Fast Batch Verification of Multiple Signatures

Jung Hee Cheon[1] and Jeong Hyun Yi[2]

[1] ISaC and Dept. of Mathematics
Seoul National University, Republic of Korea
jhcheon@snu.ac.kr
[2] Communication and Networking Lab
Samsung Advanced Institute of Technology, Republic of Korea
jeong.yi@samsung.com

Abstract. We propose an efficient batch verification of multiple signatures generated by *different signers* as well as a single signer. We first introduce a method to generate width-w Non-Adjacent Forms (w-NAFs) uniformly. We then propose a batch verification algorithm of exponentiations using w-NAF exponents, and apply this to batch verification for the modified DSA and ECDSA signatures. The performance analysis shows that our proposed method is asymptotically seven and four times as fast as individual verification in case of a single signer and multiple signers, respectively. Further, the proposed algorithm can be generalized into τ-adic w-NAFs over Koblitz curves and requires asymptotically only six elliptic curve additions per each signature for batch verification of the modified ECDSA signatures by a single singer. Our result is the first one to efficiently verify multiple signatures by multiple signers that can introduce much wider applications.

Keywords: Batch verification, exponentiation, sparse exponent, non-adjacent form, elliptic curve, Koblitz curve, Frobenius map

1 Introduction

Batch verification was introduced by Naccache *et al.* to verify multiple signatures more efficiently [NMVR94]. Their method is to use a set of small exponents to verify multiple exponentiations simultaneously: Let G be an abelian group with a generator g. Given a batch instance of n 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 if $g^{\sum_{i=1}^{n} x_i s_i} = \prod_{i=1}^{n} y_i^{s_i}$ for randomly chosen $s_i \in S$, where the exponent set S is taken to be the set of e-bit prime integers for small e. This test was improved by adopting small exponent set $\{0,1\}^{\ell}$ by Yen and Laih [YL95] and Bellare *et al.* [BGR98]. Another improvement [CL06] was obtained by taking longer integers of small weights, so called *sparse exponents*, as elements of S rather than small integers.

In this paper, we improve the previous results by employing generalized sparse exponents, so called *width-w non-adjacent forms* (w-NAFs for short). A w-NAF of weight t is a radix 2 representation satisfying: (1) each nonzero

digit is an odd integer less than 2^w, (2) at most one of any w consecutive digits is nonzero, and (3) the number of nonzero digits is t. We first introduce a method to generate w-NAFs uniformly and then propose a batch verification algorithm of exponentiations using w-NAF exponents. The performance analysis shows that N exponentiations can be verified with $16N + 241$ multiplications over a finite field. In the previous method, it was $40N + 241$ and $19N + 241$ using small exponent test [BGR98] and sparse exponent test [CL06], respectively. Our verification cost becomes $14N + 235$ elliptic curve additions over elliptic curves in which a subtraction is as efficient as an addition.

To apply batch verification technique to DSA [DSA], one needs to slightly modify the signature scheme as in [NMVR94]. We apply the proposed algorithm for batch verification of the modified DSA and ECDSA signatures. The verification can be asymptotically 7.1 and 8.3 times as fast as individual verifications in a finite field and an elliptic curve with 160 bit security, respectively. Furthermore, for digital signatures by the multiple signers with the same system parameters the proposed verification performs asymptotically 4.3 and 4.8 times faster than the individual verifications in a finite field and an elliptic curve, respectively. Our result is the first one about the batch verification of signatures by different signers.

We further generalize our method to τ-adic w-NAFs over Koblitz curves. In [CL06], the authors proposed a batch verification algorithm for the modified ECDSA signatures by one signer, in which only 9 elliptic curve additions are required for one additional signature. Using τ-adic w-NAFs, we reduce it to 6 elliptic curve additions. It is very surprising that only 6 elliptic curve additions are required asymptotically to verify one signature.

Applications Batch verification will be useful in any settings where multiple signatures need to be verified at once. We have a variety of applications in which our proposed method can be employed. In some cases, we may need to adjust our techniques. For example, in e-cash applications, merchants and/or consumers need to verify the validity of lots of electronic coins signed by the bank. E-voting systems need to verify huge number of signed ballots as fast as possible. In the outsourced database applications [MNT04], numbers of clients' query request messages need to be authenticated by servers. Another example is authenticated routing based on public key cryptography, in which network packets are signed and verified in each node and each router has to verify many signatures. We also are able to apply to Mixnet [Abe99] for making systems or protocols privacy-preserving, and VSS (Verifiable Secret Sharing) [Fel87] scheme which is a fundamental technique for fault-tolerant and secure distributed computations such as reliable broadcast, peer group membership management, and Byzantine agreement.

Organization The rest of paper is organized as follows: we first define batch verification and introduce fast exponentiation methods in Section 2. An efficient batch verification algorithm is proposed in Section 3, and its applications to

signature schemes are described in Section 4. We then present more efficient algorithm over Koblitz curves in Section 5 and conclude in Section 6.

