

## A New Version of ElGamal Signature Scheme (Versi Baru Skema Tandatangan ElGamal)

EDDIE SHAHRIL ISMAIL & YAHYA ABU HASAN

### ABSTRACT

*In the original ElGamal signature scheme and its variants, two secret integers-private key and one-time secret key are required to produce a signature on a message,  $m$ . The private key of a system will be used throughout the life of the system whereas the one-time secret key only be used once and must be regenerated (different one-time secret key) when signing different message. This paper introduces a new version of ElGamal signature that eliminates the use of one-time secret key. This property will make all attacks, aiming at revealing the one-time secret key irrelevant. The scheme also can be regarded as 'a right notion of signature scheme' because we use only one secret key to sign messages.*

*Keywords: cryptography; ElGamal signature scheme; discrete logarithm problem*

### ABSTRAK

*Dalam tandatangan ElGamal asal dan versi-versinya, dua integer sulit-kunci rahsia dan kunci sulit satu-masa diperlukan untuk menurunkan tandatangan pada suatu mesej,  $m$ . Kunci rahsia akan diguna sepanjang hayat sistem itu manakala kunci sulit satu-masa hanya diguna sekali dan mesti dijana semula (kunci sulit satu-masa yang baru) apabila menandatangani mesej seterusnya. Kertas ini memperkenalkan versi baru tandatangan ElGamal yang menghapuskan penggunaan kunci sulit satu-masa. Melalui ciri ini, semua serangan yang bertujuan mendedahkan kunci sulit satu-masa menjadi tidak lagi relevan. Sistem ini boleh dianggap sebagai 'skema tandatangan yang tepat' kerana memerlukan hanya satu kunci rahsia untuk menanda tangan mesej-mesej.*

*Katakunci: kriptografi; skema tandatangan ElGamal; masalah logaritma diskret*

### INTRODUCTION

ElGamal (1985) was the first to introduce a digital signature based on the difficulty of computing discrete logarithm problem in certain finite fields. His system required two secret integers to produce a valid signature on a message, that is, his private key and one-time secret key. The one-time secret key is used to generate a signature on a message and must be regenerated when signing different message. This property constantly followed in any ElGamal's version like Schnorr (1990), NIST (1993), Michels et al. (1996) and Lim & Lee (1998). In this paper, we do not follow this tradition but present a new version of ElGamal signature with two new attractive properties. First, the scheme requires no one-time secret key to sign messages. The scheme uses a private key of the scheme to sign all messages and this private key 'acts' as an exponent when generating a part of signature. Second, the scheme also prevents any attack that aiming at revealing or manipulating the one-time secret key and this consequently reduces the risk that faced by most of ElGamal-type signature schemes.

### RELATED WORK

Since the proposal of ElGamal signature scheme (ElGamal 1985) the first scheme to base on the discrete logarithm problem, there have been many approaches suggested to generalize the scheme. Schnorr (1990) proposed a variation of ElGamal signature scheme by introducing a second prime  $q$  to be a divisor of  $p - 1$ . The variation selects a generator  $g$  of order  $q$  but in the original ElGamal's scheme  $g$  was chosen such that it has an order  $p - 1$ . Schnorr also computes a second part of signature as  $s \bmod q$ . The Schnorr's proposal aimed to reduce the length of a signature on a message  $m$ . Later the National Institute for Standardization (NIST 1994) proposed an ElGamal-like signature scheme to reduce additionally the parameter  $r \bmod q$ . Then in their respective papers, Schnorr (1991) and Knobloch (1993) suggested using a hash function or a one-way function where to reduce the size of  $s \bmod q$  and where  $2^{50} \leq q \leq 2^{160}$  to reduce the size of  $s \bmod q$  and  $r \equiv g^k \pmod{p}$ .

