Signature Schemes Based on a Group of Hidden Order

Wandi Wei
Department of Mathematics
Xiamen University
Xiamen, Fujian 361005, China

Tran van Trung
Institute for Experimental Mathematics, University of Essen
Ellernstrasse 29, 45326 Essen, Germany
trung@exp-math.uni-essen.de

Spyros S. Magliveras
Department of Mathematical Sciences
Florida Atlantic University
Boca Raton, FL 33431, U.S.A
spyros@fau.edu

Abstract

In this paper we exploit the idea of combining the intractability of the integer factorization problem and the discrete logarithm problem to construct signature schemes based on groups of hidden order. Using this notion we describe two signature schemes. The first scheme arises by applying the idea to the ElGamal signature scheme. We show that the new scheme withstands all known attacks to the ElGamal scheme. The second scheme, considered as an application of the idea to the Schnorr signature scheme, provides a much shorter signature, as does the Schnorr scheme, and enhanced security.

Key words. Signature schemes, discrete logarithm problem, integer factorization problem, groups with hidden order.

1 Introduction

Let $p$ be a prime number such that the discrete logarithm problem in $\mathbb{Z}_p^*$ is intractable, and $\alpha$ a primitive element of $\mathbb{Z}_p^*$. Let $a$ be an integer with the property that finding the value of $a$ from $\beta$ and $\alpha$ is computationally infeasible, where $\beta = \alpha^a \mod p$. Here we conventionally use $c \mod m$ to denote the smallest nonnegative residue of $c$ modulo $m$.

Let

\[
K_{pub} = \{p, \alpha, \beta\}, \quad (1.1)
\]
\[
K_{pri} = \{a\}, \quad (1.2)
\]

be the public key and the private key, respectively. And let

\[
K = \{K_{pub}, K_{pri}\}.
\]
Then the ElGamal signature scheme can be described as follows [2, 3, 8]. To sign a message \( x \in \mathbb{Z}_p^* \), use the signing algorithm \( \text{sig}_k \): Choose a (secret) random number \( k \in \mathbb{Z}_{p-1}^* \), and then compute the signature \((\gamma, \delta)\) by

\[
\text{sig}_k(x, k) = (\gamma, \delta),
\]

where

\[
\begin{align*}
\gamma &= \alpha^k \mod p, \\
\delta &= (x - a\gamma)k^{-1} \mod (p - 1).
\end{align*}
\]

To verify the signature, use the verification algorithm \( \text{ver}_{\text{pub}} \):

\[
\text{ver}_{\text{pub}}(x, \gamma, \delta) = \text{true}
\]

if and only if

\[
\begin{align*}
0 < \gamma < p, \\
\beta^\gamma \delta &\equiv \alpha^x \pmod p.
\end{align*}
\]

For security considerations, to be discussed in a later section, a cryptographic hash function \( h : \{0, 1\}^* \rightarrow \mathbb{Z}_p \) should be used [3]. In this case, the only changes are: using

\[
\delta = (h(x) - a\gamma)k^{-1} \mod (p - 1)
\]

and

\[
\beta^\gamma \delta \equiv \alpha^{h(x)} \pmod p
\]

to replace (1.4) and (1.6), respectively.

All known attacks to the ElGamal signature scheme [3][8] involve performing computations in modular arithmetic with the order of the group \( G = \mathbb{Z}_p^* = \langle \alpha \rangle \) as a modulus. This order is \((p - 1)\), which is known from the public key. Now, if we could use some group \( G \) whose order is not publicly known, all those attacks would not work and this would enhance the security of the scheme.

In this paper we present signature schemes in which the order of such a group \( G \) is hidden. The main idea is to combine the intractability of the integer factorization problem and the intractability of the discrete logarithm problem. In Section 2 we describe \emph{signature scheme I} using a group of hidden order and discuss its security. In particular, we show the distinctness of the signatures of scheme I and give a thorough discussion of the fact that the scheme is secure against all the known attacks to ElGamal. We also present the performance behaviour of the scheme. In Section 3 we discuss a modification of scheme I to form \emph{signature scheme II} of \textquotedblleft Schnorr type\textquotedblright{} but using a group of hidden order as well. In the conclusions of Section 4 we announce some results of related work.