2 Preliminary

2.1 Batch Verification

Let G be a cyclic group of prime order p with a generator g. Given a subset S of $\mathbb{Z}_p$, we define a *batch verifier* $\mathcal{V}_S$ following [BGR98,CL06]:

1. Input a batch instance $\{(x_i, y_i) \in \mathbb{Z}_p \times G | i = 1, 2, \ldots, N\}$
2. $\mathcal{V}_S$ takes N elements $c_1, c_2, \ldots, c_N$ uniformly from the exponent set S
3. $\mathcal{V}_S$ computes $x = \sum_{i=1}^{N} c_i x_i$ and $y = \prod_{i=1}^{N} y_i^{c_i}$
4. If $g^x = y$ output 1 and otherwise output 0

We say a batch instance $\{(x_i, y_i) \in \mathbb{Z}_p \times G | i = 1, 2, \ldots, N\}$ is correct if $g^{x_i} = y_i$ for all i and incorrect otherwise. Note that the verifier $\mathcal{V}_S$ outputs 1 for a correct instance regardless of S. We define the $Fail(V_S)$ to be the maximum probability that an incorrect batch instance passes the test. That is,

$$Fail(\mathcal{V}_S) = \max_{\text{A batch instance } X} \{Prob[\mathcal{V}_S(X) = 0]\}, \tag{1}$$

where probability is over the random choice of $c_1, \ldots, c_N$ uniformly from S. Then, if $c_1, c_2, \ldots, c_N$ are uniformly chosen from S, we have

$$Fail(\mathcal{V}_S) = \max_{\alpha \in \mathbb{Z}_p^N \setminus \{(0,\ldots,0)\}} \frac{|\{(c_1, \ldots, c_n) | c_1, \ldots, c_n \in S, g^{c_1\alpha_1 + \cdots + c_n\alpha_n} = 1\}|}{|\{(c_1, \ldots, c_n) | c_1, \ldots, c_n \in S\}|}. \tag{2}$$

Theorem 1 in [CL06] shows that it is upper-bounded by $1/|S|$; that is, we have $g^{x_i} = y_i$ for all i with probability at least $1 - 1/|S|$.

2.2 Fast Exponentiations

To evaluate the performance of the proposed algorithm, we apply the most up-to-date fast exponentiation methods to our batch verification and individual ones. Following [HHM00], Lim-Lee method (fixed based comb) and the window method appeared to be most efficient methods for a fixed base and a non-fixed base, respectively. We consider an exponentiation on a group of m-bit prime order. Lim-Lee method with window size w requires at average $(m/w - 1)$ doublings and $(m/w - 1)(1 - 2^{-w})$ additions. Window method with window size w requires at average $(m/(w+1) + 2^{w-1} - s)$ multiplications and $(m+1)$ squarings over a finite field, and $(m/(w+1) + 2^{w-1} - s)$ additions and $(m+1)$ doublings over an elliptic curve. Refer to [LL94,HHM00] for more details.

In this paper, we will consider a finite field of 160 bit order, an elliptic curve of 160 bit order and Koblitz curve $K163$ as a base group G. $K163$ is given by $E : y^2 + xy = x^3 + x^2 + 1$ over $\mathbb{F}_{2^{163}}$ and has 162 bit order (cofactor=2). Notation used in the rest of paper is summarized in Table 1. We present the number of group operations for fast exponentiations in Table 2.

Table 1. Notation

w	window size
m	bit length of exponents
t	Hamming weight
Mem	number of finite field elements or elliptic curve points to be stored
$\mathcal{E}_f$	finite field exponentiation
$\mathcal{M}_f$	finite field multiplication
$\mathcal{S}_f$	finite field squaring
$\mathcal{M}_e$	scalar multiplication in elliptic curves
$\mathcal{A}_e$	elliptic curve addition
$\mathcal{D}_e$	elliptic curve doubling

Table 2. Performance of Fast Exponentiation Algorithms

Group	Method	w	Mem	$\mathcal{M}_f$ or $\mathcal{A}_e$	$\mathcal{S}_f$ or $\mathcal{D}_e$
Finite Field	Window NAF	4	7	39	161
	Lim-Lee	4	14	38	40
Elliptic Curve	w-NAF	4	3	36	161
	Lim-Lee	4	14	38	40
Koblitz Curve	τ-adic w-NAF	5	7	34	0
	Fixed-based τ-adic w-NAF	6	15	23	0

3 Batch Verification of Exponentiations on Abelian Groups

Let $w \geq 2$ be an integer. A radix 2 representation is called a *width-w nonadjacent Form* (w-NAF, for short) if it satisfies: (1) each nonzero digit is an odd integer with absolute value less than 2^{w-1}, and (2) for any w consecutive digits, at most one is nonzero [MS06].

Although w-NAF gives an efficient exponentiation on a group admitting fast inversion, it is not useful for a group such as a multiplicative subgroup of a finite field in which an inversion is much slower than a multiplication. We here introduce a generalized version of w-NAF with a digit set D.

Definition 1. *Let w be an integer ≥ 2 and $D = \{\alpha_1, \alpha_2, \ldots, \alpha_{2^{w-1}}\}$ where α_i's are nonzero odd integers and distinct modulo 2^w. A* w-NAF with the digit set D *is a sequence of digits satisfying the following conditions:*

1. *Each digit is zero or an element in D.*
2. *Among any w consecutive digits, at most one is nonzero.*

A w-NAF with the digit set D is denoted by $a = (a_{m-1} \cdots a_1 a_0)_2$ or $a = \sum_{i=0}^{m-1} a_i 2^i$ where $a_i \in D \cup \{0\}$.

Definition 2. *Let $a = (a_{m-1} a_{m-2} \ldots a_0)_2$ be a w-NAF with the digit set D. Then the* length *of a, denoted by* $\mathrm{len}(a)$*, is defined to be the smallest i such that*

$a_{i-1} \neq 0$. *By notation, we let* $\mathrm{len}(0) = 0$. *The number of nonzero digits in its representation is called the* weight *of* a *and denoted by* $\mathrm{wt}(a)$.

The uniqueness of the representation can be easily shown as follows. The argument is a simple generalization of Proposition 2.1 in [MS06].

Theorem 1. *Let* q *be a positive integer. All* w*-NAFs of length* $\leq m$ *with the digit set* D *are distinct modulo* q *if* $m \leq \log_2(q/C)$ *where* $C = \max\{|x - y| : x, y \in D \cup \{0\}\}$.

