Use of Sparse and/or Complex Exponents in Batch Verification of Exponentiations

Jung Hee Cheon[1] and Dong Hoon Lee[2]

[1] Department of Mathematics, Seoul National University
jhcheon@math.snu.ac.kr,
[2] National Security Research Institute (NSRI)
dlee@etri.re.kr

Abstract. Modular exponentiation in an abelian group is one of the most frequently used mathematical primitives in modern cryptography. *Batch verification* is to verify many exponentiations simultaneously. We propose two fast batch verification algorithms. The first one makes use of exponents with small weight, called *sparse exponents*, which is asymptotically 10 times faster than the individual verification and twice faster than the previous works without security loss. The second one is applied only to elliptic curves defined over small finite fields. Using sparse Frobenius expansion with small integer coefficients, we propose a complex exponent test which is four times faster than the previous works. For example, each exponentiation in one batch requires asymptotically 9 elliptic curve additions in some elliptic curves for 2^{80} security.

Keywords: Batch verification, modular exponentiation, sparse exponent, Frobenius map.

1 Introduction

Batch verification is an algorithm to verify many exponentiations simultaneously: Let G be an abelian group with a generator g. Given a batch instance of n exponentiation pairs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ with $x_i \in \mathbb{Z}$ and $y_i \in G$, the algorithm checks at once if $g^{x_i} = y_i$ for all i. The small exponent test is to check if $g^{\sum_{i=1}^{n} x_i s_i} = \prod_{i=1}^{n} y_i^{s_i}$ for randomly chosen ℓ bit integers s_i when the screening parameter is ℓ [2]. We extend the small exponent test to more general exponent sets and improve the performance using *sparse exponents*.

Our idea is to take random exponents which have smaller Hamming weights and longer bit sizes. The probability that a wrong batch instance passes the proposed test with an exponent set S is at most $1/|S|$, as we will show, when all elements in S are distinct modulo the order of a given group. Therefore it is important how to take a exponent set S and how to maximize the cardinality of elements that are distinct modulo the order of the group.

If we take S to be a set of non-negative integers in $\mathbb{Z}_p$ with Hamming weight $k \leq 19$ from $\mathbb{Z}_p$ for an 160 bit prime p, $|S|$ is greater than 2^{80}. In this case, our sparse exponent test requires at most $208 + 19n$ multiplications to verify n exponents with security 2^{80}, while the small exponent test requires $272 + 40n$ multiplications at average. Another variant of the algorithm is to use sparse signed binary exponents. It is proper for verifying scalar multiplications on elliptic curves where a subtraction is as efficient as an addition. More savings are obtained by using even larger exponents than the order of the underlying group. It is applicable to signature schemes the order of whose underlying group is secret such as RSA and Guillou-Quisquater (GQ) signatures [9].

We can generalize the exponents set to complex endomorphisms in elliptic curves. When an elliptic curve is defined over a small finite field $\mathbb{F}_q$, we have a very efficient complex endomorphism, called the Frobenius map. We take as an exponent set S a polynomial of the Frobenius map with small integer coefficients. In most previous works on Frobenius expansions, the set of coefficients have at most q cardinality due to the uniqueness of representation (For more references, refer to [14]). If we consider this coefficient set, we have the same efficiency with the sparse exponent set. In our method, however, we enlarge the coefficient set to all integers whose absolute values are less than $q^2/2$ and coprime to q and prove that each element of S is distinct as an endomorphism on the cyclic subgroup of the elliptic curve if each element of S has no adjacent non-zero coefficients. As a result, we obtain the batch verification four times faster than the previous works. For example, each exponentiation requires asymptotically 9 elliptic curve additions in Koblitz curves for 2^{80} security.

Batch Cryptography Batch cryptography was first introduced by Fiat [8]. He introduced an algorithm to obtain two exponentiations by one full exponentiation and several small exponentiations. More techniques on batch computations can be found in [3, 12].

Batch verification of exponentiations was first proposed by Naccache *et al.* [15] to verify efficiently modified DSA signatures. They used the subset consisting of e-bits prime numbers for small e as an exponent set. Yen and Laih [18] improved the test by adopting small exponents set $\{0, 1\}^{\ell}$ for the screening parameter ℓ. Bellare *et al.* [2] gave systematic approaches with two more tests: the atomic test uses $S = \{0, 1\}$ and the bucket test is the combination of the atomic test and the small exponent test. Later, Boyd and Pavlovski [4] indicated the weakness of the tests when they are used in non-prime order subgroups.

Current batch verifications are efficient for verifying many signatures by one signer at once. The applications include: 1)an authenticated database in which each data is signed by a data owner [13], 2) electronic commerce where shops or banks need to verify all the coins, usually issued by one bank, very efficiently, and 3) voting protocols where the tally needs to authenticate huge number of votes and an effective vote should be accompanied by the signature signed by the voting center.

Organization This paper is organized as follows: In Section 2, we introduce general approach for batch verifications and discuss its security. In Section 3, we propose a batch verification algorithm using sparse binary exponents and its variants using sparse signed binary exponents and sparse long exponents. In Section 4, we consider complex exponents on elliptic curves. We conclude in Section 5.

