On the Fly Authentication and Signature Schemes based on Groups of Unknown Order

Marc Girault[1], Guillaume Poupard[2], and Jacques Stern[3]

[1] France Telecom Research & Development, 42 rue des Coutures
BP 6243, F-14066 Caen Cedex 4, France
marc.girault@francetelecom.com

[2] DCSSI Crypto Lab, 51 boulevard de La Tour-Maubourg
F-75700 Paris 07 SP, France
Guillaume.Poupard@m4x.org

[3] École normale supérieure, Département d'informatique
45 rue d'Ulm, F-75230 Paris Cedex 05, France
Jacques.Stern@ens.fr

Abstract. In response to the current need for fast, secure and cheap public-key cryptography, we propose an interactive zero-knowledge identification scheme and a derived signature scheme that combine provable security based on the problem of computing discrete logarithms in any group, short keys, very short transmission and minimal on-line computation. This leads to both efficient and secure applications well suited to implementation on low cost smart cards.

We introduce GPS, a Schnorr-like scheme that does not require knowledge of the order of the group nor of the group element. As a consequence, it can be used with most cryptographic group structures, including those of unknown order. Furthermore, the computation of the prover's response is done over the integers, hence can be done with very limited computational capabilities. This paper provides complete security proofs of the identification scheme. From a practical point of view, the possible range of parameters is discussed and a report on the performances of an actual implementation on a cheap smart card is included: a complete and secure authentication can be performed in less than 20 milliseconds with low cost equipment.

Key words. Identification scheme, digital signature, discrete logarithm problem, minimal on-line computation, low cost smart cards.

1 Introduction

The rapid world-wide development of electronic transactions stimulates a strong demand for fast, secure and cheap public-key cryptography. Besides confidentiality, cryptographers need to solve two important problems: authentication and providing digital signatures or, in plain words, how to prove one's identity and how to digitally sign a document. Several proposals have addressed those questions, putting forward elegant solutions, many of them based on the concept of zero-knowledge introduced in 1985 by Goldwasser, Micali and Rackoff [30].

In order to assess the performances of proposed schemes, three main properties have to be considered. The most important concern is, of course, security. Obviously, a system can be supported by the claim that nobody has been able to jeopardize it so far. This is of course important but, in many applications, it is not a satisfactory enough guarantee. A much better paradigm tries to prove security in a mathematical sense, i.e. to establish theorems claiming that illegal actions such as impersonation are as difficult as solving a specific problem, whose difficulty is well-established. Among these problems are integer factorization, or the computation of discrete logarithms in a finite group. Half way between heuristic validation and formal proofs are proofs in a model where concrete objects are replaced by some ideal substitutes: applying this paradigm to hash functions yields the so-called random oracle model described by Bellare and Rogaway in [3]. Although this approach may not be considered as offering absolute proofs of security for cryptographic schemes, it provides a strong guarantee that their general design is not flawed.

Next, the size of the data involved in the scheme is of crucial practical significance. We usually need short public and private keys, mainly when they have to be stored in portable devices like chip cards, which may have small storage capabilities. We also want to reduce the amount of transmissions and the length of the signatures. The latter is an important parameter in applications for which many signatures have to be stored (e.g. electronic commerce) or transmitted (e.g. pay TV).

Another key property is the time complexity, since it directly controls the cost of the devices on which a scheme may be implemented. Here, we have to distinguish between precomputations that can be performed off-line and stored in memory, and calculations that have to be done on-line during authentication or signature computation. The latter is often the bottleneck of many applications, especially when smart cards are used. Naccache *et al.* [34] proposed to precompute *use & throw coupons* in order to make the DSA signature process much more efficient. However this attempt for designing *on the fly* signature schemes is not optimal since it still requires a modular multiplication. Another approach is much more general in character: Even, Goldreich and Micali [15] proposed the concept of *on-line/off-line* digital signature and described a construction to transform any signature scheme in such a way that most of the computations can be done off-line. This was further improved by Shamir and Tauman [44].

In this paper, we study an interactive zero-knowledge identification scheme, called GPS for short, and a derived signature scheme. They combine provable security based on the discrete logarithm problem over an arbitrary finite group, short keys, short transmissions and signature size and minimal on-line computation.

The coupon-based signature algorithm GPS allows to implement public-key signature or identification schemes on low cost smart cards, without crypto-processor. Another promising application is the implementation of such schemes on contactless smart cards. Such cards just look like credit cards but they have an electronic microchip and an embedded antenna. These components allow the card to communicate with an antenna/coupler unit without any physical contact. Contactless cards are the ideal solution when transactions must be processed very quickly, as in mass-transit or toll collection but, since the power supply comes from electromagnetic induction, heavy-consumption crypto-processors hardly can be used.

A typical application of GPS is "on the fly" authentication at a toll. The basic idea is to equip each authorized car with a low cost contactless smart card. When a car goes through a toll, it does not have to stop but just performs a GPS authentication in order to prove that it is a legitimate user. In such an application, the time allowed to transmit data and to perform on-line calculations is very short, about 100 milliseconds.

The main feature of GPS is to use public key cryptography in this setting (very short authentication time and low cost devices), thus achieving a high level of security. Furthermore, in such an independent application that does not require interoperability with other systems, the coupons can be computed and stored in the card by the authority who also acts as a verifier.

Notice that symmetric cryptography could also be used to solve this problem efficiently. But the advantage of using public key cryptography is that no secret master key has to be stored by the verifier (here the toll). Consequently, the system is much more secure against piracy.

Earlier announcements. The GPS scheme was first proposed by Girault at Eurocrypt '91 [22] as an example of a scheme with *self-certified* public keys but without security analysis. Then, the main results of this paper appeared in a preliminary version at Eurocrypt '98 [40]. The main technical differences are a more precise security model and a complete proof of security. Many technicalities have been streamlined and we only assume the intractability of computing discrete logarithms with short exponents.

The GPS scheme has been submitted to the European NESSIE project and labelled by this project as a strong cryptographic primitive [12]. Various modes of use are described in [26]. Finally, the GPS identification scheme has been standardized by ISO/IEC (International Organization for Standardization/ International Electrotechnical Commission) [32], while the signature scheme is currently at an earlier stage.

Paper organization. In this paper, we show that GPS achieves a combination of the strongest properties that one can demand in authentication applications. In section 2 we first recall the Schnorr scheme and several of its variants. Then we describe the GPS identification scheme and we recall how it can be turned into a signature scheme. In section 3, we develop our security model for identification, and we study the security of the GPS identification scheme. We prove that GPS is secure against active adversaries provided the so-called discrete logarithm with short exponents problem is hard. In section 4, we establish the security of the derived signature scheme using the random oracle model in order to validate the proposed design. The last section is more practical in character: we discuss how to choose secure parameters in order to resist the known attacks against factorization and the discrete logarithm problem. Then we explain how to optimize size of the data and, finally, report on the performances of a smart card application.

2 Description of GPS

2.1 Identification schemes based on the discrete logarithm problem

In 1989, C. Schnorr [42] proposed a nice proof of knowledge of a discrete logarithm in groups of known prime order. This proof is a more efficient version of previous proposals of Chaum *et al.* [11, 10] and Beth [4]. Such a proof can be used as an identification scheme, and also converted into a signature scheme using the Fiat-Shamir paradigm [18]. In these schemes, the size of the data is short, and the computation load is quite acceptable.

Towards a more precise description, we let p be a prime number. We denote by $\mathbb{Z}_p^*$ the set of invertible elements modulo p. Let q be a large prime divisor of $p-1$ and g an element of $\mathbb{Z}_p^*$ of order q, i.e. such that $g^q = 1 \bmod p$ but $g \neq 1$. The prover knows a secret element s in $\mathbb{Z}_q$ and he wants to prove that he knows the discrete logarithm of $I = g^{-s} \bmod p$ in base g. We first notice that any verifier can immediately check that p is prime, q is a prime divisor of $p-1$, g is of order q and that I belongs to the subgroup of $\mathbb{Z}_p^*$ generated by g. This last verification just consists in checking that $I^q = 1 \bmod p$.

In order to prove knowledge of s, the prover first generates $r \in \mathbb{Z}_q^*$ at random and sends the commitment $x = g^r \bmod p$ to the verifier who answers a challenge c randomly chosen in the interval $[0, B-1]$, where B is a publicly known system parameter. Next, the prover computes $y = r + sc \bmod q$ and sends y to the verifier who checks the equation $x = g^y I^c \bmod p$. This elementary round can be repeated sequentially; we denote by ℓ the number of repetitions.

The security analysis of the scheme shows that a prover accepted with probability substantially greater than $1/B^\ell$ must know the discrete logarithm of I, i.e. the secret s; the proof is sound. Furthermore, even a dishonest verifier cannot learn any additional information about the secret, whatever the number of authentications may be, if B and ℓ are polynomial in a security parameter, i.e. asymptotically "not too large"; the proof is perfectly zero-knowledge.

Many modifications of the Schnorr scheme, that achieve additional properties, have been proposed. Firstly, one can use a composite modulus instead of a prime modulus and keep the factorization of the modulus secret. As a consequence, the order of the multiplicative group in which the computations are performed may remain secret.

Furthermore, the order of the publicly known base g can also be public or private. In the Schnorr scheme, both the order $p-1$ of the group and the order q of g are known. We will see that in the GPS scheme, both of those parameters can remain unknown to provers and verifiers. Other schemes, classified in figure 1, achieve different combinations. We now briefly review those protocols.

