SOME VARIANTS OF THE SCHNORR SIGNATURE SCHEMA ON THE FINITE FIELD AND THE ELLIPTIC CURVE

Hong Dung Luu[1,*], *Minh Duc Tong*[1], *The Truyen Bui*[1]

doi:10.56651/lqdtu.jst.v11.n02.532.ict

Abstract

Schnorr signature schemas and the version on its elliptic curve (EC-Schnorr) are signature schemas that are not only highly secure but are also rated as having the highest performance among signature schemas based on discrete logarithm problems (on the F_p finite field and on the elliptic curve). The Schnorr and EC-Schnorr schemas have been incorporated into ISO/IEC 14888-3 and the BSI standard of the Federal Republic of Germany. Currently, the development of signature schemas based on Schnorr and EC-Schnorr schemas for BlockChain applications and digital currencies is of great interest. In this article the authors suggest several variations of the Schnorr schema on the finite field and the elliptic curve. The analyses in the article shows that the new schemas proposed here have the same security and performance as the Schnorr/EC-Schnorr signature schema in practical applications.

Index terms

Digital signature schema, Schnorr signature schema, discrete logarithm problem, elliptic curve cryptography.

1. Introduction

The Schnorr digital signature schema was proposed by C.P. Schnorr [1] from the development of the ElGamal signature schema [2]. Among the signature schemas constructed on discrete logarithms, Schnorr is a digital signature schema with many advantages: a) High performance: So far the Schnorr schema is still rated as the most efficient signature schema among the ElGamal family of signature schemas; b) High security: This schema has a strong and tight security proof, the research results in [3] have proven the security of the Schnorr schema in the Random Oracle model, the results in [4] proved the security of the Schnorr schema in the General Group model.

Currently, with the strong development of BlockChain technology and digital currencies in electronic transactions over the Internet, Schnorr-style digital signature schemas are very interested in researching for application in these systems.

[1]Faculty of Information Technology, Le Quy Don Technical University
[*]Corresponding author, email: luuhongdung@gmail.com

In this article, the authors propose some variations of the Schnorr signature schema based on the discrete logarithm problem on the finite field and on the elliptic curve defined on the finite field. The schemas proposed here have the same security and performance as the Schnorr signature schema (on the finite fields and on the elliptic curve) in practical applications.

2. The Schnorr signature schema

2.1. The finite field F_p based Schnorr signature schema

The finite field F_p based Schnorr signature schema was proposed by C.P. Schnorr in 1990 to accommodate applications that have an interaction between smart cards and terminals, where there are limitations in computing capacity, storage space, channel bandwidth, or energy consumption. The Schnorr signature schema is included in ISO/IEC 14888-3 [5] as SDSA (Schnorr Digital Signature Algorithm).

a) Parameter and key generating algorithm

- Domain parameters:
  The domain parameter set for SDSA in accordance with ISO/IEC 14888-3 includes:
  + p is a prime number, where: $2^{\alpha-1} < p < 2^{\alpha}$.
  + q is a prime divisor of $p-1$, where: $2^{\beta-1} < p < 2^{\beta}$.
  + g is a generator of a subgroup of order q in the multiplicative group of $GF(p)$, such that $1 < g < p$.

  Four choices for the α and hash function (α, H) are allowed in SDSA, namely: (1024, SHA-1), (2048, SHA-224), (2048, SHA-256) and (3072, SHA-256).
  Corresponding β should be selected according to α in section 5.1.3.1, table 1 of ISO/IEC 14888-3.
  The parameters p, q and g are generated as specified in ISO/IEC 14888-3. The parameters p and q can be generated using the prime generation techniques given in ISO/IEC 18032 [5].
- Key Pair Generation:
  The secret key of a signing entity is a secretly generated random or pseudo-random integer x such that $0 < x < q$. The corresponding public key y is:

$$y = g^{-x} \mod p$$

  The p, q, g parameters are system parameters or domain parameters created by the service provider and (x, y) are the secret, public key pair of the signer. The parameter and key generating algorithm (Algorithm 1) is described as follows:

Algorithm 1:

Input: L_p, L_q.
Output: p, q, g, H, x, y.

