Department of
Computer Science and Engineering
计算机科学与工程系

CSE5014 CRYPTOGRAPHY AND NETWORK SECURITY

Dr. Qi Wang

Department of Computer Science and Engineering
Office: Room413, CoE South Tower
Email: wangqi@sustech.edu.cn

Private-key schemes

- We have seen how to construct schemes based on various lower-level primitives
  - Stream ciphers / PRGs
  - Block ciphers / PRFs
  - Hash functions
- How do we construct these primitives?

Hash functions

- Main goal is *collision resistance*
  - Want *optimal* birthday security

Hash functions

- Main goal is *collision resistance*
  - Want *optimal* birthday security

- Also want *preimage resistance*, *2nd-preimage resistance*
  - Want *optimal* security here as well

- "*Optimal*" measured relative to a random function
  - Why not design H to be a "random function"?

The random-oracle (RO) model

- Treat H as a public, *random* function
- Then $H(x)$ is *uniform* for any x
  - Unless the attacker computes $H(x)$ explicitly

The random-oracle (RO) model

- Treat H as a public, *random* function
- Then $H(x)$ is *uniform* for any x
  - Unless the attacker computes $H(x)$ explicitly
- Intuitively
  - Assume the hash function "*is random*"
  - Models attacks that are *agnostic* to the specific hash function being used
  - Security in the real world as long as "*no weaknesses found*" in the hash function

The random-oracle (RO) model

- Formally
  - Choose a *uniform* hash function as part of the security experiment
  - Attacker can *only* evaluate H via *explicit* queries to an oracle
  - Simulate H for the attacker as part of the security proof/reduction

The random-oracle (RO) model

- Formally
  - Choose a *uniform* hash function as part of the security experiment
  - Attacker can *only* evaluate H via *explicit* queries to an oracle
  - Simulate H for the attacker as part of the security proof/reduction
- In practice
  - Prove security in the RO model
  - Instantiate the RO with a "*good*" hash function
  - Hope for the best

Pros and cons of the RO model

- Cons
  - There is no such a thing as a public hash function that "*is random*"
    - Not even clear what this means formally
  - Known counterexamples
    - There are (contrived) schemes secure in the RO model, but insecure when using any real-world hash function
  - Sometimes *over-abused* (arguably)

Pros and cons of the RO model

- Pros
  – No known example of "*natural*" scheme secure in the RO model being attacked in the real world

  – If an attack is found, just replace the hash

  – Proof in the RO model better than no proof at all
    – Evidence that the basic design principles are sound

Groups

- A *group* is a set G and a binary operation $\circ$ defined on G such that:
  - (*Closure*) For all $g, h \in G$, $g \circ h$ is in G
  - (*Identity*) There is a unique element $e \in G$ such that $e \circ g = g$ for all $g \in G$
  - (*Inverse*) Every element $g \in G$ has an *inverse* $h \in G$ such that $g \circ h = e$
  - (*Associativity*) For all $f, g, h \in G$, $f \circ (g \circ h) = (f \circ g) \circ h$

- A *group* is a set G and a binary operation $\circ$ defined on G such that:
  - (*Closure*) For all $g, h \in G$, $g \circ h$ is in G
  - (*Identity*) There is a unique element $e \in G$ such that $e \circ g = g$ for all $g \in G$
  - (*Inverse*) Every element $g \in G$ has an *inverse* $h \in G$ such that $g \circ h = e$
  - (*Associativity*) For all $f, g, h \in G$, $f \circ (g \circ h) = (f \circ g) \circ h$
- A *group* is called *abelian* if the further property holds.
  - (*Commutativity*) For all $g, g \in G$, $g \circ h = h \circ g$

Groups

- A *group* is a set G and a binary operation $\circ$ defined on G such that:
  - (*Closure*) For all $g, h \in G$, $g \circ h$ is in G
  - (*Identity*) There is a unique element $e \in G$ such that $e \circ g = g$ for all $g \in G$
  - (*Inverse*) Every element $g \in G$ has an *inverse* $h \in G$ such that $g \circ h = e$
  - (*Associativity*) For all $f, g, h \in G$, $f \circ (g \circ h) = (f \circ g) \circ h$
- A *group* is called *abelian* if the further property holds.
  - (*Commutativity*) For all $g, g \in G$, $g \circ h = h \circ g$
- The *order* of a finite group G is $\#$ of elements in G.