2 Batch Verification

For the definition of batch verification, we will follow the notation in [2]. Let G be a cyclic group of order p with a generator g and $R(\cdot)$ a Boolean relation on the set of instances $\{(x,y)|x \in \mathbb{Z}_p, y \in G\}$. We say that $R(x,y) = 1$ or the instance (x,y) is correct if and only if $g^x = y$. A batch instance for relation R is a sequence $inst_1, \cdots, inst_n$ of instances for R. We say that the batch instance is correct $(R(inst_1, \ldots, inst_n) = 1)$ if $R(inst_i) = 1$ for all $i = 1, \ldots, n$, and incorrect $(R(inst_1, \ldots, inst_n) = 0)$ if there is some $i \in \{1, \ldots, n\}$ for which $R(inst_i) = 0$.

Definition 1. *A* batch verifier *for relation R is a probabilistic algorithm V that takes a batch instance $X = (inst_1, \ldots, inst_n)$ for R and a screening parameter ℓ, and satisfies:*

1. *If X is correct then V outputs 1.*
2. *If X is not correct then the probability that V outputs 1 is at most $2^{-\ell}$.*

In this case, we say the error of V is $2^{-\ell}$.

Now we consider a subset S of $\mathbb{Z}$ such that the difference of any two elements of S is coprime to p. Then we define a batch verifier V_S as Figure 1.

```
Input: A batch instance $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $x_i \in \mathbb{Z}_p$ and $y_i \in G$.
Check: That $g^{x_i} = y_i$ for all $1 \le i \le n$.

 1. Choose $n$ elements $s_1, \ldots, s_n$ randomly from $S$.
 2. Compute $g^{\sum_{i=1}^n s_i x_i}$ and $\prod_{i=1}^n y_i^{s_i}$.
 3. Accept if they coincide, else reject.
```

Fig. 1. The Batch Verifier V_S with a Exponent Set S

Note that any correct batch instance always passes the test, but even a wrong instance may pass the test with some probability. If we let $\alpha_i \in \mathbb{Z}_p$ be such that $g^{\alpha_i} = y_i/g^{x_i}$, the test is passed if and only if $g^{\sum_{i=1}^n s_i\alpha_i} = 1$. Thus the error of the batch verifier V_S is

$$Err(V_S) = \max_{\alpha} \frac{|\{(s_1, \ldots, s_n) \mid g^{\sum s_i\alpha_i} = 1 \text{ for } s_i \in S\}|}{|\{(s_1, \ldots, s_n) | s_1, \ldots, s_n \in S\}|}, \tag{1}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n)$ runs through all but $(0, 0, \cdots, 0)$ in $\mathbb{Z}_p^n$, if all elements of S are equally likely to be chosen in the test. Now we introduce the formula for $Err(V_S)$.

Theorem 1. *Let S be a subset of nonnegative integers such that the difference of any two elements in S is coprime to p. Then we have*

$$Err(V_S) \le 1/|S|.$$

Proof. Consider an instance of n exponentiation pairs $(x_1, y_1), \ldots, (x_n, y_n)$ with $x_i \in \mathbb{Z}_p$ and $y_i \in G$ where $y_i = g^{x_i+\alpha_i}$ for some $\alpha_i \in \mathbb{Z}_p$. Given a test parameter $(s_1, \ldots, s_n) \in S^n$, the instance passes the test if and only if $g^{s_1\alpha_1+\cdots+s_n\alpha_n} = 1$.

If the instance contains an incorrect pair, there must be at least one i such that $\alpha_i \not\equiv 0 \bmod p$. In that case, we have at most one s_i satisfying $s_i\alpha_i \equiv -\sum_{j\neq i}^n s_j\alpha_j \bmod p$, because any two elements s_i, s_i' satisfying the equation should have $\gcd(s_i - s_i', p) \neq 1$ which contradicts the assumption of S. Hence, for any given α, we have

$$|\{(s_1, \ldots, s_n) | \sum s_i\alpha_i \equiv 0 \bmod p \text{ for } s_i \in S\}| \le |S|^{n-1}.$$

From Equation (1), we have $Err(V_S) \le 1/|S|$. □

In [2], Bellare *et al.* suggested three tests. In the atomic test, S is taken to be $\{0, 1\}$. In the small exponent test, S is taken to be $\{s \in \mathbb{Z} | 0 \le s < 2^\ell\}$ for a given screening parameter ℓ. The bucket test is the combination of the atomic test and the small exponent test. In this paper, we consider different types of a set consists of *sparse exponents.*

2.1 How to Select Screening Parameters

The security level of today's signature schemes is usually set on 2^{80}, which is equivalent to that of 1024-bit RSA or 160-bit elliptic curve cryptosystem. But, the screening level of batch verifier can be different because even the owner of signing key can produce a set of wrong signatures that passes the verifiers, with at most $2^{-\ell}$ probability for the screening parameter ℓ. In [2] and [4], $\ell = 60$ was proposed for most practical use.

Usually, selection of the screening parameter depends on the situation where each application deploys. If an application performs only small number of verifications, it would be very unlikely to accept the wrong instance in its lifetime even with the small screening parameter. When we take the small screening parameter, the batch verification algorithm is more efficient. Hence it is desirable to set the screening parameter as small as possible in tolerable ranges for each application.

