SOME REMARKS ON THE LOGARITHMIC SIGNATURES OF FINITE ABELIAN GROUPS

THUONG T. DANG [1], TRI T. TON [2], VAN H. DANG [1], THUC D. NGUYEN [1]

Abstract. In the paper about the cryptosystem MST3, Svaba and Trung proposed a way to build a cryptosystem based on the concept of logarithmic signatures, and they choose Suzuki's group, which is not abelian for implementing. Recently, to reason why these methods cannot be applied to abelian groups; Svaba, Trung and Wolf developed some algorithms to factorize the fused transversal logarithmic signatures (FTLS). Their attacks can be avoided by some modifications, which is the aim of this paper, where we will use the weakness of the discrete logarithm problem (DLP) to propose two cryptosystems. The first one is based on the new concept about quasi-logarithmic signature of finite solvable groups, which is the generalization of logarithmic signatures. The second is built on the logarithmic signatures of finite cyclic 2-groups, which include two interesting examples on Pell's curves and elliptic curves over finite fields.

Keywords. logarithmic signatures, quasi-logarithmic signatures, cyclic 2-group, elliptic curves

Contents

[1] Faculty of Information Technology, University of Sciences, Ho Chi Minh City, Viet Nam.

[2] Faculty of Mathematics and Applications, Saigon University, Ho Chi Minh City, Viet Nam.

1. Introduction

The DLP is one of the fundamental problems of cryptography, which is stated "Let G be a cyclic group generated by g. Given $x \in G$, can we find $k \in \mathbb{Z}$ in polynomial time such that: $g^k = x$?" If G has larger prime order, in general, there is no known algorithm in polynomial time to solve this problem. Otherwise, if the order of G can be factorized into products of small primes, then the DLP of G can be solved in polynomial time, as we will discuss later. The raised question is "Can we use the ease of the DLP on such groups to build other cryptosystems, which are still secure?" The ideas from logarithmic signatures can be applied to give the answer of this question.

In the next section, we will recall some basic concepts about logarithmic signatures, and some cryptographic aspects, including the result of Svaba et al. about factorization of FTLS on abelian group [10]. The third section will be devoted to introduce the generalized concept of logarithmic signatures, named quasi-logarithmic on finite abelian groups (or more general cases: on solvable groups) to avoid this attack, and to reduce the storage of trapdoor information of MST3 cryptosystem. Then we will use the logarithmic signatures to give an algorithm to solve the weak DLP on cyclic 2-group in the fourth section, and this method can be used to solve the DLP on any cyclic group, whose order consist of small prime factors. Some nontrivial examples are given in Section 5. Then, the last section will be devoted to introduce the other cryptosystem based on the concept of logarithmic signatures, and designed to avoid the attack of Svaba et al.

2. Preliminaries

In this section, we briefly describe the definition and properties of logarithmic signatures used by Svaba and Trung [9].

Definition 2.1. Let G be a finite group, and $A_1, A_2, ..., A_s$ be subsets of G. For each $g \in G$, if we can express $g = a_{1,k_1}a_{2,k_2}...a_{s,k_s}$ where $a_{i,k_i} \in A_i$ in exactly one way, then the set $\alpha = \{A_1, A_2, ..., A_s\}$ is called the *logarithmic signatures* of G.

The simplest way to build such a logarithmic signature can be done as follows: if G be a finite group, then there exists a series of subgroup of G

$$G_0 = G \geq G_1 \geq ... \geq G_s = 1$$

where G_{i+1} is the subgroup of $G_i (0 \leq i \leq s-1)$. We denote $A_i = \{a_{i,1}, ..., a_{i,r_i}\}$ the coset representatives of G_{i+1} in G_i, then $\alpha = \{A_0, ..., A_{s-1}\}$ is the logarithmic signature of G. In [10], the authors call such α the *transversal logarithmic signatures* (TLS) of G. According to the definition of logarithmic signatures, there exists a bijective map

$$\breve{\alpha} : \mathbb{Z}_{r_1} \times ... \times \mathbb{Z}_{r_s} \to G$$
$$(k_1, ..., k_s) \mapsto a_{1,k_1}...a_{s,k_s}$$

If $\breve{\alpha}^{-1}$ can be computed efficiently, α is called *tame* (or *factorizable*). Here is the important definition that transforms a tame logarithmic signatures to a *not* tame (or *wild*) logarithmic signatures.

Definition 2.2. Given a logarithmic signature $\alpha = \{A_1, ..., A_s\}$, the following transformations are called *fused operations* on α

(1) **Block permutation.** Permutes the block A_i's of α.
(2) **Element permutation.** Permutes the elements within blocks A_i of α.
(3) **Block replacement.** Replaces block A_i with $A_i g$ for some $g \in G, A_i \in \alpha$.
(4) **Block mix.** Replaces two blocks A_i and A_j with a single block $A_i A_j = \{xy | x \in A_i, y \in A_j\}$.

Consequently, we obtain the following proposition.

