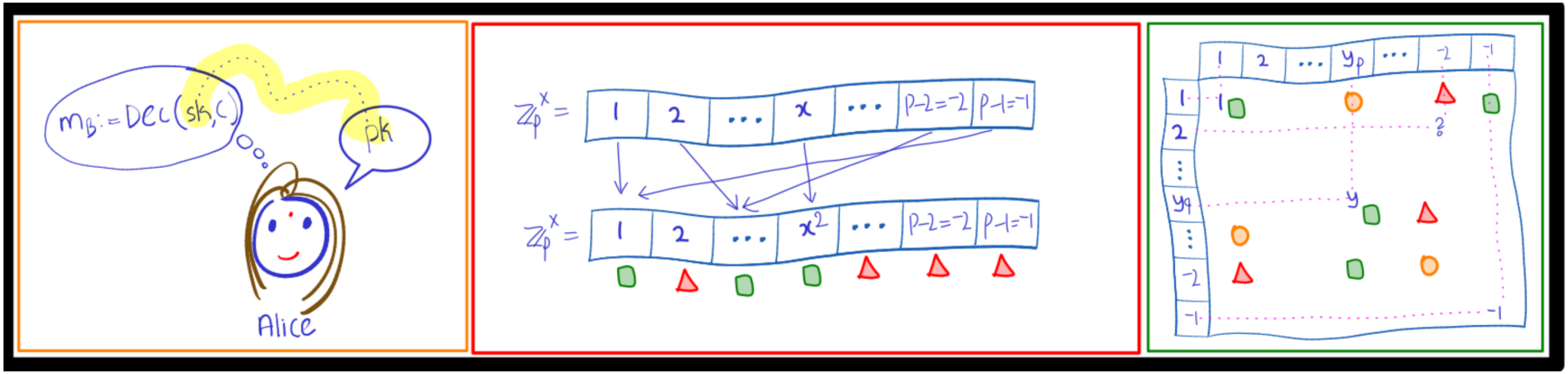
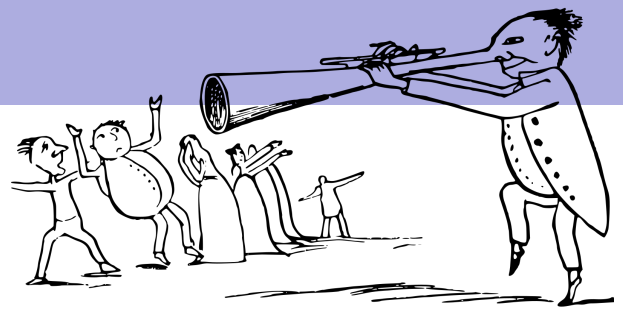


CS409m: Introduction to Cryptography

Lecture 14 (01/Oct/25)

Instructor: Chethan Kamath

Announcements



- Quiz 2 moved to 10/Oct (next Friday), 08:25-09:25, in CC103
- ⚠ Bounty on Problem 7.3 still on!
  - Come up with a *simple* construction of MAC from weak PRF
  - Construction provided in solution set is too complex!

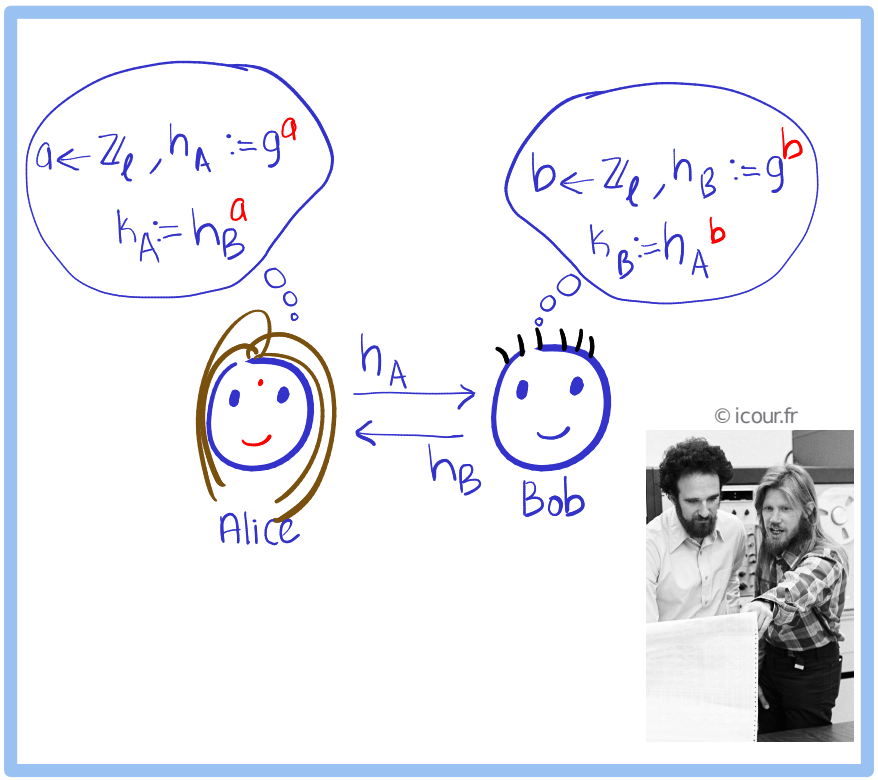
Recall from Last Two Lectures...

- Tasks: key exchange (KEx) and public-key encryption (PKE)
- 2-Round KEx $\Leftrightarrow$ PKE

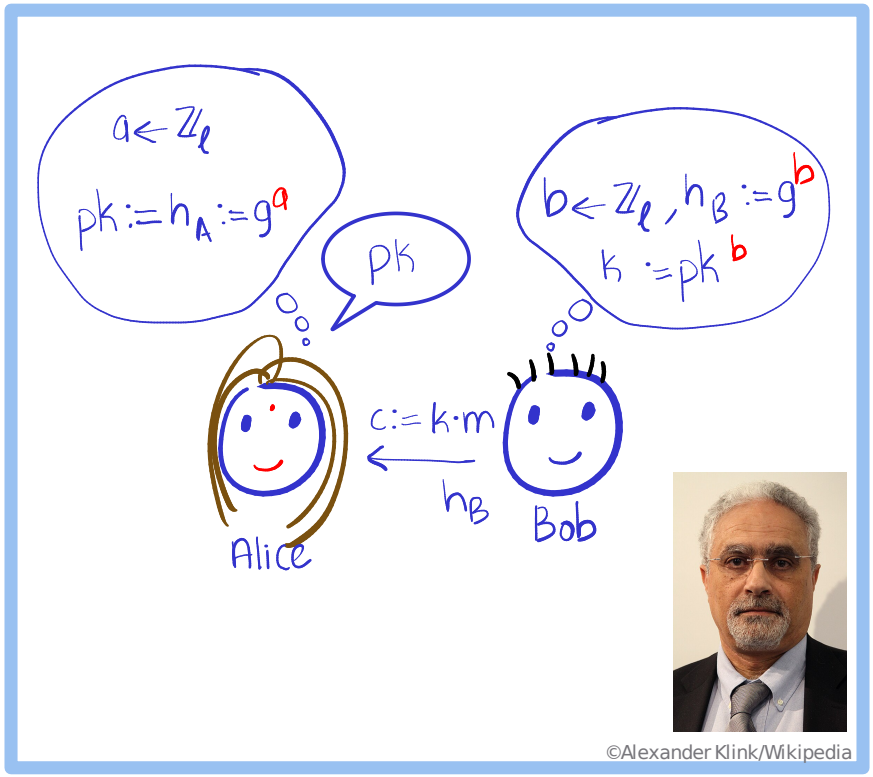
Recall from Last Two Lectures...

- Tasks: key exchange (KEx) and public-key encryption (PKE)
- 2-Round KEx $\Leftrightarrow$ PKE

Diffie-Hellman KEx



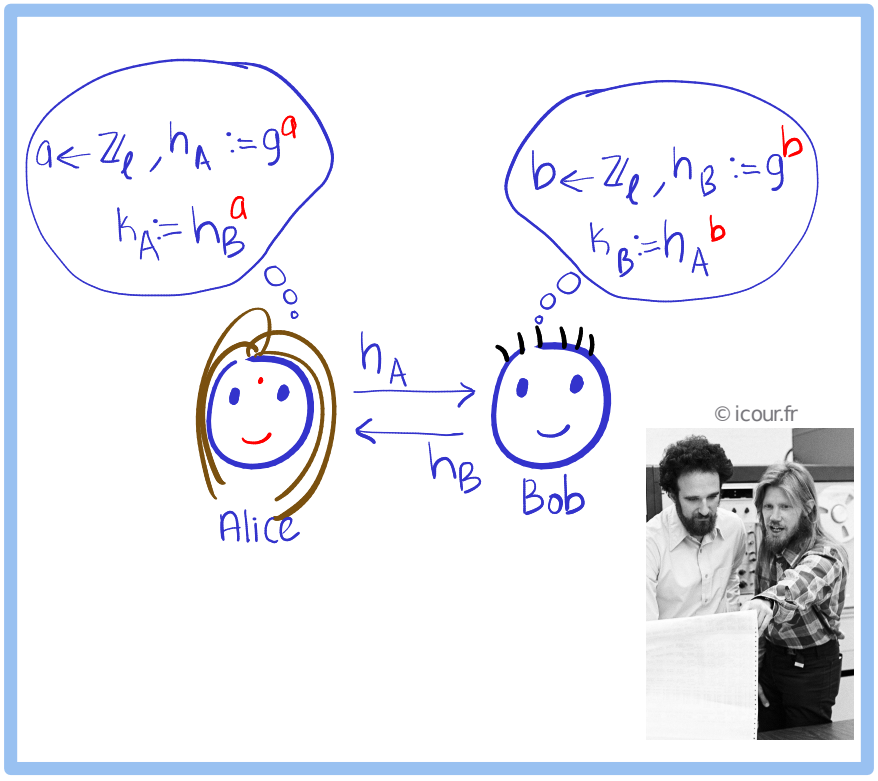
Elgamal PKE



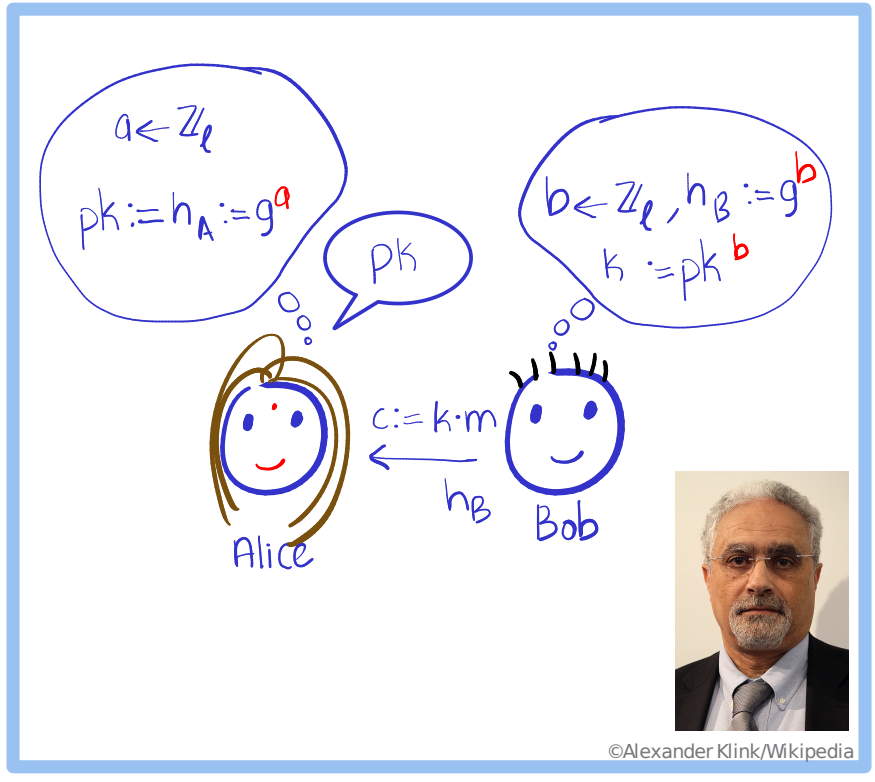
Recall from Last Two Lectures...