We may consider an iteration of several weak batch verifications to achieve stronger batch verifications. If we perform t times the batch verification with screening level ℓ independently, then the screening level becomes $t\ell$. This enables us a *parallelization* of batch verification. Moreover we can design a *cascade filter* of signatures consisting of successive weak batch verifiers. If the test fails in the first filter, the set of signatures should be moved to the step of identifying wrong signatures. If the test succeeds, it goes to the second filter and so on until the intended screening level is achieved. This would be useful for fast screening DoS attacks.

Note that when there is only one invalid signature in the set of n instances and the batch verifier fails the verification, we can identify the invalid one with at most $\lceil \log_2 n \rceil + 1$ batch verifications by binary tree search technique. If the number of wrong signatures are k, the maximum number of batch verifications becomes $k(\lceil \log_2 n \rceil - \lceil \log_2 k \rceil) + 2^{\lceil \log_2 k \rceil + 1} - 1$

3 Sparse Exponent Test

3.1 Basic Sparse Exponent Test

Let ℓ be the screening parameter and let the order of G be a prime p. In sparse exponent test, we consider the set S of exponents

$$S = \{s \in \mathbb{Z} | 0 \leq s < 2^{\lfloor \log p \rfloor}, \quad wt(s) \leq k\},$$

where $wt(s)$ is the Hamming weight of s in the binary representation. Then every element is distinct modulo p and so we have $Err(V_S) \leq 1/|S|$

by Theorem 1. To bound $Err(V_S)$ by $2^{-\ell}$, k must satisfy

$$|S| = \sum_{i=0}^{k} \binom{\lfloor \log p \rfloor}{i} \geq 2^{\ell}.$$

Table 1 gives minimum k values for various screening parameters ℓ that satisfy the above equation when $\lfloor \log p \rfloor = 160$. Observe that k is much less than $\ell/4$. In particular, $k \leq \ell/5$ for $\ell \leq 50$.

Table 1. k values for various screening parameters when $\lfloor \log p \rfloor = 160$

ℓ	10	20	30	40	50	60	70	80
k	2	3	5	8	10	13	16	19

Our sparse exponent test is described in Figure 2. In order to perform the test efficiently, we use the technique of simultaneous multiplications as usual.

```
Input: $(x_i, y_i)$ for $1 \leq i \leq n$.
Check: That $g^{x_i} = y_i$ for all $1 \leq i \leq n$.

 1. Choose $n$ random elements $s_1, s_2, \ldots, s_n$ from $S$ defined in Equation (3.1)
 2. $y \leftarrow 1$ and $y_0 \leftarrow g$.
 3. $s_0 \leftarrow -\sum_{i=1}^{n} s_i x_i \mod p$.
 4. For $j = \lfloor \log p \rfloor$ to 1, do
    (a) For $i = 0$ to $n$, $y \leftarrow y y_i$ if the $j$-th bit of $s_i$ is one.
    (b) $y \leftarrow y^2$ unless $j = 1$.
 5. Accept if $y$ is one, else reject.
```

Fig. 2. Sparse exponent test using simultaneous multiplications

Table 2 gives the average numbers of operations which are required for verifying n instances for the naïve test, the small exponent test (SE), and our sparse exponent test (SPET). Naïve test verifies n instances independently. In the table, Exp, Mul, and Sqr represent the exponentiation, the multiplication, and the squaring on G respectively. P denotes a prime field (Sqr=0.8 Mul) and B-PB and B-NB denote a binary field with a polynomial basis (Sqr=0.1 Mul) and a binary field with a normal basis (Sqr is for free), respectively. Multiplication modulo p operations are ignored in total estimations.

Table 2. Average number of operations for verifying n-instances when $\lfloor \log_p \rfloor = 160$.

		Naïve	SE	SPET(ours)
Exp		n	1	0
Mul		0	$\ell n/2$	$kn + \lfloor \log_p \rfloor/2$
Sqr		0	ℓ	$\lfloor \log_p \rfloor$
Total	P	$208n$	$10n + 224$	$3n + 208$
(in Mul)	B-PB	$96n$	$10n + 98$	$3n + 96$
$\ell = 20$	B-NB	$80n$	$10n + 80$	$3n + 80$
Total	P	$208n$	$40n + 272$	$19n + 208$
(in Mul)	B-PB	$96n$	$40n + 104$	$19n + 96$
$\ell = 80$	B-NB	$80n$	$40n + 80$	$19n + 80$

Compared with the small exponent test, our method removes one exponentiation and reduces the number of multiplications at the cost of increasing only a small number of squarings (which is independent of n). Moreover squaring is generally more efficient than multiplication [5, 10].[3] From Table 2, we can see that our test is expected to be about $2 \sim 3$ times faster than the small exponent test as n grows.

The sparse exponent test can be applied to signature schemes based on modular exponentiations. Especially it is useful for modified DSA as in [15]. But we need to take primes p and q such that $(p-1)/(2q)$ has no divisor less than q, and regard two group elements to be equal if they are the same up to the sign due to the attack by Boyd and Pavlovski [4].

