

# COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Spring 2018

# (Non-interactive) Commitment Syntax

Message space  **$\mathcal{M}$**

Ciphertext Space  **$\mathcal{C}$**

(suppressing security parameter)

**$\text{Com}(\mathbf{m}; \mathbf{r})$** : outputs a commitment  **$\mathbf{c}$**  to  **$\mathbf{m}$**

# Commitments with Setup

Message space  **$\mathcal{M}$**

Ciphertext Space  **$\mathcal{C}$**

**Setup()**: Outputs a key  **$k$**

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

# Using Commitments

Reveal Stage

Commit Stage



$m$

$r \leftarrow R$

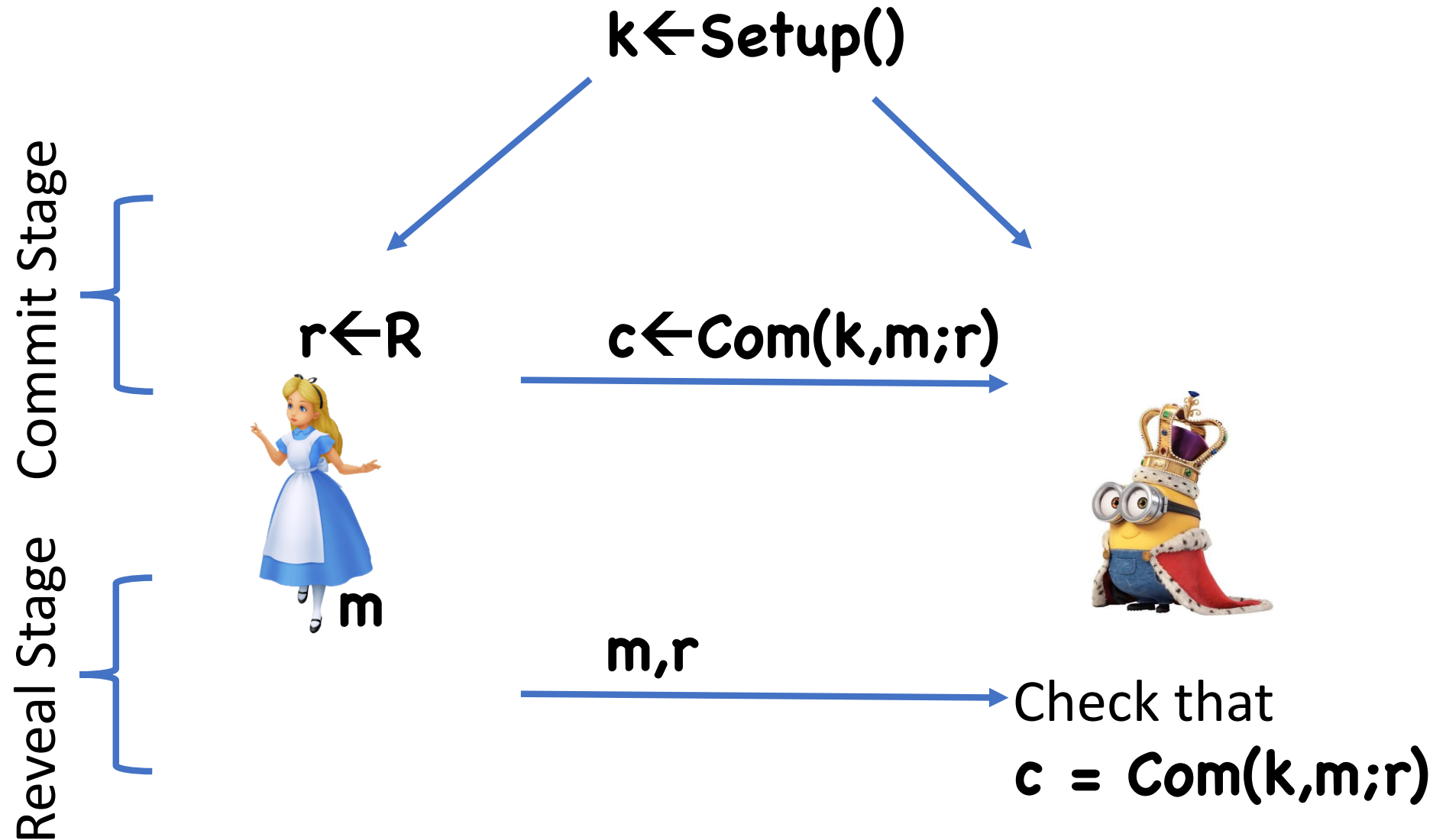
$c \leftarrow \text{Com}(m;r)$



$m, r$

Check that  
 $c = \text{Com}(m;r)$

# Using Commitments (with setup)



# Security Properties (with Setup)

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

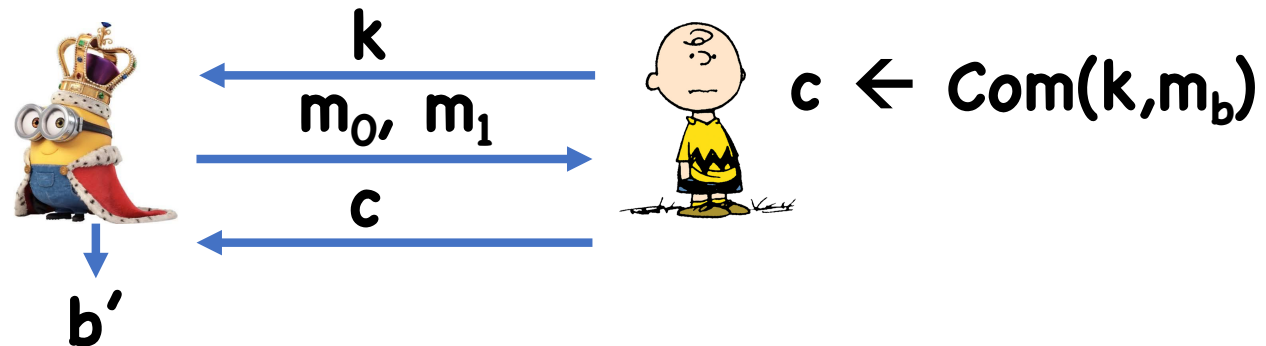
- Perfect hiding: for any **m<sub>0</sub>**, **m<sub>1</sub>**,

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

- Statistical hiding: for any **m<sub>0</sub>**, **m<sub>1</sub>**,

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

- Computational hiding:



