

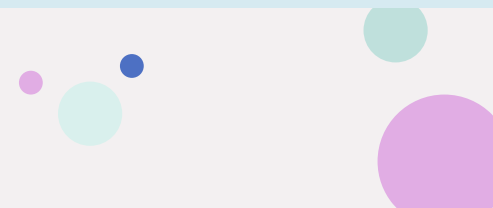


Lecture 12

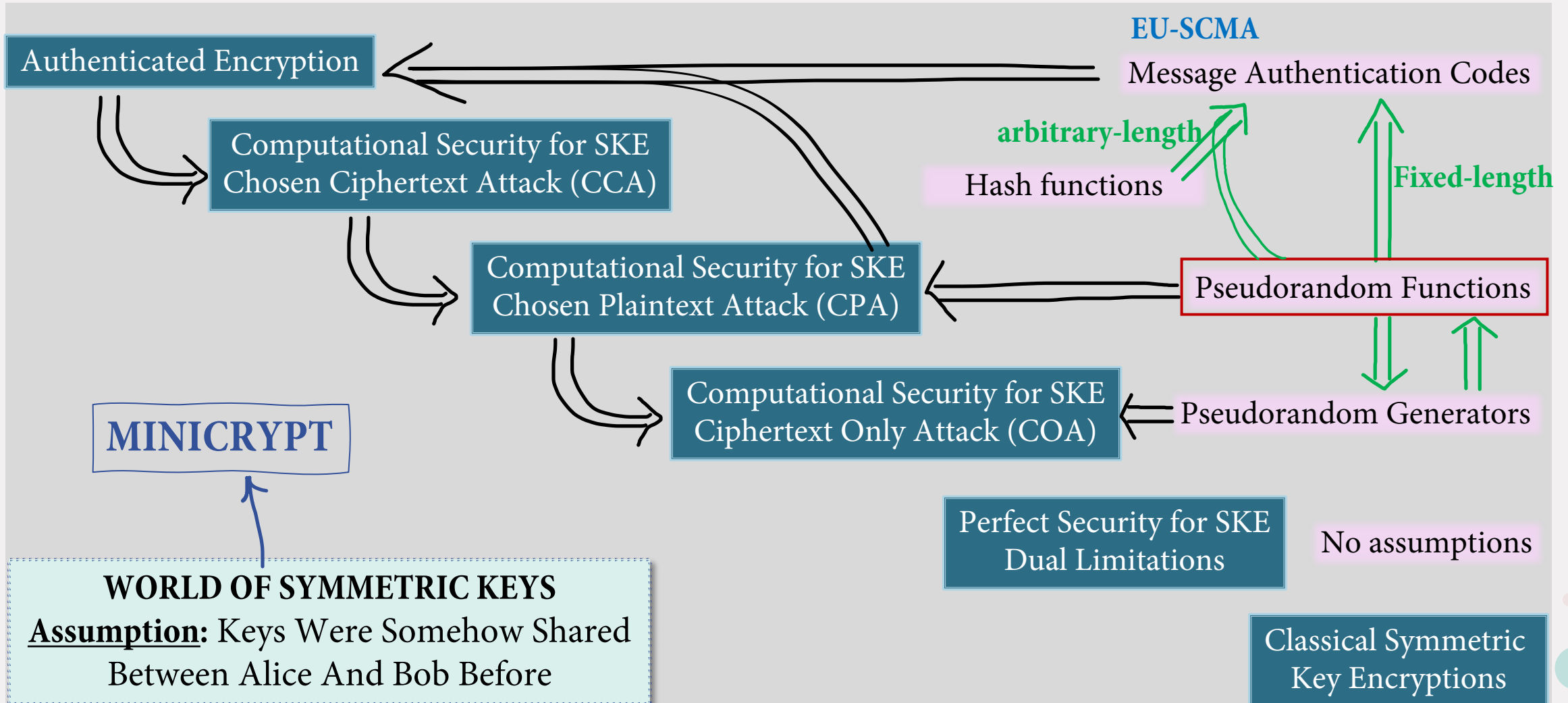
Basic Number Theory, Hardness Assumptions