3.2 Bucket Test based on Sparse Exponents

The bucket test is a combination of the atomic test and the small exponent test [2]. We can get another bucket test based on sparse exponents by replacing the small exponent test by the sparse exponent test appearing in the bucket test: Given an instance of n exponentiation pairs (x_i, y_i), n pairs are randomly partitioned into 2^m groups and the x_i's and y_i's in the same partition are added and multiplied to form a new exponentiation pair. Then the 2^m pairs are verified using the sparse exponent test with the screening parameter m. This procedure is repeated independently

[3] According to [5, 10], the speed ratio of the squaring over the multiplication is 0.8∼0.86 for prime fields and 0.1∼0.13 for binary fields.

$\lceil \ell/(m-1) \rceil$ times for the screening parameter ℓ. The m is chosen to optimize the performance of the test.

We compare the performance of the bucket test based on sparse expoents with the previous tests in Table 3. In the table, we set p to be 160 bit primes and $\ell = 60$ as in [2]. Hence $k = 13$ is enough for our test. K denotes the unit of 1000.

Table 3. Comparison of Average Numbers of Multiplications for Different Tests

n	Naïve	Small Exp	Bucket	Sparse Exp	Buc.+Spar.
5	1 K	0.41 K	4.35 K	0.245 K	2.99 K
100	20 K	3.26 K	5.78 K	1.48 K	4.08 K
200	40 K	6.26 K	7.17 K	2.78 K	5.08 K
500	100 K	15.3 K	10.8 K	6.68 K	8.08 K
1K	200 K	30.2 K	16.6 K	13.2 K	13.1 K
10K	2000 K	300 K	100 K	130.1 K	85.6 K

Observe that our sparse exponent test improves the small exponent test by a factor of about two and our bucket test based on sparse exponent test gives better performance than the original bucket test.

3.3 A Group with Secret Order: a Composite Modulus

Let N be a product of two strong primes p_N and q_N such that $p_N = 2p'_N + 1$ and $q_N = 2q'_N + 1$ for two primes p'_N and q'_N. We assume that N is hard to factorize. Then we may take even larger exponents than N for batch verification.

We consider

$$S = \{s \in [0, 2^L) | wt(s) \leq k\},$$

where L is a size of the exponent set S. We may assume that the difference of any two elements in S is coprime to $\phi(N)/4 = p'_N q'_N$. Otherwise, we can get a nontrivial divisor of $p'_N q'_N$ or a multiple of $\phi(N)$ which gives a factorization of N [11]. Hence we may assume that the batch verifier with exponent set $|S|$ has the error at most $1/|S|$ by Theorem 1.

The cardinality of $|S|$ is given by

$$|S| = \sum_{i=0}^{k} \binom{L}{i}.$$

For the screening parameter ℓ, k is the minimum integer satisfying $|S| > 2^\ell$. Table 4 gives several k values for various exponent sizes L and screening parameters ℓ.

Table 4. k values for various exponent sizes L and screening parameters ℓ

L	160	192	256	512	1024	2048
k when $\ell = 20$	4	3	3	3	3	2
k when $\ell = 40$	8	7	7	6	5	5
k when $\ell = 80$	19	18	16	13	11	9

The sparse *long* exponent test is similar to the sparse exponent test (Figure 2). The difference is that the order p of G is secret and the bit length of exponents is longer. Since p is unknown, Step 3 of Figure 2 uses normal integer operations. This is useful for verifying the modified GQ signatures. We need to modify the scheme a little bit as in the DSA scheme [15, 4]. Let N be the product of two primes p_N and q_N and e be a 160 bit integer. Take random J and compute $a \equiv J^{-e^{-1}} \mod N$. Then the public key is (N, J, e) and the private key is (p_N, q_N, a).

The GQ signature for a message $m \in \mathbb{Z}_N$ is $(r = (k^e \bmod N), \sigma = (ka^h \bmod N))$ for randomly chosen $k \in \mathbb{Z}_N$ and h a hash output of m and r. The verification is to check if $\sigma^e J^h \equiv r \mod N$. Assume we are given n signatures (m_i, r_i, σ_i). We take n random s_i from S to apply our batch verification and computes

$$\left(\prod_{i=1}^{n} \sigma_i^{s_i}\right)^e J^{\sum_{i=1}^{n} s_i h_i} \text{ and } \prod_{i=1}^{n} r_i^{s_i},$$

and verify if they are equal modulo N. It requires $\frac{(L+160)\lfloor \log n \rfloor}{2} + 2kn + 80$ multiplications and $(L + 160)\lfloor \log n \rfloor + 2L + 160$ squarings at average.

It can be also applied to the RSA, but it is efficient only when relatively large public exponent is used. Some applications may require relatively large public exponent e [17] to reduce the private exponent. Also the similar technique can be applied to signatures based on strong RSA problems. One example is one of two verification equalities in the verification in the Cramer-Shoup signature [6].

3.4 Sparse Signed Binary Exponent Tests

In elliptic curves, subtractions are as efficient as additions. In this case, we may consider signed binary representation of exponents. Let S be a subset of nonnegative integers

$$S = \{s = \sum_{i=0}^{\lfloor \log p \rfloor - 1} s_i 2^i \geq 0 | s_i = 0, \pm 1, s_i s_{i+1} = 0\},$$

where the number of nonzero a_i is at most k. That is, an element of S has a non-adjacent representation of the weight $\leq k$ and the most significant bit is one. Note that an integer has the unique non-adjacent form (NAF) [16]. By modifying the proof, we can show that all elements in S are distinct modulo p.