Proposition 2.3. *Applying a finite number of four transformations defined above to a given logarithmic signatures α, the obtained result β is also a logarithmic signatures.*

Such β in Proposition 2.3 is called *fused transversal logarithmic signatures* (FTLS), and the finite number of transformations (1), (2), (3), (4) to transform α to β is called *trapdoor information*.

In [10], the authors developed some algorithms to factorize the FTLS of finite abelian groups. Their main idea is trying to find the group structure in the FTLS. Note that $A_s = G_{s-1}/G_s$, and for all i, the mixed block $A_s A_i$ always contains a group as a subset, because A_i have the identity element. Although we can use the fused operations to break its group structure, we can normalize it, that means, making it have the identity element in each block. By this way, Svaba et al. developed a way to factorize the FTLS on abelian groups.

To avoid this factorizing, we propose two ways. The first one is to completely break the group structure, even in A_s. But to do this, we have to modify the logarithmic signatures structure and it leads us to the concept of quasi-logarithmic signatures, which is the generalization of logarithmic signatures. And the other way is to hide the FTLS, and to consider it trapdoor information also. These ideas will be discussed in detail in our next sections.

3. Quasi-logarithmic signatures

We turn to the generalized definition of logarithmic signatures.

Definition 3.1. Let $A_1, A_2, ..., A_s$ be subsets of a group G. We call $\alpha = \{A_1, ..., A_s\}$ *quasi-logarithmic signatures* (QLS) if for any $g \in G$ we can uniquely choose $A_{g_1}, ..., A_{g_i} (1 \leq g_1 < ... < g_i \leq s)$ from α such that:

$$g = (a_{1,k_{11}} ... a_{1,k_{1u_1}})(a_{2,k_2} ... a_{2,k_{21,u_2}}) ... (a_{i,k_{i1}} ... a_{i,k_{iu_i}})$$

in exactly one way, where $a_{j,k_{jv}} \in A_{g_j} (1 \leq j \leq i, 1 \leq v \leq u_j)$

In the definition of logarithmic signatures, it is only possible for us to choose one element in each A_i for the factorization of $g \in G$. Therefore, if α is a logarithmic signatures of G, then it is QLS. For QLS, to factorize an element of G, we can uniquely choose several or no element from a certain block. We will discuss a way to construct such a QLS. If G is a finite solvable group, then there exists a composition series of G

$$G = G_0 \geq G_1 \geq ... \geq G_s = \{1\}$$

such that: G_{i-1}/G_i are cyclic groups, whose prime orders, which are denoted p_i. That means, there exists a generator $\bar{a_i} = a_iG_i$ in G_{i-1}/G_i, and for every element $\bar{g} = gG_i$ in G_{i-1}/G_i, there exists an integer k_i such that $1 \leq k_i \leq p_i$ and $\bar{a_i}^{k_i} = \bar{g}$, Then we can express k_i in base 2

$$k_i = b_{i,0} + b_{i,1}2^1 + ... + b_{i,\lceil \log_2 p_i \rceil}2^{\lceil \log_2 p_i \rceil}$$

where $\lceil x \rceil$ is the ceiling function. Hence,

$$\bar{g} = \bar{a_i}^{k_i} = (\bar{a_i})^{b_{i,0}}(\bar{a_i})^{b_{i,1}2^1}...(\bar{a_i})^{b_{i,\lceil \log_2 p_i \rceil}2^{\lceil \log_2 p_i \rceil}}(b_{i,j} \in \{0,1\})$$

That means, there exists $g_i \in G_i$ such that:

$$g = (a_i)^{b_{i,0}}(\bar{a_i})^{b_{i,1}2^1}...(a_i)^{b_{i,\lceil \log_2 p_i \rceil}2^{\lceil \log_2 p_i \rceil}}g_i$$

Using this method, we can prove the following

Proposition 3.2. *If G is a finite solvable group, then there exists a QLS of G.*

Proof. We can denote the composition series of G as above, and $\bar{a}_{i-1} = a_iG_i$ is the generator of G_{i-1}/G_i. Then for any g in G, there exists g_1 in G_1, such that:

$$g = (a_0)^{b_{0,0}}...(a_0)^{b_{0,\lceil \log_2 p_0 \rceil}2^{\lceil \log_2 p_0 \rceil}}g_1$$

Similarly, there exists $g_2 \in G_2$ such that

$$g_1 = (a_1)^{b_{1,0}}...(a_1)^{b_{1,\lceil \log_2 p_1 \rceil}2^{\lceil \log_2 p_1 \rceil}}g_2$$

This process will end after s steps, and we can express:

$$g_s = (a_s)^{b_{s,0}}...(a_s)^{b_{s,\lceil \log_2 p_s \rceil}2^{\lceil \log_2 p_s \rceil}}$$

Now, if we denote $A_i = \{a_i, a_i^2, ...a_i^{\lceil \log_2 p_i \rceil}\}$ then $\alpha = \{A_1, ..., A_s\}$ is the QLS. □

Actually, at each step to express g, we have to solve the DLP on the group G_{i-1}/G_i, whose prime order. And it can be done very fast if we choose G_{i-1}/G_i having small order, and an elementary abelian group, whose order is p^n, where p is a small prime, can be applied for this case. We call such QLS constructed by this way *transversal quasi-logarithmic signature*(TQLS).