- Tasks: key exchange (KEx) and public-key encryption (PKE)
- 2-Round KEx $\Leftrightarrow$ PKE

Diffie-Hellman KEx



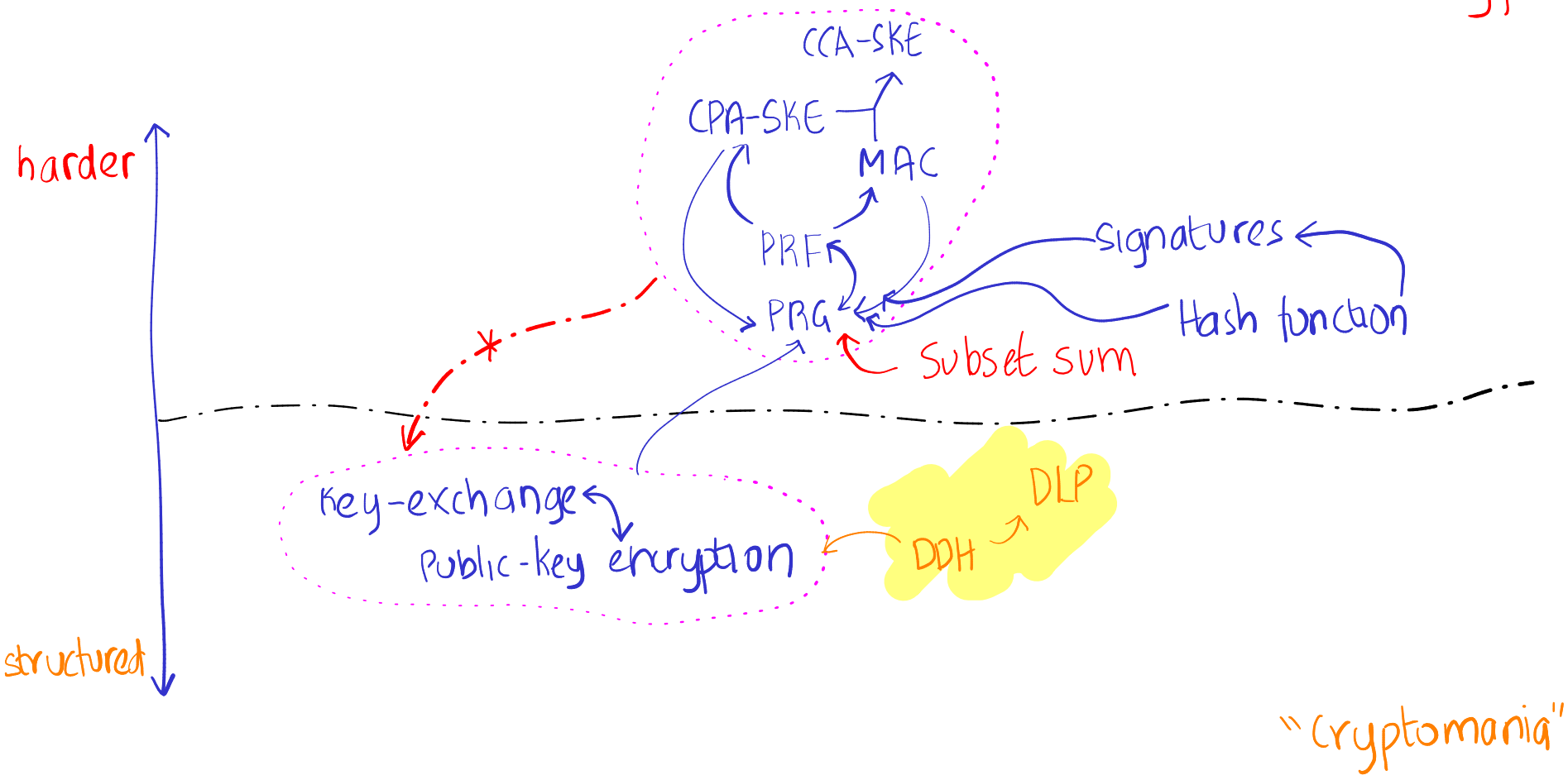
Elgamal PKE



Structure we exploited: $(g^a)^b = g^{ab} = (g^b)^a$

Recall from Last Two Lectures...

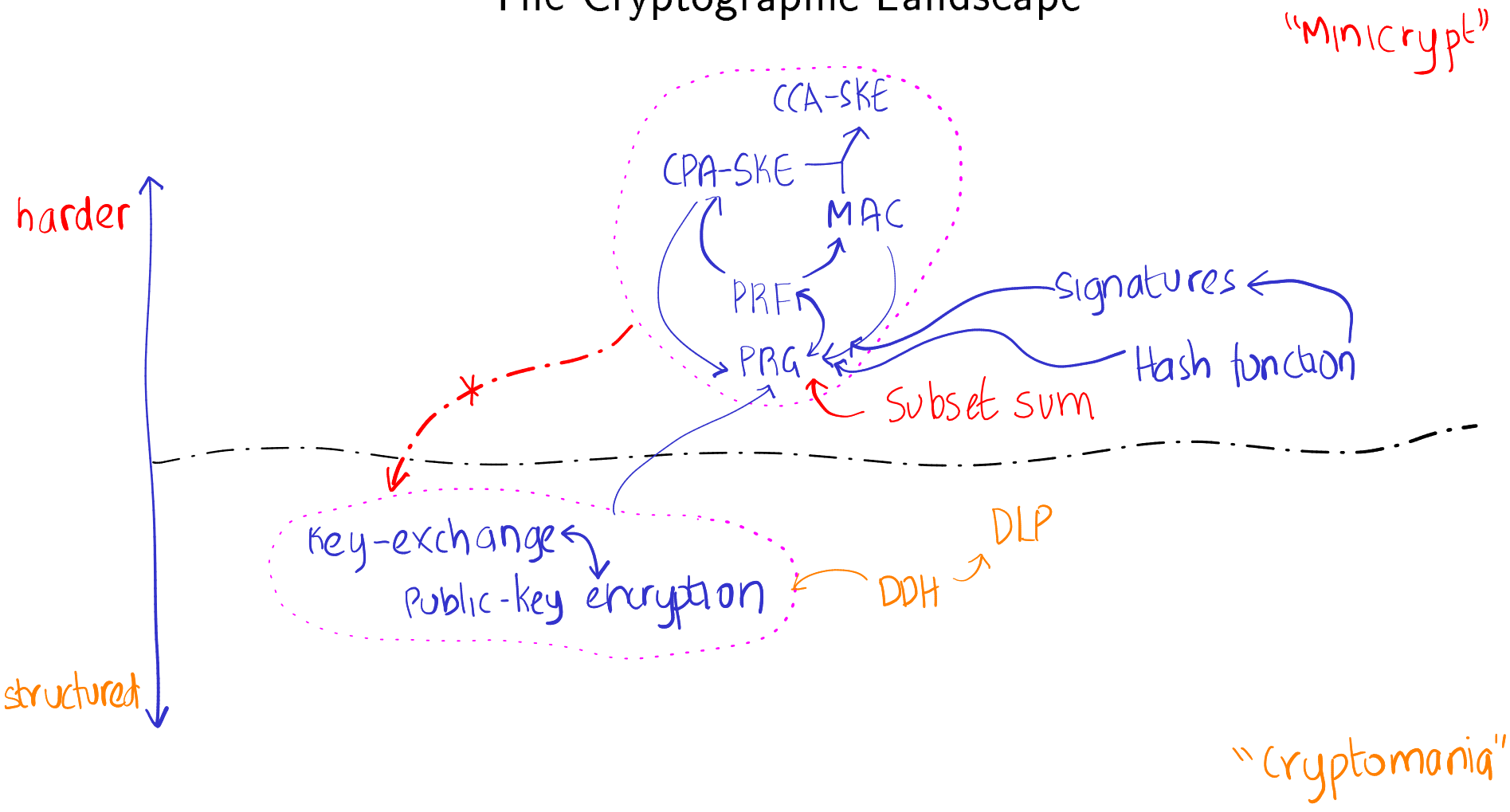
The Cryptographic Landscape



Group-based hard problems: DLP, CDH and DDH

Plan for Today's Lecture...

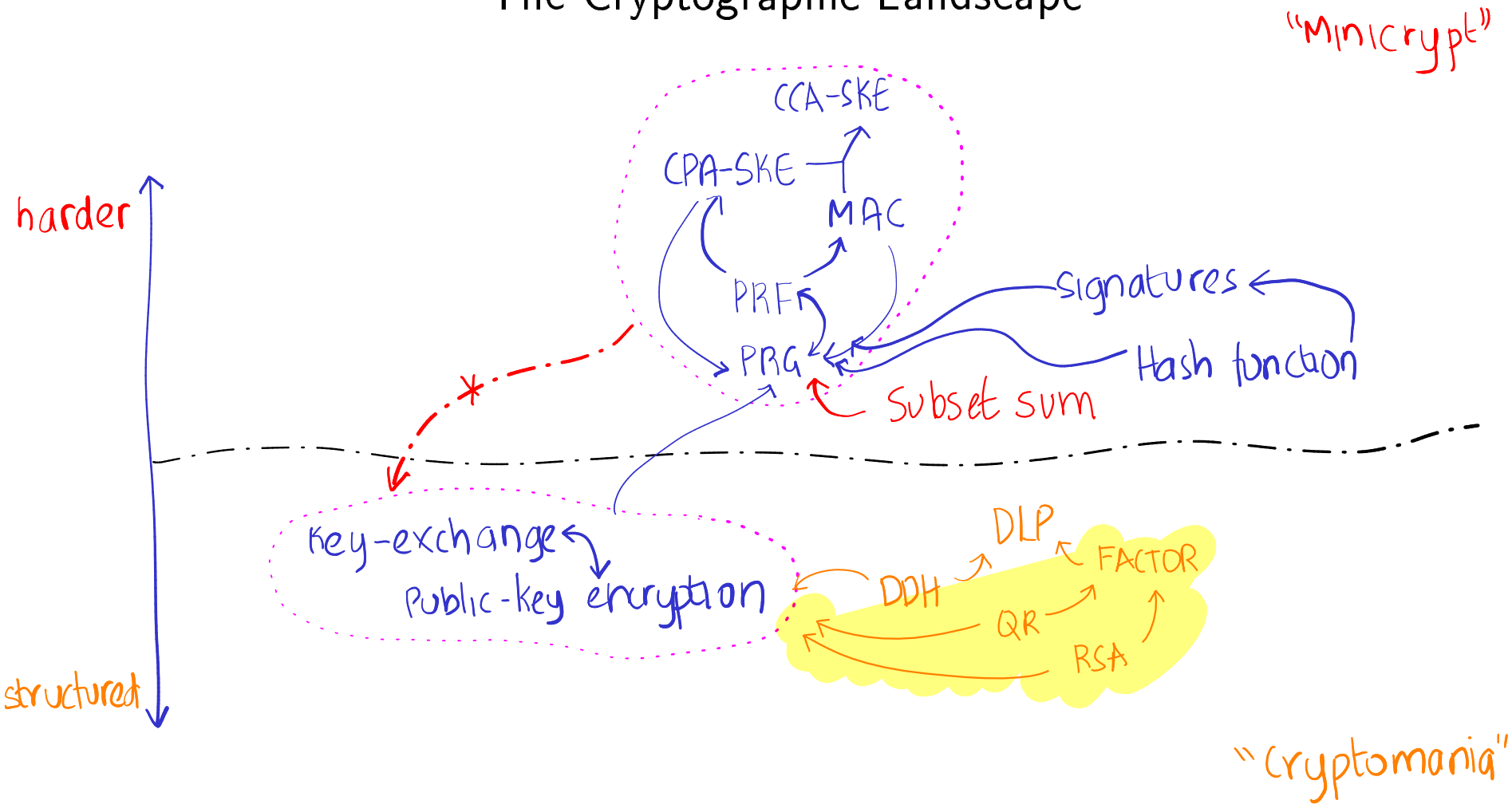
The Cryptographic Landscape



Today: Integer Factoring and its friends!

