Journal of Information Security and Applications 37 (2017) 15–27

Contents lists available at ScienceDirect

Journal of Information Security and Applications

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

Batch verification of Digital Signatures: Approaches and challenges

Apurva S. Kittur*, Alwyn Roshan Pais

Information Security Research Lab, Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, 575025, India

ARTICLE INFO

Article history:

Keywords:
Digital signatures
Batch verification
Modular exponentiation

ABSTRACT

Digital Signatures can be considered analogous to an ordinary handwritten signature for signing messages in the Digital world. Digital signature must be unique and exclusive for each signer. Multiple Digital Signatures signed by either single or multiple signers can be verified at once through Batch Verification. There are two main issues with respect to Batch Verification of Digital Signatures; first is the security problem and the second is the computational speed. Due to e-commerce proliferation, quick verification of Digital Signatures through specific hardware or efficient software becomes critical. Internet companies, banks, and other such organizations use Batch verification to accelerate verification of large number of Digital Signatures. Many Batch Verification techniques have been proposed for various Digital Signature algorithms. But most of them lack the security requirements such as signature authenticity, integrity, and non-repudiation. Hence there is a need for the study of batch verification of Digital Signatures. The main contributions of our survey include: (a) Identifying and categorizing various Batch verification techniques for RSA, DSS, and ECDSA(includes schemes based on Bilinear Pairing) (b) Providing a comparative analysis of these Batch Verification techniques (c) Identifying various research challenges in the area of Batch verification of signatures.

1. Introduction

In modern world, Digital Signatures [6,18,66] have variety of applications [17,21,57,62]. The applications in industries and organizations have established e-Signature standards based on digital signature technology and certified CAs (Certificate Authority) [68]. Digital Signatures are part of the X.509 standard [33], which is an international, well-understood, standard-based technology. This standard also helps to prevent forgery and modifications to documents once the signature is generated.

Batch Verification technique verifies multiple Digital Signatures quickly at once. This technique has various applications due to its low computation load and time at the verifier, with secured communication. Some of the applications of Batch Verification include air traffic controlling, secured communication in sensor networks, hospital management, online banking transactions etc. Therefore deployment of Batch Verification of Digital Signatures in such scenarios helps in faster and more secure communication.

Fiat [25] was the first to introduce the Batch Verification concept for digital signatures where multiple modular exponentiation operations were performed at the cost of a single Modular Exponentiation. Later many techniques were introduced to verify RSA Digital Signatures in batches. The techniques by L. Harn [31] and others [35,50], identify the existence of invalid signatures in a batch of RSA Digital Signatures. There were several other techniques [16,45,60,63] introduced which can even identify the index of invalid signature/s in a batch of RSA signatures.

Naccache et al. [52] was the first to introduce techniques for batch verification of DSS signatures. Then [30], [64] proposed techniques for identifying existence of invalid signatures in a batch of DSS signatures. Identification of invalid signatures will be more efficient if the techniques also identify the location of the signatures.Lin et al. [47] and Pastuszak et al. [58] introduced techniques which identify the location of bad signature in a given batch of DSS digital signatures.

The Elliptic Curve Cryptography(ECC) was introduced by Koblitz [39] and Miller [49] independently, which was one of the most secure cryptosystems. The Signature scheme based on ECC is known as ECDSA(Elliptic Curve Digital Signature Algorithm). And there are many batch verification techniques introduced to verify these signatures in batches. Few batch verification schemes are developed to speed up the verification of ECDSA signatures [2,19,36] and few are developed to verify signatures based on Bilinear pairing [14,24,29]. In our survey of Batch Verification schemes, we have not considered ID - based techniques because of their certain limitations. One of them is, during any attack, the revocation of credentials for the specific user in the network becomes tedious and

* Corresponding author.
E-mail address: apurva.cs15fv03@nitk.edu.in (A.S. Kittur).

https://doi.org/10.1016/j.jisa.2017.09.005
2214-2126/

time consuming and therefore it is not feasible to deploy in real time scenario.

Various Batch verification techniques have been introduced to verify multiple signatures signed either by single or multiple signers. Our survey makes a comparative study of these techniques and also discusses their properties and instances where they perform the best. RSA, DSS, and ECDSA are the most popularly used Digital Signature algorithms. Therefore we have reviewed batch verification techniques introduced for these algorithms.

The rest of the paper is organized as follows: Section 2 provides the formal definitions and the generic notations followed throughout our survey. Then Section 3 discusses the properties of Digital Signature algorithms, applications and various kinds of attacks possible on the various batch verification schemes. Section 4 is the study of various batch verification techniques proposed for RSA Digital Signatures. Similarly, the Sections 5 and 6 elaborate the Batch Verification techniques proposed for DSS and ECDSA signatures respectively. And in Section 7 we discuss the various Research Challenges. We conclude our paper with Section 8.

2. Definitions

In this section we provide formal definitions of various notions.

- A **Digital Signature Scheme** is actually a systematic study of three probabilistic algorithms *(Gen, Sign, Vrfy)* [37]:
  - *Gen* is the Key Generation algorithm, which takes security parameter 1^n as input and generates the *(pk, sk)* as output, where *pk* is public key and *sk* is private key. We assume that *pk* and *sk* each have length at least *n*, and that *n* can be determined from *pk* and *sk*.
  - *Sign* is the Signing algorithm that takes the private key *sk* and the message *m* as inputs and outputs signature *s*, which can be written as $s \leftarrow Sign_{sk}(m)$.
  - *Vrfy* is the Verification algorithm, which takes the public key *pk*, message *m* and the signature *s* as inputs and outputs *b* whose value is either '1', if the signature is valid or '0', if the signature is invalid. It can be shown as $b \leftarrow Vrfy_{pk}(m, s)$.