\section{Signature Scheme I}

\subsection{Description of the Signature Scheme I}

Let

\[
\eta = p_1 p_2,
\]
be the product of two large distinct primes $p_1$ and $p_2$ such that the discrete logarithm problem in $\mathbb{Z}_{p_i}^*$ is intractable for each $i = 1, 2$. Let $\alpha_i$ be a primitive element in $\mathbb{Z}_{p_i}$, and $a_i$ an element of $\mathbb{Z}_{p_i}^*$ with the property that

$$\gcd(p_1 - 1, p_2 - 1) \mid (a_1 - a_2) \quad (2.9)$$

Let

$$\beta_i = \alpha_i^{a_i} \mod p_i, \quad \text{for } i = 1, 2.$$

Under condition (2.9), the congruence equations

$$a \equiv a_1 \pmod{p_1 - 1},$$
$$a \equiv a_2 \pmod{p_2 - 1}$$

have a unique solution $a$ modulo $t$, where

$$t = (p_1 - 1)(p_2 - 1)/\gcd(p_1 - 1, p_2 - 1) = \phi(n)/\gcd(p_1 - 1, p_2 - 1).$$

Since $p_1$ and $p_2$ are distinct primes, the equations

$$\alpha \equiv \alpha_1 \pmod{p_1},$$
$$\alpha \equiv \alpha_2 \pmod{p_2}$$

have a unique solution $\alpha$ modulo $n$. Note that the order of $\alpha$ is $t$. Also, the equations

$$\beta \equiv \beta_1 \pmod{p_1},$$
$$\beta \equiv \beta_2 \pmod{p_2}$$

have a unique solution $\beta$ modulo $n$.

Let

$$K_{\text{pub}} = \{n, \alpha, \beta\},$$
$$K_{\text{pri}} = \{p_1, p_2, a\},$$

be the public and private keys of the scheme, respectively. Further, let

$$K = (K_{\text{pub}}, K_{\text{pri}}),$$

and $m = \lfloor 2^{k\phi(n)}/2 \rfloor$. To sign a message $x \in \mathbb{Z}_m$, use the signing algorithm $\text{sig}_K$: Choose a (secret) random number $k \in \mathbb{Z}_n$ with $\gcd(k, \phi(n)) = 1$, and then compute the signature $(\gamma, \delta)$ by

$$\text{sig}_K(x, k) = (\gamma, \delta),$$

where

$$\gamma = \alpha^k \mod n \quad (2.10)$$
$$\delta = (x - a\gamma)k^{-1} \mod t \quad (2.11)$$

To verify the signature, use the verification algorithm $\text{ver}_{K_{\text{pub}}}$:

$$\text{ver}_{K_{\text{pub}}}(x, \gamma, \delta) = \text{true}$$

3
if and only if

\[ \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod{n} \]  (2.12)

It is clear that signature scheme \( I \) works in the group \( G = \langle \alpha \rangle \). The important fact is that if the factors of \( n \) are not known, the order \( t \) of \( G \), known to the owner of the private key \( K_{pri} \), cannot computationally be determined by using only the knowledge of the public key \( K_{pub} = \{ n, \alpha, \beta \} \), even though the group is explicitly defined. This is because if the attacker could determine the value of \( t \), this would apparently help him to factorize \( n \), and therefore the RSA signature scheme would be broken. Thus the order of \( G \) is actually hidden.

2.2 Correctness of the Signature Scheme I

This signature scheme is correct, since

\[
\beta^\gamma \gamma^\delta \equiv \beta_1^\gamma (\alpha_1^k)^{(x-a_1)k^{-1}} \\
\equiv \beta_1^\gamma (\alpha_1)^{x-a_1} \\
\equiv \beta_1^\gamma ((\alpha_1)^x)^\gamma (\alpha_1)^x \\
\equiv \beta_1^\gamma \beta_1^{-\gamma} (\alpha_1)^x \\
\equiv (\alpha_1)^x \\
\equiv \alpha^x \pmod{p_1},
\]

and similarly,

\[ \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod{p_2}, \]

and then

\[ \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod{n}. \]

2.3 Injectivity of the Signature Scheme I

In this subsection we prove that the signatures on distinct messages produced by our signature scheme \( I \), are distinct. We first prove the following:

**Lemma 1** Let \( N \) be a natural number and \( p_1, p_2 \) primes such that \( p_1 > p_2 \geq 3 \) and \( 2^N < p_1 p_2 < 2^{N+1} \). Then, \( (p_1 - 1)(p_2 - 1)/\gcd(p_1 - 1, p_2 - 1) > 2^{N/2} \).

