A Study of Non-Abelian Public Key Cryptography

Tzu-Chun Lin
Department of Applied Mathematics, Feng Chia University
100, Wenhwa Road, Taichung 40724, Taiwan, R.O.C.
(Email: lintc@fcu.edu.tw)
(Received Feb. 12, 2017; revised and accepted June 12, 2017)

Abstract

Nonabelian group-based public key cryptography is a relatively new and exciting research field. Rapidly increasing computing power and the futurity quantum computers [52] that have since led to, the security of public key cryptosystems in use today, will be questioned. Research in new cryptographic methods is also imperative. Research on nonabelian group-based cryptosystems will become one of contemporary research priorities. Many innovative ideas for them have been presented for the past two decades, and many corresponding problems remain to be resolved. The purpose of this paper, is to present a survey of the nonabelian group-based public key cryptosystems with the corresponding problems of security. We hope that readers can grasp the trend that is examined in this study.

Keywords: Conjugacy Search Problem; Nonabelian Groups; Public Key Cryptography

1 Introduction

The development of public key cryptography was a revolutionary concept that emerged during the twentieth century. The first published study on public key cryptography was a key agreement scheme that was described by W. Diffie and M.E. Hellman in 1976 [19]. The most common public key cryptography presently in use, such as the Diffie-Hellman cryptosystem, the RSA cryptosystem, the ElGamal cryptosystem and the elliptic curve cryptosystem are number theory based and hence depend on the structure of abelian groups. Their security depends on difficulties regarding resolving some hard problems of the number theory. For instance, the RSA algorithm depends on integer factorization problem. The Diffie-Hellman, ElGamal and ECC algorithms also depend on discrete logarithmic problems (DLP). Although there have not been any successful attacks on the above public key cryptosystems the security of public key cryptosystems in use today, will be questioned due to rapidly increasing computing power and the futurity quantum computers. In 1997 [52], P.W. Shor pointed out that there are polynomial-time algorithms for solving the factorization and discrete logarithmic problems based on abelian groups during the functions of a quantum computer. Research in new cryptographic methods is also imperative, as research on nonabelian group-based cryptosystems will be one of new research priorities. In fact, the pioneering work for nonabelian group-based public key cryptosystem was proposed by N. R. Wagner and M. R. Magyarik [61] in 1985. Their idea just is not suitable for practical applications. For nearly two decades, numerous nonabelian groups have been discussed to design efficient cryptographic systems. The most frequently discussed nonabelian settings include matrix groups, braid groups, semidirect products, logarithmic signatures and algebraic erasers.

In this paper, we give an overview of known public key cryptography designed by the above mentioned nonabelian groups. These proposed nonabelian group-based public key cryptosystems rely on either encryption-decryption or on key exchange agreement. A standard model for a public key cryptographic scheme is phrased as two parties, which are referred to as Alice and Bob. Suppose that Alice wants to send a message M to Bob. A general model of encryption scheme is the following. Alice uses the encryption map f_{k_1} to encrypt the message $C = f_{k_1}(M)$, where f_{k_1} is a one-way function and is public. After receiving the cipher C, Bob uses the corresponding decryption map g_{k_2} to decode $g_{k_2}(f_{k_1}(M)) = M$, where g_{k_2} should be known only by Bob.

Many non-abelian group-based key establishment protocols are related to the Diffie-Hellman (DH) protocol, and we therefore provide a brief description of the DH-protocol. The Diffie-Hellman (DH) protocol functions as follows: Let G be a cyclic group with a generator g. Suppose that Alice and Bob want to generate a shared secret key K. Alice then randomly selects an integer $1 < a < o(g)$ and sends $A := g^a$ to Bob. Similarly, Bob randomly selects an integer $1 < b < o(g)$ and sends $B := g^b$ to Alice. Alice computes $K = B^a$, while Bob computes $K = A^b$. The security of the DH-protocol relies on the Diffie-Hellman problem (or the Discrete Logarithmic Problem).

Problem 1. *(Diffie-Hellman Problem) Let G be a group. If $g, g^x, g^y \in G$ are known, find the value of g^{xy}.*

Problem 2. *(Discrete Logarithmic Problem) Let G be a group. If $h, g \in G$ such that $h = g^x$ and h, g are known. Find the integer x.*

Problem 3. *(Conjugacy Search Problem) Let G be a nonabelian group. Let $g, h \in G$ be known such that $h = g^x$ for some $x \in G$. Find x. Here g^x stands for $x^{-1}gx$.*

Nonabelian group-based public key cryptography is a relative new research field. In contrast to abelian groups the conjugacy search problem and its variant versions are hard problems on some nonabelian groups. The conjugacy search problem and its variant versions play an important role for the security on nonabelian group-based public key cryptography.

In this paper, we give a survey of the representative nonabelian group-based public key cryptosystems so far. Their algorithms are very different.

2 Matrix Groups

2.1 Yamamura's Encryption Scheme

At PKC'98, A. Yamamura [64] proposed a public key encryption scheme based on the modular group $SL(2, \mathbb{Z})$. It is well known that $SL(2, \mathbb{Z})$ is generated by two matrices $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ and $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, where the orders of both generators are $o(S) = 4$ and $o(T) = \infty$ and the matrix $ST = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}$ is of order 6. Also, $SL(2, \mathbb{Z})$ is generated by the matrices S and ST subject to the relations $S^4 = (ST)^4 = I$ and $(ST)^3 = S^2$. For a matrix $N \in SL(2, \mathbb{Z})$, the matrices $A := N^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix} N$ and $B := N^{-1} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} N$ satisfy the relations $A^6 = B^4 = I$ and $A^3 = B^2$. Therefore, the matrices A and B generate $SL(2, \mathbb{Z})$.

1) Key Generation: Bob
   a. chooses two matrices $V_1 := (BA)^i$ and $V_2 := (BA^2)^j \in SL(2, \mathbb{Z})$ for some $i, j \in \mathbb{N}$.
   b. chooses matrices $M \in GL_2(\mathbb{C})$ and $F_1(X), F_2(X) \in Mat_2(\mathbb{C}[X])$ and $a \in \mathbb{C}$ such that $F_1(a) = V_1$ and $F_2(a) = V_2$.
   c. computes $W_1(X) := M^{-1}F_1(X)M$ and $W_2(X) := M^{-1}F_2(X)M$.
   d. Bob's public key: $W_1(X), W_2(X)$.
   Bob's private key: M, a.

2) Encryption: Let $b_1 \cdots b_n \in \{0,1\}^n$ be the message. Alice computes the ciphertext
$$C(X) := W_2(X) \prod_{i=1}^{n} (W_1(X)^{b_i+1} W_2(X)).$$

3) Decryption: From the ciphertext $C(X)$ and Bob's private key (M, a) the message $b_1 \cdots b_n$ can be recovered by means of a procedure described in [64].

4) Security Analysis:
The protocol is based on conjugacy search problem and root problems. But, R. Steinwandt [55] in 1992) pointed out that the Yamamura's Encryption Scheme is insecure. Suppose that an adversary Eve intercepted to the cipher $C(X)$. She can compute
$$D(X) := W_2(X)^{-1}C(X) = \prod_{i=1}^{n} (W_1(X)^{b_i+1} W_2(X)).$$
The entries of the matrix
$$((W_1(X)^{b_i+1} W_2(X))^{-1} D(X)$$
should be polynomials over $\mathbb{C}$. Beginning with the first bit b_1, if at least one of the entries of $D_1 := ((W_1(X)^2 W_2(X))^{-1} D(X)$ involves a non-constant denominator then we can conclude $b_1 = 0$; otherwise $b_1 = 1$. Similarly, if the matrix $D_2 := ((W_1(X)^2 W_2(X))^{-1} D_1(X)$ contains a non-polynomial entry then we can conclude $b_2 = 0$; otherwise $b_2 = 1$. The process continues until all bits $b_i, i = 1, \cdots, n$ are recovered. This means that the plaintext $b_1 \cdots b_n \in \{0,1\}^n$ can be recovered efficiently from ciphertext $C(x)$ and the public data alone.

2.2 Two Rososhek-Matrix Cryptosystems

In 2013, S. K. Rososhek [49] proposed a ElGamal-like encryption scheme -called BMMC ((Basic Matrix Modular Cryptosystem) - by using matrices over $\mathbb{Z}_n$.

1) BMMC: Let n be a large positive integer and let $G(\alpha, \beta, \gamma)$ be a free subgroup of the general linear group $\mathrm{GL}(2, \mathbb{Z}_n)$ generated by three generators A, B and C, where $\alpha, \beta, \gamma \in \mathbb{Z}$ with $\mid \alpha \mid, \mid \beta \mid, \mid \gamma \mid \geq 3$, $A = \begin{pmatrix} 1 & 0 \\ \alpha & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 & \beta \\ 0 & 1 \end{pmatrix}$ and $C = \begin{pmatrix} 1-\gamma & r \\ -\gamma & \gamma+1 \end{pmatrix}$. Let q be the order of the group $\mathrm{GL}(2, \mathbb{Z}_n)$. All the data above is public.
   a. Key Generation: Bob
      i. chooses two random matrices P_1 and U in $G(\alpha, \beta, \gamma)$ with $P_1 U \neq U P_1$ and three integers k, s, l with $-q \leq k, s \leq q$ and $2 \leq q$.
      ii. computes $P_2 := U^{-s} P_1^k U^s$ and $P_3 := U^l$.
      iii. The public key: n, P_1, P_2, P_3.
      The private key: U, k, s.
   b. Encryption: Let the message $m \in Mat(2, \mathbb{Z}_n)$ be a matrix. Alice chooses integers $r, t \in \mathbb{Z}_n$ and then computes the ciphertext
$$(C_1, C_2) := (P_3^{-r} P_1^t P_3^r, m P_3^r P_2^{-t} P_3^{-r}).$$

c. Decryption: Bob computes m by using his private key k, s:

$$C_2U^{-s}C_1^kU^s = m.$$

d. Security Analysis:
If Eve want to break the system, Eve has to solve the transformation and hybrid problems that are more complicated than the discrete logarithm problem in the group of the same cardinality. The both hard problems are described as follows.