Now, we can see that the group structure in the TQLS is completely broken, because A_s is not a group anymore. Besides, our TQLS did not contain the identity element. And hence, the attack applied to TLS may not be applied to TQLS, including the algorithms of Svaba et al. in [10]. In terms of the storage cost, for TLS, we need to keep p_i elements, which are representatives of the quotient group G_{i-1}/G_i, but for TQLS, we just need to save $\mathrm{O}(\log_2 p_i)$ elements.

To make QLS become fused quasi-logarithmic signatures (FQLS), which is similar to the way of building FTLS from given logarithmic signatures, we assume that G is a group and $\alpha = A_1, ..., A_s$ is the QLS of G, such that: $A_i \cap A_j = \emptyset$, for all $i \neq j$. We can do this easily from the different choices of coset representatives of TQLS. Let us define the actions on α, which are also called *fused operations.*

(1) **Block concatenation.** Replace two blocks A_i, A_j by their union $A_i \cup A_j$.
(2) **Element permutation.** Permute elements within block A_i.
(3) **Block separation.** Divide a block A_i into disjoint smaller block $A_{i,1}, ..., A_{i,k}$ and $\cup_{j=1}^{k} A_{i,j} = A_i$.
(4) **Block replacement.** Replace block A_i with gA_i where g is an element of G, such that: $gA_i \cap A_j = \emptyset$, for all $i \neq j$.

There are some differences between fused operations of logarithmic signature and QLS. For logarithmic signature, we used the block mix in their fused operations, that means, replacing A_i, A_j with $A_iA_j = \{a_ia_j | a_i \in A, a_j \in A_j\}$, because by this way, it preserves the logarithmic signatures structure, i.e. after applying finite fused operation, we will obtain another logarithmic signature β. But for QLS, we need to use the disjoint union of sets to preserve the QLS structure as the following proposition points out. And from this, the block permutation in logarithmic signatures is not necessary for the fused operations of QLS, because a permutation of blocks $A_1, ..., A_s$ is actually a specific case of permutation of elements within the block $\cup_{i=1}^{k} A_i$. Instead of block permutation, we propose another operation, that is block separation. Finally, the block replacement is also modified to be suitable for the structure of QLS, because we need to keep the condition $A_i \cap A_j = \emptyset$, for the next round (if necessary). Note that the choice of such g in the fourth operation is possible, because the size of each block A_i is very small compared to G (about $\log \log |G|$). To see how these operations preserve the QLS structure, let us prove the following proposition.

Proposition 3.3. *Given a QLS α of an abelian group G, then β obtained after applying finite fused transformations is also QLS.*

Proof. The proposition is obvious to the fused operations (2) and (3) because we do nothing but renumbering our blocks and elements. For operation (1), if the factorization of h by QLS α contains some elements from A_i, and some elements from A_j, then we can choose these elements in the union $A_i \cup A_j$, because $A_i \cap A_j = \emptyset$, and this choice is unique. Hence,

$$\beta = \{A_1, ..., A_{i-1}, A_i \cup A_j, A_{i+1}, ..., A_{j-1}, A_{j+1}, ..., A_s\}$$

is also a QLS. About operation (4), for any $h \in G$, we can express $g^{-1}h$ uniquely by the QLS α, that means, we can uniquely express h by $\beta = \{A_1, ..., gA_i, ..., A_s\}$. Therefore, β is also QLS. □

In the proposition above, for simplicity, we just consider the case G is an abelian group. If G is not abelian, we will change a little bit the block replacement, which replaces A_i with $g_iA_ig_{i+1}^{-1}$, as long as $g_iA_ig_{i+1}^{-1} \cap A_j = \emptyset$ for all $i \neq j$. The proposition also lets us know that after applying finite fused operations on TQLS α, we also obtain a QLS β, which is called *fused transversal quasi-logarithmic signatures* (FTQLS). As mentioned earlier, we can build such QLS easily from the composition series of elementary abelian groups. And with the similar ideas to logarithmic signatures in [9], we can build such a cryptosystem based on the concept of QLS. The main advantage of this concept is that we can avoid known attacks for cryptosystems on logarithmic signatures, and reduce the storage cost. For brute force attack, the attacker has to know at a very least the permutation at our first fused operations, and it takes O$((\log|G| \log\log|G|)!)$. Our next sections will be devoted to discuss another ideas, which is based on the logarithmic signatures on cyclic 2-groups with some interesting examples.

4. The DLP on finite cyclic 2-groups

Let G be a finite group and $\alpha = \{A_1, ..., A_s\}$ is the TLS of G. With the same notations in section 2, where we built such TLS, in [10], under the assumption "There exists an algorithm in polynomial time such that for any $g_i \in G_i$, we can find $a_{i,k_i} \in A_i$ such that $g_i \in a_{i,k_i}G_{i+1}$" the authors pointed out any transversal logarithmic signatures are tame.

