Computer Standards & Interfaces 40 (2015) 42–52

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Improved batch verification of signatures using generalized sparse exponents

Jung Hee Cheon [a], Mun-Kyu Lee [b,*]

[a] Department of Mathematical Sciences, Seoul National University, Seoul 151-742, Republic of Korea
[b] Department of Computer and Information Engineering, Inha University, Incheon 402-751, Republic of Korea

ARTICLE INFO

Article history:
Received 19 August 2014
Accepted 2 December 2014
Available online 30 January 2015

Keywords:
Public key cryptosystem
Digital signature
Elliptic curve
Batch verification
Non-adjacent form

ABSTRACT

We propose an efficient method for batch verification of exponentiation using width-w Non-Adjacent Forms (w-NAFs), which can be applied to modified DSA and ECDSA signatures. We further generalize this method to use tau-adic w-NAF scalars on elliptic curves with complex multiplication such as Koblitz curves. The theoretical analyses and experimental results show that our method accelerates the individual verification by a factor of up to 7.49 in the single-signer case and by up to 1.47 in the multiple-signer case for 1000 instances over a Koblitz curve K233. Our method can also be exploited to accelerate batch verification of pairing-based signatures.

1. Introduction

Batch verification was introduced by Naccache et al. to verify multiple signatures efficiently [25]. 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 a small exponent set $\{0, 1\}^{\ell}$ for small ℓ by Yen and Laih [30] and Bellare et al. [4]. Another improvement [11] was obtained by taking longer integers with small Hamming weights, so called *sparse exponents*, as elements of S rather than small integers.

In this paper, we improve the previous results by adopting generalized sparse exponents which we call *width-w Non-Adjacent Form* (w-NAF for short). First, we define a *w-NAF of weight t with digit set D* as a radix-2 representation satisfying that; (1) each nonzero digit is an element in D, (2) at most one of any w consecutive digits is nonzero, and (3) the number of nonzero digits is t. Its radix-τ analog, i.e., a τ-adic w-NAF, is defined similarly for a complex number τ. Then, by taking random s_is from a set of sparse (τ-adic) w-NAFs of a fixed small weight t, we significantly reduce the cost for signature verification.

We first provide efficient sampling algorithms for w-NAF exponents, which are based on the unique representations for integer w-NAFs and τ-adic w-NAFs, respectively. By using w-NAF exponents together with the legacy simultaneous exponentiation technique, we can show that N exponentiations can be verified with approximately $21N$ multiplications under a medium level of security defined by NIST and ECRYPT II [17], i.e., assuming that the error probability of a batch verification is $< 1/2^{112}$. Note that the costs are $57N$ and $27N$ in the small exponent test [4] and the sparse exponents test [11], respectively. Over an elliptic curve group where a point subtraction is as efficient as a point addition, our verification cost becomes approximately $19N$ point additions for batch verification of N scalar multiplications. This figure drops further over a Koblitz curve. Using sparse τ-adic w-NAFs and our parallelized version of the BGMW scalar multiplication [8], we reduce the verification cost to $8N$ over the standard curve K233.

We also present asymptotic complexity analyses for our methods. According to these analyses, the number of multiplications per exponentiation in our method is sublinear in the size of security parameter, l, i.e., $O(l/\log l)$, which is smaller by a factor of $\log l$ than that of an individual verification, $O(l)$.

We remark that the practical performance can be different from the theoretical estimation. For the basic small exponent tests, some implementations have been reported in the context of pairing-based signatures [16], after the conference version of this paper [12] has been published, and the results comply with the theoretical estimation. For the sparse exponent test, however, it was not clear if there is any gap between theory and practice. In this paper, we attempt to verify the performance of our methods by a real implementation of modified ECDSA [3]. This is the first implementation of batch verification with sparse

* Corresponding author.
E-mail addresses: jhcheon@snu.ac.kr (J.H. Cheon), mklee@inha.ac.kr (M.-K. Lee).

http://dx.doi.org/10.1016/j.csi.2014.12.004
0920-5489/

exponents, and also the first implementation of (modified) ECDSA with secure batch verification.

According to our experimental results, the fastest method, i.e., batch verification using sparse τ-adic w-NAFs, accelerates the individual verification by a factor of 7.49 over K233 when the batch size is 1000 and all the signatures are signed by a single party. This is possible thanks to the surprising result that only 8 elliptic curve additions are sufficient for the proposed method to verify one signature. We obtain similar performance gains for K163 and K283, where our method requires only 6 and 10 point additions per signature, respectively.