Problem 4. *(The Transformation Problem): Find all solutions (Z, y) of the equation $ZP_1Z^{-1} = P_1^y$, where $Z \in \mathrm{GL}(2, \mathbb{Z}_n)$ and $\mid y \mid < q$ is an integer.*

Problem 5. *(The Hybrid Problem): Find all solutions (Y, x) of the equation $Z_0 = Y^x$, where $Y \in \mathrm{GL}(2, \mathbb{Z}_n)$ and $\mid x \mid < q$ is an integer.*

2) MMMC1: The BMMC requires three matrix modular exponentiations for key generation. There are three exponentiation under encryption and two exponentiations under decryption. In order to speed the algorithm, S. K. Rososhek [50] gave two modified schemes named MMMC1 (Modified Matrix Modular Cryptosystem one) and MMMC2. The both modified schemes are similar. We only introduced the MMMC1 here.

a. Key Generation: Bob

i. computes the integer n, where n may be either a power of a prime p^r or a product $n = pq$ of two distince primes.

ii. determines two invertible matrices $V, W \in \mathrm{GL}(2, \mathbb{Z}_n)$ in order to define two commuting inner automorphisms α, β of the ring $Mat(2, \mathbb{Z}_n)$: $\alpha(D) := V^{-1}DV$ and $\beta(D) := W^{-1}DW$, for all $D \in Mat(2, \mathbb{Z}_n)$.

iii. computes two automorphisms $\phi := \alpha^2\beta$ and $\psi := \alpha\beta^2$.

iv. chooses a matrix $L \in \mathrm{GL}(2, \mathbb{Z}_n)$ such that $L \notin G$.

v. The public key: $n, \phi(L), \psi(L^{-1})$.
The private key: V, W, α, β.

b. Encryption: Let the message $m \in Mat(2, \mathbb{Z}_n)$ be a matrix. Alice

i. chooses $Y \in G$ and define an inner automorphism ζ of the ring $Mat(2, \mathbb{Z}_n)$ by $\zeta(D) := Y^{-1}DY$.

ii. computes the matrices $\zeta(\phi(L)), \zeta(\psi(L^{-1}))$ and $m\zeta(\phi(L))$.

iii. chooses a unit $\gamma \in \mathbb{Z}_n$.

iv. computes the ciphertext

$$(C_1, C_2) = (\gamma^{-1} \cdot \zeta(\psi(L^{-1}),\ \gamma \cdot m \cdot \zeta(\phi(L))).$$

c. Decryption: Bob decrypts the message using his private key

$$C_2 \cdot \alpha^{-1}\beta(C_1) = m.$$

d. Security Analysis:
The security of the scheme is based on the "random salt" conjugacy search problem. This is for the given matrices A, B in $Mat(2, \mathbb{Z}_n)$ to find an invertible matrix $X \in \mathrm{GL}(2, \mathbb{Z}_n)$ and an integer $0 < \gamma < n$ such that $X^{-1}AX = \gamma B$.
If the integer γ in the encryption algorithm is removed, then the system is insecure. This is because the usual conjugacy search problem on the general linear group $\mathrm{GL}(2, \mathbb{Z}_n)$ is not hard. The equation $C_1 = Y^{-1}\psi(L^{-1})Y$ can be transformed to a system of four linear equations with four unknowns. On the other hand, the author [50] claimed that the "salt" γ can be found only under brute force attack and for large n this problem becomes intractable.

More about public key cryptosystems based on matrices, see for example [9,21,24,27,32,49,50,55–57,64] for an example.

3 Braid Groups

The braid groups were first introduced explicitly by E. Artin in 1925 [4]. There are several ways to represent braids, but the most common is through the use of Artin generators and the fundamental braid [15]. The (Artin's) braid groups, denoted as B_n, are groups of braids on n strands defined by the following presentation

$$B_n :=< \sigma_1, \cdots, \sigma_{n-1} \mid \sigma_i\sigma_{i+1} = \sigma_{i+1}\sigma_i,$$
$$\sigma_{i+1}\sigma_i\sigma_{i+1} = \sigma_i\sigma_{i+1}\sigma_i, 1 \le i \le n-1 > .$$

These are non-abelian torsion-free groups. The precise description, in particular is the geometric interpretation of Artin braid groups, see *e.g.* [2,10,17,18,30,36]. Due to their efficient computational quality, Artin's braid groups seemed to be a good candidate as a platform group for cryptographic applications.

At the beginning of the twenty-first century, some braid group-based public key cryptosystems were proposed. The pioneering papers for braid group-based cryptography include the Anshel-Anshel-Goldfeld scheme [2] in 1999 and the Ko-Lee *et al.* scheme [36] in 2000. Since then, braids group-based cryptography has attracted a great deal of attention. The security of the most proposed braid group cryptographic schemes is based on the conjugacy search problem or its variant versions, *e.g.* the membership search problem.

Problem 6. *(Membership Search Problem (or, Multiple Conjugacy Search Problem)) Given elements $x, a_1, a_2, \cdots, a_n$ of a group G, find an expression of x as a word in $a_1, a_2, \cdots, a_n$ (if it exists).*

Unfortunately, the conjugacy search problem on linear groups is not hard, and braid groups are linear groups [11,37]. Although the most proposed braid group-based cryptographic schemes are vulnerable to several announced attack methods [28], the research of the braid groups for cryptography has not decreased. Apart from the conjugacy search problem, there are other hard problems in braid groups that have not been studied extensively. We therefore present the works of Ko-Lee *et al.* and Anshel *et al.* and the corresponding attacks based on linear representations of the braid groups. The algorithms can be applied not only to braid groups, but also to any nonabelian groups.

3.1 Two Ko-Lee *et al.* Schemes

Let LB_k and RB_{n-k} be commuting subgroups of the braid group B_n, where $0 < k < n$, consisting of braids made by braiding left k strands and by braiding right $n-k$ strands among n strands, respectively. For any $a \in LB_k$ and $b \in RB_{n-k}$, the commutative rule holds: $ab = ba$.

1) The Key Agreement Scheme [36]: The algorithm is a Diffie-Hellman like algorithm.
   a. The public key: braids groups B_n, LB_k, RB_{n-k} and a braid $x \in B_n$.
   b. Alice chooses a random secret braid $a \in LB_k$ and sends $y_1 := axa^{-1}$ to Bob.
      Bob chooses a random secret braid $b \in RB_{n-k}$ and sends $y_2 := bxb^{-1}$ to Alice.
   c. Alice receives y_2 and computes the shared key $K = ay_2a^{-1}$.
      Bob receives y_2 and computes the shared key $K = by_1b^{-1}$.
2) The Encryption Scheme [36]:
   a. Bob's public key: x, y, where $x \in B_n$, $y := axa^{-1}$ and the hash function $H : B_n \to \{0,1\}^l$.
      Bob's private key: $a \in LB_k$.
   b. Encryption: Let $m \in \{0,1\}^l$ be a plaintext. Alice
      i. chooses a braid $b \in RB_{n-k}$ at random.
      ii. computes the ciphertext (c, d), where
      $$c = bxb^{-1}, \quad d = H(byb^{-1}) \oplus m.$$
   c. Decryption: Bob uses the prime key a to recover the message
      $$m = H(aca^{-1}) \oplus d.$$
3) Security Analysis:
   The security of both of these schemes is based on the conjugacy search problem on braid groups. To break the both schemes, it suffices for Eve to solve the Braid Diffie-Hellman Conjugacy Problem.

Problem 7. *(Diffie-Hellman Conjugacy Problem) Let A and B be commuting subgroups of a group G with $[A, B] = 1$, and let $g \in G$ be given. Given a pair (g^a, g^b) with $a \in A$ and $b \in B$, find g^{ab}.*

The Ko-Lee *et al.* key agreement scheme can be attacked by using the Lawrence-Krammer representation

$$\mathcal{K} : B_n \longrightarrow \mathrm{GL}(\frac{n(n-1)}{2}, \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]).$$

The proposed algorithm to solve the braid DH problem is described roughly as follows. Suppose that Eve can find a matrix A such that

$$\begin{aligned} \mathcal{K}(y_1)A &= A\mathcal{K}(y_1), \\ \mathcal{K}(\sigma_i)A &= A\mathcal{K}(\sigma_i), \end{aligned}$$

for all generators $\sigma_i \in LB_k$. Then, $A\mathcal{K}(y_2)A^{-1} = A\mathcal{K}(b)\mathcal{K}(x)\mathcal{K}(b)^{-1}A^{-1} = \mathcal{K}(b)\mathcal{K}(y_1)\mathcal{K}(b)^{-1} = \mathcal{K}(K)$. Note that the Lawrence-Krammer representation is faithful and one can effectively find the image $\mathcal{K}(g)$ for every $g \in B_n$. Moreover, one can effectively recover $K \in B_n$ from its image $\mathcal{K}(K)$ by using the Cheon-Jun inversion algorithm [16, 59].