Theorem 4.1 (**10-Theorem 3.1**). *Let G be a finite abelian group, then any transversal logarithmic signatures are tame.*

Generally, it is not easy to satisfy the condition the authors assume above because it is equivalent to deciding whether $a_{i,k_i}^{-1}g_i \in G_{i+1}$ or not, i.e. it is the membership problem, which has not been solved in general cases [10]. However, we can choose some specific groups in which this assumption is easy to satisfy, such as groups, whose orders consist of small prime factors, and cyclic 2-groups are examples. We first prove the following

Proposition 4.2. *Let G be a cyclic 2-group, whose order is 2^s, and g is the generator of G, then the set $\alpha = \{A_0, ..., A_{s-1}\}$ is the TLS of G, where $A_i = \{1, g^{2^i}\}(i = 0, ..., s-1)$*

Proof. Let $G_i = \langle g^{2^i} \rangle$, then it is easy to see that

$$G = G_0 \geq G_1 \geq ... \geq G_{s-1} = \{1\}$$

is the normal series of G and $G_i/G_{i+1} = G_{i+1}, g^{2^i}G_{i+1}$. Applying the result from the first section, we can see that α is the transversal logarithmic signatures of G. □

Based on Proposition 4.2, for any $x \in G$, we can express x uniquely in the form

$$x = \prod_{i=0}^{s-1} g^{a_i 2^i} \tag{1}$$

where $a_i = 0$ or 1. If we can know exactly the value of a_i, then we can solve the DLP on this group. It can be stated in the following theorem

Theorem 4.3. *Let G be a cyclic group of order 2^s. Then the DLP of G can be solved in polynomial time.*

Proof. Let g be the generator of G and $x \in G$. We now find $k \in \mathbb{Z}$ such that $g^k = x$. Based on equality (1), we can express x in the form

$$x = \textstyle\prod_{i=0}^{s-1} g^{a_i 2^i} = g^{a_0} g^{a_1 2} ... g^{a_{s-1} 2^{s-1}} (a_i \in \{0, 1\})$$

Firstly, to determine a_0, we have to know whether $x \in \langle g^2 \rangle$ or not. By basic facts from algebra, it can be checked when we compare $x^{2^{s-1}}$ to 1. If $x^{2^{s-1}} = 1$, then $x \in \langle g^2 \rangle$, and $a_0 = 0$. Otherwise, $a_0 = 1$.

From the expansion of x, we can see $\frac{x}{g^{a_0}}$ is in $\langle g^2 \rangle$, and $\frac{x}{g^{a_0}}$ lies either in $\langle g^4 \rangle$ (when $a_1 = 0$) or $g^2 \langle g^4 \rangle$ (when $a_1 = 1$). Because G is cyclic, this can be checked by comparing $(\frac{x}{g_0^a})^{2^{s-2}}$ to 1.

Similarly, to determine a_i from known values $a_0, ..., a_{i-1}$, we need to compute $(\frac{x}{g^{a_0} g^{a_1 2} ... g^{a_{i-1} 2^{i-1}}})^{2^{s-(i+1)}}$ and then comparing it to 1. If the equality holds, then $a_i = 0$. Otherwise, $a_i = 1$.

Continuing this step to $i = s - 1$, we can obtain the value of $k = \sum_{i=0}^{s-1} a_i 2^i$. Each of the step can be done in polynomial time. Therefore, the DLP on such groups can be solved efficiently. □

It is worth noting that our proof implies an algorithm and we can use this method to solve the DLP on groups whose order consist of small prime factors. And in the next section, we will give non-trivial examples of cyclic 2-groups, that can be used for our proposed cryptosystem in the last section.

5. Examples

5.1. **Cyclic 2-groups on Pell's curves.** The Pell's curves that we mention here is the set of all roots of the Pell's equation over finite fields. In [5-Theorem 6], the authors proved that the DLP on Pell's curves over finite fields can be reduced to the DLP on finite fields in polynomial time. We will prove that the DLP on Pell's curves over $\mathbb{F}_p$ where p is a Mersenne prime can be solved in polynomial time, because they are cyclic 2-groups, as theorem 4.3 pointed out. We first recall the definition and properties of Pell's curves over finite fields.

Definition 5.1. Let p be a prime, and D in $\mathbb{F}_p$ is non-quadratic residue modulo p, then the *Pell's curve* over $\mathbb{F}_p$ is the set

$$P(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 | x^2 - Dy^2 = 1\}$$

It is easy to see that $(1, 0) \in P(\mathbb{F}_p)$. For all $(x_i, y_i) \in \mathbb{F}_p (i = 1, 2)$, the addition law in $P(\mathbb{F}_p)$ can be defined as follows.

$$(x_1, y_1) + (x_2, y_2) = (x_1x_2 + Dy_1y_2, x_1y_2 + x_2y_1)$$

Under this addition, $P(\mathbb{F}_p)$ forms an abelian group whose $(1, 0)$ is neutral element. In [5-Theorem 5], the authors pointed out that $P(\mathbb{F}_p) \cong \mathbb{Z}_{p+1}$. When $p = 2^s - 1 (s > 2)$ is a Mersenne prime, we have $P(\mathbb{F}_p) \cong \mathbb{Z}_{2^s}$, that means, it is a cyclic 2-group. According to Theorem 4.3, we can infer following theorem.

