COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2017

Announcements

Homework 5 Up

- Due April 4

Today

Number Theory

So Far…

Two ways to construct cryptographic schemes:

- Use others as building blocks
  - PRGs → Stream ciphers
  - PRFs → PRPs
  - PRFs/PRPs → CPA-secure Encryption
  - …

- From scratch
  - RC4, DES, AES, etc

In either case, ultimately scheme or some building block built from scratch

Cryptographic Assumptions

Security of schemes built from scratch relies solely on our inability to break them

- No security proof
- Perhaps arguments for security

We gain confidence in security over time if we see that nobody can break scheme

Number-theory Constructions

Goal: base security on hard problems of interest to mathematicians

- Wider set of people trying to solve problem
- Longer history

Discrete Log

Discrete Log

Let **p** be a large integer (maybe prime)

Given $\mathbf{g} \in \mathbb{Z}_p^*$, $\mathbf{a} \in \mathbb{Z}$, easy to compute $\mathbf{g^a \bmod p}$
- Time **poly(log a, log p)**

However, no known efficient ways to recover **a** from **g** and $\mathbf{g^a \bmod p}$

Discrete Log Assumption: Let $\mathbf{p}$ be a $\boldsymbol{\lambda}$-bit integer.

Then the function $\mathbf{(g,a) \rightarrow (g,g^a \bmod p)}$ is one-way, where

- $\mathbf{g} \in \mathbb{Z}_{\mathbf{p}}^{*}$
- $\mathbf{a} \in \mathbb{Z}_{\Phi(\mathbf{p})}$

Generalizing Discrete Log

Let $\mathbf{G_\lambda}$ be multiplicative groups of size $\mathbf{n_\lambda}$

Definition: The discrete log assumption holds on $\{G_\lambda\}$ if the function $F: G_\lambda \times \{0,\ldots,n_\lambda - 1\} \rightarrow G_\lambda^2$ is one-way, where

$$F(g,a) = (g,g^a)$$

Plausible examples:

- $G = \mathbb{Z}_p^*$ for a prime p, $n = p-1$
- $G =$ subgroup of $\mathbb{Z}_p^*$ of order q, where $q | p-1$
- $G =$ "elliptic curve groups"

Non-example

G = additive group of integers mod **p**

- $g*h = g+h \bmod p$
- $g^{-1} = -g \bmod p$
- $g^a = g*g*g\ldots*g \bmod p = ag \bmod p$

Discrete log?

Generalizing Discrete Log

Often, the group $\mathbf{G}$ will be:

- Cyclic: $\mathbf{G} = \{1,g,g^2,\ldots,g^{|G|-1}\}$, $\mathbf{g}$ is a "generator"
- Of prime size

This means that every element except for the identity is a generator of $\mathbf{G}$

- $\mathbf{G} = \{1,g,g^2,\ldots,g^p\}$

Hardness of Discrete Log

Brute force search: **$O(n)$**

Better generic algorithm: **$O(n^{1/2})$**

- Known to be optimal for generic algorithms

Much better algorithms are known for $\mathbb{Z}_p^*$

$$\exp(C (\log p)^{1/3} (\log \log p)^{2/3})$$

(still super-polynomial)

For elliptic curves, best known attack is **$O(n^{1/2})$**

Applications of Discrete Log

One-way functions

Collision resistance

- Key space = $\mathbb{G}^2$, $\mathbb{G}$ has prime order p
- Domain: $\mathbb{Z}_p^2$
- Range: $\mathbb{G}$
- $H(\ (g,h),\ (x,y)\ ) = g^x h^y$

Collision Resistance from Discrete Log

$$H(\ (g,h),\ (x,y)\ ) = g^x h^y$$

Theorem: If the discrete log assumption holds on G, then **H** is collision resistant

Goal: show that from collision, can compute discrete log of **g** and **h**: **a** where **$h=g^a$**

Blum-Micali PRG

Let $\mathbf{G}=\mathbb{Z}_p^*$

Let $\mathbf{g}$ be a generator of $\mathbf{G}$

Let $\mathbf{h}:\mathbf{G}\rightarrow\{0,1\}$ be $h(x) = 1$ if $0<x<(p-1)/2$

Seed space: $\mathbb{Z}_p^*$

Algorithm:

- Let x_0 be seed
- For $i=0,\ldots$
  - Let $x_{i+1} = g^{x_i} \bmod p$
  - Output $h(x_i)$

