Two Notes on the Security of Certificateless Signatures

Rafael Castro* and Ricardo Dahab

UNICAMP, Brazil
rafael.castro@gmail.com, rdahab@ic.unicamp.br
http://www.lca.ic.unicamp.br/

Abstract. We discuss two common pitfalls found in proofs of security of various certificateless signature (CLS) schemes. As a result of the first observation, we are able to show that a CLS scheme ([Goy06]), previously thought to be secure, is vulnerable to a key replacement attack. We then proceed to define a class of CLS schemes whose security is provable by standard techniques, leading to a more efficient version of a known CLS scheme ([ARP03]) and a (previously unknown) security proof for another ([LCS05]).

Keywords: Certificateless Public-Key Cryptography, Forking Lemma, Signature Schemes.

1 Introduction

Certificateless Public-Key Cryptography *(CL-PKC)* was introduced by Al-Riyami and Paterson in [ARP03] as a compromise between ID-Based Cryptography *(ID-PKC)* [Sha85] and traditional PKIs. Its main design goal is to avoid the inherent key escrow of ID-PKC while taming the complexity of running a full-blown PKI. To accomplish this, the responsibility for the generation of the private key is shared by a trusted Key Generation Center *(KGC)* and the user.

The peculiar setting of certificateless signatures *(CLS)* makes proving security of such schemes somewhat tricky. Otherwise widely used techniques must be applied with care when used in this setting. The Oracle Replay Technique [PS00] is a good example of such a technique. It is a very important tool for proving the security of a large class of signature schemes, the so-called *generic* signature schemes, such as Schnorr's [Sch91], that previously had no security proof. Assuming the existence of an adversary capable of generating signature forgeries, two related signatures on the same message are obtained; these forgeries can then be used to solve a hard problem, thus producing a reductionist security proof. The original formulation of the oracle replay technique, however, does not directly apply to the certificateless setting, since it does not cover the possibility of public-key replacement attacks. Nonetheless, most CLS schemes are proved secure using this technique, leaving open the possibility of such attacks. In this

* This work was supported by FAPESP grant number 2006/06146-3.

W. Susilo, J.K. Liu, and Y. Mu. (Eds.): ProvSec 2007, LNCS 4784, pp. 85–102, 2007.

work we identify the occurrence of such shortcomings in the security proofs of a few CLS schemes, while also providing improved versions for two of these schemes.

Additionally, we present an attack on the CLS scheme from [Goy06], identifying another common inaccuracy, namely the unjustified assumption that an adversary knows the private key corresponding to the public key used in a forgery.

1.1 Organization

In Section 2 we review a few important concepts used throughout this paper. In Sections 3 and 4 we discuss two common pitfalls in the security proofs of CLS schemes, leading us to the main results in this paper. In Section 5 we make a quick summary of the available CLS schemes. Section 6 brings our concluding remarks.

2 Preliminaries

2.1 Bilinear Maps

Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ be groups such that $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T|$. A bilinear map is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ that satisfies the following properties.

1. **Bilinearity.** For all $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}$, $e(aP, bQ) = e(P, Q)^{ab}$
2. **Non-degeneracy.** Let Q be a generator of $\mathbb{G}_2$ and $\psi()$ an homomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$. Then $e(\psi(Q), Q) \neq 1$.

Additionally, we want the map e to be efficiently computable. Such a bilinear map is called *admissible*. In the particular case where $\mathbb{G}_1 = \mathbb{G}_2$, the map is called *symmetric*. Examples of bilinear maps widely used in cryptography are the Weil pairing (as in [BF01]) and the Tate pairing.

2.2 Security Assumptions and Hard Problems

We base our security reductions on a few important definitions presented below.

Definition 1. ***Decision Diffie-Hellman Problem*** **(DDHP).** *Given a multiplicative group* $(\mathbb{G}, .)$*, and* $\alpha, \alpha^a, \alpha^b, \alpha^c \in \mathbb{G}$*, decide whether* $c = ab$*.*

Definition 2. ***Computational Diffie-Hellman Problem*** **(CDHP).** *Given a multiplicative group* $(\mathbb{G}, .)$*, and* $\alpha, \alpha^a, \alpha^b \in \mathbb{G}$*, compute* $X = \alpha^{ab}$*.*

Definition 3. ***q-Strong Diffie-Hellman Problem*** **(q-SDHP).** *Given multiplicative groups* $\mathbb{G}_1, \mathbb{G}_2$*, both with prime order* p*, and the* $(q+2)$*-tuple* $(P, Q, \alpha Q, \alpha^2 Q, \ldots, \alpha^q Q)$*, with* $P \in \mathbb{G}_1$ *and* $Q \in \mathbb{G}_2$*, compute the pair* $(c, \frac{1}{c+\alpha} P)$*, for* $c \in \mathbb{Z}_p^*$*.*