Theorem 5.2. *Let $p > 3$ be a Mersenne prime, and D is non-quadratic residue modulo p, then the DLP on the Pell's curve can be solved in polynomial time.*

5.2. Cyclic 2-group on elliptic curves.

The use of elliptic curves over finite fields in cryptography was first introduced by N. Koblitz [4] and V. Miller [5], who proposed a cryptosystem based on the DLP on the elliptic curves.

Definition 5.3. Let $\mathbb{F}_q$ be the finite field, $\text{char}(\mathbb{F}_q) \neq 2, 3$, and $a, b \in \mathbb{F}_q$ such that $4a^3 + 27b^2 \neq 0$. An *elliptic curve* over $\mathbb{F}_q$ is the set

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q^2 | y^2 = x^3 + ax + b\} \cup \{\infty\}$$

where ∞ is the point at infinity.

With the addition law defined in [11-Theorem 2.1], $E(\mathbb{F}_p)$ forms an abelian group, whose neutral element is ∞. According to the addition law, the point of order 2 of $E(\mathbb{F}_q)$ has the form $(x, 0)$.

In 1991, Menezes-Okamoto and Vanstone [6] introduced a algorithm to reduce the DLP on *supersingular elliptic curves* to the DLP on finite fields by Weil pairings. An elliptic curve we mention here is also a supersingular elliptic curve, which play an important role in cryptography and it is often used in applications relating to pairing-based cryptography (the readers can refer [2] or [1-Chapter 5]). Let $p > 3$ be prime and $p \equiv 3 \pmod 4$, we consider the elliptic curve

$$E(\mathbb{F}_p) : y^2 = x^3 + x$$

It is a supersingular elliptic curve, which means $\#E(\mathbb{F}_p) = p + 1$ [3-Table 1]. This curve has only one point of order 2 as the following lemmas points out.

Lemma 5.4. *The equation $x^2 + 1 = 0$ has no solution on the field $\mathbb{F}_p$ where $p \equiv 3$ (mod 4).*

Proof. It is obvious from the Euler's criterion. □

Lemma 5.5. *There exists only one point of order 2 on the elliptic curve defined above.*

Proof. The point of order 2 will have the form $(x, 0)$, i.e. they are roots of the equation $x^3 + x = 0$. And it is equivalent to $x(x^2 + 1) = 0$. If $x = 0$, the point $(0, 0)$ has order 2 on the curve. On the other hand, according to Lemma 5.4, the equation $x^2 + 1 = 0$ has no solution on $\mathbb{F}_p$. Hence, there exists only one point of order 2 on this curve. □

Additionally, from the result of Schoof [8- Theorem 4.8], we know either $E(\mathbb{F}_p) \cong \mathbb{Z}_2 \oplus \mathbb{Z}_{\frac{p+1}{2}}$ or $E(\mathbb{F}_p) \cong \mathbb{Z}_{p+1}$. In the first case, $E(\mathbb{F}_p)$ has more than one point of order 2, and it contradicts to the Lemma 5.5. Therefore, $E(\mathbb{F}_p) \cong \mathbb{Z}_{p+1}$, and it is a cyclic group. In particular, when $p > 3$ is a Mersenne prime, we can write $p = 2^s - 1$, and $E(\mathbb{F}_p) \cong \mathbb{Z}_{2^s}$ Thus, it is a cyclic 2-group. Again, using theorem 4.3, we have the following theorem.

Theorem 5.6. *Let $p > 3$ be a Mersenne prime and $p = 2^s - 1$, then the DLP on the elliptic curve $E(\mathbb{Z}_p) : y^2 = x^3 + x$ can be solved efficiently.*

6. A cryptosystem based on the ease of the DLP

In this section, we will propose a cryptosystem on a finitely cyclic 2-group, where the DLP is easy. The idea is based on the cryptosystem MST3 mentioned in [9].

6.1. **FTLS of finite abelian groups.** We first prove the following

Theorem 6.1 (10-Theorem 4.5). *Let $\beta = \{B_0, B_1, ..., B_t\}$ be the FTLS constructed from TLS $\alpha = \{A_0, A_1, ..., A_s\}$ of an abelian group G using four operations in Definition 2.2, then β is tame if the trapdoor information is known.*

Remark. We will give the independent proof, which implies an algorithm we use in the decryption process of the cryptosystem proposed below.

Proof. From α we can use four fused operations mentioned in section 2 to obtain its FTLS β. Denote $\alpha = \{A_0, A_1, ..., A_s\}, \beta = \{B_0, B_1, ..., B_t\}$, where $A_i = \{a_{i,1}, ..., a_{i,r_i}\}, B_j = \{b_{j,1}, ..., b_{j,s_j}\}$.