1: **generate:** p, q : len$(p) = L_p$, len$(q) = L_q, q|(p-1)$
2: **select** δ : $1 < \delta < p$
3: $g = \delta^{\frac{p-1}{q}} \mod p$
4: **if** $((g = 1))$ **then**
goto 2.
5: **end if**
6: $y = g^{-x} \mod p$
7: **return** $\{p, q, g, H, x, y\}$

Comment:

- L_p, L_q: length (in bits) of prime numbers p, q.
- len(.): the function calculates the length (in bits) of an integer.

b) Signing algorithm

The signing algorithm of the Schnorr schema with input data are system parameters (p, q, g), secret key of the signer (x) and message to sign (M), giving the output as the signature (e, s). The algorithm is described as follows:

Algorithm 2:

Input: p, q, g, H, x, M.
Output: (e, s).

1: $k = RGN(\{1, 2, \ldots, q-1\}$
2: $r = g^k \mod p$
3: $e = H(r \| M)$
4: **if** $(e = 0)$ **then**
goto 1
5: **end if**
6: $s = (k + x \times e) \mod q$
7: **if** $(s = 0)$ **then**
goto 1.
8: **end if**
9: **return** (e, s).

Comment:

- $RGN(.)$: random/pseudo-random number generator.

c) Verification algorithm

The verification algorithm with input data are system parameters (p, q, g), signatures (e, s), public key of the signer (y) and message to be verified (M). If the return result is *TRUE*, then the integrity and origin of M is confirmed. Conversely, if the result is *FALSE*, M is denied of origin and integrity. The verification algorithm of the schema (Algorithm 3) is described as follows:

Algorithm 3:

Input: $p, q, g, H, y, M, (e, s)$.
Output: *TRUE / FALSE.*

1: **if** $(e = 0$ **or** $s = 0)$ **then**
    return (*FALSE*)
2: **end if**
3: $u = g^s \times y^e \mod p$
4: $v = H(u \| M)$
5: **if** $(v = e)$ **then**
    return (*TRUE*).
6: **else**
    return (*FALSE*).
7: **end if**

d) The performance of the Schnorr signature schema The performance of the digital signature schema can be assessed by the computational cost, it is the number of calculations to be performed of the schema, here the convention uses the symbols:
N_{exp} : the number of modulo exponentiations,
N_{mul} : the number of modulo multiplications,
N_{inv} : the number of modulo division (inversion), N_h : the number of one-way or key-ed hash operations.
The computational cost of the signing algorithm and verification algorithm of the Schnorr schema compared to the DSA schemas in the U.S. DSS signature standard [6] or GOST R34.10–94 of the Russian Federation [7] is indicated in table 1 and table 2 as follows:

Table 1. The computational cost of the signing algorithms

	N_{exp}	N_{mul}	N_{inv}	N_h
DSA	1	2	1	1
GOST R34-10.94	1	2	0	1
Schnorr	1	1	0	1

Table 2. The computational cost of the verification algorithms

	N_{exp}	N_{mul}	N_{inv}	N_h
DSA	2	3	1	1
GOST R34-10.94	3	3	0	0
Schnorr	2	1	0	1

Results from table 1 and table 2 show that the performance efficiency of the Schnorr schema is the highest among the schemas compared.
Remark:
- Parameter and key generation algorithm only need to be done once with every schema. Therefore, the computational cost of parameter and key generation algorithm can be ignored when comparing the performance of schemas.

2.2. The elliptic curve based Schnorr signature schema

The elliptic curve based Schnorr signature schema is included in the German Federal Republic's digital signature standard in BSI [8] under the name EC-Schnorr and is also included in ISO/IEC 14888-3 [5] as EC-SDSA (Elliptic Curve Schnorr Digital Signature Algorithm).

a) Parameter and key generating algorithm

- Domain parameters:
  The set of domain parameters includes:
  + p is a prime number specifying the underlying finite field F_p.
  + $E(F_p)$ is elliptic curve $E(a, b)$ defined on the finite field F_p by equation:

$$y^2 = x^3 + ax + b$$

    with $a, b \in F_p$ and satisfied $4a^3 + 27b^2 \neq 0 \mod q$.
  + G is a point of order q on the elliptic curve $E(F_p)$, called the base point in $E(F_p)$.
  + q is the order of G in $E(F_p)$.

