Lecture 1 - Digital Signatures

Rylan Schaeffer and Vincent Yang

March 23, 2016

Note: This lecture is based on Princeton University's BTC-Tech: Bitcoin and Cryptocurrency Technologies Spring 2015 course.

Cryptography

- If you want to make information secret, hide its existence or make it intelligible
- Symmetric and Asymmetric Cryptography

Symmetric Cryptography

- An encryption scheme *SE = (K, E, D)* in which the sender and receiver use the same key to encrypt and decrypt information.

  Randomized *key generation* algorithm K that returns a string k. *Keys(SE)* is the set of possible bitstrings that K can output (keys).

  Encryption Algorithm E that takes a key $k \in K$.

  decryption algorithm D that inputs a key $k \in$ *Keys(SE)* and *plaintext* $M \in \{0,1\}*$.
  $M \leftarrow D_{\mathrm{K}}(C)$

- Message Space

  The set of possible messages

  Key Space

  Ciphertext Space/Cipher Space

- Caesar Cipher

  Rotate each letter n letters down the alphabet

  Key Space 26 - 1

- One Time Pad

- Diffie-Hellman Key Exchange

  A method of publically creating cryptographic keys.

  Trapdoor Function: A function that is easy to compute in one direction, but not the other.

  Perfect Forward Secrecy: The property in which compromised long-term keys do not invalidate the integrity of a session key.

  modulus = remainder

1. Choose a modulus $p = 23$ and base $g = 5$.

   g is a Primitive root modulo. A number g is a *Primitive root modulo* if every number a coprime to n is congruent to a power g of modulo emphn

   Restated: For every integer a coprime to n, there exists an integer k such that $g^{\mathrm{k}} \equiv a \ (mod\ n)$.

   Restated: g is a generator for the multiplicative group of integers modulo n.

2. Alice chooses a secret integer $a = 6$, and sends Bob $A = g^a \ mod\ p$.

   $A = 5^6 \ mod\ 23 = 8$

3. Bob chooses secret integer $b = 15$, and sends Alice $B = g^b \ mod\ p$.

   $B = 5^{15} \ mod\ 23 = 19$

4. Alice computes $s = B^{\mathrm{a}} \ mod\ p$.

   $s = 19^6 \ mod\ 23 = 2$

5. Bob computes $s = A^{\mathrm{b}} \ mod\ p$.

   $s = 8^{15} \ mod\ 23 = 2$

6. They reached the same number because under mod p:

   $A^{\mathrm{b}} \ mod\ p = g^{\mathrm{ab}} \ mod\ p = g^{\mathrm{ba}} \ mod\ p = B^{\mathrm{a}} \ mod\ p$

   $= (g^a mod\ p)^b \ mod\ p = (g^b \ mod\ p)^a \ mod\ p$

Asymmetric Cryptography

- Properties

  Only you can create a signature, but anyone can verify its validity

  Tied to a document

  Ex: Actual mail

- Public/Private keys

- Digital signature scheme

  $(sk, pk) := generateKeys(keySize)$ Randomized key generation

  $sig := sign(sk, message)$ Encryption Algorithm

  $isValid := verify(pk, message, sig)$ Deterministic Decryption Algorithm

- Ensure only one person can decrypt your message

- Ensure a message was created by someone

- Unforgeability Game

  Challenger has $(sk,\ pk)$ and attacker has pk.

  Attacker can send over $m_0, m_1, m_2 ... m_n$ and get back $sign(sk, m_0)$.

  Attacker has to send $M,\ sig$ where $M \notin m_0, m_1, m_2, ...m_n$.

  Challenger verifies: $verify(pk, M, sig)$.

  Existential Forgery: Creation of a message/signature pair (m, σ), where σ isn't created by signer.

  Existentially Unforgeable: The chance of successfully forging a message is negligible to the point that it will never happen in practice.

- RSA

  Ron Rivest, Adi Shamir, Leonard Adleman

  - Basic Principle: $(m^{\mathrm{e}})^{\mathrm{d}} \ mod\ n = m$

    Even with m, e, and n, it is extremely difficult to find d.

- Key Distribution

  Distribute public key (n, e).

- Encryption

  Change M to integer m. Make sure $0 \leq m < n$ and $gcd(m, n) = 1$ through an agreed padding scheme.

  Compute ciphertext c through $c \equiv (m^{\mathrm{e}})^{\mathrm{d}} \equiv m \ mod \ n$

- Decryption

  Use private key exponent d with $c^{\mathrm{d}} \equiv (m^{\mathrm{e}})^{\mathrm{d}} \equiv m \ mod \ n$

  Then, reverse emphm to M using the padding scheme.

