# Security Analysis of a Practical "on the fly" Authentication and Signature Generation

Guillaume Poupard and Jacques Stern

École Normale Supérieure, Laboratoire d'informatique
45 rue d'Ulm, F-75230 Paris Cedex 05, France
email: {Guillaume.Poupard,Jacques.Stern}@ens.fr

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

We develop complete proofs of completeness, soundness and statistical zero-knowledge property of the identification scheme. The security analysis of the signature scheme leads to present a novel number theoretical lemma of independent interest and an original use of the "forking lemma" technique. From a practical point of view, the possible choice of parameters is discussed and we submit performances of an actual implementation on a cheap smart card. As an example, a complete and secure authentication can be performed in less than 20 ms with low cost equipment.

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

## 1 Introduction

Besides confidentiality, cryptographers face two important problems: authentication and signature or, in plain words, how to prove one's identity and how to digitally sign a document. Recently, several proposals have already addressed those two questions, putting forward elegant solutions, many of them based on the concept of zero-knowledge introduced in 1985 by Goldwasser, Micali and Rackoff [11]. In order to assess the performances of those schemes, four properties have to be considered:

- The most important concern is, of course, security. Basically, a system is 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 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 [4] or the computation of discrete logarithms in a finite group [23]. Half way between heuristic validation and formal proofs are proofs in a model where concrete objects are replaced by some ideal counterparts: applying this paradigm to hash functions yields the random oracle model described by Bellare and Rogaway in [1]. Although this approach may not be considered as offering absolute proofs of security for specific schemes, it provides a strong guarantee about their general design.

- Next, the size of the data involved in the scheme is of crucial practical significance. We usually need short public and private keys, we want to reduce the amount of transmissions and, lastly, the length of the generated signatures is an important parameter in many applications.
- Another key property is the time complexity of the scheme which controls the cost of the devices on which it may be implemented. There, we have to distinguish between the time needed for the precomputations that can be performed off-line possibly by a trusted authority (use & throw coupons [16]) and the calculations that have to be done on-line during authentication or signature. The latter is often the bottleneck of many applications, especially when smart cards are used.
- Finally, in situations where a directory is not available, public keys need to be certified. A few paradigms are known; the simplest consists in having some authority sign the key in order to produce a certificate. This often degrades drastically the performances of a scheme, especially in terms of size. A simpler and more efficient technique has been proposed by Shamir [19] and just consists in using public keys so closely related to the identity that they do not have to be further certified.

In this paper, we study an interactive zero-knowledge identification scheme and a derived signature scheme that combine provable security based on the general discrete logarithm problem (which difficulty is equivalent to the one of factoring integers and computing discrete logarithms modulo prime numbers), identity based short keys, very short transmissions and signature size and minimal on-line computation. This provides a solution for applications which require efficient and secure identification or signature generation while using only low cost individual devices. A typical example is "on the fly" authentication at a toll where the time needed to transmit data and to perform on-line calculations is very short, about 0.1 seconds.

## Related Work

In 1989, C. Schnorr [18] proposed an identification and a signature scheme based on the problem of computing discrete logarithms in groups of prime order. In these schemes the size of the data is short and the computations load is quite acceptable. Moreover, they are provably secure in the random oracle model [18, 17, 21]. Towards a more precise description, let $p$ be a prime number, $q$ a large prime divisor of $p-1$ and $g$ an element of $\mathbb{Z}_p^*$ of order $q$. The user receives a secret key $s$ in $\mathbb{Z}_q^*$ and the corresponding public key $v = g^{-s} \bmod p$. If he wants to be identified, he generates $x \in \mathbb{Z}_q^*$ at random and sends the commitment $t = g^x \bmod p$ to the verifier who answers a challenge $c$ randomly chosen in $[0, k[$. Then the prover computes $y = x + sc \bmod q$ and sends $y$ to the verifier who check the equation $t = g^y v^c \bmod p$.

Modifications of the Schnorr scheme that achieve additional properties have already been proposed [2, 7]. Basically, they use a composite number in place of $p$ to provide identity-based public keys. A way to improve the 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 [16] setting. Therefore, in order to further reduce the on-line computation to a very simple operation, it is tempting to eliminate the second modulus $q$ by performing operations in $\mathbb{Z}$. This has been proposed by Girault in [8] as an example of protocol allowing "self-certified" public keys.

### Our Results

In this paper we show that, using the above scheme, it is possible to achieve a combination of the strongest properties that one can demand. In section 2 we study the identification scheme and we prove that it is secure against active adversaries for any modulus $n$, provided the computation of discrete logarithms modulo $n$ is hard. Next, we prove that, with suitable parameters, the protocol is statistically zero-knowledge. In section 3, we introduce a derived signature scheme and we use the random oracle model in order to validate the proposed design, showing that if an adversary is able to forge a signature under an adaptively chosen message attack then he is able to compute discrete logarithms modulo $n$. Finally, section 4 is more practical on character: we discuss how to choose secure parameters in order to resist to the most efficient known attacks against factorization and discrete logarithm (4.1); we show that the schemes support identity based public keys (4.2) and explain how to optimize the data size (4.3), and finally we give the performances of two smart cards applications.

It should be clear that we have not invented a new scheme. Rather, we have given a thorough theoretical and practical treatment of a scheme that had not received strong attention from the crypto community, presumably because it lacked this type of analysis and, accordingly, might have been considered as dangerous despite its potential advantages. This is consistent with a current trend of today's cryptography and other examples appear in the recent literature [20].

## 2 Identification Scheme

Let us first introduce some notation. For any integer $x$, $|x|$ is the number of bits ($\lfloor \log_2(x) \rfloor + 1$) of $x$. We use functions $\boldsymbol{\delta}$, defined by $\boldsymbol{\delta}\,(\mathit{true}) = 1$ and

$\delta(false) = 0$, and $\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}(n)$ is the highest order of any element of $\mathbb{Z}_n^*$. It is well known that if the factorization of an odd integer $n$ is $\prod_i {p_i}^{e_i}$ then $\boldsymbol{\lambda}(n) = \mathrm{lcm}_i\left({p_i}^{e_i-1}(p_i-1)\right)$. Our computing model is the probabilistic polynomial time Turing machine (PPTM($|n|$)), which running time is a polynomial in $|n|$. Finally, we recall a well-known (see [17]) probabilistic lemma:

**Lemma 1.** *Let $A \subset X \times Y$, such that $\Pr_{x,y}\{A(x,y)\} \geq \varepsilon$,*

*and $\Omega = \left\{a \in X / \Pr_{y}\{A(a,y)\} \geq \varepsilon/2\right\}$ then $\Pr_{x}\{x \in \Omega\} \geq \varepsilon/2$.*

We now describe the parameters of the system. Let $n$, $S$, $X$ and $k$ be four integers and $g$ be an element of $\mathbb{Z}_n^*$ of order $\boldsymbol{\lambda}(n)$. The relations between those parameters are analyzed in the next section. We let $\Phi := (k-1)(S-1)$.