Examples

- $\mathbb{Z}$ under addition
  $\mathbb{Z} \setminus \{0\}$ under multiplication
  $\mathbb{Q}$ under addition
  $\mathbb{Q} \setminus \{0\}$ under multiplication
  $\mathbb{R}$ under addition
  $\mathbb{R} \setminus \{0\}$ under multiplication
  $\{0, 1\}^*$ under concantenation
  $\{0, 1\}^n$ under bitwise XOR
  2×2 real matrices under addition
  2×2 invertible, real matrices under multiplication

Groups

- The group operation can be written *additively* or *multiplicatively*
  - I.e., instead of $g \circ h$, write $g + h$ or gh
  - Does *not* mean that the group operation corresponds to (integer) addition or multiplication

Groups

- The group operation can be written *additively* or *multiplicatively*
  - I.e., instead of $g \circ h$, write $g + h$ or gh
  - Does *not* mean that the group operation corresponds to (integer) addition or multiplication
- Identity denoted by 0 or 1, respectively
- Inverse of g denoted by $-g$ or g^{-1}, respectively
- Group *exponentiation*: $m \cdot a$ or a^m, respectively

Useful example

- $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ under addition modulo n
  - *Identity* is 0
  - *Inverse* of a is $-a$ mod N
  - *Associativity*, *commutativity* obvious
  - *Order* N

Useful example

- $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ under addition modulo n
  - *Identity* is 0
  - *Inverse* of a is $-a$ mod N
  - *Associativity*, *commutativity* obvious
  - *Order* N

- What happens if we consider *multiplication* modulo N?

Useful example

- $\mathbb{Z}_N = \{0, 1, \ldots, N-1\}$ under addition modulo n
  - *Identity* is 0
  - *Inverse* of a is $-a \bmod N$
  - *Associativity*, *commutativity* obvious
  - *Order* N

- What happens if we consider *multiplication* modulo N?

- $\mathbb{Z}_N$ is *not* a group under this operation!
  - 0 has no inverse
  - Even if we exclude 0, there is, e.g., *no* inverse of 2 modulo 4

Example

- What happens if we consider *multiplication* modulo N?
- Consider instead the *invertible* elements modulo N, under multiplication modulo N

Example

- What happens if we consider *multiplication* modulo N?
- Consider instead the *invertible* elements modulo N, under multiplication modulo N
- $\mathbb{Z}_N^* = \{0 < x < N : \gcd(x, N) = 1\}$
  - *Closure*
  - *Identity* is 1
  - *Inverse* of a is $a^{-1} \bmod N$
  - *Associativity*, *commutativity* obvious

Example

- What happens if we consider *multiplication* modulo N?
- Consider instead the *invertible* elements modulo N, under multiplication modulo N
- $\mathbb{Z}_N^* = \{0 < x < N : \gcd(x, N) = 1\}$
  - *Closure*
  - *Identity* is 1
  - *Inverse* of a is $a^{-1} \bmod N$
  - *Associativity*, *commutativity* obvious
- If p is prime, then $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$
  - $\mathbb{Z}_p$ is a (prime) *field*

Permutation Group

- Let $s_n =< 1, 2, \ldots, n >$ denote a *sequence* of integers 1 through n. Denote by P_n the set of all *permutations* of the sequence s_n.

Permutation Group

- Let $s_n = < 1, 2, \ldots, n >$ denote a *sequence* of integers 1 through n. Denote by P_n the set of all *permutations* of the sequence s_n.

  For example, $s_3 = < 1, 2, 3 >$
  $P_3 = \{< 1, 2, 3 >, < 1, 3, 2 >, < 2, 1, 3 >, < 2, 3, 1 >, < 3, 1, 2 >, < 3, 2, 1 >\}$

Permutation Group

- Let $s_n =< 1, 2, \ldots, n >$ denote a *sequence* of integers 1 through n. Denote by P_n the set of all *permutations* of the sequence s_n.

  For example, $s_3 =< 1, 2, 3 >$
  $P_3 = \{< 1, 2, 3 >, < 1, 3, 2 >, < 2, 1, 3 >, < 2, 3, 1 >, < 3, 1, 2 >, < 3, 2, 1 >\}$

- Define a binary operation $\circ$ on the elements of P_n: for $\rho, \pi \in P_n$, $\pi \circ \rho$ denotes a *re-permutation* of the elements of ρ according to the elements of π.

Permutation Group

- Consider $s_3 =< 1, 2, 3 >$, and $P_3 = \{< p_1, p_2, p_3 > | p_1, p_2, p_3 \in s_3 \text{ with } p_1 \neq p_2 \neq p_3\}$.

Permutation Group

- Consider $s_3 =< 1, 2, 3 >$, and $P_3 = \{< p_1, p_2, p_3 > | p_1, p_2, p_3 \in s_3$ with $p_1 \neq p_2 \neq p_3\}$.
- $\pi =< 3, 2, 1 >$, $\rho =< 1, 3, 2 >$, what is $\pi \circ \rho$?

