So, the expectation of N is $E[N] = \sum_t E[N \mid J = t] \cdot \Pr[J = t]$ and then is

$$\begin{aligned}
&\leq \frac{141}{\nu} \times \sum_t \left(\frac{\alpha^{t+1}}{\alpha - 1}\right) \times \Pr[J \geq t] \\
&\leq \frac{165}{\varepsilon} \cdot \left[\sum_{t=0}^{t=\ell-1} \left(\frac{\alpha^{t+1}}{\alpha-1}\right) + \sum_{t \geq \ell} \left(\frac{\alpha^{t+1}}{\alpha-1}\right) \times \gamma^{t-\ell}\right] \\
&\leq \frac{165}{\varepsilon} \cdot \frac{\alpha^{\ell+1}}{\alpha-1} \cdot \left[\frac{1}{\alpha-1} + \sum_t (\alpha\gamma)^t\right] \\
&\leq \frac{165}{\varepsilon} \cdot \frac{\alpha^{\ell+1}}{\alpha-1} \cdot \left(\frac{1}{\alpha-1} + \frac{1}{1-\alpha\gamma}\right).
\end{aligned}$$

Using the definition of ℓ and the values of α and γ, we obtain

$$E[N] \leq \frac{165}{\varepsilon} \cdot \frac{64Q}{7} \cdot (7 + 49) = \frac{84480Q}{\varepsilon}.$$

Hence the following theorem.

Theorem 1. *(The Forking Lemma). Let $(\mathcal{G}, \Sigma, V)$ be a generic digital signature scheme with security parameter k. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote by Q the number of queries that $\mathcal{A}$ can ask to the random oracle. Assume that, within time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 7Q/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$. Then there is another machine which has control over $\mathcal{A}$ and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$, in expected time $T' \leq 84480TQ/\varepsilon$.*

3.1.2. *The Schnorr Digital Signature Scheme*

We now apply the previous result in the simple setting of the Schnorr signature scheme (see Fig. 8).

Firstly, we briefly describe the protocol. For any security parameter k, an authority chooses two large prime integers p and q, such that $2^{k-1} \leq q < 2^k$ holds and q divides $p - 1$ as well as an element g from $\mathbb{Z}_p^\star$ of order q. The triple (p, q, g) is published together with a public hash function f whose output domain is identified to $\mathbb{Z}_q^\star$. The security parameter k is then equal to $\lceil \log q \rceil$, whereas the size of the public key, denoted by n, is equal to $\lceil \log p \rceil$. Furthermore, we assume that $k \gg \log n$. Any user randomly chooses his secret key x in $\mathbb{Z}_q^\star$, and publishes $y = g^{-x} \bmod p$.

In order to sign a message m, the user chooses a random element K in $\mathbb{Z}_q^\star$ and computes the commitment $r = g^K \bmod p$. He gets the challenge $e = f(m, r)$ and computes $s = K + xe \bmod q$. The signature is the triple (r, e, s), which satisfies the tests $r \stackrel{?}{=} g^s y^e \bmod p$ and $e \stackrel{?}{=} f(m, r)$.

Theorem 2. *Assume that, within a time bound T, an attacker $\mathcal{A}$ performs an existential forgery under a no-message attack against the Schnorr signature, with probability $\varepsilon \geq 7Q/q$. We denote by Q the number of queries that $\mathcal{A}$ can ask to the random oracle. Then the discrete logarithm in subgroups of prime order can be solved in expected time less than $84480QT/\varepsilon$.*

Proof. As we have previously seen, this scheme satisfies all the required properties of a generic signature scheme. From the Forking Lemma (Theorem 1), after a polynomial replay of the attacker $\mathcal{A}$, we obtain two valid signatures (m, r, e, s) and (m, r, e', s') with $e \neq e'$. Then we have the following equalities $r = g^s y^e \bmod p$ and $r = g^{s'} y^{e'} \bmod p$, from which we obtain the discrete logarithm $\log_g y = (s - s')/(e' - e) \bmod q$. □

3.2. *Adaptively Chosen-Message Attacks*

We now focus on the adaptively chosen-message scenario. As in the previous section, we first give a generic result and we apply the technique to the Schnorr signature scheme.

3.2.1. *Generic Results*

As was previously observed, in a no-message scenario, only the least powerful kind of adversaries is assumed to attack the signature scheme. For many applications, resistance to this type attack is not considered sufficient. If we want to assess the "security" of a signature scheme, we should prove its resistance against adaptively chosen-message attacks. In such a scenario, the attacker uses the signer as an oracle, and asks any signature he wants. If it is possible to simulate the signer Σ by a simulator $\mathcal{S}$ who does not know the secret key (see Fig. 9), 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 obtain two distinct signatures with a suitable relation.

Lemma 4. *Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and R the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signer. Assume that, within a time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 10(R+1)(R+Q)/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then, a replay of the attacker $\mathcal{A}$, where interactions*

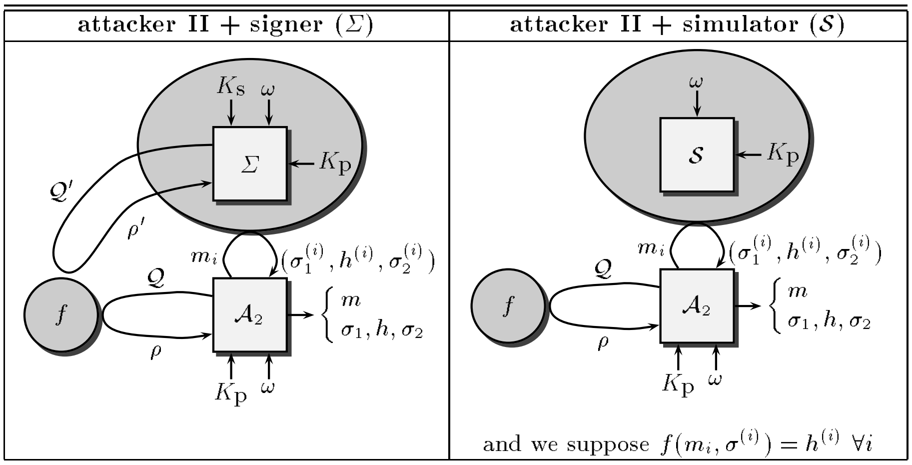


Fig. 9. Adaptively chosen message scenario.

with the signer are simulated, outputs two valid signatures $(m, \sigma_1, h, \sigma_2)$ *and* $(m, \sigma_1, h', \sigma_2')$ *such that* $h \neq h'$, *within time* $T' \leq 23QT/\varepsilon$ *and with probability* $\varepsilon' \geq \frac{1}{9}$.

Proof. As in the previous proof, we let $\mathcal{Q}_1, \ldots, \mathcal{Q}_Q$ denote the Q distinct queries to the random oracle, $\rho_1, \ldots, \rho_Q$ the respective answers, and $m_1, \ldots, m_R$ the R queries (possibly all the same) to the signing oracle. Using the simulator, we can simulate the answers of the signer without knowledge of the secret key. For a message m_i, the simulator answers a triple $(\sigma_1^{(i)}, h^{(i)}, \sigma_2^{(i)})$. Then, the attacker assumes that $f(m_i, \sigma_1^{(i)}) = h^{(i)}$ and stores it. The previous proof can be exactly mimicked, except for the problem added by the simulations: there is some risk of "collisions" of queries, or supposed queries, to the random oracle. Recall that in the definition of generic digital signature schemes, we made the assumption that the probability for a "commitment" $\sigma_1^{(i)}$ to be output by the signing oracle is less than $2/2^k$. Then, two kinds of collisions can appear:

- A pair $(m_i, \sigma_1^{(i)})$ that the simulator outputs also appears in the list of questions asked to the random oracle by the attacker (some question $\mathcal{Q}_j$). The probability of such an event is less than $QR \times 2/2^k \leq \varepsilon/5$.
- A pair $(m_i, \sigma_1^{(i)})$ that the simulator outputs is exactly similar to another pair produced by this simulator (some question $(m_j, \sigma_1^{(j)})$). The probability of such an event is less than $R^2/2 \times 2/2^k \leq \varepsilon/10$.