Lemma 1. *Any two elements in S are distinct modulo p.*

Proof. Suppose that two elements $x = \sum_{i=0}^{\lfloor \log p \rfloor - 1} x_i 2^i$ and $y = \sum_{i=0}^{\lfloor \log p \rfloor - 1} y_i 2^i$ in S are congruent modulo p and have different representation in NAF. Then two elements are between zero and $p - 1$ and so it should be identical in $\mathbb{Z}$. Let j be the first bit from the least significant bit such that $x_j \neq y_j$. Then $(x - y)/2^j \not\equiv 0 \mod 4$. This is a contradiction. □

The cardinality of S is given in the next theorem.

Theorem 2.

$$|S| = \sum_{i=1}^{k} \binom{\lfloor \log p \rfloor + 1 - i}{i} 2^{i-1} + 1.$$

Proof. We partition S into two disjoint subsets S_0 and S_1, where

$$S_0 = \{ \sum_{i=0}^{\lfloor \log p \rfloor - 1} s_i 2^i \in S | s_{\lfloor \log p \rfloor - 1} = 0\}$$

and $S_1 = S \setminus S_0$. First, we count the cardinality of S_0. We choose i positions from $\lfloor \log p \rfloor - i$ positions, each i position except the first position from the most significant bit is filled with 01 or $0\bar{1}$, and the first position is filled with 01 because an element of S should be positive. Since they are all distinct and covers all elements of weight i in S_0, we have

$$|S_0| = 1 + \sum_{i=1}^{k} \binom{\lfloor \log p \rfloor - i}{i} 2^{i-1}.$$

When we count the cardinality S_1, we first fix the first bit of each element by one. Then by similar argument but without the constraint on the first position, we have

$$|S_1| = \sum_{i=0}^{k-1} \binom{\lfloor \log p \rfloor - 1 - i}{i} 2^i = \sum_{i=1}^{k} \binom{\lfloor \log p \rfloor - i}{i - 1} 2^{i-1}.$$

Using the well-known formula $\binom{a}{b-1} + \binom{a}{b} = \binom{a+1}{b}$, we have the lemma. □

In Table 5, we compare the k values for sparse binary exponents and sparse signed binary exponents. Table 5 shows that if we use sparse signed binary exponents, then the batch verification could be more efficient when the screening parameter $\ell \geq 40$. ECDSA that is the elliptic curve analogue of DSA is a good example that admits a sparse signed binary exponent tests [7]. We have to slightly modify the ECDSA for batch verification in the same way [15]. Remark that if we take an elliptic curve whose order is prime, the Boyd and Pavlovski attack [4] can not be applied.

Table 5. k values for various screening parameters when $\lfloor \log p \rfloor = 160$

ℓ	10	20	30	40	50	60	70	80
k (Sparse Binary)	2	3	5	8	10	13	16	19
k (Sparse Signed Binary)	2	3	5	6	8	10	13	15

4 Complex Exponent Test

Consider an ordinary elliptic curve E defined over $\mathbb{F}_q$ with $\#E(\mathbb{F}_q) = q + 1 - t$ and $\gcd(q, t) = 1$. The Frobenius map ϕ is defined as follows:

$$\phi : E(\overline{\mathbb{F}}_q) \to E(\overline{\mathbb{F}}_q); (x, y) \mapsto (x^q, y^q),$$

where $\overline{\mathbb{F}}_q$ is the algebraic closure of $\mathbb{F}_q$. The Frobenius map ϕ satisfies $\phi^2 - t\phi + q = 0$ on $E(\overline{\mathbb{F}}_q)$. The endomorphism ring of E contains $\mathbb{Z}[\phi] = \mathbb{Z}[x]/(x^2 - tx + q)$. Then the norm function $N(\cdot)$ on $\mathbb{Z}[\phi]$ is defined by $N(a + b\phi) := (a + b\phi)(a + b\bar{\phi}) = a^2 + tab + qb^2, a, b \in \mathbb{Z}$, where $\bar{\phi}$ is the dual of ϕ corresponding to the complex conjugate of ϕ in $\mathbb{C}$. (We may regard ϕ as a complex number $(t + \sqrt{t^2 - 4q})/2$ and $\mathbb{Z}[\phi]$ as a subset of $\mathbb{C}$.)

We denote $E(\mathbb{F}_{q^m})$ by the subgroup of $E(\overline{\mathbb{F}}_q)$ consisting of $\mathbb{F}_{q^m}$-rational points. Let G be the subgroup of $E(\mathbb{F}_{q^m})$ generated by P with a prime order p satisfying $p^2 \nmid \#E(\mathbb{F}_{q^m})$ and $p \nmid E(\mathbb{F}_q)$. Then we have $\phi^m = 1$ and

$$\phi^{m-1} + \phi^{m-2} + \cdots + \phi + 1 = 0$$

on the group G since $G \cap E(\mathbb{F}_q) = \{O\}$.