3.2 Anshel-Anshel-Goldfeld Scheme

In contrast to Ko-Lee *et al.* schemes the Anshel-Anshel-Goldfeld key agreement scheme [2] requires no commuting subgroups. Let G be a public nonabelian group and $a_1, \cdots, a_k, b_1, \cdots, b_m \in G$ be public.

1) The Algorithm.
   a. Alice chooses a random secret $x = x(a_1, \cdots, a_k) \in G$ as a word in $a_1, \cdots, a_k$ and sends $b_1^x, \cdots, b_m^x$ to Bob.
      Bob chooses a random secret $y = y(b_1, \cdots, b_m) \in G$ as a word in $b_1, \cdots, b_k$ and sends $a_1^y, \cdots, a_k^y$ to Alice.
   b. Alice computes $x(a_1^y, \cdots, a_k^y) = x^y = y^{-1}xy$ and $x^{-1}(y^{-1}xy) = K$.
   c. Bob computes $y(b_1^x, \cdots, b_m^x) = y^x = x^{-1}yx$ and $(y^{-1}(x^{-1}yx))^{-1} = K$.
2) Security Analysis:
   In the paper [2], braid groups are selected as platform groups for the scheme. The security of the AAG scheme is based on the multiple conjugacy search problem, which is otherwise called the membership search problem. However, for Eve to extract the shared key K out of the public information, it suffices to solve the Commutator KE Problem, which is otherwise called the Anshel-Anshel-Goldfeld Problem, in polynomial time.

Problem 8. *(Commutator Key Exchange Problem) Let G be a group. Let $a_1, \cdots, a_k, b_1, \cdots, b_k \in G$ and*

let $a \in < a_1, \cdots, a_k >$ *and* $b \in < b_1, \cdots, b_k >$. *Given* $a_1, \cdots, a_k, b_1, \cdots, b_k, a_1^b, \cdots, a_k^b, b_1^a, \cdots, b_k^a$, *compute* $a^{-1}b^{-1}ab$.

Let S be a subset of $Mat(n, \mathbb{F})$. The centralizer of S is the set

$$C(S) := \{c \in Mat(n, \mathbb{F}) \mid cs = sc, \ \forall \ s \in S\}.$$

The centralizer $C(S)$ is a subspace of the vector space $Mat(n, \mathbb{F})$ over a field $\mathbb{F}$. The proposed algorithm [59] to solve the commutator key exchange problem is described roughly as follows. See [59] for details.

a. Use the method of Cheon and Jun [16] to reduce the commutator key exchange problem in matrix groups over fields.

$$B_n \overset{\mathcal{K}}{\hookrightarrow} \mathrm{GL}(\frac{n(n-1)}{2}, \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]) \twoheadrightarrow \mathrm{GL}(\frac{n(n-1)}{2}, \mathbb{F}),$$

where $\mathbb{F} = \mathbb{Z}[t]/ < p, f(t) >= \mathbb{Z}[t^{\pm 1}, \frac{1}{2}]/ < p, f(t) >$ is a finite field of the order $p^{\deg f(t)}$, p is a prime and $f(t)$ is an irreducible polynomial.

b. Compute a basis for $C(C(b_1, \cdots, b_k))$.

c. Find a matrix x (and its inverse x^{-1}) that satisfies the following homogeneous system of linear equations

$$\begin{aligned} b_1 \cdot x &= x \cdot b_1^a, \\ &\vdots \\ b_k \cdot x &= x \cdot b_k^a. \end{aligned}$$

Thus, $b_i^x = b_i^a$ and $xa^{-1} \in C(b_1, \cdots, b_k)$.

d. If $y \in C(C(b_1, \cdots, b_k))$, then $(xa^{-1})y = y(xa^{-1})$ and $(xa^{-1})y^{-1} = y^{-1}(xa^{-1})$. Therefore,

$$\begin{aligned} x^{-1}y^{-1}xy &= x^{-1}y^{-1}(xa^{-1})ay \\ &= x^{-1}(xa^{-1})y^{-1}ay \\ &= a^{-1}a^y. \end{aligned}$$

e. Find a matrix $y \in C(C(b_1, \cdots, b_k))$ (and its inverse y^{-1}) that satisfies the following homogeneous system of linear equations

$$\begin{aligned} a_1 \cdot y &= y \cdot a_1^b, \\ &\vdots \\ a_k \cdot y &= y \cdot a_k^b. \end{aligned}$$

f. Let $a = a_{i_1}^{\epsilon_1} \cdots a_{i_m}^{\epsilon_m}$. Then, compute

$$\begin{aligned} a^y &= (a_{i_1}^{\epsilon_1})^y \cdots (a_{i_m}^{\epsilon_m})^y = (a_{i_1}^y)^{\epsilon_1} \cdots (a_{i_m}^y)^{\epsilon_m} \\ &= (a_{i_1}^b)^{\epsilon_1} \cdots (a_{i_m}^b)^{\epsilon_m} = (a_{i_1}^{\epsilon_1})^b \cdots (a_{i_m}^{\epsilon_m})^b \\ &= a^b. \end{aligned}$$

g. Compute $a^{-1}a^y = a^{-1}a^b = a^{-1}b^{-1}ab$.

h. Recover the shared key K from $a^{-1}b^{-1}ab$ by using the Cheon-Jun inversion algorithm.

For more about braid groups and braid group-based public key cryptosystems, see for example [2, 4, 10, 11, 15–18, 23, 28, 30, 36, 37, 39, 44, 45, 47, 53, 59].

4 Stickel's Schemes

In 2003, E. Stickel presented the algorithms based on the Diffie-Hellman type of nonabelian groups. The algorithms cover the key agreement, authentication and digital signature purposes. Let G be a finite nonabelian group and let $a, b \in G$ with $ab \neq ba$ and $o(a) = N, o(b) = M > 1$.

1) Stickel's Key Agreement Scheme

a. Alice chooses two random natural number $n < N, m < M$ and sends $u := a^n b^m$ to Bob.
Bob chooses two random natural number $r < N, s < M$ and sends $v := a^r b^s$ to Alice.

b. Alice computes the shared secret key $K = a^n v b^m$.
Bob computes the shared secret key $K = a^r u b^s$.

2) Security Analysis:
Suppose that Eve wants to break the system and she has intercepted the values u and v. In order to get the secret shared key K, Eve does not have to find a pair of integers (n, m) (or (r, s)), but to solve the decomposition search problem [45, 54].

Problem 9. *(Decomposition Search Problem) Given a recursively presented (semi)group* G, *two recursively generated sub(semi)groups* $A, B \in G$, *and two elements* $u, w \in G$. *Find two elements* $x \in A$ *and* $y \in B$ *such that* $x \cdot w \cdot y = u$, *provided at least one such pair of elements exists.*

Suppose that Eve can find a pair $x, y \in G$ which satisfies the system

$$\begin{cases} xa &= ax \\ yb &= by \\ u &= xwy \end{cases}$$

then Eve can use Bob's transmission v to compute

$$xvy = xa^r wb^s y = a^r xwyb^s = a^r ub^s = K.$$

3) Suggested Platforms: In the paper [56], it was suggested that the general linear group $\mathrm{GL}_k(\mathbb{F}_{2^l})$ is used as the platform group G. Then the above system of three equations including a nonlinear equation can be translated to a system of three linear equations

$$\begin{cases} x^{-1}a &= ax^{-1} \\ yb &= by \\ xu &= wy. \end{cases}$$

It makes also the protocol vulnerable to linear algebra attacks. However, the system is worth preserving. The author of the paper [54] suggested semigroups with a great deal of non-invertible elements, and then the linear algebra attack would not work. Whether a semigroup (with a lot of non-invertible elements) as the platform makes the protocol vulnerable to another attacks is unclear.

5 Semidirect Products

In [47] (2001), S.-H. Paeng *et al.* described a public key encryption protocol based on a semidirect product of abelian groups connecting with the inner automorphism.

Definition 5.1. Let G and H be groups and $\rho : H \longrightarrow \mathrm{Aut}(G)$ be a group homomorphism. The semidirect product $G \rtimes_\rho H$ is a nonabelian group

$$G \rtimes_\rho H := \{(g, h) \mid g \in G, h \in H\}$$

under the group operation

$$(g_1, h_1) \cdot (g_2, h_2) := (g_1 \rho(h_1)(g_2), h_1 h_2).$$

Definition 5.2. Let G be a nonabelian group. Fix an element $g \in G$. An automorphism $\mathrm{Inn}(g) : G \longrightarrow G$ defined by

$$\mathrm{Inn}(g)(x) := gxg^{-1}, \ \forall\, x \in G$$

is called the inner automorphism of G by g.