<table>
<tr><td rowspan="2"></td><td colspan="2">Order of the multiplicative group</td></tr>
<tr><td>KNOWN</td><td>UNKNOWN</td></tr>
<tr><td>Order of g
KNOWN</td><td>Chaum, Evertse, van de Graaf and Peralta [11, 10],
Beth [4], Schnorr [42],
Okamoto [36]</td><td>Girault [21],
Biham and Shulman [6]</td></tr>
<tr><td>Order of g
UNKNOWN</td><td>Brickell and McCurley [7]</td><td>GPS [22, 40], RDSA [5]
Poupard and Stern [41]</td></tr>
</table>

Fig. 1. Discrete logarithm related schemes classified according to the need for the order of the group and/or of the base g to be known by provers and verifiers

The Okamoto scheme. The Schnorr scheme is known to be perfectly zero-knowledge if the parameters B and ℓ remain polynomial in a security parameter. The one-round ($\ell = 1$) variant remains sound if B is super-polynomial, but, as a consequence, this variant is perfectly zero-knowledge only w.r.t a honest verifier, i.e. a verifier who randomly chooses the challenges. However it is unknown how to prove the zero-knowledge property if the verifier can bias the distribution of his challenges. This means that, for large challenges, we can only prove the security of Schnorr identification against passive adversaries who just observe regular authentications. Exactly the same remark applies to the GPS scheme.

A solution proposed by Okamoto [36] consists in using two bases g_1 and g_2 and to prove the knowledge of a "representation" (s_1, s_2) such that $I = g_1^{s_1} g_2^{s_2} \bmod p$. While this protocol is not proven to be zero-knowledge, it nonetheless is witness indistinguishable [17]. As a consequence, provided the computation of the discrete logarithm of g_1 in base g_2 modulo p is intractable, the scheme is provably secure against active adversaries, even for large challenges (note that so is GPS, as explained in section 3).

The Brickell-McCurley scheme. In the protocol proposed in [7], the computations are still done modulo a prime number p but instead of using a base g of publicly known prime order q, the parameters are chosen such that $p - 1$ is divisible by the product $q \times w$ of two secret prime numbers. The rest of the scheme is similar to the Schnorr protocol; it uses a base g of order q but the answer y is equal to $r + sc \bmod p - 1$ in order not to reveal information about q. The main advantage of this variant is to base the security on the intractability of the discrete logarithm problem modulo p or on the factorization of $p-1$. This means that it is secure if at least one of those two problems is difficult.

The Girault scheme. The idea of [21] is to choose a composite modulus $n = (2fp+1) \times (2fq+1)$ where $2fp+1$, $2fq+1$, p, q and f are prime. The integer f is public, the base g has order f and the answer y is computed modulo f. A public key of a prover of identity $\textsc{Id}$ is obtained with the formula $I = \textsc{Id}^{1/e} g^s \bmod n$ where the e-th root of $\textsc{Id}$ is computed by an authority who knows the factorization of n and s is a secret key chosen by the prover. In this setting, an identification, in spirit, is a Schnorr proof of knowledge of the discrete logarithm $(e \times s)$ of $I^e/\textsc{Id} \bmod n$ in base g.

The GPS scheme. A way of improving the Schnorr protocol efficiency is to get rid of modular reductions during identification or signature. Exponentiation modulo p can be performed off-line by the user's device or precomputed by an authority in a *use & throw coupons* [34] setting. Therefore, in order to further reduce the on-line computation to a very simple operation, it is natural to eliminate the second modulus q by performing the operations $y = r + sc$ in $\mathbb{Z}$. This has first been proposed by Girault in [22] and the security analysis of this protocol is precisely the subject of the present paper in a more general setting. Note that in [24], the on-line operation is reduced to a single, but much longer, addition.

Other schemes. A variant of GPS, called RDSA, has been proposed in [5] and analyzed in [19]. We can also note that the scheme described in [41] is based on the intractability of the factorization problem, but it can be seen as a proof of knowledge of discrete logarithm where the order of the group and the order of the base are secrets owned by the prover. More recently, another factorization-based scheme has been proposed, in which the key pair is a RSA key pair [25].

2.2 GPS identification scheme

We now describe precisely the GPS identification scheme. The security analysis appears in the next section.

Choice of the underlying mathematical structure. The GPS identification scheme is defined on a generic group $\mathcal{G}$ and uses a specific element, namely the base $g \in \mathcal{G}$. In the theoretical security analysis of the next section, we only assume the intractability of computing discrete logarithms in the group $\mathcal{G}$, in base g, for exponents in the range $[0, S-1]$ where S is a public parameter of the scheme.

More precisely, we assume the existence of a randomized algorithm $\mathcal{PP}(\omega_{pp}, k)$ that generates public parameters $\mathcal{G}$ and g according to a security parameter k using a random tape ω_{pp}.

In practice, several mathematical structures can be used; the most interesting choices for $\mathcal{G}$ are listed below:

– $\mathcal{G} = \mathbb{Z}_p^*$ with p a prime number s.t. $p - 1$ has a large prime factor q; the order of the base g should be q. We obtain a variant of the Schnorr identification scheme in which on-line computation is twice as fast for the same security.
– the set $\mathcal{G} = \mathbb{Z}_n^*$ of invertible elements modulo an RSA modulus n, i.e. a composite integer with typically two prime factors of almost the same size. Note that the factors of n are no longer required so they can be discarded after the generation of n. Then g can be randomly chosen. However, the generation of n and g must be done by a trusted party since the computation of "short" exponents, typically of 160 bits, can be done very easily using partial Pohlig-Hellman techniques [45] if the order of g is known and if it has many small prime factors. In practice, we advise the use of modulus n which is the product of two strong primes, i.e. primes p s.t. $(p-1)/2$ is also prime. We also advise the use of the base $g = 2$ for efficiency reasons.
– $\mathcal{G}$ can also be derived from an elliptic curve. Analogs of GPS in the elliptic curve setting can be defined in a straightforward manner; see for example [13].
– Much more sophisticated mathematical structures can also be used; the only constraint is the intractability of the discrete logarithm with short exponent problem is such groups. An example of such an approach is proposed in [5].

Other public parameters. Besides the upper bound S for the secret keys, additional parameters of the GPS scheme are the number ℓ of elementary rounds and two integer bounds A and B, defined below. The relations between those parameters are analyzed in section 3. We just summarize some facts about these parameters in order to make their meaning more explicit:
– the probability of impersonation is $1/B^\ell$,
– the computation of discrete logarithms in base g in the group $\mathcal{G}$ must be intractable for exponents in the interval $[0, S-1]$,
– A must be significantly larger than $S \times B$ since it defines the size of some random data used to mask the secret.

Public/private keys. The private keys s are integers chosen in the range $[0, S-1]$ and the related public keys I are computed in the group $\mathcal{G}$ by the relation $I = g^s$.

Protocol (see figure 2). We let $\Phi = (B-1)(S-1)$. A round of identification consists for the prover in randomly choosing an integer r in $[0, A-1]$, and computing the *commitment* $x = g^r$. Next, he sends x to the verifier, who answers a *challenge* c randomly chosen in $[0, B-1]$. The prover checks $c \in [0, B-1]$ and computes the integer $y = r + c \times s$. He sends y to the verifier who checks $g^y = x \times I^c$ and $y \in [0, A + \Phi - 1]$.

A complete identification consists in repeating ℓ times the elementary round. The theoretical analysis shows that ℓ should be super-logarithmic in the security parameter in order to be able to prove the security of the scheme against active adversaries, as in the Schnorr scheme. However, in many practical applications, ℓ will often be equal to $\ell = 1$.

Parameters: ℓ, A, B and S integers
g an element of a multiplicative group $\mathcal{G}$
Secret key: $s \in [0, S-1]$
Public key: $I = g^s$

Prover		Verifier
Repeat ℓ times		
choose r in $[0, A-1]$		
$x = g^r$	$\xrightarrow{x}$	
check $c \in [0, B-1]$	$\xleftarrow{c}$	choose c in $[0, B-1]$
$y = r + c \times s$	$\xrightarrow{y}$	check $g^y \stackrel{?}{=} x \times I^c$ and $y \in [0, A + (B-1) \times (S-1) - 1]$

Fig. 2. GPS identification scheme

As usual in this kind of scheme, many straightforward variants can be designed, such as choosing $I = g^{-s}$ and/or $y = r - c \times s$, with some trivial impact on the rest of the protocol.

2.3 GPS signature scheme

We can turn the identification scheme into a signature scheme by following the technique originally proposed by Fiat and Shamir [18], and used by Schnorr [42] and many others: challenges c are no longer randomly chosen by a verifier but computed by means of a hash function h with output range $[0, B-1]$, with B larger than in the identification scheme.

The signature of a message m is computed by taking a random r in $[0, A-1]$ and computing $x = g^r$, $c = h(m, x)$ and $y = r + cs$. This produces the signature (x, c, y) that may be checked by anybody using the equations $c = h(m, x)$, $y \in [0, A + (B-1) \times (S-1) - 1]$ and $g^y = xI^c$.

Furthermore, well known optimizations, described in section 5.2, such as reducing the signature to the pair (c, y) can be applied. This leads to the following signature scheme:

Input:

- public parameters $(\mathcal{G}, g, A, B, S)$
- hash function $h(.)$, with output range $[0, B-1]$
- signer's private key s
- message encoded as an integer m

Operations: signature (c, y) shall be computed by the following sequence of steps:

1. Randomly generate an integer r from the range $[0, A-1]$.
2. Compute $x = g^r$.
3. Compute $c = h(m, x)$.
4. Compute $y = r + c \times s$.
5. Output (c, y).

