COS433/Math 473: Cryptography

Mark Zhandry
Princeton University
Spring 2020
Announcements

HW5 Due Today
PR2 Due April 19th
Previously on COS 433...
Integer Factorization
Integer Factorization

Given an integer $N$, find it’s prime factors

Studied for centuries, presumed difficult

• Grade school algorithm: $O(N^{1/2})$
• Better algorithms using birthday paradox: $O(N^{1/4})$
• Even better assuming G. Riemann Hyp.: $O(N^{1/6})$
• Still better heuristic algorithms:
  \[ \exp\left( C (\log N)^{1/3} (\log \log N)^{2/3} \right) \]
• However, all require super-polynomial time in bit-length of $N$
Factoring Assumption: For any factoring algorithm running in polynomial time, \( \exists \) negligible \( \varepsilon \) such that:

\[
\Pr[((p, q) \leftarrow \mathcal{R}(N): \\
N = pq \\
p, q \leftarrow \text{random } \lambda\text{-bit primes}] \leq \varepsilon(\lambda)
\]
Chinese Remainder Theorem

Let $N = pq$ for distinct prime $p,q$

Let $x \in \mathbb{Z}_p$, $y \in \mathbb{Z}_q$

Then there exists a unique integer $z \in \mathbb{Z}_N$ such that

- $x = z \mod p$, and
- $y = z \mod q$

Proof: $z = [py(p^{-1} \mod q)+qx(q^{-1} \mod p)] \mod N$
Quadratic Residues

**Definition:** $y$ is a quadratic residue mod $N$ if there exists an $x$ such that $y = x^2 \mod N$. $x$ is called a “square root” of $y$

Ex:

- Let $p$ be a prime, and $y \neq 0$ a quadratic residue mod $p$. How many square roots of $y$?
- Let $N=pq$ be the product of two primes, $y$ a quadratic residue mod $N$. Suppose $y \neq 0 \mod p$ and $y \neq 0 \mod q$. How many square roots?
QR Assumption: For any algorithm running in polynomial time, \( \exists \) negligible \( \varepsilon \) such that:

\[
\Pr[y^2 = x^2 \mod N: \quad \begin{align*}
y &\leftarrow \mathcal{U}(N, x^2) \\
N = pq, &\quad p, q \leftarrow \text{random } \lambda\text{-bit primes} \\
x &\leftarrow \mathbb{Z}_N
\end{align*}
\] \leq \varepsilon(\lambda)
This Time

Factoring continued

Public key cryptography
Theorem: If the factoring assumption holds, then the QR assumption holds.
Proof

To factor $N$:
• $x \leftarrow \mathbb{Z}_N$
• $y \leftarrow \mathbb{R}(N,x^2)$
• Output $\text{GCD}(x-y,N)$

Analysis:
• Let $\{a,b,c,d\}$ be the 4 square roots of $x^2$
• has no idea which one you chose
• With probability $\frac{1}{2}$, $y$ will not be in $\{+x,-x\}$
• In this case, we know $x = y \mod p$ but $x = -y \mod q$
Collision Resistance from Factoring

Let $N=pq$, $y$ a QR mod $N$
Suppose $-1$ is not a QR mod $N$

Hashing key: $(N,y)$
Domain: $\{1,...,(N-1)/2\} \times \{0,1\}$
Range: $\{1,...,(N-1)/2\}$

$H((N,y), (x,b))$: Let $z = y^b x^2 \mod N$
• If $z \in \{1,...,(N-1)/2\}$, output $z$
• Else, output $-z \mod N \in \{1,...,(N-1)/2\}$
**Theorem:** If the factoring assumption holds, $H$ is collision resistant

**Proof:**

- Collision means $\langle x_0, b_0 \rangle \neq \langle x_1, b_1 \rangle$ s.t.
  
  $$y^{b_0} x_0^2 = \pm y^{b_1} x_1^2 \mod N$$

- If $b_0 = b_1$, then $x_0 \neq x_1$, but $x_0^2 = \pm x_1^2 \mod N$
  
  - $x_0^2 = -x_1^2 \mod N$ not possible. Why?
  
  - $x_0 \neq -x_1$ since $x_0, x_1 \in \{1, \ldots, (N-1)/2\}$

- If $b_0 \neq b_1$, then $(x_0/x_1)^2 = \pm y^{\pm 1} \mod N$
  
  - $-y$ case not possible. Why?
  