Theorem: If the discrete log assumption holds on $\mathbb{Z}_p^*$, then the Blum-Micali generator is a secure PRG

We will prove this next time

Stronger Assumptions on Groups

Sometimes, the discrete log assumption is not enough

Instead, define stronger assumptions on groups

Computational Diffie-Hellman:

- Given (g, g^a, g^b), compute g^{ab}

Decisional Diffie-Hellman:

- Distinguish (g, g^a, g^b, g^c) from (g, g^a, g^b, g^{ab})

Hard Problems on Groups

Increasing Difficulty

DLog:

- Given (g, g^a), compute a

CDH:

- Given (g, g^a, g^b), compute g^{ab}

DDH:

- Distinguish (g, g^a, g^b, g^c) from (g, g^a, g^b, g^{ab})

Stronger Assumptions

Another PRG

Group **G** of order **p**

Seed space: $\mathbb{Z}_p^2$
Range: $\mathbf{G}^3$

PRG(a,b) = (g^a, g^b, g^{ab})

Naor-Reingold PRF

Domain: $\{0,1\}^n$
Key space: $\mathbb{Z}_p^{n+1}$
Range: $\mathbb{G}$

$$F(\ (a,b_1,b_2,\ldots,b_n),\ x\ ) = g^{a\ b_1^{x1}\ b_2^{x2}\ \ldots\ b_n^{xn}}$$

Theorem: If the discrete log assumption holds on **G,** then the Naor-Reingold PRF is secure

Proof by Hybrids

Hybrids **0**: $H(x) = g^{a \, b_1^{x1} \, b_2^{x2} \, \ldots \, b_n^{xn}}$

Hybrid **i**: $H(x) = H_i(x_{[1,i]})^{b_{i+1}^{x_{i+1}} \, \ldots \, b_n^{xn}}$

- H_i is a random function from $\{0,1\}^i \rightarrow G$

Hybrid **n**: $H(x)$ is truly random

Proof

Suppose adversary can distinguish Hybrid **i-1** from Hybrid **i** for some **i**

Easy to construct adversary that distinguishes:

$$x \rightarrow H_i(x) \text{ from } x \rightarrow H_{i-1}(x_{[1,i-1]})^{b^{x_i}}$$

Proof

Suppose adversary makes **2q** queries

- Assume wlog that queries are in pairs **x||0**, **x||1**

What does the adversary see?

- $H_i(x)$: **2q** random elements in **G**

- $H_{i-1}(x_{[1,i-1]})^{b_i^{x_i}}$: **q** random elements in **G**, $h_1,\ldots,h_q$
  as well as $h_1^b, \ldots, h_q^b$

Lemma: Assuming the DDH assumption on **G**, for any polynomial q, the following distributions are indistinguishable:

$$(g, g^{x_1}, g^{y_1}, \ldots, g^{x_q}, g^{y_q}) \text{ and}$$
$$(g, g^{x_1}, g^{b\,x_1}, \ldots, g^{x_q}, g^{b\,x_q})$$