A signature is verified using the following scheme:
Input:

- public parameters $(\mathcal{G}, g, S, A, B)$
- hash functions $h(.)$
- signer's public key I
- message encoded as an integer m
- signature to be verified (c, y), a pair of integers

Output: "valid" if m and (c, y) are consistent given the public key; "invalid" otherwise

Operations: output shall be computed by the following sequence of steps:

1. If c is not in $[0, B-1]$ or y is not in $[0, A + (B-1) \times (S-1) - 1]$ output "invalid" and stop.
2. Compute $x' = g^y / I^c$.
3. Compute $c' = h(m, x')$.
4. If $c' = c$ then output "valid" else output "invalid".

3 Security analysis of the GPS identification scheme

The aim of this section is to formally prove the security of the GPS identification scheme. We first define the security model we use. Next, in order to prove the security of the GPS protocol against active adversaries, we follow the approach of Feige, Fiat and Shamir [16], proving completeness, zero-knowledge and soundness.

Another strategy to demonstrate the security against active adversaries is to prove that GPS is witness indistinguishable [17]. In [38], Pointcheval proved that the GPS scheme enjoys this property for some specific group $\mathcal{G}$ and base $g \in \mathcal{G}$.

3.1 Security model

By means of an identification scheme a prover convinces a verifier of his identity. Both the prover and the verifier are modeled as probabilistic polynomial time Turing machines (Pptm). They have a special tape, denoted ω, initially filled with randomly and uniformally chosen bits. They also have additional tapes where they can read and/or write the messages that they exchange. See [30] for a complete definition of interactive Pptms.

We consider the following scenario for identification; firstly a randomized algorithm generates public parameters on input the security parameter k. Its running time is polynomial in k. Next a second probabilistic algorithm, using random tape ω_K, generates pairs of public and private keys (pk,sk) and sends the secret key to the prover while the related public key is made available to anybody, including of course the prover and the verifier. Finally, the identification is an interactive protocol between the prover and the verifier which respectively use random tapes ω_P and ω_V. At the end, the verifier accepts or not.

We now modify this scenario, where everybody is honest, in order to add an attacker whose aim is to impersonate the prover, i.e. to be accepted by a verifier with the public key of the prover. In this model, we consider an attacker that does not corrupt public parameters and key generation. Thus, there are only two ways for him to obtain information. Firstly, the attacker can passively observe the communication during regular authentications between the prover and the verifier. Secondly, he can take control over the verifier. The difference between the passive and the active attacks is that the active attacker can make the verifier deviate from the protocol in order to try to extract more information about the prover's secret key.

In the active scenario, we can view the attacker and the verifier under control as a single machine. Accordingly, an attacker is made of two probabilistic polynomial time Turing machines; the first one, "attacker A_1", interacts with a prover and sequentially executes a polynomial number of identifications while the second one, "attacker A_2", acts as a prover and tries to impersonate the original prover. Of course, the first attacker can transmit some information to the second one but the contrary is not allowed. Notice that such a security model does not take into account concurrent attacks where the attacker performs parallel authentications with the prover [14] or reset attacks where he can reset the prover in a former state [9, 2]. Furthermore, classical man-in-the-middle attacks cannot be performed since we separate interactions with the prover from those with the verifier.

We can now define what is a secure identification protocol in this model: a protocol is secure if the probability for any probabilistic polynomial time attacker (A_1, A_2) to be accepted is negligible:

$$\forall d \in \mathbb{N} \ \ \exists k_0 \ \ \forall k > k_0 \quad \Pr\left[\text{Verifier accepts } A_2\right] < \frac{1}{k^d}$$

where the probability is computed over all the random tapes.

3.2 The discrete logarithm with short exponent problem

For efficiency reasons, GPS secret keys are chosen in the range $[0, S-1]$ and not modulo the (possibly unknown) order of g. As a consequence, the security of GPS is not reduced to the discrete logarithm problem but, more precisely, to the so-called *discrete logarithm with short exponent problem*. Among other studies, this problem has been used by [45] in the context of the Diffie-Hellman key agreement scheme and also by [37, 20] in the context of provably secure pseudorandom generators. Of course, if S is chosen greater or equal to the order of g then the security assumption is reduced to the ordinary intractability of computing discrete logarithms in $\mathcal{G}$ in base g.

We assume the existence of a randomized algorithm $\mathcal{PP}(\omega_{pp}, k)$ that generates public parameters $\mathcal{G}$, g and S from a security parameter k using a random tape ω_{pp}. The *discrete logarithm with short exponent problem* consists, given inputs of $\mathcal{G}$, g, S and g^x s.t. $x \in [0, S-1]$, to find x.

The intractability assumption we will further use in the security proof of GPS is as follows:

Discrete Logarithm with short exponent assumption. For every polynomial Q and every probabilistic polynomial time Turing machine $\mathcal{M}$ running on random tape ω_M, for sufficiently large k,

$$\Pr_{\omega_{pp}, \omega_M} \left[\mathcal{M}(\mathcal{G}, g, S, g^x) = x \text{ where } (\mathcal{G}, g, S) \leftarrow \mathcal{PP}(\omega_{pp}, k) \text{ and } x \in [0, S-1]\right] < \frac{1}{Q(k)}$$

3.3 Security analysis of GPS

In the case of GPS, some public parameters are the group $\mathcal{G}$, the element $g \in \mathcal{G}$ and the bound S which are generated according to the security parameter k. The exact way of generating those parameters depends on the kind of cryptographic group $\mathcal{G}$ that is used but, roughly speaking, k defines the "size" of $\mathcal{G}$ and S in such a way that the discrete logarithms with short exponent problem may be assumed to be intractable, i.e. that there should not exist any polynomial algorithm in the security parameter k able to solve this problem.

In order to prove the security of GPS, we first prove in theorem 1 that honest provers are correctly authenticated. Next, we need to prove than an attacker (A_1, A_2), as described in section 3.1, cannot be accepted with non-negligible probability. Firstly, A_1 interacts with a prover. The zero-knowledge property, proven in theorem 2, shows that the communication between the A_1 and a prover can be simulated. This means that the prover can be replaced by a simulator, who does not know any secret. Since the simulated communication is indistinguishable from real ones, the attacker cannot detect this change. So, the attacker learns as much information from the simulator as from the real prover and we conclude that no information about the secret is leaked during the execution of the protocol.

Then, we show that if the second part of the attacker, A_2, is accepted with non-negligible probability, it can be used to solve in polynomial time a problem that is assumed to be intractable. Such a proof modifies the key generation algorithm but the distribution of the keys remains indistinguishable from the real one. In conclusion, we obtain that, if a PPTM attacker exists in our model, the discrete logarithm with short exponent problem can be solved in expected polynomial time in the security parameter k. In applications where this problem is assumed to be intractable, we conclude that the GPS scheme is secure against the active adversaries we consider.

Let us introduce some notation. For any integer x, $|x|$ is the number of bits $(\lfloor \log_2(x) \rfloor + 1)$ of x, and $\mathrm{abs}(x)$ denotes the absolute value of x. We use function $\boldsymbol{\delta}$, defined by $\boldsymbol{\delta}\,(true) = 1$ and $\boldsymbol{\delta}\,(false) = 0$. Finally, we denote by $\wedge$ the logical operator "and".

Implicitly, we consider that S, A, B and ℓ are functions of the security parameter k. In order to simplify notations, we do not explicitly write the dependencies on k but, when we say that a positive expression f is negligible, this means that f depends on k and that, for any constant d and for large enough k, $f(k) < 1/k^d$.

Theorem 1 (Completeness). *The execution of the protocol between a prover who knows the secret key corresponding to his public key and a verifier is always successful.*

Proof. At the end of each round, the verifier obtains $x = g^r$ and $y = r + cs$, which can be easily computed by the prover if he knows the secret key s. Consequently,

$$g^y = g^{r+sc} = g^r \times I^c = xI^c$$

Furthermore, $0 \leq y = r + cs \leq (A-1) + (B-1) \times (S-1) < A + \varPhi$. □

Theorem 2 (Zero-knowledge). *The GPS protocol is statistically zero-knowledge if ℓ and B are polynomial and $\ell SB/A$ is negligible.*

Proof. We describe an expected polynomial time simulation of the communication between a prover P and a dishonest verifier A_1 who can use an adaptive strategy to bias the choice of the challenges in order to try to obtain information about s. In this case, the challenges are no longer randomly choosen and this must be taken into account in the security proof. If we focus on the i^{th} round of identification, A_1 has already obtained data, denoted by $hist$, from previous interactions with P. Then the prover sends the commitment x_i and A_1 chooses, possibly using $hist$, x_i and bits from its random tape ω_A, the challenge $c_i(x_i, hist, \omega_A)$.

Here is an algorithm that uses a random tape ω_M to simulate the i^{th} round of identification by the usual method of resettable simulation:

step 1. using ω_M, choose random values $\overline{c_i} \in [0, B-1]$ and $\overline{y_i} \in [\varPhi, A-1]$ (recall that $\varPhi = (B-1)(S-1)$),

step 2. compute $\overline{x_i} = g^{\overline{y_i}}/I^{\overline{c_i}}$,

step 3. if $c_i(\overline{x_i}, hist, \omega_A) \neq \overline{c_i}$ then return to step 1 and try again with another pair $(\overline{c_i}, \overline{y_i})$, else return $(\overline{x_i}, \overline{c_i}, \overline{y_i})$.

The rest of the proof shows that, provided Φ is much smaller than A, this simulation algorithm outputs triplets statistically indistinguishable from real ones, for any fixed random tape ω_A. The main goal is to justify the intuition according which the distribution of $\overline{x_i} = g^{\overline{y_i}}/I^{\overline{c_i}}$, as computed by the simulator, and the distribution of g^r, as chosen by the real prover, are statistically close.