**Proof.** Since \( \gcd(p_1 - 1, p_2 - 1) \leq p_2 - 1 \), we have

\[ (p_1 - 1)(p_2 - 1)/\gcd(p_1 - 1, p_2 - 1) \geq p_1 - 1. \]

On the other hand, since \( p_1 > p_2 \) are primes, we have

\[ p_1(p_1 - 2) \geq p_1 p_2. \]

Hence,

\[ 2^N < p_1 p_2 \leq p_1^2 - 2p_1 < p_1^2 - 2p_1 + 1 = (p_1 - 1)^2. \]

Since \( p_1 > 2 \) the above inequality implies:

\[ 2^{N/2} < p_1 - 1. \]
and therefore,
\[(p_1 - 1)(p_2 - 1)/\gcd(p_1 - 1, p_2 - 1) \geq p_1 - 1 > 2^{N/2}\]

We now prove the assertion stated at the beginning of this subsection.

**Proposition 1** Assume the notation above, including \(m = [2^{\log_2 n}]/2\). Then, for any two distinct messages \(x_1, x_2 \in \mathbb{Z}_m\), scheme I will always generate two distinct signatures for \(x_1\) and \(x_2\).

**Proof.** First note that \(m < t\) by Lemma 1, where \(t = \phi(n)/\gcd(p_1 - 1, p_2 - 1)\). Let \((\gamma_1, \delta_1)\) be a signature for \(x_1\) and \((\gamma_2, \delta_2)\) a signature for \(x_2\). Let \(\gamma_i = \alpha^{k_i} \mod n, k_i \in \mathbb{Z}_m\) and \(\gcd(k_i, \phi(n)) = 1\), for \(i = 1, 2\).

We will show that \((\gamma_1, \delta_1) = (\gamma_2, \delta_2)\) implies that \(x_1 = x_2\).

Assume that \((\gamma_1, \delta_1) = (\gamma_2, \delta_2)\). Then, \(\gamma_1 = \gamma_2\), which implies that \(k_1 = k_2 = k\). Moreover, \(\delta_1 = \delta_2\), which means that
\[(x_1 - a\gamma_1)k^{-1} \equiv (x_2 - a\gamma_2)k^{-1} \pmod{t}.
\]

Since
\[
\gamma_1 = \gamma_2
\]
we have
\[x_1 \equiv x_2 \pmod{t}.
\]

Since \(m < t\), we have
\[x_1 = x_2.
\]

2.4 Security of the Signature Scheme I

In this section we discuss the security of signature scheme I under the assumption that the integer factorization and discrete logarithm problems are computationally intractable.

1. Suppose the RSA signature scheme is secure. Then the public key \(K_{pub} = (n, \alpha, \beta)\) in our scheme I does not release any information that can be used to compute any component of the private key \(K_{pri} = (p_1, p_2, a)\).

Suppose to the contrary that \(K_{pub} = (n, \alpha, \beta)\) can be used to get one of \(p_1, p_2\) or \(a\). We construct an RSA signature scheme with public key \((n, b)\) and private key \((p_1, p_2, a)\), where \(b = a^{-1} \mod \phi(n)\). Now choose \(\alpha\) as a message, then \(\beta\) is the RSA-signature for \(\alpha\). Now, if \(p_1\) (hence also \(p_2\)) or \(a\) can be computed from \(K_{pub}\), the value of \(a\) will be known in either case. Thus the RSA signature scheme is totally broken, a contraction to the assumption.
2. All possible known attacks discussed in [3] and [8] to ElGamal signature scheme will no longer work against our signature scheme \( I \) due to the fact that the group order is hidden. We discuss them here in some detail.