Problem 10. *(Special Conjugacy Search Problem) Let G be a nonabelian group. Given an element $\mathrm{Inn}(g) \in \mathrm{Inn}(G)$. Find $g' \in G$ such that $\mathrm{Inn}(g') = \mathrm{Inn}(g)$.*

5.1 The Encryption Scheme

Let Γ be a semidirect product of the groups $\mathrm{SL}(2, \mathbb{Z}_p)$ and $\mathbb{Z}_p$. $\mathrm{Inn}(\Gamma) := \{\mathrm{Inn}(g) \mid g \in \Gamma\}$ be the inner automorphism group of Γ. The encryption scheme in [47]:

1) Bob's public key: $\mathrm{Inn}(g), \mathrm{Inn}(g^a), g := (x, y) \in \Gamma \setminus \mathcal{Z}(\Gamma)$.
   Bob's private key: $a \in \mathbb{Z}_{|\Gamma|}$.

2) Encryption: Let $m := (m_1, m_2) \in \Gamma \setminus \mathcal{Z}(\Gamma)$ with $mg \neq gm$ be a message. Alice

   a. chooses $b \in \mathbb{Z}$ and computes $\mathrm{Inn}(g^a)^b$.

   b. computes $E = \mathrm{Inn}(g^{ab})(m)$.

   c. computes $\phi = \mathrm{Inn}(g)^b$.

   Alice sends to Bob the cipher (E, ϕ).

3) Decryption: Bob computes $m = \phi^{-a}(E)$.

4) Security Analysis:
   The security of the encryption scheme is based on the difficulty of the special conjugacy search problem and the discrete logarithmic problem.

   If Eve wants to break the system, she has to find the private key a. Eve can try to find an element g_0 such that $\mathrm{Inn}(g_0) = \mathrm{Inn}(g^a)$. Then, it holds that $g_0 = g^a z$ for some $z \in \mathcal{Z}(\Gamma)$. After that Eve has to check whether $g_0 z^{-1} \in < g >$. If that is the case, then it goes back to solve the discrete logarithmic problem in the cyclic group $< g >$. Indeed, if the subgroup $\mathcal{Z}(\Gamma)$ of Γ is large, then it is less efficient to determine whether $g_0 z^{-1} \in < g >$.

   On the other hand, Eve considers directly solving the discrete logarithmic problem in the group $< \mathrm{Inn}(g) >$. The most efficient known method- the index calculus- cannot applied to the group $< \mathrm{Inn}(g) >$. In general case, the expected run times for solving he discrete logarithmic problem are $\mathcal{O}(\sqrt{p})$.

5) Suggested Platforms:
   The Author [47] employ a semi-direct product Γ of groups as the platform group of the system. Let $\Gamma = SL(2, \mathbb{Z}_p) \rtimes_\rho \mathbb{Z}_p$ an let $\rho := \mathrm{Inn} \circ \rho_1 : \mathbb{Z}_p \to \mathrm{Aut}(\mathrm{SL})(2, \mathbb{Z}_\mathrm{p}))$ be the automorphism, which is a composition of an inner automorphism Inn with an isomorphism ρ_1. For the DLP to be a hard problem in $< \mathrm{Inn}(g) >$, we choose 160-bit prime p. Then the security of the system is comparable to 1024-bit RSA.

5.2 HKKS-Key Exchange Protocol

In 2013, M. Habeeb *et al.* [29] proposed a new key agreement protocol (HKKS) by using semidirect products which is very different from the S.-H. Paeng *et al.* scheme. Let G be a (semi)group and let $\Gamma = G \rtimes H$ be a semidirect product, where $H \leq \mathrm{Aut}(G)$ is a subgroup. Let $(g, \phi) \in \Gamma$ be the public key for the protocol.

1) The HKKS-Key Exchange Portocol in [29]:

   a. Alice chooses a private number $m \in \mathbb{N}$ and computes $(g, \phi)^m = (\phi^{m-1}(g) \cdots \phi^2(g)\phi(g)g, \ \phi^m)$.
   Alice sends to Bob the first component

   $$a := \phi^{m-1}(g) \cdots \phi^2(g)\phi(g)g.$$

   Bob chooses a private number $n \in \mathbb{N}$ and computes $(g, \phi)^n = (\phi^{n-1}(g) \cdots \phi^2(g)\phi(g)g, \ \phi^n)$.
   Bob sends to Alice the first component

   $$b := \phi^{n-1}(g) \cdots \phi^2(g)\phi(g)g.$$

   b. Alice chooses any $x \in H$ and computes $(b, x)(a, \phi^m) = (\phi^m(b) \cdot a, \ x \cdot \phi^m)$.
   Bob chooses any $y \in H$ and computes $(a, y)(b, \phi^n) = (\phi^n(a) \cdot b, \ y \cdot \phi^n)$.

   c. The shared secret key K of Alice and Bob is the first component of $(b, x)(a, \phi^m) = (a, y)(b, \phi^n) = (g, \phi)^{m+n}$

   $$K = \phi^n(a) \cdot b = \phi^m(b) \cdot a.$$

2) Suggested Platforms:
In this paper [29], the authors consider the semigroup G of 3×3 matrices over the group ring $\mathbb{Z}_7[A_5]$, where A_5 is the alternating group on 5 letters and an extension of G by an inner automorphism to get a platform for the protocol. The public map ϕ is defined as an inner automorphism by using a fixed invertible matrix $h \in G$

$$\phi(g) := h^{-1}gh \text{ and } \phi^k(g) := h^{-k}gh^k$$

for any matrix $g \in G$ and any integer $k \geq 1$.

In order to reduce key size and to speed up computation of the algorithm, the authors of the paper [33] consider the semigroup G of 2×2 matrices over the binary field $\mathbb{F}_{2^{127}}$ and an extension of G by an endomorphism ϕ, which is a composition of a conjugation by a matrix $h \in \mathrm{GL}(2, \mathbb{F}_{2^{127}})$ with the endomorphism ψ as the platform semigroup. The public map ϕ is thus defined as follows

$$\begin{aligned} \phi(g) &:= h^{-1}\psi(g)h, \text{ and} \\ \phi^k(g) &:= (\prod_{i=0}^{k-1}\psi^i(h^{-1})\psi^k(g)(\prod_{i=k-1}^{0}\psi^i(h)), \end{aligned}$$

for any matrix $g \in G$ and any integer $k \geq 1$. The change of G reduces the bit complexity of a public matrix g from 1620-bits into 508-bits. All computation are done by well known methods of fast computation in finite binary fields. However, the choice of the both platforms makes the protocol unable to resist a linear algebra attack and a linear decomposition attack.

3) Security Analysis:
We give the linear algebra attack in the original version. Other attacks referred to [20, 29, 33, 34, 48]. If the inner automorphism ϕ_h by a matrix h over a field, i.e. $\phi_h(x) = hxh^{-1}$, is selected as the automorphism ϕ, then the HKKS-key exchange protocol is vulnerable to the linear algebra attack. The reason is described as follows:

a. Recall that Alice sends the matrix a to Bob

$$a = (\prod_{i=m-1}^{1} h^{-i}gh^i)\cdot g = h^{-m}(hg)^m,$$

and Bob sends the matrix b to Alice

$$b = (\prod_{i=n-1}^{1} h^{-i}gh^i)\cdot g = h^{-n}(hg)^n.$$

The shared secret key K is

$$K = \phi^m(b)\cdot a = \phi^n(a)\cdot b = h^{-(m+n)}(gh)^{m+n}.$$

b. Suppose that Eve wants to break the protocol. She finds two matrices X and Y satisfying the system of two linear and one nonlinear equations

$$\begin{cases} Xh &= hX, \\ Y(hg) &= (hg)Y, \\ XY &= h^{-m}(hg)^m. \end{cases}$$

If the matrix X is invertible, the system can be translated to the system of three linear equations

$$\begin{cases} X^{-1}h &= hX^{-1}, \\ Y(hg) &= (hg)Y, \\ Y &= X^{-1}h^{-m}(hg)^m. \end{cases}$$

It makes also the protocol vulnerable to a linear algebra attack, since Eve can thus compute the shared secret key by applying the solution (X, Y)

$$X(h^{-n}(hg)^n)Y = h^{-n}(XY)(hg)^n = K.$$

The next key exchange protocol is a modified version in order to prevent the linear algebra and linear decomposition attacks.

5.3 Modified HKKS-Key Exchange Protocol [33]

Let the automorphism ϕ be the inner automorphism ϕ_h by an invertible matrix h, i.e. $\phi_h(x) = hxh^{-1}$ and let $\mathrm{Ann}(hg) := \{x \mid x\cdot(hg) = O\}$ be the annihilator of the matrix hg, where O is denoted as the zero matrix. Alice and Bob agree on public matrices g and h, where h is invertible and g is not.