- Key Generation

  Choose different prime numbers p and q. Find $n = pq$.

  n is the key. $len(n) = keyLength$.

  Find $\phi(n)$.

  1. Choose two large, different primes p and q. Find $n = p \cdot q$

     n is the key. $len(n) = key_length$.

  2. Find $\phi(n)$. This turns out to be $(p-1)(q-1)$

     ϕ is Euler's totient function. $\phi(n)$ is the number of positive integers less than n that are coprime to n. $\phi(1) = 1$.

     Two numbers are *Coprime* when the only positive integer that divides them is 1.

     Given n, a prime number, $\phi(n) = n - 1$. e.g. $n = 5$, so $1, 2, 3, 4$ are coprime to 5.

     (a) However, for composite numbers, it works for some but not others.

     (b) e.g. $15 = 3 \cdot 5$ and $\phi(15) = \phi(3) \cdot \phi(5) = 2 \cdot 4 = 8$.

     (c) But doesn't hold for 4, 8, 9.

     If m and n are coprime, $\phi(m \cdot n) = \phi(m) \cdot \phi(n)$.

  3. Find an integer e where $1 < e < \phi(n)$ and $gcd(e, \phi(n)) = 1$. This means e and ϕn are coprime.

  4. Find d, the modular multiplicative inverse of $e(mod \ \phi(n))$

     This means: Solve for d where $d \cdot e \equiv 1 \ (mod \ \phi(n))$

     e should have short length, and is usually $2^{16} + 1 = 65,537$.

     e is the public key, along with n.

     d is the private key.

- Example:

  1. Find two distinct prime numbers

     $p = 11$ and $q = 7$

  2. Find $n = pq$

     $n = 11 \cdot 7 = 77$

  3. Compute the totient of n

     $\phi(77) = (11 - 1)(7 - 1) = 60$

  4. Choose e such that $1 < e < 60$, where e is coprime to *60*.

     Let $e = 17$; check that 60 is not divisible by 17.

  5. Compute d. Process is below.

     $d = 53$

     $d \cdot e \ mod \ \phi(n) = 1$

     $53 \cdot 17 \ mod \ 60 = 1$

  6. The public key is $n = 77$ and $e = 17$.

     $c(m) = m^{17} \ mod \ 77$

  7. The private key is $d = 53$.

     $m(c) = c^{53} \ mod \ 77$

8. To encrypt $m = 65$,
   $c = 65^{17}\ mod\ 77 = 32$
9. To decrypt $c = 32$,
   $m = 32^{53}\ mod\ 77\ = 65$.

- Calculating d for above: use Extended Euclid's Algorithm

  Basically finding *gcd* with Euclid's Algorithm, but reversed.

$$
\begin{aligned}
\phi(77) &= 60 \\
e \cdot d\ mod\ 60 &= 1 \\
17 \cdot d\ mod\ 60 &= 1 \\
60 &= 3(17) + 9 \\
17 &= 1(9) + 8 \\
9 &= 1(8) + 1
\end{aligned}
$$

  Once we find 1, rewrite as:

$$
\begin{aligned}
1 &= 9 - 1(8) \\
8 &= 2(9) - 17 \\
9 &= 60 - 3(17)
\end{aligned}
$$

  Therefore,

$$
\begin{aligned}
1 &= 9 - (17 - 1(9)) \\
1 &= 2(9) - 17 \\
1 &= 2(60 - 3(17)) - 17 \\
1 &= 2(60) - 7(17) \\
\phi(77) - 7 &= d \\
d &= 53
\end{aligned}
$$

- Calculating d for above: use Extended Euclid's Algorithm

  Basically finding *gcd* with Euclid's Algorithm, but reversed.
- Fermat's little theorem states $a^{p} = a\ mod\ p$. This is equal to $a^{p-1} = 1\ mod\ p$.
- For RSA, this is insufficient. You need the Euler-Fermat generalisation:

  $a^{\phi(n)} = 1$

  Group Theory: Multiplication for some sets of integers makes a group under modulo, if all the elements are coprime to the modulo used.

  E is coprime to $\phi(pq)$. Group theory says there exists some integer that acts uniquely as an inverse and transforms under multiplication to the identity.

  The identity element for multiplication is 1. The inverse is d.

Source: `http://crypto.stackexchange.com/questions/388/what-is-the-relation-between-rsa-fermats-little-theorem?lq=1`