Plan for Today's Lecture...

The Cryptographic Landscape

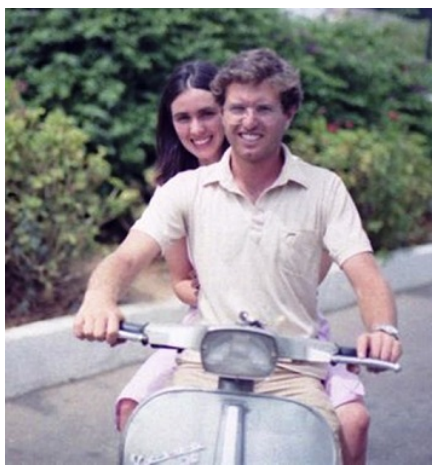


Today: Integer Factoring and its friends!

Plan for Today's Lecture...

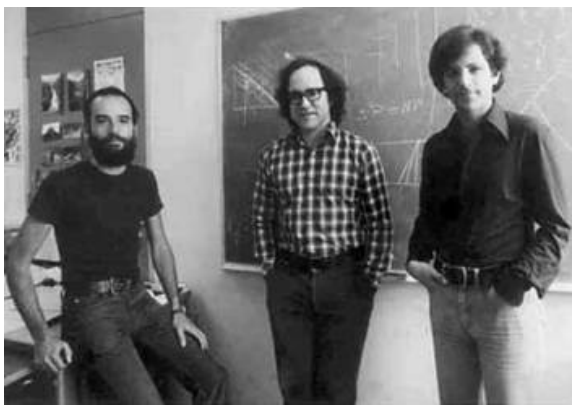
- Task: public-key encryption (PKE)
- Threat model: IND-CPA

Goldwasser-Micali PKE



©Oded Goldreich

RSA PKE

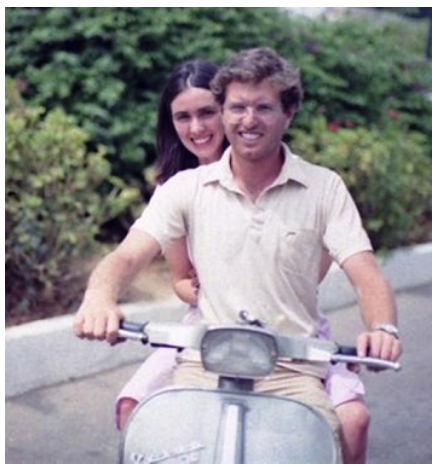


©cs.miami.edu (Rosenberg)

Plan for Today's Lecture

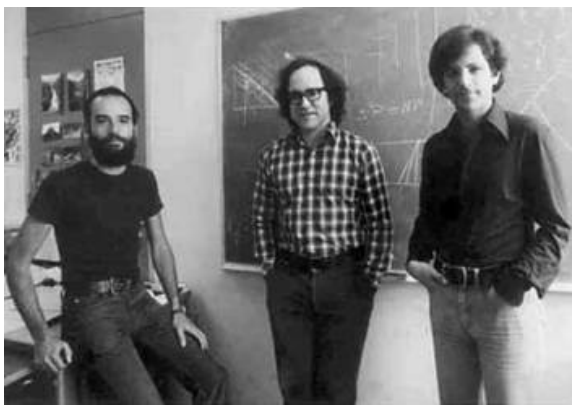
- Task: public-key encryption (PKE)
- Threat model: IND-CPA

Goldwasser-Micali PKE



©Oded Goldreich

RSA PKE



©cs.miami.edu (Rosenberg)

Algebraic setting: multiplication modulo *semiprime* (RSA group)

Plan for Today's Lecture

- Task: public-key encryption (PKE)
- Threat model: IND-CPA

Goldwasser-Micali PKE

RSA PKE

©Oded Goldreich

©cs.miami.edu (Rosenberg)

 Algebraic setting: multiplication modulo *semiprime* (RSA group) 

Integer Factoring...

- Problem: Given a integer N, find $1 < p < N$ that divides N

Integer Factoring...

- Problem: Given a integer N, find $1 < p < N$ that divides N

Can you think of an algorithm that takes $\sqrt{N}$ steps? 

Integer Factoring...

- Problem: Given a integer N, find $1 < p < N$ that divides N

Can you think of an algorithm that takes $\sqrt{N}$ steps? 

Pseudocode 1

NaiveFactor(N):

1. For $1 \leq i \leq \lceil \sqrt{N} \rceil$:
   - If $i^2 = N$ then output i

Integer Factoring...

- Problem: Given a integer N, find $1 < p < N$ that divides N

Can you think of an algorithm that takes $\sqrt{N}$ steps? 

Pseudocode 1

NaiveFactor(N):

1. For $1 \leq i \leq \left\lceil \sqrt{N} \right\rceil$:
   - If $i^2 = N$ then output i
2. Output "Prime!"

- Let's try to sample hard-to-factor integer N

What about a *random* integer N?

Integer Factoring...

- Problem: Given a integer N, find $1 < p < N$ that divides N

Can you think of an algorithm that takes $\sqrt{N}$ steps? 

Pseudocode 1

NaiveFactor(N):

1. For $1 \leq i \leq \lceil \sqrt{N} \rceil$:
   - If $i^2 = N$ then output i
2. Output "Prime!"

- Let's try to sample hard-to-factor integer N
  - What about a *random* integer N? N even with probability $1/2$ 
  - What about a random *odd* integer N?

Integer Factoring...

- Problem: Given a integer N, find $1 < p < N$ that divides N

Can you think of an algorithm that takes $\sqrt{N}$ steps? 

Pseudocode 1

NaiveFactor(N):

1. For $1 \leq i \leq \lceil \sqrt{N} \rceil$:
   - If $i^2 = N$ then output i
2. Output "Prime!"

- Let's try to sample hard-to-factor integer N
  - What about a *random* integer N? N even with probability $1/2$ 
  - What about a random *odd* integer N? N divisible by 3 with probability $1/3$ 
  - What seems hardest?

Integer Factoring...

- Problem: Given a integer N, find $1 < p < N$ that divides N

Can you think of an algorithm that takes $\sqrt{N}$ steps? 

Pseudocode 1

NaiveFactor(N):

1. For $1 \leq i \leq \lceil \sqrt{N} \rceil$:
   - If $i^2 = N$ then output i
2. Output "Prime!"

- Let's try to sample hard-to-factor integer N
  - What about a *random* integer N? N even with probability $1/2$ 
  - What about a random *odd* integer N? N divisible by 3 with probability $1/3$ 
  - What seems hardest? *Semiprime*, i.e., N product of two primes

Integer Factoring...

Pseudocode 2

Semiprime sampler $S(1^n)$:

- Sample two random primes p and q of length $\approx n$
- Output $N := pq$

Integer Factoring...

Pseudocode 2

Semiprime sampler $\mathsf{S}(1^n)$:

- Sample two random primes p and q of length $\approx n$
- Output $N := pq$

Assumption 1 (Factoring assumption w.r.to S . . .)

. . . holds if for all PPT A, the following is negligible:

$$\left|\delta(n) := \Pr_{N \leftarrow \mathsf{S}(1^n)}[\mathsf{A}(N) \text{ divides } N]\right|$$

Integer Factoring...

Pseudocode 2

Semiprime sampler $\mathsf{S}(1^n)$:

- Sample two random primes p and q of length $\approx n$
- Output $N := pq$

Assumption 1 (Factoring assumption w.r.to S . . .)

. . . holds if for all PPT A, the following is negligible:

$$\left|\delta(n) := \Pr_{N \leftarrow \mathsf{S}(1^n)}[\mathsf{A}(N) \text{ divides } N]\right|$$


Best known algorithm (Number-Field Sieve) requires $\approx 2^{|N|^{\frac{1}{3}}}$ time


Assumption does not hold against *quantum* adversaries!

- Shor's algorithm factors in *polynomial time* on quantum computer

Integer Factoring in the Wild!

WIKIPEDIA
The Free Encyclopedia

Donate Create account Log in

RSA Factoring Challenge

6 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

The **RSA Factoring Challenge** was a challenge put forward by RSA Laboratories on March 18, 1991[1] to encourage research into computational number theory and the practical difficulty of factoring large integers and cracking RSA keys used in cryptography. They published a list of semiprimes (numbers with exactly two prime factors) known as the RSA numbers, with a cash prize for the successful factorization of some of them. The smallest of them, a 100-decimal digit number called RSA-100 was factored by April 1, 1991. Many of the bigger numbers have still not been factored and are expected to remain unfactored for quite some time, however advances in quantum computers make this prediction uncertain due to Shor's algorithm.

RSA250[b]	250	829		Feb 28, 2020[16]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann
RSA260	260	862			
RSA270	270	895			
RSA896	270	896	US$75,000[d]		

Integer Factoring in the Wild!

WIKIPEDIA
The Free Encyclopedia

Donate Create account Log in

RSA Factoring Challenge

6 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

The **RSA Factoring Challenge** was a challenge put forward by RSA Laboratories on March 18, 1991[1] to encourage research into computational number theory and the practical difficulty of factoring large integers and cracking RSA keys used in cryptography. They published a list of semiprimes (numbers with exactly two prime factors) known as the RSA numbers, with a cash prize for the successful factorization of some of them. The smallest of them, a 100-decimal digit number called RSA-100 was factored by April 1, 1991. Many of the bigger numbers have still not been factored and are expected to remain unfactored for quite some time, however advances in quantum computers make this prediction uncertain due to Shor's algorithm.

RSA250[b]	250	829		Feb 28, 2020[16]	F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann
RSA260	260	862			
RSA270	270	895			
RSA896	270	896	US$75,000[d]		
RSA617	617	2048			
RSA2048	617	2048	US$200,000[d]		

Plan for Today's Lecture

- Task: public-key encryption (PKE)
- Threat model: IND-CPA

Goldwasser-Micali PKE

RSA PKE

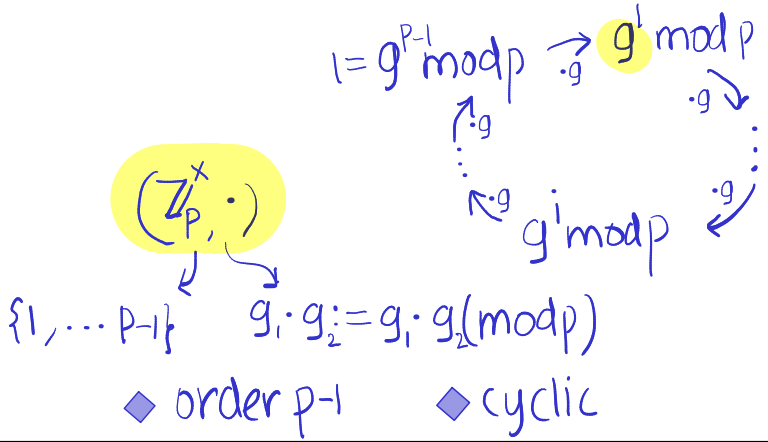
©Oded Goldreich

©cs.miami.edu (Rosenberg)

Algebraic setting: multiplication modulo *semiprime* (RSA group)

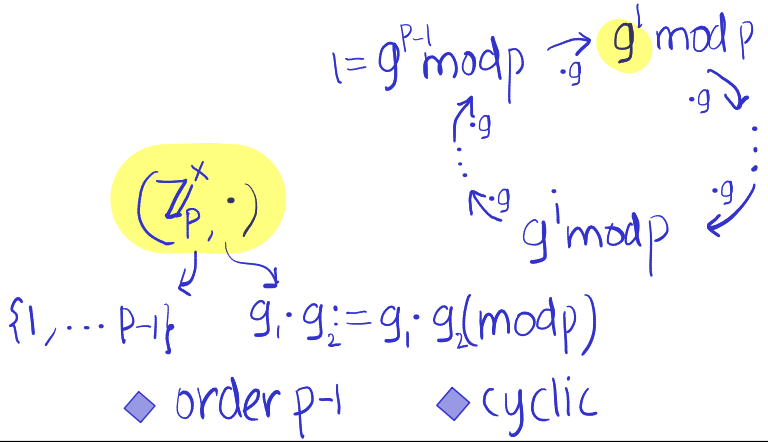
Multiplication Modulo Semiprime $N = pq$