In this section, the exponent sets can be extended to this endomorphism ring, that is why we call complex exponents. Now we consider two candidates for the exponent set.

$$S_1 = \{\sum_{i=0}^{e} a_i\phi^i \mid a_i \in \mathbb{Z},\ |a_i| \le q-1,\ a_i a_{i+1} = 0\},$$
$$S_2 = \{\sum_{i=0}^{e} a_i\phi^i \mid a_i \in \mathbb{Z},\ q \nmid a_i,\ |a_i| < q^2/2,\ a_i a_{i+1} = 0\},$$

where the number of non-zero a_i is at most k.

We note that each S_i is a set of generalized signed ϕ-adic NAF expansions. To count the number of elements in S_i which are distinct as endomorphisms of G, we need the following lemma.

Lemma 2. *Let $e \ge 5$ and $f(x) = \sum_{i=0}^{e} a_i x^i$ be a polynomial in $\mathbb{Z}[x]$ with $|a_i| \le M$. If $30q^{e+1}M^2 \le p$, $f(\phi)P = 0$ implies that $x^2 - tx + q$ divides $f(x)$ in $\mathbb{Z}[x]$.*

Proof. By division algorithm, we have $g_i, r_i \in \mathbb{Z}$ such that

$$f(x) = (x^2 - tx + q)(g_{e-2}x^{e-2} + \cdots + g_1 x + g_0) + r_1 x + r_0.$$

Let $M_0 = 0$, $M_1 = M$ and $M_i = M + |t|M_{i-1} + qM_{i-2}$ for $i \ge 2$. Then, by equating the coefficients of x^i, we have

$$\begin{aligned} |g_{e-2}| &= |a_e| \le M_1, \\ |g_{e-3}| &= |a_{e-1} + tg_{e-2}| \le M + |t|M_1 = M_2, \\ |g_{e-i-1}| &= |a_{e-i+1} + tg_{e-i} - qg_{e-i+1}| \\ &\le M + |t|M_{i-1} + qM_{i-2} = M_i \end{aligned}$$

for $2 \le i \le e-1$. Also we have

$$\begin{aligned} |r_1| &= |a_1 + tg_0 - qg_1| \le M_e, \\ |r_0| &= |a_0 - qg_0| \le M + qM_{e-1}. \end{aligned}$$

If we take $M'_i = M_i + M'$ $(i \ge 0)$ for $M' = M/(|t| + q - 1)$, we have $M'_i = |t|M'_{i-1} + qM'_{i-2}$ for $i \ge 2$. This recurrence relation has the unique solution $M'_i = c_1\beta_1^i + c_2\beta_2^i$ where β_1 and β_2 are the solutions of $x^2 - |t|x + q$ in $\mathbb{C}$ and $c_1 = \frac{M'\beta_2 + M - M'}{\beta_1 - \beta_2}$ and $c_2 = \frac{-M'\beta_1 + M - M'}{\beta_2 - \beta_1}$. Thus we have for $i \ge 4$

$$|M_i| \le \max\{c_1, c_2\}|\beta_1^i + \beta_2^i| + M' \le \frac{21}{8}q^{i/2}M,$$

because

$$|c_i| \leq \frac{M}{\sqrt{4q-t^2}} \left(\frac{\sqrt{q}+1}{|t|+q-1} + 1 \right) \leq \frac{5}{4}M$$

and

$$|\beta_1^i + \beta_2^i| = |q^i + 1 - \#E_1(\mathbb{F}_{q^i})| \leq 2\sqrt{q^i}$$

by Hasse-Weil Theorem, where E_1 is E if $t > 0$ and the twist of E (with $q + 1 + t$ $\mathbb{F}_q$-rational points) if $t < 0$. Hence we have

$$|r_1| \leq \frac{21}{8} q^{e/2} M$$

and

$$|r_0| \leq M + qM_{e-1} \leq (1 + \frac{21}{8} q^{(e+1)/2}) M \leq \frac{22}{8} q^{(e+1)/2} M.$$

Using this bound, we have $N(f(\phi)) = N(r_1\phi + r_0) = r_1^2 q + t r_1 r_0 + r_0^2 < 30q^{e+1}M^2$. Suppose $f(\phi)P = O$. Then $N(f(\phi))P = f(\bar{\phi})f(\phi)P = O$ and $N(f(\phi)) \in \mathbb{Z}$, and so $N(f(\phi))$ is divisible by p. Therefore, if we take $30q^{e+1}M^2 \leq p$, then $N(f(\phi)) = 0$ and so $r_1 = r_0 = 0$, which completes proof. □

Now we can show that each element of S_i is distinct as an endomorphism of G under some condition.

Theorem 3. *If* $5 \leq e \leq \frac{\log p - 2\log M - 5}{\log q}$, *then each element of* S_i *is a distinct endomorphism of* G *where* $M = 2(q-1)$ *for* S_1 *and* $M = 2\lfloor \frac{q^2-1}{2} \rfloor$ *for* S_2.

Proof. Suppose that $f_1(\phi)P = f_2(\phi)P$ for two different polynomials $f_1(x) = \sum_{i=0}^{e} a_i x^i$ and $f_2(x) = \sum_{i=0}^{e} b_i x^i$ in $\mathbb{Z}[x]$ with $|a_i| \leq M/2$ and $|b_i| \leq M/2$. Then $f_1(x) - f_2(x)$ is divisible by $x^2 - tx + q$ by Lemma 2.

