Winter 2018 CS 485/585 Introduction to Cryptography

LECTURE 15

Portland State University
Lecturer: Nate Launchbury

Feb. 27, 2018

DRAFT NOTE. VERSION: March 6, 2018. Email fang.song@pdx.edu for comments and corrections.

Agenda

- (Last time) PKE
- Digital signature INTRO
- Review of HW3
- Quiz3

Defining digital signatures

Definition 1 (KL-12.1)**.** A digital signature scheme consists PPT algorithms (G, S, V) such that:

1. G: $(pk, sk) \leftarrow G(1^n)$.
2. S: on input sk and message m, outputs $\sigma \leftarrow S_{sk}(m)$.
3. V: on input pk and message-signature pair (m, σ), output $V_{pk}(m, \sigma) =$ acc/rej.

Software update example How to use a signature scheme?

- MS generates (pk, sk). Publish pk.
- Software patch m is signed under sk: $\sigma \leftarrow S_{sk}(m)$.
- User downloads (m, σ), and verifies σ with pk, $V_{pk}(m, \sigma) = b$. If $b =$ acc, execute m and complete update.

Comparison to MAC. DS and MAC both protect date integrity. One drawback of DS, as in PubKE, is that it is usually less efficient than MAC. Otherwise DS are advantageous in many aspects.

- No need to share a private-key with each user. Everyone just generates its (pk_i, sk_i) pair and makes pk_i public.
- Publicly verifiable and transferable.
- Non-repudiation. Once a sender signs a message, s/he cannot later deny having done so.

A common misconception. Signature is often considered the “inverse” of public-key encryption. Use decryption as signing, and encrypting as verification. This is *totally unsound*!

Security of Digital signature. Intuitively, we would need that no adversary without knowing the secret key can produce valid signatures. The formal definition is similar to that of MAC, considering chosen-message-attacks where an adversary may ask to see signatures on messages of its choice.

FS NOTE: Draw sig-forge game

1. CH generates $(pk, sk) \leftarrow G(1^n)$.
2. $\mathcal{A}$ is given pk and access to signing oracle $S_{sk}(\cdot)$. Let $\mathcal{L} := \{m_i\}$ be the set of messages that $\mathcal{A}$ has queried. At the end $\mathcal{A}$ outputs (m^*, σ^*).
3. We say that $\mathcal{A}$ succeeds if 1) $V_{pk}(m^*, \sigma^*) = \text{acc}$; and 2) $m^* \notin \mathcal{L}$. Define the output of the game $\textsf{Sig-forge}_{\mathcal{A},\Pi}(n) = 1$ iff. $\mathcal{A}$ succeeds.

Figure 1: Signature forgery game $\textsf{Sig-forge}_{\mathcal{A},\Pi}(n)$

Note that implicitly, $\mathcal{A}$ also has access to the verification procedure $V_{pk}(\cdot)$.

Definition 2 ([KL: Def. 12.2]). A signature scheme $\Pi = (G, S, V)$ is existentially unforgeable under an adaptive chosen-message attack (eu-cma-secure), if for all PPT $\mathcal{A}$,

$$\Pr[\textsf{Sig-forge}_{\mathcal{A},\Pi}(n) = 1] \leq \text{negl}(n)\,.$$

We will see constructions next time.

Hash-and-Sign

Suppose that we already have a secure signature scheme that can sign messages of fixed length $\ell(n)$. How to sign longer messages? Hash function will save our day, as in hash-and-MAC.

Let $\Pi = (G, S, V)$ be a signature scheme for messages of length $\ell(n)$, and let $H : \{0,1\}^* \rightarrow \{0,1\}^{\ell(n)}$ be hash function. Construct a new signature scheme $\Pi' = (G', S', V')$

1. $G' = G$: $(pk, sk) \leftarrow G(1^n)$.
2. S': on input message $m \in \{0,1\}^*$, output $\sigma \leftarrow S_{sk}(H(m))$.
3. V': accept (m, σ) iff. $V_{pk}(H(m), \sigma) = 1$.

Theorem 3 ([KL: Thm. 12.4]). *If Π is a secure signature and H a collision resistant hash, then construction above Π' is a secure signagure (for arbitrary-length messages).*

Review HW3