Multiplication mod prime p



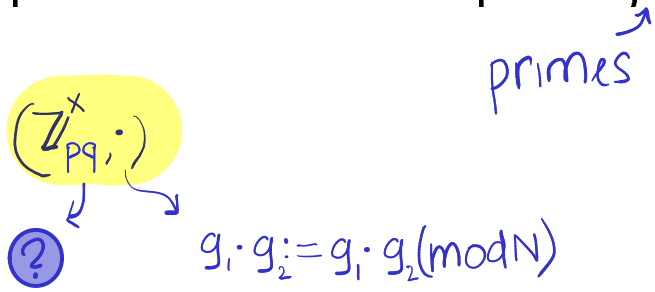
Multiplication Modulo Semiprime $N = pq$

Multiplication mod prime p



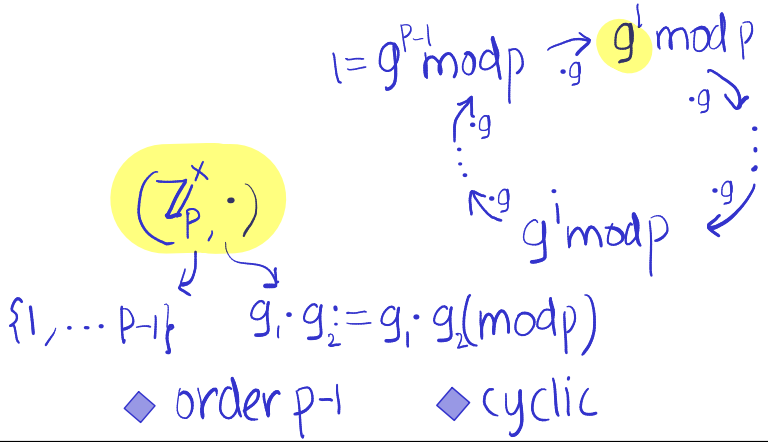
- order $p-1$
- cyclic

Multiplication mod semiprime pq

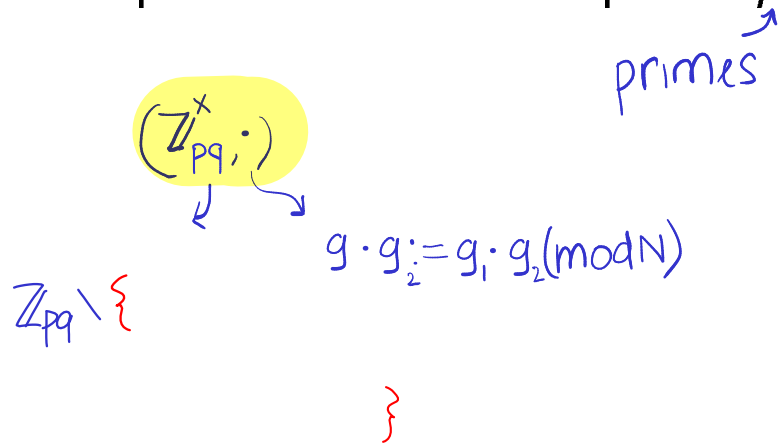


Multiplication Modulo Semiprime $N = pq$

Multiplication mod prime p



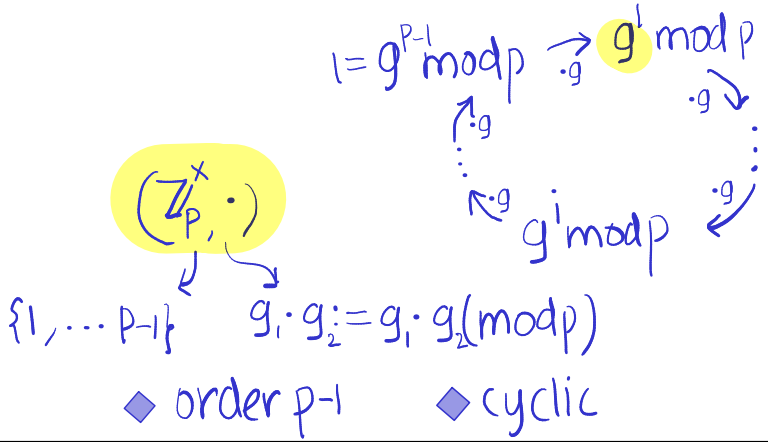
Multiplication mod semiprime pq



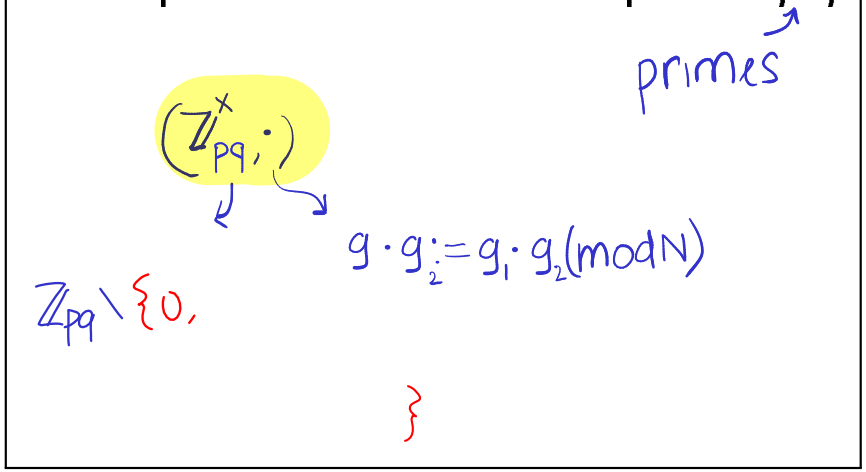
? What are the elements in $\mathbb{Z}_{pq}^\times$? Every $0 \leq a < N$ that is invertible

Multiplication Modulo Semiprime $N = pq$

Multiplication mod prime p

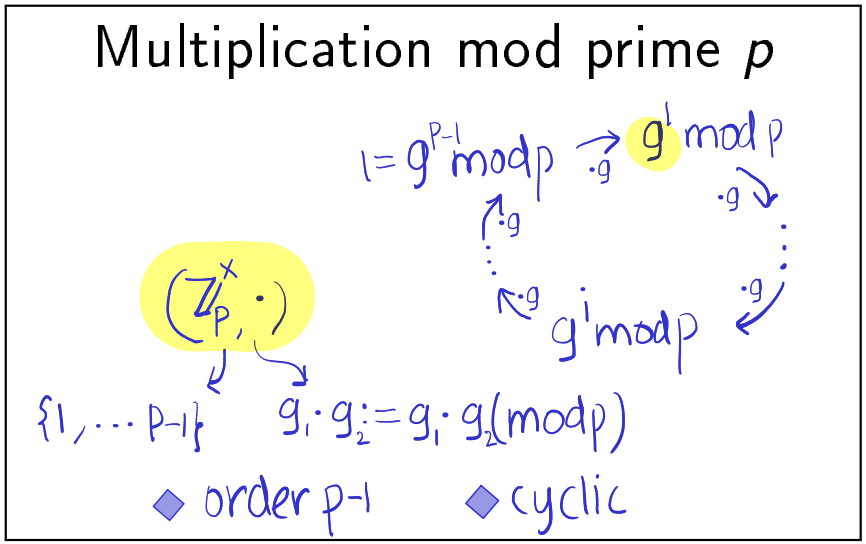


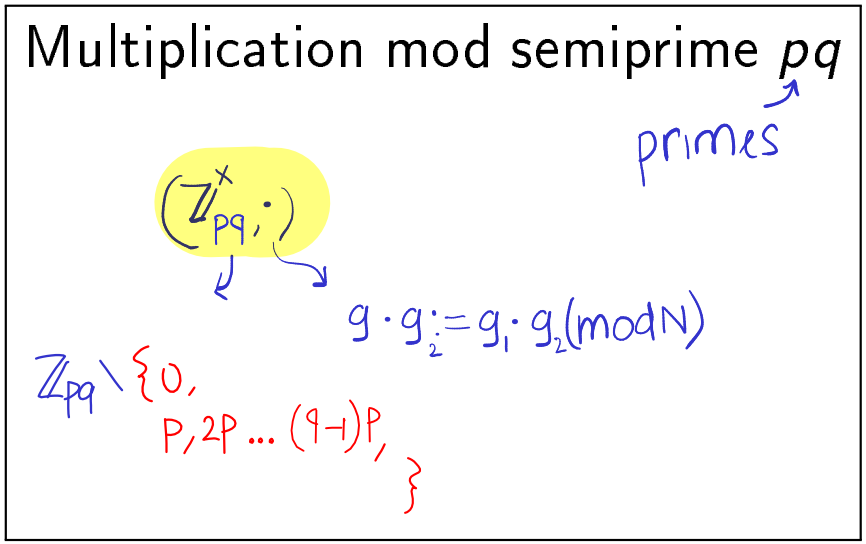
Multiplication mod semiprime pq



- What are the elements in $\mathbb{Z}_{pq}^\times$? Every $0 \leq a < N$ that is invertible
  - 0 is not invertible

Multiplication Modulo Semiprime $N = pq$

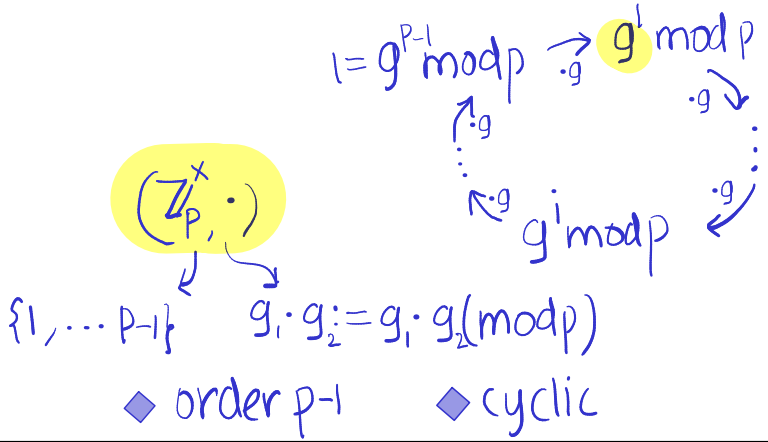




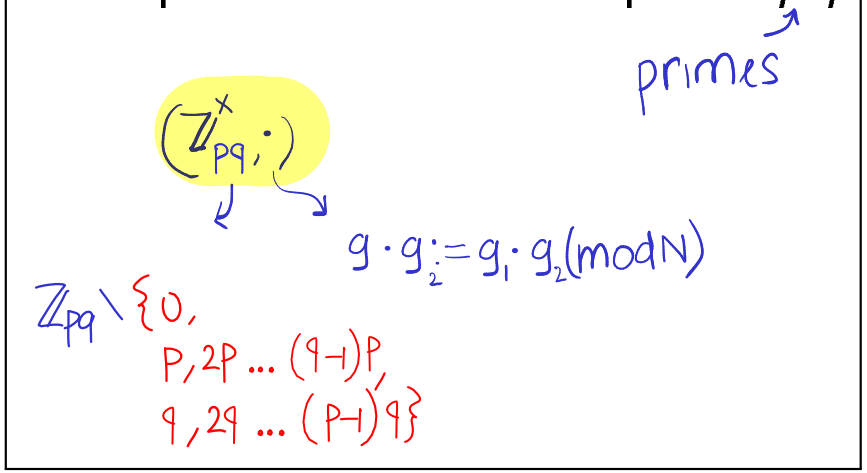
- What are the elements in $\mathbb{Z}_{pq}^\times$? Every $0 \leq a < N$ that is invertible
  - 0 is not invertible
  - p and its multiples are not invertible (proof on whiteboard)