# Security Properties (with Setup)

Binding: Impossible to change committed value

- Perfect binding: For any  $\mathbf{k}, \mathbf{c}$ ,  $\exists$  at most a single  $\mathbf{m}$  such that  $\mathbf{c} = \mathbf{Com}(\mathbf{k}, \mathbf{m}; \mathbf{r})$  for some  $\mathbf{r}$
- Statistical binding: except with negligible prob over  $\mathbf{k}$ , for any  $\mathbf{c}$ ,  $\exists$  at most a single  $\mathbf{m}$  such that  $\mathbf{c} = \mathbf{Com}(\mathbf{k}, \mathbf{m}; \mathbf{r})$  for some  $\mathbf{r}$
- Computational binding: no efficient adversary, given  $\mathbf{k} \leftarrow \mathbf{Setup}()$ , can find  $(\mathbf{m}_0, \mathbf{r}_0), (\mathbf{m}_1, \mathbf{r}_1)$  such that  $\mathbf{Com}(\mathbf{k}, \mathbf{m}_0; \mathbf{r}_0) = \mathbf{Com}(\mathbf{k}, \mathbf{m}_1; \mathbf{r}_1)$   
 $\mathbf{m}_0 \neq \mathbf{m}_1$

# A Simple Commitment Scheme

Let **H** be a hash function

$$\mathbf{Com(m;r) = H(m \parallel r)}$$

Binding?

Hiding?



# Statistically Hiding Commitments?

Let  $H$  be a collision resistant hash function with domain  $X = \{0,1\} \times R$  and range  $Z$

**Setup():**  $k \leftarrow K$ , output  $k$

**Com(  $k, m; r$  )** =  $H(k, (m,r))$

Binding?

Hiding?