The private keys $s$ are chosen in $[0, S[$ and the public keys $v$ are computed by the relation $v = g^{-s} \bmod n$. A round of identification consists for the prover in randomly choosing an integer $x$ in $[0, X[$ and computing the *commitment* $t = g^x \bmod n$. Then he sends $t$ to the prover who answers a *challenge* $c$ randomly chosen in $[0, k[$. The prover computes $y = x + sc$ and sends it to the verifier who checks $t = g^y v^c \bmod n$. A complete identification consists in repeating $\ell$ times the elementary round.

## Security Analysis

In order to prove the security of this protocol against active adversaries, we follow the approach of Feige, Fiat and Shamir [5], proving completeness, soundness and the zero-knowledge property. Later on we consider that the security parameter is $|n|$ and that $S$, $X$ and $\ell$ are functions of $|n|$. For technical reasons related to the proof of the zero-knowledge property, $k$ is considered as a constant. In order to simplify the notations, we do not write the dependences on $|n|$ but when we say that an expression $f$ is negligible, this means that $f$ depends on $|n|$ and that, for any constant $c$ and for large enough $|n|$, $f(|n|) < 1/|n|^c$.

**Theorem 2 (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.* Just notice that $g^y v^c = g^{x+sc} g^{-sc} = g^x = t \bmod n$. □

The proof of soundness consists in proving that, if someone is correctly identified then, with overwhelming probability, he must know the secret key associated with his public key. We need two lemmas.

**Lemma 3.** *Let $n$ be any integer and $L$ be any multiple of $\boldsymbol{\lambda}(n)$. Then there exists a* PPTM($|n|$) *which, on input $(n, L)$, output the factorization of $n$ in time $O(|n||L|)$.*

*Proof.* This lemma is due to Miller and is proved in [15]. □

**Lemma 4.** *Assume that some* PPTM($|n|$) *adversary* $\widetilde{P}$ *is accepted by honest verifiers with probability* $\geq 1/k^{\ell} + \varepsilon$, $\varepsilon > 0$. *Then there exists a probabilistic Turing machine which outputs the private key $s$ from the public data with overwhelming probability in time* $O(|n|k\ell\tau/\varepsilon + |n|^{O(1)})$, *where $\tau$ is the average running time of a round of identification.*

*Proof.* Assume that some PPTM($|n|$) adversary $\widetilde{P}(\omega)$, running on random tape $\omega$, is accepted with probability $\geq 1/k^{\ell} + \varepsilon$. We write $ACC(\widetilde{P}(\omega), c_1, ...c_{\ell})$ the result of the identification of $\widetilde{P}(\omega)$ when the challenges $c_1, ...c_{\ell}$ are used. By assumption, the probability of success of $\widetilde{P}$ is $\Pr_{\omega, c_1, ...c_{\ell}} \{ACC(\widetilde{P}(\omega), c_1, ...c_{\ell})\} \geq 1/k^{\ell} + \varepsilon$. Let $\Omega$ be the set of random tapes $\omega$ such that $\Pr_{c_1, ...c_{\ell}} \{ACC(\widetilde{P}(\omega), c_1, ...c_{\ell})\} \geq 1/k^{\ell} + \varepsilon/2$. Using lemma 1, we obtain that $\Pr_{\omega} \{\omega \in \Omega\} \geq \varepsilon/2$.

We now explain how to use $\widetilde{P}$ in order to obtain a machine which on input $(n, g, v)$ answers $(\alpha, \beta)$ such that $v^{\alpha} = g^{\beta} \bmod n$, with $-k < \alpha < k$, in time $O(|n|k\ell\tau/\varepsilon)$ where $\tau$ is the average running time of a round of identification.

We first choose a random tape $\omega$. Then, let $i$ vary from 1 to $\ell$. For each $i$, we let $\widetilde{P}(\omega)$ produce the $i^{\text{th}}$ commitment $t_i$ and note $S_i$ the state reached by $\widetilde{P}(\omega)$. We ask $\widetilde{P}(\omega)$ the $k$ possible challenges and, each time, we check the answer and we reset $\widetilde{P}(\omega)$ at state $S_i$. After those $k$ steps, three cases may appear:

- if $\widetilde{P}(\omega)$ has correctly answered two challenges $c$ and $c'$, with $y$ and $y'$, return $(\alpha = c' - c, \beta = y - y')$.

- if $\widetilde{P}(\omega)$ cannot answer any challenge, return `Fail`.

- if $\widetilde{P}(\omega)$ answers exactly one challenge, keep on with the loop. If the end of the loop is reached, return `Fail`.

With probability $\geq \varepsilon/2$, $\omega \in \Omega$ and therefore this machine returns $(\alpha, \beta)$ such that $v^{\alpha} = g^{\beta} \bmod n$ after at most $O(k\ell\tau)$ time units. If we repeat $2|n|/\varepsilon$ times this procedure with other random tapes, $(\alpha, \beta)$ is obtained with overwhelming probability in time $O(|n|k\ell\tau/\varepsilon)$.

We first use the machine with the input $(n, g, v_0)$, where $v_0 = g^{\gamma} \bmod n$ for $\gamma$ chosen much greater than $n$. It returns $(\alpha, \beta)$ such that ${v_0}^{\alpha} = g^{\beta} \bmod n$ so $L = \gamma\alpha - \beta$ is a multiple of $\boldsymbol{\lambda}(n)$ and $L \neq 0$ with high probability because even if the machine is able to compute discrete logarithms modulo $n$ in base $g$, it only learns the value of $\gamma$ modulo $\boldsymbol{\lambda}(n)$. Using the result of lemma 3, $n$ is factored and accordingly $\boldsymbol{\lambda}(n)$ is computed in polynomial time.

