CSE 594 : Modern Cryptography 03/21/2017

Lecture 13: Digital Signatures

Instructor: Omkant Pandey Scribe: Meghana Doppalapudi, Venkata Kedarnath Pakala

1 Digital Signatures

Earlier we studied about message authentication codes where both signer and verifier need to share a key. In contrast, digital signatures are used in real life scenarios where sender needs to communicate with multiple receivers.

2 Properties of Digital Signatures

2.1 Public verifiability

If a receiver verifies that signature is legitimate then it is legitimate for all other receivers. Unlike MAC, this reduces overhead on the sender as he has to compute a single signature. This feature is not present in MAC where sender shares a seperate key with each receiver and computes seperate MAC tag for each receiver.

2.2 Transferability

A signature σ on message m can be shown to a third party who can then verify that σ is a legitimate signature on message m with respect to sender's public key. The signature can be copied and transferred from third party to other parties to prove the authenticity of sender.

2.3 Non repudiation

Once the sender has signed the message, later he cannot deny his signature. This is important if the recipient has to prove to the third party that the message was indeed signed by the sender.

3 Definition

It is similar to definition of MAC with the difference that the key generation generates a pair of keys and verify stage uses a public key instead of secret key of the sender.

By definition, a scheme is digital signature if it satisfies the following 3 conditions :

1. Key generation : $Gen(1^n)$ is a PPT algorithm that generates a pair of keys (p_k, s_k). These are called public and private keys respectively.

2. Signing Algorithm : $Sign_{s_k}(m)$ is a PPT algorithm that takes a private key s_k of sender and message m and outputs a signature σ, denoted as $\sigma \leftarrow Sign_{s_k}(m)$

3. Verification : $Vrfy_{p_k}(m, \sigma)$ is PPT algorithm that takes a public key p_k, message m and signature σ and outputs a bit b which is 1 if the signature is valid and 0 if invalid, denoted as $b = Vrfy_{p_k}(m, \sigma)$.

Similar to MAC the above scheme must satisfy

1. Correctness : If σ is a valid signature on message m corresponding to the public key p_k then the algorithm must give 1 for $Vrfy_{p_k}(m, \sigma)$
2. Unforgeability : No non uniform PPT algorithm can forge or create a valid signature σ on message m s.t. it passes the verification step. i.e. $Pr[Adv$ forges the signature$] \leq v(n)$, where v(n) is a negligible function

4 Security of Digital Signatures

A digital signature scheme is secure if an adversary cannot output a forgery even if he is allowed to obtain many messages of his choice i.e. if adversary gets message , signature pairs $(m_1, \sigma_1), (m_2, \sigma_2), ...$ from the challenger, adversary should not be able to generate a σ for a new message

Game Definition:
Signature Forging Game :
A signature scheme is considered secure if an adversary wins a forging game with negligible probability. The game proceeds between adversary and challenger in 3 steps:

1. Initialization : Challenger runs $Gen(1^n)$ to obtain key pairs (p_k, s_k)
2. Learning : Adversary sends messages $m_i \in M$ to Challenger
   Challenger sends back a signature $\sigma_i \leftarrow Sign(s_k, m_i)$
   Let L=m_i be set of messages that A sends to challenger.
3. Guess : A outputs a message signature pair (m, σ) for a message not in the set L and A wins if and only if $m \notin L \wedge Verify(m, k, \sigma) = 1$

5 Lamport's One Time Signature : Construction

A signature scheme is unforgeable under a single message attack or is a one time signature scheme, if for all probabilistic polynomial time adversaries A, there exists a negligible function v(n) such that

$$\mathbb{P}[Sig - forge_A^{1-time}(n)] \leq v(n)$$

Let f be one way function
The public key p_k consists of n bits defined as
$p_k : \begin{pmatrix} y_1^0 & y_2^0 & .... & y_n^0 \\ y_1^1 & y_2^1 & .... & y_n^1 \end{pmatrix}$ where $y_i^b \leftarrow \{0,1\}^n \forall i \in [n]$ and b $\in \{0,1\}$
Similarly, $s_k : \begin{pmatrix} x_1^0 & x_2^0 & .... & x_n^0 \\ x_1^1 & x_2^1 & .... & x_n^1 \end{pmatrix}$ where $x_i^b \leftarrow \{0,1\}^n \forall i \in [n] and b \in \{0,1\}$
$Sign_{s_k}(m) : \sigma := \begin{pmatrix} x_1^{m_1} & x_2^{m_2} & .... & x_n^{m_n} \end{pmatrix}$

$Ver_pk(m, \sigma)$: Accept if $f(\sigma_i) = y_i^{m}{}_i \forall i \in [n]$; reject otherwise

6 Security of One-time Signature Scheme

6.1 Proof by contradiction

Let A be a probabilistic polynomial time adversary who wins the SigForgingGame with noticeable probability ϵ. We construct a new adversary B that inverts the one-way function f with probability $\frac{\epsilon}{poly(n)}$
Let m and m' be two different messages chosen by A. A requests the signing oracle for signature σ of message m and as A wins the forging game it outputs σ' for new message m'.