  EC-Schnorr's domain parameters are generated as specified in BSI [8], while EC-SDSA's parameters are generated as specified in ISO/IEC 14888-3 [9].
- Key Pair Generation:
  The secret key of the signature entity is a random or pseudo-random integer d that is secretly created so that $0 < d < q$. The corresponding public key P is: $P = d.G$
  The p, a, b, G, q parameters are system parameters or domain parameters generated by the service provider and (d, P) are the secret, public key pair of the signer. The key generating algorithm is described as follows:

Algorithm 4:

Input: $E(F - p) = (p, a, b, G, q)$.
Output: (d, P)

1: $d = RNG(\{1, 2, \ldots, q - 1\})$
2: $P = (x_P, y_P) = d.G$
return (d, P).

b) Signing algorithm

The signing algorithm of the EC-Schnorr schema with inputs is system parameters (p, a, b, G, q, H), the signer's secret key (d) and the message to be signed (M), giving the output as a signature (r, s). The algorithm is described as follows:

Algorithm 5:

Input: $E(F_p) = (p, a, b, G, q), H, d, M.$
Output: (r, s)

1: $k = RNG(\{1, 2, \ldots, q-1\})$
2: $R = (x_R, y_R) = k.G$
3: $r = H(M \| x_R)$
If $(r = 0 \mod q)$ **goto** 1.
4: $s = (k - r \times d) \mod q$
If $(s = 0 \mod q)$ **goto** 1.
return (r, s).

c) Verification algorithm

The verification algorithm with input data being system parameters (p, a, b, G, q, H), signature (r, s) and the public key of the signer (P) and the message to be verified (M). If the return result is *TRUE*, then the integrity and origin of M is confirmed. Conversely, if the result is *FALSE*, M is denied of origin and integrity. The verification algorithm of the schema (Algorithm 6) is described as follows:

Algorithm 6:

Input: $E(F_p) = (p, a, b, G, q), H, P, M, (r, s).$
Output: *TRUE / FALSE*

1: **if** $(r = 0 \mod q$ or $s = 0 \mod q)$ **return** (*FALSE*)
2: $Q = (x_Q, y_Q) = s.G + r.P$
3: $v = H(M \| x_Q)$
4: **If** $(v = r)$ **then return** *TRUE*
5: **else return** *FALSE.*

d) The performance of the EC-Schnorr signature schema

Compare the computational cost of the signature algorithm and the verification algorithm of the EC-Schnorr schema with the ECDSA schema in the U.S. DSS signature standard [6] or GOST R34.10–2012 of the Russian Federation [10] indicated in table 3 and table 4 are as follows:

Table 3. The computational cost of the signature algorithms

	N_{mp}	N_{mul}	N_{inv}	N_h
EC DSA	1	2	1	1
GOST R34.10-2012	1	2	0	1
EC-Schnorr	1	1	0	1

Table 4. The computational cost of the verification algorithms

	N_{mp}	N_{mul}	N_{inv}	N_h
EC DSA	2	2	1	1
GOST R34.10-2012	2	2	1	1
EC-Schnorr	2	0	0	1

where:
N_{mp} : the number of multiplications on $E(F_p)$,
N_{mul} : the number of modulo multiplications,
N_{inv} : the number of modulo division (inversion),
N_h : the number of one-way or key-ed hash operations.

3. Variants of the Schnorr signature schema

3.1. The variant of Schnorr signature schema based on finite field F_p

The variant of the Schnorr signature schema based on finite field F_p (MTA V22-1) includes a parameter-key generation algorithm, a signing algorithm, and a verification algorithm as follows:

a) Parameter and key generating algorithm
The MTA V22-1's domain parameters here can be created as prescribed in ISO/IEC 14888-3 [9] or FIPS 186-4 [6] or GOST R34.10-94 [7]. The parameter and key generating algorithm (Algorithm 7) of the proposed schema is described as follows:

Algorithm 7:

Input: L_p, L_q.
Output: p, q, g, H, x, y.