Horster et al. (1994) gives a complete account on variations of ElGamal-like signature scheme. They illustrate eight kinds of generalizations and describe their respective

advantages and disadvantages and ended by stating the most secure and efficient scheme. Two variants of the ElGamal signature schemes have been standardized in USA as digital signature standard (DSS) (NIST 1994) and in Russia as GOST 34.10 (Michels et al. 1996). Korea also proposed KCDSA (Korean Certificate-based Digital Signature Algorithm) (Lim & Lee 1998) a candidate algorithm for Korean digital signature standard and now being standardized by the Korean Government. Recently, two variants of ElGamal-like signature schemes have been proven secure against adaptive attack for existential forgery under the random oracle model where the hash function is replaced with an oracle producing a random value for each new query. First variant  $H(m)$  is replaced with  $H(m || r)$  as in the Schnorr signature scheme. Pointcheval and Stern (1996) proved this variant is secure. Second variant, Pointcheval and Vaudenay (1996) claimed that the variant of DSA (NIST 1994) with  $r$  replaced by  $r \equiv H(g^k \text{ mod } p)$  is also secure in the random oracle model.

The abovementioned works use one-time secret key in their signing algorithms. This is due to several attacks as pointed out by ElGamal (1985), Lim and Lee (1998) and Michels et al. (1996). However, we will prove that, these attacks could be eliminated by using the life time private key as an exponent when generating a part of signature.

#### THE SYSTEM SETUP

Before we present our actual scheme, it is natural to define a general digital signature scheme. Our schemes involve three standard players, a signer who issues a signature on a message  $m$ , a verifier who convinces him or herself the authenticity of a signature and an adversary (Adv) who always attempting is at forging signatures.

#### DEFINITION 1

A digital signature scheme  $\Sigma$  is defined by the following:

- 1) The key generation algorithm GEN: On input  $1^k$ , where  $k$  is the security parameter, the algorithm GEN produces a pair (PK, SK) of public and secret keys. Algorithm GEN is probabilistic.
- 2) The signing algorithm SIGN: Given message  $m$  and a pair of public and secret key (PK, SK), SIGN produces a signature  $\sigma$ . The signing algorithm might be probabilistic and in some schemes it might receive other inputs as well.
- 3) The verification algorithm VER: Given a signature  $\sigma$ , a message  $m$  and a public key PK, VER tests whether  $\sigma$  is a valid signature of  $m$  with respect to PK. In general, the verification algorithm need not be probabilistic.

A signer first initializes the scheme by choosing and computing some parameters and keys required in GEN.

#### GEN Algorithm:

1. Select at random two large primes  $2^{511} < p < 2^{512} < q < 2^{160}$  where  $q$  is a divisor of  $p - 1$ .
2. Pick a generator  $g \in \mathbb{Z}_p^*$  that has an order  $q \text{ mod } p$  that is  $g$  where  $g$  must not equal to 1.
3. Choose an integer  $1 < x < q$  and computes  $y \equiv g^x \text{ (mod } p)$
4. Public key  $PK = (p, q, g, y)$  and private key  $SK = x$ .

To sign a message  $H(m || r)$ , the signer does the following:

#### SIGN Algorithm:

1. Compute
2. Compute
3. Calculate
4. A valid signature on a message  $m$  is given by  $(r, s)$ .

#### VER Algorithm:

Verifier accepts the pair  $(r, s)$  as valid signature if and only if the congruence  $r \equiv H(y^r g^s \text{ mod } p)$  holds.

The following theorem shows the correctness of our new version of ElGamal signature.

#### THEOREM 1

If a pair  $(r, s)$  is correctly derived from SIGN using keys generated in GEN then the validation of signature in VER is correct.

Proof:

Since  $(r, s)$  is a genuine signature, we then have

$$H(y^r g^s \text{ mod } p) \equiv H(g^{-xr} g^{H(m)^s - xr} \text{ mod } p) \equiv H(g^{H(m)^s} \text{ mod } p) \equiv r$$

#### A SIMPLE EXAMPLE

Now we present a simple example of this new version of ElGamal's signature. Let say Zainal wishes to have Suzie's signature on his document,  $m$  and say  $H(m) = 99$ . Suzie signs the message 99 using all parameters and keys generated in GEN algorithm and say all the outputs are given as below:

Suzie then signs the message by performing the following calculations:

1. Compute  $\lambda \equiv 99^{11} \equiv 73 \pmod{101}$
  2. Calculate  $r \equiv H(7^{73} \pmod{607}) \equiv H(601)$  and say  $H(601) = 66$
  3. Compute
- and sends her valid signature (66,54) on  $m$  to Zainal.