Proof. Suppose there are two different w-NAFs which represent the same integer. Let $(a_{\ell-1}a_{\ell-2}\cdots a_0)_2$ and $(b_{\ell'-1}b_{\ell'-2}\cdots b_0)_2$ are different representation such that

$$a = \sum_{i=0}^{\ell-1} a_i 2^i = \sum_{i=0}^{\ell'-1} b_i 2^i. \tag{3}$$

Assume ℓ is the smallest integer satisfying the above property.

If $a_0 = b_0$, we have two different and shorter w-NAFs which stand for the same integer. Thus it should be $a_0 \neq b_0$. If a is even, both of a_0 and b_0 should be zero and $a_0 = b_0$. It therefore should be odd and both of a_0 and b_0 should be nonzero. Since the representations are w-NAFs, $a_0 \neq 0$ and $b_0 \neq 0$ implies $a_1 = \cdots = a_w = 0$ and $b_1 = \cdots = b_w = 0$. From the equation (3), we have $a_0 \equiv b_0 \mod 2^w$. Since all elements in D are distinct modulo 2^w, we must have $a_0 = b_0$, which contradicts with the minimality of ℓ. Thus each integer has only one w-NAF with the digit set D.

Moreover, let C_1 and C_2 be the maximal and minimal element in $D \cup \{0\}$. Then $C = C_1 - C_2$. The largest w-NAF of length $\leq m$ is less than $C_1 2^m$. The smallest w-NAF of length $\leq m$ is greater than $C_2 2^m$. Thus the difference of any two w-NAFs is less than $(C_1 - C_2)2^m = C2^m \leq q$ for $m \leq \log(q/C)$. Therefore any two w-NAF of length $\leq m$ must be distinct modulo q or identical.

Theorem 2. *The number of* w*-NAFs of length* $\leq m$ *and weight* t *with the digit set* D *is*

$$\binom{m - (w-1)(t-1)}{t} 2^{(w-1)t}.$$

Proof. Consider an algorithm to choose t positions out of $m - (w-1)(t-1)$ positions and fill each of them by $w-1$ consecutive zeros followed by an element in D. This algorithm gives a w-NAF of length $m + (w-1)$. Then its first $(w-1)$ positions should be always zero since each nonzero digit is preceded by $(w-1)$ consecutive nonzeros. By discarding the first $(w-1)$ zeros, we get a w-NAF of length $\leq m$. Since the algorithm covers all w-NAFs of length $\leq m$ and the algorithm outputs one of $\binom{m-(w-1)(t-1)}{t}2^{(w-1)t}$ strings, we have the theorem.

From the proof of Theorem 2, we introduce Algorithm 1 to produce a random secret exponent in a finite field of 2^n elements.

Using the set of w-NAFs, we can perform efficient batch verification of exponentiations on a group as in Algorithm 2. Here we use simultaneous multiplication methods and online precomputation method.

Algorithm 1 (Generation of w-NAF exponents of weight t)

Input: m, w, t and the digit set D
Output: w-NAF of length $\leq m$

1: Choose t positions out of $n-(w-1)(t-1)$ positions.
2: Fill each position by $(w-1)$ consecutive zeros followed by an element in D.
3: Discard the first $(w-1)$ positions of the string.
4: Print the string which is a w-NAF of length $\leq m$

Algorithm 2 (Batch Verification of Exponentiations using w-NAF Exponent)

Input: m, w, t, D, and N exponentiation pairs $(x_i, y_i) \in \mathbb{Z}_q \times G$ for an abelian group G of order q with a generator g
Output: True or false

1: Take N random exponents $c_1, c_2, \ldots, c_N$ from the set of w-NAFs of length $\leq m$ and weight t, where $c_i = \sum_{j=0}^{m} c_{ij}2^j$ and $c_{ij} \in D \cup \{0\}$.
2: **for** $\alpha \in D$ **do**
3: $\quad y_{i,\alpha} \leftarrow y_i^{\alpha}$ /* precomputation */
4: **end for**
5: $y \leftarrow 1$
6: **for** $j = m-1$ downto 0 **do**
7: $\quad y \leftarrow y^2$
8: $\quad$ **for** $i = 1$ upto N **do**
9: $\quad\quad$ **if** $c_{ij} = \alpha \in D$ **then**
10: $\quad\quad\quad y \leftarrow y \cdot y_{i,\alpha}$
11: $\quad\quad$ **end if**
12: $\quad$ **end for**
13: **end for**
14: Compute g^x for $x = \sum_{i=1}^{N} c_i x_i \mod q$.
15: **if** $y = g^x$ **then**
16: $\quad$ Accept all of N instances
17: **else**
18: $\quad$ Reject
19: **end if**

We need to take an appropriate digit set for each of specific groups. For a multiplicative subgroup of a finite field in which an inversion is much slower than a multiplication, we take $D = \{1, 3, \ldots, 2^w - 1\}$. It then requires the precomputation that takes one squaring and 2^{w-1} multiplications. Steps 6-13 take $m - 1$ squarings and $tN - 1$ multiplications since each exponent has t nonzero digit. Hence the total complexity is m squarings and $N(t + 2^{w-1})$ multiplications plus one exponentiation using memory for $2^{w-1} - 1$ group elements.

For an elliptic curve group in which a subtraction is as efficient as an addition, we take $D = \{\pm 1, \pm 3, \ldots, \pm(2^{w-2} - 1)\}$. In this case, the precomputation cost reduces to one elliptic doubling and 2^{w-2} elliptic additions. Hence the total complexity is m elliptic doublings and $N(t + 2^{w-2})$ elliptic additions plus one scalar multiplication using memory for $2^{w-2} - 1$ elliptic curve points.