1: **generate** $p, q : \text{len}(p) = L_p, \text{len}(q) = L_q, q|(p-1)$
2: **select** $\alpha : 1 < \alpha < p$
3: $g = \alpha^{\frac{p-1}{q}} \mod p$.
4: **If** $(g = 1)$ **then goto** 2
5: **select** $x : 1 < x < q$.
6: $y = g^{x^{-1}} \mod p$.
7: **If** $(y = 1)$ **then goto** 4
return (p, q, g, H, x, y)

b) Signing algorithm
The signing algorithm (Algorithm 8) of the proposed schema is described as follows:

Algorithm 8:

Input: p, q, g, H, x, M.
Output: (e, s)

1: $k = RNG(\{1, 2, \ldots, q-1\})$
2: $r = g^k \mod p$
3: $e = H(r \| M)$.
4: **If** $(e = 0)$ **then goto** 1
5: $s = x \times (k - e) \mod q$.
6: **If** $(s = 0)$ **then goto** 1
return (e, s).

c) Verification algorithm

The verification algorithm (Algorithm 9) of the proposed schema is described as follows:

Algorithm 9:

Input: $p, g, H, y, M, (e, s)$.
Output: *TRUE/FALSE*

1: **if** $(e = 0$ **or** $s = 0)$ **return** (*FALSE*)
2: $u = y^s \times g^e \mod p$
3: $v = H(u \| M)$.
4: **if** $(v = e)$ **then return** (*TRUE*)
5: **else return** *(FALSE)*

d) The correctness of the proposed new schema

We have:

$$\begin{aligned} u &= y^s \times g^e \mod p \\ &= \left(g^{x^{-1}} \mod p\right)^{x(k-e)} \times g^e \mod p \\ &= g^{x^{-1}x(k-e)} \times g^e \mod p \\ &= g^{k-e+e} \mod p \\ &= g^k \mod p = r \end{aligned}$$

Here's what needs to be proved:

$$v = H(u \| M) = H(r \| M) = e.$$

3.2. The variant of Schnorr signature schema based on the elliptic curve

The variant of Schnorr signature schema based on the elliptic curve (MTA V22-2) includes a parameter and key generating algorithm, a signing algorithm and a verification algorithm as follows:

a) Parameter and key generating algorithm

- Domain parameters
  The set of domain parameters includes:
  + p is a prime number specifying the underlying finite field F_p.
  + $E(F_p)$ is elliptic curve defined on the finite field F_p by equation $E(a, b)$:

$$y^2 = x^3 + ax + b$$

  with: $a, b \in F_p$ and satisfied: $4a^3 + 27b^2 \neq 0 \mod q$.
  + G is the base point in $E(F_p)$.
  + q is the order of G in $E(F_p)$.

  MTA V22-2's domain parameters here can be generated as specified in ISO/IEC 14888-3 [9], BSI [8], FIPS 186 – 4 [6] or GOST R34.10–2012 [10].

- Key Pair Generation
  The secret key of the signature entity is a random or pseudo-random integer d that is secretly created so that $0 < d < q$. The corresponding public key P is:

$$P = (d^{-1} \mod q).G$$

  The p, a, b, G, q parameters are system parameters or domain parameters generated by the service provider and (d, P) are the secret, public key pair of the signer. The key generating algorithm is described as follows:

Algorithm 10:

Input: $E(F_p) = (p, a, b, G, q)$.
Output: (d, P).

1: $d = RNG(\{1, 2 \ldots, q-1\})$
2: $P = (x_P, y_P) = (d^{-1} \mod q).G$
return (d, P).

b) Signing algorithm

The signing algorithm of the schema with input data is the system parameters (p, a, b, G, q, H), the secret key of the signer (d) and the newsletter to sign (M), giving the output as the signature (r, s). The algorithm is described as follows:

Algorithm 11:

Input: $E(F_p) = (p, a, b, G, q), H, d, M$.
Output: (r, s).