CS409: Introduction to Cryptography

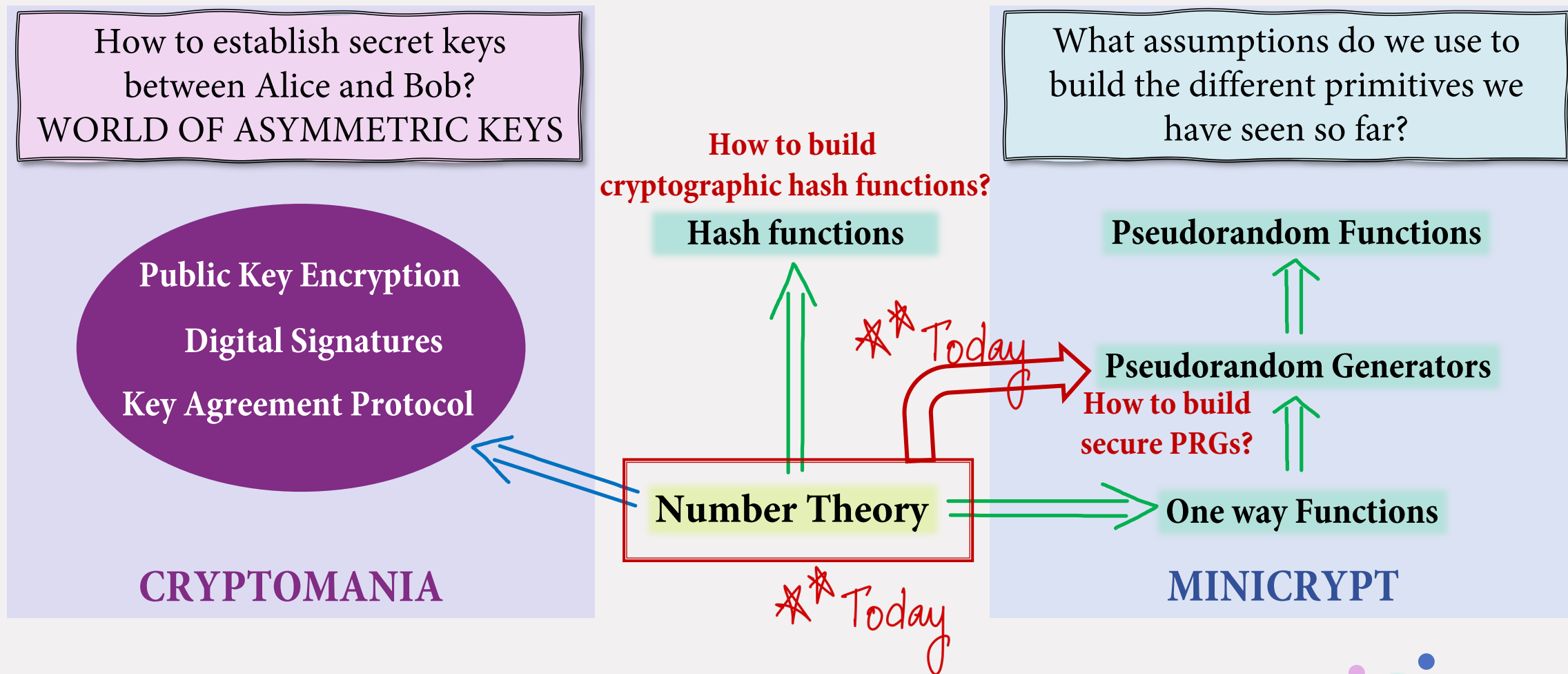
Sruthi Sekar



Journey so far



Journey Ahead



Today

- **Groups and their properties**
- **Basic computations: efficient algorithms**
 - Primes, primality testing, generating random primes
 - Efficient computation on groups
 - Chinese Remainder Theorem
- **Cryptographic Hardness Assumptions**
 - Discrete log problem (Dlog)
 - Quadratic Residues (QR)
 - Diffie-Hellman Assumptions
- **A PRG construction from DDH**

Visual example of Groups



Suppose I give you a solved Rubik's Cube. Imagine every possible way you can scramble it just by turning the faces — no peeling off stickers!

What is the mathematical object that consists of all these possible moves?



Set: all possible moves

Operation: Composition of different moves

Defining Groups

- Group $(\mathbb{G}, *)$ consists of **set** $\mathbb{G}$ and **operation** $* : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}$ that satisfies:
 - Closure Property**: For all $a, b \in \mathbb{G}$, $a * b \in \mathbb{G}$ (denote $*(a, b)$ as $a * b$)
 - Associativity**: For all $a, b, c \in \mathbb{G}$, $a * (b * c) = (a * b) * c$
 - Identity**: There exists an $e \in \mathbb{G}$ such that $a * e = a = e * a, \forall a \in \mathbb{G}$
 - Inverse**: For every $a \in \mathbb{G}$, there exists $a^{-1} \in \mathbb{G}$ such that $a * a^{-1} = e = a^{-1} * a$
 - Commutativity (additional property)**: For all $a, b \in \mathbb{G}$, $a * b = b * a$
- Order** of a group $|\mathbb{G}|$: number of elements in $\mathbb{G}$.

Abelian
groups

IS THIS A GROUP? WHAT IS THE ORDER?

✓ $\mathbb{Z} = (\text{Integers}, +)$

$\mathbb{Z}_n = (\text{Integers modulo } n, + \bmod n)$ ✓

✗ $\mathbb{R} = (\text{Real Numbers}, \times)$

✗ $\mathbb{Z}_n = (\text{Integers modulo } n, \times \bmod n)$

Properties of Groups

Lagrange's Theorem

For any $g \in \mathbb{G}$, $g^{|\mathbb{G}|} = g * g * \dots * g$ ($|\mathbb{G}|$ times) = identity, i.e., $|g|$ divides $|\mathbb{G}|$

Notation: g^i is used to denote $g * g * \dots * g$ (i times), $|g|$, the order of $g \in \mathbb{G}$ is $\min_i (g^i = \text{identity})$