2.1. An adversary might attempt to forge a signature for a message \( x \) by selecting a random integer \( k \) and computing \( \gamma = \alpha^k \mod n \). The adversary could proceed in one of the following two approaches.

i) In the first approach, the adversary may wish to further determine the third component \( \delta = (x - a\gamma)k^{-1} \mod t \) of the signature. As the integer-factoring problem is computationally infeasible, the modulus value \( t \) can not be determined by the adversary. Even if \( t \) is known, the adversary still needs to compute \( a \), i.e., he needs to solve the discrete logarithm problem. Therefore, he can do no better than to choose a \( \beta \) at random, but then the success probability is only \( 1/t \), which is negligible for large \( p_1 \) and \( p_2 \).

At this point it is clear that signature scheme \( I \) is based on both the difficulty of the discrete logarithm problem and the integer-factoring problem.

ii) In the second approach, having computed \( \gamma \), the adversary tries to find \( \delta \) from \( \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod n \). But here, he must compute the discrete logarithm \( \log_\alpha \beta^\gamma \gamma^\delta \) in the group \( G = \langle \alpha \rangle \) whose order is unknown.

2.2. The adversary might attempt to forge a signature for a message \( x \) by selecting a random integer \( \delta \) and then trying to compute the corresponding \( \gamma \). Then he is faced with the problem of solving the congruence equation

\[
\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod n.
\]

As stated in [8], this is a problem for which no feasible solution is known.

2.3. The adversary might select \( \gamma \) and \( \delta \) and try to find a message \( x \) such that \( (\gamma, \delta) \) is a valid signature for \( x \). First, he is faced with computing the value of \( \log_\alpha \beta^\gamma \gamma^\delta \), again an instance of discrete logarithm problem for group \( G \), whose order is unknown to him. Thereafter, he has to break the hash function \( h \), if such is used.

3. Some ways of breaking the ElGamal signature scheme, when the scheme is used carelessly, are discussed in [3] and [8]. We will see here that none of these attacks can be employed to break our signature scheme \( I \).

3.1. Suppose that the same value for \( k \) is used for signing two different messages. In this case, the private key value \( a \) in the ElGamal signature scheme can be determined as stated in [8]. But this will not be a problem for our scheme. Suppose \( k \) has been used to sign two distinct messages \( x_1 \) and \( x_2 \), and the corresponding signatures are \( (\gamma_1, \delta_1) \) and \( (\gamma_2, \delta_2) \), respectively. Since the modulus value \( t \) is not publicly known, the value of \( a \) can not be computed from

\[
\delta_1 = (x_1 - a\gamma_1)k^{-1} \mod t
\]

and

\[
\delta_2 = (x_2 - a\gamma_2)k^{-1} \mod t.
\]

6
3.2. Suppose that no hash function is used for the message. It is indicated in [3] that the following existential forgery attack to the ElGamal signature scheme is possible. Select any pair of integers \((u, v)\) with \(\gcd(v, p - 1) = 1\). Let 
\[
\gamma = \alpha^v \beta^v \mod p \text{ and } \delta = -\gamma v^{-1} \mod (p - 1).
\]
Then \((\gamma, \delta)\) is a valid signature for \(x = \delta \alpha\). Now, if an adversary attempts to apply this attack to our signature scheme \(I\), he must first choose \(v\) with \(\gcd(v, t) = 1\), which is not easy because \(t\) is unknown. Suppose he could obtain such a \(v\), then after having computed \(\gamma = \alpha^v \beta^v \mod n\), he is still faced with computing \(\delta = -\gamma v^{-1} \mod t\), which is again impossible.

Of course, using a cryptographic hash function together with our scheme will make the scheme much stronger.

3.3. Suppose that the first step in the verification algorithm of the ElGamal signature scheme is neglected.

In this case, the adversary can sign a message, say \(x'\), of his choice provided a valid signature \((\gamma, \delta)\) for a message \(x\) is known. This is because \((\gamma', \delta')\) is a valid signature on \(x'\), where \(\gamma'\) is a solution to the congruence equations

\[
\begin{align*}
\gamma' & \equiv \gamma u \pmod{p - 1} \\
\gamma' & \equiv \gamma \pmod{p}
\end{align*}
\]

with

\[
u \equiv h(x')h(x)^{-1} \pmod{p - 1},
\]

and

\[
\delta' \equiv \delta u \pmod{p - 1}.
\]

If this attack is applied to our signature scheme \(I\), the following computations must be performed

\[
\begin{align*}
\gamma u \mod t, \\
h(x')h(x)^{-1} \mod t, \\
\delta u \mod t,
\end{align*}
\]

which are impossible because \(t\) is unknown.

