CSIT 495/595 - Introduction to Cryptography
Public-Key Encryption and RSA

Bharath K. Samanthula
Department of Computer Science
Montclair State University

Public-Key Encryption

- Introduced by Diffie and Hellman in their seminal paper "New Directions in Cryptography" (1976)
- Suppose Alice wants to send a message to Bob
- In private-key encryption, Alice and Bob agree on some secret key
  - encryption and decryption uses the same secret key
- In Public-key Encryption, Bob generates a pair of keys (pk_b, sk_b), called the public key and private key, respectively
  - Alice uses pk_b to encrypt the message and sends it to Bob
  - Bob decrypts it using her secret key sk_b

Sharing of Public Key

Two approaches

1. Once Bob knows that Alice wants to send him a message, Bob generates (pk_b, sk_b) and sends pk_b to Alice
2. Bob generates (pk_b, sk_b) in advance and disseminates his public key pk_b using one of the following techniques
    - publish pk_b on his webpage
    - put it on his business card
    - place it in a public directory

In either approach, the communication channel between Alice and Bob can be public and thus the attacker might know pk_b

Private-Key vs. Public-Key

Private-Key	Public-Key
Symmetric	Asymmetric
All keys remain private	Only secret key remains private
Secret key is used for communication between those two parties	Multiple senders can communicate with a receiver using his public key

- Observation 1: Public-key encryption is roughly 2 to 3 orders of magnitude slower than private-key encryption
- Observation 2: It can be a challenge to implement public-key encryption in resource-constrained devices, such as smartcards and wireless sensors

Comparison of Cryptographic Primitives

	Private-Key Setting	Public-Key Setting
Secrecy	Private-key encryption	Public-key encryption
Integrity	MACs	Digital Signatures

Secure Distribution of Public keys

- What happens if the adversary actively interferes with the communication?
  - If the honest parties share no keys in advance, then privacy cannot be achieved
- Example:
  - Suppose the attacker is able to replace Alice's public key with pk' (either during transmission or in public directory)
  - When Bob sends a message using pk', the attacker can easily decrypt it
- We assume that senders obtain legitimate copy of receiver's public key (we will see how to get rid of this assumption later on)

Definition

Def: a public-key encryption system is a triple of algs. (G, E, D)

- G(): randomized alg. outputs a key pair (pk, sk)
- E(pk, m): randomized alg. that takes $m \in M$ and outputs $c \in C$
- D(sk,c): det. alg. that takes $c \in C$ and outputs $m \in M$ or $\perp$

Consistency: $\forall$(pk, sk) output by G :

$$\forall m \in M: \quad D(sk, E(pk, m)) = m$$

RSA

- Invented in **1978** by Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman
  - Published as R L Rivest, A Shamir, L Adleman, "*On Digital Signatures and Public Key Cryptosystems*", Communications of the ACM, vol 21 no 2, pp120-126, Feb 1978
- Security relies on the difficulty of factoring large composite numbers
- Essentially the same algorithm was discovered in 1973 by Clifford Cocks, who works for the British intelligence

Recap: Number Theory

Let $N = p \cdot q$ where p,q are prime

$Z_N = \{0,1,2,\ldots,N-1\}$; $(Z_N)^* = \{\text{invertible elements in } Z_N\}$

Facts: $x \in Z_N$ is invertible $\Leftrightarrow$ $\gcd(x,N) = 1$

– Number of elements in $(Z_N)^*$ is $\varphi(N) = (p-1)(q-1) = N-p-q+1$

Euler's thm: $\boxed{\forall x \in (Z_N)^* : x^{\varphi(N)} = 1}$

RSA Key Generation

- Selecte two large prime numbers of the same size p, q
- Compute $N = p * q$ and $\Phi(n) = (p - 1)(q - 1)$
- Select a random integer e such that $1 < e < \Phi(n)$ and $\gcd(e, \Phi(n)) = 1$
- Compute d such that $1 < d < \Phi(n)$ and $ed = 1 \bmod \Phi(n)$
- Output:
  - Public key pk = (N, e)
  - Private Key $sk = d$

Textbook RSA Encryption Scheme

- Encryption:
  given a message m, where $0 < m < N$
  use $pk = (N, e)$ to encrypt it
  $c = m^e \bmod N$
- Decryption:
  given a ciphertext c, use $sk = d$ to decrypt it
  $m = c^d \bmod N = m^{ed} \bmod N = 1,$

Textbook RSA: Example

- Let $p = 17$ and $q = 23 \rightarrow N = 391$
- $\Phi(391) = 16 * 22 = 352$
- Suppose $e = 3$, then $d = 235$
- Public key: (391, 3) and the private key is 235
- Given $m = 158$:
  - *Encryption*: $c = (158)^3 \bmod 391 = 295$
  - *Decryption*: $c^d \bmod 391 = (295)^{235} \bmod 391 = 158$

RSA Security

- Given $c = m^e \bmod N$, can the attacker compute m?
- Security of RSA is based on two assumptions
  - *Factoring* large numbers (say of size 2048 bits)
  - *RSA problem*: compute eth roots modulo N of c (the best known solution to this problem requires factoring N)

Is Textbook RSA Secure??

- Never use textbook RSA
- Key Observation: Textbook RSA is deterministic, that is, it is not CPA-secure
- What is the Solution?
  - Padded-RSA: choose a uniform bit-string r and encrypt $m' = r||m$ instead of m

Useful References

- Chapter 11, Introduction to Modern Cryptography by Jonathan Katz and Yehuda Lindell, 2nd Edition, CRC Press, 2015.
- http://www.cis.upenn.edu/~jean/RSA.pdf
- https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture12.pdf
- http://searchsecurity.techtarget.com/definition/RSA