We then use the machine again with the public key $v$ whose discrete logarithm is unknown; it answers $(c, y)$ such that $v^c = g^y \bmod n$ and, solving the equation $y + sc = 0 \bmod \boldsymbol{\lambda}(n)$, we obtain $s_0 = s \bmod \lambda_0$ with $\lambda_0 = \boldsymbol{\lambda}(n)/\gcd(c, \boldsymbol{\lambda}(n))$. Since $k$ is assumed to be a constant and $-k < c < k$, we can test whether $v = g^{-s_0 - i \times \lambda_0}$ for all the positive integers $i$ less than $\gcd(c, \boldsymbol{\lambda}(n)) < k$ and obtain the secret $s$. □

**Theorem 5 (Soundness).** *Assume that some* PPTM($|n|$) *adversary* $\widetilde{P}$ *is accepted with non-negligible probability by honest verifiers and that* $\log(|n|) = o(\ell)$

*and that $\ell$ is polynomial in $|n|$. Then there exists a* $\mathrm{PPTM}(|n|)$ *which outputs $s$ from public data with overwhelming probability.*

*Proof.* If $\pi(|n|)$ is the non-negligible probability of success of $\widetilde{P}$, there exists an integer $d$ such that $\pi(|n|) \geq 1/|n|^d$ for infinitely many values $|n|$. Furthermore, for $|n|$ large enough, $1/k^\ell < 1/2|n|^d$ because $\log(|n|) = o(\ell)$. So, taking $\varepsilon = \pi(|n|)/2$ in lemma 4 we conclude that it is possible to compute $s$ in time $O(|n|k\ell\tau/\varepsilon + |n|^{O(1)})$. If we assume that $\ell$ is polynomial in $|n|$, then there exists a $\mathrm{PPTM}(|n|)$ which outputs $s$ from public data with overwhelming probability. □

**Theorem 6 (Zero-knowledge).** *The protocol is statistically zero-knowledge if $ST/X$ is negligible, where $T(|n|)$ is the maximal number of repetitions of the protocol with the same keys.*

*Proof.* We describe the polynomial time simulation of the communication between a prover $P$ and a dishonest verifier $\widetilde{V}$. We assume that, in order to try to obtain information about $s$, $\widetilde{V}$ does not randomly choose the challenges. If we focus on the $i^{\text{th}}$ round of identification, $\widetilde{V}$ has already obtained data, noted $Data_i$, from previous interactions with $P$. Then the prover sends the commitment $t_i$ and $\widetilde{V}$ chooses, possibly using $Data_i$ and $t_i$, the challenge $c_i(Data_i, t_i)$.

Here is a simulation of the $i^{\text{th}}$ round of identification: choose random values $c_i{}' \in [0, k[$ and $y_i{}' \in [\Phi, X[= [(k-1)(S-1), X[$, compute $t_i{}' = g^{y_i{}'} v^{c_i{}'} \bmod n$. If $c_i(D_i, t_i{}') \neq c_i{}'$ then try again with another pair $(c_i, y_i{}')$, else return $(t_i{}', c_i{}', y_i{}')$.

For any function $Q$ from $\mathbb{Z}_n$ to $\mathbb{Z}_k$, any integer $A$ and any positive constant $\Delta$, we define $\mathcal{S}(Q, A, \Delta)$ as the number of pairs $(c, y)$ in $[0, k[\times[A, A + \Delta[$ such that $Q(g^y v^c) = c$. Let us prove that $\mathcal{S}$ verifies $\Delta - \Phi \leq \mathcal{S}(Q, A, \Delta) \leq \Delta + \Phi$ (remember $\Phi = (k-1)(S-1)$). First notice that $g^y v^c = g^{y+d\times s} v^{c+d} \bmod n$ and that if $Q(g^y v^c) = c$ then $Q(g^{y+d\times s} v^{c+d}) = c \neq c + d$. The inequalities follow from a partition of the pairs $(c, y)$ into $\Delta - s(k-1)$ subsets containing exactly one pair $(c, y)$ such that $Q(g^y v^c) = c$ and $2s(k-1)$ subsets that have at most one such pair.

Let us denote by $p_i(t, c, y)$ the probability to obtain the triplet $(t, c, y)$ in the actual round and by $p_i{}'(t, c, y)$ the probability to obtain the same triplet during the simulation. Using the strategy $c_i(Data_i, t) = Q(t)$ of $\widetilde{V}$ to choose the challenges, we first calculate $p$:

$$p(\alpha, \beta, \gamma) = \sum_{0 \leq x < X} \Pr \left\{ \begin{array}{l} g^x = \alpha \bmod n \\ Q(g^x) = \beta \\ x + sQ(g^x) = \gamma \end{array} \right\} = \frac{1}{X} \delta \left( \begin{array}{c} 0 \leq \gamma - s\beta < X \\ Q(\alpha) = \beta \\ \alpha = g^\gamma v^\beta \bmod n \end{array} \right)$$

If $(\alpha, \beta, \gamma)$ are such that $\Phi \leq \gamma < X$, $Q(\alpha) = \beta$ and $\alpha = g^\gamma v^\beta \bmod n$, we have $p(\alpha, \beta, \gamma) = 1/X$. Consequently, the sum of the probabilities for $\alpha \in [\Phi, X[$ and all $\beta$ and $\gamma$ is equal to $\mathcal{S}(Q, \Phi, X - \Phi)/X$ by definition of $\mathcal{S}$.

We can also calculate the conditional probability $p'$, using the Bayes' law:

$$\begin{aligned}
p'(\alpha,\beta,\gamma) &= \Pr_{0\le c<k,\varPhi\le y<X}\left\{ c=\beta, y=\gamma, \alpha=g^{\gamma}v^{\beta} \bmod n \,\middle|\, Q(\alpha)=\beta\right\} \\
&= \Pr_{\substack{0\le c<k\\ \varPhi\le y<X}}\left\{\begin{matrix} c=\beta, y=\gamma, Q(\alpha)=\beta, \\ \alpha=g^{\gamma}v^{\beta} \bmod n\end{matrix}\right\} \Big/ \Pr_{\substack{0\le c<k\\ \varPhi\le y<X}}\{Q(g^{y}v^{c})=c\} \\
&= \frac{\delta\left(\varPhi\le\gamma<X, Q(\alpha)=\beta, \alpha=g^{\gamma}v^{\beta} \bmod n\right)}{k(X-\varPhi)} \Big/ \frac{\mathcal{S}(Q,\varPhi,X-\varPhi)}{k(X-\varPhi)}
\end{aligned}$$