Permutation Group

- Consider $s_3 =< 1, 2, 3 >$, and $P_3 = \{< p_1, p_2, p_3 > | p_1, p_2, p_3 \in s_3$ with $p_1 \neq p_2 \neq p_3\}$.
- $\pi =< 3, 2, 1 >$, $\rho =< 1, 3, 2 >$, what is $\pi \circ \rho$?

  $\pi \circ \rho =< 2, 3, 1 > \in P_3$

Permutation Group

- Consider $s_3 =< 1, 2, 3 >$, and
  $P_3 = \{< p_1, p_2, p_3 > | p_1, p_2, p_3 \in s_3$ with $p_1 \neq p_2 \neq p_3\}$.
- $\pi =< 3, 2, 1 >$, $\rho =< 1, 3, 2 >$, what is $\pi \circ \rho$?

  $\pi \circ \rho =< 2, 3, 1 >\in P_3$
- We can verify the other three properties.

  $\rho_1 \circ (\rho_2 \circ \rho_3) = (\rho_1 \circ \rho_2) \circ \rho_3$

  $< 1, 2, 3 > \circ \rho = \rho \circ < 1, 2, 3 >= \rho$

  For each $\rho \in P_3$, there exists another unique $\pi \in P_3$ such that

  $\rho \circ \pi = \pi \circ \rho =< 1, 2, 3 >$

- Consider $s_3 =< 1, 2, 3 >$, and
  $P_3 = \{< p_1, p_2, p_3 > | p_1, p_2, p_3 \in s_3 \text{ with } p_1 \neq p_2 \neq p_3\}$.
- $\pi =< 3, 2, 1 >$, $\rho =< 1, 3, 2 >$, what is $\pi \circ \rho$?

  $\pi \circ \rho =< 2, 3, 1 >\in P_3$
- We can verify the other three properties.

  $$\rho_1 \circ (\rho_2 \circ \rho_3) = (\rho_1 \circ \rho_2) \circ \rho_3$$

  $$< 1, 2, 3 > \circ \rho = \rho \circ < 1, 2, 3 >= \rho$$

  For each $\rho \in P_3$, there exists another unique $\pi \in P_3$ such that

  $$\rho \circ \pi = \pi \circ \rho =< 1, 2, 3 >$$

$(P_n, \circ)$ is called a *permutation group*.

Permutation Group

- Consider $s_3 = < 1, 2, 3 >$, and $P_3 = \{< p_1, p_2, p_3 > | p_1, p_2, p_3 \in s_3 \text{ with } p_1 \neq p_2 \neq p_3\}$.
- $\pi = < 3, 2, 1 >$, $\rho = < 1, 3, 2 >$, what is $\pi \circ \rho$?

  $\pi \circ \rho = < 2, 3, 1 > \in P_3$
- We can verify the other three properties.

  $\rho_1 \circ (\rho_2 \circ \rho_3) = (\rho_1 \circ \rho_2) \circ \rho_3$

  $< 1, 2, 3 > \circ \rho = \rho \circ < 1, 2, 3 > = \rho$

  For each $\rho \in P_3$, there exists another unique $\pi \in P_3$ such that

  $\rho \circ \pi = \pi \circ \rho = < 1, 2, 3 >$

$(P_n, \circ)$ is called a *permutation group*.
$(P_n, \circ)$ is not abelian.

Ring

- If $(R, +)$ is an *abelian group*, we define one more operation (denoted as *multiplication* $\times$ for convenience) to have a *ring* $(R, +, \times)$ satisfying the following properties.

Ring

- If $(R, +)$ is an *abelian group*, we define one more operation (denoted as *multiplication* $\times$ for convenience) to have a *ring* $(R, +, \times)$ satisfying the following properties.

  Closure: R must be closed w.r.t. $\times$

  Associativity: $(a \times b) \times c = a \times (b \times c)$

  Distributivity: $a \times (b + c) = a \times b + a \times c$
  $(a + b) \times c = a \times c + b \times c$

Ring

- If $(R, +)$ is an *abelian group*, we define one more operation (denoted as *multiplication* $\times$ for convenience) to have a *ring* $(R, +, \times)$ satisfying the following properties.

  Closure: R must be closed w.r.t. $\times$

  Associativity: $(a \times b) \times c = a \times (b \times c)$

  Distributivity: $a \times (b + c) = a \times b + a \times c$
  $(a + b) \times c = a \times c + b \times c$

  Example:
  $(\mathbb{Z}, +, \times)$, $(\mathbb{Q}, +, \times)$, $(\mathbb{R}, +, \times)$, $(\mathbb{M}_{n\times n}, +, \cdot)$?