Altogether, the probability of collisions is less than $3\varepsilon/10$. Therefore,

$$\Pr_{\omega,f}[\mathcal{A} \text{ succeeds and no-collisions}] \geq \Pr_{\omega,f}[\mathcal{A} \text{ succeeds}] - \Pr_{\omega,f}[\text{collisions}] \;\geq\; \varepsilon(1 - \tfrac{3}{10}) \;\geq\; 7\varepsilon/10.$$

This is clearly greater than $7Q/2^k$. We can then apply the previous Forking Lemma (lemma 2). Such a replay succeeds with probability $\varepsilon' \geq \frac{1}{9}$, within time $T' \leq 16QT \times 10/7\varepsilon \leq 23QT/\varepsilon$. □

Theorem 3 (The Forking Lemma). *Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and R the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signer. Assume that, within a time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 10(R+1)(R+Q)/2^k$, a valid signature $(m, \sigma_1, h, \sigma_2)$. If the triples (σ_1, h, σ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from $\mathcal{A}$ replacing interaction with the signer by simulation and produces two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma_2')$ such that $h \neq h'$ in expected time $T' \leq 120686QT/\varepsilon$.*

Proof. The collusion of the attacker $\mathcal{A}$ and the simulator $\mathcal{S}$ defines a machine $\mathcal{B}$ which performs a no-message attack. An execution of $\mathcal{B}$ is successful if it outputs a forgery, and if there is no collisions of queries to the random oracle during the process. Then, within a time bound T, $\mathcal{B}$ has a probability of success greater than $7\varepsilon/10 \geq 7Q/2^k$. Using Theorem 1, within an expected number of steps bounded by $84480Q/(7\varepsilon/10)$, one can provide two valid signatures. □

3.2.2. *Application to the Schnorr Digital Signature Scheme*

Theorem 4. *Let $\mathcal{A}$ be an attacker which performs, within a time bound T, an existential forgery under an adaptively chosen-message attack against the Schnorr signature, with probability ε. We denote respectively by Q and R the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signing oracle. Assume that $\varepsilon \geq 10(R+1)(R+Q)/q$, then the discrete logarithm in subgroups of prime order can be solved within expected time less than $120686QT/\varepsilon$.*

Proof. We only have to prove that the triples (r, e, s) produced by the signer and the random oracle can be simulated without the knowledge of the signer's secret. Once this is done, the result directly follows from Theorem 3, using the same proof as for Theorem 2.

Lemma 5. *The following distributions are the same:*

$$\delta = \left\{ (r,e,s) \,\middle|\, \begin{array}{c} K \in_R \mathbb{Z}_q^\star \\ e \in_R \mathbb{Z}_q \\ r = g^K \ mod\ p \\ s = K + xe \ mod\ q \end{array} \right\} \quad and \quad \delta' = \left\{ (r,e,s) \,\middle|\, \begin{array}{c} K \in_R \mathbb{Z}_q \\ e \in_R \mathbb{Z}_q \\ s = K \\ r = g^s y^e \ mod\ p \\ r \neq 1 \ mod\ p \end{array} \right\}.$$

Proof. First we choose a triple $(\varepsilon, \beta, \gamma)$ from the set of the signatures: let $\varepsilon \in \mathbb{Z}_p^\star$, $\gamma \in \mathbb{Z}_q$ and $\beta \in \mathbb{Z}_q$ such that $g^\gamma y^\beta = \varepsilon \neq 1 \bmod p$. We then compute the probability of appearance of this triple following each distribution of probabilities:

$$\begin{aligned} \Pr_\delta[(r,e,s) = (\varepsilon,\beta,\gamma)] &= \Pr_{K \neq 0, e}\begin{bmatrix} g^K = \varepsilon; e = \beta \\ K + xe = \gamma \end{bmatrix} = \frac{1}{q(q-1)} \\ \Pr_{\delta'}[(r,e,s) = (\varepsilon,\beta,\gamma)] &= \Pr_{r,e}\left[\begin{array}{c} \varepsilon = r = g^K y^e \\ e = \beta; s = K = \gamma \end{array} \,\middle|\, r \neq 1 \bmod p \right] \\ &= \frac{1}{q(q-1)}. \end{aligned}$$

□

From the above, the following simulator $\mathcal{S}$ produces triples (r, e, s) with an identical distribution from those produced by the signer. In order to sign the message m, $\mathcal{S}$ randomly chooses $e \in \mathbb{Z}_q$ and $K \in_R \mathbb{Z}_q$, and sets $r = g^K y^e \bmod p$ and $s = K$. In the (unlikely) situation where $r = 1 \bmod p$, we discard the results and restart the simulation. Then it returns the triple (r, e, s). □

3.2.3. *Further Results*

It is clear that identical results can be obtained for any signature scheme which is the transformation of a honest-verifier zero-knowledge identification protocol, and *a fortiori* of the parallelization of a zero-knowledge identification protocol (Fiat-Shamir [21], Guillou-Quisquater [30], the Permuted Kernel Problem [53], the Syndrome Decoding problem [54], the Constrained Linear Equations [55], the Permuted Perceptrons Problem [41], etc.). In fact, the zero-knowledge property is exactly what we need for our notion of simulation. For each of these schemes, existential forgery under an adaptively chosen-message attack in the random oracle model is equivalent to the mathematical problem on which the identification scheme relies. Furthermore, our results may also provide security arguments for other schemes. In the following section we study a signature scheme of the El Gamal type.

3.3. *Application to the El Gamal Signature Scheme*

The original El Gamal signature scheme [17] was proposed in 1985 but its security was never proved equivalent to the discrete logarithm problem nor to the Diffie-Hellman problem. As will be seen, the Forking Lemma provides a security argument for a very slight variant of this scheme.

3.3.1. *The Original Scheme*

Description of the original scheme. We begin with a description of the original scheme [17], where k denotes, as usual, the security parameter:

- The key generation algorithm: it chooses a random large prime p, of length n polynomial in k, and a generator g of $\mathbb{Z}_p^\star$, both public. Then, for a random secret key $x \in \mathbb{Z}_{(p-1)}$, it computes the public key $y = g^x \bmod p$.
- The signature algorithm: in order to sign 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 $K \in \mathbb{Z}_{(p-1)}^\star$, to compute the exponentiation $r = g^K \bmod p$ and finally to solve the linear equation $m = xr + Ks \bmod (p-1)$. The algorithm finally outputs (r, s).
- The verification algorithm checks both $1 < r < p$ and $g^m = y^r r^s \bmod p$.

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 5. *The original El Gamal signature scheme is existentially forgeable.*

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

1. The one-parameter forgery: let $e \in_R \mathbb{Z}_{(p-1)}$, 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 $e \in_R \mathbb{Z}_{(p-1)}$ and $v \in_R \mathbb{Z}^{\star}_{(p-1)}$, 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 slightly modify the original scheme by using a hash function.

3.3.2. *The Modified El Gamal Signature Scheme – MEG*

In this variant we replace m by the hash value of the entire part of the computation bound not to change, once the commitment has been computed, namely $f(m, r)$, where f is a public hash function which outputs k-bit long elements.

Definition 7. 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 $R \leq |p|^{\alpha}$, where $|p|$ denotes the length of the integer p.

Remark 2. Those prime moduli are precisely those used for cryptographic applications of the discrete logarithm problem.

We describe the Modified El Gamal Signature Scheme:

- The key generation algorithm: it chooses a random large α-hard prime p, greater than 2^k, of length n polynomial in k. It also randomly chooses a generator g of $\mathbb{Z}^{\star}_p$. They are both published. Then, for a random secret key $x \in \mathbb{Z}_{(p-1)}$, it computes the public key $y = g^x \bmod p$.
- The signature algorithm: in order to sign a message m, one generates a pair (r, s) such that $g^{f(m,r)} = y^r r^s \bmod p$. 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.

3.3.3. *Security Results*

In this section we see that the above modification allows us to offer security arguments for the resulting scheme even against an adaptively chosen-message attack, at least for a large variety of moduli.

Security against a no-message attack. Firstly, we study the resistance of the MEG signature scheme against no-message attacks.