Let us prove that the distribution of the generated triplets is *statistically indistinguishable* from the distribution of real triplets, i.e. formally that

$$\Sigma_1 = \sum_{\alpha,\beta,\gamma} \left| \Pr_{\omega_P}\left[(x,c,y) = (\alpha,\beta,\gamma)\right] - \Pr_{\omega_M}\left[(\overline{x},\overline{c},\overline{y}) = (\alpha,\beta,\gamma)\right] \right|$$

is negligible. This means that the two distributions cannot be distinguished by any algorithm, even using an infinite computational power, but only accessing a polynomial number of triplets of both distribution. We refer to [30] for more details on this definition.

Let (α,β,γ) be a fixed triplet. Let us evaluate the respective probabilities to obtain such a triplet during one round of proof and during simulation.

We assume that the prover is honest, i.e. follows the protocol. Consequently,

$$\begin{aligned}
&\Pr_{\omega_P}\left[(x,c,y) = (\alpha,\beta,\gamma)\right] \\
&= \Pr_{r\in[0,A-1]}\left[\alpha = g^r \wedge \beta = c(\alpha, hist, \omega_A) \wedge \gamma = r + \beta\times s\right] \\
&= \sum_{r\in[0,A-1]} \frac{1}{A}\boldsymbol{\delta}\left(\alpha = g^\gamma/I^\beta \wedge \beta = c(\alpha, hist, \omega_A) \wedge r = \gamma - \beta\times s\right) \\
&= \frac{1}{A}\boldsymbol{\delta}\left(\alpha = g^\gamma/I^\beta \wedge \beta = c(\alpha, hist, \omega_A) \wedge \gamma - \beta\times s \in [0, A-1]\right) \\
&= \frac{1}{A}\times\boldsymbol{\delta}\left(\alpha = g^\gamma/I^\beta\right)\times\boldsymbol{\delta}\left(\beta = c(\alpha, hist, \omega_A)\right)\times\boldsymbol{\delta}\left(\gamma - \beta\times s\in[0,A-1]\right) \qquad (\star)
\end{aligned}$$

We now consider the probability $\Pr_{\omega_M}\left[(\overline{x},\overline{c},\overline{y}) = (\alpha,\beta,\gamma)\right]$ to obtain the triplet (α,β,γ) during the simulation described above. This is a conditional probability given by:

$$\Pr_{\overline{y}\in[\Phi,A-1],\overline{c}\in[0,B-1]}\left[\alpha = g^{\overline{y}}/I^{\overline{c}} \wedge \beta = \overline{c} \wedge \gamma = \overline{y} \;\middle|\; \overline{c} = c(g^{\overline{y}}/I^{\overline{c}}, hist, \omega_A)\right]$$

Using the definition of conditional probabilities, this can be written as

$$\frac{\Pr_{\overline{y}\in[\Phi,A-1],\overline{c}\in[0,B-1]}\left[\alpha = g^{\overline{y}}/I^{\overline{c}} \wedge \beta = \overline{c} = c(\alpha, hist, \omega_A) \wedge \gamma = \overline{y}\right]}{\Pr_{\overline{y}\in[\Phi,A-1],\overline{c}\in[0,B-1]}\left[\overline{c} = c(g^{\overline{y}}/I^{\overline{c}}, hist, \omega_A)\right]}$$

Let $Q = \sum_{\overline{y}\in[\varPhi,A-1],\overline{c}\in[0,B-1]} \boldsymbol{\delta}\left(\overline{c} = c(g^{\overline{y}}/I^{\overline{c}}, hist, \omega_A)\right)$. We obtain that the denominator of the previous fraction is

$$\Pr_{\overline{y}\in[\varPhi,A-1],\overline{c}\in[0,B-1]}\left[\overline{c} = c(g^{\overline{y}}/I^{\overline{c}}, hist, \omega_A)\right] = \frac{Q}{(A-\varPhi)\times B}$$

We now return to the evaluation of $\Pr_{\omega_M}[(\overline{x},\overline{c},\overline{y}) = (\alpha,\beta,\gamma)]$

$$\begin{aligned}
&= \sum_{\overline{c}\in[0,B-1]} \frac{1}{B} \Pr_{\overline{y}\in[\varPhi,A-1]}\begin{bmatrix}\alpha = g^{\overline{y}}/I^{\beta} \ \wedge\ \gamma = \overline{y}\ \wedge \\ \beta = \overline{c} = c(\alpha, hist, \omega_A)\end{bmatrix} \Bigg/ \frac{Q}{(A-\varPhi)\times B} \\
&= \Pr_{\overline{y}\in[\varPhi,A-1]}\begin{bmatrix}\alpha = g^{\gamma}/I^{\beta} \ \wedge\ \gamma = \overline{y}\ \wedge \\ \beta = c(\alpha, hist, \omega_A)\end{bmatrix} \times \frac{A-\varPhi}{Q} \\
&= \sum_{\overline{y}\in[\varPhi,A-1]} \frac{1}{A-\varPhi} \times \boldsymbol{\delta}\begin{pmatrix}\alpha = g^{\gamma}/I^{\beta} \ \wedge\ \gamma = \overline{y}\ \wedge \\ \beta = c(\alpha, hist, \omega_A)\end{pmatrix} \times \frac{A-\varPhi}{Q} \\
&= \frac{1}{Q} \times \boldsymbol{\delta}\left(\alpha = g^{\gamma}/I^{\beta}\right) \times \boldsymbol{\delta}\left(\beta = c(\alpha, hist, \omega_A)\right) \times \boldsymbol{\delta}\left(\gamma \in [\varPhi, A-1]\right) \qquad (\star\star)
\end{aligned}$$

Comparing $(\star)$ and $(\star\star)$, we see that, in order to proceed with the indistinguishability proof, we have to show that Q is close to A. The question is how many pairs $(\overline{c},\overline{y}) \in [0,B-1]\times[\varPhi,A-1]$ satisfy $\overline{c} = c(g^{\overline{y}}/I^{\overline{c}}, hist, \omega_A)$? The answer is provided by the following combinatorial lemma the proof of which appears in appendix A:

Lemma 3. *If f is a function from $\mathcal{G}$ to $[0,B-1]$ and $I \in \{g^s; s\in[0,S-1]\}$ then the total number N of solutions $(c,y)\in[0,B-1]\times[\varPhi,A-1]$ of the equation $c = f(g^y/I^c)$ satisfies $A-2\varPhi \le N \le A$.*

We can specialize the result of lemma 3 by setting f to the function which computes $c(g^{\overline{y}}/I^{\overline{c}}, hist, \omega_A)$ from $(\overline{c},\overline{y})$. Consequently we obtain that Q is between $A-2\varPhi$ and A.

We are now able to bound the distance $\varSigma_1$ between the actual and simulated distributions:

$$\begin{aligned}
\varSigma_1 &= \sum_{\alpha,\beta,\gamma}\left|\Pr_{\omega_P}[(x,c,y) = (\alpha,\beta,\gamma)] - \Pr_{\omega_M}[(\overline{x},\overline{c},\overline{y}) = (\alpha,\beta,\gamma)]\right| \\
&= \sum_{\alpha,\beta,\gamma\in[\varPhi,A-1]}\left|\Pr_{\omega_P}[(x,c,y) = (\alpha,\beta,\gamma)] - \Pr_{\omega_M}[(\overline{x},\overline{c},\overline{y}) = (\alpha,\beta,\gamma)]\right| \\
&\quad + \sum_{\alpha,\beta,\gamma\notin[\varPhi,A-1]} \Pr_{\omega_P}[(x,c,y) = (\alpha,\beta,\gamma)]
\end{aligned}$$

$$= \sum_{\gamma\in[\Phi,A-1],\beta\in[0,B-1],\alpha=g^\gamma/I^\beta} \left| \begin{array}{l} \frac{1}{A}\times \boldsymbol{\delta}\left(\beta = c(\alpha, hist, \omega_A)\right) \\ -\frac{1}{Q}\times \boldsymbol{\delta}\left(\beta = c(\alpha, hist, \omega_A)\right) \end{array} \right|$$

$$+ \left(1 - \sum_{\alpha,\beta,\gamma\in[\Phi,A-1]} \Pr_{\omega_P}\left[(x,c,y) = (\alpha,\beta,\gamma)\right]\right)$$

$$= \left(\left|\frac{1}{A} - \frac{1}{Q}\right| \times Q\right) + 1 - \sum_{\gamma\in[\Phi,A-1],\beta\in[0,B-1],\alpha=g^\gamma/I^\beta} \frac{1}{A}\boldsymbol{\delta}\left(\beta = c(\alpha, hist, \omega_A)\right)$$

$$= \frac{|Q-A|}{A} + 1 - \frac{Q}{A} \le 2\frac{|Q-A|}{A} \le \frac{4\Phi}{A} < \frac{4SB}{A}$$

This proves that the real and simulated distributions are statistically indistinguishable if SB/A is negligible.

We finally explain the reason why the machine M runs in expected polynomial time. Step 3 outputs a triplet $(\overline{x_i}, \overline{c_i}, \overline{y_i})$ if $c(\overline{x_i}, hist, \omega_A) = \overline{c_i}$. We have already proven that

$$\Pr_{\overline{y}\in[\Phi,A-1],\overline{c}\in[0,B-1]}\left[\overline{c} = c(g^{\overline{y}}/I^{\overline{c}}, hist, \omega_A)\right] = \frac{Q}{(A-\Phi)\times B}$$

and that $A - 2\Phi \le Q \le A$ so the probability of success at step 3 is bounded beween $\frac{1}{B}\left(1 - \frac{\Phi/A}{1-\Phi/A}\right)$ and $\frac{1}{B}\left(\frac{1}{1-\Phi/A}\right)$. Since SB/A is negligible, this probability is essentially $1/B$ and the expected number of executions of the loop is B. Consequently, the complexity of the simulation of the ℓ rounds is $O(\ell \times B)$.