Commutative Ring, Integral Domain

- A *ring* is *commutative* if the multiplication operation is *commutative* for all elements in the ring. $(ab = ba)$

Commutative Ring, Integral Domain

- A *ring* is *commutative* if the multiplication operation is *commutative* for all elements in the ring. ($ab = ba$)

- An *integral domain* $(R, +, \times)$ is a *commutative ring* that satisfies the following two additional properties.

  Identity element for multiplication: $a1 = 1a = a$
  Nonzero product for any two nonzero elements:
  if $ab = 0$, then either a or b must be 0.

Commutative Ring, Integral Domain

- A *ring* is *commutative* if the multiplication operation is *commutative* for all elements in the ring. $(ab = ba)$

- An *integral domain* $(R, +, \times)$ is a *commutative ring* that satisfies the following two additional properties.

  Identity element for multiplication: $a1 = 1a = a$
  Nonzero product for any two nonzero elements:
  if $ab = 0$, then either a or b must be 0.

  Example:
  $(\mathbb{Z}, +, \times)$, $(\mathbb{Q}, +, \times)$, $(\mathbb{R}, +, \times)$?
  $(\mathbb{Z}_m, +, \times)$, $(\mathbb{M}_{n\times n}, +, \cdot)$?

Field

- A *field*, denoted by $(F, +, \times)$, is an *integral domain* whose elements satisfy the following additional property.

  Inverse for multiplication: For every $a \in F$, there exists an element b, denoted by a^{-1}, such that $ab = ba = 1$.

Field

- A *field*, denoted by $(F, +, \times)$, is an *integral domain* whose elements satisfy the following additional property.

  Inverse for multiplication: For every $a \in F$, there exists an element b, denoted by a^{-1}, such that $ab = ba = 1$.

- **Example**:
  $(\mathbb{Z}, +, \times)$, $(\mathbb{Q}, +, \times)$, $(\mathbb{R}, +, \times)$?
  $(\mathbb{Z}_p, +, \times)$?

Field

- A *field*, denoted by $(F, +, \times)$, is an *integral domain* whose elements satisfy the following additional property.

  Inverse for multiplication: For every $a \in F$, there exists an element b, denoted by a^{-1}, such that $ab = ba = 1$.

- **Example**:
  $(\mathbb{Z}, +, \times)$, $(\mathbb{Q}, +, \times)$, $(\mathbb{R}, +, \times)$?
  $(\mathbb{Z}_p, +, \times)$?
- If $\mathbb{F}$ is finite, $\mathbb{F}$ is called a *finite field*.

Field

- A *field*, denoted by $(F, +, \times)$, is an *integral domain* whose elements satisfy the following additional property.

  Inverse for multiplication: For every $a \in F$, there exists an element b, denoted by a^{-1}, such that $ab = ba = 1$.

- **Example**:
  $(\mathbb{Z}, +, \times)$, $(\mathbb{Q}, +, \times)$, $(\mathbb{R}, +, \times)$?
  $(\mathbb{Z}_p, +, \times)$?

- If $\mathbb{F}$ is finite, $\mathbb{F}$ is called a *finite field*.

- $\mathbb{F}_q = \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ with the operations *addition*, *multiplication* of integers modulo p, is called a *prime field*
  - The properties can be verified

Prime subfield and characteristic

- Consider a *finite field* $\mathbb{F}$, define
  $S_r = 1 + 1 + \cdots + 1$ as sum of r 1's for a positive integer r
  - Let p be the smallest positive number with $S_p = 0$.
    If such a p exists, it must be *prime*

Prime subfield and characteristic

- Consider a *finite field* $\mathbb{F}$, define $S_r = 1 + 1 + \cdots + 1$ as sum of r 1's for a positive integer r
  - Let p be the smallest positive number with $S_p = 0$. If such a p exists, it must be *prime*
  - If $p = a \cdot b$ with $0 < a, b < p$, then by *distributivity*, $0 = S_p = S_a \cdot S_b$. Then one of S_a, S_b must be 0, contradicting the minimality of p.

Prime subfield and characteristic

- Consider a *finite field* $\mathbb{F}$, define
  $S_r = 1 + 1 + \cdots + 1$ as sum of r 1's for a positive integer r
  - Let p be the smallest positive number with $S_p = 0$.
    If such a p exists, it must be *prime*
  - If $p = a \cdot b$ with $0 < a, b < p$, then by *distributivity*,
    $0 = S_p = S_a \cdot S_b$. Then one of S_a, S_b must be 0,
    contradicting the minimality of p.
- This p is called the *characteristic* of the field $\mathbb{F}$.