Theorem 6. *Consider a no-message attack in the random oracle model against the MEG signature scheme using α-hard prime moduli. Probabilities are taken over the common generator g, random tapes, random oracles and the public key y. If an existential forgery has nonnegligible probability of success, then the discrete logarithm problem with α-hard prime moduli can be solved in polynomial time for any pair (g, y).*

Proof. Using the Forking Lemma (Theorem 1), 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 $\mathbb{Z}_p^\star$, there exist t and x such that $g^t = r \bmod p$ and $g^x = y \bmod p$. Therefore,

$$hs' - h's = xr(s' - s) \bmod p - 1, \quad (1)$$

$$h' - h = t(s - s') \bmod p - 1. \quad (2)$$

Since h and h' come from "oracle replay", we may further assume that $h - h'$ 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, (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, (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 nonnegligible probability, $x \in \mathbb{Z}_{(p-1)}$ such that $y = g^x \bmod p$ (case 1) or $b \in \mathbb{Z}_R$ and $t \in \mathbb{Z}_{(p-1)}$ such that $bq = g^t \bmod p$ (case 2). Probabilities are taken over g, y, and the random tapes of $\mathcal{M}$. Using the Splitting-Lemma (Lemma 1), let $\mathcal{G}$ be a nonnegligible set of g's such that whenever $g \in \mathcal{G}$, the set of y's which provides the above witnesses is nonnegligible. 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 or equal to $\varepsilon/2$. 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 nonnegligible part of (g, y), the discrete logarithm of y relative 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 nonnegligible subset $G_{bad}(b)$ of bad g's such that, with nonnegligible 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 nonnegligible probability, some t such that $g^t = bq \bmod p$. Running $\mathcal{M}(yg^\ell, z')$, for random ℓ and z', we get, with nonnegligible 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 nonnegligible part of (g, y), the discrete logarithm of y relative to g.

Now, we fix g and y. Running the machine on (g^u, yg^v) with random u and v, we obtain, with nonnegligible 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. As we have seen before, the only thing we have to show is how the signer can be simulated.

Lemma 6. *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 the generation algorithm, on the security parameter k, outputs p and q such that $qR > 2^k \geq q$.

First, one can remark that we can easily compute $x \bmod R$, since R is polynomially bounded. Then, using the two-parameter forgery for the q modular part, and the $x \bmod R$ value for the other part, we can obtain an indistinguishable simulation: we first randomly choose $e \in \mathbb{Z}_q$ and $v \in \mathbb{Z}_q^\star$. We then randomly choose $\ell \in \mathbb{Z}_R^\star$. We let $r = (g^{Re} y^{Rv}) \times g^{q\ell} \bmod p$.

Therefore, one may remark that r is a generator of $\mathbb{Z}_p^\star$ if and only if $\gcd(eR + xvR + q\ell, p-1) = 1$. Since $\ell \in \mathbb{Z}_R^\star$, this greatest common divisor is equal to $\gcd(e + xv, q) \cdot \gcd(\ell, R)$ and so equals 1 with overwhelming probability. We start the simulation again in the (unlikely) situation where r is not a generator of $\mathbb{Z}_p^\star$. Our approach corresponds to dealing separately 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 want h and s to satisfy $h = xr + R(e + xv)s \bmod q$. Then we can set $s = -r(Rv)^{-1} \bmod q$ and $h = -erv^{-1} \bmod q$. For the R modular part, we randomly choose $h \bmod R$ until $h \in H$ (in a first step, h is uniformly distributed in $\mathbb{Z}_{qR}$, then h is uniformly distributed in H) and we compute $s = (h - rx)(q\ell)^{-1} \bmod R$. Then the triple (r, h, s) satisfies $g^h = y^r r^s \bmod p$, therefore it is a valid signature of a message m as soon as $h = f(m, r)$.

Let $(r, h, s) \in \mathbb{Z}_p^\star \times H \times \mathbb{Z}_{(p-1)}$ such that $g^h = r^s y^r \bmod p$ and r is a generator of $\mathbb{Z}_p^\star$. Then there exists a unique exponent K prime to q such that $r = g^K \bmod p$. So, exactly one execution of the signature algorithm can produce this triple. Trying to output this signature through our simulation yields the following system of equations where v and e are the unknowns:

$$\begin{cases} hv + re = 0 \bmod q, \\ \;\; xv + e = KR^{-1} \bmod q. \end{cases}$$

If $h \neq xr \bmod q$, the determinant is nonzero modulo q so that there is exactly one solution and therefore one way for $\mathcal{S}$ to generate such a signature. In the other case, $r^s = g^{h-xr} \bmod p$, so that $s = 0 \bmod q$. Furthermore, the first equation can be written $r(xv + e) = rK = 0 \bmod q$, so that $r = 0 \bmod q$. Since $h = xr \bmod q$, $h = 0 \bmod q$. Consequently, $\mathcal{S}$ can generate such a signature only if $r = h = s = 0 \bmod q$. In this case the system admits $q - 1$ solutions.

Since our simulation only outputs r which are generators, the latter case contributes to the overall distance by some term bounded by $R/2^k$ which is less than $n^{2\alpha+1}/2^n$, a negligible value, where $n = |p|$. □

Theorem 3 is then applicable, therefore we can state:

Theorem 7. *Consider an adaptively chosen-message attack in the random oracle model against MEG using α-hard prime moduli. Probabilities are taken over the common generator g, random tapes, random oracles and the public key y. If an existential forgery of this scheme has nonnegligible probability of success, then the discrete logarithm problem with α-hard prime moduli can be solved in polynomial time for any pair (g, y).*

3.3.4. *Remarks*

We conclude the section by the following two remarks.

Exact security. Because of the intricate reduction, we do not try to compute the complexity nor the expected time of the resulting discrete logarithm algorithm exactly. In any case, this reduction is rather inefficient and we cannot infer from it any form of "exact security" [3]. Accordingly, it cannot be used practically to infer the security of the MEG signature scheme. Nevertheless, it is the first security argument for a variant of the well-known El Gamal signature scheme and, as such, validates the design of this scheme.

The Bleichenbacher attack. At Eurocrypt '96, Bleichenbacher presented an attack [4] against the original El Gamal signature scheme which is also applicable to our variant. However as explained in [56], the apparent contradiction between our security arguments and this attack vanishes since our arguments are correct for almost all choices of the parameters whereas Bleichenbacher uses very specific values. More precisely, the MEG is secure provided not only the keys but also the generator g of $\mathbb{Z}_p^\star$ are chosen at random. Otherwise, there is some danger that a trapdoor has been added. Thus, a reasonable requirement would be that the authority issues some sort of proof that g has been fairly generated, as was suggested for the modulus p of the Digital Signature Standard [35].

4. Security Arguments for Blind Signatures

This last section investigates the possibility of designing provably secure blind signature schemes. As in the previous section, we present a generic lemma providing security arguments for blind signature schemes. This extends our previous results of [42].

We first describe a general format of the schemes to which those proofs apply. We then propose several schemes for which one can provide security arguments relative to the discrete logarithm problem or to RSA.

4.1. *Witness Indistinguishability*

Previous methods of proofs used to establish security arguments for signature schemes no longer work since, during the collusion between the signer, the attacker and the random oracle, we lose control over the value that the signer receives: it no longer comes from the random oracle, but from the attacker. As a consequence, the signer cannot be simulated without the secret key, otherwise the signature scheme would be universally forgeable.

In order to overcome this problem, we use the concept of the "witness indistinguishable" proofs. This notion was defined by Feige and Shamir in [18] for the purpose of identification. In such a proof system:

- Many secret keys are associated to a same public key.
- The views of two proofs using two distinct secret keys (witnesses) associated to a same public key are indistinguishable, even from the point of view of the verifier.
- The knowledge of two distinct secret keys associated to a same public one provides the solution of a difficult problem.

For example, in the Fiat-Shamir protocol [21], the verifier cannot distinguish which square root the prover uses, and with probability $\frac{1}{2}$, two distinct square roots provide the factorization of the modulus. Okamoto, in [37], proposed a witness indistinguishable adaptation of both the Schnorr [50] and the Guillou-Quisquater [30] identification schemes.

As was already remarked, the technical difficulty to be overcome comes from the fact that, in the colluding step, we can no longer simulate the signer without the secret key. We use a scheme which admits more than one secret key for a given public key. This makes the collusion possible and we constrain the attacker to output a different secret key.