# Statistically Hiding Commitments

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

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

**Setup():**  $f \leftarrow \mathbf{F}$ ,  $k \leftarrow \mathbf{K}$ , output  $(f,k)$

**Com(  $(f,k)$ ,  $m$ ;  $r$  )** =  $\mathbf{H}(k, f(m,r))$

**Theorem:** If  $H$  is  $(t, \epsilon)$ -collision resistant, then  $(\text{Setup}, \text{Com})$  is  $(t - t', \epsilon + |X|^2 / |Y|)$ -computationally binding

Proof:

- Suppose  $|Y| = |X|^2 \times \gamma$
- For any  $x_0 \neq x_1$ ,  $\Pr[f(x_0) = f(x_1)] < 1 / (|X|^2 \times \gamma)$
- Union bound:  
$$\Pr[\exists x_0 \neq x_1 \text{ s.t. } f(x_0) = f(x_1)] < 1/\gamma$$
- Therefore,  $f$  is injective  $\Rightarrow$  any collision for Com must be a collision for  $H$

**Theorem:** If  $|X|$  is “sufficiently large”, then  $(\text{Setup}, \text{Com})$  has statistical hiding

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

# Min-entropy

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

$$- \max_x \log_2( \Pr[x \leftarrow \mathbf{D}] )$$

Examples:

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

# 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

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

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

Intuitively,  $f(\mathbf{D})$  looks random, provided the min-entropy of  $\mathbf{D}$  is large enough

# “Crooked” 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  $\mathbf{Y}$ , and  $\mathbf{h}$  be any function from  $\mathbf{Y}$  to  $\mathbf{Z}$ . Then

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

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

Makes sense even for  $\mathbf{f}$  with large output

**Theorem:** If we set  $|R|=|Z|^3$ , then **(Setup,Com)** is  $(4/|Z|)$ -statistically hiding

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

Let  $D=(0,r)$ , min-entropy  $\log |R|$

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

Then  $\log |Z| \leq H_\infty(D) + 2 \log \epsilon - 1$



**Theorem:** If we set  $|R|=|Z|^3$ , then **(Setup,Com)** is  $(4/|Z|)$ -statistically hiding

For any  $k$ ,

$$\Delta( (f, H(k, f(0,r))) , (f, H(k, U)) ) \leq \varepsilon$$

Thus

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

Therefore

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

# Statistically Binding Commitments

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

**Setup():** choose and output a random  $3\lambda$ -bit string  $\mathbf{k}$

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

**Theorem:**  $(\text{Setup}, \text{Com})$  is  $(2^{-\lambda})$ -statistically binding

**Theorem:** If  $\mathbf{G}$  is a  $(t, \epsilon)$ -secure PRG, then  $(\text{Setup}, \text{Com})$  is  $(t - t', 2\epsilon)$ -computationally hiding

**Theorem:** If  $G$  is a  $(t, \epsilon)$ -secure PRG, then  $(\text{Setup}, \text{Com})$  is  $(t - t', 2\epsilon)$ -computationally hiding

Hybrids:

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

**Theorem: (Setup, Com) is  $(2^{-\lambda})$ -statistically binding**

Proof:

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

By union bound:

$$\begin{aligned} & \Pr[\exists r, r' \text{ such that } \text{Com}(k, 0) = \text{Com}(k, 1)] \\ &= \Pr[\exists r, r' \text{ such that } G(r) = G(r') \oplus k] < 2^{-\lambda} \end{aligned}$$

# More Problems with Anagrams

## Huygens Discovers Saturn's moon Titan

- Sends the following to Wallis

**ADMOVERE OCULIS DISTANTIA SIDERA NOSTRIS,  
UUUUUUUCCCRH-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

## Huygens Discovers Saturn's moon Titan

- Wallis replies with

AAAAAAAAA B CCCCC DDDD EEEEEEEEE F H  
IIIIIIIIII LLL MMMMM NNNNNN OOOOOO PPPP  
Q RRRRRRRRRR SSSSSSSSSSSS TTTTTTT  
UUUUUUUUUUUUUUUUUUU 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

# So Far...

Two ways to construct cryptographic schemes:

- Use others as building blocks
  - PRGs  $\rightarrow$  Stream ciphers
  - PRFs  $\rightarrow$  PRPs
  - PRFs/PRPs  $\rightarrow$  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.  $xy \bmod N = 1$
- For prime  $N$ ,  $\mathbb{Z}_N^* = \{1, \dots, N-1\}$

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

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

# Discrete Log

# Discrete Log

Let  $p$  be a large number (usually prime)

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

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

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



# Cyclic Groups

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

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


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


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

- If  $\mathbf{g}_0$  is a generator, then  $\mathbf{g} = \mathbf{g}_0^2$  is not:

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

- How to test for generator?