Prime subfield and characteristic

- Consider a *finite field* $\mathbb{F}$, define $S_r = 1 + 1 + \cdots + 1$ as sum of r 1's for a positive integer r
  - Let p be the smallest positive number with $S_p = 0$. If such a p exists, it must be *prime*
  - If $p = a \cdot b$ with $0 < a, b < p$, then by *distributivity*, $0 = S_p = S_a \cdot S_b$. Then one of S_a, S_b must be 0, contradicting the minimality of p.
- This p is called the *characteristic* of the field $\mathbb{F}$.
- The subset $\{0, S_1, S_2, \ldots, S_{p-1}\} \subseteq \mathbb{F}$ is *isomorphic* to $\mathbb{F}$ (prime field)

Prime subfield and characteristic

- Consider a *finite field* $\mathbb{F}$, define $S_r = 1 + 1 + \cdots + 1$ as sum of r 1's for a positive integer r
  - Let p be the smallest positive number with $S_p = 0$. If such a p exists, it must be *prime*
  - If $p = a \cdot b$ with $0 < a, b < p$, then by *distributivity*, $0 = S_p = S_a \cdot S_b$. Then one of S_a, S_b must be 0, contradicting the minimality of p.
- This p is called the *characteristic* of the field $\mathbb{F}$.
- The subset $\{0, S_1, S_2, \ldots, S_{p-1}\} \subseteq \mathbb{F}$ is *isomorphic* to $\mathbb{F}$ (prime field)
- Any finite field $\mathbb{F}$ is a *finite dimensional vector space* over $\mathbb{F}_p$, with $n = \dim_{\mathbb{F}_p}(\mathbb{F})$, $|\mathbb{F}| = p^n$, i.e., the cardinality of $\mathbb{F}$ must be a prime power

Finite fields

- For any prime power q, there is essentially only one finite field of order q. Any two finite fields of order q are the same except that the labelling used to represent the field elements may be different

Finite fields

- For any prime power q, there is essentially only one finite field of order q. Any two finite fields of order q are the same except that the labelling used to represent the field elements may be different

- *Binary field* – *characteristic-2* finite fields $\mathbb{F}_{2^m}$
  - Elements are polynomials over $\mathbb{F}_2$ of degree $\leq m-1$
  - $\mathbb{F}_{2^m} := \{a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2 + a_1x + a_0 : \ a_i \in \mathbb{F}_2\}$

Finite fields

- For any prime power q, there is essentially only one finite field of order q. Any two finite fields of order q are the same except that the labelling used to represent the field elements may be different

- *Binary field* – *characteristic-2* finite fields $\mathbb{F}_{2^m}$
  - Elements are polynomials over $\mathbb{F}_2$ of degree $\leq m-1$
  - $\mathbb{F}_{2^m} := \{a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2 + a_1x + a_0 : \ a_i \in \mathbb{F}_2\}$

- An *irreducible polynomial* $f(x)$ of degree m is chosen: $f(x)$ cannot be factered as a product of binary polynomials each of degree less than m
  - *Addition*: usual
  - *Multiplication*: modulo $f(x)$

Hard problems

- So far, we have only discussed number-theoretic problem that are *easy*
  - E.g., addition, multiplication, modular arithmetic, exponentiation

Hard problems

- So far, we have only discussed number-theoretic problem that are *easy*
  - E.g., addition, multiplication, modular arithmetic, exponentiation

- Some problems are (conjectured to be) *hard*

Factoring

- Multiplying two numbers is *easy*; factoring a number *hard*
  - Given x, y, *easy* to compute $x \cdot y$
  - Given N, *hard* (in general) to find $x, y > 1$ such that $x \cdot y = N$

Factoring

- Multiplying two numbers is *easy*; factoring a number *hard*
  - Given x, y, *easy* to compute $x \cdot y$
  - Given N, *hard* (in general) to find $x, y > 1$ such that $x \cdot y = N$
- Compare:
  - Multiply 10101023 and 29100257
  - Find the factors of 293942365262911

Factoring

- It is not hard to factor most numbers
  - 50% of the time, random number is even
  - 1/3 of the time, random number is divisible by 3

Factoring

- It is not hard to factor most numbers
  - 50% of the time, random number is even
  - 1/3 of the time, random number is divisible by 3
- The hardest numbers to factor are those that are the *product* of two, equal-length primes

Factoring

- It is not hard to factor most numbers
  - 50% of the time, random number is even
  - 1/3 of the time, random number is divisible by 3
- The hardest numbers to factor are those that are the *product* of two, equal-length primes
- The *RSA problem* is related to *factoring*

The RSA problem

- Let $N = pq$ with p and q distinct, odd primes
- $\mathbb{Z}_N^* =$ *invertiable* elements under multiplication modulo N
  - The order of $\mathbb{Z}_N^*$ is $\phi(N) = (p-1) \cdot (q-1)$
  - $\phi(N)$ is *easy* to compute if p, q are known
  - $\phi(N)$ is *hard* to compute if p, q are not known
    - Equivalent (believed) to factoring N

