COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2020

Announcements

HW4 Due Today
HW5 Due April 9th
PR2 Due April 19th

Previously on COS 433…

Anagrams and Astronomy

<u>Galileo and the Rings of Saturn</u>

- 1610: Galileo observed the rings of Saturn, but mistook them for two moons

- Galileo wanted extra time for verification, but not to get scooped

- Circulates anagram
  SMAISMRMILMEPOETALEUMIBUNENUGTTAUIRAS

- When ready, tell everyone the solution:
  altissimum planetam tergeminum observavi
  (“I have observed the highest planet tri-form”)

(Non-interactive) Commitment Syntax

Message space **M**
Ciphertext Space **C**
(suppressing security parameter)

Com(m; r): outputs a commitment **c** to **m**

- Why have **r**?

Commitments with Setup

Message space **M**
Ciphertext Space **C**
(suppressing security parameter)

Setup(): Outputs a key **k**
Com(k, m; r): outputs a commitment **c** to **m**

Using Commitments

Commit Stage

$r \leftarrow R$

$c \leftarrow Com(m;r)$

m

Reveal Stage

m,r

Check that

$c = Com(m;r)$

Using Commitments (with setup)

k←Setup()

Commit Stage

r←R

c←Com(k,m;r)

m

Reveal Stage

m,r

Check that

c = Com(k,m;r)

Security Properties

Hiding: **c** should hide **m**

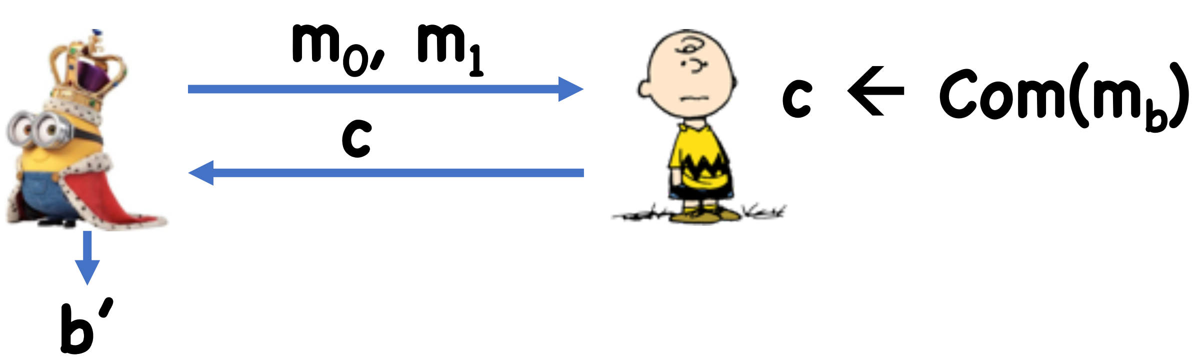
- Perfect hiding: for any m_0, m_1,

$$\mathrm{Com}(m_0) \stackrel{d}{=} \mathrm{Com}(m_1)$$

- Statistical hiding: for any m_0, m_1,

$$\Delta(\ \mathrm{Com}(m_0),\ \mathrm{Com}(m_1)\ ) < \mathrm{negl}$$

- Computational hiding:



Security Properties (with Setup)

