Computers and Mathematics with Applications 61 (2011) 1705–1723

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Provably secure server-aided verification signatures

Wei Wu[a,*], Yi Mu[a], Willy Susilo[a], Xinyi Huang[b]

[a] Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

[b] Institute for Infocomm Research (I^2R), Singapore

ARTICLE INFO

Article history:
Received 28 June 2010
Received in revised form 26 January 2011
Accepted 26 January 2011

Keywords:
Server-aided verification
Provably secure
Digital signatures

ABSTRACT

A server-aided verification signature scheme consists of a digital signature scheme and a server-aided verification protocol. With the server-aided verification protocol, some computational tasks for a signature verification are carried out by a server, which is generally untrusted; therefore, it is very useful for low-power computational devices. In this paper, we first define three security notions for server-aided verification signatures, i.e., existential unforgeability, security against collusion attacks and security against strong collusion attacks. The definition of existential unforgeability includes the existing security requirements in server-aided verification signatures. We then present, on the basis of existing signature schemes, two novel existentially unforgeable server-aided verification signature schemes. The existential unforgeability of our schemes can be formally proved both without the random oracle model and using the random oracle model. We also consider the security of server-aided verification signatures under collusion attacks and strong collusion attacks. For the first time, we formally define security models for capturing (strong) collusion attacks, and propose concrete server-aided verification signature schemes that are secure against such attacks.

1. Introduction

For power-constrained devices such as smart cards and mobile terminals, additional care must be taken with cryptographic algorithms due to the limited computational ability of those devices. Several techniques (e.g., pre-computation and off-line computation) have thus been introduced and adopted in order to improve the efficiency of cryptographic protocols. While such techniques can reduce the computational load, the computational requirement of many cryptographic systems (especially those with excellent security features) still remains too heavy for low-power devices. Pairing computation on elliptic curves is an example. Due to its elegant properties, pairing has been widely employed as a building block for lots of cryptographic schemes, in particular in the construction of identity-based encryption and short signatures. However, performing a pairing on an elliptic curve requires much more computational cost than executing an exponentiation computation. It remains a challenging task to reduce the computational cost in pairing-based cryptography.

A promising solution is to employ a powerful server assisting the low-power device (we refer to this as the *client*) to carry out cryptographic operations. This is known as "server-aided computation". If the server is fully trusted, computations can be easily done through a secure channel between the client and the server. As an example, the client can send his/her private key to the server, which then can act on behalf of the client to decrypt ciphertexts or sign messages, and return the result to the client. However, the assumption of a fully trusted server seems too strong in practice, since clients are more likely to

[*] Corresponding author.
E-mail addresses: ww986@uow.edu.au, weiwu81@gmail.com (W. Wu), ymu@uow.edu.au (Y. Mu), wsusilo@uow.edu.au (W. Susilo), xyhuang81@gmail.com (X. Huang).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.01.036

face an untrusted server which could try to extract the secret of the client (in decryption or signing) or respond with a false result (in encryption or signature verification).

Many schemes for server-aided computation [1–12] have been proposed in the literature. A server-aided-verification signature scheme SAV-Σ consists of a digital signature scheme (Σ) and a server-aided verification protocol. Signatures can be verified by executing the server-aided verification protocol with the server, where the verification requires less computation than the original verification algorithm of Σ. This notion was introduced by Quisquater and De Soete [13] for speeding up RSA verification with a small exponent. Lim and Lee [14] introduced this idea into discrete-logarithm-based schemes, by proposing efficient protocols for speeding up the verification of discrete-logarithm-based identity proofs and signatures. Their constructions are based on the "randomization" of the verification equation [14]. A different approach was introduced by Girault and Quisquater [15], which does not require pre-computation or randomization. Their server-aided verification protocol [15] is computationally secure, based on the hardness of a sub-problem of the underlying complexity problem in the original signature scheme. Hohenberger and Lysyanskaya considered server-aided verification in the situation where the server is made up from two untrusted software packages, which are assumed not to communicate with each other [14]. Under this assumption, it allows a very light public computation task (typically one modular multiplication in the Schnorr scheme). Girault and Lefrance [16] proposed a more generalized model of server-aided verification without the assumption in [14]. A generic server-aided verification protocol for digital signatures from bilinear maps was also proposed [16]. Their protocol can be applied to signature schemes with similar constructions, such as the BB signature scheme [17] and the ZSS signature scheme [18].

