Easy Simple Factoring-based Digital Signature Scheme

Conference Paper · December 2020

DOI: 10.23919/ICITST51030.2020.9351341

4 authors, including:

Nor Haniza Sarmin
Universiti Teknologi Malaysia
475 PUBLICATIONS **871** CITATIONS

SEE PROFILE

Eddie Shahril Ismail
Universiti Kebangsaan Malaysia
70 PUBLICATIONS **482** CITATIONS

SEE PROFILE

Ji-Jian Chin
Multimedia University
42 PUBLICATIONS **179** CITATIONS

SEE PROFILE

Easy Simple Factoring-based Digital Signature Scheme

Sook-Yan Hue
Department of Mathematical Sciences
Universiti Teknologi Malaysia
Johor, Malaysia
sook.yan.hsy@gmail.com

Nor Haniza Sarmin
Department of Mathematical Sciences
Universiti Teknologi Malaysia Johor,
Malaysia
nhs@utm.my

Eddie Shahril Ismail
Department of Mathematical Sciences
Universiti Kebangsaan Malaysia
Selangor, Malaysia
esbi@ukm.edu.my

Ji-Jian Chin
Faculty of Engineering
Multimedia University
Cyberjaya, Malaysia
jjchin@mmu.edu.my

2020 15th International Conference for Internet Technology and Secured Transactions (ICITST) | 978-1-913572-21-1/20/$31.00 ©2020 23919/ICITST51030.2020.9351341

***Abstract*—A newly developed signature scheme, namely the Easy Simple Factoring-based (ESF) signature scheme is proposed in this paper. First, we construct a standard identification scheme based on the ESF assumption. Then, the standard identification scheme is converted to a signature scheme by utilizing the Fiat-Shamir transformation. Our signature scheme is provable secure in the random oracle model, and the security proof is shown. The efficiency analysis of our scheme is conducted and compared with the Full Domain Hash (FDH) signature scheme. Our scheme has a more powerful efficiency performance in signing and verifying algorithms than the FDH signature scheme when the public keys u and γ in our signature scheme are fixed to small bit lengths.**

***Index Terms*—digital signature scheme, provably security, random oracle model, identification scheme, Fiat-Shamir transformation**

I. Introduction

A digital signature is a public key cryptographic algorithm that is created to guarantee message integrity, authentication and non-repudiation. In 2018, Ismail *et al.* [1] invented an Easy Simple Factoring-based (ESF) cryptosystem, a variation of the Rivest-Shamir-Adleman (RSA) cryptosystem [2], based on the hardness of the factoring problem. The level of efficiency of the ESF cryptosystem is higher compared to the RSA cryptosystem.

In 1986, Fiat and Shamir [3] introduced a method to transform canonical identification schemes into digital signature schemes. Fiat-Shamir (FS) transformation creates an efficient scheme for a digital signature and maintains its security against chosen message attacks.

In this paper, we proposed a signature scheme based on the ESF assumption [1], followed by its security and efficiency analysis. The ESF signature design begins with the construction of an identification scheme based on the ESF assumption. Then, FS transformation [3] is utilized to convert the identification scheme into a digital signature scheme.

A. Our Contribution

Based on the ESF assumption, we design a new digital signature scheme. We propose a signature scheme that serves as an alternative to the RSA-based signature scheme, the Full Domain Hash (FDH) signature scheme [4]. Our scheme provides provable security and performs faster in signing and verifying algorithms compared to the FDH signature scheme if we fix the public keys u and γ in our scheme to small bit lengths.

To achieve this, we first come out with an identification scheme based on the ESF assumption that secure against passive attack, followed by its security proof. Then we convert the identification scheme to a signature scheme by utilizing the FS transformation.

Additionally, we prove that our scheme is secure by presenting its security proof, reducing the hardness of an existential forgery under the chosen message attack (EUF-CMA) to the intractability of solving the ESF assumption. Lastly, we show an implementation with higher efficiency performance in signing and verifying algorithms compared with the RSA-FDH signature scheme. As a result, our signature scheme is more efficient than FDH signature scheme.