Finite Cyclic Group

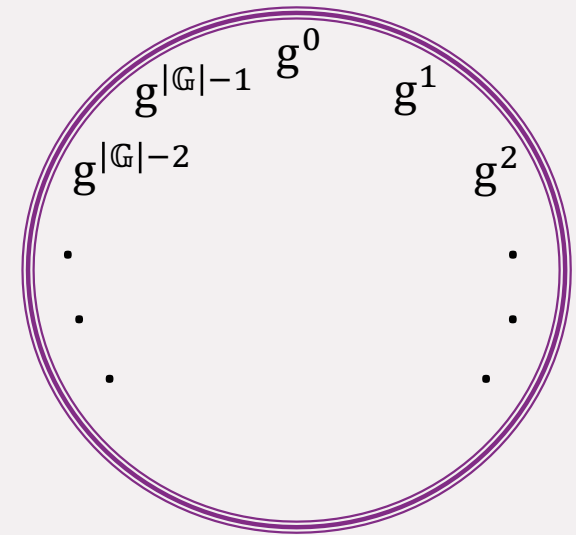
$\exists g \in \mathbb{G}$ such that $\mathbb{G} = \{g^0, g^1, \dots, g^{|\mathbb{G}|-1}\}$. g is called **generator** of the cyclic group.

EXAMPLE:

$\mathbb{Z}_n, + \text{mod } n$: generator?

Theorem: Prime order groups are cyclic

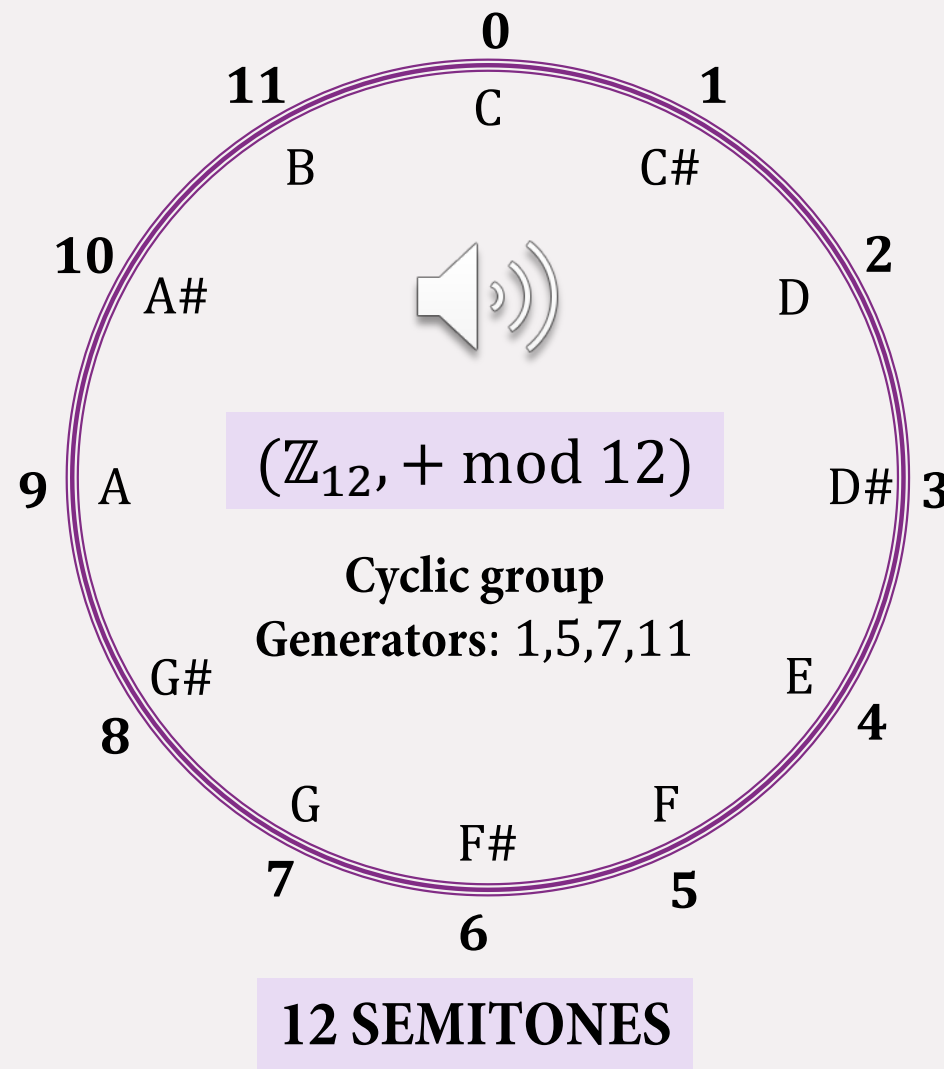
Every group $\mathbb{G}$ with $|\mathbb{G}| = p$, where p is a prime, is cyclic.
Every element other than the identity is a generator of $\mathbb{G}$.