1) The modified version:

a. Alice chooses a secret number $m \in \mathbb{N}$ and a secret nonzero matrix $R \in \mathrm{Ann}(hg)$, and then computes

$$\begin{aligned} (g,\phi)^m &= (\phi^{m-1}(g)\cdots\phi^2(g)\phi(g)g,\ \phi^m) \\ &= (h^{-m}(hg)^m, \phi^m) = (a,\ \phi^m). \end{aligned}$$

Alice sends to Bob the matrix $a + R$.
Bob chooses a secret number $n \in \mathbb{N}$ and a secret nonzero matrix $S \in \mathrm{Ann}(hg)$, and then computes

$$\begin{aligned} (g,\phi)^n &= (\phi^{n-1}(g)\cdots\phi^2(g)\phi(g)g,\ \phi^n) \\ &= (h^{-n}(hg)^n, \phi^m) = (b,\ \phi^n). \end{aligned}$$

Bob sends to Alice the matrix $b + S$.

b. Alice chooses any $x \in H$ and computes

$$(b+S, x)(a+R, \phi^m) = (\phi^m(b)\cdot a,\ x\cdot\phi^m).$$

Bob chooses any $y \in H$ and computes

$$(a+R, y)(b+S, \phi^n) = (\phi^n(a)\cdot b,\ y\cdot\phi^n).$$

c. The shared secret key K of Alice and Bob is then
$$K = \phi^m(b) \cdot a = \phi^n(a) \cdot b.$$

2) Security Analysis:
The protocol uses R and S to hide a and b, respectively. Here, Eve would be looking for matrices X, Y and Z of the system of four equations
$$\begin{aligned} Xh &= hX, \\ Y(hg) &= (hg)Y, \\ Z \cdot (hg) &= O, \\ XY + Z &= h^{-m}(hg)^m + R. \end{aligned}$$
The last equation is not linear. The linear algebra attack does not work against this protocol. However, it is vulnerable against the linear decomposition attack which is described as follows [20, 48].

a. First, construct a linear space W generated by all elements of the form $h^{-k}(hg)^k, k \in \mathbb{N} \cup \{0\}$, with a basis $\{e_1, \cdots, e_l\}$, where $e_i = h^{-k_i}(hg)^{k_i}$, $k_i \in \mathbb{N}$. Choose a basis $\{f_1, \cdots, f_t\}$ of the linear space $\mathrm{Ann}(hg)$ such that $\{e_1, \cdots, e_l, f_1, \cdots, f_t\}$. This consists of a basis of the space $W + \mathrm{Ann}(hg)$.

b. For public data $a+R$ and $b+S$ in $W+\mathrm{Ann}(hg)$, we can effectively find one matrix $S_1 \in \mathrm{Ann}(hg)$ and the coefficients $\eta_i, \nu_j \in \mathbb{F}$ such that
$$\begin{aligned} b + S &= h^{-n}(hg)^n + S \\ &= \sum_{i=1}^{l} \eta_i h^{-k_i}(hg)^{k_i} + \sum_{j=1}^{t} \nu_j f_j, \end{aligned}$$
where $S_1 := \sum_{j=1}^{t} \nu_j f_j$ may bot be S.

c. Compute the shared secret key
$$\begin{aligned} & \sum_{i=1}^{l} \eta_i h^{-k_i}(a+R)(hg)^{k_i} \\ = & \sum_{i=1}^{l} \eta_i [h^{-k_i+m}(hg)^{k_i+m} + h^{k_i} \cdot R \cdot (hg)^{k_i}] \\ = & h^m (\sum_{i=1}^{l} \eta_i h^{-k_i}(hg)^{k_i})(hg)^m \\ = & h^{-m}(h^{-n}(hg)^n + S_1)(hg)^m \\ = & h^{-(m+n)}(hg)^{m+n} = K. \end{aligned}$$

More about public key cryptosystems based on semidirect products, see [20, 21, 29, 32–35, 42, 48] for an example.

6 Logarithmic Signatures

The logarithmic signatures were first used in the cryptography in order to construct a symmetric key cryptosystem PGM [40]. Nearly twenty years later, S.S. Magliveras *et al.* [38, 41] proposed three public key encryption schemes, called MTS_1, MTS_2 and MTS_3, based on logarithmic signatures for finite groups. Their security relies on the following hard factorization problem (Problem 11).

Let G be a finite (nonabelian) group and let $A_i := [a_{i1}, a_{i2}, \cdots, a_{ir_i}]$ be a finite sequence of elements of G, where r_i is called the length of A_i and $\overline{A_i}$ denotes the element $\sum_{j=1}^{r_i} a_{ij}$ in the group ring $\mathbb{Z}G$. An ordered sequence $\alpha := [A_1, A_2, \cdots, A_s]$ of A_i can be viewed as an $s \times r$ matrix $\alpha = (a_{ij})$, where $r = \max\{r_i\}$ and $a_{ij} = 0$ for $j > r_i$. Let $\overline{A_1} \cdot \overline{A_2} \cdots \overline{A_s} = \sum_{a_g \in G} a_g g$, where $a_g \in \mathbb{Z}$.

Definition 6.1. A sequence $\alpha = [A_1, A_2, \cdots, A_s]$ described as above is said to be

1) a cover for G if $a_g > 0$ for all $g \in G$.

2) a logarithmic signature for G if $a_g = 1$ for all $g \in G$.

3) a $[s, r]$-mesh cover if α is a cover for G, all A_i have the same length r and the distribution of the set $\{a_g \mid g \in G\}$ is approximately uniform.

Note that if $\alpha = [A_1, A_2, \cdots, A_s]$ is a logarithmic signature for G then for each element y of G there is a unique factorization with $q_i \in A_i, 1 \le i \le s$
$$y = q_1 \cdot q_2 \cdots q_s. \tag{1}$$
In general, it is not the case for covers.

A logarithmic signature α is called tame if the complexity of the factorization of y in (1) is in polynomial time. Otherwise, α is called wild.

Let $\alpha = [A_1, A_2, \cdots, A_s]$ be a cover of type $(r_1, r_2, \cdots, r_s)$ for G and let $m = \prod_{i=1}^{s} r_i$. Then the cover α induces an efficiently computable surjective mapping
$$\breve{\alpha} : \mathbb{Z}_m \longrightarrow G. \tag{2}$$
If α is a logarithmic signature, then the induced mapping $\breve{\alpha}$ is bijective. Moreover, if a logarithmic signature α is tame, then the inverse $\breve{\alpha}^{-1}$ is efficiently computable.

Proposition 6.2. If α is a wild logarithmic signature and β is a tame logarithmic signature for a finite group G, then the mapping $\breve{\alpha}\breve{\beta}^{-1} : \mathbb{Z}_{|G|} \longrightarrow \mathbb{Z}_{|G|}$ is a one-way permutation.

Let $\gamma : \{e\} = G_0 < G_1 < \cdots < G_{s-1} < G_s$ be a sequence of subgroups of G and let A_i be an ordered, complete set of right coset representatives of G_{i-1} in G_i. Then the sequence $[A_1, A_2, \cdots, A_s]$ forms a logarithmic signature α for G, and is called exact-transversal with respect to γ. If we set $B_i := g_{i-1}^{-1} A_i g_i$, $i = 1, \cdots, s$, where $g_0, g_1, \cdots, g_s \in G$, then the sequence $\beta : [B_1, B_2, \cdots, B_s]$ is again a logarithmic signature for G. When $g_0 = g_s = 1$, then β is said to be a sandwich of α.

Definition 6.3. A logarithmic signature α for a finite group G is called

1) transversal, if α is the sandwich of an exact-transversal logarithmic signature for G.

2) non-transversal, if it is not transversal.

3) totally non-transversal, if none of its blocks is a coset of a non-trivial subgroup of G.

Problem 11. *(Factorization Problem for Logarithmic Signatures/Covers) Given* $g \in G$ *and* $\alpha = [A_1, A_2, \cdots, A_s] = (a_{ij})$. *Find* $a_{ij_i} \in A_i, i = 1, \cdots, s$, *such that* $g = a_{1j_1} a_{2j_2} \cdots a_{sj_s}$.

6.1 MST$_1$- Cryptosystem

Let G be a finite permutation group with a sequence γ of subgroups of G.

1) Key Generation: Bob generates

   a. a totally non-transversal logarithmic signature α, a transversal logarithmic signature β for G and .

   b. a short sequence of exact-transversal logarithmic signatures $\theta_1, \theta_2, \cdots, \theta_k$ such that $\sigma := \breve{\alpha}\breve{\beta}^{-1} = \breve{\theta}_1 \cdot \breve{\theta}_2 \cdots \breve{\theta}_k$, where k is a small integer > 1.

   c. The public key: α, β, G.
   The private key: $\theta_1, \cdots, \theta_k$.

2) Encryption: Let $m \in \mathbb{Z}_{|G|}$ be a message. Alice computes the cipher

$$C = \sigma(m) = (\breve{\alpha}\breve{\beta}^{-1})(m).$$

3) Decryption: Bob recovers the message m by computing

$$\breve{\theta}_k^{-1}(\breve{\theta}_{k-1}^{-1}(\cdots(\breve{\theta}_1^{-1}(C)\cdots))) = m.$$

4) Security Analysis:
The security of MST$_1$ relies on the hardness of the factorization problem with respect to wild logarithmic signatures. In [41] the authors assume that totally non-transversal logarithmic signatures are "wild like". Unfortunately, J.M. Bohli *et al.* [14] have proved that totally non-transversal logarithmic signatures can be tame. This means that not any totally non-transversal logarithmic signature is suitable for being used as a key in MST$_1$. In addition to that there are still no practical implementations of MST$_1$ in sight.

6.2 MST$_2$- Cryptosystem

Let G and H be large groups.

1) Key Generation: Bob

   a. generates an epimorphism $f : G \to H$ and a random $[s, r]$-mesh cover $\alpha = (a_{ij})$ for G

   b. computes $\beta = (b_{ij}) = f(\alpha) = (f(a_{ij}))$.

   c. The public key: α, β.
   The private key: f.

2) Encryption: Let $m \in H$ be a message. Alice

   a. chooses $R \in \mathbb{Z}_{r^s}$.

   b. computes $y_1 = \breve{\alpha}(R)$, $y_2 = \breve{\beta}(R)$, $y_3 = my_2$.

   c. Alice transmits the cipher (y_1, y_3) to Bob.