Multiplication Modulo Semiprime $N = pq$

Multiplication mod prime p



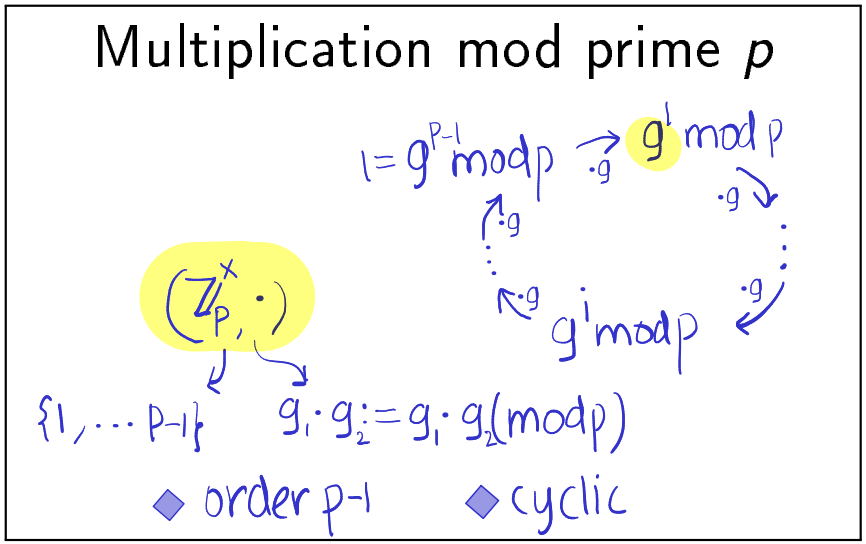
Multiplication mod semiprime pq

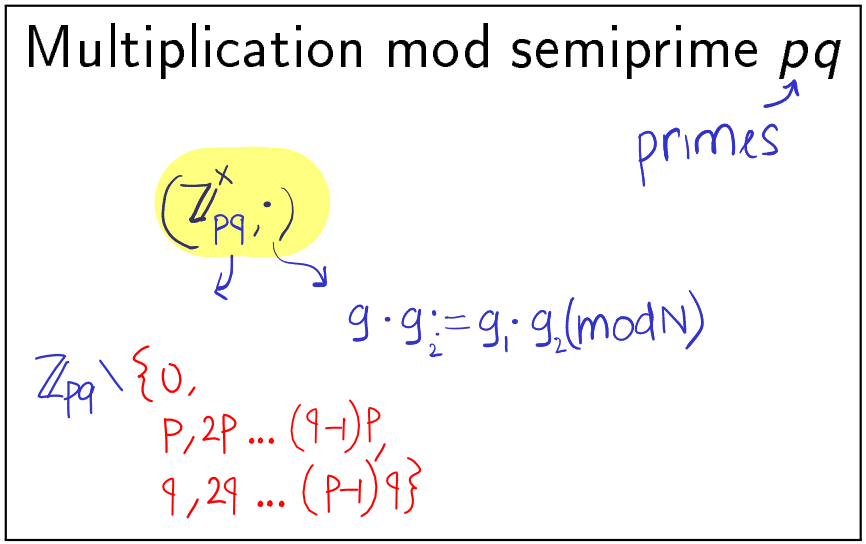


? What are the elements in $\mathbb{Z}_{pq}^\times$? Every $0 \leq a < N$ that is invertible

- 0 is not invertible
- p and its multiples are not invertible (proof on whiteboard)
- q and its multiples are not invertible

Multiplication Modulo Semiprime $N = pq$

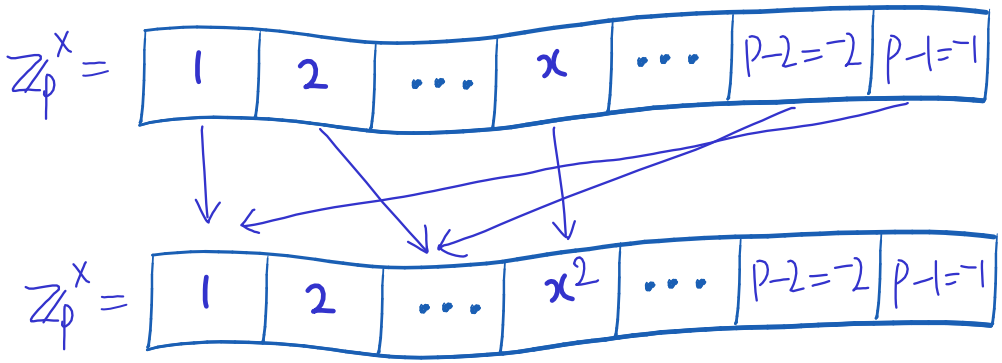




- What are the elements in $\mathbb{Z}_{pq}^\times$? Every $0 \leq a < N$ that is invertible
  - 0 is not invertible
  - p and its multiples are not invertible (proof on whiteboard)
  - q and its multiples are not invertible
- What is the order of the $\mathbb{Z}_{pq}^\times$?

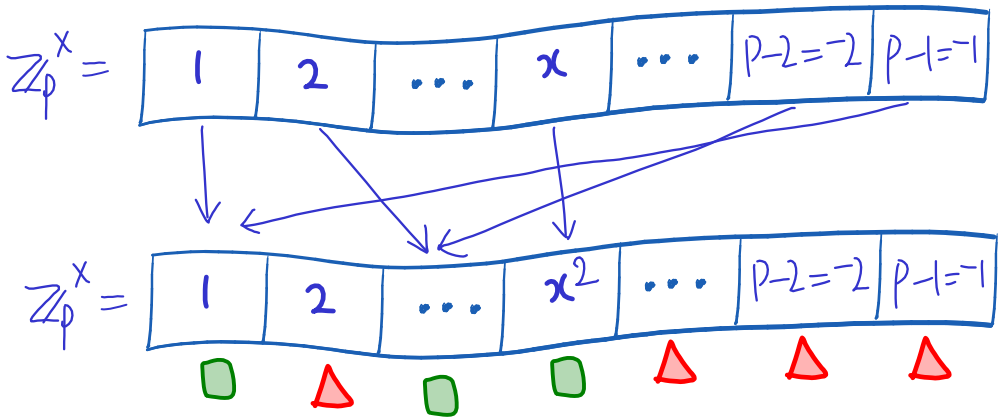
$$\begin{aligned} pq - 1 - (q-1) - (p-1) &= pq - p - q - 1 \\ &= (p-1)(q-1) =: \phi(N) \end{aligned}$$

Squaring Map $x \mapsto x^2 \bmod p$



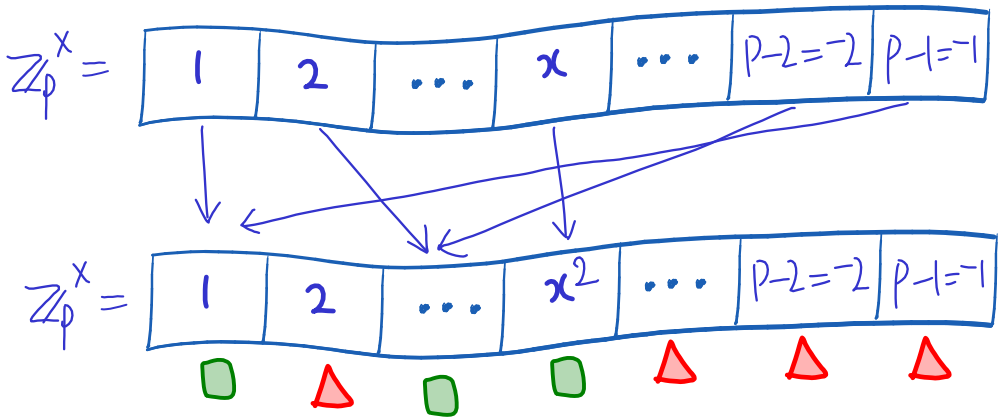
- 2-1 map

Squaring Map $x \mapsto x^2 \bmod p$



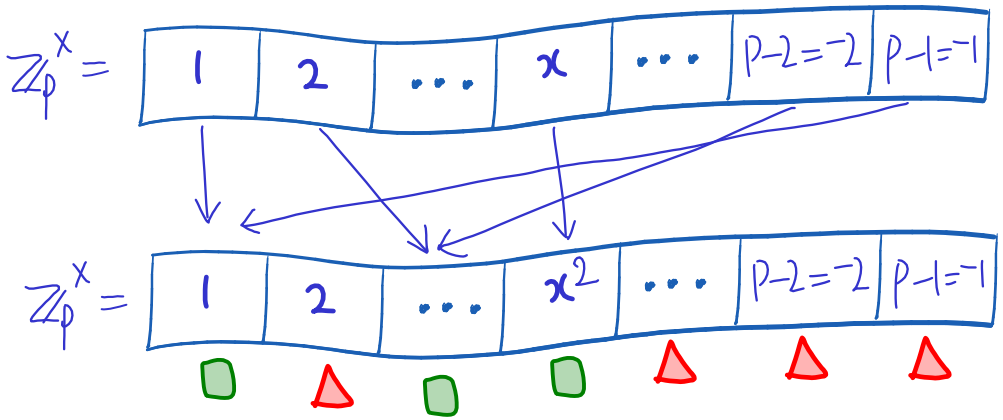
- 2-1 map $\Rightarrow$ Half the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$?

Squaring Map $x \mapsto x^2 \bmod p$



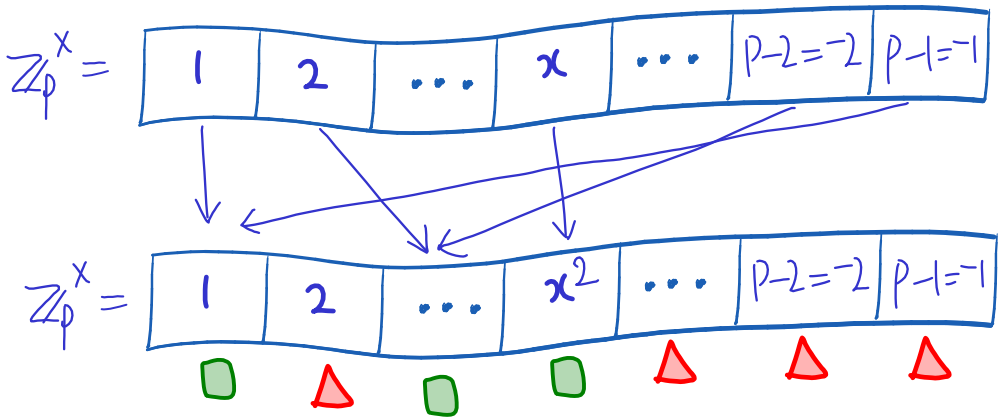
- 2-1 map $\Rightarrow$ Half the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  1. Compute discrete log x of y w.r.to *some* generator g
  2. Output "square" if x is *even*

Squaring Map $x \mapsto x^2 \bmod p$



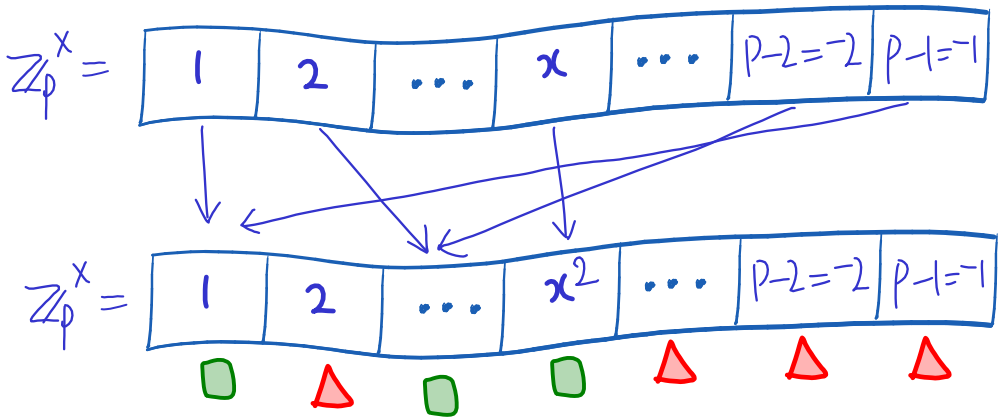
- 2-1 map $\Rightarrow$ Half the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  1. Compute discrete log x of y w.r.to *some* generator g
  2. Output "square" if x is *even*
- Is it possible to *efficiently* test if $y \in \mathbb{Z}_p^\times(+)$?