In conclusion, if $\ell SB/A$ is negligible and if ℓ and B are polynomial, the GPS protocol is statistically zero-knowledge. □

Since GPS is statistically zero-knowledge, we know that interactions with a prover cannot help an attacker in our model. Consequently, the end of the security proof of GPS consists in proving that, if the verifier accepts, then, with overwhelming probability, the prover must know the discrete logarithm of I in base g. Intuitively, after one commitment x has been sent, if the prover can correctly answer with probability $> 1/B$, he must be able to answer to two different challenges c and c' with y and y', smaller than $A + \Phi$, such that $g^y/I^c = x = g^{y'}/I^{c'}$. Let $\sigma = y - y'$ and $\tau = c - c'$; we obtain $g^\sigma = I^\tau$. The following lemma, where ε is implicitly assumed to depend on the security parameter k, turns those ideas in more formal terms:

Lemma 4. *Assume that some prover is accepted for a public key I with probability $\varepsilon > 1/B^\ell$. Then there exists an algorithm which outputs*

$$\sigma \in [-(A+\Phi-1), A+\Phi-1] \quad \text{and} \quad \tau\in[1,B-1] \quad \text{such that} \quad g^\sigma = I^\tau$$

with probability $> \frac{1}{6}\left(\frac{\varepsilon - 1/B^\ell}{\varepsilon}\right)^2$. The expected running time is $< 2/\varepsilon \times T$, where T is the average running time of an execution of the identification protocol.

Proof. The proof of this lemma appears in appendix B. It is quite similar to the extractor of the Schnorr scheme [43].

We now meet the main difference between the Schnorr proof and GPS. If the order of g were known and relatively prime with any integer in the range $[1, B-1]$, then, exactly as in the Schnorr scheme where g is of prime order $\mathrm{ord}(g) = q$, it would be very easy to recover the secret s from the equation $g^\sigma = I^\tau$, just by solving the equation $\sigma - s\tau = 0 \bmod \mathrm{ord}(g)$. When the order of g is unknown, we cannot solve this equation. A consequence is that GPS is not a proof of knowledge of a discrete logarithm because logarithms cannot be extracted from accepted provers. However, we can prove its security in our model, assuming the sole intractability of computing short discrete logarithms in base g, modulo n. Let us first recall a well known probabilistic lemma (see for example [39]) :

Lemma 5 (Splitting Lemma). *Let $A \subset X \times Y$, such that $\Pr_{x,y}[A(x,y)] \geq \varepsilon$, and $\Omega = \left\{ a \in X \mid \Pr_y[A(a,y)] \geq \varepsilon/2 \right\}$ then $\Pr_x[x \in \Omega] \geq \varepsilon/2$.*

Theorem 6 (Security of GPS). *Assume that an adversary (A_1, A_2) is such that after interactions between A_1 and a prover, A_2 is accepted with non-negligible probability by honest verifiers. Further assume that ℓ and B are polynomial while $\ell SB/A$ and $1/B^\ell$ are negligible, relatively to the security parameter k. Then there exists an algorithm that solves the discrete logarithm with short exponent problem in expected polynomial time.*

Proof. The basic idea of the proof is to show that if an adversary can impersonate a prover, he can compute discrete logarithms with short exponent in expected polynomial time. We have already seen in lemma 4 that if a prover is accepted with non negligible probability, he must know integers σ and τ such that $g^\sigma = I^\tau$. Unfortunately, we cannot immediately deduce the discrete logarithm of I in base g from this equation since we do not know the order of g. However, we show in this proof that only two situations are possible and that in those two cases we can finally compute discrete logarithms.

In the first case, the exponents σ are most of the time multiples of τ so it is easy to simplify the equation $g^\sigma = I^\tau$ and to compute $\log_g I$. In the second case, we consider the opposite situation in which τ does not usually divide σ; we can no longer compute $\log_g I$ but, if we already know that $I = g^{s_0}$, we learn that $\sigma - s_0\tau$ is a non-zero multiple of the multiplicative order of g. Then, this information finally enables to solve the discrete logarithm problem for values I' for which we do not previously know the logarithm. Some technical details are now provided.

In the proof, we fix the group $\mathcal{G}$ and the base g; we consider an adversary accepted with probability π, where the probability is considered over the random tapes ω_K (for the choice of the private key s), ω_A (for the attacker random choices) and ω_V (for the verifier random choices). We let $\mathcal{I} = \{I = g^s,\ s \in [0, S-1]\}$ the set of all public keys. Let I_0 be such a key chosen in $\mathcal{I}$. We now describe an algorithm that uses the adversary (A_1, A_2) to compute the discrete logarithm of I_0 in base g.

Notice that, in order to make the proof as simple as possible, we present a non-uniform algorithm for computing discrete logarithms, i.e. an algorithm that depends on the probability of success of the attacker. However, since the actual value of this probability is not used but just allows to estimate the complexity, it could easily be transformed into a uniform algorithm, by just running in parallel all Turing machines described below.

Firstly, notice that the interaction between A_1 and a real prover can be simulated in expected polynomial time as explained in theorem 2. Consequently, the information transmitted by A_1 to A_2 in our security model can be output by a probabilistic Turing machine that does not know any secret. Furthermore, the program of this machine can even be included in the program of A_2.

Then, in our security model, the probability of success π for an attacker is

$$\Pr_{I \in \mathcal{I}, \omega_A, \omega_V} [\text{the adversary } (A_1, A_2) \text{ is accepted with the public key } I] = \pi$$

The probability is taken over those random tapes and also over the public keys so that the probability of success can be much smaller for some specific keys. However, for a non-negligible part of the keys, the probability of success is “not too small”. More formally, let $\mathcal{I}_0$ be the subset of the public keys I such that

$$\Pr_{\omega_A, \omega_V} [\text{the adversary is accepted with the public key } I] \geq \pi/2$$

Lemma 5 proves that the probability for a public key I to be in this subset $\mathcal{I}_0$ is greater than $\pi/2$. Since the probability of success of the attacker π is non-negligible while $1/B^\ell$ is assumed to be negligible, we consider large enough values of the security parameter k for which $\pi/2 > 1/B^\ell$. In this case we can use the result of lemma 4 that shows the existence of a PPTM $\mathcal{M}(I)$ which outputs

$$\sigma \in [-(A + \Phi - 1), A + \Phi - 1] \quad \text{and} \quad \tau \in [1, B-1] \quad \text{such that} \quad g^\sigma = I^\tau$$

with probability $\varepsilon > (\pi/2 - 1/B^\ell)^2/(6(\pi/2)^2)$, in time $T' < 4/\pi \times T$.

Thus, the probability that a public key g^s is in $\mathcal{I}_0$ and that $\mathcal{M}(I)$ outputs (σ, τ) such that $g^\sigma = I^\tau$ is larger than $\pi/2 \times \varepsilon$.

$$\Pr_{s \in [0, S-1]} [g^s \in \mathcal{I}_0 \wedge \mathcal{M}(g^s) \text{ outputs } (\sigma, \tau)] \geq \frac{\pi\varepsilon}{2}$$

Two situations can occur depending on the probability for $\mathcal{M}(I)$ to output σ and τ such that $\sigma - s\tau = 0$:

– First case: if most of the time $\mathcal{M}(I)$ outputs (σ, τ) such that $\sigma - s\tau = 0$, we immediately obtain the discrete logarithm $s = \sigma/\tau$ that we are looking for,
– Second case: on the contrary, if $\mathcal{M}(I)$ outputs (σ, τ) such that $\sigma - s\tau \neq 0$, we obtain a multiple of the multiplicative order of g in $\mathcal{G}$. Then this information enables to solve equations such as $\sigma' - x\tau' = 0 \bmod \mathrm{ord}(g)$ and consequently to compute discrete logarithms from the outputs of $\mathcal{M}(I)$.

First case: if the probability that a public key g^s is in $\mathcal{I}_0$ and that the PPTM $\mathcal{M}(I)$ outputs (σ, τ) such that $\sigma - s\tau = 0$ is greater than $\pi\varepsilon/4$,

$$\Pr_{s \in [0, S-1]} [g^s \in \mathcal{I}_0 \wedge \mathcal{M}(g^s) \text{ outputs } (\sigma, \tau) \wedge \sigma - s\tau = 0] \geq \frac{\pi\varepsilon}{4}$$

we can immediately compute the discrete logarithm of the target public key I_0 by means of the following algorithm:

step 1. choose $r \in [-(S-1), S-1]$,

step 2. compute $I' = I_0 \times g^r$,

step 3. run $\mathcal{M}$ on input I',

step 4. if $\mathcal{M}(I')$ outputs (σ, τ) such that τ divides σ, $\sigma/\tau - r \in [0, S-1]$ and $I_0 = g^{\sigma/\tau - r}$, output $\log_g I_0 = \sigma/\tau - r$; otherwise restart at step 1.

Notice that if $I_0 \in \mathcal{I}$ and $r \in [-(S-1), S-1]$, the probability for I' to be in $\mathcal{I}$ is $1/2$. Furthermore, if $I' \in \mathcal{I}$, it is uniformly distributed. Consequently, using the fact we are in the "first case", we obtain that this algorithm finds $\log_g I_0$ after about $8/(\pi\varepsilon)$ executions of the loop on average. Each loop calls $\mathcal{M}$ once so the expected running time of this algorithm is $O(T'/(\pi\varepsilon)) = O(T/\pi^2\varepsilon)$.