The inequalities $X-2\varPhi \le \mathcal{S}(Q,\varPhi,X-\varPhi) \le X$ lead to an upper bound of the sum of differences $\varSigma_0 = \sum_{\alpha,\beta,\gamma} |p_i(\alpha,\beta,\gamma) - p_i'(\alpha,\beta,\gamma)|$ because

$$\begin{aligned}
\varSigma_0 &= \sum_{\substack{\varPhi\le\gamma<X\\ 0\le\beta<k\\ \alpha=g^{\gamma}v^{\beta} \bmod n}} \left|\frac{1}{X} - \frac{1}{\mathcal{S}(Q,\varPhi,X-\varPhi)}\right| \delta(Q(\alpha)=\beta) + \sum_{\gamma\notin[\varPhi,X[,\beta,\gamma} p(\alpha,\beta,\gamma) \\
&= \frac{X-\mathcal{S}(Q,\varPhi,X-\varPhi)}{X.\mathcal{S}(Q,\varPhi,X-\varPhi)} \times \sum_{\substack{\varPhi\le\gamma<X\\ 0\le\beta<k\\ \alpha=g^{\gamma}v^{\beta} \bmod n}} \delta(Q(\alpha)=\beta) + \left(1 - \frac{\mathcal{S}(Q,\varPhi,X-\varPhi)}{X}\right)
\end{aligned}$$

Therefore $\varSigma_0 = 2(1-\mathcal{S}(Q,\varPhi,X-\varPhi)/X)$ and consequently $\varSigma_0$ is bounded by $4\varPhi/X < 4kS/X$. So the simulation of one round is statistically indistinguishable from the actual distribution if $kS/X$ is negligible.

We then use recursively this reasoning to simulate all the $T$ rounds of identification between $P$ and $\tilde{V}$ and we obtain

$$\sum_{(\alpha_i,\beta_i,\gamma_i),i\le T} \left|\Pr\{(\alpha_i,\beta_i,\gamma_i)=(t_i,c_i,y_i)\} - \Pr\{(\alpha_i,\beta_i,\gamma_i)=(t_i',c_i',y_i')\}\right| < \tfrac{4kST}{X}$$

This is rigorously proved in appendix A. Therefore the protocol is statistically zero-knowledge as soon as $ST/X$ is negligible. □

## 3 Signature Scheme

We can turn the identification scheme into a signature scheme in a way initially proposed by Fiat and Shamir [6] and used by Schnorr [18] and others. In order to perform the transformation, the challenges $c$ are no longer randomly chosen by a verifier but computed through a hash function $h$. The signature of a message $m$ is computed by taking a random $x$ in $[0,X[$ and computing $t=g^{x} \bmod n$, $c=h(m,t)$ and $y=x+sc$. This produces the signature $(t,c,y)$ that may be checked by anybody with the equations $c=h(m,t)$ and $t=g^{y}v^{c} \bmod n$.

We now note $[0,k[$ the output range of $h$. Note that, in the identification scheme, $k$ was a fixed constant but, in the signature setting, we need to let $k$ depend on the security parameter $|n|$, like $S$, $X$ and $T$.

## Security Analysis

In order to prove the security of this protocol, we show that, if someone is able to forge valid signatures after having obtained signatures of messages he chose, then we can use him to compute the secret $s$. The random oracle model [1] is used to model the behavior of the hash function $h$ so the proofs validate the overall design. We first need a lemma, proving that a PPTM($|n|$) can statistically simulate actual signature generation. We also use a number theoretical result which is of independent interest and which allows a precise estimate of the greatest common divisor of two random numbers.

**Lemma 7.** *The outputs of the signature algorithm can be statistically simulated by a* PPTM($|n|$) *if $kS/X$ is negligible.*

*Proof.* Valid triplets $(t, c, y)$, i.e. verifying $t = g^y v^c \bmod n$, generated by the signature algorithm in the random oracle model can be simulated with the following probabilistic algorithm: choose randomly $c' \in [0, k[$ and $y' \in [\Phi, X[$, compute $t' = g^{y'} v^{c'} \bmod n$ and return $(t', c', y')$ as a valid signature.

If we note $p(t, c, y)$ the probability to obtain $(t, c, y)$ with the signature algorithm and $p'(t, c, y)$ with the simulator, we see that

$$p(\alpha,\beta,\gamma) = \frac{\delta\begin{pmatrix} g^\gamma v^\beta = \alpha \bmod n \\ 0 \le \beta < k \\ 0 \le \gamma - s\beta < X \end{pmatrix}}{kX} \quad \text{and} \quad p'(\alpha,\beta,\gamma) = \frac{\delta\begin{pmatrix} g^\gamma v^\beta = \alpha \bmod n \\ 0 \le \beta < k \\ \Phi \le \gamma < X \end{pmatrix}}{k(X - \Phi)}$$

Therefore the sum of all the differences $|p(\alpha, \beta, \gamma) - p'(\alpha, \beta, \gamma)|$ can be bounded like in the proof of theorem 6 and is less than $2kS/X$. □

**Lemma 8.** *Let $c$ be a fixed constant in $[0, k[$. Given any positive integer $B$,*

$$\Pr_{0 \le c', c'' < k} \{\gcd(c - c', c - c'') > B\} < \frac{2.7}{B}$$

*Proof.* appears in appendix B. □

An attacker who existentially forges the signature scheme can be modeled as a PPTM($|n|$) $\mathcal{A}(\omega)$, running on random tape $\omega$, which is able, for infinitly many values of the security parameter $|n|$ and for a non-negligible fraction of the public keys, to find with probability $\varepsilon(|n|)$ a message $m$ and a valid signature $(t, c, y)$.

Two scenario of attacks are considered, the *no-message attack* during which $\mathcal{A}(\omega)$ can ask $Q$ queries $Q_1 ... Q_Q$ to a random oracle $f$ and the *adaptively chosen message attack* where $\mathcal{A}(\omega)$ can also ask the signatures $(t_i, c_i, y_i)$ of $R$ messages $m_1 ... m_R$ he chooses to a signature oracle. This oracle knows the private key associated with the public one, follows the signature algorithm and consequently uses $f$ to compute the challenges. We note $T$ the maximal number of messages signed with a fixed key and $\tau_{\mathcal{A}}$ the average running time of an attacker $\mathcal{A}$.

