COMS W4995 Introduction to Cryptography **October 30, 2003**

Lecture 18: Digital Signature Schemes

Lecturer: Tal Malkin *Scribes: Hua Shuo Cai*

Summary

Definition of Digital Signature scheme. Brief description of Paradox. And construction of Lamport's One-Time Signature for one bit message.

1 Authenticity

How do you know that relieved message is from who you think it is? And that it wasn't tampered with?

Adv can intercept the cipher and change the cipher.

We have two authentication schemes:

- *DigitalSignature* - Public Key World.
- *MessageAuthenticationCode* - Private Key World.

2 Digital Signature

Definition 1 *A Digital Signature Scheme is a triple of algorithms* $\{GEN, SIGN, VER\}$ *such that:*

- *GEN (The key generate algorithm) - a randomized algorithm that on input* 1^k *outputs* (PK, SK).
- *SIGN (The Signing algorithm)- a (Possibly randomized algorithm) that on input* SK*,m output a signature* σ.

$$\sigma \leftarrow SIGN_{SK}(m)$$

- *VER (The Verification algorithm) - takes as input PK,m,σ outputs accept or reject.*

$$\{accept, reject\} \leftarrow VER_{PK}(m, \sigma)$$

Associated with the scheme is a message space M_k. We require $\forall m \in M_k$,

$$VER_{PK}(m, SIGN_{SK}(m)) = accept.$$

For every signature that VER accept is valid, otherwise, the signature is invalid.

Example 1 *: Here is a first attempt at designing a signature scheme based on any TDP, such as RSA. Assume RSA keys $PK = (n, e)$, $SK = (n, d)$*

$$\begin{aligned} SIGN_{SK}(m) &= m^d \ mod \ n \\ Ver_{PK}(m, \sigma) \ : \ if \sigma^e &= m \ mod \ n, accept \\ else \ reject & \end{aligned}$$

This is not secure. For example, Eve can choose σ and compute $m = \sigma^e \ mod \ n$, and she has a valid SIGN(m,σ). Also, if Eve know, σ_1 a sign for m_1,σ_2 a sign for m_2, Eve can compute $\sigma = \sigma_1\sigma_2 \ mod \ n$, which is a valid signature on (m_1m_2)

So we see, that while the RSA assumption provides some amount of security (e.g, Eve cannot *always* forge a signature on every m). It is not really secure: she *can* forge many messages. Our definition of security will disallow it.

3 Security Definition

We require that even with chosen message attack *no* signature can be forged.

Definition 2 (GMR 84) *: A sign scheme $\{GEN, SIGN, VER\}$ is secure (existentially unforgeable under adaptive chosen message attack) if $\forall$ PPT A $\exists$ a negligible ε, such that*

$$\Pr\left[(PK, SK) \overset{\mathrm{R}}{\leftarrow} GEN(1^k) \ : \ A^{SIGN_{SK}()}(PK) forges\right] \leq \varepsilon(k)$$

Where "A forges" means A produces (m, σ) such that:

- $VER_{PK}(m, \sigma) = accept.$
- *m was never a query to the signing oracle*

4 The Paradox

Before 1984, people believed that no secure signature scheme exist,using the following (wrong) argument

To prove a signature scheme secure, show that if A breaks the scheme, then it can construct B that breaks your security assumption (e.g. RSA). B runs A, and uses A's forgery to break assumption. But B cannot use A unless it can itself answer As CMA queries. $\Rightarrow$ cannot prove forgery impossible without forging signatures yourself. However, as we will see, this reasoning is wrong.

5 Lamport's one-time signature

We start with signature scheme that are only secure for signing one message. (The definition will stay the same, except only one query to the oracle is allowed).

Let f be an OWF. We show how to sign one-bit message.

$$\begin{aligned} GEN(1^k), choose X_0, X_1 \xleftarrow{\mathrm{R}} \{0,1\}^k & \\ Set \quad & Y_0 = f(X_0),\ Y_1 = f(X_1) \\ Output \quad & PK = Y_0, Y_1\ SK = X_0, X_1 \end{aligned}$$

$SIGN_{SK}(b) = \begin{cases} X_0 & \text{if b} = 0 \\ X_1 & \text{if b} = 1 \end{cases}$
$VER_{PK}(b, \sigma) =$ accept if and only if $f(\sigma) = Y_b$.

Claim 1 *: This is a secure one-time signature, if f is a OWF.*

Proof: : Assume $\exists$ PPT A that forges with non-negligible probability δ, then B that inverts f with non-negligible probability will work as follows:

- B:(Y)
  - choose b $\in$ 0,1 at random.
  - set $Y_b = Y$
  - set $Y_{1-b} = f(X_{1-b})$ for random X_{1-b}
  - run A (Y_0, Y_1)
    if A asks for Sign(b), abort
    if A asks for Sign(1-b), give it X_{1-b}
    output A's output

If A success in forging. Its output is (b, X_b) then B inverted Y. $\Rightarrow$ B inverts f with probability $\delta/2$ (non-negligible).

■