Cryptology DATE: 12.11.2015

Lecture 14: RSA Variants; Digital Signatures and Certificates; PKI

Instructor: Goutam Paul *Scribe: Arghya Bhattacharjee*

Attacks on RSA

1. Short Message Small Exponent Attack
   Suppose c is the ciphertext corresponding to the message m. Then, $c = m^e \bmod N$. If $m \ll N$ and $e \ll N$, then $m^e < N$. So c becomes m^e and m becomes $c^{\frac{1}{e}}$.

2. Common Modulus Attack
   Suppose c_1 and c_2 are two ciphertexts corresponding to the exponents e_1 and e_2 respectively, and a common modulus N. Then, $c_1 = m^{e_1} \bmod N$ and $c_2 = m^{e_2} \bmod N$. If $\gcd(e_1, e_2) = 1$ then $\exists x, y$ such that $e_1 x + e_2 y = 1$. Then,

$$\begin{aligned} c_1^x c_2^y \bmod N &= (m_1^e)^x (m_2^e)^y \bmod N \\ &= m^{e_1 x + e_2 y} \bmod N \\ &= m \bmod N. \end{aligned}$$

Theorem (Chinese Remainder Theorem)**.** *Suppose, $a_1 = x \bmod N_1$, $a_2 = x \bmod N_2$, ..., $a_m = x \bmod N_m$, where x is unknown, but $a_1, a_2, \ldots, a_m$ are known along with $N_1, N_2, \ldots, N_m$. If $\gcd(N_i, N_j) = 1, \forall i \neq j$, then there exists a unique solution $x \bmod N_1 N_2 \ldots N_m$.*

Proof. Let $M_i = \prod_{j=1, i \neq j}^{m}$. Then $\gcd(M_i, N_i) = 1, \forall i \in \{1, 2, \ldots, m\}$. Let $b_i = M_i^{-1} \bmod N_i$, $\forall i \in \{1, 2, \ldots, m\}$ and $x = (\sum_{i-1}^{m} a_i M_i b_i) \bmod (N_1 N_2 \ldots N_m)$. Then x is the solution for the m equations mentioned in the statement of the theorem, and is unique. □

3. Common Exponent Attack
   Suppose c_1, c_2 and c_3 are three ciphertexts corresponding to the moduli N_1, N_2 and N_3 respectively, where N_1, N_2 and N_3 are co-prime, and a common exponent e. Then, $c_1 = m^e \bmod N_1$, $c_2 = m^e \bmod N_2$ and $c_3 = m^e \bmod N_3$. Then m^e will have a unique solution modulo $N = N_1 N_2 N_3$(from Chinese Remainder Theorem). If m and e are small enough, such that $m^e < N$, then the eth root of c gives m.

Padded RSA

The idea is to randomly pad the message before encrypting. A general paradigm for this approach is shown in this construction. The construction is defined based on a parameter l that determines the length of messages that can be encrypted.

Construction. Let GenRSA be as before, and let l be a function with $l(n) \leq 2n - 2$ for all n. Define a public key encryption scheme as follows:

1. Key-generation algorithm Gen: On input 1^n, run $\mathsf{GenRSA}(1^n)$ to obtain (N, e, d). Output the public key $pk = \langle N, e \rangle$, and the private key $sk = \langle N, d \rangle$.

2. Encryption algorithm Enc: On input a public key $pk = \langle N, e \rangle$ and a message $m \in \{0,1\}^{l(n)}$, choose a random string $r \leftarrow \{0,1\}^{\|N\|-l(n)-1}$ and interpret $r\|m$ as an element of $\mathbb{Z}_N$ in the natural way. Output the ciphertext $c := [(r\|m)^e \bmod N]$.

3. Decryption algorithm Dec: On input a private key $sk = \langle N, d \rangle$ and a ciphertext $c \in \mathbb{Z}_N^*$, compute $\hat{m} := [c^d \bmod N]$, and output the $l(n)$ low-order bits of $\hat{m}$.

Digital Signature

Definition (signature scheme - syntax)**.** A signature scheme is a tuple of probabilistic polynomial-time algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ satisfying the following:

1. The key generation algorithm Gen takes as input a security parameter 1^n and outputs a pair of keys (pk, sk). These are called the public key and the private key, respectively. We assume for convenience that pk and sk each have length at least n, and that n can be determined from pk, sk.

2. The signing algorithm Sign takes as input a private key sk and a message m from some underlying message space(that may depend on pk). It outputs a signature σ, and we write this as $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$.

3. The deterministic verification algorithm Vrfy takes as input a public key pk, a message m and a signature σ. It outputs a bit b, with $b = 1$ meaning valid and $b = 0$ meaning invalid. We write this as $b := \mathsf{Vrfy}_{pk}(m, \sigma)$.

We require that for every n, every (pk, sk) output by $\mathsf{Gen}(1^n)$, and every message m in the appropriate underlying plaintext space, it holds that $\mathsf{Vrfy}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1$.
We say σ is a valid signature on a message m(with respect to some public key pk that is understood from the context) if $\mathsf{Vrfy}_{pk}(m, \sigma) = 1$.

Secure DSA

Construction. Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a signature scheme for message of length n and $\Pi' = (\overline{Gen}, \overline{H})$ be a hash function, where the output of H has length n on security parameter 1^n.

1. Gen', on input 1^n, runs $\mathsf{Gen}(1^n)$ to obtain (pk, sk) and runs $\overline{Gen}(1^n)$ to obtain s. The public key is $pk' = \langle pk, s \rangle$ and the private key is $sk' = \langle sk, s \rangle$.

2. Sign', on input a private key $sk' = \langle sk, s \rangle$ and a message $m \in \{0,1\}^*$, computes $\sigma' \leftarrow \mathsf{Sign}_{sk}(H^s(m))$.

3. Vrfy', on input a public key $pk' = \langle pk, s \rangle$, a message $m \in \{0,1\}^*$, and a signature σ, outputs 1 if and only if $\mathsf{Vrfy}_{pk}(H^s(m), \sigma) \stackrel{?}{=} 1$.

We can construct a new signature scheme $\Pi' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Vrfy}')$ for arbitrary-length messages as follows: the public key contains a public key pk output by Gen as well as a key s output by $\overline{Gen}$; the private key is simply the one corresponding to sk(that was also output by Gen). To sign a message $m \in \{0,1\}^*$, the signer simply computes $\sigma \leftarrow \mathsf{Sign}_{sk}(\overline{H}^s(m))$. Verification is performed by checking that $\mathsf{Vrfy}_{pk}(\overline{H}^s(m), \sigma) \stackrel{?}{=} 1$.

A Simple Forge on DSA

Suppose σ_1 and σ_2 are two signatures corresponding to the messages m_1 and m_2 respectively. Then, $\sigma_1 = m_1^d \bmod N$ and $\sigma_2 = m_2^d \bmod N$. Then $\sigma_1\sigma_2 \bmod N$ is a valid signature of $m_1 m_2$.

Digital Certificate

In cryptography, a public key certificate (also known as a digital certificate or identity certificate) is an electronic document used to prove ownership of a public key. The certificate includes information about the key, information about its owner's identity, and the digital signature of an entity that has verified the certificate's contents are correct. If the signature is valid, and the person examining the certificate trusts the signer, then they know they can use that key to communicate with its owner. In a typical public-key infrastructure (PKI) scheme, the signer is a certificate authority (CA), usually a company which charges customers to issue certificates for them. In a web of trust scheme, the signer is either the key's owner (a self-signed certificate) or other users ("endorsements") whom the person examining the certificate might know and trust.