The RSA problem

- Let $N = pq$ with p and q distinct, odd primes
- $\mathbb{Z}_N^*$ = *invertiable* elements under multiplication modulo N
  - The order of $\mathbb{Z}_N^*$ is $\phi(N) = (p-1) \cdot (q-1)$
  - $\phi(N)$ is *easy* to compute if p, q are known
  - $\phi(N)$ is *hard* to compute if p, q are not known
    - Equivalent (believed) to factoring N
- Fix e with $\gcd(e, \phi(N)) = 1$
  - Raising to the e-th power is a permutation of $\mathbb{Z}_N^*$

The RSA problem

- Let $N = pq$ with p and q distinct, odd primes
- $\mathbb{Z}_N^* =$ *invertiable* elements under multiplication modulo N
  - The order of $\mathbb{Z}_N^*$ is $\phi(N) = (p-1)\cdot(q-1)$
  - $\phi(N)$ is *easy* to compute if p, q are known
  - $\phi(N)$ is *hard* to compute if p, q are not known
    - Equivalent (believed) to factoring N
- Fix e with $\gcd(e, \phi(N)) = 1$
  - Raising to the e-th power is a permutation of $\mathbb{Z}_N^*$
- If $ed \equiv 1 \bmod \phi(N)$, raising to the d-th power is the *inverse* of raising to the e-th power
  - I.e., $(x^e)^d \equiv x \bmod N$
  - x^d is the e-th root of x modulo N

The RSA problem

- If p, q are known:
  $\Rightarrow \phi(N)$ can be computed
  $\Rightarrow d = e^{-1} \bmod \phi(N)$ can be computed
  $\Rightarrow$ possible to compute e-th roots modulo N

The RSA problem

- If p, q are known:
  $\Rightarrow \phi(N)$ can be computed
  $\Rightarrow d = e^{-1} \bmod \phi(N)$ can be computed
  $\Rightarrow$ possible to compute e-th roots modulo N

- If p, q are *not* known:
  $\Rightarrow$ computing $\phi(N)$ is as hard as factoring N
  $\Rightarrow$ computing d is as hard as factoring N

The RSA problem

- If p, q are known:
  $\Rightarrow \phi(N)$ can be computed
  $\Rightarrow d = e^{-1} \bmod \phi(N)$ can be computed
  $\Rightarrow$ possible to compute e-th roots modulo N

- If p, q are *not* known:
  $\Rightarrow$ computing $\phi(N)$ is as hard as factoring N
  $\Rightarrow$ computing d is as hard as factoring N

  $\mathcal{Q}$: Given d and e, can we factor N?

The RSA problem

- If p, q are known:
  $\Rightarrow \phi(N)$ can be computed
  $\Rightarrow d = e^{-1} \bmod \phi(N)$ can be computed
  $\Rightarrow$ possible to compute e-th roots modulo N

- If p, q are *not* known:
  $\Rightarrow$ computing $\phi(N)$ is as hard as factoring N
  $\Rightarrow$ computing d is as hard as factoring N
  $\mathcal{Q}$: Given d and e, can we factor N?

- Very useful for *public-key* cryptography

The RSA problem

- Informally: given N, e, and uniform element $y \in \mathbb{Z}_N^*$, compute the e-th root of y

The RSA problem

- Informally: given N, e, and uniform element $y \in \mathbb{Z}_N^*$, compute the e-th root of y
- *RSA assumption*: this is a hard problem!

The RSA problem

- Informally: given N, e, and uniform element $y \in \mathbb{Z}_N^*$, compute the e-th root of y
- *RSA assumption*: this is a hard problem!
- Formally:
- GenRSA: on input 1^n, outputs (N, e, d) with $N = pq$ a product of two distinct n-bit primes, with $ed = 1 \bmod \phi(N)$

The RSA problem

- Informally: given N, e, and uniform element $y \in \mathbb{Z}_N^*$, compute the e-th root of y
- *RSA assumption*: this is a hard problem!
- Formally:
- GenRSA: on input 1^n, outputs (N, e, d) with $N = pq$ a product of two distinct n-bit primes, with $ed = 1 \bmod \phi(N)$
- Experiment RSA-inv$_{A,GenRSA}(n)$:
  - Compute $(N, e, d) \leftarrow GenRSA(1^n)$
  - Choose uniform $y \in \mathbb{Z}_N^*$
  - Run $A(N, e, y)$ to get x
  - Experiment evaluates to 1 if $x^e = y \bmod N$

The RSA assumption (formal)