Let j be the smallest index such that $a_j \neq b_j$. Then $g(x) = (f_1(x) - f_2(x))/x^j = \sum_{i=0}^{e-j} c_i x^i \in \mathbb{Z}[x]$ is divisible by $x^2 - tx + q$. So c_0 must be a multiple of q. Let $c_0' = c_0/q \in \mathbb{Z}$. Then $(g(x) - c_0'(x^2 - tx + q))/x = \sum_{i=3}^{e-j} c_i x^{i-1} + (c_2 - c_0')x + (c_1 + tc_0')$ is also divisible by $x^2 - tx + q$, hence $c_1 + tc_0'$ must be divided by q again. That is, q^2 divides $qc_1 + tc_0$.

If either $a_j = 0$ or $b_j = 0$, then q does not divide c_0 by the definition of S_1 and S_2. If both of a_j and b_j are non-zero, then $a_{j+1} = b_{j+1} = 0$ and $q^2 \nmid qc_1 + tc_0 = tc_0$. In both cases, $g(x)$ can not be divided by $x^2 - tx + q$, which is a contradiction. □

The cardinality of S_i is given in the next theorem.

Theorem 4. *The cardinality of S_i is given by*

1. $|S_1| = \sum_{i=0}^{k} \binom{e+2-i}{i} (2q-2)^i$,
2. $|S_2| = \sum_{i=0}^{k} \binom{e+2-i}{i} (q^2-q)^i$.

Proof. At first, we will count the cardinality of S_1. The proof is similar to the proof of Theorem 2. The difference is that a negative coefficient is allowed in the most significant nonzero position. The number of elements whose most significant bit is zero is

$$\sum_{i=0}^{k} \binom{e+1-i}{i} (2q-2)^i.$$

The number of elements whose most significant bit is nonzero is

$$(2q-2) \sum_{i=0}^{k-1} \binom{e-i}{i} (2q-2)^{i-1}.$$

By summing up the above values, we have

$$|S_1| = \sum_{i=0}^{k} \binom{e+2-i}{i} (2q-2)^i.$$

For the cardinality of S_2, it is essentially same as the above case. Only difference is the number of coefficients, which is $q^2 - q$. □

Our complex exponent test using Frobenius maps is described in Figure 3. In order to perform the test efficiently, we utilize BGMW multiplications [1].

In Table 6, given a power of prime q we present the maximum e such that all elements of S_i are distinct as endomorphisms of G. Also we compute the maximum weight k satisfying that the cardinality S_i is greater than 2^ℓ for various screening parameter ℓ. For example, when $q = 8$, the maximum e becomes 47 and we need only 9 elliptic curve additions for each exponentiation when the number of batch instances n is large. More precisely, to verify n exponentiations, we need $9n + 80$ elliptic curve additions when a normal basis is used for representation of field elements.

```
Input: $(x_i, Q_i)$ for $1 \leq i \leq n$.
Check: $x_i P = Q_i$ for all $1 \leq i \leq n$ where $P$ is a generator of $E$.

1. Choose $n$ random elements $\sigma_1, \sigma_2, \cdots, \sigma_n$ from $S_1(e, k, q)$ or $S_1(e, k, q)$
   Denote that $\sigma_i = \sum_{j=0}^{e} c_j \phi^j$ and $\epsilon_j = c_j / |c_j|$ for nonzero $c_j$ for each $i$.
2. $\sigma \leftarrow \sum_{i=1}^{n} \sigma_i x_i \mod (\phi^{m-1} + \phi^{m-2} + \cdots + \phi + 1)$.
3. $R[i] \leftarrow O$ for $0 \leq i \leq M$
4. For $j = 0$ to $e$, do
   (a) For $i = 0$ to $n$, do
       if $c_j$ of $\sigma_i$ is not zero, then $R[|c_j|] \leftarrow R[|c_j|] + \epsilon_j \phi^j (Q_i)$.
5. $Q \leftarrow R[M]$, $T \leftarrow R[M]$
6. For $i = M - 1$ to 1 do
   (a) $T \leftarrow T + R[i]$
   (b) $Q \leftarrow Q + T$
7. Accept if $Q = \sigma P$, else reject.
```

Fig. 3. Complex exponent test using Frobenius maps

Table 6. The maximum weight k in each S_1 and S_2

ℓ	10	20	40	60	80	ℓ	10	20	40	60	80
S_1 $(q = 2, e = 153)$	2	3	6	10	15	S_1 $(q = 3, e = 95)$	2	3	6	10	15
S_2 $(q = 2, e = 153)$	2	3	6	10	15	S_2 $(q = 3, e = 94)$	2	3	6	9	13
S_1 $(q = 4, e = 74)$	2	3	6	10	14	S_1 $(q = 5, e = 64)$	2	3	6	9	14
S_2 $(q = 4, e = 73)$	2	3	5	8	12	S_2 $(q = 5, e = 62)$	1	3	5	8	11
S_1 $(q = 8, e = 49)$	2	3	6	9	13	S_1 $(q = 7, e = 52)$	2	3	6	9	13
S_2 $(q = 8, e = 47)$	1	2	4	7	9	S_2 $(q = 7, e = 51)$	1	2	5	7	10

5 Conclusion

In this paper, we developed two batch verification algorithms using sparse exponents with small weights. The first one makes use of exponents of (signed) binary representation. We can take exponents whose weight is less than $\ell/4$ with binary expansion and $\ell/5.5$ with signed binary expansion while the average weight of exponents is $\ell/2$ in the small exponent test. Hence we expect that our test is about twice faster. If we use a group whose order is secret, e.g. RSA group, we can extend our algorithm to allow longer exponents to reduce the weight of exponents.