- Let i be the first bit-position where m and m' differ from one another i.e., $m_i \neq m'_i$. It is sure that such an i exists as $m \neq m'$
- This means A inverts f at position i. It sees either ${y_i}^0$ or ${y_i}^1$ but not both and still outputs the second one as forgery.
- We construct B who gets a challenge z such that z = f(x) for One-Way function. It chooses random location (i, b) and sets ${y_i}^b$ = z
- Now, B uses adversary A and inverts y with probability more than $\frac{\epsilon}{2n}$. But security of f implies ϵ to be negligible which means one-time signature is secure.

7 One-time Signatures for Long Messages

7.1 Recap of CHRF

For a single hash function h which compresses its input, it is difficult to guarantee collision resistance i.e., there may certainly exist two inputs x and x′ that comprise a collision. Hence, family of collision resistant hash functions is introduced.

7.1.1 Definition (CHRF)

A set of functions H = $\{h_i : D_i \rightarrow R_i\}$ i∈I is a family of collision resistant hash functions (CRHF) if:

- (Easy to sample) There is a PPT algorithm Gen s.t Gen(1^n) ∈ I
- (Compression) $|R_i| \leq |D_i|$
- (Easy to evaluate) There is a PPT algorithm for evaluation (Eval) s.t $\forall x \in |D_i|$ and $\forall i \in I, Eval(x, i) = h_i(x)$
- (Collision resistance) ∀ non uniform PPT adversary A, there is a negligible function μ such that ∀ n ∈ N:
  Pr[i ← $Gen(1^n)$, (x, $x\prime$) ⇐ A(1^n, i): x ≠ x' ∧$h_i(x) \neq h_i(x')$] ≠ $\mu(n)$

7.2 One-time Digital Signature Scheme for $\{0,1\}^*$

We use family of CRHFs to construct a one-time signature scheme $(Gen', Sign', Ver')$ for messages in $\{0,1\}^*$. We define One-time Digital Signature for long messages by:

$Gen'(1^n)$: Run the generator (pk,sk) $\leftarrow Gen_{Sig}(1^n)$ and sampling function i $\leftarrow Gen_{CRH}(1^n)$. Output $pk' =$ (pk, i) and $sk' =$ (sk, i).

$Sign'_{sk}(m)$: Sign the hash of message m. Output $Sign_{sk}(h_i(m))$

$Ver'_{pk}(\sigma, m)$: Verify σ on the hash of m: Output $Ver_{pk}(h_i(m), \sigma)$

7.3 Multi-message Signatures (via chain)

The idea is to generate new keys for each new message that should be signed. So, one-time digital signature scheme (Gen, Sign, Ver) can still be used. We start with a key-pair $(sk_0, pk_0) \leftarrow Gen(1^n)$.

To sign first message m_1:

- Generate a new key-pair for the next message: $(sk_1, pk_1) \leftarrow Gen(1^n)$
- Create signature $\sigma_1 = Sign_{sk_0}(m_1||pk_1)$ on the concatenation of message m_1 and new public key pk_1
- Output $\sigma_1{}' = (1, \sigma_1, m_1, pk_1)$

So, each signature attests to next public key. To sign second message m_2, generate key pair (sk_2, pk_2), create signature $\sigma_2 = Sign_{sk_1}(m_2||pk_2)$ and output $\sigma_2{}' = (1, \sigma_2, \sigma_1{}', m_2, pk_2)$
Previous signature $\sigma_1{}'$ is included to show that previous public key is correct.

7.3.1 Improving size of signature

One way to improve many-message secure digital scheme is to attest two new key pairs instead of one at every step. This builds a balanced binary tree of depth n of key pairs. Each leaf in the tree is associated with one public-private key pair pk, sk and each non-leaf node attests two child nodes. Each of the 2^n nodes can be used. Hence, such a digital signature algorithm can perform up to 2^n signatures with signature size n.

7.3.2 Full-fledged Signature Schemes

Using Merkele trees and other ideas, we can construct a full-fledged scheme from Universal One way hash functions.