**Theorem 9.** *If an existential forgery of the signature scheme under a no-message attack has a probability of success $\varepsilon \geq 8Q/k$ then the discrete logarithm can be computed with probability $\geq \varepsilon^4/(2^{10} \times Q^5)$ within time $3 \times \tau_{\mathcal{A}} + O(Q^2/\varepsilon^2)$.*

*Proof.* The proof is based on a technique developed by Pointcheval and Stern [17] and known as the *forking lemma.* It consists in making the attacker run with different random oracles in order to obtain valid signatures $(t, c_i, y_i)$ of a message $m$ with the same commitment $t$ and consequently equations of the form $g^{y_i}v^{c_i} = g^{y_j}v^{c_j} \bmod n$. Then, like in the proof of soundness of the identification scheme, this leads to the computation of the discrete logarithm of the public key.

We describe a PPTM($|n|$) $\mathcal{M}$ which obtains the valid signatures. It is far from being the most efficient one but it is the easiest to analyse. $\mathcal{M}$ first chooses an index $\beta$ for which we assume that the question $Q_\beta$ asked by the attacker to the random oracle will be $(m, t)$. Then it chooses a random tape $\omega$ and three random oracles which answer identiquely the $\beta-1$ first queries. Finally $\mathcal{M}$ makes the attacker $\mathcal{A}(\omega)$ running with each oracle and we hope to obtain three valid signatures $(t, c, y)$, $(t, c', y')$ and $(t, c'', y'')$ of the same message $m$. The following analysis proves that $\mathcal{M}$ succeeds with probability $\geq \varepsilon^4/(2^{10} \times Q^5)$ and further that $0 < gcd(c - c', c - c'') \leq 173 \times Q^2/\varepsilon^2$.

Then, in order to compute a discrete logarithm modulo $n$, we first use $\mathcal{M}$ with a public key $v_0$ which discrete logarithm is known and this leads to the value of $\lambda(n)$, just like in the proof of soundness of the identification scheme. Then we use it again with the public key $v$ which discrete logarithm $(-s)$ is searched and we obtain $s_0 = s \bmod \lambda(n)/\gcd(c - c', c - c'', \lambda(n))$. This explains why we need three valid signatures instead of two in the original forking lemma: in the identification scheme, $k$ was fixed so $\gcd(c - c', \lambda(n)) < k$ was always "small"; here, $k$ depends on $|n|$ and is "very large" so we need a second "forking" to be sure, thanks to the result of the lemma 8, to obtain a "small" gcd whatever $\lambda(n)$ may be and therefore to be able to find $s \bmod \lambda(n)$ after an exhaustive search.

We now analyse why $\mathcal{M}$ succeeds in finding valid signatures. First, we can assume that $\mathcal{A}(\omega)$ does not ask the random oracle twice the same query so the oracle can be replaced by a the list of its Q random answers $\rho_1, ...\rho_Q$. Then, because of the randomness of the random oracle, the probability to produce a valid signature $(t, c, y)$ of $m$ without asking $(m, t)$ is less than $1/k$. Since $\varepsilon > 8Q/k > 2/k$, $\mathcal{A}(\omega)$ asks the query $(m, t)$ and forges a signature with probability $\geq \varepsilon/2$. Consequently, with probability $\geq 1/Q$, $\mathcal{M}$ guesses an index $\beta$ such that the attacker succeds with $Q_\beta = (m, t)$ with probability $\geq \varepsilon/2Q$. Using lemma 1 with $X = \{0,1\}^* \times ([0...k[)^{\beta-1}$ and $Y = ([0...k[)^{Q-\beta+1}$, we distinguish the subset $\Omega$ of tuples $(\omega, \rho_1, ...\rho_{\beta-1})$ such that the probability of success only taken over $(\rho_\beta, ...\rho_R)$ is $\geq \varepsilon/4Q$. $\mathcal{M}$ chooses $\omega$ and $(\rho_1, ...\rho_{\beta-1})$ that belong to $\Omega$ with probability $\geq \varepsilon/4Q$. Then, with probability $\geq \varepsilon/4Q$ the last answers of the first oracle leads $\mathcal{A}(\omega)$ to output a valid signature.

The probability of success with the two other random oracles is $\geq (\varepsilon/4Q)^2$. Furthermore $gcd(\rho_\beta - \rho'_\beta, \rho_\beta - \rho''_\beta) = 0$ iif $\rho_\beta = \rho'_\beta = \rho''_\beta$ so the probability to have $gcd(\rho_\beta - \rho'_\beta, \rho_\beta - \rho''_\beta) = 0$ is $1/k^2$ and it is smaller than $1/4 \times (\varepsilon/4Q)^2$. Lemma 8 proves that the probability to have $gcd(\rho_\beta - \rho'_\beta, \rho_\beta - \rho''_\beta) > B$ is less than $2.7/B$

so, with $B = 173 \times Q^2/\varepsilon^2$, this probability is also smaller than $1/4 \times (\varepsilon/4Q)^2$. In conclusion, $\mathcal{M}$ succeeds with probability $\geq 1/Q \times \varepsilon/4Q \times \varepsilon/4Q \times 1/4(\varepsilon/4Q)^2$ and $(c, c', c'') = (\rho_\beta, \rho'_\beta, \rho''_\beta)$ are such that $0 < gcd(c - c', c - c'') \leq 173 \times Q^2/\varepsilon^2$. □

**Theorem 10.** *Assume that $kST/X$ and $1/k$ are negligible. If an existential forgery of the signature scheme under adaptatively chosen message attack has a non-negligible probability of success then the discrete logarithm can be computed in time polynomial in $|n|$.*

*Proof.* First, the signature oracle can be replaced by the PPTM($|n|$) $\mathcal{S}(\omega')$ of lemma 7 which simulates valid signatures if $kST/X$ is negligible. This modifies only negligibly the probability of succes of the attacker because overwise it would be able to distinguish between actual and simulated signatures and this would contradict lemma 7. Then, we can assume that $\mathcal{A}(\omega)$ does not ask the oracle a query whose answer is already known, i.e. already asked by $\mathcal{A}(\omega)$ or generated by the simulator. On the other hand $\mathcal{S}(\omega')$ can simulate twice the answer to the same query or one already asked by $\mathcal{A}(\omega)$. The probability to have such a "collision" is the probability to have $(m_i, t_i) = \mathcal{Q}_j$ or $(m_i, t_i) = (m_j, t_j)$. But if $x$ is fixed in $[0, X[$ and $x'$ randomly chosen in the same set, the probability to have $g^x = g^{x'} \bmod n$ is $\lceil X/\lambda(n) \rceil / X < 1/\lambda(n) + 1/X$. So the probability of collision is less than $(RQ + R(R-1)) \times (1/\lambda(n) + 1/X)$. We can further assume that $1/\lambda(n)$ and $1/X$ are negligible because overwise it would be possible to find the secret key by an exhaustive search (eventually using a known signature) in polynomial time. Consequently the probability is negligible since $Q$ and $R$ are polynomials in $|n|$ and for large enough values of the security parameter it becomes less than $\varepsilon/2$.

