International Journal of Cryptology Research 9(2): 31 - 44 (2019)

A New Signing Scheme Based on BFHP and DLP

Amir Hamzah Abd Ghafar[*1] and **Muhammad Rezal Kamel Ariffin**[1,2]

[1]*Al-Kindi Laboratory of Cryptography, Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*
[2]*Department of Mathematics, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*

E-mail: amirghafar87@gmail.com
[*]*Corresponding author*

ABSTRACT

The notion of digital signing scheme was introduced in the new era of public key cryptography. Since then, many signing schemes have been designed utilizing one-way functions such as discrete logarithm, integer factorization, elliptic curve and quadratic residue. This paper will put forward a signing scheme which deploys the discrete logarithm problem (DLP) and bivariate function hard problem (BFHP).

Keywords: digital signing scheme, discrete logarithm problem, diophantine equations, bivariate function hard problem

1 INTRODUCTION

A digital signing scheme is very important in digital world today. It serves (a) to authenticate the originality of a digital message or document; (b) to check whether the message or document has been altered before; and (c) to prevent the sender to deny having sent the message or document. Many signing schemes have been introduced before by ElGamal (1985), Rivest et al. (1978) and Schnorr (1991). Furthermore, a standard digital signature scheme has been introduced to be practised by public users (Gallagher, 2013). All of the schemes mentioned either uses integer factorization problem (IFP) or discrete logarithm problem (DLP) which many have considered as one-way functions in the mathematical field. These functions ensure the authenticity, integrity and non-repudiation stated before.

A digial signing scheme must be efficient in computational complexity since there are millions of worldwide signing and verification process happen everyday in the digital realm. The vast number of applications today also require the scheme to be more flexible especially for small devices. In this paper, we will show how our signing scheme competes with other digital signing schemes with respect to their major operations such as modular exponentiation.

1.1 Contribution of This Paper

In our paper, we will introduce a scheme which combines discrete logarithm problem (DLP) and a newly defined one-way function called bivariate function hard problem (BFHP), introduced by Ariffin et al. (2013). Based on the abbreviations of both problems, we named this new scheme as **DLBF digital signing scheme**. This scheme will have probabilistic key-generation and signing algorithms which take randomized values as their inputs. Since it is newly introduced, not many attacks or analysis against DLBF scheme have been discovered. It is upon the readers that we depend the attacks will come from. Nevertheless, basic security and performance analysis are shown in this paper.

1.2 Outline of the Paper

We provide some mathematical backgrounds used in this paper in Section 2. It includes brief introduction to DLP and BFHP; two mathematical hard problem utilized by the new scheme. Then Section 3 introduces the new scheme in its full form. In Section 4 and 5, we give brief security and performance analysis against the scheme, respectively. Finally, we conclude the paper in Section 6.

2 MATHEMATICAL BACKGROUND

In this section, we provide brief introduction to discrete logarithm problem. Then, the application of linear diophantine equations with infinitely many solutions is also briefed since the notion of the application is crucial to introducing bivariate function hard problem (BFHP) afterwards. Both DLP and BFHP are two hard problems employed by the new scheme as its security strength. Finally, we define the term computational reduction to be used in our security analysis later in this paper.

2.1 Discrete Logarithm

The definition of discrete logarithm problem is as follows:

Definition 2.1 (Discrete Logarithm Problem)**.** *Given $h \in \mathbb{Z}_N$ is an element in group N and g is the generator of the same group, discrete logarithm problem is a problem to find x if $g^x \equiv h$ modulo N.*

Diffie-Hellman Key Exchange (Diffie and Hellman, 1976), the first scheme that introduced asymmetric key cryptography to the world, specifically uses discrete logarithm to achieve its security goal. As of today, there is still no polynomial time i.e effective algorithm has been discovered to solve this problem. Hence, DLP is still considered hard to be solved. The best algorithm to solve it, index calculus method, has $O\left(e^{(1+O(1))\left(\sqrt{ln(p)ln(ln(p))}\right)}\right)$ of time complexity which is in sub-exponential time (Schirokauer et al., 1996). For DLP in prime fields, number field sieve method has been proposed to solve it; also in sub-exponential time (Coppersmith et al., 1986).