**$(p, t, \epsilon)$ -Discrete Log Assumption:** For any discrete log algorithm  running in time at most  $t$ ,

$$\Pr[a \leftarrow \text{ (g, g^a \bmod p):$$
$$g \leftarrow \text{generator of } \mathbb{Z}_p^*,$$
$$a \leftarrow \mathbb{Z}_{p-1} \quad ] \leq \epsilon$$

# Hardness of DLog

For prime  $p$ , best known algorithms:

- Brute force:  $O(p)$
- Better algs based on birthday paradox:  $O(p^{1/2})$
- Even better heuristic algorithms:  
 $\exp( C (\log p)^{1/3} (\log \log p)^{2/3} )$

- Therefore, plausible assumption:

$$(N, t=2^{(\log p)^{1/3}}, \epsilon=2^{-(\log p)^{1/3}})$$

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

# Collision Resistance from DLog

Let **p** be a prime

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

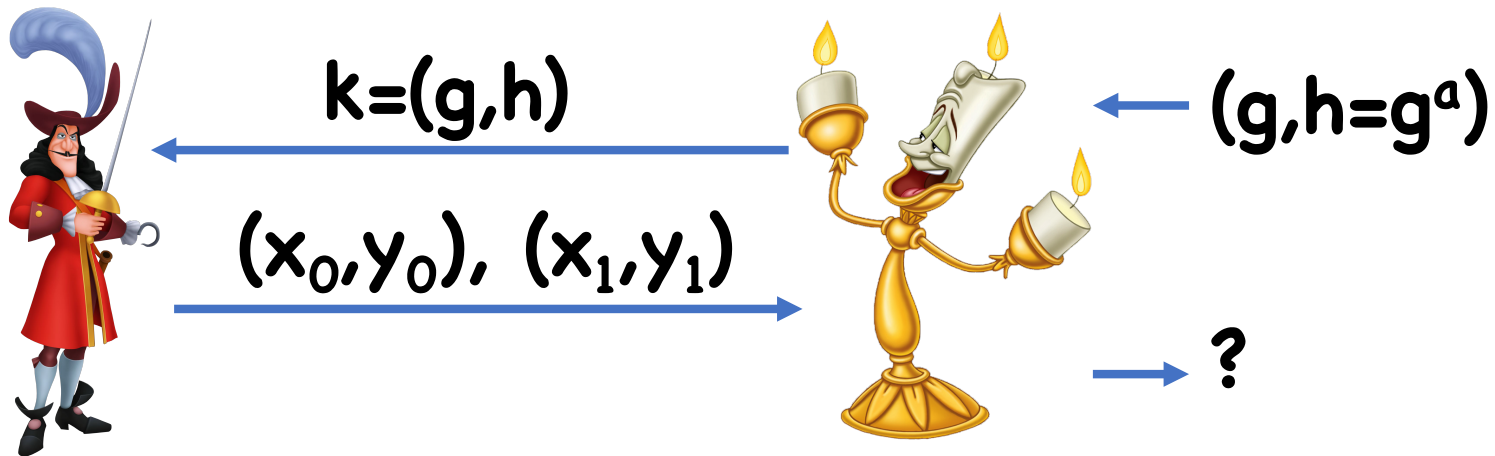
To generate key, choose random  **$g, h \in \mathbb{Z}_p^*$**

- Require **g** a generator

# Collision Resistance from Discrete Log

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

**Theorem:** If  $(p,t,\epsilon)$ -Discrete Log assumption holds, then  $H$  is  $(t-t',\epsilon)$ -collision resistant



# Collision Resistance from Discrete Log

Proof idea:

- Input to **H** is equation for a line **line(a)=ay+x**
- **H(line) = g<sup>line(a)</sup>** (evaluation “in the exponent”)
- A collision is two different lines that intersect at **a**
- Use equations for two lines to solve for **a**:

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

# 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  $p$  be a prime

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

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

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

Algorithm:

- Let  $x_0$  be seed
- For  $i=0, \dots$ 
  - 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

# 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



**$(p, t, \epsilon)$ -Computational Diffie Hellman:** For any algorithm  running in time at most  $t$ ,

**$\Pr[g^{ab} \leftarrow \text{ (g, g^a, g^b):$**   
 **$g \leftarrow \text{generator of } \mathbb{Z}_p^*, a, b \leftarrow \mathbb{Z}_{p-1} ] \leq \epsilon$**