To validate the received signature, Zainal checks the equality of the congruence  $r \equiv H(570^{66} 7^{54} \pmod{607}) \equiv H(601) = 66$ . Since it holds, Zainal accepts the signature as valid.

In our scheme, we require no one-time secret key but we use only our private key  $x$  as in RSA signature scheme which uses only one secret key to sign messages. Note that, in the Korean Certificate Digital Signature Algorithm (KCDSA) (Lim and Lee 1998) and Schnorr signature scheme (Schnorr 1990), the integer  $r$  is calculated as

where the one-time secret key  $k$  must

be chosen at random. Thus to maintain this randomness, we suggest that the integers  $r$  in our scheme must be calculated as  $r \equiv H(g^{\omega H(m)^x} \pmod{p})$  where  $\omega$  is randomly chosen.

#### SECURITY AND EFFICIENCY PERFORMANCES

This section introduces some possible attacks in our signature scheme. Normally, the security properties are discussed due to the generation of parameters, algorithm for signing messages and verifying signatures.

In GEN, we require the two selected primes are safe. The number  $(p - 1)/2q$  must have prime factors greater than  $q$  so that attacks using small order subgroups of  $Z_p^*$  is not possible. The factors of  $p - 1$  then should not contain smooth integers to prevent from a key recovery attack (Lim & Lee 1997). Next, some information on generator  $g$  should not be available to the Adv otherwise it could be used to get some valuable tips of secret prior to the signer. The poor chosen primes and generator would give an opportunity to the Adv to forge a signature without knowing a private key of a signer as was pointed out in (Bleichenbacher 1996). This attack is called 'the parameters manipulation'.

Particularly, attacks on the schemes can be divided into two types. The first type recovers the signing private key  $x$  and the second type is to forge signatures without recovering  $x$ . Consider the first type of attack.

Attack 1. The obvious attempt is by brute-force attack to find the private key of the signer. Unfortunately, this is impossible due to the use of large primes  $p$  and  $q$ .

Attack 2. Let say the Adv is given a set of  $t$  signatures

$\{(r_i, s_i) | i = 1, \dots, t\}$  with their corresponding messages  $\{m_i | i = 1, \dots, t\}$ . Then the Adv may try to solve  $t$  equations of  $s = (H(m)^x - rx) \pmod{q}$ . These equations however yield  $t + 1$  unknowns. Thus the private key cannot be found because the number of solutions is large. Now, consider  $t = 2$  then we have  $s_1 = (H(m_1)^x - r_1 x) \pmod{q}$  and  $s_2 = (H(m_2)^x - r_2 x) \pmod{q}$ . Solving these two equations simultaneously is difficult since the term  $H(m_i)^x$  cannot be eliminated.

Attack 3. Trying to solve equation of the form  $g^{H(m)^x} = y^r g^s \pmod{p}$  is always equivalent to compute discrete logarithm in certain finite fields.

Consider the second type of attack aiming at forging signature without using the signing private key  $x$ .

Attack 4. To forge a signature of a message  $m$ , the Adv may first try to compute  $\beta = g^{H(m)\alpha} \pmod{p}$  for some  $\alpha$  chosen at random. Then the Adv has two ways of forging. (1) The Adv fixes  $r$  and tries to find  $s$  or (2) first fixes  $s$  and tries to find  $r$ . The two methods require the Adv to solve the congruence  $\beta = g^s y^r \pmod{p}$ . However, these are equivalent to solving a discrete logarithm problem in certain finite fields.

Attack 5. The Adv first chooses at random two integers  $r$  and  $s$  such that, it satisfies the congruence  $\beta = g^s y^r \pmod{p}$  for some known  $\beta$ . Adv then checks that  $\beta = g^{H(m)^x} \pmod{p}$  holds. However, we believe this happens with non-negligible probability.

For efficiency performance, our scheme achieves the same level of efficiency consideration of Schnorr signature scheme. The scheme should use a 'good' generator to speed up the modular exponentiation as it dominates the overall designated operations.