Our candidate scheme is one of the schemes designed by Okamoto in [37]. For the reader's convenience, Okamoto's adaptation of the Schnorr scheme appears in Fig. 10.

4.2. *The Okamoto-Schnorr Blind Signature Scheme*

The scheme uses two large primes p and q such that $q \mid (p-1)$, and two elements $g, h \in \mathbb{Z}_p^\star$ of order q. The authority chooses a secret key $(r, s) \in (\mathbb{Z}_q^\star)^2$ and publishes the public key, $y = g^{-r} h^{-s} \bmod p$. We assume that the function f outputs

Prover		Verifier
	p and q are prime integers such that $q \mid (p-1)$ g and h are elements of $\mathbb{Z}_p^\star$ of order q	
secret : $r, s \in \mathbb{Z}_q$ public : $y = g^{-r}h^{-s} \bmod p$		
$t, u \in \mathbb{Z}_q$ $a = g^t h^u \bmod p$	$\xrightarrow{a}$	
	$\xleftarrow{c}$	$c \in \mathbb{Z}_{2^t}$
$R = t + cr \bmod q$ $S = u + cs \bmod q$	$\xrightarrow{R, S}$	
		$a \stackrel{?}{=} g^R h^S y^c \bmod p$

Fig. 10. The Okamoto adaptation of the Schnorr identification scheme

elements in $\mathbb{Z}_q$ and that $\lceil \log q \rceil = k$, where k is, as usual, the security parameter. The protocol (Fig. 11) by which the user obtains a blind signature of the message m is as follows:

Authority		User
	p and q are prime integers such that $q \mid (p-1)$ g and h are elements of $\mathbb{Z}_p^\star$ of order q	
secret : $r, s \in \mathbb{Z}_q$ public : $y = g^{-r}h^{-s} \bmod p$		
$t, u \in \mathbb{Z}_q$ $a = g^t h^u \bmod p$	$\xrightarrow{a}$	
		$\beta, \gamma, \delta \in \mathbb{Z}_q$ $\alpha = a g^\beta h^\gamma y^\delta \bmod p$ $\varepsilon = f(m, \alpha)$
	$\xleftarrow{e}$	$e = \varepsilon - \delta \bmod q$
$R = t + er \bmod q$ $S = u + es \bmod q$	$\xrightarrow{R, S}$	
		$a \stackrel{?}{=} g^R h^S y^e \bmod p$ $\rho = R + \beta \bmod q$ $\sigma = S + \gamma \bmod q$
	Then $\alpha = g^\rho h^\sigma y^\varepsilon \bmod p$	

Fig. 11. The Okamoto-Schnorr blind signature scheme

- The authority chooses $(t, u) \in (\mathbb{Z}_t^\star)^2$, computes and sends the commitment $a = g^t h^u \bmod p$.
- The user chooses $\beta, \gamma, \delta \in \mathbb{Z}_q$ and blinds a into $\alpha = a g^\beta h^\gamma y^\delta \bmod p$. He computes the challenge $\varepsilon = f(m, \alpha)$ and sends $e = \varepsilon - \delta \bmod q$ to the authority.
- The authority computes $R = t + er \bmod q$ and $S = u + es \bmod q$, and sends the pair (R, S) which satisfies $a = g^R h^S y^e \bmod p$;
- the user computes $\rho = R + \beta \bmod q$ and $\sigma = S + \gamma \bmod q$.

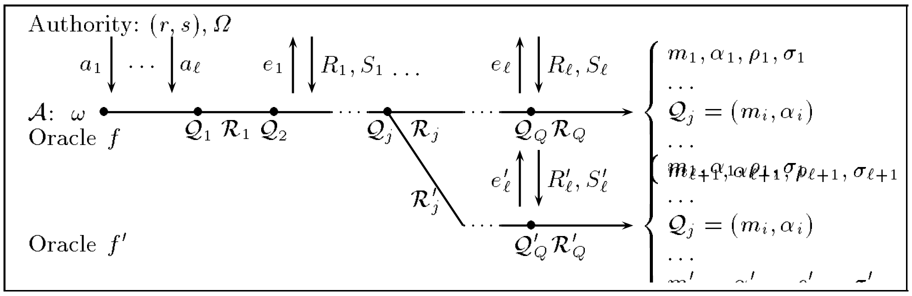


Fig. 12. Forking Lemma.

Straightforward computations show that $\alpha = g^\rho h^\sigma y^\varepsilon \bmod p$, with $\varepsilon = f(m, \alpha)$. Security arguments follow from the theorem below.

Lemma 7. *Consider the Okamoto-Schnorr blind signature scheme in the random oracle model. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and ℓ the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the authority, with $Q, \ell \geq 1$. Assume that, within the time bound T, $\mathcal{A}$ produces, with probability $\varepsilon \geq 4Q^{\ell+1}/q$, an $(\ell, \ell+1)$-forgery. Then, within time $T' \leq 97Q\ell^2T/\varepsilon$, and with probability $\varepsilon' \geq 1/96\ell$, a polynomial replay of this machine provides the discrete logarithm of h relative to g.*

Proof. We first give an outline of the proof. Then we describe the reduction we use. Finally, we evaluate the probability of success and the cost of the reduction.

Outline We start with an $(\ell, \ell+1)$-attacker $\mathcal{A}$, which is a probabilistic polynomial time Turing machine with random tape ω. During the attack, this machine asks a polynomial number Q of queries which we assume to be distinct: $\mathcal{Q}_1$, $\ldots$, $\mathcal{Q}_Q$. Furthermore, $\mathcal{A}$ performs ℓ interactions with the authority, denoted by (a_i, e_i, R_i, S_i) for $i \in \{1, \ldots, \ell\}$. Finally, with probability ε, $\mathcal{A}$ returns $\ell+1$ valid signatures, $(m_i, \alpha_i, \varepsilon_i, \rho_i, \sigma_i)$ for $i = 1, \ldots, \ell+1$. These signatures satisfy the required equations with $\varepsilon_i = f(m_i, \alpha_i)$.

The public data consist of two large primes p and q such that $q \mid (p-1)$ and two elements, g and h, of $\mathbb{Z}_p^\star$ of order q. The authority possesses a secret key (r, s) associated to public key $y = g^{-r}h^{-s} \bmod p$, and a random tape Ω.

Through a collusion (presented in Fig. 12) of the authority and the attacker, we want to compute the discrete logarithm of h relative to g. We use the oracle replay technique that was previously formalized. We hope that, after polynomially many replays of $\mathcal{A}$, we obtain two distinct representations of some α_i relative to g and h. From $\alpha_i = g^a h^b = g^c h^d \bmod p$ with $a \neq c$ we get $\log_g h = (a-c)(d-b)^{-1} \bmod q$.

Cleaning up notations. In the collusion, the pair (r, s) is the secret key used by the authority and the random tape Ω of the authority determines the pairs (t_i, u_i) such that $a_i = g^{t_i}h^{u_i} \bmod p$ for $i = 1, \ldots, \ell$. Note that the distribution of (r, s, y) where r and s are random and $y = g^{-r}h^{-s} \bmod p$ is equal to the distribution of (r, s, y) where r, y are random and s is the unique element in $\mathbb{Z}_q^\star$ such that $y = g^{-r} h^{-s} \bmod p$. Accordingly, we replace (r, s) by (r, y) and, similarly, each (t_i, u_i) by (t_i, a_i).

In the following, we group $(\omega, y, a_1, \ldots, a_\ell)$ under variable ν, and τ represents the ℓ-tuple $(t_1, \ldots, t_\ell)$.

As observed in the previous section, if a query has not been asked during the attack, then the probability for one ε_i to be equal to $f(m_i, \alpha_i)$ is less than $1/q$. Thus, with probability $\rho \geq \varepsilon - (\ell+1)/q \geq 3Q^{\ell+1}/q$, the machine $\mathcal{A}$ performs a forgery with all the outputs (m_i, α_i) asked to the random oracle during the attack, and, accordingly, we define Ind_i to be the index j such that $\mathcal{Q}_j = (m_i, \alpha_i)$.

Finally, we denote by $\mathcal{S}$ the set of all successful data, *i.e.* quadruples (ν, r, τ, f) such that the attack succeeds and every index Ind_i is well-defined. Then, we have $\Pr_{\nu,r,\tau,f}[(\nu, r, \tau, f) \in \mathcal{S}] = \rho \geq 3Q^{\ell+1}/q$.