- GenRSA: on input 1^n, outputs (N, e, d) with $N = pq$ a product of two distinct n-bit primes, with $ed = 1 \bmod \phi(N)$
- Experiment RSA-inv$_{A,GenRSA}(n)$:
  – Compute $(N, e, d) \leftarrow GenRSA(1^n)$
  – Choose uniform $y \in \mathbb{Z}_N^*$
  – Run $A(N, e, y)$ to get x
  – Experiment evaluates to 1 if $x^e = y \bmod N$

The RSA assumption (formal)

- GenRSA: on input 1^n, outputs (N, e, d) with $N = pq$ a product of two distinct n-bit primes, with $ed = 1 \bmod \phi(N)$
- Experiment RSA-inv$_{A,GenRSA}(n)$:
  - Compute $(N, e, d) \leftarrow GenRSA(1^n)$
  - Choose uniform $y \in \mathbb{Z}_N^*$
  - Run $A(N, e, y)$ to get x
  - Experiment evaluates to 1 if $x^e = y \bmod N$

- The *RSA problem* is *hard* relative to GenRSA if for all PPT algorithms A,

$$\Pr[\text{RSA-inv}_{A,GenRSA}(n) = 1] < negl(n)$$

Implementing GenRSA

- One way to implement GenRSA:
  - Generate uniform n-bit primes p, q
  - Set $N := pq$
  - Choose arbitrary e with $\gcd(e, \phi(N)) = 1$
  - Compute $d := e^{-1} \bmod \phi(N)$
  - Output (N, e, d)

Implementing GenRSA

- One way to implement GenRSA:
  - Generate uniform n-bit primes p, q
  - Set $N := pq$
  - Choose arbitrary e with $\gcd(e, \phi(N)) = 1$
  - Compute $d := e^{-1} \bmod \phi(N)$
  - Output (N, e, d)

- Choice of e?
  - Does *not* seem to affect hardness of the *RSA problem*
  - $e = 3$ or $e = 2^{16} + 1$ for *efficient* exponentiation

RSA and factoring

- If factoring moduli output by GenRSA is easy, then the *RSA problem* is easy relative to GenRSA
  - Factoring is easy $\Rightarrow$ RSA problem is easy

RSA and factoring

- If factoring moduli output by GenRSA is easy, then the *RSA problem* is easy relative to GenRSA
  - Factoring is easy $\Rightarrow$ RSA problem is easy

- Hardness of the *RSA problem* is not known to be implied by hardness of factoring
  - Possible factoring is hard but *RSA problem* is easy
  - Possible both are hard but *RSA problem* is "easier"
  - Currently, RSA is believed to be *as hard as factoring*

Trapdoor functions

- **Definition 10.1** (*Trapdoor functions*) A *trapdoor function collection* is a collection $\mathcal{F}$ of finite functions such that every $f \in \mathcal{F}$ is a one-to-one function from some set S_f to a set T_f. The following properties are requried.
  - Efficient generation, computation and inversion
  There is a PPT algorithm G that on input 1^n outputs a pair (f, f^{-1}), where these are two $poly(n)$ size strings that describe the functions f, f^{-1}
  - Efficient sampling There is a PPT algorithm that given f can output a random element of S_f
  - One-wayness The function f is hard to invert without knowing the *invertion key*. For all PPT A there is a negligible function ϵ s.t.

$$\Pr_{(f,f^{-1})\leftarrow_R G(1^n),\ x\leftarrow_R S_f}[A(1^n, f, f(x)) = x] < \epsilon(n)$$

RSA trapdoor function

- **Keys**: choose P, Q as random primes of length n, $N = P \cdot Q$. Choose
  e at random from $\{1, \ldots, \phi(N) - 1\}$ with $\gcd(e, \phi(N)) = 1$
  Forward **Key**: N, e
  Backward **Key**: d with $ed \equiv 1 \bmod \phi(N)$
  Function: $RSA_{N,e}(X) = X^e \pmod{N}$
  Inverse: If $Y = RSA_{N,e}(X) = X^e \bmod N$, then $Y^d \bmod N = X$.

RSA trapdoor function

- **Keys**: choose P, Q as random primes of length n, $N = P \cdot Q$. Choose e at random from $\{1, \ldots, \phi(N) - 1\}$ with $\gcd(e, \phi(N)) = 1$
  Forward **Key**: N, e
  Backward **Key**: d with $ed \equiv 1 \bmod \phi(N)$
  Function: $RSA_{N,e}(X) = X^e \pmod N$
  Inverse: If $Y = RSA_{N,e}(X) = X^e \bmod N$, then $Y^d \bmod N = X$.

- **RSA Assumption**: the RSA function is indeed a *trapdoor function*
  – This is stronger than the assumption that factoring is hard

Rabin's trapdoor function

- Assume that *factoring* random *Blum integers* is hard. A *Blum integer* is a number $n = pq$ where $p, q \equiv 3 \pmod 4$.