Hiding: **c** should hide **m**

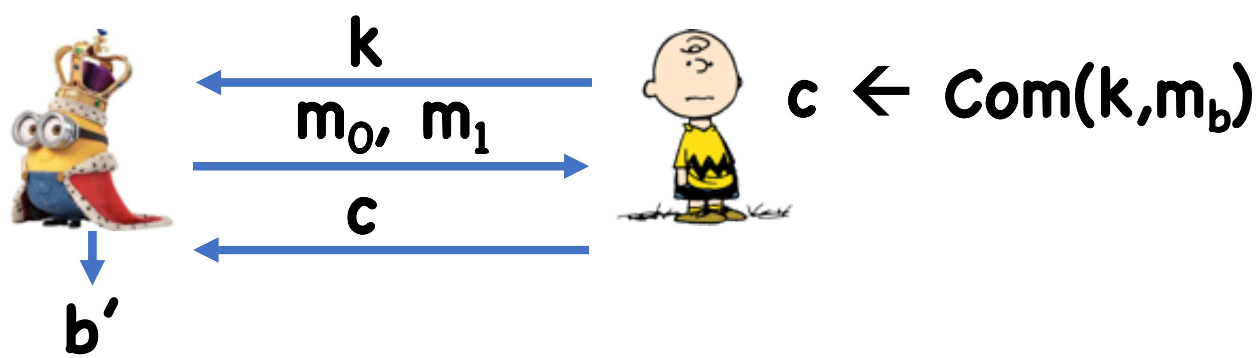
- Perfect hiding: for any $\mathbf{m_0}$, $\mathbf{m_1}$,

$$k, Com(k, m_0) \stackrel{d}{=} k, Com(k, m_1)$$

- Statistical hiding: for any $\mathbf{m_0}$, $\mathbf{m_1}$,

$$\Delta(\ [k, Com(k, m_0)],\ [k, Com(k, m_1)]\ ) < negl$$

- Computational hiding:



Security Properties

Binding: Impossible to change committed value

- Perfect binding: For any $\mathbf{c}$, $\exists$ at most a single $\mathbf{m}$ such that $\mathbf{c = Com(m;r)}$ for some $\mathbf{r}$

- Computational binding: no efficient adversary can find $\mathbf{(m_0,r_0),(m_1,r_1)}$ such that:

  $\mathbf{Com(m_0;r_0)=Com(m_1;r_1)}$

  $\mathbf{m_0 \neq m_1}$

Security Properties (with Setup)

Binding: Impossible to change committed value

- Perfect binding: For any **k,c**, $\exists$ at most a single **m** such that **c = Com(k,m;r)** for some **r**

- Statistical binding: except with negligible prob over **k**, for any **c**, $\exists$ at most a single **m** such that **c = Com(k,m;r)** for some **r**

- Computational binding: no PPT adversary, given $\mathbf{k \leftarrow Setup()}$, can find $(\mathbf{m_0},\mathbf{r_0}),(\mathbf{m_1},\mathbf{r_1})$ such that

  $\mathbf{Com(k,m_0;r_0)=Com(k,m_1;r_1)}$

  $\mathbf{m_0 \neq m_1}$

Statistically Hiding Commitments

Let $\mathbf{F}$ be a pairwise independent function family with domain $\mathbf{X=\{0,1\}\times R}$ and range $\mathbf{Y}$

Let $\mathbf{H}$ be a collision resistant hash function with domain $\mathbf{Y}$ and range $\mathbf{Z}$

Setup(): f←F, k←K, output **(f,k)**
Com((f,k), m; r) = H(k, f(m,r))

Theorem: If **H** is collision resistant and $|X|^2/|Y|$ is negligible, then **(Setup,Com)** is computationally binding

Proof:

- Suppose $|Y| \times \gamma = |X|^2$
- For any $x_0 \neq x_1$, $\Pr[f(x_0)=f(x_1)] < \gamma/(|X|^2)$
- Union bound:

$$\Pr[\exists \ x_0 \neq x_1 \text{ s.t. } f(x_0)=f(x_1)] < \gamma$$

- Therefore, **f** is injective $\Longrightarrow$ any collision for Com must be a collision for **H**

Min-entropy

Definition: Given a distribution $\mathbf{D}$ over a set $\mathbf{X}$, the min-entropy of $\mathbf{D}$, denoted $\mathbf{H_\infty(D)}$, is

$$\min_x -\log_2(\Pr[x \leftarrow D])$$

Examples:

- $H_\infty(\{0,1\}^n) = n$
- $H_\infty($ random $\mathbf{n}$ bit string with parity $0) = ?$
- $H_\infty($ random $i>0$ where $\Pr[i] = 2^{-i}) = ?$