Reduction. The reduction is as follows:

1. We first run the attack with random ν, r, τ, and f until we obtain a success, or at most $1/\varepsilon$ times.

   In case of success, we denote respectively by $\tilde{Q}$ and $\tilde{\ell}$ the number of queries that $\mathcal{A}$ has asked to the random oracle and the number of interactions that $\mathcal{A}$ has had with the authority, then $\mathcal{A}$ outputs $\tilde{\ell}+1$ valid signatures. Note that $\tilde{Q} \leq Q$ and $\tilde{\ell} \leq \ell$.
2. For $i = 1, \ldots, \tilde{\ell}+1$:

   we let $j = Ind_i(\nu, r, \tau, f)$ and run the attack, with identical ν, r, τ, but a different oracle f' such that the $j-1$ first answers are unchanged, *i.e.* $f'_j = f_j$, until we obtain again a success with $Ind_i(\nu, r, \tau, f') = j$, or at most $48Q\ell/\varepsilon$ times.

We expect that, with nonnegligible probability, both successes output a common α_i coming from the jth oracle query having two distinct representations relative to g and h.

Success of the Reduction After $1/\varepsilon$ repetitions of the attack, with probability greater than one half, we have had at least one success $(\nu, r, \tau, f) \in \mathcal{S}$. Therefore, for all $i \in \{1, \ldots, \ell+1\}$, $\alpha_i = g^{\rho_i} h^{\sigma_i} y^{\varepsilon_i} = g^{\rho_i - r\varepsilon_i} h^{\sigma_i - s\varepsilon_i} \bmod p$. We randomly choose $i \in \{1, \ldots, \ell+1\}$. Then we replay with identical ν, r, τ, but a different oracle f' such that the $j-1$ first answers are unchanged, where $j = Ind_i(\nu, r, \tau, f)$. We will prove that we obtain a new representation of α_i:

$$\alpha_i = g^{\rho'_i - r\varepsilon'_i} h^{\sigma'_i - s\varepsilon'_i} \bmod p \quad \text{with} \quad \rho'_i - r\varepsilon'_i \neq \rho_i - r\varepsilon_i \bmod q.$$

The main question we have to study is whether or not the random variable $\chi_i = \rho_i - r\varepsilon_i$ is sensitive to queries asked at steps j, $j+1$, etc. We expect that the answer is yes. A way to grasp the question is to consider the most likely value taken by this random variable when (ν, r, τ) and the $j-1$ first answers of f are fixed. We are thus led to consider two functions $c_{i,j}(\nu, r, \tau, f)$ and $C_i(\nu, r, \tau, f)$ which we now define. Set

$$\lambda_{i,j}(\nu, r, \tau, f, c) = \Pr_{f'}\left[\begin{array}{c} \Big((\nu, r, \tau, f') \in \mathcal{S}\Big) \ \& \ \Big(Ind_i(\nu, r, \tau, f') = j\Big) \\ \& \ \Big(\chi_i(\nu, r, \tau, f') = c\Big) \end{array} \middle| f'_j = f_j \right],$$

where f_j denotes, as above, the restriction of f to queries of index strictly less than j. Let $c_{i,j}(\nu, r, \tau, f)$ be any value c such that $\lambda_{i,j}(\nu, r, \tau, f, c)$ is maximal. Furthermore, let $C_i(\nu, r, \tau, f) = c_{i, Ind_i(\nu,r,\tau,f)}(\nu, r, \tau, f)$. Accordingly, we define a partition of $\mathcal{S}$: the "good" subset $\mathcal{G}$ whose elements satisfy, for all i, $\chi_i(\nu, r, \tau, f) = C_i(\nu, r, \tau, f)$, and the "bad" $\mathcal{B}$ its complement in $\mathcal{S}$. The aim of the following is to prove that, with non-negligible probability, the success obtained at the first step of the reduction lies in $\mathcal{B}$.

In order to prove this fact, we define the following transformation.

Definition 8. We denote by Φ the transformation which maps any quadruple (ν, r, τ, f) to $(\nu, r+1, \tau - e, f)$, where $\tau - e = (t_1 - e_1, \ldots, t_\ell - e_\ell)$.

This transformation has useful properties (see Fig. 13).

Lemma 8. *Both executions corresponding to (ν, r, τ, f) and $\Phi(\nu, r, \tau, f)$ are totally identical with respect to the view of the attacker. Especially, outputs are the same.*

Proof. Let (ν, r, τ, f) be an input for the collusion. Replay with $r' = r+1$ and $\tau' = \tau - e$, the same ν and the same oracle f. The answers of the oracle are unchanged and the interactions with the authority become

$$R'_i(r', t'_i, e_i) = t'_i + r'e_i = (t_i - e_i) + (r+1)e_i = t_i + re_i = R_i(r, t_i, e_i).$$

Thus, everything remains the same. □

Corollary 1. *Φ is a one-to-one mapping from $\mathcal{S}$ onto $\mathcal{S}$.*

The following lemma shows that Φ sends the set $\mathcal{G}$ into $\mathcal{B}$, except for a negligible part.

Lemma 9. *For fixed (ν, r, τ),*

$$\Pr_f[((\nu, r, \tau, f) \in \mathcal{G}) \ \& \ (\Phi(\nu, r, \tau, f) \in \mathcal{G})] \leq Q^{\ell+1}/q.$$

Proof. In order to prove this statement, we argue by contradiction. Assume that there exists a triplet (ν, r, τ) for which the above probability is strictly greater

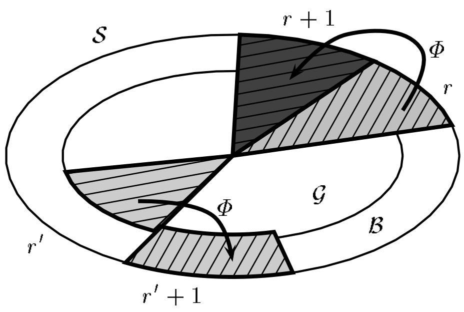


Fig. 13. Properties of Φ.

than $Q^{\ell+1}/q$. Then there exist an $\ell+1$-tuple $(u_1,\ldots,u_{\ell+1}) \in \{1,\ldots,Q\}^{\ell+1}$ and an ℓ-tuple $(\tilde{e}_1,\ldots,\tilde{e}_\ell) \in (\mathbb{Z}_q)^\ell$ such that

$$\Pr_f\left[\begin{array}{c}((\nu,r,\tau,f)\in\mathcal{G})\ \&\ (\Phi(\nu,r,\tau,f)\in\mathcal{G})\\ \&\ ((\forall i)\ Ind_i(\nu,r,\tau,f)=u_i)\ \&\ ((\forall i)\ e_i=\tilde{e}_i)\end{array}\right] > 1/q^{\ell+1}.$$

Thus, there exist an index i and two oracles f and f' which provide distinct answers for the u_ith query, *i.e.*, $f(\mathcal{Q}_{u_i}) \neq f'(\mathcal{Q}_{u_i})$, and are such that answers to queries not of the form $\mathcal{Q}_{Ind_j}$ are similar. We denote by i the smallest such index, and $j = Ind_i(\nu,r,\tau,f) = Ind_i(\nu,r,\tau,f') = u_i$. Then $f_j = f'_j$ and $\varepsilon_i \neq \varepsilon'_i$. Furthermore, we have $(\nu,r,\tau,f)\in\mathcal{G}$, $\Phi(\nu,r,\tau,f)\in\mathcal{G}$. Similarly, $(\nu,r,\tau,f')\in\mathcal{G}$, $\Phi(\nu,r,\tau,f')\in\mathcal{G}$. Because of the property of Φ (see lemma 8), and by definition of the subset $\mathcal{G}$,

$$\begin{aligned}
c_{i,j}(\nu,r,\tau,f) &= \rho_i(\nu,r,\tau,f) - r\varepsilon_i = \rho_i(\Phi(\nu,r,\tau,f)) - r\varepsilon_i\\
&= c_{i,j}(\nu,r+1,\tau-e,f) + ((r+1)-r)\cdot\varepsilon_i\\
c_{i,j}(\nu,r,\tau,f') &= \rho_i(\nu,r,\tau,f') - r\varepsilon'_i = \rho_i(\Phi(\nu,r,\tau,f')) - r\varepsilon'_i\\
&= c_{i,j}(\nu,r+1,\tau-e',f') + ((r+1)-r)\cdot\varepsilon'_i
\end{aligned}$$