Given $g \in G$, we can uniquely factorize $g = a_{0,k_0}...a_{s,k_s}$ where $a_{i,k_i} \in A_i$. Our task is to give the factorization of G by FTLS β when four fused operations are known. We now consider four cases, which corresponds to four transformations of α.

Case 1. If we permute s block of α by the permutation π in S_s to obtain β, where S_s is the permutation of a set whose s elements, i.e. $B_i = A_{\pi(i)}$ then a_{i,k_i} is actually $b_{\pi^{-1}(i),k_i}$.

Case 2. If the elements within a block A_i is permuted to obtain a block B_i, then there exists a permutation ϕ of the set $\{1, ..., r_i\}$ such that

$$A_i = \{a_{i,1}, ..., a_{i,r_i}\} \to B_i = \{a_{i,\phi(1)}, ..., a_{i,\phi(r_i)}\} = \{b_{i,1}, ..., b_{i,r_i}\}$$

That means $b_{i,j} = a_{i,\phi(j)}$. Hence, $a_{i,j}$ will be replaced with $b_{i,\phi^{-1}(j)}$ in the factorization of g.

Case 3. Replace a block A_i with $B_i = A_i h$, where $h \in G$, then we can express gh^-1 as follows

$$gh^-1 = a_{0,k'_0}...a_{i,k'_i}...a_{s,k'_s} \Rightarrow g = a_{0,k'_0}...(a_{i,k'_i}h)...a_{s,k'_s}$$

And we can replace a_{j,k_j} with $a_{j,k'_j}(= b_{j,k'_j})(j \neq i)$, and a_{i,k_i} with $a_{i,k'_i}h(= b_{i,k'_i})$.

Case 4. Replace two block $A_i, A_j (i < j)$ with a single block $B_i = A_i A_j = \{xy | x \in A_i, y \in A_j\}$, and remove both blocks A_i, A_j to obtain β. Then we can see that the number of blocks will be reduced 1. Because G is an abelian group, we have

$$\begin{gathered} g = a_{0,k_0}...a_{i,k_i}...a_{j,k_j}...a_{s,k_s} = \\ a_{0,k_0}...a_{i-1,k_{i-1}}(a_{i,k_i}a_{j,k_j})a_{i+1,k_{i+1}}...a_{j-1,k_{j-1}}a_{j+1,k_{j+1}}...a_{s,k_s} = \\ b_{0,k_0}...b_{i-1,k_{i-1}}(a_{i,k_i}a_{j,k_j})b_{i+1,k_{i+1}}...b_{j-1,k_{j-1}}b_{j+1,k_{j+1}}...b_{s,k_s} \end{gathered}$$

That means, we can replace a_{u,k_u} with $b_{u,k_u}(0 \leq u \leq i-1, i+1 \leq u \leq j-1)$, a_{v,k_v} with $b_{v-1,k_v}(j+1 \leq v \leq s)$ and replace $a_{i,k_i}a_{j,k_j}$ with $b_{i,t}$ in B_i where t can be obtained by suitable numbering.

Through four cases, we can see that any FTLS of α are tame if α is tame and the information about transformations are known. □

From proposition 4.2 and theorem 6.1, we obtain the following

Corollary 6.2. *Let G be a finite cyclic 2-group, whose order 2^s, and g is the generator of G, then any FTLS of α is tame if the trapdoor information is known, where $\alpha = \{A_0, ..., A_{s-1}\}, A_i = \{1, g^{2^i}\}$.*

6.2. **A proposed cryptosystem.** Based on theorem 6.1, we know that if the trapdoor information is not given, then the FTLS obtained from a TLS by applying four transformations is not tame. And to avoid the attack from [10], we can also hide the information about the FTLS. Using this remark, we will propose a cryptosystem as follows.

Key generation

Step 1. Alice will public a large Mersene prime and an elliptic curve

$$E(\mathbb{F}_p) : y^2 = x^3 + x$$

From Section 5.2, we know that $E(\mathbb{F}_p)$ is a cyclic group of order 2^s.

Step 2. She chooses the generator P of $E(\mathbb{F}_p)$ and obtain the TLS

$$\rho = \{P_0, ..., P_{s-1}\} \text{ where } P_i = \{P_{i,1}, P_{i,2}\} = \{\infty, 2^i P\}$$

Step 3. At this step, she uses four fused operations defined in Definition 2.2 to obtain the FTLS of ρ, called α, where

$$\alpha = \{A_0, ..., A_t\} \text{ where } A_i = A_{i,1}, ..., A_{i,r_i}$$

Step 4. Alice chooses $B_i \subset E(\mathbb{F}_p)$ randomly such that $|B_i| = |A_i| = r_i(i = 0, ..., t)$ and points of B_i are denoted $B_{i,j}(j = 1, ..., r_i)$. Let $\beta = \{B_1, ..., B_t\}$ and she defines the map $\breve{\beta}$ as follows

$$\begin{array}{rccc} \breve{\beta}: & \mathbb{Z}_{r_0} \times ... \times \mathbb{Z}_{r_t} & \to & B_0 + B_1 + ... + B_t \\ & (k_0, ..., k_t) & \mapsto & B_{0,k_0} + B_{1,k_1} + ... + B_{t,k_t} \end{array}$$

Step 5. Alice chooses $(t+2)$ points $T_i \in E(\mathbb{F}_p)(i = 0, ..., t+1)$ secretly.