B. Organization

The structures of this paper are organized as follows: The paper begins with the introduction in Section I. Then, we include the preliminaries in Section II. Our signature scheme is presented in Section III followed by its security proof in Section IV and efficiency analysis in Section V. Finally, we conclude our paper in Section VI.

II. Preliminaries

First, we recall the formal definition of the ESF Assumption introduced in [1]. Later, the concept of identification scheme

and FS transformation are revisited. Then, we present the definition of the signature schemes and the security of signature schemes.

A. Easy Simple Factoring-based (ESF) Assumption

The ESF assumption proposed in [1] where given (N, X_1, X_2, u, γ) and compute (v, λ, ω, x). Let $N = pq$ such that p and q are primes. Let $\omega = s(p-1)+1 = r(q-1)+1$ where r/s is the simplest ratio of $(p-1)/(q-1)$. A Linear Diophantine equation $uv + \gamma\lambda = \omega$ is formed where u and γ are public keys while v and λ are private keys.

Then, the functions $f_{N,u} : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ and $f_{N,\gamma} : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ are given by $f_{N,u} = x^u = X_1 \mod n$ and $f_{N,\lambda} = x^\lambda = X_2 \mod N$, respectively. In other words, for every pair of $X_1, X_2 \in \mathbb{Z}_N^*$ there exists a unique $x \in \mathbb{Z}_N^*$ that satisfying $x^{u+\gamma} \equiv x^u \cdot x^\gamma \mod N \equiv X_1 \cdot X_2 \mod N$. For x, X_1 and X_2 satisfying this relation, we write $X_1^v \cdot X_2^\lambda \equiv (x^u)^v \cdot (x^\gamma)^\lambda \equiv x^{uv+\gamma\lambda} \equiv x^\omega \equiv x \mod N$. Hence, we define f^{-1} as follow:

$$f^{-1}(X_1 \cdot X_2) = X_1^v \cdot X_2^\lambda = (x^u)^v \cdot (x^\gamma)^\lambda \equiv x^{uv+\gamma\lambda} \equiv x^\omega \equiv x \mod N$$

Let $\mathcal{K}$ be a probabilistic polynomial-time algorithm (PPT) that, on input 1^k, outputs a modulus N that is the product of 1024-bit primes. Compute $\omega = s(p-1)+1 = r(q-1)+1$. Choose the public keys u and γ randomly. Compute the private keys v and λ from the Linear Diophantine equation $uv+\gamma\lambda = \omega$ by extended Euclidean algorithm.

Definition 1. *We say that the ESF problem is hard relative to $\mathcal{K}$ if for all PPT algorithms A, the probability of adversary to able to extract x is negligible as follows:*

$$Pr[(N, u, \gamma, v, \lambda, \omega) \leftarrow \mathcal{K}(1^k); X_1, X_2 \leftarrow \mathbb{Z}_n^*; x \leftarrow A(N, u, \gamma, X_1, X_2) : x^{u+\gamma} \equiv X_1 \cdot X_2 \mod n] < \epsilon$$

B. Standard Identification

The standard identification scheme is a canonical three-move protocol as defined by Bellare and Palacio [5]. First, Prover $\mathcal{P}$ generates commitment Cmt and sends it as a message to $\mathcal{V}$. Verifier $\mathcal{V}$ selects a challenge Ch uniformly from a random set, called challenge set $ChSet_\mathcal{V}$ associated to its input, and sends the challenge to $\mathcal{P}$. Prover $\mathcal{P}$ generates a response Rsp and sends it to $\mathcal{V}$. Lastly, $\mathcal{V}$ deterministically outputs a value $d \leftarrow Veri(Cmt, Ch, Rsp)$ such that $d = 1(accept)$ while $d = 0(reject)$.

Theorem 1. *The canonical identification protocol is secure against a passive attack if the protocol is honest-verifier zero knowledge and satisfies soundness.* [6].