The equality $f_j = f'_j$ implies $c_{i,j}(\nu,r,\tau,f) = c_{i,j}(\nu,r,\tau,f')$. Since we have assumed $(e_1,\ldots,e_\ell) = (e'_1,\ldots,e'_\ell) = (\tilde{e}_1,\ldots,\tilde{e}_\ell)$, then

$$c_{i,j}(\nu,r+1,\tau-e,f) = c_{i,j}(\nu,r+1,\tau-e',f').$$

Thus $\varepsilon_i = \varepsilon'_i$, which contradicts the hypothesis. $\square$

We can partition the set $\mathcal{G}$ into two subsets: the subset $\mathcal{G}_g$ whose elements have their image by Φ in $\mathcal{G}$, and its complement $\mathcal{G}_b$ whose elements have their image by Φ in $\mathcal{B}$. From the previous theorem, and since Φ is a bijection from $\mathcal{S}$ into $\mathcal{S}$, $\Pr[\mathcal{G}] = \Pr[\mathcal{G}_g] + \Pr[\mathcal{G}_b] \leq Q^{\ell+1}/q + \Pr[\mathcal{B}]$. Then $\mathcal{B}$ is a nonnegligible set since

$$\Pr[\mathcal{B}] \geq \Pr[\mathcal{S}] - \Pr[\mathcal{B}] - \frac{Q^{\ell+1}}{q} \geq \frac{1}{2}\Big(\rho - \frac{Q^{\ell+1}}{q}\Big) \geq \frac{\rho}{3},$$

where ρ has been defined as $\Pr[\mathcal{S}]$.

With this lower bound on the size of $\mathcal{B}$, we complete the evaluation of the probability of success of the reduction.

First, for any i and j, we define

$$\mathcal{B}_i = \{(\nu,r,\tau,f)\in\mathcal{B}\ \&\ C_i \neq \chi_i\} \quad\text{and}\quad \mathcal{B}_{i,j} = \{(\nu,r,\tau,f)\in\mathcal{B}_i\ \&\ Ind_i = j\}.$$

Then, we can remark that $\sum_{i=1}^{i=\ell+1}\Pr[\mathcal{B}_i] \geq \Pr[\cup_{i=1}^{i=\ell+1}\mathcal{B}_i] = \Pr[\mathcal{B}] \geq \rho/3$. Therefore, there exists $i\in\{1,\ldots,\ell+1\}$ such that $\Pr[\mathcal{B}_i] \geq \rho/3(\ell+1)$. In the following, we assume that i has been chosen so that this inequality holds.

We now define the set $J_i = \{j \mid \Pr[\mathcal{B}_{i,j} \,|\, \mathcal{B}_i] \geq 1/2Q\}$. As in a previous situation (lemma 3), we observe that

$$\Pr\left[\bigcup_{j\in J_i}\mathcal{B}_{i,j} \,\Big|\, \mathcal{B}_i\right] = \Pr[Ind_i \in J_i \,|\, \mathcal{B}_i] \geq \frac{1}{2}.$$

For any $j \in J_i$, $\Pr[\mathcal{B}_{i,j}] \geq \rho/6(\ell+1)Q$, so the Splitting-Lemma (Lemma 1) ensures that there exists a subset $\Omega_{i,j}$ such that

– for any $(\nu, r, \tau, f) \in \Omega_{i,j}$,

$$\Pr_{f'}[(\nu, r, \tau, f') \in \mathcal{B}_{i,j} \mid f'_j = f_j] \geq \rho/12(\ell+1)Q,$$

– $\Pr[\Omega_{i,j} \mid \mathcal{B}_{i,j}] \geq \frac{1}{2}$.

Since all the subsets $\mathcal{B}_{i,j}$ are disjoint, for any fixed i,

$$\Pr_{\nu,r,\tau,f}[\exists j \in J_i : (\nu, r, \tau, f) \in \Omega_{i,j} \cap \mathcal{B}_{i,j} \mid \mathcal{S}] = \sum_{j \in J_i} \Pr[\Omega_{i,j} \cap \mathcal{B}_{i,j} \mid \mathcal{S}]$$

$$= \sum_{j \in J_i} \Pr[\Omega_{i,j} \mid \mathcal{B}_{i,j}] \cdot \Pr[\mathcal{B}_{i,j} \mid \mathcal{B}_i] \cdot \Pr[\mathcal{B}_i \mid \mathcal{S}]$$
$$\geq \left(\sum_{j \in J_i} \Pr[\mathcal{B}_{i,j} \mid \mathcal{B}_i] \right) /6(\ell+1) \geq 1/12(\ell+1).$$

Globally, the first step provides a tuple (ν, r, τ, f) in $\mathcal{S}$ such that for some i, $(\nu, r, \tau, f) \in \Omega_{i,j} \cap \mathcal{B}_{i,j}$, where we note $j = Ind_i$, with probability greater than $1/12(\ell+1) \geq 1/24\ell$. Assume that we know this index i. We denote by d the value $\chi_i(\nu, r, \tau, f)$ and by c the value $C_i(\nu, r, \tau, f)$. Then two cases appear relatively to the value $\lambda_{i,j}(\nu, r, \tau, f, d)$:

1. If $\lambda_{i,j}(\nu, r, \tau, f, d) \geq \rho/24Q(\ell+1)$, then, by definition of C_i, we know that

$$\Pr_{f'}\left[(\nu, r, \tau, f') \in \mathcal{S} \,\&\, \chi_i(\nu, r, \tau, f') \neq d \,\&\, Ind_i(\nu, r, \tau, f) = j \middle| f'_j = f_j\right]$$
$$\geq \lambda_{i,j}(\nu, r, \tau, f, c) \geq \rho/24Q(\ell+1).$$

2. otherwise,

$$\Pr_{f'}\left[(\nu, r, \tau, f') \in \mathcal{S} \,\&\, \chi_i(\nu, r, \tau, f') \neq d \,\&\, Ind_i(\nu, r, \tau, f) = j \middle| f'_j = f_j\right]$$
$$= \Pr_{f'}\left[(\nu, r, \tau, f') \in \mathcal{S} \,\&\, Ind_i(\nu, r, \tau, f) = j \middle| f'_j = f_j\right] - \lambda_{i,j}(\nu, r, \tau, f, d)$$
$$\geq \Pr_{f'}[(\nu, r, \tau, f') \in \mathcal{B}_{i,j} \mid f'_j = f_j] - \lambda_{i,j}(\nu, r, \tau, f, d)$$
$$\geq \rho/24Q(\ell+1).$$

Both cases lead to

$$\Pr_{f'}\left[\begin{array}{c} (\nu, r, \tau, f') \in \mathcal{S} \\ \&\ \chi_i(\nu, r, \tau, f') \neq d \\ \&\ Ind_i(\nu, r, \tau, f) = j \end{array} \middle| f'_j = f_j \right] \geq \frac{\rho}{24Q(\ell+1)}.$$

Thus, after at most $48Q\ell/\varepsilon$ replays with the same keys and random tapes but another random oracle f' such that $f'_j = f_j$, we obtain, with probability at least $\frac{1}{2}$, a new success with $Ind_i(\nu, r, \tau, f') = j$ and $\chi_i(\nu, r, \tau, f') \neq d$. Then both executions provide two different representations of α_i relative to g and h.