Step 6. She then chooses $G_i \subset E(\mathbb{F}_p)(0 \leq i \leq t)$ such that $|G_i| = |B_i| = |A_i| = r_i$ and G_i consists of r_i points $G_{i,j}(j = 1, ..., r_i)$ that satisfy

$$G_{i,j} = (T_i - T_{i+1}) + B_{i,j} + A_{i,j}$$

and the map

$$\begin{array}{rccc} \breve{\gamma}: & \mathbb{Z}_{r_0} \times ... \times \mathbb{Z}_{r_t} & \to & G_0 + G_1 + ... + G_t \\ & (k_0, ..., k_t) & \mapsto & G_{0,k_0} + G_{1,k_1} + ... + G_{t,k_t} \end{array}$$

After six steps, she will

- Public the information $(E(\mathbb{F}_p), \beta, \breve{\beta}, \gamma, \breve{\gamma})$
- Keep (ρ, α, T_i) secret together with the trapdoor information.

Encryption

Step 1. Bob wants to encrypt data $M \in E(\mathbb{F}_p)$.

Step 2. He chooses $r = (k_0, ..., k_t) \in \mathbb{Z}_{r_0} \times ... \times \mathbb{Z}_{r_t}$ and then compute

$$C_1 = \breve{\beta}(r) + M, C_2 = \breve{\gamma}(r) + M$$

Step 3. He sends (C_1, C_2) to Alice.

Decryption

Step 1. Alice computes

$$C = C_2 - C_1 = \breve{\gamma}(r) - \breve{\beta}(r) = T_0 - T_{t+1} + \sum_{i=0}^{t} A_{i,k_i}$$

Step 2. Alice then compute

$$D = C - T_0 + T_{t+1} = \sum_{i=0}^{t} A_{i,k_i}$$

Step 3. She uses the ease of the DLP (theorem 4.3) to recover D from the TLS ρ

$$D = \sum_{i=0}^{s-1} P_{i,k_i}$$

Step 4. By Corollary 6.2 and the algorithm implied in the proof of Theorem 6.1, she can obtain the factorization of D into TLS α by using trapdoor information

$$D = \sum_{i=0}^{t} A_{i,k_i}$$

and obtain the value $r = (k_0, .., k_t) \in \mathbb{Z}_{r_0} \times ... \times \mathbb{Z}_{r_t}$.

Step 5. She can recover M by computing

$$M = C_1 - \breve{\beta}(r)$$

6.3. **Security Issues.** The reason why we choose the cyclic 2-group on elliptic curves is that the addition law in these curves is much more complicated than that in $\mathbb{Z}_{2^s}$, but the computing time for point addition is acceptable. The security of our scheme is discussed in more details in [9], where the author first use the concept of logarithmic signatures to build a cryptosystem based on Suzuki's 2-groups. We will briefly mention some similar cases here, because our contribution is to strengthen the security in the abelian cases, and take advantage of the ease of the DLP and the tameness of FTLS for designing trapdoor information.

For brute force attack, the attacker Eve has to know about the TLS α, but at least, it will require the complexity time of $\mathrm{O}(s!)$ (if we only use the first transformation when permuting s blocks). Another attempt is trying to guess P or ρ is also impossible, because we have 2^{n-1} generators for a cyclic group of order 2^n.

Our β is chosen randomly, and each B_i has the same element as A_i. Hence, there is no relation between them. In contrast, there are some connections between $B_{i,j}$, $A_{i,j}$ and $G_{i,j}$ through the system of equations

$$G_{i,j} - B_{i,j} = T_i - T_{i+1} + A_{i,j}(0 \leq i \leq t, 1 \leq j \leq r_i)$$

The values of all $G_{i,j}$ and $B_{i,j}$ are public, hence to recover T_i and $A_{i,j}$, Eve must solve a system of $(r_0 \times ... \times r_t)$ equations and $(r_0 \times ... \times r_t) + (t+2)$ unknowns. Hence, it is infeasible to recover T_i and $A_{i,j}$ from $B_{i,j}$ and $G_{i,j}$.

To read message M, he can get started from the pair (C_1, C_2), and he can compute the point $C = C_2 - C_1 = T_0 - T_{t+1} + \sum_{i=0}^{t} A_{i,k_i}$, but for next steps, he cannot know the two points T_0 and T_{t+1} because of the reasons mentioned above.