2.2 Linear Diophantine Equations with Infinitely Many Solutions

Definition 2.2. *The successful process of prf-solving a Diophantine equation which has infinitely many solutions is the process of determining a preferred solution from a set of infinitely many solutions for the Diophantine equation.*

To further understand and obtain the intuition of Definition 2.2, we will now observe a remark by Herrmann and May (2008). It discusses the ability to retrieve variables from a given linear Diophantine equation. But before that we will put forward a famous theorem of Minkowski that relates the length of the shortest vector in a lattice to the determinant:

Theorem 2.1 (Minkowski's theorem)**.** *In a ω-dimensional lattice, l there exists a non-zero vector v with*

$$||\, v \,|| \leq \omega^{0.5} det(l)^{\frac{1}{\omega}} \tag{1}$$

We now put forward the remark.

Remark 2.1 (Herrmann and May, 2008)**.** *There is a method for finding small roots of linear modular equations $a_1x_1+a_2x_2+\cdots+a_nx_n \equiv 0 \pmod{N}$ with known modulus N. It is further assumed that gcd$(a_i, N) = 1$. Let X_i be upper bound on $|x_i|$. The approach to solve linear modular equation requires to solve the shortest vector in a certain lattice. We assume that there is only one linear independent vector that fulfills Minkowski bound (Theorem 2.1) for the shortest vector. Herrmann and May (2008) showed that under heuristic assumption that the shortest vector yields the unique vector $(y_1, \cdots, y_n)$ whenever*

$$\prod_{i=1}^{n} X_i \leq N. \tag{2}$$

If in turn we have

$$\prod_{i=1}^{n} x_i > N^{1+\epsilon}$$

Then the linear equation usually has N^{ϵ} many solutions, which is exponential in the bit-size of N. So there is no hope to find efficient algorithms that in general improve on this bound, since one cannot even output all roots in polynomial time.

We now put forward a corollary.

Corollary 2.1. *A linear Diophantine equation $f(x_1, x_2, \cdots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n = N$ that satisfies*

$$\prod_{i=1}^{n} x_i > N^{1+\epsilon}$$

is able to ensure secrecy of the sequence $x = \{x_i\}$.

Remark 2.2. *In fact if one were to try to solve the linear Diophantine equation $N = a_1x_1 + a_2x_2 + \cdots + a_nx_n$, where*

$$\prod_{i=1}^{n} x_i > N^{1+\epsilon}$$

any method will first output a sequence of short vectors $(x'_1, x'_2, \cdots, x'_n)$ as the initial condition. Then there will be infinitely many values from this initial condition that is able to reconstruct N.

2.3 Bivariate Function Hard Problem

Bivariate function hard problem or BFHP is a particular case of a linear Diophantine equations in two variables. It is first introduced by (Ariffin et al., 2013).

Definition 2.3. *$Z^{+}_{(2^{m-1}, 2^m-1)}$ is defined as a set of positive integers in the interval $\left(2^{m-1}, 2^m - 1\right)$. In other words, if $x \in Z^{+}_{(2^{m-1}, 2^m-1)}$, then x is a m-bit positive integer.*

Proposition 2.1. *Let $F(x_1, x_2, \cdots, x_n)$ be a multiplicative one-way function that maps $F : Z^n \to \mathbb{Z}^{+}_{(2^{m-1}, 2^m-1)}$. Let F_1 and F_2 be such function (either identical or non-identical) such that $a_1 = F(x_1, x_2, \cdots, x_n)$,*

$a_2 = F(y_1, y_2, \cdots, y_n)$ and gcd$(a_1, a_2) = 1$. Let $u, v \in \mathbb{Z}^+_{(2^{n-1}, 2^n-1)}$. Let (a_1, a_2) be public parameters and (u, v) be private parameters. Let

$$G(u, v) = a_1u + a_2v, \tag{3}$$

with the domain of the function G is $\mathbb{Z}^2_{(2^{n-1}, 2^n-1)}$ since the pair of positive integers $(u, v) \in \mathbb{Z}^2_{(2^{n-1}, 2^n-1)}$ and $Z^+_{(2^{m+n-1}, 2^{m+n}-1)}$ is the codomain of G since $a_1u + a_2v \in \mathbb{Z}^+_{(2^{m+n-1}, 2^{m+n}-1)}$. If at minimum $n - m - 1 = k$ where 2^k is exponentially large for any probabilistic polynomial time (PPT) adversary through all possible answers, it is infeasible to determine (u, v) over $\mathbb{Z}$ from G. In addition to that, (u, v) is unique for $G(u, v)$ with high probability.