In SIGN, for the first part of signature, the signer needs only two exponentiations modulo  $p$  with a  $|q|$  bit exponent. For the second part of signature, the signer needs one exponentiation modulo  $q$  with a  $|q|$  bit exponent, one multiplication modulo  $q$  and involves no inverse operation. VER needs only two exponentiations of modulo  $p$  with a  $|q|$  bit exponent and involve no inverse computation.

#### CONCLUSION

We proposed a new version of ElGamal-like signature scheme. The presented scheme required no one-time secret key when generating the first part of signature. This new approach is strongly believed to provide the same level of security as previous ElGamal's versions of signature. Someone might claim that two unknown parameters (secret numbers) are better than only one. However, in our scheme we have proven that the use of one secret number is sufficient to provide a same level of security as previous two secret numbers-like ElGamal's signatures and hence reduces the time (operations) and cost generally when running the SIGN algorithm. Our signature scheme can easily be extended to any other signature scheme, for example, directed signature scheme. This means the verification

procedure can only be run by an intended verifier. This is done by replacing the generator  $g$  in  $r$  by a public where is a private key of the intended verifier.

## REFERENCES

- Bleichenbacher, D. 1996. Generating ElGamal Signatures Without Knowing the Secret In *Advances in Cryptology-Eurocrypt '96*, LNCS 1070, Springer-Verlag, 10-18.
- ElGamal, T. 1985. A Public Key Cryptosystem and A signature Scheme Based on Discrete Logarithm Problem. *IEEE Trans. Info. Theory*, IT-31, 469-472.
- Horster, P., Michels, M. & Petersen, H. 1994. Generalized ElGamal Signature Schemes for One Message Block. In *Proc. 2<sup>nd</sup> Int. Workshop on IT-Security*, 66-81.
- Knobloch, H.J. 1993. *A Remark on the Size of ElGamal-Type Digital Signatures*. Draft Version.
- Lim, C.H. & Lee, P.J. 1997. A Key Recovery Attack on Discrete Log Based Schemes Using A Prime Order Subgroup. In *Advances in Cryptology-Crypto '97*, LNCS 1294, Springer-Verlag, 249-263.
- Lim, C. H. & Lee, P. J. 1998. A Study on the Proposed Korean Digital Signature Algorithm. In *Advances in Cryptology-ASIACRYPT '98*, LNCS 1514. Springer-Verlag. 175-186.
- Michels, M., Naccache, D. & Petersen, H. 1996. GOST 34.10-A Brief Overview of Russia's DSA. *Computers and Security*, 15(8), 725-732.
- National Institute of Standards and Technology. 1993. Digital Encryption Standard. *FIPS PUB 42-6*. U.S Department of Commerce.
- National Institute of Standards and Technology. 1994. Digital Signature Standard. *FIPS PUB 186*. U.S Department of Commerce.
- Pointcheval, D. & Stern, J. 1996. Security Proof for Signature Schemes, In *Advances in Cryptology-Eurocrypt '96*, LNCS 1070, Springer-Verlag, 387-398.
- Pointcheval, D. & Vaudenay, S. 1996. On Provable Security for Digital Signature Algorithms. <http://www.dmi.ens.fr/~poinche/>.
- Schnorr, C.P. 1990. Efficient Identification and Signatures For Smart Cards. In *Advances in Cryptology-Crypto '89*, LNCS 435, Springer-Verlag, 235-251.
- Schnorr, C.P. 1991. Comment on DSA: Comparison of the Digital Signature Algorithm and the Signature Schemes of ElGamal and Schnorr, Letter to the Director of CSL/NIST.

Eddie Shahril Ismail  
 Pusat Pengajian Sains Matematik  
 Fakulti Sains & Teknologi  
 Universiti Kebangsaan Malaysia  
 43600 UKM Bangi  
 Selangor D.E  
 MALAYSIA

Yahya Abu Hasan  
 Pusat Pengajian Sains Matematik  
 Universiti Sains Malaysia  
 11800 USM Minden  
 Penang  
 MALAYSIA