Suffices to finish proof of NR-PRF

Proof of Lemma

Hybrids **0**: $(g, g^{x_1}, g^{b\,x_1}, \ldots, g^{x_q}, g^{b\,x_q})$

Hybrid **i:**

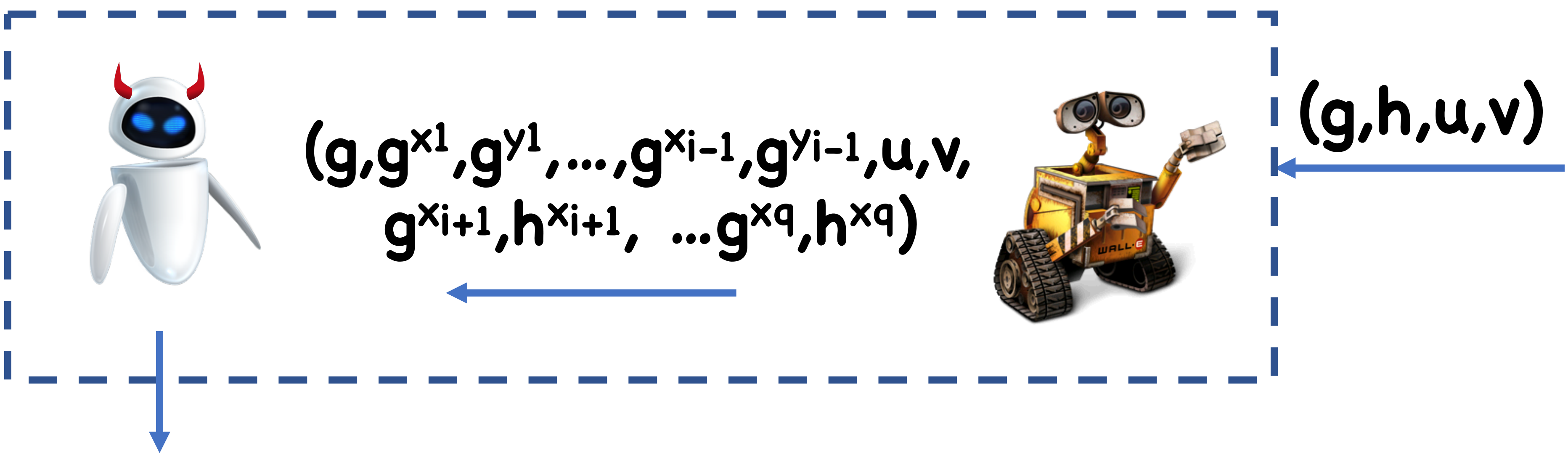
$(g, g^{x_1}, g^{y_1}, \ldots, g^{x_i}, g^{y_i}, g^{x_{i+1}}, g^{b\,x_{i+1}}, \ldots g^{x_q}, g^{b\,x_q})$

Hybrid **q**: $(g, g^{x_1}, g^{y_1}, \ldots, g^{x_q}, g^{y_q})$

Proof of Lemma

Suppose adversary distinguishes Hybrid i-1 from Hybrid i

Use adversary to break DDH:

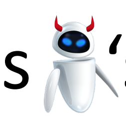


Proof of Lemma

$(g, g^{x_1}, g^{y_1}, \ldots, g^{x_{i-1}}, g^{y_{i-1}}, u, v, g^{x_{i+1}}, h^{x_{i+1}}, \ldots g^{x_q}, h^{x_q})$

If $(g,h,u,v) = (g, g^b, g^{x_i}, g^{b\,x_i})$, then Hybrid **i-1**

If $(g,h,u,v) = (g, g^b, g^{x_i}, g^{y_i})$, then Hybrid **i**

Therefore, 's advantage is the same as 's

Further Applications

From NR-PRF can construct:

- CPA-secure encryption
- Block Ciphers
- MACs
- Authenticated Encryption

Parameter Size in Practice?

G = subgroup of $\mathbb{Z}_p^*$ of order **q**, where **q| p-1**

- In practice, best algorithms require $p \geq 2^{1024}$ or so

- **G =** ”elliptic curve groups”
- Can set $p \approx 2^{256}$ to have security

  $\Rightarrow$ best attacks run in time 2^{128}

Therefore, elliptic curve groups tend to be much more efficient $\Rightarrow$ shift to using in practice

Integer Factorization

Integer Factorization

Given an integer **N**, factor **N** into its prime factors

Studied for centuries, presumed computationally difficult

- Grade school algorithm: $O(N^{1/2})$
- Much better algorithms:

$$\exp(C (\log N)^{1/3} (\log \log N)^{2/3})$$

- However, all require super-polynomial time

Factoring Assumption: Let $\mathbf{p}$, $\mathbf{q}$ be two random $\boldsymbol{\lambda}$-bit primes, and $\mathbf{N = pq}$. Then any PPT algorithm, given $\mathbf{N}$, has at best a negligible probability of recovering $\mathbf{p}$ and $\mathbf{q}$

One-way Functions From Factoring

P_λ = {λ–bit primes}

$F: P_\lambda^2 \rightarrow \{0,1\}^{2\lambda}$

$F(p,q) = p\times q$

Trivial Theorem: If factoring assumption holds, then **F** is one-way

Sampling Random Primes

Prime Number Theorem: A random **λ**-bit number is prime with probability **≈1/λ**

Primality Testing: It is possible in polynomial time to decide if an integer is prime

Fermat Primality Test (randomized, some false positives):

- Choose a random integer $\mathbf{a} \in \{0,\ldots,N-1\}$
- Test if $\mathbf{a^N = a \bmod N}$
- Repeat many times