Squaring Map $x \mapsto x^2 \bmod p$



- 2-1 map $\Rightarrow$ Half the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  1. Compute discrete log x of y w.r.to *some* generator g
  2. Output "square" if x is *even*
- Is it possible to *efficiently* test if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  1. Compute *sign* $y^{(p-1)/2} \in \{\pm 1\}$ (Legendre symbol)
  2. Output "square" if sign is $+1$

Squaring Map $x \mapsto x^2 \bmod p$

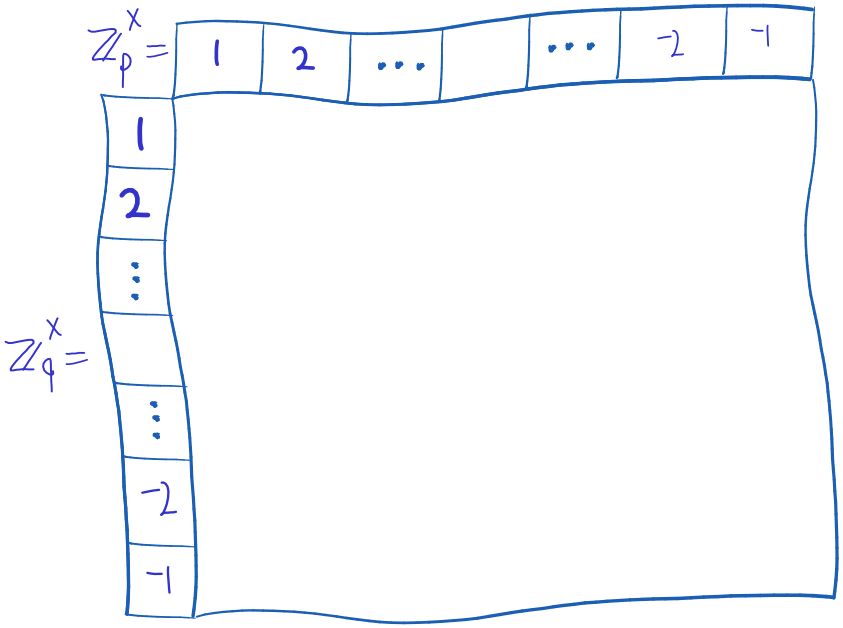


- 2-1 map $\Rightarrow$ Half the elements $\mathbb{Z}_p^\times(+) \subset \mathbb{Z}_p^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  1. Compute discrete log x of y w.r.to *some* generator g
  2. Output "square" if x is *even*
- Is it possible to *efficiently* test if $y \in \mathbb{Z}_p^\times(+)$? Yes:
  1. Compute *sign* $y^{(p-1)/2} \in \{\pm 1\}$ (Legendre symbol)
  2. Output "square" if sign is $+1$

Exercise 1 (Exercise 2, Assignment 4)

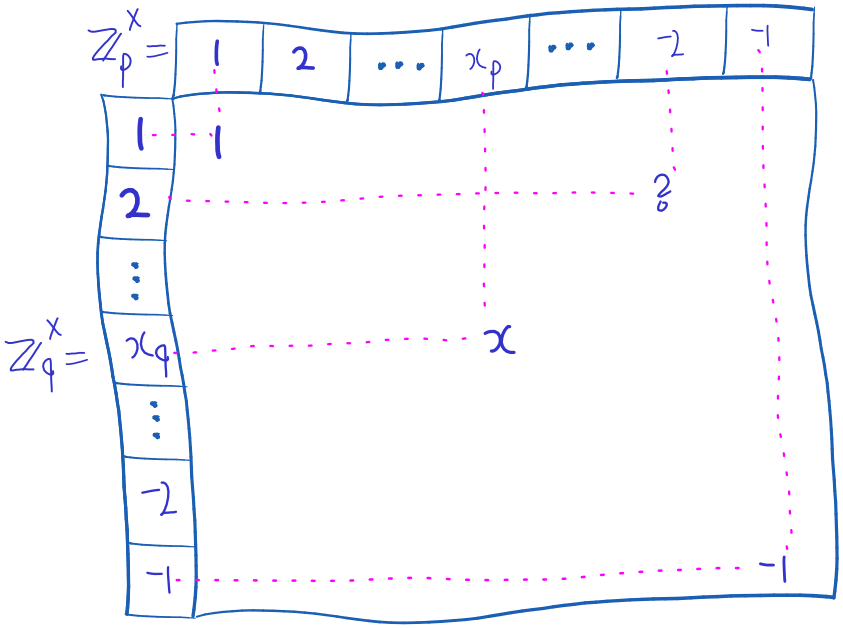
Show that DDH assumption doesn't hold in $(\mathbb{Z}_p^\times, \cdot)$

Squaring Map $x \mapsto x^2 \bmod pq$...



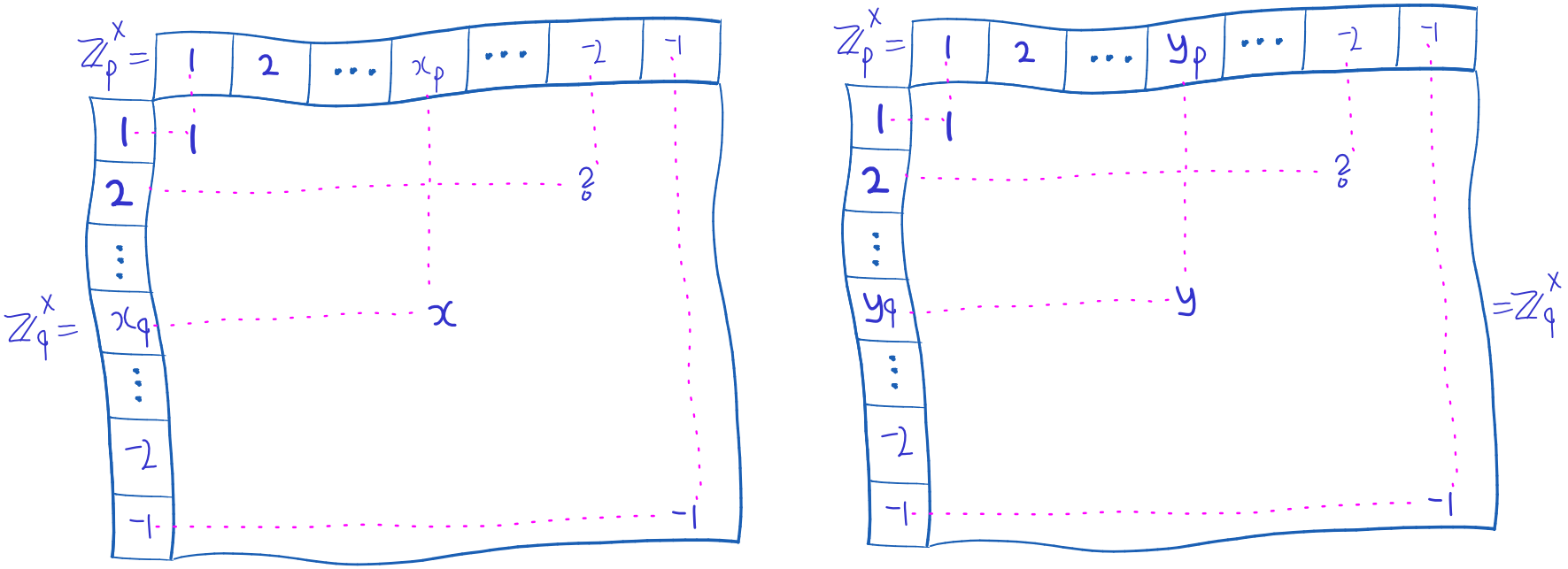
- Chinese Remaindering Theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ (on whiteboard)

Squaring Map $x \mapsto x^2 \bmod pq$



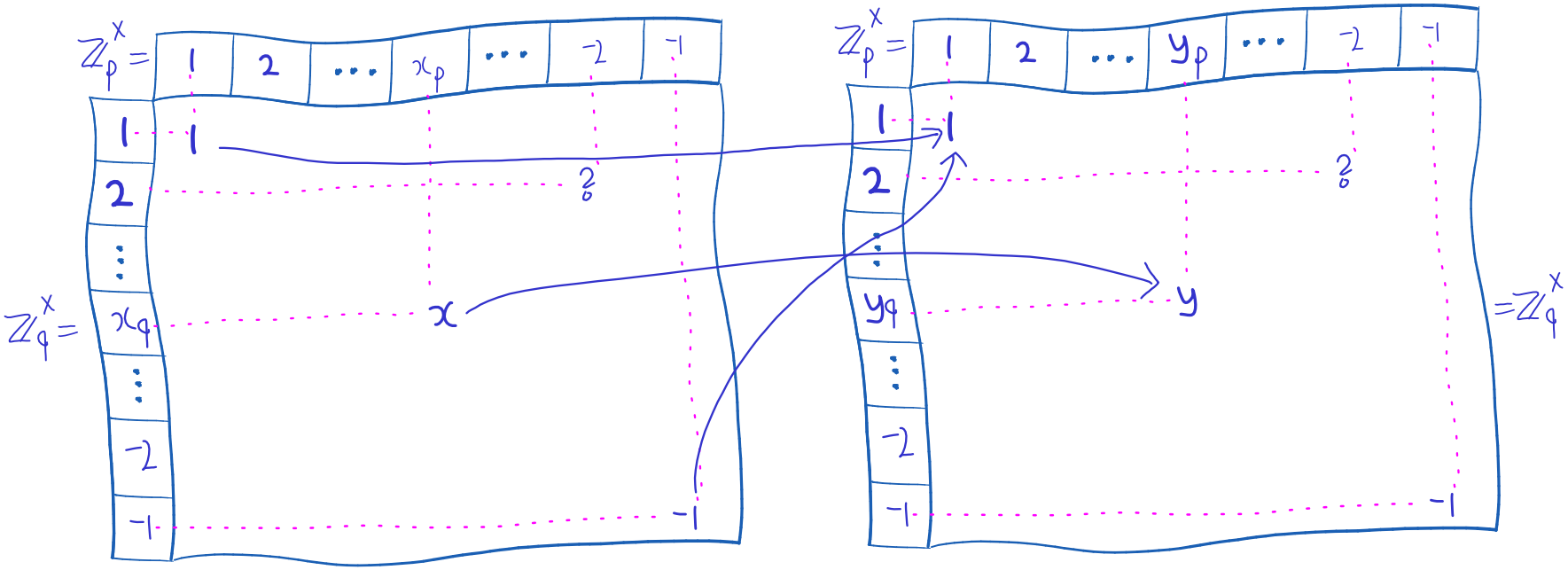
- Chinese Remaindering Theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ (on whiteboard)