It should be noted that through our implementation, we verified that the Frobenius map computation is not negligible, especially in the case that the number of point additions per scalar is extremely small as in batch verification. Therefore, the number of Frobenius maps should be considered as another criterion when we select a verification method from many possible candidates. This observation led us to develop a new batch verification algorithm (Algorithm 4) using τ-adic w-NAFs, which significantly reduces the number of Frobenius maps compared to the algorithm in the conference version of this paper [12]. Another finding is that the secondary operations such as message hashing and big integer operations modulo the subgroup order also have a minor but non-negligible effect on performance.

In this paper, we also speed up the batch verification of signatures generated by multiple signers. The batch verification problem for signatures on different messages generated by different signers was initiated by the conference version of this paper [12]. According to our complexity analysis, our sparse integer w-NAF method is more than 4 times faster than individual verification when the batch size is 1000. Our experimental results show that even for Koblitz curves where the individual verification is already fast due to the dramatic reduction in the number of point additions, our τ-adic w-NAF method accelerates the individual verification by 1.47, 1.47, and 1.55 times in K163, K233, and K283, respectively.

Finally, we show that the generalized sparse exponents can be applied to accelerate the batch verification of various pairing-based signatures, too. To be precise, we consider two example schemes, BLS [6] and ΠΙ-IBS [9], and verify that their batch verification can be improved by 13% and 10%, respectively, if we use generalized sparse exponents instead of small exponents.

1.1. Applications

Batch verification will be useful in any settings where multiple signatures need to be verified at once. We have a variety of applications to which our proposed method may be applied. 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 [23], the query request messages from clients need to be authenticated by servers. Another example is authenticated routing based on public key cryptography, in which network packets are signed and verified by each node and each router has to verify many signatures in real time. We are also able to apply batch verification to Mix-Net [1] to design privacy-preserving systems or protocols. Another promising application would be VSS (Verifiable Secret Sharing) [15], which is a fundamental building block for fault-tolerant and secure distributed computations such as reliable broadcast, peer group membership management, and Byzantine agreement.

1.2. Related works

After the conference version of this paper was presented [12], Camenisch, Hohenberger, and Pedersen [9] proposed the batch verifier for pairing-based short signatures and identity-based signatures. They used the small exponent test to reduce the total number of pairings required for signature verification. Ferrara et al. showed through implementation that the technique in [9] performs well in practice [16]. Akinyele et al. proposed an automated tool for generating batch verification code from a high level representation of a pairing-based signature scheme [2]. As already mentioned above, we also improve the batching algorithm in [9] by applying our generalized sparse exponents in Section 6. On the other hand, there was an attempt [19] to batch verify the original ECDSA instead of the modified ECDSA. The authors of [19] verified the practical performance of their technique through implementation. However, this method was shown to be insecure because it did not randomize the linear combination being verified [5]. If randomization is considered, the method in [19] does not give significant performance gain compared to the individual verification, in particular in the multiple-signer case [20].

1.3. Organization

The remainder of this paper is organized as follows. In Section 2, we define batch verification and explain the modified versions of DSA [25] and ECDSA [3] for batch verification. In Section 3, we propose the new batch verification methods for exponentiation and scalar multiplication based on the sparse w-NAF representations. The proof of uniqueness of these representations is also given as well as the corresponding selection algorithms. In addition, the asymptotic complexities of the new methods are analyzed, and the optimal parameters minimizing the verification cost are presented. The new methods are applied to signature schemes in Section 4, where both the single-signer case and the multiple-signer case are considered. The performance of the proposed verification method is compared with those of the previous methods including the individual verification and the small exponent test through the experiments in Section 5, where we also analyze the various hidden factors that affect the overall performance. In Section 6, we apply our method to pairing-based signatures and estimate the performance. Finally, we conclude in Section 7.

2. Preliminaries

2.1. Notations

Notations used throughout this paper are summarized in Table 1.

Table 1
Notation.

p	Prime number
r	Prime order of a subgroup
G	Abelian group
g	Generator of a multiplicative group
P	Generator of an elliptic curve group
x	Private key
y	Public key over a multiplicative group
Q	Public key over an elliptic curve group
w	Window size
m	Length of exponents
t	Hamming weight
Mem	Number of group elements to be stored
$\mathscr{E}_f$	Finite field exponentiation
$\mathscr{M}_f$	Finite field multiplication
$\mathscr{S}_f$	Finite field squaring
$\mathscr{M}_e$	Scalar multiplication over an elliptic curve
$\mathscr{A}_e$	Elliptic curve addition
$\mathscr{D}_e$	Elliptic curve doubling
$\mathscr{F}_e$	Frobenius mapping