3) Decryption: Bob

   a. computes $f(y_1) = y_2$, and

   b. computes $m = y_3 y_2^{-1}$.

4) Security Analysis:
If Eve wants to break the system, she has to find the value y_2. There are also two theoretical methods to recover y_2. One is to find a value $R^* \in \mathbb{Z}_{r^s}$ such that $\breve{\alpha}(R^*) = y_1$. If it is the case, then we can compute $\breve{\beta}(R^*) = y_2$. In order to effectively compute R^* such that $y_1 = \breve{\alpha}(R^*)$, Eve has to factorize the public data y_1 with respect to α. This means that Eve has to solve the Factorization Problem 11. This problem is in general an intractable problem for large groups. Second, Eve can try to find a homomorphism $f^* : G \to H$ such that $\beta = f^*(\alpha)$. S.S. Magliveras *et al.* [41] claimed that if the symmetric group S_n is used as the platform group of the scheme MST$_2$ and the private key $f : G \to G$ is conjugation by an element g in S_n, then the scheme MST$_2$ is vulnerable to the second attack.

6.3 MST$_3$- Cryptosystem

Let G be a nonabelian group with nontrivial center $\mathcal{Z}(G)$.

1) Key Generation: Bob

   a. generates a tame logarithmic signature $\beta = [B_1, B_2, \cdots, B_s] := (b_{ij})$ of type $(r_1, r_2, \cdots, r_s)$ for $\mathcal{Z}(G)$ and a random cover $\alpha = [A_1, A_2, \cdots, A_s] := (a_{ij})$ of the same type as β for a certain subset $\mathcal{F}$ of G such that $A_1, A_2, \cdots, A_s \subseteq G \setminus \mathcal{Z}(G)$.

   b. chooses $t_0, t_1, \cdots, t_s \in G \setminus \mathcal{Z}(G)$.

   c. computes $\breve{\alpha} := [\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_s]$, where $\tilde{A}_i = t_{i-1}^{-1} A_i t_i$.

   d. computes $\breve{\gamma} = (h_{ij})$, where $h_{ij} := b_{ij}\tilde{a}_{ij}$.

   e. The public key: $\breve{\alpha}, \breve{\gamma}$.
   The private key: $\beta, t_0, t_1, \cdots, t_s$.

2) Encryption: Let $m \in \mathbb{Z}_{|\mathcal{Z}(G)|}$ be a message. Alice

   a. computes $y_1 := \breve{\alpha}(m)$ and $y_2 := \breve{\gamma}(m)$.

   b. Alice transmits the cipher (y_1, y_2) to Bob.

3) Decryption: Bob

   a. computes $y_2 t_s^{-1} y_1^{-1} t_0 = \breve{\beta}(m)$.

   b. computes $m = \breve{\beta}^{-1}(y_2 t_s^{-1} y_1^{-1} t_0)$.

4) Security Analysis:
If Eve tries to obtain the private logarithmic signature β and the pair (t_0, t_s) of elements of G from the public information $(\breve{\alpha}, \breve{\gamma})$, it is sufficient for Eve to determine a sandwich transform β‘ of β which is equivalent to β, i.e., $\breve{\beta}$‘ $= \breve{\beta}$. Thus, it is sufficient to assume that the first element b_{1j} of each block B_j, except for the last block B_s of B, is the identity $1 \in G$. From the equation $h_{11} = t_0^{-1} a_{11} t_1$, Eve chooses an element of $G \setminus \mathcal{Z}(G)$ as the value of t_0. On the other hand, due to the elements t_i and $t_i z$ for $z \in \mathcal{Z}(G)$ lie in the same coset $t_i \mathcal{Z}(G)$. It is then sufficient to choose one t_0 only from each distinct coset of G modulo $\mathcal{Z}(G)$.

5) Suggested Platforms: The authors [38] of MTS_3 employ the Suzuki-2-group of the order q^2 as the platform group of MTS_3, where $q = 2^m$ is the order of the center $\mathcal{Z}(G)$ and the integer m is not a power of 2. Thus, if the Suzuki-2-group is implemented as the platform group of MTS_3, then there are $(q-1)q$ possible choices for the pair (t_0, b_{s1}). If q is large, the type of the attack is in not feasible.

More about public key cryptosystems based on logarithmic signatures see [14, 38, 40, 41, 58, 60] for an example.

7 Algebraic Eraser

The Algebraic Eraser is a binary operation consists of a semidirect product and a homomorphism of monoids and an action of a nonabelian group on a monoid. The main purpose of building the Algebraic Eraser is to design lightweight public key cryptosystems. The Algebraic Eraser key agreement scheme was introduced by Anshel, Anshel, Goldfeld and Lemieaux in 2004; the corresponding paper [1] appeared in 2006. The Algebraic Eraser and the Algebraic Eraser-based protocol are specially designed for commercial purposes. The company SecureRF owns the trademark of them. It claims a security level of 2^{128} for their preferred parameter sizes. The authors in the paper [1] gave a concrete realization of the Algebraic Eraser key agreement protocol using infinite braid groups named the colored Burau key agreement protocol (CBKAP). This Diffie-Hellman-like protocol has been proposed as a standard in ISO JTC-1/SC-31 (29167-20) to protect various communication protocols like RFID, NFC, or Bluetooth for devices associated with ISO-18000 and the Internet of Things [3, 8].

Let M, N be monoids and let S be a nonabelian group which acts on M on the left, and does not act on N. The semidirect product $M \rtimes S$ of M and S is then a monoid whose internal binary operation is given by

$$(m_1, s_1) \cdot (m_2, s_2) := (m_1^{s_1} m_2, s_1 s_2).$$

The Algebraic Eraser (AE) $\mathbf{E}$ is the binary operation specified within the 6-tuple

$$(M \rtimes S, N, \Pi, \mathbf{E}, A, B)$$

termed the $\mathbf{E}$-structure. where $\Pi : M \longrightarrow N$ is a monoid homomorphism, $\mathbf{E}$ is the function

$$\mathbf{E} : (N \times S) \times (M \rtimes S) \longrightarrow N \times S$$

given by $\mathbf{E}((n, s), (m_1, s_1)) := (n\Pi({}^s m_1), s s_1)$, and A, B are submonoids of $M \rtimes S$ satisfying $\mathbf{E}$-Commuting, i.e., they satisfy the equation

$$\mathbf{E}(\Pi(a), s_a, (b, s_b)) = \mathbf{E}(\Pi(b), s_b, (a, s_a)),$$

for all $(a.s_a) \in A$, $(b, s_b) \in B$. For simplicity, the symbol $\star$ will be used to replace the symbol $\mathbf{E}$ as follows:

$$\mathbf{E}((n, s), (m_1, s_1)) = (n, s) \star (m_1, s_1).$$

The operation $\star$ satisfies the associated property as follows: Given $(n, s) \in N \times S$ and $(m_1, s_1), (m_2, s_2) \in M \rtimes S$,

$$((n, s) \star (m_1, s_1)) \star (m_2, s_2) = (n, s) \star ((m_1, s_1) \cdot (m_2, s_2)).$$

7.1 The Key Agreement Scheme based on the Algebraic Eraser

The Algebraic Eraser key agreement scheme designed by Anshel, Anshel, Goldfeld and Lemieaux in [3] is a type of Diffie-Hellman protocol: Let N_A and N_B be submonoids of N so that they commute elementwise.

1) Alice selects her private key: $n_a \in N_A$ and $(a_i, s_{a_i}) \in A$.
Bob selects his private key: $n_b \in N_B$ and $(b_j, s_{b_j}) \in B$.

2) Alice computes P_A

$$P_A = (n_a, id) \star (a_i, s_{a_i}),$$

and transmits P_A to Bob.
Bob computes P_B

$$P_B = (n_b, id) \star (b_1, s_{b_1}),$$

and transmits P_B to Alice.

3) The shared secret key K of Alice and Bob is

$$((n_a, id) \cdot P_B) \star (a_i, s_{a_i}) = ((n_b, id) \cdot P_A) \star (b_j, s_{b_j}).$$

7.2 The CBKAP

The braid groups is used in order to implement the CBKAP. E-multiplication is an action of the braid group on pairs of matrices over a field and permutations. Recall that there is a surjective homomorphism from the Artin braid group B_n onto the symmetric group S_n. With help of the colored Burau representation of B_n, that is an extended version of the reduced Burau representation [3, 10, 17], the semi-direct product $M \rtimes S_n$ is defined as a group generated by the set $\{(x_1(t), s_1), \cdots, (x_{n-1}(t), s_{n-1})\}$, where $x_i(t)$ is a colored Burau matrix of B_n and $s_i = (i\ i+1)$ is a transposition of S_n, for all $i = 1, \cdots, n-1$.

The nonabelian group $M \rtimes S_n$ is also called the colored Burau group. The authors of the paper [3] then give an example to concretely realize the above algorithm by using the colored Burau matrices. The protocol is also called the colored Burau key agreement protocol (CBKAP).

