Cryptography
CS 555

Topic 17: Textbook RSA encryption
Outline and Readings

- Outline
  - One-way functions
  - RSA

- Readings:
  - Katz and Lindell: Chapter 6.0, 6.1.1, 6.1.2, 7.2
Towards One-Way Function

• We know how to use Pseudo-Random Generator (PRG) and Pseudo-Random Function (PRF) to construct encryption schemes and MAC.

• We know what algorithms that are used in practice in instantiate PRG and PRF.
  – But we cannot prove that they are PRG or PRF; we can only assume that they are

• Can we prove that some constructions are PRG or PRF based on something else?
One-Way Function

- A function $f$ is one-way if
  - It is easy to compute
  - It is hard to invert, that is, given $y=f(x)$, where $x$ is randomly chosen, it is difficult to find $x'$ such that $f(x')=y$
- A one-way permutation is length-preserving (input and output have the same size) and one-to-one.
- Candidates for one-way functions
  - Multiplication: $f(x,y) = xy$
Relationship of One-Way Functions and Cryptography

• Secure encryption and MAC schemes imply/require the existence of one-way functions
• Given a one-way function, one can construct PRG, PRF, PRP
  – Thus one can construct secure encryption and MAC schemes
  – Details are more suitable for 655
• One-way functions are foundation of modern cryptography theory
Trapdoor One-way Functions

Definition:
A function $f: \{0,1\}^* \rightarrow \{0,1\}^*$ is a trapdoor one-way function iff $f(x)$ is a one-way function; however, given some extra information it becomes feasible to compute $f^{-1}$: given $y$, find $x$ s.t. $y = f(x)$
Public-Key Encryption Needs One-way Trapdoor Functions

• Given a public-key crypto system,
  – Alice has public key $K$
  – $E_K$ must be a one-way function, knowing $y = E_K[x]$, it should be difficult to find $x$
  – However, $E_K$ must not be one-way from Alice’s perspective. The function $E_K$ must have a trapdoor such that knowledge of the trapdoor enables one to invert it
RSA Algorithm

- Invented in 1978 by Ron Rivest, Adi Shamir and Leonard Adleman
- 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
Let \( p \) and \( q \) be two large primes.
Denote their product \( n=pq \).
\( Z_{pq}^* \) contains all integers in the range \([1,pq-1]\) that are relatively prime to both \( p \) and \( q \).
The size of \( Z_n^* \) is
\[
\Phi(pq) = (p-1)(q-1) = n-(p+q)+1
\]
For every \( x \in Z_{pq}^* \), \( x^{(p-1)(q-1)} \equiv 1 \)
Exponentiation in $\mathbb{Z}_{pq}^*$

- Motivation: We want to use exponentiation for encryption

- Let $e$ be an integer, $1 < e < (p-1)(q-1)$

- When is the function $f(x) = x^e$, a one-to-one function in $\mathbb{Z}_{pq}^*$?
- If $x^e$ is one-to-one, then it is a permutation in $\mathbb{Z}_{pq}^*$. 
Review: Euler’s Theorem

Euler’s Theorem
Given integer n > 1, such that gcd(a, n) = 1 then
\[ a^{\phi(n)} \equiv 1 \pmod{n} \]

Corollary: Given integer n > 1, such that gcd(a, n) = 1 then \( a^{\phi(n)-1} \mod n \) is a multiplicative inverse of a mod n.

Corollary: Given integer n > 1, x, y, and a positive integers with gcd(a, n) = 1. If \( x \equiv y \pmod{\phi(n)} \), then
\[ a^x \equiv a^y \pmod{n} \]

Corollary (Fermat’s “Little” Theorem): \( a^{p-1} \equiv 1 \pmod{p} \)
Exponentiation in $\mathbb{Z}_{pq}^*$

• Claim: If $e$ is relatively prime to $(p-1)(q-1)$ then $f(x) = x^e$ is a one-to-one function in $\mathbb{Z}_{pq}^*$

• Proof by constructing the inverse function of $f$. As $\gcd(e, (p-1)(q-1)) = 1$, then there exists $d$ and $k$ s.t. $ed = 1 + k(p-1)(q-1)$

• Let $y = x^e$, then $y^d = (x^e)^d = x^{1+k(p-1)(q-1)} = x \pmod{pq}$, i.e., $g(y) = y^d$ is the inverse of $f(x) = x^e$. 
RSA Public Key Crypto System

Key generation:
Select 2 large prime numbers of about the same size, p and q
Compute $n = pq$, and $\Phi(n) = (q-1)(p-1)$
Select a random integer $e$, $1 < e < \Phi(n)$, s.t. $\gcd(e, \Phi(n)) = 1$
Compute $d$, $1 < d < \Phi(n)$ s.t. $ed \equiv 1 \pmod{\Phi(n)}$

Public key: $(e, n)$
Private key: $d$
RSA Description (cont.)

Encryption
Given a message \( M, 0 < M < n \quad M \in \mathbb{Z}_n - \{0\} \)
use public key \((e, n)\)
compute \( C = M^e \mod n \quad C \in \mathbb{Z}_n - \{0\} \)

Decryption
Given a ciphertext \( C \), use private key \((d)\)
Compute \( C^d \mod n = (M^e \mod n)^d \mod n = M^{ed} \mod n = M \)
RSA Example

- $p = 11$, $q = 7$, $n = 77$, $\Phi(n) = 60$
- $d = 13$, $e = 37$ \,(ed = 481; \, ed \mod 60 = 1)

- Let $M = 15$. Then $C \equiv M^e \mod n$
  - $C \equiv 15^{37} \pmod{77} = 71$

- $M \equiv C^d \mod n$
  - $M \equiv 71^{13} \pmod{77} = 15$
Coming Attractions …

- RSA Security
- Prime number generation
- Reading: Katz & Lindell: 7.2