Another OWF

Fix a large integer $\mathbf{N = pq}$

- Primes $\mathbf{p,q}$ random, unknown

$$F_N(x) = x^2 \bmod N$$

Theorem: If the factoring assumption holds, then F is one-way: given $\mathbf{y}$, computaitonally infeasible to compute an x such that $x^2 = y \bmod N$

Chinese Remainder Theorem

Let $\mathbf{N} = \mathbf{pq}$ for distinct prime $\mathbf{p,q}$

Let $\mathbf{x} \in \mathbb{Z}_{\mathbf{p}}$, $\mathbf{y} \in \mathbb{Z}_{\mathbf{q}}$

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

- $\mathbf{x = z \bmod p}$, and
- $\mathbf{y = z \bmod q}$

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

Quadratic Residues

Definition: y is a quadratic residue mod N if there exists an x such that $y = x^2 \bmod N$. 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?
- Let $N=pq$ be the product of two primes, y a quadratic residue mod N. Suppose $y \neq 0 \bmod p$ and $y \neq 0 \bmod q$. How many square roots?

Another OWF

Fix a large integer **N = pq**

- Primes **p,q** random, unknown

$F_N(x) = x^2 \bmod N$

Theorem: If the factoring assumption holds, then **F** is one-way: given QR **y**, computationally infeasible to compute an **x** such that $x^2 = y \bmod N$

Proof

Let  be an adversary that, given a random quadratic residue **y** mod **N**, finds a square root **x**.

How to factor:

- Choose a random **z mod N**
- Compute **y = z**^{2} **mod N**
- Run  on **y** to get a root **x**
- Let **p = GCD(z-x, N), q = N/p**

Analysis

Let x be the output of .

Given a y, z was equally likely to be each of the 4 quadratic residues:

- x
- $-x$
- w: $w = x \bmod p$, $w = -x \bmod q$
- $-w$

With probability ½, $z = \pm w$

Analysis

Suppose **z = w**

$\Rightarrow$ **z = x mod p, z = -x mod q**

$\Rightarrow$ **z-x=0 mod p, z+x=0 mod q**

Therefore, **GCD(z-x,N) = p**

- Algorithm succeeds

z = -w case identical, except algorithm flips **p** and **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: $\{1,\ldots,(N-1)/2\}\times\{0,1\}$
Range: $\{1,\ldots,(N-1)/2\}$

H((N,y), (x,b)): Let $z = y^b x^2 \bmod N$
- If $z\in\{1,\ldots,(N-1)/2\}$, output z
- Else, output $-z \bmod N \in\{1,\ldots,(N-1)/2\}$

Theorem: If the factoring assumption holds, **H** is collision resistant

Proof:

- Collision means $(x_0,b_0)\neq(x_1,b_1)$ s.t.

$$y^{b0}\ x_0^2 = \pm\ y^{b1}\ x_1^2 \bmod N$$

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

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

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 $\mathbf{p}$ be a prime, and suppose $\mathbf{p = 3 \bmod 4}$

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

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

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

Solving Quadratic Equations

In general, solving quadratic equations is:

- Easy over prime moduli
- As hard as factoring over composite moduli

Other Powers?

What about $\mathbf{x \rightarrow x^4 \bmod N}$? $\mathbf{x \rightarrow x^6 \bmod N}$?

The function $\mathbf{x \rightarrow x^3 \bmod N}$ appears quite different

- Suppose **3** is relatively prime to $\mathbf{p-1}$ and $\mathbf{q-1}$
- Then $\mathbf{x \rightarrow x^3 \bmod p}$ is injective for $\mathbf{x \neq 0}$
  - Let $\mathbf{a}$ be such that $\mathbf{3a = 1 \bmod p-1}$
  - $\mathbf{(x^3)^a = x^{1+k(p-1)} = x(x^{p-1})^k = x \bmod p}$
- By CRT, $\mathbf{x \rightarrow x^3 \bmod N}$ is injective for $\mathbf{x} \in \mathbb{Z}_N^*$

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 = GCD(z-x, N), q = N/p**

RSA Problem

Given

- **$N = pq$**,
- **e** such that **$GCD(e,p-1)=GCD(e,q-1)=1$**,
- **$y=x^e \bmod N$** for a random **x**

Find **x**

Injectivity means cannot base hardness on factoring, but still conjectured to be hard

Later, we will see why this version is useful

Roadmap

Next week:

- OWF → almost everything we've seen so far

After that:

- Public key cryptography