Squaring Map $x \mapsto x^2 \bmod pq$



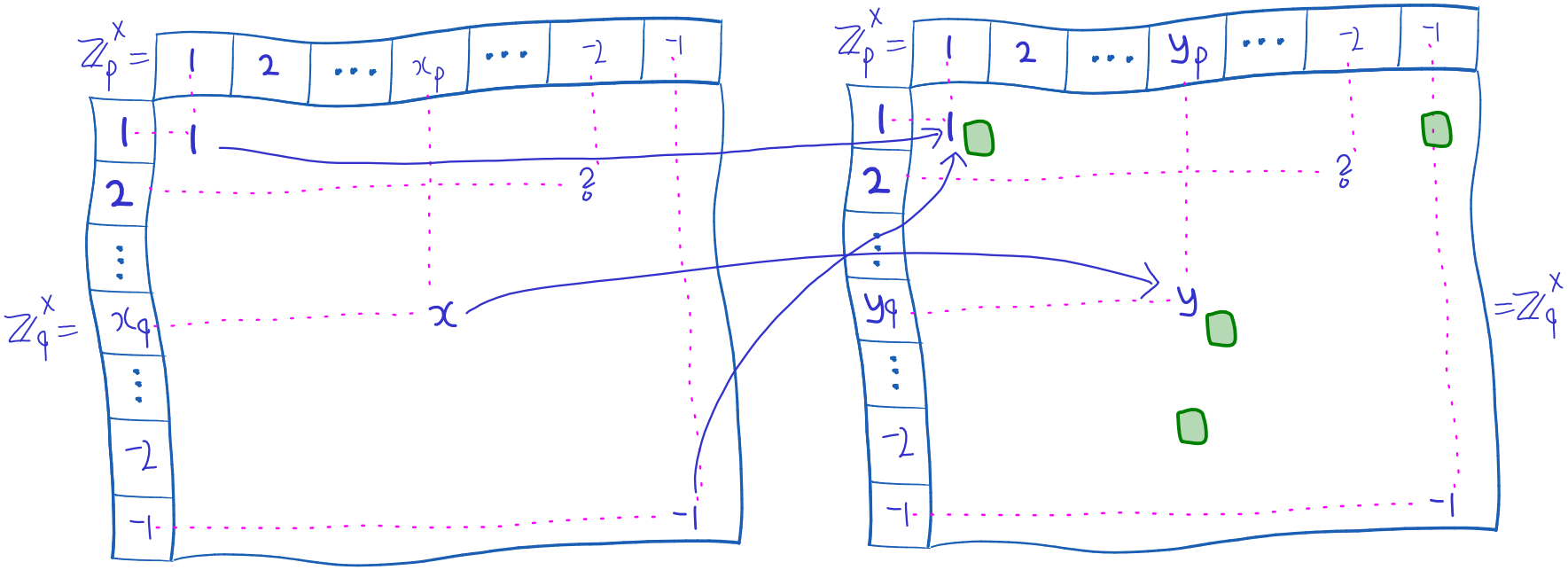
- Chinese Remaindering Theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ (on whiteboard)
- $\Rightarrow$ 4-1 map $\Rightarrow$ 1/4 of elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots

Squaring Map $x \mapsto x^2 \bmod pq$



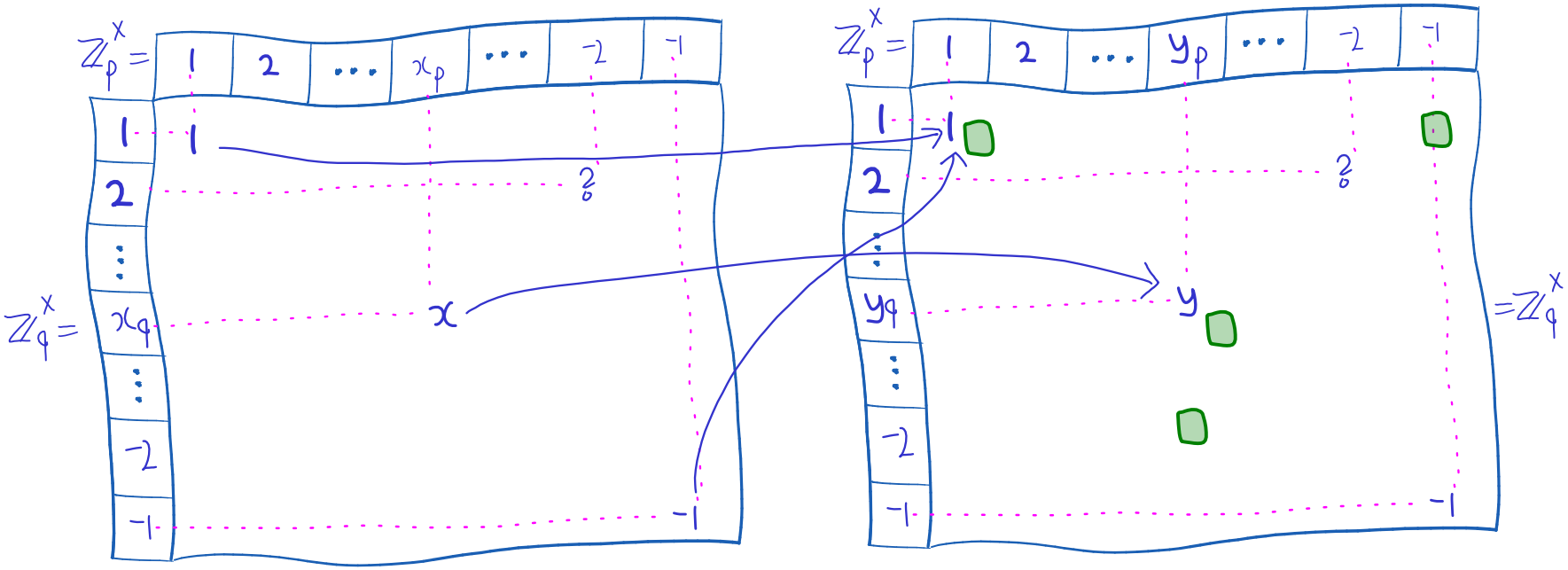
- Chinese Remaindering Theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ (on whiteboard)
- $\Rightarrow$ 4-1 map $\Rightarrow$ 1/4 of elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots

Squaring Map $x \mapsto x^2 \bmod pq$



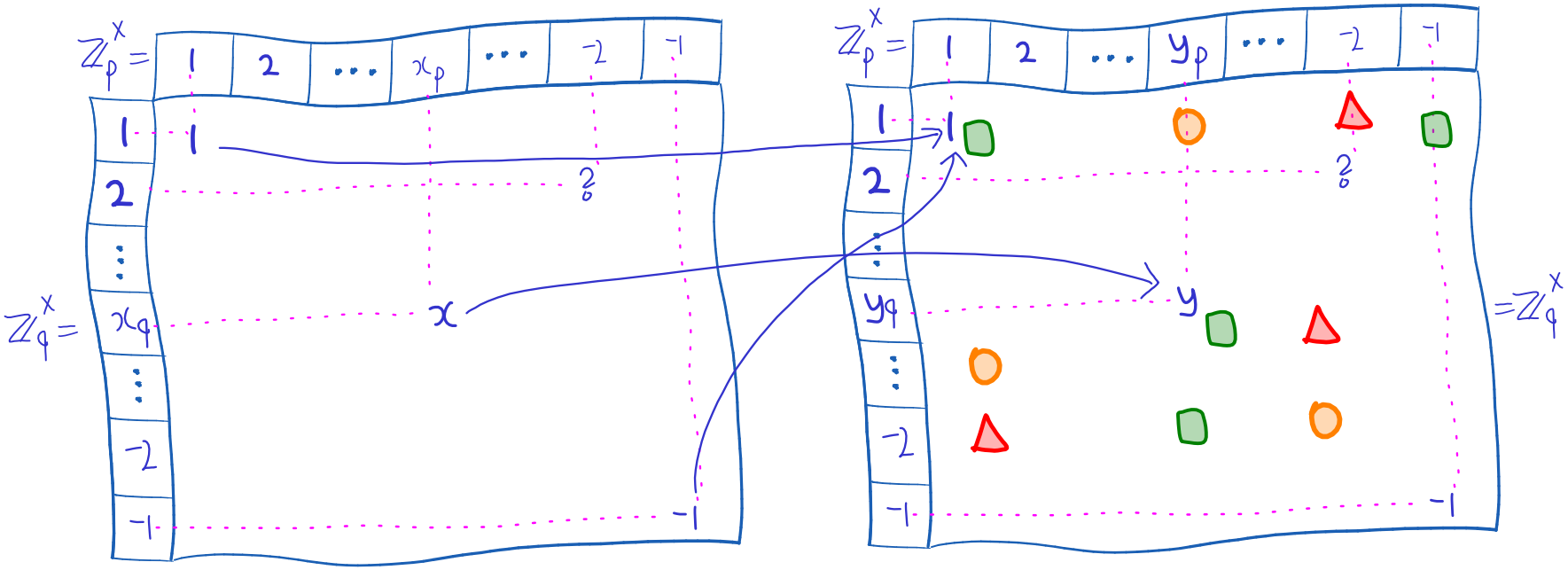
- Chinese Remaindering Theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ (on whiteboard)
- $\Rightarrow$ 4-1 map $\Rightarrow$ 1/4 of elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots

Squaring Map $x \mapsto x^2 \bmod pq$



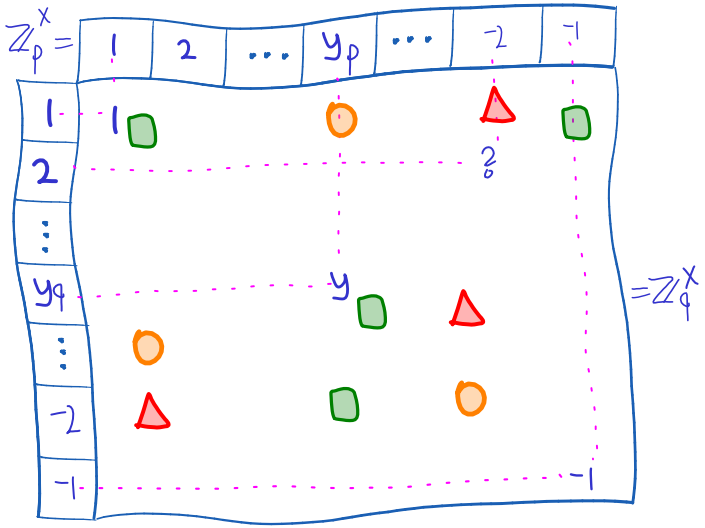
- Chinese Remaindering Theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ (on whiteboard)
- $\Rightarrow$ 4-1 map $\Rightarrow$ 1/4 of elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_N^\times(+,+)$? Yes:
  - Output "square" if $y \in \mathbb{Z}_p^\times(+)$ *and* $y \in \mathbb{Z}_q^\times(+)$

Squaring Map $x \mapsto x^2 \bmod pq$



- Chinese Remaindering Theorem: $\mathbb{Z}_N^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ (on whiteboard)
- $\Rightarrow$ 4-1 map $\Rightarrow$ 1/4 of elements $\mathbb{Z}_N^\times(+,+) \subset \mathbb{Z}_N^\times$ have square roots
- Is it possible to *test* if $y \in \mathbb{Z}_N^\times(+,+)$? Yes:
  - Output "square" if $y \in \mathbb{Z}_p^\times(+)$ *and* $y \in \mathbb{Z}_q^\times(+)$
- Is it possible to *efficiently test* if $y \in \mathbb{Z}_N^\times(+,+)$? Unclear
- Can efficiently distinguish $\mathbb{Z}_N^\times(+,+) \cup \mathbb{Z}_N^\times(-,-)$ from $\mathbb{Z}_N^\times(-,+) \cup \mathbb{Z}_N^\times(+,-)$: compute Jacobi symbol

Squaring Map $x \mapsto x^2 \bmod pq$...

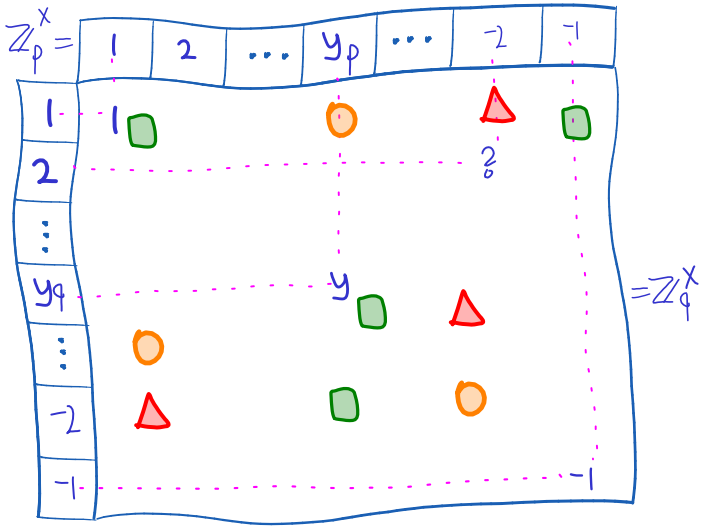


Assumption 2 (Quadratic residuosity (QR) assumption w.r.to S...)