  - $(x_0/x_1)$ or $(x_1/x_0)$ is a square root of $y$
Choosing \( N \)

How to choose \( N \) so that \(-1\) is not a QR?

By CRT, need to choose \( p, q \) such that \(-1\) is not a QR mod \( p \) or mod \( q \)

Fact: if \( p = 3 \mod 4 \), then \(-1\) is not a QR mod \( p \)
Fact: if \( p = 1 \mod 4 \), then \(-1\) is a QR mod \( p \)
Is Composite $N$ Necessary for SQ to be hard?

Let $p$ be a prime, and suppose $p = 3 \mod 4$

Given a QR $x \mod p$, how to compute square root?

Hint: recall Fermat: $x^{p-1}=1 \mod p$ for all $x\neq 0$

Hint: what is $x^{(p+1)/2} \mod p$?
Solving Quadratic Equations

In general, solving quadratic equations is:

• Easy over prime moduli

• As hard as factoring over composite moduli
Other Powers?

What about $x \rightarrow x^4 \mod N$? $x \rightarrow x^6 \mod N$?

The function $x \rightarrow x^3 \mod N$ appears quite different

• Suppose 3 is relatively prime to $p-1$ and $q-1$

• Then $x \rightarrow x^3 \mod p$ is injective for $x \neq 0$
  - Let $a$ be such that $3a = 1 \mod p-1$
  - $(x^3)^a = x^{1+k(p-1)} = x(x^{p-1})^k = x \mod p$

• By CRT, $x \rightarrow x^3 \mod N$ is injective for $x \in \mathbb{Z}_N^*$
$$x^3 \mod N$$

What does injectivity mean?

Cannot base of factoring:
Adapt alg for square roots?

- Choose a random $z \mod N$
- Compute $y = z^3 \mod N$
- Run inverter on $y$ to get a cube root $x$
- Let $p = \text{GCD}(z-x, N)$, $q = N/p$
RSA Problem

Given

• \(N = pq\),
• \(e\) such that \(\gcd(e, p-1) = \gcd(e, q-1) = 1\),
• \(y = x^e \mod N\) for a random \(x\)

Find \(x\)

Injectivity means cannot base hardness on factoring, but still conjectured to be hard
RSA Assumption: For any algorithm running in polynomial time, \( \exists \) negligible \( \varepsilon \) such that:

\[
\Pr[x \leftarrow (N, x^3 \mod N) \\
N = pq \text{ and } p, q \text{ random } \lambda\text{-bit primes s.t.} \\
\gcd(3, p-1) = \gcd(3, q-1) = 1 \\
x \leftarrow \mathbb{Z}_N^* ] \leq \varepsilon(\lambda)
\]
Application: PRGs

Let $F(x) = x^3 \mod N$, $h(x) =$ least significant bit

Theorem: If RSA Assumption holds, then $G(x) = (F(x), h(x))$ is a secure PRG
Public Key Cryptography
How do Alice & Bob get k?
Limitations

Time consuming

Not realistic in many situations
• Do you really want to send a courier to every website you want to communicate with

Doesn’t scale well
• Imagine 1M people communicating with 1M people

If not meeting in person, need to trust courier
Public Key Distribution
Public Key Distribution
Public Key Distribution
Public Key Distribution
Interactive Protocols

Pair of interactive (randomized) algorithms $A$, $B$

Write $(\text{Trans}, o_A, o_B) \leftarrow (A, B)(x, y)$
Public Key Distribution

Pair of interactive algorithms $A, B$

Correctness:
\[
\Pr[o_A = o_B: (\text{Trans}, o_A, o_B) \leftarrow (A, B)()] = 1
\]