Cost of the reduction. After at most $(1 + 48Q\ell \cdot (\ell + 1))/\varepsilon \leq 97Q\ell^2/\varepsilon$ iterations of $\mathcal{A}$, the probability of success is greater than $\frac{1}{2} \times 1/24\ell \times \frac{1}{2}$ and so is upper bounded by $1/96\ell$, where ε is the probability of success of an $(\ell,\ \ell + 1)$-forgery and Q is the number of queries asked to the random oracle. □

As for the security of signatures (Theorem 1), we can present an expected polynomial time Turing machine $\mathcal{M}$:

1. $\mathcal{M}$ initializes $r = 0$;
2. $\mathcal{M}$ runs $\mathcal{A}$ until it outputs a successful tuple $(\nu, r, \tau, f) \in \mathcal{S}$ and denotes by N_r the number of calls to $\mathcal{A}$ to obtain this success, and by ℓ_r the number of interactions with the signer during this success;
3. for $i = 1, \ldots, \ell_r + 1$, $\mathcal{M}$ replays, at most $120 N_r \alpha^r$ times, the machine $\mathcal{A}$ with fixed (ν, r, τ) and random f' such that $f'_j = f_j$, where $j = \mathit{Ind}_i(\nu, r, \tau, f)$ and $\alpha = 1 + 1/54k$;
4. $\mathcal{M}$ increments r and returns to 2, until it gets a successful forking.

For any execution of $\mathcal{M}$, we denote by J the last value of r and by N the total number of calls to $\mathcal{A}$. We want to compute the expectation of N. Since $\rho = \Pr[\mathcal{S}]$, and $N_r \geq 1$, then $\Pr[N_j \geq 1/5\rho] \geq \frac{3}{4}$. We define $L = \lceil \log_\alpha Q(\ell + 1) \rceil$, so that, $120 N_r \alpha^r \geq 24Q(\ell + 1)/\rho$ for any $r \geq L$, whenever $N_r \geq 1/5\rho$. Therefore, for any $r \geq L$, when we get a success (ν, r, τ, f) at the first step, with probability greater than $1/12(\ell{+}1)$, there exists $i \in \{1, \ldots, \ell_r + 1\}$ such that $\beta = \mathit{Ind}_i(\nu, r, \tau, f) \in J_i$ and $(\nu, r, \tau, f) \in \Omega_{i,\beta} \cap \mathcal{S}_{i,\beta}$. Furthermore, with probability greater than $\frac{3}{4}$, $N_r \geq 1/5\rho$. Therefore, with the same conditions as before, that is $\varepsilon \geq 4Q^{\ell+1}/q$, the probability of getting a successful fork after at most $24Q(\ell + 1)/\rho$ iterations at step 3 is greater than $\frac{3}{5}$. Then, for any t greater than L, the probability for J to be greater or equal to t is less than $(1 - 1/12(\ell + 1) \times \frac{3}{4} \times \frac{3}{5})^{t-L}$. Therefore, this probability is less than γ^{t-L}, with $\gamma = 1 - 1/27(\ell + 1)$. Furthermore,

$$\begin{aligned}
E[N \mid J = t] &\leq \sum_{r=0}^{r=t} \Big(E[N_r] + 120 E[N_r] \cdot (\ell_r + 1)\alpha^r \Big) \\
&\leq \frac{121(\ell + 1)}{\rho} \times \sum_{r=0}^{r=t} \alpha^j \leq \frac{121(\ell + 1)}{\rho} \times \frac{\alpha^{t+1}}{\alpha - 1}.
\end{aligned}$$

Since $\varepsilon \geq 4Q^{\ell+1}/q$, we get $\ell + 1 \leq k$, and therefore $\alpha\gamma \leq 1 - 1/54k$. Finally, the expectation of N is

$$E[N] \leq \frac{162(\ell + 1)}{\varepsilon} \cdot \frac{\alpha^{L+1}}{\alpha - 1} \cdot \left(\frac{1}{\alpha - 1} + \frac{1}{1 - \alpha\gamma} \right) \leq \frac{162(\ell + 1)}{\varepsilon} \cdot 57kQ(\ell{+}1) \cdot 108k.$$

Hence the following theorem:

Theorem 8. *(Forking Lemma). Consider the Okamoto-Schnorr blind signature scheme in the random oracle model. Let $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and ℓ the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the authority. Assume that, with a* *time bound T, $\mathcal{A}$ performs, with probability $\varepsilon \geq 4Q^{\ell+1}/q$, an $(\ell, \ell+1)$-forgery. Then there is another machine which has control over $\mathcal{A}$ and solves the discrete logarithm of h relative to g in expected time $T' \leq 10^6(\ell+1)^2 k^2 QT/\varepsilon$,*

$$
\begin{array}{lcr}
\boxed{\text{Authority}} & & \boxed{\text{User}} \\
\multicolumn{3}{c}{N = pq \text{ and } \lambda \text{ prime and prime with } \varphi(N)} \\
\multicolumn{3}{c}{a \in \mathbb{Z}_N^\star \text{ of order greater than } \lambda} \\
\text{secret} \quad r \in \{0, \ldots, \lambda-1\} & & \\
s \in \mathbb{Z}_N^\star & & \\
\text{public} \quad v = a^{-r}s^{-\lambda}[N] & & \\
t \in \{0, \ldots, \lambda-1\} & & \\
u \in \mathbb{Z}_N^\star & & \\
x = a^t u^\lambda [N] & \xrightarrow{\quad x \quad} & \alpha, \gamma \in \{0, \ldots, \lambda-1\} \\
 & & \beta \in \mathbb{Z}_N^\star \\
 & & x' = x a^\alpha \beta^\lambda v^\gamma [N] \\
 & & c' = f(m, x') \\
 & & \in \{0, \ldots, \lambda-1\} \\
y = t + cr \bmod \lambda & \xleftarrow{\quad c \quad} & c = c' - \gamma \bmod \lambda \\
w = t + cr \div \lambda & & \\
z = a^w u s^c [N] & \xrightarrow{\quad y, z \quad} & y' = y + \alpha \bmod \lambda \\
 & & w' = y + \alpha \div \lambda \\
 & & w'' = c' - c \div \lambda \\
 & & z' = a^{w'} v^{-w''} z \beta [N] \\
\multicolumn{3}{c}{\text{Then } x' = a^{y'} {z'}^\lambda v^{c'} [N]}
\end{array}
$$

Fig. 14. The Okamoto–Guillou-Quisquater blind signature scheme

These proofs can be easily modified to cover other schemes that come from witness indistinguishable protocols. Especially, the Okamoto version of the Guillou-Quisquater identification scheme provides a provably secure blind signature scheme (see Fig. 14) relative to the security of RSA. Furthermore, the authors have presented [44] blind signature schemes derived from the Fiat-Shamir identification scheme [21] and from the Ong-Schnorr identification scheme [38], which are clearly witness indistinguishable. The resulting schemes admit security arguments relative to factorization.

4.3. *Remarks*

Our result appears to be the first security result which paves the way toward provably secure electronic cash systems by providing candidates for secure blind signatures. However, it leaves an open problem: the complexity of our reduction is polynomial in the size of the key but not in ℓ. Our theorem only provides security arguments against strong "one-more" forgeries. In fact, the reduction requires $\varepsilon \geq 4Q^{\ell+1}/q$, which implies a polylogarithmically bounded number of interactions with the authority. We were unable to achieve polynomial time both in ℓ and the size of the keys.

Juels et al. [32] gave a positive answer to the question using the provably secure signature scheme of Naor and Yung [33] and the Two-Party Completeness Theorem [25]. Nevertheless, their construction is theoretical and the problem of having a practical scheme is still open.

Conclusion

As explained in the Introduction, there were several proposals for provably secure signature schemes. However, in all cases, the security was at the cost of a considerable loss in terms of efficiency. Concerning blind signatures, Damgård [15], Pfitzmann and Waidner [39] and more recently at Crypto '97, Juels et al. [32] have presented some blind signature schemes with a complexity-based proof of security. Again, the security is at the cost of inefficiency.

In the weaker setting offered by the random oracle model, we have provided security arguments for practical and even efficient digital signature schemes and blind signature schemes. On the ground of our reductions, one can justify realistic parameters, even if they are not optimal. Further improvements are expected particularly in the case of blind signatures where it should be possible to obtain a reduction polynomial in the size of the keys and in the number of interactions with the signer.

In any case, the arguments in this paper, based on the random oracle model, are a quite strong indication that the overall design of the corresponding schemes is presumably correct.