Proof. Refer to [6]. □

C. Fiat-Shamir Transformation

The Fiat-Shamir (FS) transformation is first introduced by Fiat and Shamir [3] in the year 1984. The Fiat-Shamir method is used to transform an identification scheme into a digital signature scheme. According to Katz [6], the FS transformation is having the prover run an instance of the identification protocol by itself, generating the challenge c by applying a hash function H to the first message $\mathcal{I}$ and then computing an appropriate response r. Katz [6]'s format of the Fiat-Shamir transformation's definition is recalled as follows:

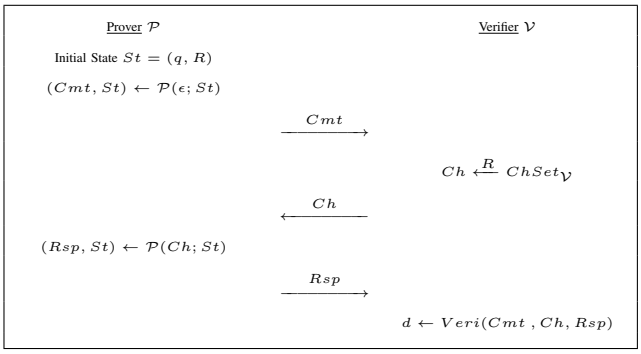

Figure 1. Standard Identification (A Canonical Protocol) [5]

Let $\Pi = (Gen, \mathcal{P}, \mathcal{V})$ be a canonical identification scheme where the verifier's challenges are chosen uniformly from Ω. Let $H : \{0,1\}^* \to \Omega$ be a hash function.

Key generation: Run $Gen(1^k)$.

Signature generation: To sign message m using secret key sk, the signer do the following.

1) Run the prover algorithm $\mathcal{P}(sk)$ to generate an initial message I.
2) Compute $c := H(I, m)$.
3) Compute the appropriate response r to the challenge c using $\mathcal{P}(sk)$.
   The signature σ is (I, r).

Signature verification: To verify the signature (I, r), the verifier proceed as follows:

1) Compute $c := H(I, m)$.
2) The signature is valid if and only if (pk, I, c, r) is an accepting transcript.

III. ESF Signature Scheme

A. Standard Identification Scheme based on ESF Assumption

We present the structure of the ESF-based standard identification scheme as follows:

Key Generation.: The parameters (N, u, γ, v, λ, ω) are generated as mentioned in Section II. Then, $x \in \mathbb{Z}_N^*$ is chosen. Next, compute $X_1 \leftarrow x^u \mod N$ and $X_2 \leftarrow x^\gamma \mod N$. Then, compute public key $pk \leftarrow ((1^k, N, u, \gamma), X_1, X_2)$ and private key $sk \leftarrow ((1^k, v, \lambda, \omega), x)$.

Identification Protocol.: The identification protocol run by prover and verifier is presented in the following list and illustrated in Fig. 2.

a) Prover $\mathcal{P}$ computes $y \leftarrow \mathbb{Z}_N^*$.
b) $\mathcal{P}$ sends the commitment $Y_1 \cdot Y_2$ such that $Y_1 \leftarrow y^u \mod N$ and $Y_2 \leftarrow y^\gamma \mod N$ to Verifier $\mathcal{V}$.
c) $\mathcal{V}$ then computes challenge $c \leftarrow \mathbb{Z}$ and send it to prover, $\mathcal{P}$.
d) $\mathcal{P}$ sends the response z to $\mathcal{V}$ where $z \leftarrow x^c y \mod N$.

e) $\mathcal{V}$ accept if $z^{u+\gamma} \equiv (X_1 \cdot X_2)^c (Y_1 \cdot Y_2)$ where Y_1, Y_2, $z \in \mathbb{Z}_N^*$, else reject.