Motivations and Contributions.

The motivations of this paper are desires to formally define the security of server-aided verification signatures and to construct new schemes that are secure under realistic security models. Giraul and Lefrance [16] made the first attempt to define the security of server-aided verification signatures. Their definition consists of two aspects, namely the existential unforgeability of the signature scheme and the soundness of the server-aided verification protocol. The former notion is the same as that for digital signatures, and the latter requires that the server be unable to prove an invalid signature as valid using the server-aided verification protocol. Although this definition captures the essence of server-aided verification signatures, it is still worthwhile to define more elaborate models for further research on server-aided verification signatures. The contributions of this paper include three security models of server-aided verification signatures and concrete schemes secure under the new models. Our contributions are outlined as follows.

First, we introduce and define the existential unforgeability of server-aided verification signatures (or EUF-SAV-Σ for short). For server-aided verification signatures, we prove that EUF-SAV-Σ includes the existential unforgeability of signature schemes and the soundness of server-aided verification protocols, under the same assumption [16], that the server does not have any valid signature of the message when it tries to prove a signature of that message as valid. An existentially unforgeable server-aided verification signature scheme ensures that even the server (without colluding with the signer) is not able to forge a signature which can be proved to be valid by using the server-aided verification protocol.

Second, we consider the security of the ZSS signature [18] with the server-aided verification protocol proposed in [16]. The analysis shows that the server-aided verification ZSS in [16] can be made secure in our model, but requires more computational cost than that claimed in [16]. This is due to the difference between the security model defined in this paper and that in [16]. In our model, the server is allowed to execute the server-aided verification protocol with the verifier before proving to the verifier that an invalid signature is valid. This, however, is not allowed in [16] when making the security analysis of the server-aided verification ZSS. We believe that our model reflects a realistic case in real life.

Third, we introduce the server-aided verification for the Waters signature [19] and the BLS signature [20], respectively. We provide the first construction of SAV-Waters and SAV-BLS. SAV-Waters inherits the desirable property of the Waters signature, which can be proven to be existentially unforgeable without random oracles under the GBDH assumption. The existential unforgeability of SAV-BLS can be reduced to the hardness of the BDH problem in the random oracle model.

Last, we consider collusion between a signer and a server, who collaboratively prove an invalid signature to be valid. Such attacks were first sketched in [16] in the definition of "auxiliary non-repudiation". Previous definitions (including EUF-SAV-Σ) are all based on the assumption that the malicious server does not have any valid signature of the message when it tries to prove an invalid signature of that message to be valid. For the first time, this paper formally defines security models for capturing the collusion attack and its stronger version in server-aided verification signatures, and proposes concrete server-aided verification signature schemes for protection against collusion attacks.

Paper Organization.

The rest of this paper is organized as follows. In Section 2, we define the notion of a server-aided verification signature scheme (SAV-Σ) whose existential unforgeability is defined in Section 3. We then analyze the existential unforgeability of a previously proposed SAV-Σ in Section 3. New constructions of existentially unforgeable SAV-Σ and the security analysis are given in Section 4. After that, we define the collusion and adaptive chosen message attacks along with a stronger version in Section 5. Concrete constructions of SAV-Σ secure against such attacks are also given in Section 5. Finally, we conclude this paper in Section 6.[1]

[1] The preliminary versions of two of the proposed protocols were published in [21] without security proofs.