Proof. Refer to Ariffin et al. (2013). □

Remark 2.3. *The preferred pair $(u, v) \in \mathbb{Z}$, is the prf-solution for (3). The preferred pair (u, v) is one of the possible solutions for (3) given by*

$$u = u_0 + a_2t \tag{4}$$

and

$$v = v_0 - a_1t \tag{5}$$

for any $t \in \mathbb{Z}$.

Remark 2.4. *We remark here that the Diophantine equation given by $G(u, v)$ is solved when the preferred parameters (u, v) over $\mathbb{Z}$ are found. That is the BFHP is prf-solved when the preferred parameters (u, v) over $\mathbb{Z}$ are found.*

Definition 2.4. *(Computational Reduction). Let Π_1 and Π_2 be two cryptographic hard problems where $\Pi_1 \neq \Pi_2$. We say that a problem Π_1 is reducible to a problem Π_2 if we are able to show that Π_2 can be solved using a polynomial time algorithm, $\mathcal{A}$ then Π_1 also can be solved by $\mathcal{A}$. We denote this as $\Pi_1 \leq_p \Pi_2$.*

Example 2.1. *Regarding to RSA equation and IFP which holds the security of RSA cryptosystem, we say that solving RSA equation $\leq_p$ solving IFP. That means if IFP is solved, then RSA equation also can be solved.* ■

3 THE PROPOSED SIGNING SCHEME

DLBF signing scheme uses DLP and BFHP as its hard mathematical problems. As an example of application of the problems, the first public-key cryptosystem by Diffie and Hellman (1976) utilizes DLP while AA_β cryptosystem by Ariffin et al. (2013) uses BFHP. In both cryptosystems, no threatening attacks that fully severe their security strength are known. As a method of verification, we use some useful properties from Schnorr signature scheme (Schnorr, 1991) in DLBF. We find it elegant to concatenate the output from the signature generation process with message M. The resultant concatenation will be passed as input for any hash function H given H is modelled as random oracle. DLBF digital signing scheme includes key generation algorithm, signing algorithm and verification algorithm.

Algorithm 1a Key Generation algorithm

```
Require: Two integers $m$ and $n$.
Ensure: Public key: $(A, B, g, p)$. Private keys: $(a, b)$
  $p \xleftarrow{\$} \mathbb{Z}_{2^m}$
  $g \leftarrow \mathbb{Z}_p$ where $g$ is a primitive root of group $\mathbb{Z}_p$.
  $a, b \xleftarrow{\$} \mathbb{Z}_{2^n}$.
  $A \leftarrow g^a \pmod{p}$.
  $B \leftarrow g^b \pmod{p}$.
  Return $(A, B, g, p)$ and $(a, b)$.
```

We can see that in Algorithm 1a, A, B, g and p have the size of 2^m-bits while a and b have the size of 2^n-bits where $n > m$. It is important for 2^m to be exponentially large to adhere with Definition 2.1.

Algorithm 1b Signing algorithm

```
Require: Private keys, $(a, b)$, message, $M$ and hash function, $H$.
Ensure: Signature, $\sigma$.
$x, y \xleftarrow{\$} \mathbb{Z}_{2^m}$.
$c = ax + by$.
$k \xleftarrow{\$} \mathbb{Z}_{2^n}$ where $c - k > 2^m$.
$r \leftarrow g^k \pmod{p}$.
$e = H(M \,||\, r)$.
$s = c - k$
Return $\sigma = (x, y, e, s)$.
```

It is also important in Algorithm 1b that 2^{n-m} to be exponentially large to follow Proposition 2.1 when calculating c.

Algorithm 1c Verification algorithm

```
Require: Signature, $\sigma$, public keys, $(A, B, g, p)$ and message, $M$.
Ensure: ACCEPT or REJECT
$A^x B^y g^{-s} \equiv r' \pmod{p}$
if $H(M \,||\, r') = e$ then
    Return ACCEPT
else
    Return REJECT.
end if
```

In the Verification algorithm, we have to check if the message digest of hash function, H with inputs concatenation of M and r' will produce the same value as e. If yes, σ is valid. Else, σ is a fraud signature.