Shared key is $k := o_A = o_B$

- Define $(\text{Trans}, k) \leftarrow (A, B)()$

Security: $(\text{Trans}, k)$ is computationally indistinguishable from $(\text{Trans}, k')$ where $k' \leftarrow K$ independent of $k$
Matrix Multiplication Approach

\[ B \leftarrow \mathbb{Z}_q^{\lambda \times \lambda} \]

\[ A \leftarrow B^{-1} \]
Matrix Multiplication Approach

\[ B \leftarrow \mathbb{Z}_q^{\lambda \times \lambda} \]
\[ A \leftarrow B^{-1} \]
\[ v \leftarrow \mathbb{Z}_q^{\lambda} \]
\[ w \leftarrow B \cdot v \]
Matrix Multiplication Approach

\[ B \leftarrow \mathbb{Z}_q^{\lambda \times \lambda} \]
\[ A \leftarrow B^{-1} \]
\[ v = A \cdot w \]

\[ v \leftarrow \mathbb{Z}_q^{\lambda} \]
\[ w \leftarrow B \cdot v \]
Matrix Multiplication Approach

\[
\begin{align*}
  B & \leftarrow \mathbb{Z}_q^{\lambda \times \lambda} \\
  A & \leftarrow B^{-1} \\
  v & = A \cdot w
\end{align*}
\]
Running Times?

Bob: $O(\lambda^2)$
Eve: $O(\lambda^3)$
Running Times?

Bob: $O(\lambda^2)$
Eve: $O(\lambda^\omega)$ where $\omega \leq 2.373$
Alice: $O(\lambda^\omega)$

Different Approach:
• Start with $A = B = I$
• Repeatedly apply random elementary row ops to $A$,
  inverse to $B$
• Output $(A, B)$
Running Times?

Bob: $O(\lambda^2)$
Eve: $O(\lambda^\omega)$ where $\omega \leq 2.373$
Alice: $O(\lambda^\omega)$

**Different Approach:**
- Start with $A = B = I$
- Repeatedly apply random elementary row ops to $A$, inverse to $B$
- Output $(A,B)$

Assuming Matrix Multiplication exponent $\omega > 2$, adversary must work harder than honest users.
Merkle Puzzles

Let $H$ be some hash function with domain $[\lambda] = \{1, \ldots, \lambda\}$

$a_1, \ldots, a_t \leftarrow [\lambda]
\quad A_i \leftarrow H(a_i)$

$b_1, \ldots, b_t \leftarrow [\lambda]
\quad b_i \leftarrow H(b_i)$
Merkle Puzzles

Let $H$ be some hash function with domain $[\lambda] = \{1, \ldots, \lambda\}$

$a_1, \ldots, a_t \leftarrow [\lambda]$

$A_i \leftarrow H(a_i)$

$a_i$ s.t. $A_i \in \{B_i\}$

$b_1, \ldots, b_t \leftarrow [\lambda]$

$b_i \leftarrow H(b_i)$

$b_i$ s.t. $B_i \in \{A_i\}$
Analysis

Protocol succeeds iff:
• \( H \) is injective (why?)
• \( \{A_i\} \cap \{B_i\} \neq \emptyset \) (equiv, \( \{a_i\} \cap \{b_i\} \neq \emptyset \))

What does \( t \) need to be to make \( \{A_i\} \cap \{B_i\} \neq \emptyset \)?

If adversary can only query \( H \) on various inputs, how many queries needed?
Limitations

Both matrix multiplication and Merkle puzzle approaches have a polynomial gap between honest users and adversaries.

To make impossible for extremely powerful adversaries, need at least $\lambda^2 > 2^{80}$

- Special-purpose hardware means $\lambda$ needs to be even bigger
- Honest users require time at least $\lambda=2^{40}$
- Possible, but expensive
Limitations

Instead, want a super-polynomial gap between honest users and adversary
• Just like everything else we’ve seen in the course
Key Distribution from Obfuscation

Software obfuscation:
• Compile programs into unreadable form (intentionally)
Key Distribution from Obfuscation

Let $F, F^{-1}$ be a block cipher

$k \leftarrow \{0,1\}^\lambda$

$P \leftarrow \text{Obf}( F(k, \cdot) )$
Key Distribution from Obfuscation

Let $F, F^{-1}$ be a block cipher

$k \leftarrow \{0,1\}^\lambda$

$P \leftarrow \text{Obf}(F(k, \cdot))$

$r \leftarrow \{0,1\}^\lambda$

$x \leftarrow P(r)$
Let $F, F^{-1}$ be a block cipher.