Fix an integer $n \geq 7$ and a prime $p > n$. Let $M \leq \mathrm{GL}(n, \mathbb{F}_p(t))$, where $t = (t_1, \cdots, t_n)$, be a subgroup, and let $S = S_n$ be the symmetric group on n symbols and $N = \mathrm{GL}(n, \mathbb{F}_p)$. Fix n elements $\tau_1, \cdots, \tau_n \in \mathbb{F}_p$, and the homomorphism $\Pi : M \longrightarrow N$ is defined by setting $\tau_i = t_i, i = 1, \cdots, n$. Let $z \in M \rtimes S_n$ be a fixed element and let $A = z \cdot \{(x_{l_1}(t), s_{l_1}), \cdots, (x_{l_\mu}(t), s_{l_\mu})\} \cdot z^{-1}$ and $B = z \cdot \{(x_{r_1}(t), s_{r_1}), \cdots, (x_{r_\nu}(t), s_{r_\nu})\} \cdot z^{-1}$, where $\mid l_i - r_j \mid \geq 2$ for $1 \geq i, j \leq n$, be two E-commuting subgroups of $M \rtimes S_n$.

1) The public key: $(M \rtimes S, N, \Pi, \mathbf{E}, A, B)$ and a matrix $m_0 \in \mathrm{GL}(n, \mathbb{F}_p)$ of order $p^n - 1$ such that two subgroups N_A, N_B of N consist of linear combinations of powers of m_0 over $\mathbb{F}_p$.
The private key: z and $(x_{l_1}(t), s_{l_1}), \cdots, (x_{l_\mu}(t), s_{l_\mu})$, $(x_{r_1}(t), s_{r_1}), \cdots, (x_{r_\nu}(t), s_{r_\nu})$.

2) Alice selects her secret key: $n_a \in N_A$, $k \in \mathbb{N}$ and some $(x_{a_i}(t), s_{a_i}) \in A$, $i = 1, \cdots, k$.
Bob selects his secret key: $n_b \in N_B$, $l \in \mathbb{N}$ and some $(x_{b_j}(t), s_{b_j}) \in B$, $j = 1, \cdots, l$.

3) Alice computes P_A

$$\begin{aligned} P_A &= (\cdots((n_a, id) \star z \star (x_{a_1}(t), s_{a_1}) \star (x_{a_2}(t), s_{a_2})) \star \\ &\qquad \cdots) \star (x_{a_k}(t), s_{a_k}) \star z^{-1} \\ &= (n_a \cdot \Pi(z) \cdot \Pi({}^{s_z}\mathcal{A}) \cdot \Pi({}^{s_z s_{\mathcal{A}}} z^{-1}),\ s_z s_{\mathcal{A}} s_{z^{-1}}), \end{aligned}$$

where $\Pi({}^{s_z}\mathcal{A}) = \Pi({}^{s_z} x_{a_1}(t))\Pi({}^{s_z s_{a_1}} x_{a_2}(t)) \cdots \cdot \Pi({}^{s_z s_{a_1} \cdots s_{a_{k-1}}} x_{a_k}(t))$ and $s_{\mathcal{A}} = s_{a_1} \cdots s_{a_k}$.
Alice transmits P_A to Bob.
Bob computes

$$\begin{aligned} P_B &= (\cdots((n_b, id) \star z \star (x_{b_1}(t), s_{b_1}) \star (x_{b_2}(t), s_{b_2})) \star \\ &\qquad \cdots) \star (x_{b_l}(t), s_{b_l}) \star z^{-1} \\ &= (n_b \cdot \Pi(z) \cdot \Pi({}^{s_z}\mathcal{B}) \cdot \Pi({}^{s_z s_{\mathcal{B}}} z^{-1}),\ s_z s_{\mathcal{B}} s_{z^{-1}}), \end{aligned}$$

and transmits P_B to Alice.

4) The secret shared key K of Alice and Bob is

$$\begin{aligned} K &= (\cdots((n_a, id) \cdot P_B \star z \star (x_{a_1}(t), s_{a_1}) \\ &\quad \star (x_{a_2}(t), s_{a_2})) \star \cdots) \star (x_{a_k}(t), s_{ak}) \star z^{-1} \\ &= (\cdots((n_b, id) \cdot P_A \star z \star (x_{b_1}(t), s_{b_1}) \\ &\quad \star (x_{b_2}(t), s_{b_2})) \star \cdots) \star (x_{b_l}(t), s_{b_l}) \star z^{-1}. \end{aligned}$$

5) Security Analysis:
For simplicity, we assume that the matrix $n_a = m_0^\alpha$ for a secret $\alpha \in \mathbb{Z}^+$. If Eve intercepts the public data P_A and P_B and tries to break the shared key K, it is sufficient to merely solve the matrix m_0^α and the element $z \in M \rtimes S_n$. First, Eve diagonalizes the matrix m_0: $Qm_0Q^{-1} = (\lambda_1, \cdots, \lambda_n)$, where $Q \in N$. Then,

$$m_0^\alpha = Q^{-1}(\lambda_1^\alpha, \cdots, \lambda_n^\alpha)Q.$$

On the other hand, by the condition that the subgroups N_A and N_B of N commute, it applies $\Pi({}^{s_z}\mathcal{A})\Pi({}^{s_z}\mathcal{B}) = \Pi({}^{s_z}\mathcal{B})\Pi({}^{s_z}\mathcal{A})$. Thus, the matrices $\Pi({}^{s_z}\mathcal{A})$ and $\Pi({}^{s_z}\mathcal{B})$ take the forms $\begin{pmatrix} X & 0 \\ 0 & I \end{pmatrix}$ and $\begin{pmatrix} I & 0 \\ 0 & Y \end{pmatrix}$, respectively.

Suppose that z were known. Then, Eve can obtain m_0^α and recover the matrices $\Pi({}^{s_z}\mathcal{A})$ and $\Pi({}^{s_z s_{\mathcal{B}} s_{\mathcal{A}}} z^{-1})$ in polynomial time. Therefore, it remains to ask how to determine the element z. The security of the CBKAP depends on the simultaneous conjugacy search problem. There are not any successful attacks to solve the simultaneous conjugacy search problem.

Problem 12. *(Simultaneous Conjugacy Search Problem) Let $w_1 = z^{-1}a_1z, \cdots, w_k = z^{-1}a_kz$. If only $w_1, \cdots, w_k$ are public, find the conjugating element z.*

For more about Algebraic Eraser key agreement scheme, see [1, 3, 5–8, 13].

8 Conclusion

There are innovative ideas to propose nonabelian group based-public key cryptography, although, most cryptographic systems seem to be vulnerable to security. For example, the conjugacy search problem on linear groups used in the mentioned protocols, *e.g.*, matrix groups and braid groups, seems to be not be hard. Nevertheless, they still have the value of reference. Some of these systems have some modifications that still have a sufficient security level. On the other hand, the efficiency and security of a cryptographic system does not only depend on the design of the algorithm, but also on the choice of platform.

References

[1] I. Anshel, M. Anshel, D. Goldfeld, S. Lemieux, "Key agreement, the algebraic eraser^{TM}, and lightweight cryptography," *Contemporary Mathematics*, vol. 418, pp. 1-34, 2006.

[2] I. Anshel, M. Anshel, D. Goldfeld, "An algebraic method for public-key cryptography," *Mathematics Research Letter*, vol. 6, pp. 287-291, 1999.

[3] I. Anshel, D. Atkins, D. Goldfeld, P. Gunnells, "Defeating the Ben-Zvi, Blackburn, and Tsaban Attack on the Algebraic Eraser," *IACR ePrint* 2016/044.

[4] E. Artin, "Theory of braids," *Annal of Mathematics*, vol. 48, pp. 101-126, 1947.

[5] D. Atkins, *Algebraic Eraser: A Lightweight, Efficient Asymmetric Key Agreement Protocol for use in No-Power, Low-Power, and IoT Devices*, 2015. (csrc.nist.gov/groups/ST/lwc-workshop2015/papers/session8-atkins-paper.pdf)

[6] D. Atkins, D. Goldfeld, *Addressing the Algebraic Eraser Diffie-Hellman Over-the Air Protocol*, 2015. (http://eprint.iacr.org/2016/205.pdf)

[7] D. Atkins, P. E. Gunnells, *Algebraic Eraser: A Lightweight, Efficient Asymmetric Key Agreement Protocol for use in No-Power, Low-Power, and IoT Devices*, 2015. (http://csrc.nist.gov/groups/ST/lwc-workshop2015/presentations/session8-atkins-gunnell.pdf)

[8] A. Ben-Zvi, S. R. Blackburn, B. Tsaban, "A Practical Cryptanalysis of the Algebraic Eraser," 2016. (https://arXiv:1511.03870v2)

[9] G. Baumslag, B. Fine, X. Xu, "Cryptosystems using Linear Groups," *Applicable Algebra in Engineering, Communication and Computing*, vol. 17, no. 3-4, pp. 205-217, 2006.

[10] G. Baumslag, B. Fine, M. Kreuzer, G. Rosenberger, *A Course in Mathematical Cryptography*, De Gruyter, 2015.

[11] S. Bigelow, "Braid groups are linear," *Journal of the American Mathematical*, vol. 14, pp. 471-486, 2001.

[12] S. R. Blackburn, C. Cid, C. Mullan, *Group Theory in Cryptography*, 2010. (https://arxiv.org/pdf/0906.5545)

[13] S. R. Blackburn, M. J. B. Robshaw, "On the security of the algebraic eraser tag authentication protocol," in *14th International Conference on Applied Cryptography and Network Security (ACNS'16)*, Lecture Notes in Computer Science, vol. 9696, pp. 3-17, 2016.