Leftover Hash Lemma

Lemma: Let $\mathbf{D}$ be a distribution on $\mathbf{X}$, and $\mathbf{F}$ a family of pairwise independent functions from $\mathbf{X}$ to $\mathbf{Y}$. Then

$$\Delta(\ (f,\ f(D))\ ,\ (f,\ R)\ )\ \leq\ \varepsilon \text{ where}$$

- $f \leftarrow F$
- $R \leftarrow Y$
- $\log |Y| \leq H_\infty(D) + 2 \log \varepsilon$

"Crooked" Leftover Hash Lemma

Lemma: Let **D** be a distribution on **X**, and **F** a family of pairwise independent functions from **X** to **Y**, and **h** be any function from **Y** to **Z**. Then

$$\Delta(\ (f,\ h(f(D)))\ ,\ (f,\ h(R))\ ) \leq \varepsilon \text{ where}$$

- $f \leftarrow F$
- $R \leftarrow Y$
- $\log |Z| \leq H_\infty(D) + 2 \log \varepsilon - 1$

This Time

Commitments continued

Number theory

Theorem: If we set $|R|=|Z|^3$ and $|Z|$ is super-poly, then **(Setup,Com)** is statistically hiding

Goal: show $(f,\ k,\ H(k,\ f(0,r))\ )$ is statistically close to $(f,\ k,\ H(k,\ f(1,r))\ )$

Let $D_b=(b,r)$, min-entropy $\log\ |R|$

Set $R\ =|Z|^3$, $\varepsilon\ =\ 2/|Z|$

Then $\log\ |Z| \leq H_\infty(D_b) + 2\ \log\ \varepsilon - 1$

Theorem: If we set $|R|=|Z|^3$ and $|Z|$ is super-poly, then **(Setup,Com)** is statistically hiding

For any **k**,**b**,

$$\Delta(\ (f,\ H(k,\ f(b,r)))\ ,\ (f,\ H(k,\ U))\ )\ \le\ \varepsilon$$

Thus (for any **k**)

$$\Delta(\ (f,\ H(k,\ f(0,r)))\ ,\ (f,\ H(k,\ f(1,r)))\ )\ \le\ 2\varepsilon$$

Therefore

$$\Delta(\ (f,\ k,\ H(k,\ f(0,r))\ )\ ,\ (f,\ k,\ H(k,\ f(1,r))\ )\ \le\ 2\varepsilon$$

Statistically Binding Commitments

Let **G** be a PRG with domain $\{0,1\}^\lambda$, range $\{0,1\}^{3\lambda}$

Setup(): choose and output a random 3λ-bit string **k**

Com(b; r): If $b=0$, output $G(r)$, if $b=1$, output $G(r)\oplus k$

Theorem: (Setup,Com) is statistically binding

Theorem: If **G** is a secure PRG, then **(Setup,Com)** is computationally hiding

Theorem: If $\mathbf{G}$ is a secure PRG, then **(Setup,Com)** is computationally hiding

Hybrids:

- Hyb 0: $\mathbf{c = Com(0;r) = G(r)}$ where $\mathbf{r \leftarrow \{0,1\}^\lambda}$
- Hyb 1: $\mathbf{c \leftarrow \{0,1\}^{3\lambda}}$
- Hyb 2: $\mathbf{c = S' \oplus k}$, where $\mathbf{S' \leftarrow \{0,1\}^{3\lambda}}$
- Hyb 3: $\mathbf{c = Com(1;r) = G(r) \oplus k}$ where $\mathbf{r \leftarrow \{0,1\}^\lambda}$

Theorem: (Setup,Com) is statistically binding

Proof:

For any r,r', $\Pr[G(r) = G(r')\oplus k] = 2^{-3\lambda}$

By union bound:

$\Pr[\exists\, r,r'$ such that $Com(k,0)=Com(k,1)]$

$= \Pr[\exists\, r,r'$ such that $G(r) = G(r')\oplus k] < 2^{-\lambda}$

More Problems with Anagrams

Huygens Discovers Saturn's moon Titan

- 1655: Sends the following to Wallis

ADMOVERE OCULIS DISTANTIA SIDERA NOSTRIS, UUUUUUUCCCRR-HNBQX

(First part meaning "to direct our eyes to distant stars")

Plaintext: **saturno luna sua circunducitur diebus sexdecim horis quatuor**

("Saturn's moon is led around it in sixteen days and four hours")

More Problems with Anagrams

<u>Huygens Discovers Saturn’s moon Titan</u>

- Wallis replies with

AAAAAAAAA B CCCCC DDDD EEEEEEEEE F H
IIIIIIIIIIII LLL MMMMMM NNNNNN OOOOOOO PPPPP
Q RRRRRRRRRRR SSSSSSSSSSSSS TTTTTTTT
UUUUUUUUUUUUUUUUU X

(Contains all of the letters in Huygens’ message, plus some)

More Problems with Anagrams

Huygens Discovers Saturn's moon Titan

- When Huygens finally reveals his discovery, Wallis responds by giving solution to his anagram:

saturni comes quasi lunando vehitur. diebus sexdecim circuitu rotatur. novas nuper saturni formas telescopo vidimus primitus. plura speramus

("A companion of Saturn is carried in a curve. It is turned by a revolution in sixteen days. We have recently observed new shapes of Saturn with a telescope. We expect more.")

- Tricked Huygens into thinking British astronomers had already discovered Titan

More Problems with Anagrams

Sometimes, hiding and binding are not enough

For some situations (e.g. claiming priority on discoveries) also want commitments to be “non-malleable”

- Shouldn’t be able to cause predictable changes to committed value

Beyond scope of this course

Number Theory and Crypto

(Handout on course website with basic number theory primer)

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

Number Theory

$\mathbb{Z}_N$: integers mod **N**

$\mathbb{Z}_N^*$: integers mod **N** that are relatively prime to **N**

- $x \in \mathbb{Z}_N^*$ iff **x** has an "inverse" **y** s.t. **xy mod N = 1**
- For prime **N,** $\mathbb{Z}_N^* = \{1,\ldots,N-1\}$

$\Phi(N) = |\mathbb{Z}_N^*|$

Euler's theorem: for any $x \in \mathbb{Z}_N^*$, $x^{\Phi(N)} \bmod N = 1$

Discrete Log

Discrete Log

Let $\mathbf{p}$ be a large number (usually prime)

Given $\mathbf{g} \in \mathbb{Z}_p^*$, $\mathbf{a} \in \mathbb{Z}$, "easy" to compute $\mathbf{g^a \bmod p}$

- Time $\mathbf{poly(\log a, \log p)}$
- How?

However, no known efficient ways to recover $\mathbf{a}$ (mod $\mathbf{\Phi(p)=p-1}$) from $\mathbf{g}$ and $\mathbf{g^a \bmod p}$

Cyclic Groups

For prime $\mathbf{p}$, $\mathbb{Z}_p^*$ is cyclic, meaning

$$\exists\ g \text{ s.t. } \mathbb{Z}_p^* = \{1, g, g^2, \ldots, g^{p-2}\}$$

(we call such a $\mathbf{g}$ a generator)

However, not all $\mathbf{g}$ are generators

- If g_0 is a generator, then $g = g_0^2$ is not:

  $g^{(p-1)/2} = g^{p-1} = 1$, so $|\{1, g, \ldots\}| \leq (p-1)/2$

- How to test for generator?