Theorem 3.1. *The proposed signing scheme is correct.*

Proof. It is easy to see that

$$\begin{aligned} A^x B^y g^{-s} &\equiv g^{ax} g^{by} g^{-(c-k)} \\ &\equiv g^{c-(c-k)} \\ &\equiv g^k \\ &\equiv r' \pmod{p} \end{aligned}$$

The correct r' will produce $e' = H(M \,||\, r') = e$. □

Example 3.1. *Take $n = 64$ and $m = 32$. Along will find $p = 2750118359 \in \mathbb{Z}_{2^{32}}$ and $g = 326$ where g is a primitive root of group $\mathbb{Z}_p$. Along will also generate randomly $a = 16597089552141307822$ and $b = 13251521636192730810$ where $a, b \in \mathbb{Z}_{2^{64}}$. These are his private keys. After that, he will calculate and find $A = 657740542$ and $B = 2608124422$. Together with g and p, these are his public keys.*

Along wants to sign a message, $M = 100$ and send it to Busu. First, he will need to generate two random parameters, $x = 3590238451$ and $y = 2499976781$ for $x, y \in \mathbb{Z}_{2^{32}}$ where gcd$(x, y) = 1$. He then will obtain $c = 92716005488189048993954786332$. He then will choose $k = 8375163739492536320792514204$ randomly from $\mathbb{Z}_{2^{64}}$ where $c - k > 2^m$ and calculate $r = 487787984$. He will concatenate r with M and find $e = H(M \,||\, r)$ where h is a hash function e.g. SHA-256. Finally, he will calculate $s = 84340841748696512673162272128$ The resultant $\sigma = (x, y, s, e)$ is Along's signature for message, M and will be sent to Busu.

To verify, Busu will calculate r'

$$\begin{aligned} &657740542^{3590238451} \cdot 2608124422^{2499976781} \cdot 326^{-84340841748696512673162272128} \\ &\qquad\qquad \equiv 487787984 \pmod{2750118359}. \end{aligned}$$

Upon obtaining r', he will concatenate it with the message to be verified, M. If $H(M \,||\, r') = e$, then he can verify that σ is indeed come from Along. Else, σ is a fraud. □

4 SECURITY ANALYSIS

The aim for this section is to provide vrief security analysis against the new scheme.

Proposition 4.1. *Solving the proposed signing scheme $\leq_p$ solving BFHP.*

Proof. If an adversary can solve BFHP, then the adversary can find the value of t such that

$$a = a_0 + xt \tag{6}$$

$$b = b_0 + yt. \tag{7}$$

As the values of a_0 and b_0 can be found from extended euclidean algorithm, then the secret a and b can also be found. □

Proposition 4.2. *Solving the proposed signing scheme $\leq_p$ solving DLP.*

Proof. If an adversary can solve DLP, then given the public keys, A, B, g and p, the adversary can find a such that $A \equiv g^a \pmod{p}$ and b such that $B \equiv g^b \pmod{p}$. □

Remark 4.1. *Propositions 4.1 and 4.2 show that to solve this new signing scheme, the adversary needs to solve either DLP or BFHP which has been considered the hard problems in Section 2.1 (see page 33) and Section 2.3 (see page 35) respectively.*

Corollary 4.1. *Forging the signature for the proposed signature scheme $\leq_p$ solving BFHP or DLP.*

Proof. If BFHP or DLP can be solved, then the values of a and b can be found. Using a and b, adversary can produce c' such that

$$c' = ax' + by' \tag{8}$$

where $x', y' \xleftarrow{\$} \mathbb{Z}_{2^n}$. An adversary can proceed to choose random $k' \xleftarrow{\$} \mathbb{Z}_{2^{2n}}$ and calculate $s' = c' - k' > 2^n$. The adversary then can forge $e' = H(M_{adv} \,||\, r')$ where M_{adv} is the forgery message and $r' \equiv g^{k'} \pmod{p}$ then send the forgery signature, $\sigma' = (x', y', s', e')$ to the intended recipient.

The recipient will use the forgery signature and find that

$$\begin{aligned} A^{x'} B^{y'} g^{-s'} &\equiv g^{ax'} g^{by'} g^{-(c'-k')} \\ &\equiv g^{c'-(c'-k')} \\ &\equiv g^{k'} \\ &\equiv r' \pmod{p} \end{aligned}$$

and verify $e' = H(M_{adv} \,||\, r')$. □

Remark 4.2. *It is still unknown if the problem to find a and b is equivalent to solving BFHP or DLP.*

5 PERFORMANCE ANALYSIS

From Proposition 2.1, values of 2^k where $n - m - 1 = k$ must be exponentially large numbers. Considering 2^{128} is still infeasible to be calculated by modern computers, we are suggesting $n > 258$ while $m = \lfloor \frac{m}{2} \rfloor$. This will causes an exponentiation of a number in size of at least 2^{m+n} for verification.