1: $k = RNG(\{1, 2 \ldots, q-1\})$
2: $R = (x_R, y_R) = k.G$
3: $r = H(M \| x_R)$
4: **if** $(r = 0 \mod q)$ **then**
goto 1
5: **end if**
6: $s = d \times (k - r) \mod q$
if $s = 0 \mod q$ **goto** 1
return (r, s).

c) Verification algorithm

The verification algorithm with input data being system parameters (p, a, b, G, q, H), signature (r, s) and the public key of the signer (P) and the newsletter to be verified (M). If the return result is *TRUE*, then the integrity and origin of M is confirmed. Conversely, if the result is *FALSE*, M is denied of origin and integrity. The verification algorithm of the schema (Algorithm 12) is described as follows:

Algorithm 12:

Input: $E(F_p) = (p, a, b, G, q), H, P, M, (r, s)$.
Output: *TRUE/FALSE.*

1: **if** $(r = 0 \mod q$ or $s = 0 \mod q)$ **then return** *FALSE*
2: $Q = (x_Q, y_Q) = r.G + s.P$
3: $r = H(M \| x_Q)$
if $v = r$ **then return** *TRUE*
else return *FALSE*

d) The correctness of the proposed schema
We have:

$$Q = r \cdot G + s \cdot P = r \cdot G + d \cdot (k - r) \cdot d^{-1} \cdot G = r \cdot G + d \cdot d^{-1} \cdot (k - r) \cdot G$$
$$= r \cdot G + (1 + n \cdot q) \cdot (k - r) \cdot G = r \cdot G + (k - r) \cdot G + n \cdot (k - r) \cdot (q \cdot G)$$
$$= r \cdot G - r \cdot G + k \cdot G + n \cdot (k - r) \cdot O_E = k \cdot G + O_E = R + O_E = R$$

Inferring: $x_Q = x_R$.
From here, we have something to prove: $v = H(M \| x_Q) = H(M \| x_R) = r$.
Notes:

+ O_E: the infinity point of $E(F_p)$ and is the unit element for the group operator on $E(F_p)$.
+ n: a positive integer.

3.3. The performance of the proposed schemas

The performance of the MTA V22-1 schema here is assessed by comparing the computational cost of this schema with the computing costs of the Schnorr, DSA digital signature schemas in the DSS standard of the United States and GOST R34.10-94 of the Russian Federation.

Table 5 shows that the computational cost of the signature algorithm of the proposed schema (MTA V22-1) is equivalent to the computational cost of the Schnorr schema and lower than the DSA and GOST R34.10-94 schemas. Table 6 also shows that the computational cost of the MTA V22-1 schema is equivalent to the computational cost of the Schnorr schema and lower than the rest of the schemas. Summarizing the results from table 5 and table 6 shows that the performance efficiency of MTA V22-1 is comparable to the Schnorr signature schema and higher than the DSA and GOST R34.10-94 schemas.

Table 5. The computational cost of the signature algorithms

	T_{exp}	T_{mul}	T_{inv}	T_h
DSA	1	2	1	1
GOST R34-10.94	1	2	0	1
Schnorr	1	1	0	1
MTA V22-1	1	1	0	1

Table 6. The computational cost of the verification algorithms

	N_{exp}	N_{mul}	N_{inv}	N_h
DSA	2	3	1	1
GOST R34-10.94	3	3	0	1
Schnorr	2	1	0	1
MTA V22-1	2	1	0	1

Compare the computational cost of the signature and verification algorithms of the MTA V22–2 schema with the EC-Schnorr, ECDSA and GOST R34.10-2012 schemas indicated in table 7 and table 8 as follows:

Table 7. The computational cost of the signature algorithms

	T_{mp}	T_{mul}	T_{inv}	T_h
EC DSA	1	2	1	1
GOST R34.10–2012	1	2	0	1
EC Schnorr	1	1	0	1
MTA V22–2	1	1	0	1

Table 8. The computational cost of the verification algorithms

	N_{mp}	N_{mul}	N_{inv}	N_h
EC DSA	2	2	1	1
GOST R34.10–2012	2	2	1	1
EC Schnorr	2	0	0	1
MTA V22–2	2	0	0	1

From the results of tables 7 and 8 it can be seen that the performance of the proposed schema is equivalent to the EC-Schnorr schema and higher than the ECDSA and GOST R34.10-2012 schemas.

3.4. The security level of the proposed schemas

From the structural characteristics of the proposed schemas versus the Schnorr and EC-Schnorr schemas, showing the security assessment methods for the Schnorr signature schema [3], [4], can be applied in a completely similar way to the proposed schemas and the results received of the evaluation of the schemas are the same when applying the same method is affirmable.

a) The security of Schnorr schema variants in the Random Oracle Model
In the following, the security of Schnorr schema variants in the random oracle model is considered in two extreme situations, the no-message attack and the adaptively chosen-message attack.

