CSE 5852, Modern Cryptography: Foundations Fall 2016

Lecture 19

Prof. Benjamin Fuller *Scribe: Chaoqun Yue*

1 Last class

Public key encryption is "autpomatically" secure for multiply messages even if they are adaptively chosen by the attacker. Also introduced 2 use of public key encryption:
1. Hybrid Encryption: use public key to encrypt symmetric key and encrypt MAC of the message by using symmetric key.
2. Key-exchange: use public key to transmit symmetric key and use this symmetric key for all following message.
*However, neither of these methods remove the attacker in the middle problem. The attacker could (blindly) change the key transmitting using the public key encryption and then we don't know how any of our methods are going to work. (This includes the MAC, we don't know what it does if the two parties have different keys.)
*What we really need is a method for an individual to send a message where its clear that the message came from them.

2 Public Key Signature

Signature: Ability for sender to send message and receiver is sure that the mseeage id from sender.
Definition of public key signature:
$(sk, vk) \leftarrow Gen(1^n)$:$sk$-signing key(private).$vk$-verification key(public)
$\sigma \leftarrow Sign(sk, m)$: Only someone who get the sk can produce the signature σ
$1/0 \leftarrow Verify(vk, m, \sigma)$: Someone hold the vk and signature can verify that whether this signature is valid.
Note that signing is not equal to Encryption because there are some public key signing exists but they are not encryption.

2.1 One-time signature Lamport 1979

Suppose we have a message with k bits and we wanna sign it. First choose a PRG function G
$V_k = S, G(S_{1,0}), G(S_{2,0}), ..., G(S_{k,0}), G(S_{1,1}), G(S_{2,1}), ..., G(S_{k,1})$
$S_k = S_{1,0}, S_{2,0}, ..., S_{k,0}, S_{1,1}, S_{2,1}, ..., S_{k,1}$

For message $m = m_1 m_2 ... m_k$, release $\sigma = S_{1,m_1}, S_{2,m_2}, ..., S_{k,m_k}$
$Vfy(V_k, m, \sigma)$ =check that PRG outputs match.
The key fact for this algorithm is that PRG is one-way function.

3 Constructing Multi-time Signature

First, let's back to the Number Theory: discrete log is hard in $\mathbb{Z}_p^*$.

3.1 Composite Modular Arithmetic

As we know that $\mathbb{Z}_p^+$ is a group, now let's consider a group with respect to multiplication mod N, and it's elements should have inverses.
Proposition: let b, N be integers, $b \geq 1, N > 1$. b has an inverse mod N if and only if $gcd(b, N) = 1$.
If b has a inverse mod N, $\exists a$ such that $ba \equiv 1 \mod N$. $ba - 1 = xN$ or $ba - xN = 1$ and $gcd(b, N)$ is the smallest $\aleph$ where α where $\aleph = by + Nx$. If $gcd(b, N) = 1$, then $by + xN = 1 \Rightarrow by \equiv 1 \mod N$.
Now we how that $\mathbb{Z}_N^* = \{b|gcd(b, N)\}$ is a group and it satisfies:

1. Identity $b \equiv 1 mod N$ is the identity.
2. Inverse: We have proved above
3. Commutativity follows from the property on the integers.
4. Commutativity follows from the property on the integers.
5. Associativity follows from the property on the integers.
6. For a, b in group ab in group, because both of them have inverse, ab has inverse b^-1a^-1 so $gcd(ab, N) = 1$

Example of group $\mathbb{Z}_N^*$, consider $N = pq = 3 * 5 = 15$

b	$gcd(b, N)$
1	1
2	1
3	3
4	1
5	5
6	3
7	1
8	1
9	3
10	5
11	1
12	3
13	1
14	1
15	15

It's clear that $|\mathbb{Z}_N^*| = 8$. $|\mathbb{Z}_N^*| = pq - p - q + 1 = (p-1)(q-1)$. We define $\phi(N) = |\mathbb{Z}_N^*|$
We have proved in homework that $a^{p-1} \equiv 1 \mod p$. Actually it holds for any modulus

Theorem 1. *$\forall N$ and $a \in \mathbb{Z}_N^*, a^{\phi(N)} \equiv 1 \mod N$*

Claim: if $i \neq j$, $aa_i \neq aa_j \mod N$
Proof. Suppose the claim is not correct, then $\exists i, j, i \neq j$ such that $aa_i \equiv aa_j \mod N \Rightarrow a_i \equiv a_j \mod N$, which is not possible.Similar to the proof of Fermat's little theorem:
$a_1 a_2 ... a_{\phi(N)} = (aa_1)(aa_2)...(aa_{\phi(N)}) \mod N$. Multiply the inverse to the left, we get:
$a \equiv a^{\phi(N)} \mod N$

Theorem 2. *Fix N and some e such that $gcd(e, \phi(N)) = 1$. $f_e(x) = x^e$ is a permutation(1-to-1) Furthermore for $d = e^{-1} \mod \phi(N)$, f_d is the inverse permutation of f_e.*

4 The Hardness of Factoring

Factoring number seems hard: Given N, find a, b such that $ab = N$. It's hard to find an algorithm run in polynomial time. One easy way to factor: division by $2, 3, ..., \sqrt{N}$, this algorithm runs in $O(\sqrt{N})$ time. Consider the following experiment:
Weak-factoring:

1. Choose random n-bit integer x_1, x_2
2. Multiply $N = x_1 x_2$
3. Give A, N
4. Get back $x_1' x_2'$ Output 1 if $x_1' x_2' = N$

We say that there is no currently known polynomial time algorithm because $\forall PPTA, Pr[Awins] \leq \epsilon(n)$. *Now notice that with good probability N will be even so we can't hope that this problem is hard on average. In fact the probability that N is divisible by a small factor is very high.
*The numbers that are hardest to factor are those that have only large factors. As before we'll assume we have methods to generate large primes. We'll define a new factoring experiment based on this.

Factoring n:

1. Choose random n-bit prime x_1, x_2
2. Multiply $N = x_1 x_2$
3. Give A, N
4. Get back $x_1' x_2'$ Output 1 if $x_1' x_2' = N$

*Note that it is not sufficient to just choose random n bits primes. As this problem has been studied more mathematicians have become better at factoring certain classes of problems so we have to restrict our attention to certain classes of primes.