Forward-Secure Threshold Signature Schemes

Michel Abdalla, Sara Miner, and Chanathip Namprempre

Department of Computer Science & Engineering
University of California at San Diego
La Jolla, California 92093
{mabdalla,sminer,meaw}@cs.ucsd.edu
http://www-cse.ucsd.edu/users/{mabdalla,sminer,cnamprem}

Abstract. We construct *forward-secure threshold* signature schemes. These schemes have the following property: even if more than the threshold number of players are compromised, it is not possible to forge signatures relating to the past. This property is achieved while keeping the public key fixed and updating the secret keys at regular intervals. The schemes are reasonably efficient in that the amount of secure storage, the signature size and the key lengths do not vary proportionally to the number of time periods during the lifetime of the public key. Both proposed schemes are based on the Bellare-Miner forward-secure signature scheme. One scheme uses multiplicative secret sharing and tolerates mobile eavesdropping adversaries. The other scheme is based on polynomial secret sharing and tolerates mobile halting adversaries. We prove both schemes secure via reduction to the Bellare-Miner scheme, which is known to be secure in the random oracle model assuming that factoring is hard.

Keywords: threshold cryptography, forward security, signature schemes, proactive cryptography.

1 Introduction

Exposure of a secret key for "non-cryptographic" reasons – such as a compromise of the underlying machine or system, human error, or insider attacks – is, in practice, the greatest threat to many cryptographic protocols. The most commonly proposed remedy is distribution of the secret key across multiple servers via secret sharing. For digital signatures, the primitive we consider in this paper, the main instantiation of this idea is threshold signature schemes [8]. The signature is computed in a distributed way based on the shares of the secret key, and a sufficiently large set of servers must be compromised in order to obtain the key and generate signatures.

Distribution of the key makes it harder for an adversary to learn the secret key, but does not remove this risk. Common mode failures — flaws that may be present in the implementation of the protocol or the operating system being run on all servers— imply that breaking into several machines may not be much harder than breaking into one. Thus, it is realistic to assume that even a distributed secret key can be exposed.

Proactive signatures address this to some extent, requiring all of the break-ins to occur within a limited time frame [13]. This again only ameliorates the key exposure

D. Naccache (Ed.): CT-RSA 2001, LNCS 2020, pp. 441-456, 2001.
© Springer-Verlag Berlin Heidelberg 2001

problem. Once a system hole is discovered, it can quite possibly be exploited across various machines almost simultaneously.

A common principle of security engineering is that one should not rely on a single line of defense. We suggest a second line of defense for threshold signature schemes which can mitigate the damage caused by complete key exposure, and we show how to provide it. The idea is to provide *forward security*.

Forward security for digital signature schemes was suggested by Anderson [2], and solutions were designed by Bellare and Miner [3]. The idea is that a compromise of the *present* secret signing key does not enable an adversary to forge signatures pertaining to the *past*. (In this light, the term "backward security" may have been more appropriate, but we decide to be consistent with existing terminology in the literature.) Bellare and Miner [3] focus on the single-signer setting and achieve this goal through the *key evolution paradigm*: the user produces signatures using different secret keys during different time periods while the public key remains fixed. Starting from an initial secret key, the user "evolves" the current secret key at the end of each time period to obtain the key to be used in the next. She then erases the current secret key to prevent an adversary who successfully breaks into the system at a later time from obtaining it. Therefore, the adversary can only forge signatures for documents pertaining to time periods after the exposure, but not before. The integrity of documents signed before the exposure remains intact.

Combining forward security and threshold cryptography will yield a scheme that can provide some security guarantees *even if* an adversary has taken control of *all* servers and, as a result, has completely learned the secret. In particular, she cannot forge signatures as if they were legitimately generated *before* the break-in. The complete knowledge of the secret signing key is useless for her with regard to signatures from "the past." [1]

It is worth noting that, at first glance, forward-secure signature schemes and signature schemes based on secret sharing can be viewed as two different alternatives for addressing the same problem, namely the key exposure problem. However, in fact, the two provide complementary security properties. Forward security prevents an adversary from forging documents pertaining to the past *even if* he is able to obtain the current secret key. On the other hand, threshold and proactive signature schemes make it harder for an adversary to learn the secret key altogether. The crucial distinction between the two notions is that forward security involves *changing the actual secret* while a secret sharing scheme distributes the secret which remains *unchanged* throughout the execution of the protocol. This is true for *both* threshold *and* proactive schemes. In particular, the refresh steps performed in a proactive scheme update the shares of the secret, but not the secret itself. Therefore, without forward security, if an adversary ever successfully obtains this secret, the validity of all documents signed by the group can be questioned, regardless of when the documents were claimed to have been signed.

Furthermore, one can think of the addition of forward security to threshold schemes as a deterrent to attempts at exposing the key. Specifically, in a forward-secure scheme, a stolen key is less useful to an adversary (i.e. it can't help her forge past signatures) than

[1] A related idea involving key evolution and distribution of secrets was presented in the context of key escrow [5]. However, in their work, the public key needs to be updated, which, in turn, requires the participation of a trusted third party in every time period.