References

1. M. Bellare and S. Micali. How to Sign Given any Trapdoor Function. *Journal of the ACM*, 39(1):214–233, January 1992.
2. M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCCS*, pages 62–73. ACM Press, New York, 1993.
3. M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.
4. D. Bleichenbacher. Generating El Gamal Signatures without Knowing the Secret Key. In *Eurocrypt '96*, LNCS 1070, pages 10–18. Springer-Verlag, Berlin, 1996.
5. A. Bosselaers, H. Dobbertin, and B. Preneel. RIPEMD-160: a Strengthened Version of RIPEMD. In *Proc of the 3rd FSE*, LNCS 1039, pages 71–82. Springer-Verlag, Berlin, 1996.
6. S. A. Brands. An Efficient Off-Line Electronic Cash System Based on the Representation Problem. Technical Report CS-R9323, CWI, Amsterdam, 1993.
7. S. A. Brands. Untraceable Off-Line Cash in Wallets with Observers. In *Crypto '93*, LNCS 773, pages 302–318. Springer-Verlag, Berlin, 1994.
8. S. A. Brands. A Note on Parallel Executions of Restrictive Blind Issuing Protocols for Secret-Key Certificates. Technical Report CS-R9519, CWI, Amsterdam, 1995.
9. S. A. Brands. More on Restrictive Blind Issuing of Secret-Key Certificates in Parallel Mode. Technical Report CS-R9534, CWI, Amsterdam, 1995.
10. S. A. Brands. Off-Line Electronic Cash Based on Secret-Key Certificates. In *LATIN '95*, LNCS 911, pages 131–166. Springer-Verlag, Berlin, 1995.
11. S. A. Brands. Restrictive Blind Issuing of Secret-Key Certificates in Parallel Mode. Technical Report CS-R9523, CWI, Amsterdam, 1995.
12. J. Camenisch, U. Maurer, and M. Stadler. Digital Payment Systems with Passive Anonymity-Revoking Trustees. *Journal of Computer Security*, 5(1):254–265, 1997.
13. D. Chaum. Blind Signatures for Untraceable Payments. In *Crypto '82*, pages 199–203. Plenum, New York, 1983.

14. D. Chaum, A. Fiat, and M. Naor. Untraceable Electronic Cash. In *Crypto '88*, LNCS 403, pages 319–327. Springer-Verlag, Berlin, 1989.
15. I. B. Damgård. A Design Principle for Hash Functions. In *Crypto '89*, LNCS 435, pages 416–427. Springer-Verlag, Berlin, 1990.
16. W. Diffie and M. E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT–22(6):644–654, November 1976.
17. T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.
18. U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *Proc. of the 22nd STOC*, pages 416–426. ACM Press, New York, 1990.
19. N. Ferguson. Extensions of Single Term Coins. In *Crypto '93*, LNCS 773, pages 292–301. Springer-Verlag, Berlin, 1994.
20. N. Ferguson. Single Term Off-Line Coins. In *Eurocrypt '93, Berlin*, LNCS 765, pages 318–328. Springer-Verlag, 1994.
21. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions of Identification and Signature Problems. In *Crypto '86*, LNCS 263, pages 186–194. Springer-Verlag, Berlin, 1987.
22. Y. Frankel, Y. Tsiounis, and M. Yung. "Indirect Disclosure Proof": Achieving Efficient Fair Off-Line E-Cash. In *Asiacrypt '96*, LNCS 1163, pages 286–300. Springer-Verlag, Berlin, 1996.
23. M. Franklin and M. Yung. Secure and Efficient Off-Line Digital Money. In *Proc. of the 20th ICALP*, LNCS 700, pages 265–276. Springer-Verlag, Berlin, 1993.
24. M. R. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, 1979.
25. O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game. In *Proc. of the 19th STOC*, pages 218–229. ACM Press, New York, 1987.
26. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proc. of the 17th STOC*, pages 291–304. ACM Press, New York, 1985.
27. S. Goldwasser, S. Micali, and R. Rivest. A "Paradoxical" Solution to the Signature Problem. In *Proc. of the 25th FOCS*, pages 441–448. IEEE, New York, 1984.
28. S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptative Chosen-Message Attacks. *SIAM Journal of Computing*, 17(2):281–308, April 1988.
29. S. Goldwasser, S. Micali, and A. Yao. Strong Signature Schemes. In *Proc. of the 15th STOC*, pages 431–439. ACM Press, New York, 1983.
30. L. C. Guillou and J.-J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory. In *Eurocrypt '88*, LNCS 330, pages 123–128. Springer-Verlag, Berlin, 1988.
31. M. Jakobsson and M. Yung. Revokable and Versatile Electronic Money. In *Proc. of the 3rd CCCS*, pages 76–87. ACM Press, New York, 1996.
32. A. Juels, M. Luby, and R. Ostrovsky. Security of Blind Digital Signatures. In *Crypto '97*, LNCS 1294, pages 150–164. Springer-Verlag, Berlin, 1997.
33. M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. In *Proc. of the 21st STOC*, pages 33–43. ACM Press, New York, 1989.
34. National Bureau of Standard U.S. Data Encryption Standard, 1977.
35. NIST. *Digital Signature Standard* (DSS). Federal Information Processing Standards PUBlication 186, November 1994.
36. NIST. *Secure Hash Standard* (SHS). Federal Information Processing Standards PUBlication 180–1, April 1995.
37. T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *Crypto '92*, LNCS 740, pages 31–53. Springer-Verlag, Berlin, 1992.
38. H. Ong and C.P. Schnorr. Fast Signature Generation with a Fiat-Shamir-Like Scheme. In *Eurocrypt '90*, LNCS 473, pages 432–440. Springer-Verlag, Berlin, 1991.
39. B. Pfitzmann and M. Waidner. How to Break and Repair a "Provably Secure" Untraceable Payment System. In *Crypto '91*, LNCS 576, pages 338–350. Springer-Verlag, Berlin, 1992.
40. J. Pieprzyk, J. Seberry, and Y. Zheng. A One-Way Hashing Algorithm with Variable Length and Output. In *Auscrypt '92*, LNCS 718, pages 83–104. Springer-Verlag, Berlin, 1993.
41. D. Pointcheval. A New Identification Scheme Based on the Perceptrons Problem. In *Eurocrypt '95*, LNCS 921, pages 319–328. Springer-Verlag, Berlin, 1995.
42. D. Pointcheval and J. Stern. Provably Secure Blind Signature Schemes. In *Asiacrypt '96*, LNCS 1163, pages 252–265. Springer-Verlag, Berlin, 1996.
43. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt '96*, LNCS 1070, pages 387–398. Springer-Verlag, Berlin, 1996.

44. D. Pointcheval and J. Stern. New Blind Signatures Equivalent to Factorization. In *Proc. of the 4th CCCS*, pages 92–99. ACM Press, New York, 1997.
45. C. Radu, R. Govaerts, and J. Vanderwalle. A Restrictive Blind Signature Scheme with Applications to Electronic Cash. In *Communications and Multimedia Security II*, pages 196–207. Chapman & Hall, London, 1996.
46. *Ripe Integrity Primitives – Final Report of RACE Integrity Primitives Evaluation (R1040)*, LNCS 1007. Springer-Verlag, Berlin, 1995.
47. R. Rivest. The MD4 Message-Digest Algorithm. RFC 1320, The Internet Engineering Task Force, April 1992.
48. R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, The Internet Engineering Task Force, April 1992.
49. 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.
50. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Crypto '89*, LNCS 435, pages 235–251. Springer-Verlag, Berlin, 1990.
51. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
52. L. A. M. Schoenmakers. An Efficient Electronic Payment System withstanding Parallel Attacks. Technical Report CS-R9522, CWI, Amsterdam, 1995.
53. A. Shamir. An Efficient Identification Scheme Based on Permuted Kernels. In *Crypto '89*, LNCS 435, pages 606–609. Springer-Verlag, Berlin, 1990.
54. J. Stern. A New Identification Scheme Based on Syndrome Decoding. In *Crypto '93*, LNCS 773, pages 13–21. Springer-Verlag, Berlin, 1994.
55. J. Stern. Designing Identification Schemes with Keys of Short Size. In *Crypto '94*, LNCS 839, pages 164–173. Springer-Verlag, Berlin, 1994.
56. J. Stern. The Validation of Cryptographic Algorithms. In *Asiacrypt '96*, LNCS 1163, pages 301–310. Springer-Verlag, Berlin, 1996.
57. S. von Solms and D. Naccache. On Blind Signatures and Perfect Crimes. *Computers & Security*, 11:581–583, 1992.