The second one can be applied only to special family of elliptic curves defined over small finite fields. It utilizes the exponents with sparse ϕ-adic expansion. By enlarging the coefficients set of the expansion, we obtained a complex exponent test much faster than sparse exponent test.

References

1. E. Brickell, D. Gordon, K. McCurley, and D. Wilson, "Fast Exponentiation with Precomputation," *Proc. of Eurocrypt'92*, LNCS Vol. 658, pp. 200-207, Springer-Verlag, 1993.
2. M. Bellare, J. Garay, and T. Rabin, "Fast Batch Verification for Modular Exponentiation and Digital Signatures," *Proc. of Eurocrypt'98*, LNCS Vol. 1403, pp. 236–250, Springer-Verlag, 1998. Full version is available via http://www-cse.ucsd.edu/users/mihir.
3. M. Beller and Y. Yacobi, "Batch Diffie-Hellman Key Agreement Systems and their Application to Portable Communications," *Proc. of Eurocrypt'92*, LNCS Vol. 658, pp. 208–220, Springer-Verlag, 1993.
4. C. Boyd and C. Pavlovski, "Attacking and Repairing Batch Verification Schemes," *Proc. of Asiacrypt 2000*, LNCS Vol. 1976, pp. 58–71, Springer-Verlag, 2000.
5. M. Brown, D. Hankerson, J. López, and A. Menezes, "Software Implementation of the NIST Elliptic Curves over Primes Fields," *Proc. of CT-RSA 2001*, LNCS, Vol. 2020, pp. 250–265, Springer-Verlag, 2001.
6. R. Cramer and V. Shoup, "Signature Schemes based on the Strong RSA Assumptions," *ACM TISSEC*, Vol. 3, No. 3, pp.161–185, 2000. A preliminary version appeared in *Proc. of Crypto'89*, LNCS Vol. 435, pp. 175–185, Springer-Verlag, 1989.
7. *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, ANSI X9.62, approved January 7, 1999.
8. A. Fiat, "Batch RSA," *J. Cryptology*, Vol. 10, No. 2, pp. 75–88, Springer-Verlag, 1997. A preliminary version appeared in *Proc. of Crypto'89*, LNCS Vol. 435, pp. 175–185, Springer-Verlag, 1989.
9. L. Guillou and J. Quisquater, "A Practical Zero-knowledge Protocol fitted to Security Microprocessor Minimizing both Transmission and Memory," *Proc. of Eurocrypt'88*, LNCS Vol. 330, pp. 123–128, Springer-Verlag, 1988.
10. D. Hankerson, J. Hernandez, and A. Menezes, "Software Implementation of Elliptic Curve Cryptography Over Binary Fields," *Proc. of CHES 2000*, LNCS, Vol. 1965, pp. 1–24, Springer-Verlag, 2000.

11. A. May, "Computing the RSA Secret Key is Deterministic Polynomial Time equivalent to Factoring," *Proc. of Crypto 2004*, LNCS Vol. 3152, pp. 213–219, Springer-Verlag, 2004.
12. D. M'Raithi and D. Naccache, "Batch Exponentiation - A Fast DLP based Signature Generation Strategy," *ACM Conference on Computer and Communications Security*, pp. 58–61, ACM, 1996.
13. E. Mykletun, M. Narasimha and G. Tsudik, "Authentication and Integrity in Outsourced Databases," *Proc. of ISOC Symposium on Network and Distributed Systems Security (NDSS04)*, 2004.
14. V. Muller, "Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two," *J. of Cryptology*, Vol. 11, pp. 219–234, Springer-Verlag, 1998.
15. D. Naccache, D. M'Raithi, S. Vaudenay, and D. Raphaeli, "Can D.S.A be Improved? Complexity trade-offs with the Digital Signature Standard," *Proc. of Eurocrypt'94*, LNCS Vol. 950, pp. 77–85, Springer-Verlag, 1994.
16. J. Solinas, "An Improved Algorithm for Arithmetic on a Family of Elliptic Curves," *Proc. of Crypto'97*, pp. 357–371, Springer-Verlag, 1997. Full version is avaiable at http://www.cacr.math.uwaterloo.ca/techreports/
17. H. Sun, W. Yang, and C. Laih, "On the Design of RSA with Short Secret Exponent," *Proc. of Asiacrypt'99*, pp. 150–164, Springer-Verlag, 1999.
18. S. Yen and C. Laih, "Improved Digital Signature suitable for Batch Veriffication," *IEEE Trans. on Computers*, Vol. 44, No. 7, pp. 957–959, 1995.