Rabin's trapdoor function

- Assume that *factoring* random *Blum integers* is hard. A *Blum integer* is a number $n = pq$ where $p, q \equiv 3 \pmod 4$.

- Define $\mathcal{B}_n := \{P \in [1 \ldots 2^n] : P \text{ prime and } P \equiv 3 \bmod 4\}$

  The Factoring Axiom For every PPT algorithm A there is a negligible function ϵ s.t.

$$\Pr_{P,Q \leftarrow_R \mathcal{B}_n}[A(P \cdot Q) = \{P, Q\}] < \epsilon(n)$$

Rabin's trapdoor function

- **Keys**: choose P, Q as random primes of length n with
  $P, Q \equiv 3 \bmod 4$, $N = P \cdot Q$.
  Forward **Key**: N
  Backward **Key**: P, Q
  Function: $Y = RABIN_N(X) = X^2 \bmod N$, which is a permutation on QR_N, where QR_N denotes the set of quadratic residues modulo N

Rabin's trapdoor function

- **Keys**: choose P, Q as random primes of length n with
  $P, Q \equiv 3 \bmod 4$, $N = P \cdot Q$.
  Forward **Key**: N
  Backward **Key**: P, Q
  Function: $Y = RABIN_N(X) = X^2 \bmod N$, which is a permutation on QR_N, where QR_N denotes the set of quadratic residues modulo N

  Inverse: Compute $A = Y \bmod P$ and $B = Y \bmod Q$. Since $P, Q \equiv 3 \bmod 4$, let $P = 4t + 3$ and $Q = 4t' + 3$.

Rabin's trapdoor function

- **Keys**: choose P, Q as random primes of length n with
  $P, Q \equiv 3 \bmod 4,\ N = P \cdot Q$.
  Forward **Key**: N
  Backward **Key**: P, Q
  Function: $Y = RABIN_N(X) = X^2 \bmod N$, which is a permutation on QR_N, where QR_N denotes the set of quadratic residues modulo N

  Inverse: Compute $A = Y \bmod P$ and $B = Y \bmod Q$. Since $P, Q \equiv 3 \bmod 4$, let $P = 4t + 3$ and $Q = 4t' + 3$.

  Compute $X_1 = A^{t+1} \bmod P$ and $X_2 = B^{t'+1} \bmod Q$. Using CRT, we find X.

Rabin's trapdoor function

- **Keys**: choose P, Q as random primes of length n with
  $P, Q \equiv 3 \bmod 4,\ N = P \cdot Q$.
  Forward **Key**: N
  Backward **Key**: P, Q
  Function: $Y = RABIN_N(X) = X^2 \bmod N$, which is a permutation on QR_N, where QR_N denotes the set of quadratic residues modulo N

  Inverse: Compute $A = Y \bmod P$ and $B = Y \bmod Q$. Since $P, Q \equiv 3 \bmod 4$, let $P = 4t + 3$ and $Q = 4t' + 3$.

  Compute $X_1 = A^{t+1} \bmod P$ and $X_2 = B^{t'+1} \bmod Q$. Using CRT, we find X.

  We know that $X = S^2 \bmod P$, then

  $$X_1 = (X^2)^{t+1} = S^{4(t+1)} = S^{P-1+2} = S^2 = X \bmod P.$$

  Similarly, $X_2 = S^2 = X \bmod Q$.

Rabin's trapdoor function

- **Lemma 10.2** Let X, Y be such that $X \not\equiv \pm Y \pmod{N}$ but $X^2 \equiv Y^2 \pmod{N}$. Then $\gcd(X - Y, N) \notin \{1, N\}$.

  Proof. easy.

Rabin's trapdoor function

- **Lemma 10.2** Let X, Y be such that $X \not\equiv \pm Y \pmod{N}$ but $X^2 \equiv Y^2 \pmod{N}$. Then $\gcd(X - Y, N) \notin \{1, N\}$.

  Proof. easy.

  Theorem 10.3 (*One-wayness of Rabin's function*)
  Rabin's function is a *trapdoor function* under the factoring axiom.

Rabin's trapdoor function

- **Lemma 10.2** Let X, Y be such that $X \not\equiv \pm Y \pmod{N}$ but $X^2 \equiv Y^2 \pmod{N}$. Then $\gcd(X - Y, N) \notin \{1, N\}$.

  Proof. easy.

  Theorem 10.3 (*One-wayness of Rabin's function*)
  Rabin's function is a *trapdoor function* under the factoring axiom.

  Proof. By contradiction. (see blackboard)

Next Lecture

- public key encryption ...