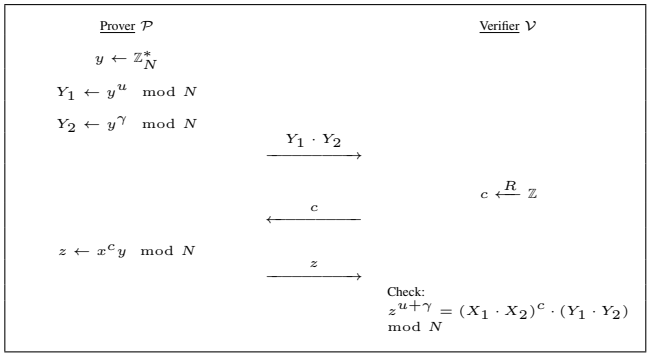

Figure 2. ESF-based Identification Protocol

Proposition 1. *Based on Theorem 1, ESF-based identification protocol is secure against passive attacks.*

Proof. Completeness is proven in the scheme since $\mathcal{P}$ knows x, then $Pr[\mathcal{V} \text{ accept }] = 1$.

Soundness is proven as follows. Suppose from two valid challenge-response pairs $(c_1, z_1), (c_2, z_2)$ for the same commitments Y_1 and Y_2 is able to extract the secret key x as follows: compute a and b by extended Euclidean algorithm such that $(c_1 - c_2)a + (u+\gamma)b = 1$. Therefore, we can compute x such that $x \leftarrow (z_1/z_2)^a (X_1 \cdot X_2)^b$.

Lastly, we show a simulator $\mathcal{S}$ such that its purpose is to output $(\tilde{Y_1} \cdot \tilde{Y_2}, \tilde{c}, \tilde{z})$ where $\tilde{z}^{u+\gamma} = (\tilde{X_1} \cdot \tilde{X_2})^{\tilde{c}} (\tilde{Y_1} \cdot \tilde{Y_2})$. The simulator $\mathcal{S}$ then outputs $(\tilde{z}^{u+\gamma}/(x_1 \cdot x_2)^{\tilde{c}}, \tilde{c}, \tilde{z})$. Thus, we have proved that $(\mathcal{P}, \mathcal{V})$ is simulatable. □

B. The Construction of ESF Signature Scheme

By utilizing the Fiat-Shamir transformation, the identification scheme in Section III-A is transformed to a signature scheme.

Key generation.: Run $Gen(1^k)$ to generate (pk, sk) such that $pk \leftarrow ((1^k, N, u, \gamma), X_1, X_2)$ and $sk \leftarrow ((1^k, v, \lambda, \omega), x)$ where $x \leftarrow \mathbb{Z}_N^*$, $X_1 \equiv x^u \mod N$ and $X_2 \equiv x^\lambda \mod N$.

Signature generation.: The signing algorithm is done by signer as follows:

1) Compute $y \leftarrow \mathbb{Z}_N^*$.
2) Compute $I := Y_1 \cdot Y_2 \mod N$ such that $Y_1 \leftarrow y^u \mod N$ and $Y_2 \leftarrow y^\gamma \mod N$.
3) Compute $c := H(I, m)$ where H is a public SHA-256 hash function, $H : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^{256}$ and m is a message.
4) Compute $z \leftarrow x^c y \mod N$.
   The signature (I, z) is generated.

Signature verification: The verification algorithm is done by verifier as follows:

1) Compute $\tilde{c} := H(I, m)$.
2) Accept the signature if and only if $z^{u+\gamma} \equiv (X_1 \cdot X_2)^{\tilde{c}} I$. Else, reject.

Proposition 2. *For the correctness, the equation $z^{u+\gamma} \equiv (X_1 \cdot X_2)^{\tilde{c}} I$ should hold.*

Proof.

$$\begin{aligned} z^{u+\gamma} &\equiv (x^{\tilde{c}} \cdot y)^{u+\gamma} \mod N \\ &\equiv (x^{u\tilde{c}} \cdot x^{\gamma\tilde{c}}) \cdot (y^u \cdot y^\gamma) \mod N \\ &\equiv (X_1^{\tilde{c}} \cdot X_2^{\tilde{c}}) \cdot (Y_1 \cdot Y_2) \mod N \\ &\equiv (X_1 \cdot X_2)^{\tilde{c}} \cdot I \mod N \end{aligned}$$

□

IV. Security Proof

We show the security proof of our signature scheme in this section. For the security proof, we use the technique proposed by Coron [7].