Table 3. Number of Multiplications for Batch Verification on Abelian Groups

Common				Finite Field		Elliptic Curve	
w	m	t	Security	Mem	Complexity	Mem	Complexity
1	159	19	$2^{80.6}$	0	19N$\mathcal{M}_f$+159$\mathcal{S}_f$+1$\mathcal{E}_f$	0	19N$\mathcal{A}_e$+159$\mathcal{D}_e$+1$\mathcal{M}_e$
2	158	15	$2^{81.2}$	1	17N$\mathcal{M}_f$+158$\mathcal{S}_f$+1$\mathcal{E}_f$	0	15N$\mathcal{A}_e$+158$\mathcal{D}_e$+1$\mathcal{M}_e$
3	157	12	$2^{79.4}$	3	16N$\mathcal{M}_f$+157$\mathcal{S}_f$+1$\mathcal{E}_f$	1	14N$\mathcal{A}_e$+157$\mathcal{D}_e$+1$\mathcal{M}_e$
4	156	11	$2^{83.9}$	7	19N$\mathcal{M}_f$+156$\mathcal{S}_f$+1$\mathcal{E}_f$	3	15N$\mathcal{A}_e$+156$\mathcal{D}_e$+1$\mathcal{M}_e$
5	155	9	$2^{79.6}$	15	25N$\mathcal{M}_f$+155$\mathcal{S}_f$+1$\mathcal{E}_f$	7	17N$\mathcal{A}_e$+155$\mathcal{D}_e$+1$\mathcal{M}_e$

Table 4. Comparison of Batch Verification of Exponentiations

Method	Finite Field	Elliptic Curve
Individual	N(39$\mathcal{M}_f$+161$\mathcal{S}_f$)	N(36$\mathcal{A}_e$+161$\mathcal{D}_e$)
[YL95,BGR98]	N(40$\mathcal{M}_f$)+80$\mathcal{M}_f$+161$\mathcal{S}_f$	N(40$\mathcal{A}_e$)+74$\mathcal{A}_e$+161$\mathcal{D}_e$
[CL06]	N(19$\mathcal{M}_f$)+80$\mathcal{M}_f$+161$\mathcal{S}_f$	N(15$\mathcal{A}_e$)+74$\mathcal{A}_e$+161$\mathcal{D}_e$
Proposed	N(16$\mathcal{M}_f$)+80$\mathcal{M}_f$+161$\mathcal{S}_f$	N(14$\mathcal{A}_e$)+74$\mathcal{A}_e$+161$\mathcal{D}_e$

Table 3 presents the performance of batch verification over a finite field and an elliptic curve and shows appropriate weight t on a group of 160-bit prime order q for various w. We take $m = 160 - w$ to guarantee the uniqueness of exponents by Theorem 1. For example, we can use 3-NAF for a finite field, which requires only $16N$ multiplications, 157 squarings and one exponentiation. For an elliptic curve, we can use 2-NAF requiring only $15N$ multiplications, 158 squarings and one exponentiation. Note that *security* in Table 3 implies the security of the batch verification with the given parameters, which is computed as $\binom{m-(w-1)(t-1)}{t}2^{(w-1)t}$ by Theorem 2.

4 Batch Verification of Multiple Signatures

To apply the batch verification of exponentiations to verification of signatures, one need to modify signature schemes. Naccache *et al.* presented a modified DSA for batch verification [NMVR94]. Our batch verification is also applicable to this modified DSA. But, considering the attack by Boyd and Pavlovski [BP00], we made a little change to the verification procedure. The performance of the batch verification algorithm is evaluated based on the screening parameter $\ell = 80$.

4.1 Modified DSA

Let p be a 1024 bit prime and q a 160 bit prime dividing $p-1$. We assume that $(p-1)/(2q)$ has no divisor less than q to resist the attack in [BP00]. Let g be a generator of a subgroup G of order q in $\mathbb{F}_p$. Take a random $x \in \mathbb{Z}_p$. The private key is x and the corresponding public key is $y = g^x$. A signature for a message $m \in \mathbb{Z}_p$ is given by

$$(r = g^k \bmod p, \quad \sigma = k^{-1}(m + xr) \bmod q)$$

for a random $k \in \mathbb{Z}_p$. It is verified by checking if $r = \pm g^a y^b \bmod p$ for $a = m\sigma^{-1} \bmod q$ and $b = r\sigma^{-1} \bmod q$. Note that $r = ((g^k \bmod p) \bmod q)$ is used in the original DSA. The verification admits only $r = g^a y^b \mod q$, but here we relax the verification to admit $r = \pm g^a y^b \mod q$ due to Boyd and Pavlovski attack, in which the security loss is only one bit.

Signatures by Multiple Signers Given N signatures (m_i, r_i, σ_i), each of which is signed by a signer with the public key y_i, we apply the batch verification by 3-NAFs with the digit set $D = \{1, 3, 5, 7\}$, which gives best performance as in Table 3. First, take random w-NAFs $c_1, \ldots, c_N$. Next, compute $a = -\sum_{i=1}^{N} a_i c_i \bmod q$ and $b_i = -r_i \sigma_i^{-1} c_i \bmod q$ for each i. Finally compute

$$g^a \prod_{i=1}^{N} y_i^{b'_i} \prod_{i=1}^{N} r_i^{c_i} \bmod p, \tag{4}$$

and if it is 1 or $p-1$, accept all N signatures.

We now evaluate the verification cost. For simplicity, we only count F_p operations. Since g is fixed, we apply Lim-Lee method of window size $w = 4$ to compute g^a, and each of g^b or $g_i^{b'_i}$ is computed by 4-NAF. Thus an individual signature verification consists of one Lim-Lee, one 4-NAF method and one multiplication. On the other hand, the batch verification consists of one Lim-Lee, N 4-NAF, $16N$ multiplications and N multiplications. Table 5 shows the achieved gains as ratio of the proposed method and individual one. Note that the measurement is conducted only in case of $\mathcal{S}_f = \mathcal{M}_f$ and $\mathcal{S}_f = 0.8\mathcal{M}_f$. Following [BHLM01], it is between 0.8 and 0.86.