4. Assume there is an oracle \(\mathcal{M}\), which generates a valid signature \((\gamma, \delta)\) for any given message \(x\) by giving any value \(\gamma, 1 \leq \gamma \leq n - 1\) for our signature scheme \(I\). Then we can prove that \(\mathcal{M}\) can be used to break the ElGamal signature scheme.

Let \(p_1, \alpha_1, a_1, \beta_1\) be qualified parameters for the ElGamal signature scheme. Then, we can find \(p_2, \alpha_2, a_2, \beta_2\) such that \(p_i, \alpha_i, a_i, \beta_i\) for \(i = 1, 2\) are qualified parameters for our scheme \(I\). Let \(n, \alpha, a, \beta\) be as defined in Section 2, and \(x \in \mathbb{Z}_{p_1}\). Then \(x\) can be regarded as a message in \(\mathbb{Z}_n\). Let \(\gamma < p_1\) and \((\gamma, \delta)\) be the valid signature produced by \(\mathcal{M}\) for \(x\). Then equation (2.12) holds. Since

\[
\begin{align*}
\alpha_1 & \equiv \alpha \pmod{p_1}, \\
\beta_1 & \equiv \beta \pmod{p_1}, \\
\delta_1 & \equiv \delta \pmod{p_1 - 1},
\end{align*}
\]

(2.12) gives

\[
\beta_1^\gamma \delta_1 \equiv \alpha_1^{p_1} \pmod{p_1},
\]

which means that \((\gamma, \delta_1)\) is a valid ElGamal signature for \(x\) since \(\gamma < p_1\).
2.5 Performance Analysis of the Signature Scheme I

Similarly to the ElGamal signature scheme, the signature generation algorithm of our scheme \( I \) is relatively fast, requiring one modular exponentiation \( \alpha^k \mod n \), the extended Euclidean algorithm for computing \( k^{-1} \mod t \), and two modular multiplications in computing \( \delta \). The exponentiation and application of the extended Euclidean algorithm can be done off-line when precomputation is possible. In this case, the signature generation requires only two on-line modular multiplications.

The expected amount of work for a signature verification is roughly \( \frac{9}{2} \left\lfloor \lg n \right\rfloor \) modular multiplications since it needs three modular exponentiations, each of which requires roughly \( \frac{3}{2} \left\lfloor \lg n \right\rfloor \) modular multiplications on average when standard techniques are used.

3 Signature Scheme II

Like scheme \( I \) described in Section 2, which is a scheme of “ElGamal type” but using a group with hidden order, we now present a modified version of scheme \( I \) to construct a new signature scheme \( II \) of “Schnorr type” using again a group of hidden order. This modified scheme \( II \), not only reduces the signature sizes drastically, as is the case for the Schnorr signature scheme \([7]\), but also strengthens the security against forgery.

Let \( p_1, p_2 \) and \( n \) be as defined in Section 2. Let \( q \) be a prime factor of \( \phi(n) \) and \( \alpha \) a \( q^{th} \) root of 1 modulo \( n \). Let \( a, \beta, K_{pub}, K \) and \( G \) be as defined in Section 2 and let

\[
K_{pri} = (p_1, p_2, a, q).
\]

Finally, let \( h : \{0, 1\}^* \rightarrow \mathbb{Z}_m \) be a cryptographic hash function with \( m < q \).