Theorem 2. *Suppose the ESF assumption is (t', ϵ')-secure. Then the ESF signature scheme is (t, ϵ)-secure if Equation 1 and 2 hold.*

$$t = t' - (q_{hash} + q_{sig} + 1) \cdot \mathcal{O}(k^3) \tag{1}$$

$$\epsilon = \frac{1}{(1 - \frac{1}{q_{sig}+1})^{q_{sig}+1}} \cdot q_{sig} \cdot \epsilon' \tag{2}$$

where q_{hash} and q_{sig} are the number of hash queries amd signature queries requested by the adversary.

Proof. Assume that there exists a (t, ϵ)-adversary $\mathcal{A}$ running in time of at most q_{hash} hash queries and at most q_{sig} signing queries against the ESF signature scheme which forge a valid signature with the probability of at least ϵ. We construct a simulator $\mathcal{S}$ that solves the ESF factoring problem with an advantage of at least ϵ' while interacting with $\mathcal{A}$.

Setup: $\mathcal{S}$ receives the ESF challenge $\{N, X_1, X_2, u, \gamma\}$ and must output $x^{v+\lambda}$. The $\mathcal{S}$ tries to find $x = f^{-1}(X_1 \cdot X_2)$ where f is defined in Equation (II-A). The private keys $\{v, \lambda, \omega, x\}$ are not known to $\mathcal{S}$.

Hash Query: $\mathcal{S}$ uses a counter i and initially set to zero. When $\mathcal{A}$ submits a fresh query on $H(I, m)$ for message m, $\mathcal{S}$ increments i such that $m_i = m$ and picks a random r_i in $\mathbb{Z}_N^*$. $\mathcal{S}$ then returns $h_i = r_i^u \mod N$ and $k_i = r_i^\gamma \mod N$ with probability p, and $h_i = x \cdot r_i^u \mod N$ and $k_i = x \cdot r_i^\gamma \mod N$ with probability $1 - p$.

Sign Query: When $\mathcal{A}$ makes a signing query for m, it has already requested the hash of m, therefore $m = m_i$ for some i. If $h_i = r_i^u \mod N$ and $k_i = r_i^\gamma \mod N$ then $\mathcal{S}$ returns r_i as the signature. Otherwise the process stops and the simulation aborts.

Forgery: $\mathcal{A}$ outputs a forgery (m, x). We make assumption that $\mathcal{A}$ has requested the hash of m before. If not, $\mathcal{S}$ goes ahead and makes the hash query itself such that $m = m_i$ for some i. Then if $h_i = x \cdot r_i^u \mod N$ and $k_i = x \cdot r_i^\gamma \mod N$, we have $X_1 \cdot X_2 = h_i^v \cdot k_i^\lambda = x^{v+\lambda} \cdot r_i \mod N$ then $\mathcal{S}$ output $x^{v+\lambda} = (X_1 \cdot X_2)/r_i \mod N$. Otherwise the process stops and the simulation has failed.

$\mathcal{S}$ answers a signature query with probability p; the probability that $\mathcal{S}$ answers to all signature queries is at least

$p^{q_{sig}}$. Then $\mathcal{S}$ outputs $x^{v+\lambda}$ for f with probability $1-p$. So $\mathcal{S}$ output a forgery $x^{v+\lambda}$ for f with probability at least $\alpha(p) = p^{q_{sig}} \cdot (1-p)$. The function $\alpha(p)$ to be maximum with

$$p_{max} = 1 - \frac{1}{q_{sig}+1}$$

and

$$\alpha(p_{max}) = \frac{1}{q_{sig}}(1 - \frac{1}{q_{sig}+1})^{q_{sig}+1}$$

. Then we obtain

$$\epsilon(k) = \frac{1}{(1-\frac{1}{q_{sig}+1})^{q_{sig}+1}} \cdot q_{sig} \cdot \epsilon'(k)$$

and if q_{sig} is large, $\epsilon(k) \approxeq exp(1) \cdot q_{sig} \cdot \epsilon'(k)$.