  It is required that except with negligible probability over (*pk, sk*) output by $Gen(1^n)$, it holds that $Vrfy_{pk}(m, Sign_{sk}(m)) = 1$ for every (legal) message *m*. Signature *s* is considered valid if $Vrfy_{pk}(m, s) = 1$
- **Batch Verification Algorithm:** Suppose *(Gen, Sign, Vrfy)* is a Digital Signature Scheme with *l* as the security parameter, *k*, $n \in poly(l)$, $PK = pk_1, \ldots, pk_k$ and $(pk_1, sk_1), \ldots, (pk_k, sk_k)$ are generated by $Gen(1^l)$, the Batch Verification Algorithm [4] should hold the following conditions:
  - If $pk_i \in PK$ and $Vrfy_{pk_i}(m_i, s_i) = 1$ for $i \in [1, n]$ then $Batch((pk_1, m_1, s_1), \ldots, (pk_n, m_n, s_n)) = 1$
  - If $pk_i \in PK$ for all $i \in [1, n]$ and $Vrfy_{pk_i}(m_i, s_i) = 0$ for some $i \in [1, n]$, then $Batch((pk_1, m_1, s_1), \ldots, (pk_n, m_n, s_n)) = 0$ except with negligible probability in *l*, over the randomness of *Batch*.
- A bad signature or an invalid signature can be stated as the one, which is signed by an unauthorized signer, or which has been intentionally or unintentionally modified before verification, or the one whose signature is illegal or forged.
- Batch verification algorithms are used to verify the signatures signed using the following three Types:
  - **Type 1**: Single signer uses his private key (*sk*) to generate signatures for multiple messages $(m_1, m_2, \ldots, m_t)$. The signatures are verified in a batch of *t* signatures $(s_1, s_2, \ldots, s_t)$ at once.
  - **Type 2**: Multiple signers use their private keys to sign multiple messages $(m_1, m_2, \ldots, m_t)$. Signatures $(s_1, s_2, \ldots, s_t)$ are verified in a batch of *t* signatures using the batch verification algorithm wherein the signatures correspond to *n* different signers $(2 \le n \le t)$.
  - **Type 3**: The signatures which can not be categorized in Type 1 and 2 can be categorized in this Type.
- Next is, we provide a set of notations followed throughout our study. Refer Table 1:

Table 1
Notations followed in the paper.

Symbol	Reference to
e	Public key in RSA Digital signature
d	Private key in RSA Digital signature
m	message to be signed
h(*m*)	hash value of the message *m*
s	signature generated by RSA Digital Signature algorithm
(*r*, *s*)	signature generated in DSS Digital Signature algorithm
t	batch size of the signatures

2.1. Fast verification of modular exponentiations

Modular exponentiation operations are major operations in digital signature verification. Fast verification of modular exponentiation operation will lead to reduction in batch verification time of the Digital Signatures. In this subsection we discuss three Generic Tests(GT) proposed by Bellare et al. [4] for faster verification of modular exponentiation operations in Digital Signatures.

For generator *g* of *G*, and batch instance $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ for batch verification with $x_i \in Z_p$ and $y_i \in G$, and a security parameter *l*, the three tests verify that $\forall i \in \{1, \ldots, n\} : y_i = g^{x_i}$

- **Random Subset (RS) Test:** It is a repetition of Atomic Random test, *l* times independently and signatures are accepted only if all sub-tests accept:
  The Atomic Random Subset Test can be explained as follows:
  1. Choose randomly $b_i \in \{0, 1\}$, for $i = 1, \ldots, n$ and consider $S = \{i : b_i = 1\}$
  2. Then compute *x* and *y*, $x = \sum_{i \in S} x_i \ mod \ q$ and $y = \prod_{i \in S} y_i$
  3. If $g^x = y$ then accept, else reject
- **Small Exponents (SE) Test:**
  1. Choose randomly $d_1, \ldots, d_n \in \{0, 1\}^l$
  2. Then compute *x* and *y*, $x = \sum_{i=1}^{n} x_i d_i \ mod \ q$, and $y = \prod_{i=1}^{n} y_i^{d_i}$
  3. If $g^x = y$ then accept, else reject
- **Bucket Test:** It considers an additional parameter $m \ge 2$ and sets $M = 2^m$. Then the atomic test is repeated independently $\lceil l/(m-1) \rceil$ times, and accepted only if all sub-tests accept:
  The Atomic Bucket Test is explained as follows:
  1. Pick randomly $t_i \in \{1, \ldots, M\}$, for each $i = 1, \ldots, n$
  2. Consider $B_j = \{i : t_i = j\}$, for $j = 1, \ldots, M$
  3. Compute c_j and d_j for each $j = 1, \ldots, M$, $c_j = \sum_{i \in B_j} x_i \ mod \ q$, and $d_j = \prod_{i \in B_j} y_i$
  4. Verify the instance $(c_1, d_1), \ldots, (c_M, d_M)$ with the Small Exponents Test with security parameter set to *m*

3. Digital Signatures: properties, applications, and threats

In this section, we discuss the properties every Digital Signature algorithm must satisfy. And we also list the applications whose efficiency is significantly increased by Batch verification of Digital Signatures. We also elaborate on the various possible threats for verification of Digital Signatures in batches.

3.1. Properties

Digital Signatures are used to verify the following properties:

- **Signature Authenticity -** Verifies that the signature is actually signed by the claimed signer.