Signatures by A Single Signer We consider N signatures (m_i, r_i, σ_i) by a single signer. We apply the batch verification by 3-NAFs with the digit set $D = \{1, 3, 5, 7\}$. First, take random w-NAFs $c_1, \ldots, c_N$. Next, compute $a = -\sum_{i=1}^{N} a_i c_i \bmod q$ and $b = -\sum_{i=1}^{N} r_i \sigma_i^{-1} c_i \mod q$. Finally compute

$$g^a \prod_{i=1}^{N} y^b \prod_{i=1}^{N} r_i^{c_i} \bmod p, \tag{5}$$

and if it is 1 or $p-1$, accept all n signatures.

Now we evaluate the verification cost. Since both of g and y are fixed, we may apply Lim-Lee method to compute g^a and y^b. The individual verification consists of two Lim-Lee and one multiplication. On the other hand, the batch verification consists of two Lim-Lee and $14N$ additions. The performance is given in Table 5.

Table 5. Performance of Batch Verifications of Signatures over a Finite Field

Signers	Individual	Proposed	Ratio($\mathcal{S}_f$=$\mathcal{M}_f$)	Ratio($\mathcal{S}_f$=0.8$\mathcal{M}_f$)
Multi.	N(78$\mathcal{M}_f$+161$\mathcal{S}_f$)	N(55$\mathcal{M}_f$)+40$\mathcal{M}_f$+161$\mathcal{S}_f$	$0.23 + 0.84/N$	$0.27 + 0.82/N$
Single	N(76$\mathcal{M}_f$+40$\mathcal{S}_f$)	N(16$\mathcal{M}_f$)+76$\mathcal{M}_f$+40$\mathcal{S}_f$	$0.14 + 2.04/N$	$0.15 + 1.90/N$

4.2 Modified ECDSA

ECDSA is an elliptic curve analogue of DSA [ECDSA]. Our batch verification algorithm is applied to the modified ECDSA [ABGLSV05] as in DSA case. The security of the modified ECDSA is equivalent to the standard ECDSA [ABGLSV05].

Let E be an elliptic curve. Assume that the order q of E is prime and $G \in E$ a generator (If E has a cofactor $\neq 1$, the signature scheme should be modified due to [BP00]). The private key is x and the corresponding public key is $Q = xG$. A signature for given message $m \in \mathbb{Z}_p$ is

$$(R = kG, \quad \sigma = k^{-1}(m + xr) \mod q)$$

where $R = (x_1, y_1)$ and $r = x_1 \mod q$ for a random $k \in \mathbb{Z}_p$. The verification is done by checking if $R = aG + bQ$ for $a = m\sigma^{-1} \bmod q$ and $b = r\sigma^{-1} \bmod q$.

Given N signatures (m_i, R_i, σ_i), we compute $t_i = \sigma_i^{-1} \bmod q$ first, and then $a_i = m_i t_i \bmod q$ and $b_i = r_i t_i \bmod q$ for each i. Next, take random $s_i \in S$ and compute $a = -\sum_{i=1}^{n} a_i s_i \bmod q$ and $b = -\sum_{i=1}^{n} b_i s_i \bmod q$. Finally compute

$$aG + bQ + \sum_{i=1}^{n} s_i R_i,$$

and if it is a point at infinity O, accept all n signatures.

Signatures by Multiple Signers Given N signatures (m_i, R_i, σ_i), each of which is signed by a signer with the public key Q_i, we apply the batch verification by 3-NAFs with the digit set $D = \{\pm 1, \pm 3\}$. First, take random w-NAFs $c_1, \ldots, c_N$. Next, compute $a = -\sum_{i=1}^{N} a_i c_i \bmod q$ and $b'_i = -r_i \sigma_i^{-1} c_i \bmod q$ for each i. Finally compute

$$aG + \sum_{i=1}^{N} b'_i Q_i + \sum_{i=1}^{N} c_i R_i, \tag{6}$$

and if it is the point at infinity O, accept all N signatures. Remark that if we take an elliptic curve whose order is prime as above, the Boyd and Pavlovski attack [BP00] can not be applied.

Signatures by A Single Signer We consider N signatures (m_i, R_i, σ_i) by a single signer. We apply the batch verification by 3-NAFs with the digit set $D = \{\pm 1, \pm 3\}$. First, take random w-NAFs $c_1, \ldots, c_N$. Next, compute $a = -\sum_{i=1}^{N} a_i c_i \bmod q$ and $b = -\sum_{i=1}^{N} r_i \sigma_i^{-1} c_i \bmod q$ for each i. Finally compute

$$aG + bQ + \sum_{i=1}^{N} c_i R_i, \tag{7}$$

and if it is the point at infinity O, accept all N signatures.

Performance comparison of individual and batch verifications of ECDSA is given in Table 6. The performance of individual verifications is evaluated based on the standard verification equation. Note that the cost can be reduced by 40 % using some special method in [ABGLSV05].