If no query is asked twice, the random oracle can be considered as an oracle who just answers random values chosen in $[0, k[$. With probability $\geq \varepsilon/2$, the attacker succeds in forging a signature in this context. Furthermore, $1/k$ is assumed to be negligible so, for large enough values $|n|$, $\varepsilon/2 > 8Q/k$. We can now use exactly the same proof as for the previous theorem to prove that we can make a PPTM($|n|$) able to compute discrete logarithms modulo $n$ with non-negligible probability. □

# 4 Applications

## 4.1 Choice of the Parameters

Let us first focus on $n$. In the theoretical analysis, we have not restricted the possible values and we have reduced the security to the difficulty of computing discrete logarithms modulo $n$. Notice that, if we know the factorization $\prod_i p_i^{e_i}$ of $n$, the Chinese remainder technique enables to reduce this problem to computing discrete logarithms modulo each factor $p_i^{e_i}$. So, in order to prevent $n$ from being factored, we must choose $|n|$ greater than 512 bits and $|n| = 1024$ should be more appropriate for secure applications (see [4]). Moreover, the prime factors of $n$ must have approximately the same size and must not be too numerous. In conclusion, $n$ may be a prime integer, like $p$ in the Schnorr scheme, or the product of a few distinct prime numbers in order to use identity-based public-keys (4.2).

The size of $S$ is conditioned by the complexity of discrete logarithms algorithms such as the Pollard lambda method which enables to compute $s$ in $O(\sqrt{S})$ operations. Furthermore, van Oorschot and Wiener have shown that this method can still be improved; we refer the reader to [23] for a precise analysis. We just notice that we should take $|S|$ greater than 140 bits and preferably $|S| \approx 180$.

The choice of the size of $k$ is related to the probability of success of an adversary. The expected security depends on the application and $|k| = 20$ ($\ell = 1$) would probably be large enough for many identification systems. A larger value such as $|k| = 80$ would of course be preferred for signature schemes.

When the parameters $S$ and $k$ are chosen, since $X/kS$ must be large enough to guarantee the statistical zero-knowledge property and to the security of the signature; we can for example take $|X| \geq |S| + |k| + 60$ for identification and $|X| \geq |S| + |k| + 80$ for signature schemes.

### 4.2 Identity Based Public Keys

Following ideas of Maurer and Yacobi [14], the public keys can be turned into identity-based keys. Assume that an authority knows the factorization of the publicly-known modulus $n$ and that it is able to compute discrete logarithms modulo each prime factor and consequently modulo $n$. Then, the authority computes secret keys associated with public keys closely related to the identity of users. Note that since $\mathbb{Z}_n^*$ is not cyclic, we have to add a small offset to the identity in order to obtain public keys which discrete logarithm exists.

We now recall a realistic scenario developped in [14]: an authority chooses four 200 bits prime integers $p_1, ...p_4$ such that, for all $i$, $(p_i - 1)/2$ is also prime. Then it searches generators $g_i$ of all the cyclic groups $\mathbb{Z}_{p_i}^*$ ($(p_i - 3)/2$ elements of $\mathbb{Z}_{p_i}^*$ generate the group). Using the Chinese remainder theorem, it obtains $g$ such that $g = g_i \bmod p_i$ for all $i$; $g$ is an element of maximal order of $\mathbb{Z}_{p_i}^*$. It publishes $n = \prod p_i$ and $g$. Then it computes the secret keys of each user, i.e. the discrete logarithm of a coding of each identity merged with a small offset to guarantee the existence of the logarithm. The subexponential-time index-calculus algorithm [13] is the most powerful known method to solve this problem. During an initial processing stage it computes once and once only a database that contains the discrete logarithms of a few elements and then any discrete logarithm computation reuses this database.

### 4.3 Optimization of the Data Size

In order to decrease the number of communication bits, Fiat and Shamir [6] have suggested not to send the all commitment in the first step of the identification but only a hash value. This trick can of course 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 $t$ is replaced in the protocol by $t' = h'(t)$ and the verification equation becomes $t' = h'(g^y v^c \bmod n)$.

Using the notion of $r$-collision-free hash functions, i.e. functions such that it is not possible to find $r$ pairwise distinct values with the same image, Girault and Stern [9] have analyzed precisely the consequences of such a modification on the security of identification schemes. We just recall the results of their study on the minimal value of $|h'|$: if we note $m$ the maximal number of queries to $h'$ (for example $m = 2^{64}$) in a reasonable time, then $|h'|$ must be greater than $128 \times m$ and than $(2m^r/r!)^{1/r-1}$. Furthermore, $k$ must be increased to $k' = k \times (r-1)$ in order to keep the same level of security $(1/k^\ell)$. All the details can be found in [9] and the tables below presents numerical results.

| $m = 2^{64}$ | r | 2 | 3 | 4 | **8** | $\geq 9$ |
|---|---|---|---|---|---|---|
| | $\lvert h' \rvert$ | 128 | 96 | 85 | **72** | 71 |

| $m = 2^{80}$ | r | 2 | 3 | 4 | 5 | **9** | $\geq 10$ |
|---|---|---|---|---|---|---|---|
| | $\lvert h' \rvert$ | 160 | 120 | 106 | 99 | **88** | 87 |

Furthermore, 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 $(x, h'(g^x \bmod n))$. This can still be improved if the random values $x$ are generated by a pseudo-random generator. This leads just to memorize the seed of the generator and the commitments, i.e. about only 10 bytes per authentication !

Finally, the signature $(t', c, y)$ of a message $m$ that one can verify using the equations $c = h(m, t')$ and $t' = h'(g^y v^c \bmod n)$ can be reduced to $(m, c, y)$ with the single verification equation $c = h(m, h'(g^y v^c \bmod n))$.