Signature scheme \( II \) works in group \( G \). To sign a message \( x \in \mathbb{Z}_n \), use the signing algorithm \( \text{sig}_K \): choose a (secret) random number \( k \in \mathbb{Z}_q \), and then compute the signature \( (\gamma, \delta) \) by

\[
\text{sig}_K(x, k) = (\gamma, \delta),
\]

where

\[
\gamma = h(x \parallel \alpha^k \mod n)
\]

and

\[
\delta = k - \alpha\gamma \mod q.
\]

To verify the signature, use the verification algorithm \( \text{ver}_{K_{pub}} \):

\[
\text{ver}_{K_{pub}}(x, \gamma, \delta) = \text{true}
\]

if and only if

\[
h(x \parallel \alpha^\delta\gamma) = \gamma.
\]

It is easy to see that for \( \gamma, \delta \) produced by the signing algorithm, we have

\[
\alpha^\delta\gamma \equiv \alpha^{k-\alpha\gamma} \alpha^{\alpha\gamma} \\
\equiv \alpha^k \mod n.
\]

In Schnorr’s signature scheme we have that \( n = p \) is a prime and \( q \) is a prime factor of \( p - 1 \). So, it is not hard to determine the value of \( q \). For the signature scheme \( II \) as
described in this section, the crucial fact is that the value of \( q \) is not published, and an attacker has no clue about its value as discussed in 2.1, above.

It is worth mentioning that even if the adversary could figure out the size of \( q \) from the value \( \delta \), he will not know \( q \) because the number of \( N \)-bit primes is astronomically large when \( N \) is large enough. In fact, if \( g(N) \) denotes the number of primes having size \( N \) bits, then \( g(N) > 2^{N-\log_2 N - 3} \).

This is easily derived as follows. Let \( \pi(x) \) denote the number of primes less than or equal to \( x \). Then we have \( \frac{x}{\ln x} < \pi(x) < 1.25506 \frac{x}{\ln x} \), for \( x \geq 17 \), see [5]. In particular, \( \pi(2^{N+1}) < 1.25506 \frac{2^{N+1}}{\ln 2^{N+1}} \), and \( \pi(2^N) > \frac{2^{N+1}}{\ln 2^{N+1}} \). Therefore,

\[
g(N) > \frac{2^{N+1}}{\ln 2^{N+1}} - 1.25506 \frac{2^N}{\ln 2^N} \geq 2^{N-\log_2 N - 3}.
\]

For example, when we choose a 200-bit prime \( q \) for scheme II, then the adversary needs to find \( q \) in a set of size \( q(200) \), which is much larger than \( 2^{189} \). With this 200-bit prime \( q \) scheme II provides 400-bit signatures, as does the Schnorr signature scheme with the same \( q \). The efficiency of scheme II is the same as that of the Schnorr scheme. The computation of \( \alpha^k \mod n \) can be carried out off-line, however.

**Remark 1** It is worth mentioning that the idea of using groups of hidden order can be applied to many known signature schemes such as the Nyberg-Rueppel scheme with message recovery [4], the Chaum-van Antwerpen undeniable signature scheme [1], etc. For example, using the same set-up as in our scheme I, a scheme of Nyberg-Rueppel type can be defined as follows: To sign a message \( x \in \mathbb{Z}_m \), choose a random \( k \in \mathbb{Z}_m \) and compute the signature \((\gamma, \delta)\) as follows:

\[
\text{sig}_K(x, k) = (\gamma, \delta),
\]

where

\[
\gamma = x\alpha^{-k} \mod n
\]

and

\[
\delta = a\gamma + k \mod t.
\]

The message \( x \) can be recovered from the signature:

\[
x = \alpha^\delta \beta^{-\gamma} \mod n.
\]

4 Conclusions

In this paper we have presented signature schemes using groups of hidden order. We have exploited the difficulty of both the integer factorization problem and the discrete logarithm problem to construct these schemes. We have discussed features of the new schemes and in particular, security improvements afforded by the new schemes in comparison to schemes based on groups of explicit, publically known order. Obviously, this idea is not restricted to the signature schemes described above. In fact, we intend to treat the case of undeniable signature schemes of Chaum-van Antwerpen type in a sequel to this paper. Further, we intend to present a public key cryptosystem which has the property that if one can break it, then he can break both the RSA and ElGamal public key cryptosystems.
Acknowledgement

The first author would like to thank the Institute for Experimental Mathematics, University of Essen, Germany, for its hospitality while carrying out parts of this work, and to Dr. D. Coppersmith for a helpful discussion.

References