- **No-Message Attacks** Based on the Forking Lemma (Theorem 1) [3]:

Theorem 1 (The Forking Lemma)**.** *Let $(G, \sum, V)$ be a generic digital signature schema with security parameter k. Let A be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote by Q the number of queries that A can ask to the random oracle. Assume that, within time bound T, A produces, with probability $\varepsilon \geq 7Q/2^k$, a valid signature $(m, \delta_1, h, \delta_2)$. Then there is another machine which controlled over A and produces two valid signatures $(m, \delta_1, h, \delta_2)$ and $(m, \delta_1, h', \delta_2')$ such that $h \neq h'$, in expected time $T' \leq 84480TQ/\varepsilon$.*

Apply to the MTA V22-1 schema, for any security parameter k, an authority chooses two large prime integers p and q, such that $2^{k-1} \leq q < 2^k$ holds and q divides $(p-1)$ as well as an element g from of $\mathbb{Z}_p^*$ order q. The triple (p, q, g) is published together with a public hash function f_H whose output domain is identified to $\mathbb{Z}_p^*$. The security parameter k is then equal to $\lceil \log q \rceil$, where as the size of the public key, denoted by n, is equal to $\lceil \log p \rceil$. Furthermore, we assume that $k >> \log n$. Any user randomly chooses his secret key x in $\mathbb{Z}_p^*$, and publishes $y = g^{x^{-1}} \mod p$. In order to sign a message m, the user chooses a random element k in $\mathbb{Z}_p^*$ and computes the commitment $r = g^k \mod p$. He gets the challenge $e = f_H(m, r)$ and computes $s = x \times (k - e) \mod q$. The signature is the triple (r, e, s), which satisfies the tests $r \stackrel{?}{=}$ and $e \stackrel{?}{=} f_H(m, r)$.

Theorem 2. *Assume that, within a time bound T, an attacker A performs an existential forgery under a no-message attack against the MTA V22-1 signature, with probability $\varepsilon \geq 7Q/q$. We denote by Q the number of queries that A can ask to the random oracle. Then the discrete logarithm in subgroups of prime order can be solved in expected time less than $84480TQ/\varepsilon$.*

Proof: Similar to the Schnorr schema, this schema satisfies all the required properties of a generic signature schema [3]. From the Forking Lemma (Theorem 1), after a polynomial replay of the attacker A, we obtain two valid signatures (m, r, e, s) and (m, r, e', s') with $e \neq e'$. Then we have the following equalities $r = g^e y^s \mod p$ and $r = g^{e'} y^{s'} \mod p$, from which we obtain the discrete logarithm $\log_g y = (e - e')/(s - s') \mod q$. ■

- **Adaptively Chosen-Message Attacks**

  Based on the Forking Lemma (Theorem 3) [3]:

  Theorem 3 (The Forking Lemma)**.** *Let A be a probabilistic polynomial time Turing machine whose input only consists of public data. We denote respectively by Q and R the number of queries that A can ask to the random oracle and the number of queries that A can ask to the signer. Assume that, within a time bound T, A produces, with probability $\varepsilon \geq 10(R+1)(R+Q)/2^k$, a valid signature $(m, \delta_1, h, \delta_2)$. If the triples (δ_1, h, δ_2) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which controlled over the machine obtained from A replacing interaction with the signer by simulation and produces two valid signatures $(m, \delta_1, h, \delta_2)$ and $(m, \delta_1, h', \delta_2')$ such that $h \neq h'$ in expected time $T' \leq 84480TQ/\varepsilon$.*

Apply to the MTA V22-1 signature schema, we have:

Theorem 4. *Let A be an attacker which performs, within a time bound T, an existential forgery under an adaptively chosen-message attack against the MTA V22-1 signature schema, with probability ε. We denote respectively by Q and R the number of queries that A can ask to the random oracle and the number of queries that A can ask to the signing oracle. Assume that $\varepsilon \geq 10(R+1)(R+Q)/q$, then the discrete logarithm in subgroups of prime order can be solved within expected time less than $120686QT/\varepsilon$.*