Audible example of Groups

Eg.: +7 semitone

CGDAEBF#C#G#D#A#FC



Multiplicative Group $\mathbb{Z}_n^*$

Greatest common divisor

$\gcd(x, n)$: largest positive integer d that divides both a and b

Multiplicative group of integers mod n

$\mathbb{Z}_n^*$, $\times \bmod n$: set consists of integers $1 \leq a < n$ such that $\gcd(a, n) = 1$

Why only integers with gcd 1?



For an integer a , there exists an integer b such that $a \times b = 1 \bmod n$ if and only if $\gcd(a, n) = 1$

Inverse of $a \in \mathbb{Z}_n^*$: Extended Euclidean Algorithm

The algorithm finds two integers X and Y such that $Xa + Yn = \gcd(a, n)$, in $\text{poly}(\|a\|, \|n\|)$ -time.

** $\|a\|$: number of bits to represent a



How does this give you the inverse?

Order of $\mathbb{Z}_p^*$ for prime p

$p - 1$

Order of $\mathbb{Z}_n^*$

called Euler totient function of n : $\varphi(n)$



Why? For $n = \prod_i p_i^{\alpha_i}$ the prime factorization of n , $\varphi(n) = \prod_i (p_i^{\alpha_i} - p_i^{\alpha_i - 1})$

Multiplicative Group $\mathbb{Z}_n^*$

Greatest common divisor

$\gcd(x, n)$: largest positive integer d that divides both a and b

Multiplicative group of integers mod n

$\mathbb{Z}_n^*$, $\times \bmod n$: set consists of integers $1 \leq a < n$ such that $\gcd(a, n) = 1$

Order of $\mathbb{Z}_n^*$

called **Euler totient function of n : $\varphi(n)$**

For $n = \prod_i p_i^{\alpha_i}$ the prime factorization of n , $\varphi(n) = \prod_i (p_i^{\alpha_i} - p_i^{\alpha_i-1})$



Is $\mathbb{Z}_n^*$ cyclic?

Yes, when $n = p$, for some prime p . Why? Order of the group?
Otherwise, need not be cyclic! Example?

Integer Operations and Modular Arithmetic

Group Operation	Time Complexity	Reference/Algorithm
$a + b$	$O(t)$	high-school algorithm
ab	$O(t \log t)$	Harvey and Van Der Hoeven
$\gcd(a, b)$	$O(t^2)$	Euclidean Algorithm
$a \bmod n$	$O(t^2)$	Division algorithm for integers
$a + b \bmod n, a - b \bmod n$	$O(t^2)$	
$ab \bmod n$	$O(t^2)$	
$a^{-1} \bmod n$	$O(t^2)$	Extended Euclidean algorithm
$a^b \bmod n$	$O(t^3)$	Repeated squaring algorithm
Check if an integer p is prime or not?	$O(t^2 \log 1/\varepsilon)$ $O(t^6)$	Miller-Rabin test (error ε) Agrawal et al. (deterministic)

**** a, b, p be integers of t -bits each.**



Efficient Computation on Groups

Group exponentiation

Given a $g \in \mathbb{G}$ and a non-negative integer b , g^b can be computed using $\text{poly}(\|b\|)$ operations.

Generating a random group element

- Choosing a group element uniformly at random $g \in_R \mathbb{Z}_n$ (randomized algorithm)
 - in time $\text{poly}(\|n\|)$
 - failure probability (not outputting a random value) is negligible $\text{negl}(\|n\|)$(similar algorithm exists for $\mathbb{Z}_n^*$)
- For a general cyclic group $\mathbb{G} = \langle g \rangle$ of order q , to choose uniformly random $h \in_R \mathbb{G}$ do:
 - Pick $x \in_R \mathbb{Z}_q$
 - Compute $h = g^x$(for any finite group $\mathbb{G}$, uniform random elements can be picked for certain representation of $\mathbb{G}$)

Efficient Computation on Groups

Testing and finding generators

Given a cyclic group $\mathbb{G}$ of order q , there exists a $\text{poly}(\|q\|)$ -time algorithm that given q and prime factorization of q

- Given $g \in \mathbb{G}$, decides whether g is a generator or not.
- Outputs a generator of $\mathbb{G}$ except with probability $\text{negl}(\|q\|)$.
(uniformly random from all generators)



How many generators does $\mathbb{G}$ have?

Chinese Remainder Theorem

Group Isomorphism denoted by $\mathbb{G} \equiv \mathbb{H}$

There exists a **bijective map** $\phi: \mathbb{G} \rightarrow \mathbb{H}$ such that $\forall a, b \in \mathbb{G}, \phi(a *_{\mathbb{G}} b) = \phi(a) *_{\mathbb{H}} \phi(b)$

Chinese Remainder Theorem (CRT):

Let $N = pq$ where $p, q > 1$ are relatively prime (i.e., $\gcd(p, q) = 1$), then

(additive group) $\mathbb{Z}_N \equiv \mathbb{Z}_p \times \mathbb{Z}_q$ and $\mathbb{Z}_N^* \equiv \mathbb{Z}_p^* \times \mathbb{Z}_q^*$ **(multiplicative group)**