[14] J. M. Bohli, M. I. Gonzaĺez Vasco, C. Martínez, R. Steinwandt, "Weak keys in MST_1," *Design, Code and Cryptography*, vol. 37, pp. 509-524, 2005.

[15] Z. Busser, *Braid Group Cryptography*, 2009.

[16] J. Cheon, B. Jun, "A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem," in *Advances in Cryptography (CRYPTO'03)*, Lecture Notes in Computer Science, vol. 2729, pp. 212-224, 2003.

[17] P. Dehornog, "Braid-based cryptography," *American Mathematical Society*, vol. 360, pp. 5-33, 2004.

[18] P. Dehornog, "Using shifted conjugacy in braid-based cryptography," *Contemporary Mathematics*, vol. 418, pp. 65-94, 2006.

[19] W. Diffie, M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory IT*, vol. 22, pp. 644-654, 1976.

[20] J. Ding, A. Miasnikov, A. Ushakov, *A Linear Attack on a Key Exchange Protocol Using Extensions of Matrix Semigroups*, 2015. (http;//eprint.iacr.org/2015/018)

[21] M. Eftekhari, *Cryptanalysis of Some Protocols using Matrices over Group Rings*, 2015. (http://arxiv.org)

[22] B. Fine, M. Habeeb, D. Kahrobaei, G. Rosenberger, *Aspects of Nonabelian Group Based Cryptography: A Survey and Open Problems*, 2011. (https://arXiv:1103.4093v2)

[23] N. Franco, J. González-Meneses, "Conjugacy problem for braid groups and garside groups," *Journal of Algebra*, vol. 266, pp. 112-132, 2003.

[24] D. Freeman, *The Discrete Logarithm Problem in Matrix Groups*, 2004.

[25] D. Grigoriev, "Public-key cryptography and invariant theory," *Journal of Mathematical Sciences*, vol. 126, no. 3, pp. 1152-1157, 2005.

[26] D. Grigoriev, A. Kojevniko, S. Nikolenko, *Invariant-based Cryptosystem and Their Security Against Provable Worst-Case Break*, Technical Report 158, Max-Planck-Inst, Preprints, 2007.

[27] D. Grigoriev, I. Ponomarenke, "Constructions in public-key cryptography over matrix groups," *Contemporary Mathematics: Algebraic Methods in Cryptography*, vol. 418, pp. 103-120, 2007.

[28] S. D. Hasapis, D. Panagopoulos, "A survey of group-based cryptogrsphy," *Journal of Applied Mathematics & Bioinformatics*, vol. 5, no. 3, pp. 73-96, 2015.

[29] M. Habeeb, D. Kahrobaei, C. Koupparis, V. Shpilrain, "Public key exchange using semidirect product of (semi)groups," *Lecture Notes in Computer Science*, vol. 7954, pp. 475-486, 2013.

[30] U. Isik, *Computational Problems in the Braid Group with Applications to Cryptography*, 2005.

[31] D. Kahrobaei, M. Anshel, "Decision and search in non-abelian Cramer-Shoup public key cryptosystem," *Group Complexity Cryptology*, vol. 1, no. 2, pp. 97-115, 2009.

[32] D. Kahrobaei, C. Koupparis, V. Shpilrain, "Public key exchange using matrices over group rings," *Group Complexity Cryptology*, vol. 5, pp. 217-225, 2013.

[33] D. Kahrobaei, H. T. Lam, V. Shpilrain, *Public Key Exchange using Extensions by Endomorphisms and Matrices over a Galois Field*, *Preprint*, 2014.

[34] D. Kahrobaei, V. Shpilrain, *Using Semidirect Product of (Semi)groups in Public Key Cryptography*, 2016. (https://arXiv:1604.05542v1)

[35] K. H. Ko, D. H. Choi, M. S. Cho, J. W. Lee, *New Signature Scheme using Conjugacy Problem*, 2002. (http://eprint.iacr.org/2002/168)

[36] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. S. Kang, C. Park, "New public key cryptosystem using braid groups," in *Advances in Cryptology (CRYPTO'00)*, pp. 166-184, 2000.

[37] D. Krammer, "Braid groups are linear," *Annals of Mathematics*, vol. 155, pp. 131-156, 2002.

[38] W. Lempken, T. van Trung, S. S. Magliveras, W. Wei, "A public key cryptosystem based on non-abelian finite groups," *Journal of Cryptology*, vol. 22, pp. 62-74, 2009.

[39] K. Mahlburg, *An Overview of Braid Group Cryptography*, Preprint, 2004.

[40] S. S. Magliveras, "A cryptosystem from logarithmic signatures of finite groups," in *Proceeding of the 29th Midwest Symposium on Circuils and Systems, Elsevier Publishing Company*, pp. 972-975, 1986.

[41] S. S. Magliveras, D. R. Stinson, T. van Trung, "New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups," *Journal of Cryptology*, vol. 15, pp. 285-297, 2002.

[42] C. Monico, M. Neusel, "Cryptanalysis of a system using matrices over group rings," *Group Complexity Cryptology*, vol. 7, no. 2, pp. 175-182, 2015.

[43] A. Myasnikov, V. Shpilrain, "A linear decomposition attack," *Group Complexity Cryptology*, vol. 7, pp. 81-94, 2015.

[44] A. Myasnikov, V. Shpilrain, A. Ushakov, "A practical attack on a braid group based cryptographic protocol," in *Advances in Cryptology (CRYPTO'05)*, pp. 86-96, 2005.

[45] A. Myasnikov, V. Shpilrain, A. Ushakov, *Group-based Cryptography*, Birkhäuser Verlag, 2008.

[46] A. D. Myasnikov, A. Ushakov, "Quantum algorithm for the discrete logarithm problem for matrices over finite group rings," *Group Complexity Cryptology*, vol. 6, pp. 31-36, 2014.

[47] S. H. Paeng, K. C. Ha, J. H. Kim, S. Chee, C. Park, "New public key cryptosystem using finite non-abelian groups," in *Advance in Cryptology (CRYPTO'01)*, Lecture Notes in Computer Science, vol. 2139, pp. 470-485, 2001.

[48] V. Roman'kov, *Linear Decomposition Attack on Public Key Exchange Protocols using Semidirect Products of (Semi)Groups*, 2015. (`https://arXiv:1501.01152v1`)

[49] S. K. Rososhek, "New practical algebraic public-key cryptosystem and Some related algebraic and computational aspects," *Applied Mathematics*, vol. 4, no, 7, pp. 1043-1049, 2013.

[50] S. K. Rososhek, "Modified matrix modular cryptosystems," *Britsh Journal of Mathematics & Computer Science*, vol. 5, no. 5, pp. 613-636, 2015.

[51] P. Svaba, *Covers and Logarithmic Signatures of Finite Groups in Cryptography*, Dissertation, 2011.

[52] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithm problems," *SIAM Journal on Computing*, vol. 26, pp. 1484-1509, 1997.

[53] V. Shpilrain, A. Ushakov, *The Conjugacy Search Problem in Public Key Cryptography: Unnecessary and Insufficient*, 2004. (`https://arXiv:math/0411644v1`)

[54] V. Shpilrain, "Cryptanalysis of stickel's key exchange scheme," *Lecture Notes in Computer Science*, vol. 5010, pp. 283-288, 2008.

[55] R. Steinwandt, "Loopholes in two public key cryptosystems using the modular group," *Lecture Notes in Computer Science*, vol. 1992, pp. 180-189, 2002.

[56] E. Stickel, *A New Public-Key Cryptosystem in Non-Abelian Groups*, 2003. (`https://www.semanticscholar.org`)

[57] E. Stickel, "A new method for exchanging secret keys," in *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*, pp. 426-430, 2005.

[58] P. Svaba, *Covers and Logarithmic Signatures of finite Groups in Cryptography*, Dissertation, 2011.

[59] B. Tsaban, "Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography," *Journal of Cryptology*, vol. 28, pp. 601-622, 2015.

[60] M. I. G. Vasco, D. Hofheinz, C. Martinez, R. Steinwandt, "On the security of two public key cryptosystems using non-abelian groups," *Designs, codes and Cryptography*, vol. 32, pp. 207-216, 2004.

[61] N. R. Wagner, M. R. Magyarik, "A public key cryptosystem based on the word problem," in *Advances in Cryptology (CRYPTO'84)*, Lecture Notes in Computer Science, vol. 196, pp. 19-36, 1985.

[62] L. Wang, L. Wang, Z. Cao, E. Okamoto, J. Shao, "New constructions of public-key encryption schemes from conjugacy search problems," in *Information Security and Cryptology*, Lecture Notes in Computer Science, vol. 6584, pp. 1-17, 2010.

[63] X. Wang, C. Xu, G. Li, H. Lin, W. Wang, "Double shielded public key cryptosystems," *Cryptology ePrint Archive Report 2014/588*, pp. 1-14, 2014.

[64] A. Yamamura, "Public-key cryptosystems using the modular group," in *1st Internationa Workshop on Practice and Theory in Public Key Cryptography (PKC'98)*, Lecture Notes in Computer Science, vol. 1431, pp. 203-216, 1998.

Biography

Tzu-Chun Lin received the PhD in Mathematics from the Faculties for Mathematics and Science of the Georg-August-University at Göttingen in Germany. She is an associate professor in the Department of Applied Mathematics of Feng Chia University in Taiwan, R.O.C.. Her current research interests include commutative algebras, invariant theory of finite groups and public-key cryptography.