Proof: The collusion of the attacker A and the simulator S defines a machine B which performs a no-message attack. An execution of B is successful if it outputs a forgery, and if there is no collisions of queries to the random oracle during the process. Then, within a time bound T, B has a probability of success greater $7\varepsilon/10 \geq 7Q/2^k$. Using Theorem 1, within an expected number of steps bounded by $84480Q/(7\varepsilon/10)$, one can provide two valid signatures. We only have to prove that the triples (r, e, s) produced by the signer and the random oracle can be simulated without the knowledge of the signer's secret. Once this is done, the result directly follows from Theorem 3, using the same proof as for Theorem 2. ■

b) The security of Schnorr schema variants in the Generic Group Model

In [4], the security (in initialization) of the Schnorr signature schema, or its elliptic curve variant, with hash functions such as SHA-1 and MD5 is analyzed in another popular idealisation, the generic group model (G) [11]. The article authors presented two real-world hash function properties, called random-prex preimage (rpp) and random-prex second-preimage (rpsp) resistance, and showed that they are at the same time necessary conditions in the standard model and sufficient conditions in the generic group model for the security of Schnorr signature schema. The rpp property is in fact equivalent to the Nostradamus attack of Kelsey and Kohno [12]. When considering a particular implementation of the hash function based on keyed compression functions, those properties become equivalent to the ePre and eSec notions in the framework of Rogaway and Shrimpton [13]. The results in [4] show that, the only way to break Schnorr signature schema is by breaking either the **rpp** problem or the **rpsp** problem associated to the hash function. In particular, it warrants the secure use of Schnorr signature schema when implemented with SHA-1/SHA-256 or MD5, as long as the **rpp** and **rpsp** problems are still believed to be hard for the respective hash functions. Using the general Forking lemma of Bellare and Neven [14] for variants of the Schnorr signature schema (MTA V22-1 schema and MTA V22-2 schema – MTA signature schemas), we can obtain their specific security limitations as follows:

Theorem 5. *If the discrete logarithm problem in G is $(t_{d\log}, \varepsilon_{d\log})$ – hard, then the MTA signature schemas is $(t_{uf-cma}, q_S, q_H, \varepsilon_{uf-cma})$ – secure for*

$$\varepsilon_{uf-cma} = \sqrt{(q_H + q_S + 1) \times \varepsilon_{d\log}} + \frac{q_H + q_S + 1}{2^n} + \frac{q_S(q_H + q_S + 1)}{q}$$

and $t_{uf-cma} = t_{d\log}/2 - q_S \times t_{exp} + O(q_H + q_S + 1)$*, where* t_{exp} *is the cost of an exponentiation in the group* G*.*

This bound clearly indicates that a hash function with $n = s/2$ output bits should be sufficient to obtain a security level of $s/2$ bits, conforming to result that H need only be **rpp** and **rpsp**-secure, and not collision resistant. Based on the results for the Schnorr signature schema in [4], apply in the same way to the MTA schemas, we also get:

Theorem 6. *Let* $H : \{0,1\}^* \to \{0,1\}^n \to \mathbb{Z}_p$ *be some hash function, and let* G *be some fixed group modelled as a generic group over the set of bit strings* G*. Let* $f : G \to \{0,1\}^d$ *be an almost-invertible conversion function. If both the* $rpp[D]$ *and the* $rpsp[D]$ *are hard for* H*, with respect to the domain* $D = \{0,1\}^d$*, then the MTA signature schemas are secure in the generic group model.*

The above theorem shows that MTA signature schemas are secure as long as the hash function used in its construction satisfies the security notions that put forth in [4].

4. Conclusion

In this article, the authors suggest two variants of the Schnorr signature schema based on the finite field and the elliptic curve. These variations are not new in construction methods compared to the original Schnorr schema. However, as the analysis has shown, the performance and security of these schemas are completely equivalent to the Schnorr signature schema, so the proposed schemas here are capable of widespread and effective application in practice similar to the Schnorr schema.

References

[1] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, pp. 161–174, 1991.

[2] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[3] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of cryptology*, vol. 13, no. 3, pp. 361–396, 2000.