With the same bijection map

$$\phi(x) := ([x \bmod p], [x \bmod q])$$

In the birthday demo we started with: we needed to efficiently find the inverse of this map.

$$\phi^{-1}((x_1, x_2)) := (c_1 x_1 + c_2 x_2) \bmod N, \text{ where } c_1 = q \cdot (q^{-1} \bmod p), c_2 = p \cdot (p^{-1} \bmod q)$$

**** ϕ^{-1} or ϕ usually need not be efficiently computable, CRT is an example where both are!**

$\mathbb{G} \times \mathbb{H}$ consists of all ordered pairs $(g, h), g \in \mathbb{G}, h \in \mathbb{H}$ with the operation: $(g, h) * (g', h') := (g *_{\mathbb{G}} g', h *_{\mathbb{H}} h')$

 [Katz and Lindell, second edition, Chapter 8] -proof using extended Euclidean algorithm! (also generalizable)

Chinese Remainder Theorem

Where does isomorphism help?

Given $\mathbb{G} \equiv \mathbb{H}$ with ϕ , one can choose to do group computations in the group where it is more efficient!

EXAMPLES

$N = pq$ for distinct primes p and q

CRT tells you that addition, multiplication and group exponentiation **mod N** can be **transformed to analogous operations mod p and mod q** .

1. **Want to compute:** $14 \times 13 \bmod 15$ (in $\mathbb{Z}_{15}^*$) $\phi_{\text{CRT}}(14) = (14 \bmod 5, 14 \bmod 3) = (4, 2)$
 $\phi_{\text{CRT}}(13) = (13 \bmod 5, 13 \bmod 3) = (3, 1)$

$$(4, 2) * (3, 1) = ((4 \times 3 \bmod 5), (2 \times 1 \bmod 3)) = (2, 2) \quad \phi_{\text{CRT}}^{-1}((2, 2)) = 2 = 14 \times 13 \bmod 15$$

2. **Want to compute:** $29^{100} \bmod 35$ $\phi_{\text{CRT}}(29) = (29 \bmod 5, 29 \bmod 7) = (-1 \bmod 5, 1)$

$$(-1 \bmod 5, 1)^{100} = ((-1)^{100} \bmod 5, 1) = (1, 1) \quad \phi_{\text{CRT}}^{-1}((1, 1)) = 1 = 29^{100} \bmod 35$$

Today

- **Groups and their properties**
- **Basic computations: efficient algorithms**
 - Primes, primality testing, generating random primes
 - Efficient computation on groups
 - Chinese Remainder Theorem
- **Cryptographic Hardness Assumptions**
 - Discrete log problem (Dlog)
 - Quadratic Residues (QR)
 - Diffie-Hellman Assumptions
- **A PRG construction from DDH**

Discrete Log Assumption

Cyclic Group $(\mathbb{G}, q, g)$

Group Order Generator

The exponentiation function

$$f_g: \mathbb{Z}_q \rightarrow \mathbb{G}: f_g(x) = g^x$$



Is f_g a group isomorphism?

Yes

inverse

Discrete Log with respect to g

$\text{Dlog}_g(h) := \text{unique } x \in \mathbb{Z}_q \text{ such that } h = g^x, \text{ for each } h \in \mathbb{G}$

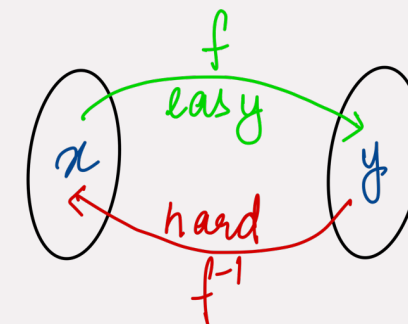


Is f_g invertible efficiently?

Depends on the group!

First cryptographic assumption

In “some groups”, for random $h \in_R \mathbb{G}$ it is “hard” to find $\text{Dlog}_g(h)$



Next class: one-way functions

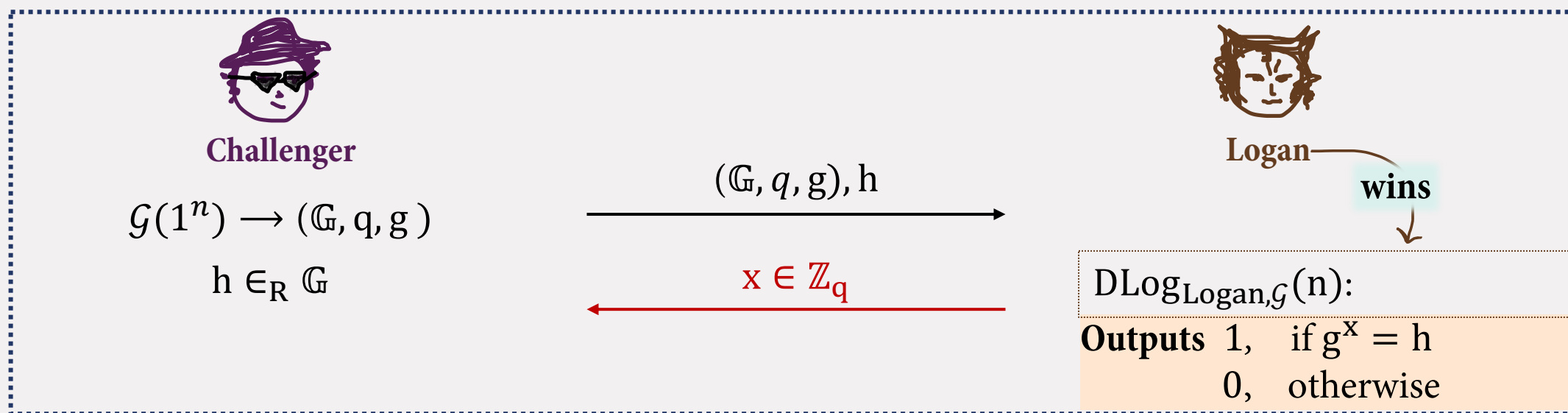
Example: Is it hard in this group?