Let $k \leftarrow \{0,1\}^\lambda$

$P \leftarrow \text{Obf}(F(k, \cdot))$

$r \leftarrow F^{-1}(k, x)$

$r \leftarrow \{0,1\}^\lambda$

$x \leftarrow P(r)$

$r$
Key Distribution From Obfuscation

For decades, many attempts at commercial code obfuscators

• Simple operations like variable renaming, removing whitespace, re-ordering operations

Really only a “speed bump” to determined adversaries

• Possible to recover something close to original program (including cryptographic keys)

Don’t use commercially available obfuscators to hide cryptographic keys!
Key Distribution From Obfuscation

Recently (2013), new type of obfuscator has been developed
• Much stronger security guarantees
• Based on mathematical tools
• Many cryptographic applications beyond public key distribution

Downside?
• Extraordinarily impractical (currently)
Practical Key Exchange

Instead of obfuscating a general PRP, we will define a specific abstraction that will enable key agreement

Then, we will show how to implement the abstraction using number theory
Trapdoor Permutations

Domain $X$

$\text{Gen}()$: outputs $(pk, sk)$

$F(pk, x \in X) = y \in X$

$F^{-1}(sk, y) = x$

Correctness:

$\Pr[ F^{-1}(sk, F(pk, x)) = x : (pk, sk) \leftarrow \text{Gen}() ] = 1$

Correctness implies $F, F^{-1}$ are deterministic, permutations
Trapdoor Permutation Security

In other words, $F(pk, \cdot)$ is a one-way function.
Key Distribution from TDPs

$$(pk, sk) \leftarrow \text{Gen()}$$

$$x \leftarrow F^{-1}(sk, y)$$

$$y \leftarrow F(pk, x)$$

$$x \leftarrow F(pk, x)$$
Analysis

Correctness follows from correctness of TDP

Security:

• By TDP security, adversary cannot compute ✗
• However, ✗ is distinguishable from a random key
Hardcore Bits

Let $F$ be a one-way function with domain $D$, range $R$

**Definition:** A function $h : D \rightarrow \{0,1\}$ is a hardcore bit for $F$ if, for any polynomial time $\mathcal{B}$, $\exists$ negligible $\varepsilon$ such that:

$$| \Pr[1 \leftarrow \mathcal{B}(F(x), h(x)), x \leftarrow D] - \Pr[1 \leftarrow \mathcal{B}(F(x), b), x \leftarrow D, b \leftarrow \{0,1\}] | \leq \varepsilon(\lambda)$$

In other words, even given $F(x)$, hard to guess $h(x)$
Examples of Hardcore Bits

Define $\text{lsb}(x)$ as the least significant bit of $x$

For $x \in \mathbb{Z}_N$, define $\text{Half}(x)$ as 1 iff $0 \leq x < N/2$
Theorem: Let \( p \) be a prime, and \( F: \mathbb{Z}_p^* \rightarrow \mathbb{Z}_p^* \) be \( F(g,x) = (g,g^x \mod p) \)

Half is a hardcore bit for \( F \) (assume \( F \) is one-way)

Theorem: Let \( N \) be a product of two large primes \( p,q \), and \( F: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^* \) be \( F(x) = x^e \mod N \) for some \( e \) relatively prime to \((p-1)(q-1)\)

Lsb and Half are hardcore bits for \( F \) (assuming RSA)

Theorem: Let \( N \) be a product of two large primes \( p,q \), and \( F: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^* \) be \( F(x) = x^2 \mod N \)

Lsb and Half are hardcore bits for \( F \) (assuming factoring)
Key Distribution from TDPs

$$(pk, sk) \leftarrow \text{Gen}()$$

$$x \leftarrow h(F^{-1}(sk, y))$$

$$y \leftarrow F(pk, x)$$

$$x \leftarrow x$$

$$h(a \text{ hardcore bit for } F(pk, \cdot ))$$
Theorem: If $h$ is a hardcore bit for $F(pk, \cdot)$, then protocol is secure.

Proof:

• $(\text{Trans},k) = ((pk,y), h(x))$
• Hardcore bit means indist. from $(pk,y), b)$
Trapdoor Permutations from RSA

**Gen()**:  
- Choose random primes \( p, q \)  
- Let \( N = pq \) 
- Choose \( e, d \) s.t \( ed = 1 \mod (p-1)(q-1) \)  
- Output \( pk = (N, e), sk = (N, d) \)

**F(pk, x)**: Output \( y = x^e \mod N \)

**F^{-1}(sk, c)**: Output \( x = y^d \mod N \)
Caveats

RSA is not a true TDP as defined
• Why???
• What’s the domain?