Table 6. Performance of Batch Verification of Signatures over an Elliptic Curve

Signers	Individual	Proposed	Ratio ($\mathcal{D}_e$=$\mathcal{A}_e$)	Ratio ($\mathcal{D}_e$=0.5$\mathcal{A}_e$)
Multiple	N(74$\mathcal{A}_e$+164$\mathcal{D}_e$)	N(50$\mathcal{A}_e$)+38$\mathcal{A}_e$+164$\mathcal{D}_e$	$0.21 + 0.84/N$	$0.33 + 0.79/N$
Single	N(76$\mathcal{A}_e$+40$\mathcal{D}_e$)	N(14$\mathcal{A}_e$)+74$\mathcal{A}_e$+164$\mathcal{D}_e$	$0.12 + 1.72/N$	$0.16 + 1.37/N$

5 Batch Verification on Koblitz Elliptic Curves

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 τ is defined as follows:

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

where $\overline{\mathbb{F}}_q$ is the algebraic closure of $\mathbb{F}_q$. The Frobenius map τ is a root of the characteristic equation $\chi_E(T) = T^2 - tT + q$ in the ring of endomorphisms End(E). We denote $E(\mathbb{F}_{q^n})$ by the subgroup of $E(\overline{\mathbb{F}}_q)$ consisting of $\mathbb{F}_{q^n}$-rational points. Let G be the subgroup of $E(\mathbb{F}_{q^n})$ generated by P with a prime order ℓ satisfying $\ell^2 \nmid \#E(\mathbb{F}_{q^n})$ and $\ell \nmid \#E(\mathbb{F}_q)$.

We now introduce a generalization of τ-adic NAF into τ-adic w-NAF, which was introduced in [Sol00] on Koblitz curves.

Definition 3. *Let w be an integer ≥ 2. A τ-adic w-NAF is a sequence of digits satisfying the following two conditions:*

1. *Each non-zero digit is an integer which is not divisible by q and whose absolute value is less than $q^w/2$.*
2. *Among any w consecutive digits, at most one is non-zero.*

A τ-adic w-NAF is denoted by $a = (a_{m-1} \cdots a_1 a_0)_\tau$ or $a = \sum_{i=0}^{m-1} a_i \tau^i$.

The length and the weight of a τ-adic w-NAF are defined similarly to w-NAFs. Note that given a τ-adic w-NAF $a = (a_{m-1} \cdots a_1 a_0)_\tau$ and a point $Q \in E$, aQ is computed as $aQ = \sum_{i=0}^{m-1} a_i \tau^i(Q)$.

Theorem 3. *Let $a = (a_{m-1}, \ldots, a_0)_\tau$ and $b = (b_{m'-1}, \ldots, b_0)_\tau$ be two τ-adic w-NAFs. Then $aQ = bQ$ for some nonzero $Q \in G$ implies that $m = m'$ and $a_i = b_i$ for all i if*

$$\max\{m, m'\} \leq M_{q,\ell,w} = \log_q \left(\frac{\ell}{(q^{w/2}+1)^2} \right) - (w-1). \tag{8}$$

Proof. Assume there is a nonzero point $Q \in G$ such that $aQ = bQ$ for two distinct τ-adic w-NAFs $a = (a_{m-1}, \ldots, a_0)_\tau$ and $b = (b_{m'-1}, \ldots, b_0)_\tau$. By adding zero digits to the front of the strings, we may assume $m = m'$. Then we have $O = aQ - bQ = \sum_{i=0}^{m-1} d_i \tau^i(Q)$ for $d_i = a_i - b_i$.

Let $F(T) = \sum_{i=0}^{m-1} d_i T^i$. Since $End(E)$ is an order of the imaginary quadratic field, $F(\tau)$ can be considered as an element of $\mathbb{Z}[i]$ divisible by ℓ. Since $\chi_E(\tau) = 0$, $F(T)$ and $\chi_E(T)$ must have a common root in the algebraic closure of $\mathbb{F}_\ell$. Thus the resultant $R = Res(T^2 - tT + q, F(T))$ satisfies $R \equiv 0 \mod \ell$.

Let τ_1 and τ_2 be the roots of χ_E. Then $R = F(\tau_1)F(\tau_2)$ and $|\tau_1| = |\tau_2| = \sqrt{q}$. For each $\tau \in \{\tau_1, \tau_2\}$, we have

$$\begin{aligned}
|F(\tau)| &\leq \sum_{i=0}^{m-1} |d_i||\tau|^i \leq \sum_{i=0}^{m-1} |a_i||\tau|^i + \sum_{i=0}^{m-1} |b_i||\tau|^i \\
&\leq 2\left(\left\lceil \frac{q^w}{2} \right\rceil - 1 \right) (\sqrt{q}^{m-1} + \sqrt{q}^{m-1-w} + \cdots + \sqrt{q}^{m-1 \bmod w}) \\
&= 2\left(\left\lceil \frac{q^w}{2} \right\rceil - 1 \right) \frac{(q^{(m+w-1)/2} - 1)}{q^{w/2} - 1} < q^{(m+w-1)/2}(q^{w/2}+1).
\end{aligned}$$

Thus, $|R| < q^{m+w-1}(q^{w/2}+1)^2 \leq \ell$. Hence $R = 0$. Because χ_E is irreducible over $\mathbb{Z}$, this implies $\chi_E | F(T)$ over $\mathbb{Z}$.

Assume that d_{i_0} is the lowest nonzero coefficient of F. Then we can write

$$T^{-i_0}F(T) = (g_0 + g_1T + \cdots + g_{m-3-i_0}T^{m-3-i_0})\chi_E(T)$$

for some $g_i \in \mathbb{Z}$. By equating the coefficients, we know $d_{i_0} = qg_0$ and $d_{i_0+j} = qg_j - tg_{j-1} + g_{j-2}$ for $1 \le j \le w-1$, where we set $g_{-1} := 0$ by convention. Since each of a_{i_0} and b_{i_0} is not divisible by q, both a_{i_0+1} and b_{i_0+1} is nonzero. Hence $d_{i_0+1} = \cdots = d_{i_0+w-1} = 0$. That is,

$$Eqn(j) = qg_j - tg_{j-1} + g_{j-2} = 0 \ \text{ for } 1 \le j \le w-1. \tag{9}$$

From $d_{i_0+1} = 0$ and $g_{-1} = 0$, we have $q|g_0$ since $\gcd(q,t) = 1$. By repeating this procedure, we have $g_0, g_1, \ldots, g_{w-2}$ are divisible by q.