Second case: we now consider the case where $\mathcal{M}$ does not directly output secret keys:

$$\Pr_{s \in [0, S-1]} [g^s \in \mathcal{I}_0 \wedge \mathcal{M}(g^s) \text{ outputs } (\sigma, \tau) \wedge \sigma - s\tau \neq 0] \geq \frac{\pi\varepsilon}{4}$$

The first step consists in computing a multiple of the multiplicative order of g. We use the following algorithm:

step 1. choose $s_0 \in [0, S-1]$,

step 2. compute $I = g^{s_0}$,

step 3. run $\mathcal{M}$ on input I,

step 4. if $\mathcal{M}$ outputs (σ, τ) such that $g^\sigma = I^\tau$ and $L_0 = \mathrm{abs}(\sigma - s_0\tau) \neq 0$ output L_0; otherwise restart at step 1.

After an expected running time $O(T'/(\pi\varepsilon))$, we obtain $L_0 \neq 0$ such that $g^{L_0} = 1$. Consequently, L_0 is a multiple, smaller than $A + \Phi$, of the order of g in $\mathcal{G}$. We can now compute discrete logarithm of I_0 in base g using the following algorithm:

step 1. choose $r \in [-(S-1), S-1]$,
step 2. compute $I' = I_0 \times g^r$,
step 3. run $\mathcal{M}$ on input I',
step 4. if $\mathcal{M}$ does not output (σ, τ) s.t. $g^\sigma = I'^\tau$ restart at step 1; otherwise output (σ, τ).

We obtain I', σ and τ such that $I'^\tau = g^\sigma$. In order to find the discrete logarithm of I', we solve the equation $\sigma - \tau x = 0 \bmod L_0$. Let $d = \gcd(\tau, L_0)$; since τ/d and L_0/d are relatively prime, the equation $(\sigma/d) - (\tau/d) \times x = 0 \bmod L_0/d$ has exactly one solution $x_0 \bmod L_0/d$,

$$x_0 = (\sigma/d) \times (\tau/d)^{-1} \bmod L_0/d$$

We now consider the equation $\sigma - \tau \times x = 0$ modulo L_0 and not only modulo L_0/d. As a consequence, we can write $x = x_0 + i \times L_0/d \bmod L_0$ with $i \in [0, d-1]$. The solution x such that $I' = g^x$ can finally be found among those d solutions. We can of course use exhaustive search since $d < \tau < B$ but a Baby-step Giant-step algorithm allows to find the solution in time $O(\sqrt{B})$.

In conclusion, in time $O(T'/(\pi\varepsilon) + \sqrt{B})$ we obtain s_0 such that $I_0 = g^{s_0}$. A final problem is that s_0 may not be in the range $[0, S-1]$; we now describe an iterative algorithm that finally outputs a short discrete logarithm in this range.

If the probability over $s \in [0, S-1]$ that $\mathcal{M}(g^s)$ outputs (σ, τ) and that the previously described algorithm computes exactly s is larger that $\pi\varepsilon/4$, we obtain $s_0 \in [0, S-1]$ in time $O(T'/(\pi\varepsilon) + B)$. Otherwise, we can run the algorithm with $I = g^s$ and we obtain s' such that $g^s = I = g^{s'}$ but $s \neq s'$. Consequently L_0 and $(s - s') \bmod L_0$ are multiples of the order of $g$so $L_0' = \gcd(L_0, (s-s') \bmod L_0)$ is also such a multiple but $L_0' \leq L_0/2$. We obtain a new value for L_0 and we restart the procedure. When the size of L_0 decreases, we are finally able to compute the discrete logarithm of I_0 in the range $[0, S-1]$. The recursive step is repeated less than $|A + \Phi|$ times because $|A + \Phi|$ is the size of the first value of L_0. Finally, the expected time complexity of this algorithm is $O(|A + \Phi|(T'/(\pi\varepsilon) + \sqrt{B}))$.

If π is non-negligible and $1/B^\ell$ is negligible, for infinitely many values of k, $1/B^\ell < \pi/4$. Consequently, the probability of success of the PPTM $\mathcal{M}$ of lemma 4 is

$$\varepsilon > \frac{1}{6}\left(\frac{\pi/2 - 1/B^\ell}{\pi/2}\right)^2 > 1/24$$

and its expected running time T' is less than $4/\pi \times T$. Finally, if B is polynomial, we obtain an expected polynomial time algorithm to compute discrete logarithms in base g, in the range $[0, S-1]$. $\square$

4 Security analysis of the GPS signature Scheme

As explained in section 2.3, the GPS identification scheme is turned into a signature scheme by following the technique originally proposed by Fiat and Shamir [18]: challenges c are no longer randomly chosen by a verifier but computed by means of a hash function h with output range $[0, B-1]$. In order to avoid the parallel execution of a super-logarithmic number of rounds, we need to increase the bound B to be super-polynomial.

In order to prove the security of the GPS signature scheme, described in section 2.3, we can show that, if an attacker is able to forge valid signatures after having obtained signatures of messages of his choice, then we can use it to compute the secret key and consequently to solve the discrete logarithm with short exponent problem. The random oracle model [3] is used to simulate the behavior of the hash function so that the proof only validates the overall design.

The GPS signature scheme uses a cryptographically secure hash function. Ideally, a security proof should only be based on some intractability assumption such as the impossibility to find collisions. However, in order to obtain security arguments, we need to simulate the hash function as a random function, following the initial idea of Bellare and Rogaway [3, 18]. In this model, the hash function is not considered as a deterministic public function but it is modeled by an oracle that randomly answers the queries. The only limitation is that this oracle provides the same answer if the same query is asked twice.

The use of the random oracle model is known to be a good engineering principle when it is not possible to provide proofs without such an additional assumption. This approach validates the design of the scheme even if we must be careful with this model as shown by Canetti *et al.* [8].

A generic result due to Abdalla *et al.* [1] shows that the use of the Fiat-Shamir paradigm to transform an identification scheme secure against passive attacks into a signature scheme leads to a secure signature scheme since even existential forgery under adaptive chosen message attack is impossible (see [31, 1] for standard definition of security of digital signature schemes). Consequently, the only property we need to prove is that the GPS identification scheme remains zero-knowledge if the bound B is increased to be super-polynomial but in a setting where the verifier is honest, i.e. randomly chooses the challenges in the range $[0, B-1]$. Going through the proof of theorem 2, we see that the simulation requires B to be polynomial. This is a well known restriction for zero-knowledge and appears in the Schnorr scheme as well. However, we can notice that if the verifier is honest the simulation complexity is only $O(\ell)$.

Theorem 7. *Under the discrete logarithm with short exponent assumption, if SB/A and $1/B$ are negligible, the GPS signature scheme is existentially unforgeable under adaptive chosen message attacks in the random oracle model.*

5 Applications

This section is more practical in character: we discuss how to choose secure parameters in order to resist the known attacks against factorization and the discrete logarithm problem. Then we explain how to optimize size of the data and, finally, report on the performance of a smart card application.

5.1 Choice of the underlying mathematical structure

Let us first focus on the group $\mathcal{G}$ and the base g. We already proposed in the description of GPS several possible options. We now clarify the practical choice of the parameters for the first two options:

- $\mathcal{G} = \mathbb{Z}_p^*$ with p a prime number s.t. $p-1$ has a large prime factor q; the order of the base g should be q and the size of q should be larger than 160 bits. The discrete logarithm problem modulo a prime integer seems currently intractable if the size of the modulus is larger than 1536 bits. For more secure applications, $|p| = 2048$ may be appropriate; we refer to specific overviews such as [33] for a more precise analysis.
- $\mathcal{G} = \mathbb{Z}_n^*$ with n an RSA modulus, i.e. a composite integer with two prime factors with almost the same size. The use of a 1536-bit modulus seems adequate to guarantee a high level of security based on the intractability of factorization for the next years. Then g can be randomly chosen. In practice, we advise the use of modulus n which is the product of two strong primes, i.e. primes p s.t. $(p-1)/2$ is also prime, in order to avoid partial Pohlig-Hellman attacks [45]. We also advise the use of the base $g = 2$ for efficiency reasons.

The security is related to the choice of $\mathcal{G}$, g and S in such a way that computing discrete logarithms in base g is intractable, even if those exponents are in the range $[0, S-1]$. Since discrete logarithms can be computed in $O(\sqrt{S})$ using Pollard *rho* algorithm or Shanks *baby-step giant-step* algorithm, S should be at least equal to 2^{160} and preferably to 2^{256} for a high level of security.

Then, the choice of the size of B is related to the probability of impersonation of an adversary. The expected security depends on the application and $B = 2^{32}$ with $\ell = 1$, i.e. using just one elementary round, would probably be large enough for many identification systems since it guarantees that an adversary cannot impersonate a user with probability larger than $1/2^{32}$. For signature applications, the use of a standard hash function such as SHA-256 [35] ($B = 2^{256}$) can be advised.

Finally, the parameter A must be s.t. A/SB is "large" in order to guarantee the statistical zero-knowledge property. We advise to take $A = S \times B \times 2^{80}$.

5.2 Optimization of coupons

In order to decrease the number of communication bits, Fiat and Shamir [18] have suggested to send a hash value of the commitment issued at the first step of the identification. This trick can be used with our scheme. Let h' be a hash function and $|h'|$ be the size of its output. The modifications are very simple: the commitment x is replaced in the protocol by $x' = h'(x)$ and the verification equation becomes $x' = h'(g^y/I^c)$.

Using the notion of t-collision-free hash functions, i.e. functions such that it is infeasible to find t distinct values with the same image, Girault and Stern [28] have precisely analyzed the consequences of such a modification on the security of identification schemes. This result can still be improved [23] if we consider that an attacker cannot perform more than a fixed number of on-line operations during the authentication process. In this setting, if we want a security level of 32 bits, we can choose $|h'| = 50$ bits only.

Finally, we have already observed that the commitments can be computed off-line, by the individual device or by an authority. In fact, we just have to compute and to keep in memory pairs of the form $(r, h'(g^r))$. Notice that the computation of the exponentiation can use the Chinese remainder theorem, in order to be more efficient, when the factorization of the modulus is known.