The running time of $\mathcal{S}$ is the running time of $\mathcal{A}$ added to the time needed to compute the h_i and k_i values. One ESF computation is equivalent to cubic time or better $\mathcal{O}(k^3)$ since modular exponentiation takes $\mathcal{O}(k^3)$ bit complexity. This gives the formula for t.

The percentage of error rates is 0.

□

V. Efficiency Analysis

We run 100 times key generation operation and each key generation operation, we run 100 times signing operation and 100 times verifying operation for the ESF signature scheme and FDH signature scheme.

The simulations are done in the platform Python 3.7 Intel Core i5-8250U 1.80GHz 4GB RAM. We fix the parameters p and q of both of the schemes to 1536 bit lengths. The public keys u and γ of our scheme are fixed to 30 bit lengths.

The results of coding, its consoles and the complete data are included in [8].

Table I
AVERAGE RUNNING TIME OF KEY GENERATION, SIGNING AND VERIFYING OPERATIONS BETWEEN FDH AND ESF SIGNATURE SCHEMES

	FDH Signature Scheme	ESF Signature Scheme
Average Time Taken of 100 Key Generation Operations	3.1273438s	3.7260937s
Average Time Taken of 100 Signing Operations	0.1138078s	0.0176016s
Average Time Taken of 100 Verifying Operations	0.1184797s	0.0147141s

From Table I, we obtain that the run time for the signing and verifying algorithms in our scheme are around 10 times faster than the FDH signature scheme since the public keys u and γ that used in the signing and verifying algorithms are fixed to 30 bit lengths.

VI. Conclusion

We propose a provable secure signature scheme that are constructed from the identification scheme based on the ESF assumption by utilizing FS transformation. Furthermore, we showed the proof of security of the ESF digital signature scheme in the ROM. Lastly, efficiency analysis is conducted towards our scheme and the result is compared with the FDH signature scheme.

Based on the result, we obtained that our signature scheme is performing faster in the signing and verifying algorithms than FDH signature scheme if the public keys u and γ in our scheme are fixed to small bit lengths, even though both of the schemes got the same bit lengths of parameters p and q.

Acknowledgment

The authors would like to thank the Ministry of Education of Malaysia in providing financial support for this work through the Fundamental Research Grant Scheme (FRGS/1/2019/ICT04/MMU/02/5).

References

[1] E. S. Ismail, M. Z. Zaharidan, and F. Samat, "ESF: Suatu kriptosistem mudah ringkas berasaskan masalah pemfaktoran," *Journal of Quality Measurement and Analysis JQMA*, vol. 14, no. 2, pp. 81–89, 2018.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, p. 120–126, Feb. 1978.

[3] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology — CRYPTO' 86*, A. M. Odlyzko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 186–194.

[4] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, ser. CCS '93. New York, NY, USA: Association for Computing Machinery, 1993, p. 62–73.

[5] M. Bellare and A. Palacio, "GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks," in *Advances in Cryptology — CRYPTO 2002*, M. Yung, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 162–177.

[6] J. Katz, *Digital signatures*. Springer Science & Business Media, 2010.

[7] J.-S. Coron, "On the exact security of full domain hash," in *Advances in Cryptology — CRYPTO 2000*, M. Bellare, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 229–235.

[8] S. Y. Hue, E. S. Ismail, N. H. Sarmin, and J. J. Chin. [Online]. Available: https://github.com/syhue/the-ESF-signature-scheme.