After replacing g_i by g_i/q in the $Eqn(1), \ldots, Eqn(c-1)$, we repeat the above procedure to obtain $q^2|g_0, g_1, \ldots, g_{w-3}$. At the end, we have $q^c|g_0$. Therefore, we have $q^w|d_{i_0}$. However, this is impossible since $|d_{i_0}| < q^w$.

The above theorem tells us that distinct τ-adic w-NAFs of length $m < M_{q,\ell,w}$ play an role of distinct group homomorphisms of G. Moreover, if $a = b$ in $\mathrm{End}(E)$, we have $R = Res(T^2 - tT + q, \sum_{i=0}^{m-1}(a_i - b_i)T^i)$ satisfies $R = 0$. By the same argument with Theorem 3, we have $a_i = b_i$ for all i regardless of k, which implies that every endomorphism of E has at most one τ-adic w-NAF.

Theorem 4. *The number of τ-adic w-NAFs of length $\le m$ and weight t is*

$$\binom{m-(w-1)(t-1)}{t} 2^t \left(\left\lfloor \frac{q^w}{2} \right\rfloor - \left\lfloor \frac{q^{w-1}}{2} \right\rfloor \right)^t.$$

Proof. As in Theorem 2, we consider an algorithm to choose t positions out of $m + (w-1) - wt$ positions and fill each of them by $w-1$ consecutive zeros followed by an integer not divisible by q whose absolute value is less than $q^w/2$. By discarding the first $(w-1)$ zeros, we get a τ-adic w-NAF of length $\le m$ with weight t. Conversely, any string with the property can be produced by the algorithm.

Now we count the number of cases. First we have $\binom{m-(w-1)(t-1)}{t}$ choices for t positions. Next, each position is filled by an integer x such that x is not divisible by q and $|x| < q^w/2$. The number of such integers is

$$2^t \left(\left\lfloor \frac{q^w}{2} \right\rfloor - \left\lfloor \frac{q^{w-1}}{2} \right\rfloor \right)^t,$$

which completes the proof.

We introduce an algorithm to output a random secret exponent in a subgroup G of order ℓ in an elliptic curve $E(\mathbb{F}_{q^n})$. Algorithm 3 produces uniformly distributed w-NAFs of length $\le m$ with weight t if $m \le M_{q,\ell,w}$

Algorithm 4 describes batch verification using τ-adic NAF on Koblitz Curves, given (x_i, Q_i) for $1 \le i \le N$. For ease of notation, we describe the algorithm

Algorithm 3 (τ-adic w-NAF Exponent of weight t)

Input: q, m, w, and t
Output: τ-adic w-NAF of length $\leq m$

1: Choose t positions out of $m-(w-1)(t-1)$ positions
2: Fill each position by $(w-1)$ consecutive zeros followed by an integer not divisible by q whose absolute value is less than $q^w/2$
3: Discard the first $(w-1)$ positions of the string
4: Print the string which is a τ-adic w-NAF of length $\leq m$

Algorithm 4 (Batch Verification using τ-adic NAF on Koblitz Curves)

Input: (x_i, Q_i) for $1 \leq i \leq N$
Output: True or false

1: Choose N random elements $c_i = \sum_{j=0}^{e} c_{ij}\tau^j$ $(1 \leq i \leq N)$ from the set of τ-adic w-NAF of length $\leq m$ and weight t, where c_{ij} is an integer not divisible by q whose absolute value is less than $q^w/2$ and $\epsilon_{ij} = c_{ij}/|c_{ij}|$ for nonzero c_{ij} for each i, j.
2: **for** $1 \leq k \leq 2^{w-2}$ **do**
3: $R[2k-1] \leftarrow O$
4: **end for**
5: **for** $j = 0$ to e **do**
6: **for** $i = 1$ to N **do**
7: **if** $c_{ij} \neq 0$ **then**
8: $R[|c_{ij}|] \leftarrow R[|c_{ij}|] + \epsilon_{ij}\tau^j(Q_i)$
9: **end if**
10: **end for**
11: **end for**
12: $Q \leftarrow R[2^{w-1}-1]$
13: $T \leftarrow R[2^{w-1}-1]$
14: **for** $k = 2^{w-2}-1$ to 2 **do**
15: $T \leftarrow T + R[2k-1]$
16: $Q \leftarrow Q + T$
17: **end for**
18: $Q \leftarrow 2Q + T + R[1]$
19: **if** $Q = cP$ **then**
20: Accept all of N instances
21: **else**
22: Reject
23: **end if**

in case of $q = 2$, but it can be easily extended into the general case. For more details, Steps 2-11 compute

$$R_k = \sum_{i=1}^{N} \sum_{j=0, c_j=k}^{m-1} sign(c_j)\tau^j(Q_i)$$

for each odd integer $1 \leq k \leq 2^{w-1} - 1$. Steps 12-18 compute

$$Q = \sum_{k=1,2\nmid k}^{2z-1} kQ_k = (R_{2z-1} + \cdots + R_1) + 2((z-1)R_{2z-1} + (z-2)R_{2z-3} + 2R_5 + R_3)$$

for $z = 2^{w-2}$ where the last term is computed using BGMW method [BGMW93]. From complexity point of view, Step 1 requires $tN - 2^{w-2}$ additions at average and Steps 5-11 require at most $3 + 2(z-1) = 2z + 1 = 2^{w-1} + 1$ additions. Hence the total complexity is at average $tN + 2^{w-2} + 1$ additions with $2^{w-2} - 1$ memory.