Hardness of DLog

For prime **p**, best know algorithms:

- Brute force: $O(p)$
- Better algs based on birthday paradox: $O(p^{1/2})$
- Even better heuristic algorithms:

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

(super polynomial in $\log p$)

For non-prime **p**, some cases are easy

Sampling Large 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 $\mathbf{a}^N = \mathbf{a} \bmod N$
- Repeat many times

Discrete Log Assumption: For any discrete log algorithm  running in time polynomial time, there exists negligible ε such that:

$\Pr[a \leftarrow$  $(p, g, g^a \bmod p)$:

$p \leftarrow$ random λ-bit prime

$g \leftarrow$ random generator of $\mathbb{Z}_p^*$,

$a \leftarrow \mathbb{Z}_{p-1}$ $] \leq \varepsilon(\lambda)$

Collision Resistance from DLog

Let **p** be a prime

- Key space = $\mathbb{Z}_p^2$
- Domain: $\mathbb{Z}_{p-1}^2$
- Range: $\mathbb{Z}_p$
- $\mathbf{H((g,h), (x,y)) = g^x h^y}$

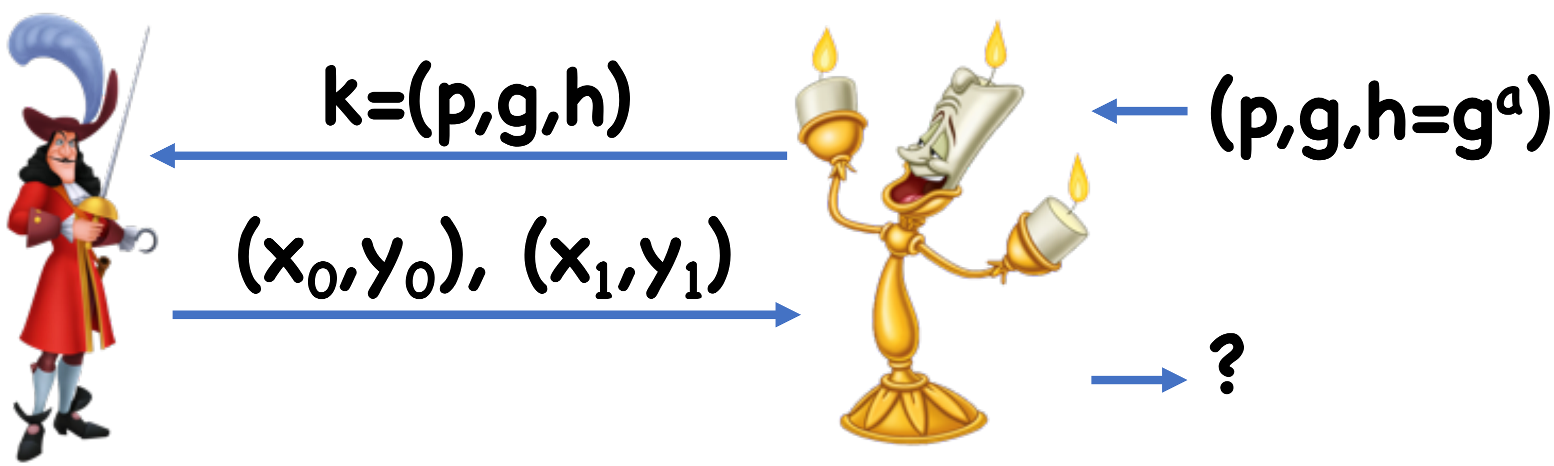
To generate key, choose random **p**, **g,h** $\in \mathbb{Z}_p^*$

- Require **g** a generator

Collision Resistance from Discrete Log

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

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



Collision Resistance from Discrete Log

Proof idea:

- Input to **H** is equation for a line **line(a)=ay+x**
- **H(line) = g**$^{\text{line(a)}}$ (evaluation "in the exponent")
- A collision is two different lines that intersect at **a**
- Use equations for two lines to solve for **a**:

$$a = -(x_1-x_0)/(y_1-y_0) \pmod{p-1}$$

Problem

For **$p>2$**, **$p-1$** is not a prime, so has some factors

Therefore, **(y_1-y_0)** not necessarily invertible mod **$p-1$**

However, possible to show that if this is the case, either:

- **(y_1-y_0)** and **(x_1-x_0)** have common factor, so can remove factor and try again, or
- **g** is not a generator (which isn't allowed)

Blum-Micali PRG

Let $\mathbf{p}$ be a prime

Let $\mathbf{g} \in \mathbb{Z}_p^*$

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

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

Algorithm:

- Let $\mathbf{x_0}$ be seed
- For $\mathbf{i=0,\ldots}$
  - 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 (if time)

Another PRG

p a prime
Let **g** be a generator

Seed space: $\mathbb{Z}_{p-1}^{2}$
Range: $\mathbb{Z}_{p}^{3}$

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

Don't know how to prove security from DLog

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})

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

Computational Diffie Hellman: For any algorithm  running in polynomial time, there exists negligible $\boldsymbol{\varepsilon}$ such that:

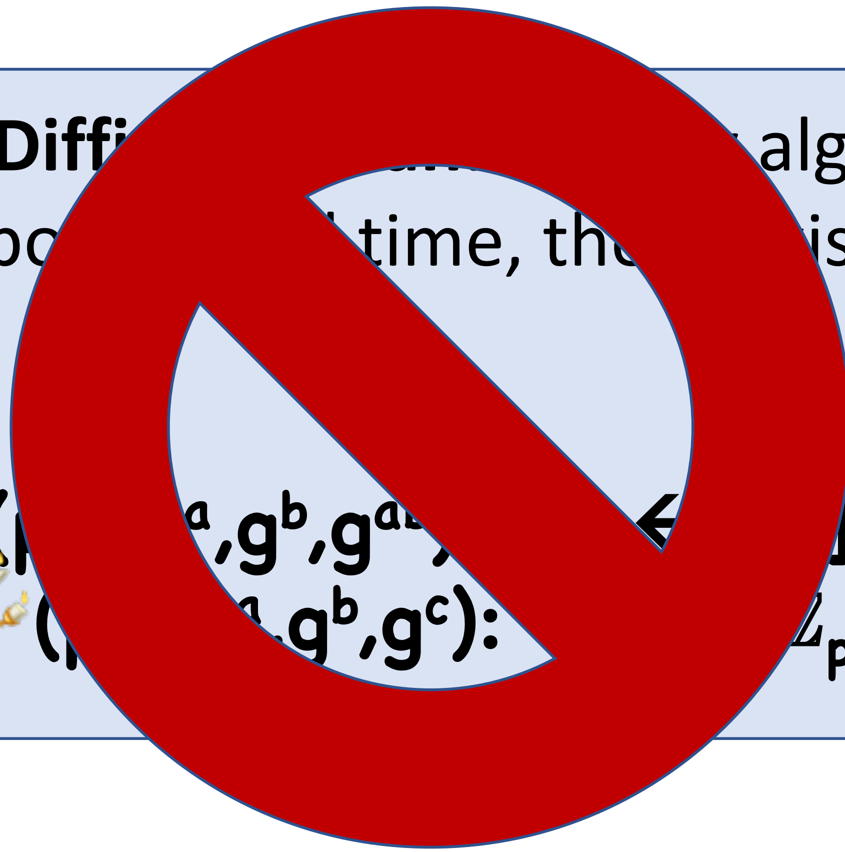
Pr[g^{ab}←  **(p,g,g^a,g^b):**

p←random **λ**-bit prime

g←random generator of $\mathbb{Z}_p^*$,

a,b←$\mathbb{Z}_{p-1}$ **] ≤ ε(λ)**

Decisional Diffi algorithm 
running in po l time, the ists negligible **ε**
such that:



|Pr[1←  **(p a,g^b,g^{ab}) ←]**
-Pr[1←  **(,g^b,g^c): $\mathbb{Z}_{p-1}$]| ≤ ε(λ)**