Nonetheless, distinction is not crucial to most applications
• In particular, works for key agreement protocol
Other TDPs?

For long time, essentially none known
• Still interesting object:
  • Useful abstraction in protocol design
  • Maybe more will be discovered...

Using obfuscation:
• Let $P$ be a PRP
• $sk = k$, $pk = Obf(P(k, \cdot ))$
Key Distribution from DH

Everyone agrees on group $G$ of prime order $p$.

$a \leftarrow \mathbb{Z}_p$

$b \leftarrow \mathbb{Z}_p$
Key Distribution from DH

Everyone agrees on group $G$ or prime order $p$
Key Distribution from DH

Everyone agrees on group $G$ or prime order $p$

$a \leftarrow \mathbb{Z}_p \quad g^a \quad g^b \quad b \leftarrow \mathbb{Z}_p$

$k = (g^b)^a = g^{ab} \quad k = (g^a)^b = g^{ab}$
Key Distribution from DH

**Theorem:** If \((t, \varepsilon)\)-DDH holds on \(G\), then the Diffie-Hellman protocol is \((t, \varepsilon)\)-secure.

**Proof:**
- \((\text{Trans}, k) = ( (g^a, g^b), g^{ab})\)
- DDH means indistinguishable from \(( (g^a, g^b), g^c)\)

What if only CDH holds, but DDH is easy?
Public Key Encryption
Public Key Encryption
Public Key Encryption

\[
c \leftarrow \text{Enc}(pk, m)
\]
Public Key Encryption

\[
\begin{align*}
sk & \quad \text{Dec}(sk, c) \\
\text{Enc}(pk, m) & \quad c \\
m & \quad \text{Alice} \\
\end{align*}
\]
Public Key Encryption

\[ \text{pk} \leftarrow \text{Enc}(\text{pk, m}) \]

\[ \text{m} \leftarrow \text{Dec}(\text{sk, c}) \]
PKE vs Key Agreement

Key agreement:

$k_{AB}$
PKE vs Key Agreement

Key agreement:

Alice: $k_{AB}$

Bob: $k_{AB}$
PKE vs Key Agreement

Key agreement:

Alice

$K_{AB}$

$K_{AC}$

Charlie

$K_{AC}$

King

$K_{AB}$
PKE vs Key Agreement

Key agreement:

For \(n\) users, need \(O(n^2)\) key exchanges
PKE vs Key Agreement

PKE:

\[ \text{sk}_A \rightarrow \text{pk}_A \]
PKE vs Key Agreement

PKE:

\[ \text{sk}_A \quad \text{pk}_A \quad \text{pk}_B \quad \text{sk}_B \]
PKE vs Key Agreement

PKE:

\[
\begin{align*}
    &pke: \\
    &sk_A, pk_A \\
    &sk_B, pk_B \\
    &sk_C, pk_C \\
\end{align*}
\]

For \(n\) users, need \(O(n)\) public keys
PKE Syntax

Message space $\mathcal{M}$

Algorithms:
• $(sk,pk) \leftarrow \text{Gen}(\lambda)$
• $\text{Enc}(pk,m)$
• $\text{Dec}(sk,m)$

Correctness:
$\Pr[\text{Dec}(sk,\text{Enc}(pk,m)) = m: (sk,pk) \leftarrow \text{Gen}(\lambda)] = 1$
Security

One-way security

Semantic Security

CPA security

CCA Security
One-way Security

\[(sk, pk) \leftarrow \text{Gen}() \]

\[m \leftarrow M\]

\[c \leftarrow \text{Enc}(pk, m)\]
Semantic Security

\[ (sk, pk) \leftarrow \text{Gen}() \]
\[ c \leftarrow \text{Enc}(pk, m_b) \]
CPA Security

\[(sk,pk) \leftarrow \text{Gen}() \]

\[c \leftarrow \text{Enc}(pk, m^b)\]
CCA Security

\[(sk, pk) \leftarrow \text{Gen}()\]

\[c \leftarrow \text{Enc}(pk, m_b)\]

\[m \neq c^*\]
Reminders

HW5 Due Today

PR2 Due April 19th