Memory space can be saved if the random values r are not stored but generated when needed by a pseudo-random generator. This finally leads to store the seed of the generator and the commitments, i.e. about only 6 bytes per authentication using [23] commitment hashing technique.

5.3 Performance

An implementation of GPS on a PC shows the very high practical efficiency of this scheme for identification and signature applications.

Parameters (see sections 2.2 for notations)	$\mathcal{G} = \mathbb{Z}_n^*$ with $\lvert n\rvert = 1536$ $(n = p \times q)$ $\lvert S\rvert = 2 \times 80 = 160$, $\ell = 1$ $\lvert B\rvert = 35$, $\lvert A\rvert = \lvert S\rvert + \lvert B\rvert + 80 = 275$
Parameters generation	≈ 1 s
Computation of commitment x	10.1 ms (5940 per minute)
Computation of answer y	< 1 μs
Verification	11.8 ms (5084 per minute)

Fig. 3. Implementation of GPS in C using GMP library on a PC PIII 450 MHz

Figure 3 shows the performances obtained with a Pentium III 450 MHz processor and a C program using the GMP multiprecision arithmetic library [29].

The public parameters and key generation needs about one second but this operation does not happen very often. The computation of commitments or of coupons just consists in computing an exponentiation. The verification is a similar operation. We see that we can perform a few thousands of such operation by minute with a simple PC. Note that the computation of the answer is so easy that we cannot really measure it. Also note that verification can be server-aided and made as efficient as Guillou-Quisquater verification [27].

5.4 Smart card application

In order to show to what extent computations are minimal and transmissions very short, we now present an application we have implemented on a low cost smart card based on a 6805 chip. The size of the program is very small, about 300 bytes. We see in figure 4 that the running time of the computation is very short and actually most of the time needed for an authentication is taken by the communication protocol between the card and the computer. Notice that, for signature, we would have to take into account the computation time of the hash function; this would probably be the bottleneck of many very fast applications. In conclusion, this demonstrates that the scheme under study is really suitable for very fast "on the fly" applications.

Parameters (see sections 2.2 for notations)	$\mathcal{G} = \mathbb{Z}_n^*$ with $\lvert n\rvert = 1536$ $\lvert S\rvert = 2 \times 80 = 160$, $\ell = 1$ $\lvert B\rvert = 35$, $\lvert A\rvert = \lvert S\rvert + \lvert B\rvert + 80 = 275$
Size of a coupon ($= \lvert h'\rvert$) Number of coupons in 4 KBytes	50 bits **655**
Running time at 3.57 MHz	$<$ **2 ms**
Amount of communication Running time at 9600 bauds at 115000 bauds	45 bytes 38 ms 3.1 ms
Total running time at 9600 bauds at 115000 bauds	 $\approx$ 40 ms $\approx$ **5 ms**

Fig. 4. Implementation of GPS on low cost smart card

Acknowledgments

We thank D. Bleichenbacher for his comment on the necessity of checking the range of the challenges and the anonymous referees for fruitful comments.

References

1. M. Abdalla, J.H. An, M. Bellare, and C. Namprempre. From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assuptions for Security and Forward-Security. In *Eurocrypt 2002*, LNCS 2332, pages 418–433. Springer-Verlag, 2002.
2. M. Bellare, M. Fischlin, S. Goldwasser, and S. Micali. Identification Protocols Secure against Reset Attacks. In *Eurocrypt 2001*, LNCS 2045, pages 495–511. Springer-Verlag, 2001.
3. M. Bellare and P. Rogaway. Random Oracles are Practical: a paradigm for designing efficient protocols. In *Proceedings of the 1st ACM-CCS*, pages 62–73. ACM press, 1993.
4. T. Beth. Efficient Zero-Knowledge Identification Scheme for Smart Cards. In *Eurocrypt '88*, LNCS 330, pages 77–86. Springer-Verlag, 1988.
5. I. Biehl, J. Buchmann, S. Hamdy, and A. Meyer. A Signature Scheme Based on the Intractability of Computing Roots. *Designs, Codes and Cryptography*, 25(3):223–236, March 2002.
6. E. Biham and A. Shulman. Memory Efficient Divisible Electronic Cash. In *CARDIS '98*, LNCS 1820. Springer-Verlag, 2000.
7. E. F. Brickell and K. S. McCurley. An Interactive Identification Scheme Based on Discrete Logarithms and Factoring. *Journal of Cryptology*, 5:29–39, 1992.
8. R. Canetti, O. Goldreich, and S. Halevi. The Random Oracle Methodology Revisited. In *Proceedings of the 30th STOC*, pages 209–218. ACM Press, 1998.
9. R. Canetti, S. Goldwasser, O. Goldreich, and S. Micali. Resettable Zero-Knowledge. In *Proceedings of the 32nd STOC*, pages 235–244. ACM Press, 2000.
10. D. Chaum, J. Evertse, and J. van de Graaf. An Improved Protocol for Demonstrating Possession of Discrete Logarithms and some Generalizations. In *Eurocrypt '87*, LNCS 304, pages 127–141. Springer-Verlag, 1988.
11. D. Chaum, J. Evertse, J. van de Graaf, and R. Peralta. Demonstrating Possession of a Discrete Logarithm without Revealing it. In *Crypto '86*, LNCS 263, pages 200–212. Springer-Verlag, 1987.
12. NESSIE consortium. *Portfolio of recommanded cryptographic primitives*, 2003. Available from `http://www.cryptonessie.org`.
13. J.S. Coron, H. Handschuh, and D. Naccache. ECC: Do We Need to Count? In *Asiacrypt '99*, LNCS 1716, pages 122–134. Springer-Verlag, 1999.
14. C. Dwork, M. Naor, and A. Sahai. Concurrent Zero-Knowledge. In *Proceedings of the 30th STOC*, pages 409–418. ACM Press, 1998.
15. S. Even, O. Goldreich, and S. Micali. On-line/off-line Digital Signatures. In *Crypto '89*, LNCS 435, pages 263–277. Springer-Verlag, 1990.
16. U. Feige, A. Fiat, and A. Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, 1:77–95, 1988.
17. U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *Proceedings of the 22nd STOC*, pages 416–426. ACM Press, 1990.
18. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Crypto '86*, LNCS 263, pages 186–194. Springer-Verlag, 1987.
19. P. A. Fouque and G. Poupard. On the Security of RDSA. In *Eurocrypt 2003*, LNCS 2656, pages 462–476. Springer-Verlag, 2003.
20. R. Gennaro. An Improved Pseudo-random Generator Based on Discrete Log. In *Crypto 2000*, LNCS 1880, pages 469–481. Springer-Verlag, 2000.
21. M. Girault. An Identity-Based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. In *Eurocrypt '90*, LNCS 473, pages 481–486. Springer-Verlag, 1991.
22. M. Girault. Self-Certified Public Keys. In *Eurocrypt '91*, LNCS 547, pages 490–497. Springer-Verlag, 1992.
23. M. Girault. Low-size Coupons for Low-cost IC Cards. In *CARDIS 2000*, volume 180 of *IFIP Conference Proceedings*, pages 39–50. Kluwer, 2000.
24. M. Girault and D. Lefranc. Public Key Authentication with one Single (on-line) Addition. In *CHES 2004*, LNCS 3156, pages 413–427. Springer-Verlag, 2004.

25. M. Girault and J.-C. Paillès. On-line/off-line RSA-like. In *Proceedings of WCC '03*, pages 173–184, 2003.
26. M. Girault, G. Poupard, and J. Stern. Some modes of use of the GPS identification scheme. In *Proceedings of the 3rd NESSIE Conference*. Springer-Verlag, 2002.
27. M. Girault and J.-J. Quisquater. GQ+GPS. Rump-Session of Eurocrypt 2002.
28. M. Girault and J. Stern. On the Length of Cryptographic Hash-Values used in Identification Schemes. In *Crypto '94*, LNCS 839, pages 202–215. Springer-Verlag, 1994.
29. *GNU Multiple Precision Arithmetic Library (GMP 4.0.1)*, 2002. Available from `http://www.swox.com/gmp`.
30. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM journal of computing*, 18(1):186–208, february 1989.
31. 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.
32. ISO/IEC 9798-5. Information Technology – Security techniques – Entity authentication – Part 5: Mechanisms using zero-knowledge techniques – Second edition, December 2004.
33. A. Lenstra and E. Verheul. Selecting Cryptographic Key Sizes. *Journal of Cryptology*, 14(4):255–293, 2001.
34. D. Naccache, D. M'Raïhi, S. Vaudenay, and D. Raphaeli. Can DSA be improved? In *Eurocrypt '94*, LNCS 950, pages 77–85. Springer-Verlag, 1995.
35. NIST. Secure Hash Standard (SHS). Federal Information Processing Standards PUBlication 180–2, august 2002.
36. T. Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In *Crypto '92*, LNCS 740, pages 31–53. Springer-Verlag, 1992.
37. S. Patel and G. Sundaram. An Efficient Discrete Log Pseudo Random Generator. In *Crypto '98*, LNCS 1462, pages 304–317. Springer-Verlag, 1998.
38. D. Pointcheval. The Composite Discrete Logarithm and Secure Authentication. In *PKC 2000*, LNCS 1751, pages 113–128. Springer-Verlag, 2000.
39. D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3), 2000.
40. G. Poupard and J. Stern. Security Analysis of a Practical "on the fly" Authentication and Signature Generation. In *Eurocrypt '98*, LNCS 1403, pages 422–436. Springer-Verlag, 1998.
41. G. Poupard and J. Stern. On The Fly Signatures based on Factoring. In *Proceedings of 6th ACM-CCS*, pages 37–45. ACM press, 1999.
42. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Crypto '89*, LNCS 435, pages 235–251. Springer-Verlag, 1990.
43. C. P. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.
44. A. Shamir and Y. Tauman. Improved Online/Offline Signature Schemes. In *Crypto 2001*, LNCS 2139, pages 355–367. Springer-Verlag, 2001.
45. P. C. van Oorschot and M. J. Wiener. On Diffie-Hellman Key Agreement with Short Exponents. In *Eurocrypt '96*, LNCS 1070, pages 332–343. Springer-Verlag, 1996.