[4] G. Neven, N. P. Smart, and B. Warinschi, "Hash function requirements for Schnorr signatures," *Journal of Mathematical Cryptology*, vol. 3, no. 1, pp. 69–87, 2009.

[5] "Information technology – security techniques – prime number generation. first edition." International Organization for Standardization, Standard, 1 2005.

[6] C. F. Kerry and P. D. Gallagher, "Digital signature standard (dss)," *FIPS PUB*, pp. 186–4, 2013.

[7] "Cryptographic data security produce and check procedures of electronic digital signature based on asymmetric cryptographic algorithm, government committee of the russia for standards." Russian Federation Standard Information Technology, Standard, 1994.

[8] "Technical guideline BSI tr-03111: Elliptic curve cryptography version 2.1, Germany." Federal Office for Information Security, Standard, 6 2018.

[9] "Information technology – security techniques – digital signatures with appendix. second edition." International Organization for Standardization, Standard, 11 2006.

[10] V. Dolmatov and A. Degtyarev, "GOST R 34.10-2012: Digital Signature Algorithm," RFC 7091, Tech. Rep. 7091, Dec. 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc7091

[11] V. Shoup, "Lower bounds for discrete logarithms and related problems," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1997, pp. 256–266.
[12] J. Kelsey and T. Kohno, "Herding hash functions and the nostradamus attack," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 183–200.
[13] P. Rogaway and T. Shrimpton, "Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance," in *International workshop on fast software encryption*. Springer, 2004, pp. 371–388.
[14] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 390–399.

Manuscript received 26-03-2022; Accepted 27-08-2022.

Hong Dung Luu graduated in Electronics and Communications from Le Quy Don Technical University in 1989, PhD at Le Quy Don Technical University in 2013; Currently working in the IT department - Le Quy Don Technical University; Research direction: Cryptography and information security. E-mail: luuhongdung@gmail.com

Minh Duc Tong graduated from Le Quy Don Technical University in 2000. Received a doctorate from University of Electrical Engineering - Russia in 2007. Currently, he is a lecturer in the Faculty of Information Technology - Le Quy Don Technical University. Research field: Image processing, object identification, information security safety. E-mail: ducmta@gmail.com

The Truyen Bui graduated from Le Quy Don Technical University in 2000. He received a doctor's degree in analysis and information processing at Moscow Aviation Institute, Russia in 2008. Currently, he is a lecturer at the Le Quy Don Technical University. His research interests are virtual reality simulation and information security. E-mail: truyenbuithe@lqdtu.edu.vn

MỘT SỐ BIẾN THỂ CỦA LƯỢC ĐỒ CHỮ KÝ SCHNORR TRÊN TRƯỜNG HỮU HẠN VÀ ĐƯỜNG CONG ELLIPTIC

Lưu Hồng Dũng, *Tống Minh Đức*, *Bùi Thế Truyền*

Tóm tắt

Lược đồ chữ ký Schnorr và phiên bản trên đường cong elip của nó (EC-Schnorr) là lược đồ chữ ký không chỉ có tính bảo mật cao mà còn được đánh giá là có hiệu năng cao nhất trong số các lược đồ chữ ký dựa trên bài toán logarit rời rạc (trên trường hữu hạn F_p và trên đường cong elip). Lược đồ Schnorr và EC-Schnorr đã được công bố tại ISO/IEC 14888-3 và tiêu chuẩn BSI của Cộng hòa Liên bang Đức. Hiện nay, việc phát triển các sơ đồ chữ ký dựa trên các sơ đồ Schnorr và EC-Schnorr cho các ứng dụng BlockChain và tiền kỹ thuật số đang nhận được sự quan tâm rất lớn. Trong bài báo này, các tác giả đề xuất một số biến thể của lược đồ Schnorr trên trường hữu hạn và đường cong elliptic. Các phân tích trong bài báo cho thấy rằng các lược đồ mới được đề xuất ở đây có cùng tính bảo mật và hiệu suất như lược đồ chữ ký Schnorr/EC-Schnorr trong các ứng dụng thực tế.

Từ khóa

Lược đồ chữ ký số, lược đồ chữ ký Schnorr, bài toán logarit rời rạc, mật mã dựa trên đường cong elliptic.