In the worst case, if he know the information about T_0 and T_{i+1}, then he will know the point $D = \sum_{i=0}^{t} A_{i,k_i}$. It is worth mentioning that our decryption process is done via the diagram:

$$\begin{tikzcd}
& \text{Sum of } 2^iP \arrow[dr, "\text{trapdoor inf.}"] & \\
D \in E(\mathbb{F}_p) \arrow[ur, "\text{ease of DLP}"] \arrow[rr] & & \text{Sum of } A_{i,j}
\end{tikzcd}$$

That means, if Eve want to read M, he must find another ways to recover the generator P or $A_{i,j}$. The diagram above also visualize the one-way function in our case, because compute sum of $A_{i,j}$ is easy, but for the converse, one has to meet much more difficulty. And to avoid attack from [10], because the logarithmic signatures ρ is easy to realize, we have to use four transformations secretly to make it fused and also hide the obtained FTLS α.

6.4. **Implement.** For implementing, we choose $p = 2^{527} - 1$ is a Mersenne prime and the curve:

$$E(\mathbb{F}_p) : y^2 = x^3 + x$$

We also run the proposed cryptosystem along with two versions of the RSA cryptosystems, RSA-1024 and RSA-2048, and compare the execution time of three systems in three phases: key generation, encryption, and decryption.

Computer Configuration
Processor: Intel(R) Core(TM) i5- 2430M CPU@ 2.40GHz.
RAM: 4.00 GB(2.98 usable).
System type: 32-bit Operating System.
OS: Windows 7 Ultimate, SP1.

Virtual Machine
Oracle VM Virtual Box 4.2.18˙ SAGE- 5.11.
OS: Fedora.
Base Memory: 512 MB.
Storage: 7.81 GB.

Execution time
The table below shows the execution time of our proposed system.

	Key generation	**Encryption**	**Decryption**
Proposed-system	**14.1740s**	**0.0730s**	**15.7800s**

The decryption phase is often slower than the encryption, because our proposed system has to solve a weak DLP in Step 3, and in Step 4, we have to recover the indexes from trapdoor information. And by the implement experiments, our proposed system above can be applied to the system of digital signatures in which verifications are performed more frequently.

7. Conclusion

In this paper, we have generalized the definition of logarithmic signature to give a concept of quasi logarithmic signatures in section 3, and from this, defining the new operations to preserve the QLS structure. These ideas can be applied to build a new cryptosystem, which is similar to the concept of MST3, and probably avoid the known attacks on cryptosystems based on logarithmic signatures, including the factorization of abelian groups from [10] and to reduce the storage memory of trapdoor information. Besides, using the concepts of logarithmic signatures, we have proposed an algorithm to solve the DLP on cyclic 2-groups efficiently. This can be generalized for cyclic groups whose orders are products of small primes. We have also pointed out two nontrivial groups, including cyclic 2-group on Pell's curves and elliptic curves, as feasible examples of our result. And using the ease of the DLP on such cyclic 2-groups, we have proposed a cryptosystem based on the concept of logarithmic signatures, and that can also avoid the attack from [10]. The security for our scheme is also discussed in this paper. For implementing, we choose an elliptic curve of $y^2 = x^3 + x$ over $\mathbb{F}_p$, where $p = 2^{527} - 1$, and the running speed is acceptable.

References

[1] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman 2008. An Introduction to Mathematical Cryptography, *Springer* (2008), 315-334

[2] Antoine Joux 2014. A One Round Protocol for Tripartite Diffie-Hellman, *Springer-Verlag Berlin-Heidelberg* (2014) 17, 263-276

[3] Antoine Joux and Kim Nguyen 2003. Separating Decision Diffie-Hellman from Diffie-Hellman in cryptographic groups, *Journal of Cryptography* (2003) 16, 239-247

[4] Neal Koblitz 1987. Elliptic Curve Cryptosystems, *Mathematics of Computation*, Volume 48 (1987), 203-209

[5] V. Miller 1986. Uses of Elliptic Curves in Cryptography, *Advances in Cryptography, Proceeding of Crypto'85, Lecture Notes in Computer Science*, 218 (1986), Springer-Verlag, 417-426

[6] Alfred J. Menezes, Tatsuaki Okamoto, and Scott Vanstone. Reducing Elliptic Curve Logarithms to Logarithms in a Finite Fields, *STOC '91 Proceedings of the twenty-third annual ACM symposium on Theory of computing* (1991), 80-89

[7] Alfred J. Menezes and Scott A. Vanstone 1992. A Note on Cyclic Groups, Finite Fields, and the Discrete Logarithm Problem, *AAECC* 3 (1992), 67-74

[8] R.Schoof 1987. Nonsingular Plane Cubic Curves over Finite Fields, *Journal of Combinatorial Theory*, A46 (1987), 183-211

[9] Pavol Svaba and Tran Van Trung 2010. Public Key Cryptosystem MST3: Cryptanalysis and realization, *J. Math. Cryptol.*, 271-315

[10] Pavol Svaba, Tran Van Trung, and Paul Wolf 2013. Logarithmic Signatures for Abelian Groups and Their Factorization, *Tatra Mt. Math. Publ.* 57 (2013), 21-33

[11] Lawrence C. Washington 2008. Elliptic Curves: Number Theory and Cryptography, 2^{nd} edition, *Chapman & Hall/CRC*, 12-18