### 4.4 Smart Card Application

In order to show to what extent the computations are minimal and the transmissions very short, we now present two applications that we have implemented on low cost smart cards based on an Intel 6805 chip. The size of the program is very short, about **300 bytes**. We see in the following table 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 do not 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.

| Application | Identification | Signature |
|---|---|---|
| Parameters | $\lvert n \rvert = 768$ $\lvert S \rvert=144$ | $\lvert n \rvert = 1024$ $\lvert S \rvert=176$ |
| | $\lvert k \rvert=24$ $\lvert X \rvert=228$ | $\lvert k \rvert=80$ $\lvert X \rvert=336$ |
| Size of a precomputation | 72 bits | 88 bits |
| Number of precomp. in 4 KB | **455** | **372** |
| Number of CPU cycles | 4678 | 14934 |
| Running time at 3.57 MHz | **1.3 ms** | **4.2 ms** |
| Amount of communication | 327 bits | 419 bits |
| Running time at 19200 bauds | 17 ms | 21.8 ms |
| **Total running time** | **18.3 ms** | **26 ms** |

## References

1. M. Bellare and P. Rogaway. Random Oracles are Practical: a paradigm for designing efficient protocols. In *Proc. of the 1st CCCS*, 62–73. ACM press, 1993.
2. E. F. Brickell and K. S. McCurley. An Interactive Identification Scheme Based on Discrete Logarithms and Factoring. *Journal of Cryptology*, 5:29–39, 1992.
3. H. Cohen. *A Course in Computational Algebraic Number Theory.* Graduate Texts in Mathematics 138. Springer, 1993.
4. J. Cowie, B. Dodson, R.M. Elkenbracht-Huizing, A. Lenstra, P. Montgomery, and J. Zayer. A World Wide Number Field Sieve Factoring Record: On to 512 Bits. In *Asiacrypt '96*, LNCS 1163, 382–394. Springer, 1996.
5. U. Feige, A. Fiat, and A. Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, 1:77–95, 1988.
6. A. Fiat and A. Shamir. How to Prove Yourself: practical solutions of identification and signature problems. In *Crypto '86*, LNCS 263, 186–194. Springer, 1987.
7. M. Girault. An Identity-Based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. In *Eurocrypt '90*, LNCS 473, 481–486, 1991.
8. M. Girault. Self-certified public keys. In *Eurocrypt '91*, LNCS 547, 490–497. Springer, 1992.
9. M. Girault and J. Stern. On the Length of Cryptographic Hash-Values used in Identification Schemes. In *Crypto '94*, LNCS 839, 202–215. Springer, 1994.
10. O. Goldreich. *Foundations of Cryptography.* Weizmann Institute of Science, 1995. (fragment of a book).
11. S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. In *Proc. of the 17th STOC*, 291–304. ACM Press, 1985.
12. D. E. Knuth. Seminumerical algorithms. In *The Art of Computer Programming*, volume 2. Addison-Wesley Publishing Company, 1969.
13. B. A. LaMacchia and A. M. Odlyzko. Computation of Discrete Logarithms in Prime Fields. *Designs, Codes and Cryptography*, 1(1):47–62, May 1991.
14. U. M. Maurer and Y. Yacobi. Non-interactive Public-Key Cryptography. In *Eurocrypt '91*, LNCS 547, 498–507. Springer, 1992.
15. G. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, (13):300–317, 1976.
16. D. Naccache, D. M'Raïhi, S. Vaudenay, and D. Raphaeli. Can DSA be improved ? In *Eurocrypt '94*, LNCS 950, 77–85. Springer, 1995.
17. D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt '96*, LNCS 1070, 387–398. Springer, 1996.
18. C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Crypto '89*, LNCS 435, 235–251. Springer, 1990.
19. A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Crypto '84*, LNCS 196, 47–53. Springer, 1985.
20. V. Shoup. On The Security of a Practical Identification Scheme. In *Eurocrypt '96*, LNCS 1070, 344–353. Springer, 1996.
21. V. Shoup. Lower Bounds for Discrete Logarithms and Related Problems. In *Eurocrypt '97*, LNCS 1233, 256–266. Springer, 1997.
22. D. R. Stinson. *Cryptography, Theory and Practice.* CRC Press, 1995.
23. P. C. van Oorschot and M. J. Wiener. On Diffie-Hellman Key Agreement with Short Exponents. In *Eurocrypt '96*, LNCS 1070, 332–343. Springer, 1996.

# A Complement to the Proof of Theorem 6

Using the notations of the theorem 6 and the ideas of [10], we prove that

$$\sum_{(\alpha_i,\beta_i,\gamma_i),i\leq T} \left|\Pr\left\{(\alpha_i,\beta_i,\gamma_i)=(t_i,c_i,y_i)\right\}-\Pr\left\{(\alpha_i,\beta_i,\gamma_i)=(t_i{}',c_i{}',y_i{}')\right\}\right| < \frac{4kST}{X}$$

*Proof.* Let us note $w_i = (t_i, c_i, y_i)$ the communication between $P$ and $\widetilde{V}$ at the round $i$ and $w'_i$ the simulated triplet. By induction, since we have already proved the case $T = 1$, assume that the property is true for $T - 1$.

The quantity $\sum_{\omega_i,i\leq T} \left|\Pr\left\{w_i=\omega_i, i\leq T\right\}-\Pr\left\{w_i{}'=\omega_i, i\leq T\right\}\right|$

can be written $\sum_{\omega_i,i<T}\sum_{\omega_T} \left|\begin{array}{l}\Pr\left\{w_i=\omega_i,i<T\right\}\Pr\left\{w_t=\omega_T/w_i=\omega_i,i<T\right\}\\ -\Pr\left\{w_i{}'=\omega_i,i<T\right\}\Pr\left\{w_t{}'=\omega_T/w_i{}'=\omega_i,i<T\right\}\end{array}\right|$.