Hardness of DDH

Need to be careful about DDH

Turns out that DDH as described is usually easy:

- For prime **$p>2$, $\Phi(p)=p-1$** will have small factors
- Can essentially reduce solving DDH to solving DDH over a small factor

Fixing DDH

Let $\mathbf{g_0}$ be a generator

Suppose $\mathbf{p-1 = qr}$ for prime $\mathbf{q}$, integer $\mathbf{r}$

Let $\mathbf{g=g_0^r}$

$\mathbf{g^q \bmod p = 1}$, but $\mathbf{g^{q'} \bmod p \neq 1}$ for any $\mathbf{q'<q}$

- So $\mathbf{g}$ has "order" $\mathbf{q}$

Let $\mathbf{G = \{1,g,g^2,\ldots\}}$ be group "generated by" $\mathbf{g}$

Generalizing Cryptographic Groups

Replace fixed family of groups with “group generator” algorithm

$$(G,g,q) \leftarrow \mathrm{GrGen}(\lambda)$$

Group

Generator for group

Group oder

Decisional Diffie Hellman for GrGen:

For any algorithm  running in polynomial time, there exists negligible ε such that:

$\Big|\ \Pr[1 \leftarrow$  $(g, g^a, g^b, g^{ab}) :$

$(G,g,q) \leftarrow \text{GrGen}(\lambda),\ a,b \leftarrow \mathbb{Z}_q]$

$- \Pr[1 \leftarrow$  $(g, g^a, g^b, g^c) :$

$(G,g,q) \leftarrow \text{GrGen}(\lambda),\ a,b,c \leftarrow \mathbb{Z}_q]\ \Big| \le \varepsilon(\lambda)$

Another PRG

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

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

Security almost immediately follows from DDH

Generalizing Cryptographic Groups

Can also define Dlog, CDH relative to general **GrGen**

In many cases, problems turns out easy
Ex: $\mathbf{G} = \mathbf{Z_q}$, where $\mathbf{g \otimes h = g+h \bmod q}$
- What is exponentiation in $\mathbf{G}$?
- What is discrete log in $\mathbf{G}$?

Essentially only two groups where Dlog/CDH/DDH is conjectured to be hard:
- $\mathbb{Z}_p^*$ and its subgroups
- “Elliptic curve” groups

Parameter Size in Practice?

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

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

- $\mathbf{G}$ = "elliptic curve" group
- Can set $\mathbf{p \approx 2^{256}}$ to have security
  $\Rightarrow$ best attacks run in time $\mathbf{2^{128}}$

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

Naor-Reingold PRF

Domain: $\{0,1\}^n$
Key space: $\mathbb{Z}_q^{n+1}$
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 DDH assumption holds on $\mathbb{G}$**,** then the Naor-Reingold PRF is secure

Proof by Hybrids

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

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

- 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:

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

Proof

Suppose adversary makes $\mathbf{2r}$ queries

- Assume wlog that queries are in pairs $\mathbf{x||0}$, $\mathbf{x||1}$

What does the adversary see?

- $\mathbf{H_i(x)}$: $\mathbf{2r}$ random elements in $\mathbf{G}$

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

Lemma: Assuming the DDH assumption on $\mathbf{G}$, for any polynomial $\mathbf{r}$, 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**: $(g, g^{x_1}, g^{b\, x_1}, \ldots, g^{x_r}, g^{b\, x_r})$

Hybrid **i:**
$(g, g^{x_1}, g^{y_1}, \ldots, g^{x_i}, g^{y_i}, g^{x_{i+1}}, g^{b\, x_{i+1}}, \ldots g^{x_r}, g^{b\, x_r})$

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

Proof of Lemma

Suppose adversary distinguishes Hybrid **i-1** from Hybrid **i**

Use adversary to break DDH:

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

$$(g, h, u, v)$$

Proof of Lemma

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

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

Reminders

HW4 Due Today
HW5 Due April 9^{th}

PR2 Due April 19^{th}