**$(p, t, \epsilon)$ -Decision** The Helman-Hellman problem for any algorithm running in time  $t$

$$|\Pr[1 \leftarrow (g, a, b, g^{ab}): \text{one}] - \Pr[1 \leftarrow (g, a, b, g^c): g^c = g^{ab} \mid a, b, c \leftarrow \mathbb{Z}_{p-1}]| \leq \epsilon$$

# 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  $g_0$  be a generator

Suppose  $p-1 = qr$  for prime  $q$ , integer  $r$

Let  $g = g_0^r$


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

- So  $g$  has “order”  $q$

Let  $G = \{1, g, g^2, \dots\}$  be group “generated by”  $g$



### **$(G, t, \epsilon)$ -Decisional Diffie Hellman:**

For any algorithm  running in time at most  $t$ ,

$$\left| \Pr[1 \leftarrow \text{candlestick} (g, g^a, g^b, g^{ab}): g \leftarrow G, a, b \leftarrow \mathbb{Z}_q] \right. \\ \left. - \Pr[1 \leftarrow \text{candlestick} (g, g^a, g^b, g^c): g \leftarrow G, a, b, c \leftarrow \mathbb{Z}_q] \right| \leq \epsilon$$

# Another PRG

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

Range:  $\mathbf{G}^3$

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

Security almost immediately follows from DDH

# Generalizing “Cryptographic Groups”

Can define Dlog, CDH, DDH over groups other than integers mod  $p$

In many cases, problems turns out easy

Ex:  $\mathbf{G} = \mathbf{Z}_q$ , where  $\mathbf{g} \otimes \mathbf{h} = \mathbf{g} + \mathbf{h} \bmod q$

- What is exponentiation in  $\mathbf{G}$ ?
- What is discrete log in  $\mathbf{G}$ ?

In other groups, conjectured to be hard

# Naor-Reingold PRF

Domain:  $\{0,1\}^n$

Key space:  $\mathbb{Z}_q^{n+1}$

Range:  $\mathbf{G}$

$$F((a, b_1, b_2, \dots, b_n), x) = g^{a b_1^{x_1} b_2^{x_2} \dots b_n^{x_n}}$$

**Theorem:** If the discrete log assumption holds on  $\mathbf{G}$ , then the Naor-Reingold PRF is secure

# Proof by Hybrids

Hybrids **0**:  $H(x) = g^{a \cdot b_1^{x_1} b_2^{x_2} \dots b_n^{x_n}}$

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

- $H_i$  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:

$$\mathbf{x} \rightarrow \mathbf{H}_i(\mathbf{x}) \text{ from } \mathbf{x} \rightarrow \mathbf{H}_{i-1}(\mathbf{x}_{[1,i-1]})^{\mathbf{b}^{\mathbf{x}}}$$

# Proof

Suppose adversary makes  **$2r$**  queries

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

What does the adversary see?

- **$H_i(x)$** :  **$2r$**  random elements in  **$G$**
- **$H_{i-1}(x_{[1,i-1]})^{b_i^{x_i}}$**  :  **$r$**  random elements in  **$G$** ,  **$h_1, \dots, h_q$**   
as well as  **$h_1^b, \dots, h_q^b$**

**Lemma:** Assuming the DDH assumption on  $\mathbf{G}$ , for any polynomial  $r$ , the following distributions are indistinguishable:

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

Suffices to finish proof of NR-PRF



# Proof of Lemma

Hybrids **0**:  $(g, g^{x_1}, g^{b \cdot x_1}, \dots, g^{x_r}, g^{b \cdot x_r})$

Hybrid **i**:

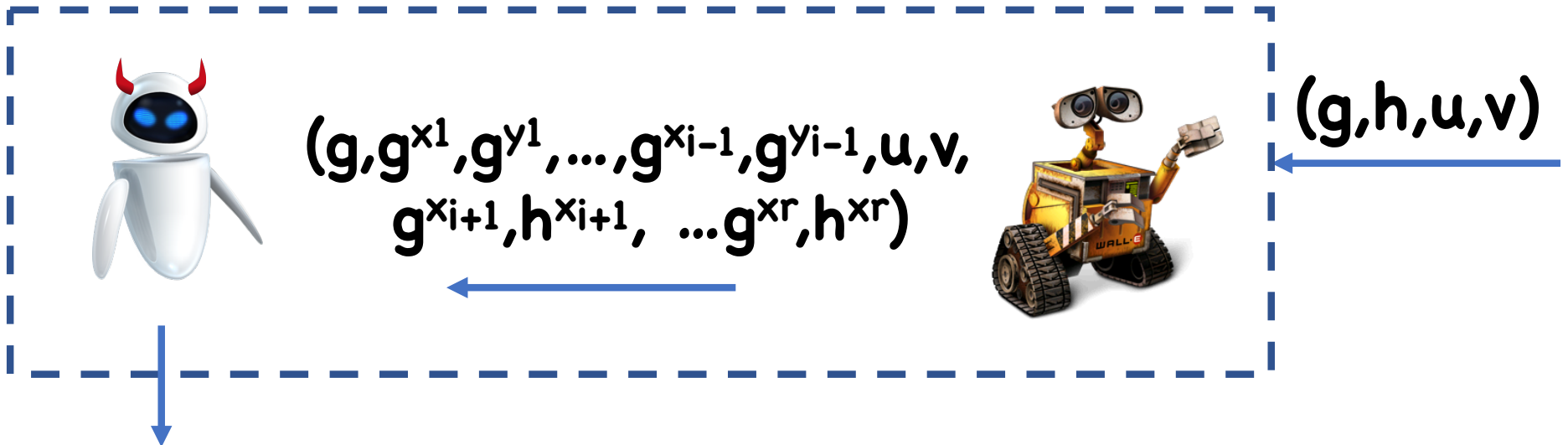
$(g, g^{x_1}, g^{y_1}, \dots, g^{x_i}, g^{y_i}, g^{x_{i+1}}, g^{b \cdot x_{i+1}}, \dots, g^{x_r}, g^{b \cdot x_r})$

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

# Proof of Lemma

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

Use adversary to break DDH:



# Proof of Lemma

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

If  $(g, h, u, v) = (g, g^b, g^{x_i}, g^{b \cdot x_i})$ , 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  **$q \mid p-1$**

- 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**, 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 Riemann Hypothesis:  $O(N^{1/5})$
- Still better heuristic algorithms:  
 $\exp( C (\log N)^{1/3} (\log \log N)^{2/3} )$
- However, all require super-polynomial time in bit-length of **N**

**$(\lambda, t, \epsilon)$ -Factoring Assumption:** For any factoring algorithm  running in time at most  $t$ ,

$\Pr[(p, q) \leftarrow \text{}(N):$   
 $N = pq$  and  $p, q$  random  $\lambda$ -bit primes]  $\leq \epsilon$

Plausible assumption:  $(\lambda, t = 2^{\lambda^{1/3}}, \epsilon = 2^{-\lambda^{1/3}})$



# Sampling Random Primes

**Prime Number Theorem:** A random  $\lambda$ -bit number is prime with probability  $\approx 1/\lambda$

**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  $a \in \{0, \dots, N-1\}$
- Test if  $a^N = a \bmod N$
- Repeat many times

# 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

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

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

# 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 of  $y$ ?
- 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?

# 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:  $\{1,\dots,(N-1)/2\} \times \{0,1\}$

Range:  $\{1,\dots,(N-1)/2\}$

$H((N,y), (x,b))$ : Let  $z = y^b x^2 \bmod N$

- If  $z \in \{1,\dots,(N-1)/2\}$ , output  $z$
- Else, output  $-z \bmod N \in \{1,\dots,(N-1)/2\}$

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

Proof:

- Collision means  $(x_0, b_0) \neq (x_1, b_1)$  s.t.  
$$y^{b_0} x_0^2 = \pm y^{b_1} x_1^2 \pmod{N}$$
- If  $b_0 = b_1$ , then  $x_0 \neq x_1$ , but  $x_0^2 = \pm x_1^2 \pmod{N}$ 
  - $x_0^2 = -x_1^2 \pmod{N}$  not possible. Why?
  - $x_0 \neq -x_1$  since  $x_0, x_1 \in \{1, \dots, (N-1)/2\}$
  - $\text{GCD}(x_0 - x_1, N)$  will give factor
- If  $b_0 \neq b_1$ , then  $(x_0/x_1)^2 = \pm y^{\pm 1} \pmod{N}$ 
  - $(x_0/x_1)$  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 **p** be a prime, and suppose  **$p = 3 \bmod 4$**

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

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

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

# Solving Quadratic Equations

In general, solving quadratic equations is:

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



# Next Time

Powers other than 2 and RSA

# Reminders

HW due tomorrow

Another homework will be out tonight

Remember to work on project!