Table 7 presents an appropriate weight t and the corresponding attack complexity for each w over Koblitz curve. The length m is taken to be the largest integer to preserve the uniqueness as in Theorem 3. *Additions* is the number of additions to be required for batch verification, where $\#A_{M_e}$ is the number of additions for one scalar multiplication. For example, $\#A_{M_e}$ can be 34 using τ-adic 5-NAF. Note that when enumerating the number of elliptic curve additions, we ignore the τ operations since their cost is negligible; they are implemented merely by a circular shift and very efficient even in polynomial basis.

Table 7. Number of Additions for Batch Verification over a Koblitz Curve

w	m	t	Mem	Additions	Complexity
2	159	15	0	$15N + 2 + \#A_{M_e}$	$2^{81.4}$
3	156	13	1	$13N + 3 + \#A_{M_e}$	$2^{84.5}$
4	154	11	3	$11N + 5 + \#A_{M_e}$	$2^{83.6}$
5	152	10	7	$10N + 9 + \#A_{M_e}$	$2^{86.2}$
6	150	9	15	$9N + 17 + \#A_{M_e}$	$2^{87.1}$
7	148	8	31	$8N + 33 + \#A_{M_e}$	$2^{86.1}$
8	146	7	63	$7N + 65 + \#A_{M_e}$	$2^{83.3}$
9	144	6	127	$6N + 129 + \#A_{M_e}$	$2^{78.5}$
10	142	6	255	$6N + 257 + \#A_{M_e}$	$2^{83.9}$

Table 8 gives a comparison with other methods. We apply the fixed-based τ-adic w-NAF method for fixed base computation with the precomputation. In a single signer case, the proposed method is asymptotically 9 times faster than the individual one.

Table 8. Comparison of Batch Verifications over Koblitz Curve

Method	Exponentiation	Single Signer	Multiple Signers
Individual	$23N$	$57N$	$57N$
[CL06]	$9N + 84$	$9N + 118$	-
Proposed	$6N + 163$	$6N + 186$	$30N + 152$
Proposed/Ind	$0.26 + 7.09/N$	$0.11 + 3.26/N$	$0.53 + 2.67/N$

6 Conclusion

We propose an efficient batch verification method of exponentiation. By applying the proposed algorithm, we can improve the efficiency of batch verification of digital signatures. To the best of our knowledge, we firstly propose a batch verification of signatures by multiple signers so that we can speed up verification of digital signatures about four times faster than individual verification thereof. In particular, our method can be applied to *any* servers or devices that need to verify multiple signatures at once. It would be an interesting problem to apply our algorithm to various applications involving many exponentiations including Mix-Net [Abe99], proof of knowledge, anonymous authentications, and authenticated routing.

Acknowledgements. The authors thank the anonymous reviewers for their valuable comments.

References

[Abe99] M. Abe, *Mix-Networks on Permutation Networks*, Advances in Cryptology - Asiacrypt'99, LNCS Vol. 1716, Springer-Verlag, pp. 258-273, 1999.

[ABGLSV05] A. Antipa, D. Brown, R. Gallant, R. Lambert, R. Struik, and S. Vanstone, *Accelerated Verification of ECDSA Signatures*, SAC 2005, LNCS Vol. 3897, pp.307-318, 2006.

[BGMW93] E. Brickell, D. Gordon, K. McCurley, and D. Wilson, *Fast Exponentiation with Precomputation*, Eurocrypt'92, LNCS Vol. 658, Springer-Verlag, pp. 200-207, 1993.

[BGR98] M. Bellare, J. Garay, and T. Rabin, *Fast Batch Verification for Modular Exponentiation and Digital Signatures*, Proc. of Eurocrypt '98, LNCS Vol. 1403, Springer-Verlag, pp. 236–250, 1998. Full version is available via http://www-cse.ucsd.edu/users/mihir.

[BP00] C. Boyd and C. Pavlovski, *Attacking and Repairing Batch Verification Schemes*, Proc. of Asiacrypt 2000, LNCS Vol. 1976, pp. 58-71, Springer-Verlag, 2000.

[BHLM01] 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.

[CL06] J. Cheon and D. Lee, "Use of Sparse and/or Complex Exponents in Batch Verification of Exponentiations," IEEE. T. on Computers, Vol. 55, No. 12, 2006.(December 2006)

[DSA] *Digital Signature Standard (DSS) (DSA, RSA, and ECDSA algorithms)*, Available at http://csrc.nist.gov/cryptval/dss.htm.

[ECDSA] *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*, ANSI X9.62, approved January 7, 1999.

[Fel87] P. Feldman, *A Practical Scheme for Non-interactive Verifiable Secret Sharing*, IEEE Symposium on Foundations of Computer Science, pp. 427-437, 1987.

[Fiat89] 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.

[HHM00] 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.

[Harn98] L. Harn, *Batch Verifying Multiple DSA-Type Digital Signatures*, Electronic Letters, Vol. 34, No. 9, pp. 870-871, 1995.

[LL94] C.H. Lim and P.J. Lee, *More Flexible Exponentiation with Precomputation*, Proc. of Crypto '94, LNCS Vol. 839, pp. 95-107, Springer-Verlag, 1994.

[MN96] 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.

[MNT04] 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.

[MS06] J. Muir and D. Stinson, *Minimality and Other Properties of the Width-w Non-Adjacent Form*, Mathematics of Computation, Vol. 75, pp. 369-384, 2006.

[NMVR94] 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.

[Sol97] 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/

[Sol00] J. Solinas, *Efficient Arithmetic on Elliptic Curves*, Design, Codes and Cryptography, Vol. 19, No. 3, pp. 195-249, 2000.

[YL95] S. Yen and C. Laih, *Improved Digital Signature suitable for Batch Veriffication*, IEEE Trans. on Computers, Vol. 44, No. 7, pp. 957-959, 1995.