Parameters (g, p, A, B) can be stored by signers and verifiers while $(a, b\ )$ is exclusively stored by signers. During signing, only **one** additional and **one** modular exponentiation required. In verification process, **three** modular exponentiation required. Because the scheme uses fixed g, there will be further speed-up in computation on signer's side.

In Table 1, we compare the number of major processes of our signing scheme with three renowned signing schemes which are RSA, ElGamal and Schnorr signature schemes. These processes are calculated from the textbook algorithm of each cryptosystems. We do not consider the processes from the further modification or implementation of the cryptosystems. Every scheme needs to hash its message in both signing and verification algorithms, thus we omit it in our table.

Signing Scheme	Number of Operations	
	Signing Algorithm	Verification Algorithm
RSA	1 modular exponentiation	1 modular exponentiation
ElGamal	1 modular exponentiation 1 modular multiplication	1 modular multiplication
Schnorr	1 modular exponentiation 1 subtraction	1 modular exponentiation
Our signing scheme	1 modular exponentiation 1 addition 1 subtraction	1 modular exponentiations

Table 1: Operations in signature schemes

Out of all three signing schemes that we have compared, only Schnorr signature scheme is proven secure Seurin (2012) using the formalization of security for signing scheme Goldwasser et al. (1988). For RSA, even though its operations are simple, but it needs to utilize the Optimal Asymmetric Encryption Padding (OAEP) to ensure its security Kaliski and Staddon (1998).

It can be seen that our signing scheme shares the same number of operations with Schnorr scheme. As addition and subtraction operations can be ignored, a drawback from the scheme is the number of modular exponentiation in verification algorithm. However, recent implementation shows that complexity of modular exponentiation can be effectively reduced using several methods including Montgomery reduction Montgomery (1985) and Karatasuba multiplication Karatsuba and Ofman (1963).

6 CONCLUSION

We have shown a new signing scheme called DLBF digital signing scheme in this paper. Its security is based on the discrete logarithm problem and bivariate function hard problem. The signing method of DLBF requires one modular exponentiation and one addition process while the verifying method needs two modular exponentiation processes. We adopted a method of verification from Schnorr signature scheme to assist us in building this scheme.

REFERENCES

Ariffin, M., Asbullah, M., Abu, N., and Mahad, Z. (2013). A new efficient asymmetric cryptosystem based on the integer factorization problem. *Malaysian Journal of Mathematical Sciences*, 7(S):19–37.

Coppersmith, D., Odlzyko, A. M., and Schroeppel, R. (1986). Discrete logarithms in $GF(p)$. *Algorithmica*, 1(1-4):1–15.

Diffie, W. and Hellman, M. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology*, pages 10–18. Springer.

Gallagher, P. (2013). Digital signature standard (dss). *Federal Information Processing Standards Publications, volume FIPS*, 186.

Goldwasser, S., Micali, S., and Rivest, R. L. (1988). A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308.

Herrmann, M. and May, A. (2008). Solving linear equations modulo divisors: On factoring given any bits. In *Advances in Cryptology-ASIACRYPT 2008*, pages 406–424. Springer.

Kaliski, B. and Staddon, J. (1998). PKCS # 1: RSA cryptography specifications version 2.0. Technical report, RFC 2437, October.

Karatsuba, A. and Ofman, Y. (1963). Multiplication of multidigit numbers on automata. In *Soviet physics doklady*, volume 7, page 595.

Montgomery, P. L. (1985). Modular multiplication without trial division. *Mathematics of computation*, 44(170):519–521.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

Schirokauer, O., Weber, D., and Denny, T. (1996). Discrete logarithms: the effectiveness of the index calculus method. In *Algorithmic number theory*, pages 337–361. Springer.

Schnorr, C.-P. (1991). Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174.

Seurin, Y. (2012). On the exact security of schnorr-type signatures in the random oracle model. In *Advances in Cryptology–EUROCRYPT 2012*, pages 554–571. Springer.