A Proof of lemma 3

Lemma 3. If f is a function from $\mathcal{G}$ to $[0, B-1]$ and $I \in \{g^s; s \in [0, S-1]\}$ then the total number N of solutions $(c, y) \in [0, B-1] \times [\Phi, A-1]$ of the equation $c = f(g^y/I^c)$ satisfies $A - 2\Phi \leq N \leq A$.

Let f be any function from $\mathcal{G}$ to $[0, B-1]$. We first notice that, since $I = g^s$, if $f(g^y/I^c) = c$ then $f(g^{y+ks}/I^{c+k}) = c$ so $f(g^{y+ks}/I^{c+k}) \neq c + k$ for any $k \neq 0$ such that $c + k \in [0, B-1]$.

We define the following sets of pairs (c, y):

$$\mathcal{P}_X = \{(c, X + cs) \quad \text{such that} \quad c \in [0, B-1] \ \text{ and } \ X + cs \in [\Phi, A-1]\}$$

Those subsets have many properties that are summarized below:

(1) $\forall X \neq X' \quad \mathcal{P}_X \cap \mathcal{P}_{X'} = \emptyset$
proof: if $(c, y) \in \mathcal{P}_X \cap \mathcal{P}_{X'}$, $y = X + cs$ and $y = X' + cs$ so $X = X'$.
(2) $\bigcup_{X \in \mathbb{Z}} \mathcal{P}_X = [0, B-1] \times [\Phi, A-1]$
proof: for any pair $(c, y) \in [0, B-1] \times [\Phi, A-1]$, $(c, y) \in \mathcal{P}_{y-cs}$.
(3) if $X < \Phi - s(B-1)$, $\mathcal{P}_X$ is empty
proof: if $X < \Phi - s(B-1)$, $\forall c \in [0, B-1] \ \ X + cs < \Phi$.
(4) if $X \geq A$, $\mathcal{P}_X$ is empty
proof: if $X \geq A$, $\forall c \in [0, B-1] \ \ X + cs \geq A$.
(5) $\text{card}(\mathcal{P}_X) = B \Leftrightarrow X \geq \Phi$ and $X < A - s(B-1)$
proof: obvious since $X + cs \in [\Phi, A-1]$ for all $c \in [0, B-1]$.
(6) $\forall X \in \mathbb{Z} \quad \forall (c, y) \in \mathcal{P}_X \quad \forall (c', y') \in \mathcal{P}_X \quad f(g^y/I^c) = f(g^{y'}/I^{c'})$
proof: this is obvious since $g^y/I^c = g^{y'}/I^{c'}$.
(7) for any $X \in \mathbb{Z}$, there is at most one pair $(c, y) \in \mathcal{P}_X$ such that $f(g^y/I^c) = c$
proof: this is an immediate consequence of property (6) because all the pairs (c, y) in a set $\mathcal{P}_X$ have different values of c.
(8) for any X such that $\text{card}(\mathcal{P}_X) = B$, there is exactly one pair $(c, y) \in \mathcal{P}_X$ such that $f(g^y/I^c) = c$
proof: a pair (c, y) such that $f(g^y/I^c) = c$ is given by $(f(g^X), X + f(g^X) \times s)$. Uniqueness follows from property (7).

Consequently, the total number N of solutions of the equation $c = f(g^y/I^c)$ is upper bounded by the number of non-empty sets $\mathcal{P}_X$, this is a consequence of property (7), and lower bounded by the number of sets $\mathcal{P}_X$ with exactly B pairs, this is a consequence of property (8). Using properties (3), (4) and (5), we obtain that N lies between $A - 2\Phi$ and A in the following way:

$$A - 2\Phi \leq A - \Phi - (B-1)s \leq N \leq A - \Phi + (B-1)s \leq A$$

B Proof of lemma 4

Assume that a prover A_2 running on random tape ω_A, is accepted with probability $\varepsilon = 1/B^\ell + \varepsilon'$ for a public key I. We write $Succ(\omega_A, c_1, ...c_\ell) \in \{true, false\}$ the result (successful of not) of the identification of $A_2(\omega_A)$ when successive challenges $c_1, ...c_\ell$ are used.

$$\Pr_{\omega_A, c_1, ...c_\ell} [Succ(\omega_A, c_1, ...c_\ell)] = \varepsilon = 1/B^\ell + \varepsilon'$$

We consider the following algorithm (inspired from [43]):

step 1. Pick a random tape ω_A and a tuple c of ℓ integers $c_1, ...c_\ell$ in $[0, B-1]$ until $Succ(\omega_A, c)$. Let u be the number of probes.

step 2. Probe up to u random ℓ-tuples c' different from c until $Succ(\omega_A, c')$. If after the u probes a successful c' is not found, the algorithm fails.

step 3. Let j be the first index such that $c_j \neq c_j{}'$; we note y_j and $y_j{}'$ the related correct answers of A_2. If $c_j > c_j{}'$, the algorithm outputs $\sigma = y_j - y_j{}'$ and $\tau = c_j - c_j{}'$ and otherwise it outputs $\sigma = y_j{}' - y_j$ and $\tau = c_j{}' - c_j$.

If this algorithm does not fail, the prover is able to correctly answer two challenges c_j and $c_j{}'$ given the same commitment x_j, with the answers y_j and $y_j{}'$. This means that $g^{y_j}/I^{c_j} = x = g^{y_j{}'}/I^{c_j{}'}$ so $g^\sigma = I^\tau$. Furthermore, $\sigma \in [-(A + \Phi - 1), A + \Phi - 1]$ and $\tau \in [1, B-1]$.

We now analyze the complexity of the algorithm. By assumption, the probability of success of A_2 is ε, so the first step finds ω_A and c with this probability. The expected number E of repetitions is $1/\varepsilon$ and the number u of probes is equal to N with probability $\varepsilon \times (1-\varepsilon)^{N-1}$.

Let Ω be the set of random tapes ω_A such that $\Pr_c[Succ(\omega_A, c)] \geq \varepsilon - \varepsilon'/2 = 1/B^\ell + \varepsilon'/2$. The probability for the random tape ω_A found in step 1 to be in Ω, conditioned by the knowledge that $Succ(\omega_A, c) = true$, can be lower bounded in the following way:

$$\Pr_{\omega_A,c}[\omega_A \in \Omega | Succ(\omega_A, c)] = 1 - \Pr_{\omega_A,c}[\omega_A \notin \Omega | Succ(\omega_A, c)]$$
$$= 1 - \Pr_{\omega_A,c}[Succ(\omega_A, c) | \omega_A \notin \Omega] \times \frac{\Pr_{\omega_A,c}[\omega_A \notin \Omega]}{\Pr_{\omega_A,c}[Succ(\omega_A, c)]} \geq 1 - \left(\frac{1}{B^\ell} + \frac{\varepsilon'}{2}\right) \times \frac{1}{\varepsilon} = \frac{\varepsilon'}{2 \times \varepsilon}$$

Thus, with probability $> \varepsilon'/(2\varepsilon)$, the random tape ω_A is in Ω and, in this case, by definition of the set Ω, the conditional probability for a tuple of challenges $c' \neq c$ to lead to success is $\geq \varepsilon'/2$. The probability to obtain such a tuple c' after less than N probes is $\geq 1 - (1 - \varepsilon'/2)^N$.

Therefore, the probability to obtain a random tape ω_A in Ω and to find an appropriate c' is greater than

$$\frac{\varepsilon'}{2\varepsilon} \times \sum_{N=1}^{+\infty} \varepsilon \times (1-\varepsilon)^{N-1} \times \left[1 - \left(1 - \frac{\varepsilon'}{2}\right)^N\right]$$
$$= \frac{\varepsilon'}{2}\left(\sum_{N=0}^{+\infty}(1-\varepsilon)^N - \left(1 - \frac{\varepsilon'}{2}\right)\sum_{N=0}^{+\infty}\left[(1-\varepsilon)\left(1 - \frac{\varepsilon'}{2}\right)\right]^N\right)$$
$$= \frac{\varepsilon'}{2}\left(\frac{1}{\varepsilon} - \frac{1 - \frac{\varepsilon'}{2}}{\varepsilon + \frac{\varepsilon'}{2} - \varepsilon \times \frac{\varepsilon'}{2}}\right) = \frac{\varepsilon'^2}{4\varepsilon^2} \times \frac{1}{1 + \frac{\varepsilon'}{2\varepsilon} - \frac{\varepsilon'}{2}} = \frac{\varepsilon'^2}{4\varepsilon^2} \times \frac{2}{3 - \left(\frac{1}{B^\ell \times \varepsilon} + \varepsilon'\right)}$$

Since $\varepsilon > 1/B^\ell$ and $1 > \varepsilon' > 0$, we obtain $0 < 1/(B^\ell \times \varepsilon) + \varepsilon' < 2$ so $\frac{2}{3 - \left(\frac{1}{B^\ell \times \varepsilon} + \varepsilon'\right)} > \frac{2}{3}$.

In conclusion, the algorithm finds (σ, τ) with probability $> {\varepsilon'}^2/(6\varepsilon^2)$ and the total expected number of executions of the protocol between the prover and a verifier is smaller than $2/\varepsilon$. $\square$