$$(\mathbb{Z}_N, + \bmod N)$$

$$(\mathbb{Z}_p^*, \times \bmod p)$$

Discrete Log Assumption

Some generic, polytime **group generation algorithm** $\mathcal{G}(1^n) \rightarrow (\mathbb{G}, q, g)$, where $\|q\| = n$ and the description of group specifies how to represent each element as a unique bit-string, such that one can compute efficiently on the elements of $\mathbb{G}$.



Discrete Log Assumption

The **discrete log problem is hard relative to $\mathcal{G}$** if for all **PPT Logan**, there exists a negligible negl such that, $\forall n$

$$\Pr[\text{Dlog}_{\text{Logan}, \mathcal{G}}(n) = 1] \leq \text{negl}(n)$$

Discrete Log Assumption

Why do we believe this is a hard assumption?

Best known algorithm to find discrete log in an **arbitrary group** $\mathbb{G}$ runs in **time** $O(\sqrt{|\mathbb{G}|})$
(Baby-step giant-step algorithm)

Can we do better?

- For $\mathbb{Z}_p^*$, there are better algorithms:

Index Calculus Algorithm ($2^{O(\sqrt{\log y \log \log y})}$),

Number Field Sieve Algorithm ($2^{O((\log y)^{1/3}(\log \log y)^{2/3})}$), where $y \sim \sqrt{|\mathbb{G}|}$

- For finite abelian group with “smooth integer” order:

Pohlig-Hellman Algorithm (if $|\mathbb{G}| = N = \prod_i p_i^{\alpha_i}$, runs in time $O(\sum_i \alpha_i (\log N + \sqrt{p_i}))$)

Moral of the story: Yes, you can do better than the naive algorithm, but still the time complexity is exponential for arbitrary groups, and sub-exponential for certain specific groups.

Quadratic Residues

Quadratic Residue in $\mathbb{Z}_p^*$ for odd prime p

$h \in \mathbb{Z}_p^*$ is called **quadratic residue mod p** if there exists a $g \in \mathbb{Z}_p^*$ such that $h = g^2 \bmod p$

$h \in \mathbb{Z}_p^*$ is called **quadratic non-residue mod p** if for all $g \in \mathbb{Z}_p^*$, $h \neq g^2 \bmod p$



Is checking if an $h \in \mathbb{Z}_p^*$ is a quadratic residue mod p or not, an efficient check?

Euler's Criterion

$h \in \mathbb{Z}_p^*$ is a **quadratic residue mod p** if and only if $h^{\frac{p-1}{2}} = 1 \bmod p$



Is checking if an $h \in \mathbb{Z}_N^*$ is a quadratic residue mod N or not, for some composite N , an efficient check?

Quadratic Residuosity (QR) Assumption

No, it is not an efficient check, for some “special N ” whose factorization is unknown!

Diffie-Hellman Problems

Cyclic Group $(\mathbb{G}, q, g)$

Group Order Generator

Diffie-Hellman Function

$$\text{DH}_g(h_1, h_2) = g^{\log_g h_1 \cdot \log_g h_2}$$

Computational Diffie-Hellman (CDH) Problem

For randomly picked $h_1, h_2 \in_R \mathbb{G}$ compute $\text{DH}_g(h_1, h_2)$

If Dlog easy relative to some $\mathcal{G} \Rightarrow \text{CDH}$ problem is easy too! $\Rightarrow \text{DDH}$ problem is easy too!



Is CDH problem hard? Depends on the group!



Is DDH problem hard? Depends on the group!

Decisional Diffie-Hellman (DDH) Problem

For randomly picked $h_1, h_2 \in_R \mathbb{G}$ distinguish $\text{DH}_g(h_1, h_2)$ from a uniform group element.

CDH and DDH Assumptions



Challenger

$$\mathcal{G}(1^n) \rightarrow (\mathbb{G}, q, g)$$

$$x, y \in_R \mathbb{Z}_q$$

CDH

$$(\mathbb{G}, q, g), g^x, g^y$$

h



Logan

wins

$$\text{CDH}_{\text{Logan}, \mathcal{G}}(n)$$

o/p 1, if $h = g^{xy}$ and 0, o.w.