Using undergraduate calculus techniques we see that this is bounded by

$$\sum_{\omega_T}\left|\begin{array}{l}\Pr\left\{w_t=\omega_T/w_i=\omega_i,i<T\right\}\\ -\Pr\left\{w_t{}'=\omega_T/w_i{}'=\omega_i,i<T\right\}\end{array}\right| + \sum_{\omega_i,i<T}\left|\begin{array}{l}\Pr\left\{w_i=\omega_i,i<T\right\}\\ -\Pr\left\{w_i{}'=\omega_i,i<T\right\}\end{array}\right|$$

and using the induction assumption and the result for just one round, this ends the proof. □

# B Proof of Lemma 8

Let $c \in [0, k[$ be a fixed value and $B$ any positive integer. The expression $\Pr_{0\leq c',c''<k}$ $\{\gcd(c-c', c-c'') > B\}$ can also be written

$$\sum_{d>B}\Pr_{c-k<d',d''\leq c}\{gcd(d',d'')=d\} = \sum_{d=B+1}^{k-1}\Pr_{c-k<d',d''\leq c}\left\{\begin{array}{l}d|d', d|d'' \text{ and}\\ \gcd(d'/d,d''/d)=1\end{array}\right\}$$

$$= \sum_{d=B+1}^{k-1}\left(\Pr_{c-k<d'\leq c}\{d|d'\}\right)^2 \Pr_{(c-k)/d<e',e''\leq c/d}\{gcd(e',e'')=1\}$$

First, it is straightforward to prove that

$$\Pr_{c-k<d'\leq c}\{d|d'\} = \left(\left\lfloor\tfrac{c}{d}\right\rfloor + \left\lfloor\tfrac{-c+k-1}{d}\right\rfloor + 1\right)/k < 1/d + 1/k$$

Furthermore we have to evaluate the probability for $e'$ and $e''$ randomly chosen in $[(c-k+1)/d, c/d]$ to be coprimes integers. We use the idea of a proof of Mertens to demonstrate a theorem of Dirichlet that states $\Pr_{u,v}\{\gcd(u,v)=1\} = 6/\pi^2$ [12, pp 324,337,595]. Let $\alpha$ and $\beta$ be two integers such that $\alpha < 0$ and $\alpha+\beta > 0$ and $q(\alpha,\beta)$ be the number of pairs $(u, v)$ such that $\alpha \leq u, v \leq \alpha+\beta$ and $\gcd(u, v) = 1$.

From the principle of inclusion and exclusion, since $\left(\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor+1\right)^2-1$ is the number of pairs $(u,v)\neq(0,0)$ such that $\ell$ divides $u$ and $v$, we infer

$$\begin{aligned}
q(\alpha,\beta) &= (\beta+1)^2-1-\sum_{p_1\text{ prime}}\left[\left(\left\lfloor\frac{\alpha+\beta}{p_1}\right\rfloor+\left\lfloor\frac{-\alpha}{p_1}\right\rfloor+1\right)^2-1\right]\\
&\quad+\sum_{p_1<p_2\text{ primes}}\left[\left(\left\lfloor\frac{\alpha+\beta}{p_1p_2}\right\rfloor+\left\lfloor\frac{-\alpha}{p_1p_2}\right\rfloor+1\right)^2-1\right]-\ldots\\
&=\sum_{\ell\geq1}\mu(\ell)\left[\left(\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor+1\right)^2-1\right]
\end{aligned}$$

where $\mu(\ell)$ is the Möbius function defined by $\mu(1)=1$, $\mu(p_1,\ldots p_r)=(-1)^r$ if $p_1,\ldots p_r$ are distinct primes and else $\mu(\ell)=0$. Let $\Delta=q(\alpha,\beta)-\beta^2\sum_{\ell\geq1}\mu(\ell)/\ell^2$. If $\ell>\beta$, $0<\alpha+\beta<\beta<\ell$ and $0<-\alpha<\beta<\ell$ so $\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor=0$.

$$\Delta=\sum_{\ell=1}^{\beta}\mu(\ell)\left[\left(\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor+1\right)^2-1-\left(\frac{\beta}{\ell}\right)^2\right]-\beta^2\sum_{\ell>\beta}\mu(\ell)/\ell^2$$

Using the formula $x-1<\lfloor x\rfloor\leq x$, we get for $\ell\leq\beta$: $-1<\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor+1-\frac{\beta}{\ell}\leq1$ and $\frac{2\beta}{\ell}-1<\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor+1+\frac{\beta}{\ell}\leq\frac{2\beta}{\ell}+1$. Consequently

$$-\frac{2\beta}{\ell}-1<\left(\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor+1\right)^2-\left(\frac{\beta}{\ell}\right)^2\leq\frac{2\beta}{\ell}+1$$

$$\left|\left(\left\lfloor\frac{\alpha+\beta}{\ell}\right\rfloor+\left\lfloor\frac{-\alpha}{\ell}\right\rfloor+1\right)^2-1-\left(\frac{\beta}{\ell}\right)^2\right|\leq\frac{2\beta}{\ell}+2$$

Furthermore, the second term $\sum_{\ell>\beta}\mu(\ell)/\ell^2$ is easily estimated because $\left|\sum_{\ell>\beta}\mu(\ell)/\ell^2\right|<\sum_{\ell>\beta}1/\ell^2<\int_\beta^{+\infty}dx/x^2\leq1/\beta$. Consequently $|\Delta|<2\beta(H_\beta+1)+\beta$ with $H_\beta=\sum_{\ell=1}^{\beta}1/\ell$. Furthermore, $\sum_{\ell\geq1}\mu(\ell)/\ell^2=6/\pi^2$ (see [12, p 337]) so $q(\alpha,\beta)<6/\pi^2\times\beta^2+2\beta\times H_\beta+3\beta$. We can apply this result and we obtain

$$\Pr_{(c-k+1)/d\leq e',e''\leq c/d}\{\gcd(e',e'')=1\}<\frac{q\left(\frac{c-k+1}{d},\left\lfloor\frac{c}{d}\right\rfloor+\left\lceil\frac{c-k+1}{d}\right\rceil\right)}{\left(\left\lfloor\frac{c}{d}\right\rfloor+\left\lceil\frac{c-k+1}{d}\right\rceil+1\right)^2}$$

Since $q(\alpha,\beta)/(\beta+1)^2<1.604$ for $\beta\geq0$ and $\sum_{d=B-1}^{k-1}(1/d+1/k)^2<\sum_{d>B}(2/d)^2<4/B$, we obtain

$$\sum_{d=B+1}^{k-1}\left(\Pr_{c-k<d'\leq c}\{d|d'\}\right)^2\Pr_{(c-k)/d<e',e''\leq c/d}\{gcd(e',e'')=1\}<\sum_{d>B}\frac{7}{d^2}<\frac{7}{B}$$

Actually a more precise analysis of the sum $\sum_{d=B-1}^{k-1}(1/d+1/k)^2$ shows that in fact it is less than $1.635/B$; this proves

$$\Pr_{0\leq c',c''<k}\{\gcd(c-c',c-c'')>B\}<\frac{2.7}{B}$$