CDH Assumption

The **CDH problem is hard relative to $\mathcal{G}$** if for all **PPT Logan**, there exists a negligible negl such that, $\forall n$

$$\Pr[\text{CDH}_{\text{Logan}, \mathcal{G}}(n) = 1] \leq \text{negl}(n)$$

DDH Assumption

$$\mathcal{G}(1^n) \rightarrow (\mathbb{G}, q, g), x, y, z \in_R \mathbb{Z}_q$$

$$(\mathbb{G}, q, g), g^x, g^y, g^{xy} \approx_c (\mathbb{G}, q, g), g^x, g^y, g^z$$

DDH problem easy in $\mathbb{Z}_p^*$

(DDH easy $\nRightarrow$ Dlog easy)

$$\mathcal{G}(1^n) \rightarrow (\mathbb{Z}_p^*, p-1, g), \quad x, y, z \in_R \mathbb{Z}_{p-1}$$

$$(\mathbb{Z}_p^*, p-1, g), g^x, g^y, g^{xy} \approx_c (\mathbb{Z}_p^*, p-1, g), g^x, g^y, g^z$$

prove it

This world has a property:

$\rightarrow g^{xy}$ is a quadratic residue mod $p \iff$ either g^x or g^y (or both) is a quadratic residue mod p

Attack on DDH:

Given: $(\mathbb{Z}_p^*, p-1, g), g^x, g^y, g^z$ (where $z = xy$ or $z \in_R \mathbb{Z}_{p-1}$)

Step I: Check if g^x, g^y and g^z are **quadratic residues or not?** (recall this is efficient in $\mathbb{Z}_p^*$!)

Step II: What is $\Pr[g^z \text{ is a quadratic residue mod } p \text{ for } z \in_R \mathbb{Z}_{p-1}]$?

Step III: If $z = xy$ then it can be decided with **probability 1** whether g^z is a QR or not.

Hence, one can distinguish the two worlds of DDH efficiently!

Solution: There are some subgroups of $\mathbb{Z}_p^*$ of prime order where DDH is assumed to be hard!



Today

- **Groups and their properties**
 - **Basic computations: efficient algorithms**
 - Primes, primality testing, generating random primes
 - Efficient computation on groups
 - Chinese Remainder Theorem
 - **Cryptographic Hardness Assumptions**
 - Discrete log problem (Dlog)
 - Quadratic Residues (QR)
 - Diffie-Hellman Assumptions
- **A PRG construction from DDH**

A PRG from DDH

DDH Assumption

$$\mathcal{G}(1^n) \rightarrow (\mathbb{G}, q, g), \quad x, y, z \in_R \mathbb{Z}_q$$

$$(\mathbb{G}, q, g), g^x, g^y, g^{xy} \approx_c (\mathbb{G}, q, g), g^x, g^y, g^z$$

PRG Candidate:

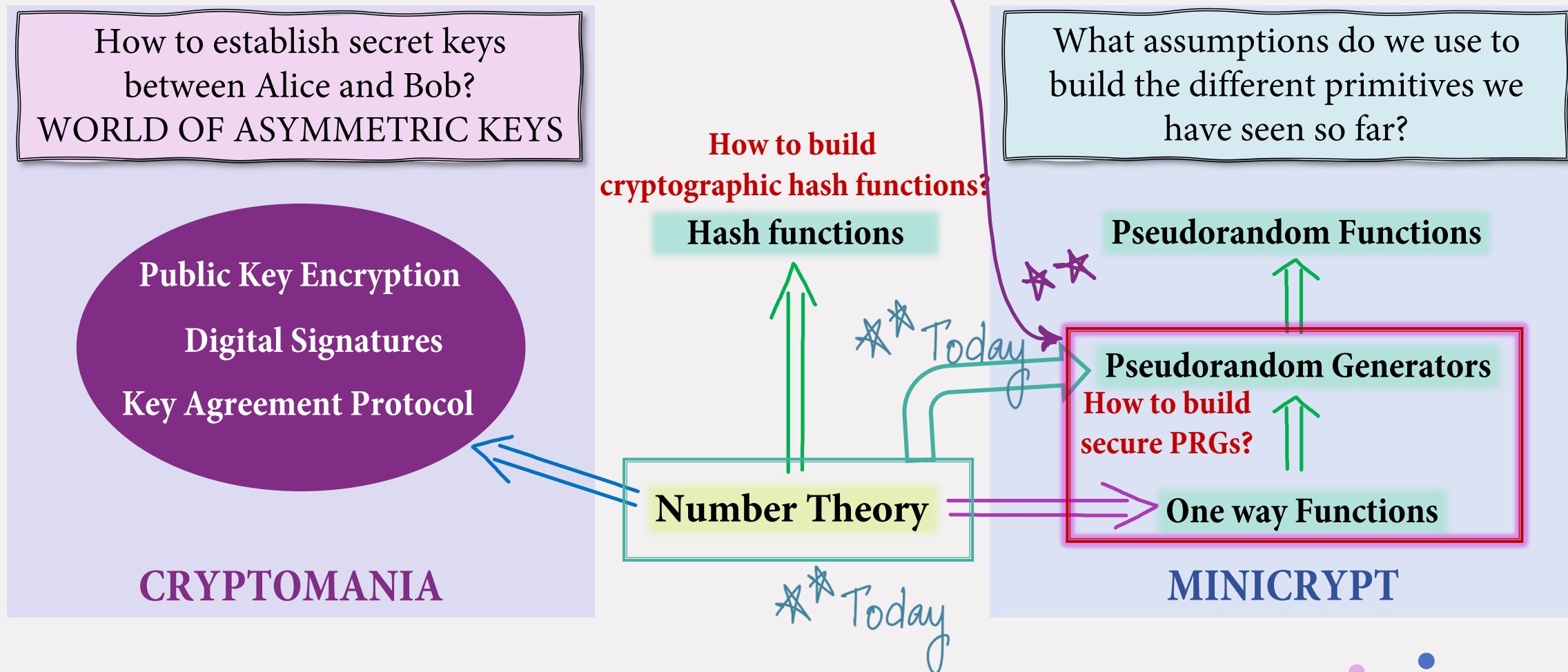
$$G_{q,g}(x, y) = (g^x, g^y, g^{xy})$$

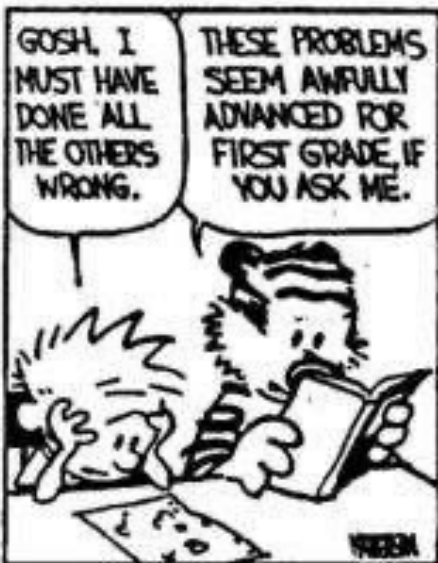
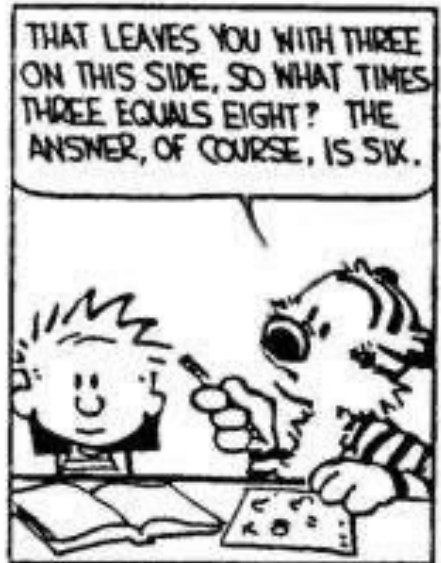
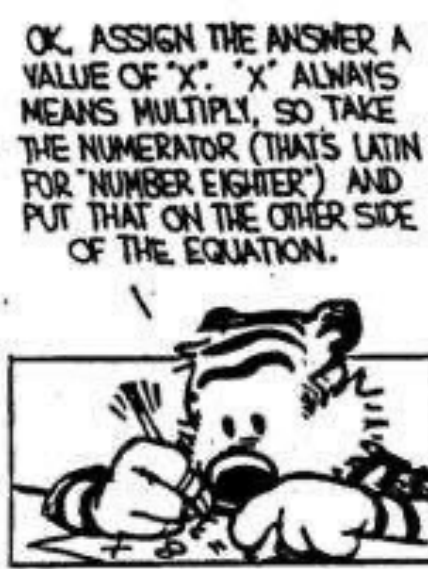
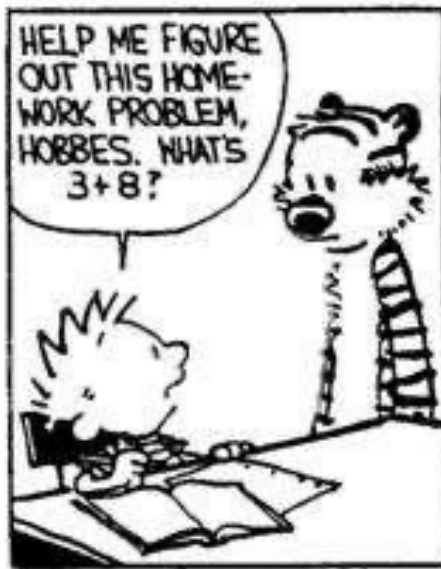
There is now a family of PRG functions above, indexed by (q, g) , such that when a random function is chosen from the family, then this function is a pseudorandom generator!

Theorem: If DDH assumption holds relative to $\mathcal{G}$ then, $\{G_{q,g}\}_{q,g}$ is a family of PRGs

Try to prove by reduction!

Next Class





THANK YOU

Quiz 2: 8th Oct, 8:30-9:30 AM
Quiz 3: 29th Oct, 8:30-9:30 AM
@CC103