...holds if for all PPT distinguishers D, the following is negligible:

$$\delta(n) := \left| \Pr_{\substack{N \leftarrow \mathsf{S}(1^n) \\ y \leftarrow \mathbb{Z}_N^\times(+,+)}} [\mathsf{D}(N, y) = 0] - \Pr_{\substack{N \leftarrow \mathsf{S}(1^n) \\ y \leftarrow \mathbb{Z}_N^\times(-,-)}} [\mathsf{D}(N, y) = 0] \right|$$

Squaring Map $x \mapsto x^2 \bmod pq$...



Assumption 2 (Quadratic residuosity (QR) assumption w.r.to S...)

...holds if for all PPT distinguishers D, the following is negligible:

$$\delta(n) := \left| \Pr_{\substack{N \leftarrow \mathsf{S}(1^n) \\ y \leftarrow \mathbb{Z}_N^\times(+,+)}} [\mathsf{D}(N, y) = 0] - \Pr_{\substack{N \leftarrow \mathsf{S}(1^n) \\ y \leftarrow \mathbb{Z}_N^\times(-,-)}} [\mathsf{D}(N, y) = 0] \right|$$

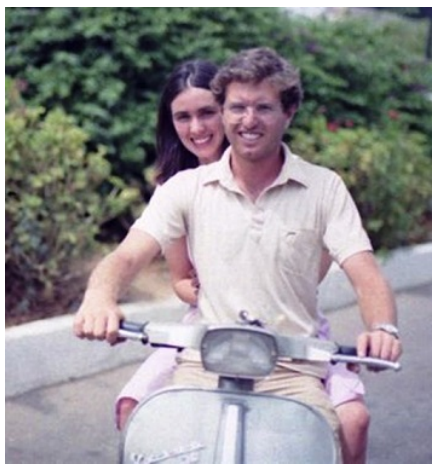
Exercise 2

1. Show that QR assumption implies Factoring assumption
2. Show that *computing* square root mod N is equivalent to factoring

Plan for Today's Lecture

- Task: public-key encryption (PKE)
- Threat model: IND-CPA

Goldwasser-Micali PKE



©Oded Goldreich

RSA PKE

©cs.miami.edu (Rosenberg)

Algebraic setting: multiplication modulo *semiprime* (RSA group)

Goldwasser-Micali Public-Key (Bit) Encryption

Key idea: encode message in the "sign"

- $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$

Goldwasser-Micali Public-Key (Bit) Encryption

Key idea: encode message in the "sign"

- $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
- Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

Pseudocode 3 (Goldwasser-Micali PKE for $\mathcal{M}_n := \{0,1\}$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. Output $(pk := N, sk := (p,q))$

Goldwasser-Micali Public-Key (Bit) Encryption

Key idea: encode message in the "sign"

- $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
- Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

Pseudocode 3 (Goldwasser-Micali PKE for $\mathcal{M}_n := \{0,1\}$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. Output $(pk := N, sk := (p,q))$

Goldwasser-Micali Public-Key (Bit) Encryption

Key idea: encode message in the "sign"

- $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
- Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

Pseudocode 3 (Goldwasser-Micali PKE for $\mathcal{M}_n := \{0,1\}$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. Output $(pk := N, sk := (p,q))$
- Encryption $\mathsf{Enc}(pk, m)$:
  1. Sample random $r \leftarrow \mathbb{Z}_N^\times$
  2. Output $c := (-1)^m \cdot r^2 \bmod N$

Goldwasser-Micali Public-Key (Bit) Encryption

Key idea: encode message in the "sign"

- $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
- Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

Pseudocode 3 (Goldwasser-Micali PKE for $\mathcal{M}_n := \{0,1\}$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. Output $(pk := N, sk := (p,q))$
- Encryption $\mathsf{Enc}(pk, m)$:
  1. Sample random $r \leftarrow \mathbb{Z}_N^\times$
  2. Output $c := (-1)^m \cdot r^2 \bmod N$
- Decryption $\mathsf{Dec}(sk, c)$: output

$$\begin{cases} 0 & \text{if } c \in \mathbb{Z}_N^\times(+,+) \cong \mathbb{Z}_p^\times(+) \times \mathbb{Z}_q^\times(+) \\ 1 & \text{otherwise} \end{cases}$$

Goldwasser-Micali Public-Key (Bit) Encryption

Key idea: encode message in the "sign"

- $0 \mapsto \mathbb{Z}_N^\times(+,+)$ and $1 \mapsto \mathbb{Z}_N^\times(-,-)$
- Exploit the fact that $-1 \in \mathbb{Z}_N^\times(-,-)$

Pseudocode 3 (Goldwasser-Micali PKE for $\mathcal{M}_n := \{0,1\}$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors $(N,(p,q)) \leftarrow \mathsf{S}(1^n)$
  2. Output $(pk := N, sk := (p,q))$
- Encryption $\mathsf{Enc}(pk, m)$:
  1. Sample random $r \leftarrow \mathbb{Z}_N^\times$
  2. Output $c := (-1)^m \cdot r^2 \bmod N$
- Decryption $\mathsf{Dec}(sk, c)$: output

$$\begin{cases} 0 & \text{if } c \in \mathbb{Z}_N^\times(+,+) \cong \mathbb{Z}_p^\times(+) \times \mathbb{Z}_q^\times(+) \\ 1 & \text{otherwise} \end{cases}$$

- Correctness of decryption: since $r^2 \in \mathbb{Z}_N^\times(+,+)$, $c \in \mathbb{Z}_N^\times(+,+)$ iff $m = 0$

Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

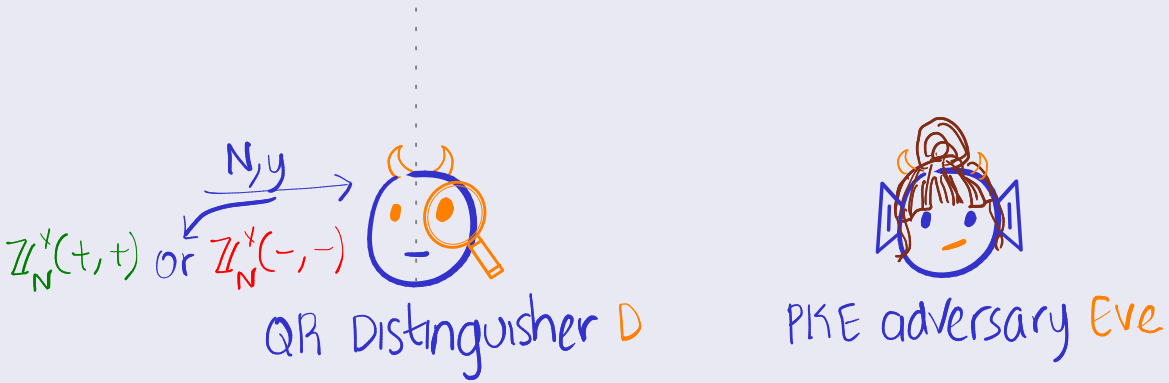
Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.

Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.

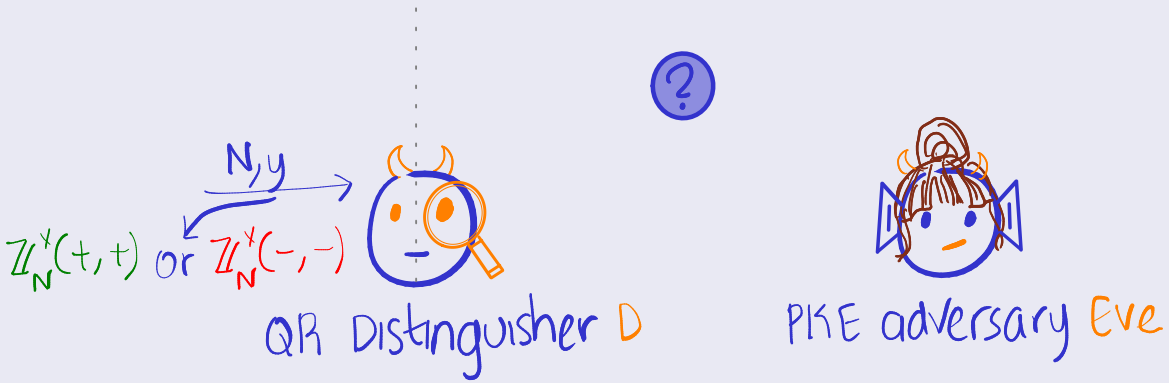


Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.

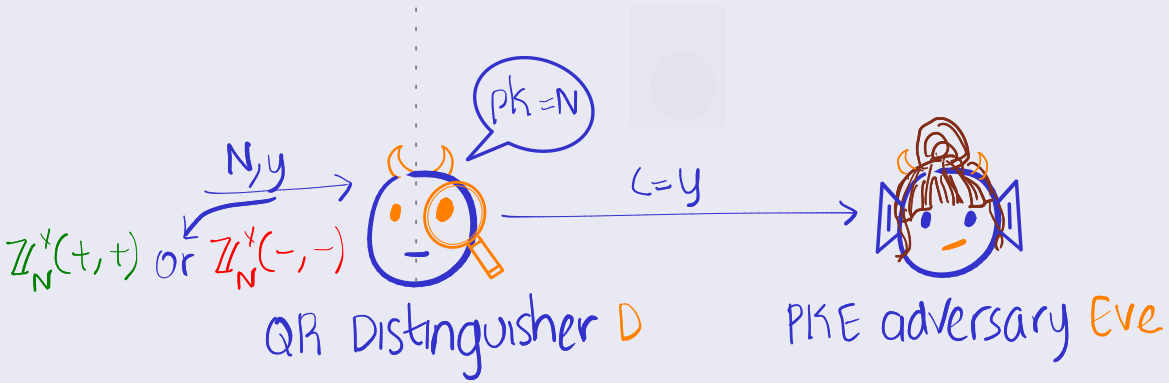


Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.

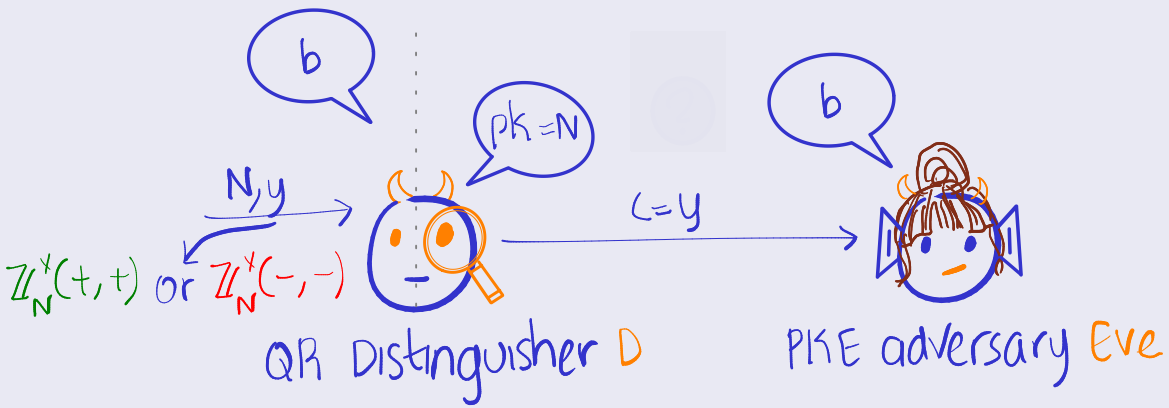


Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.

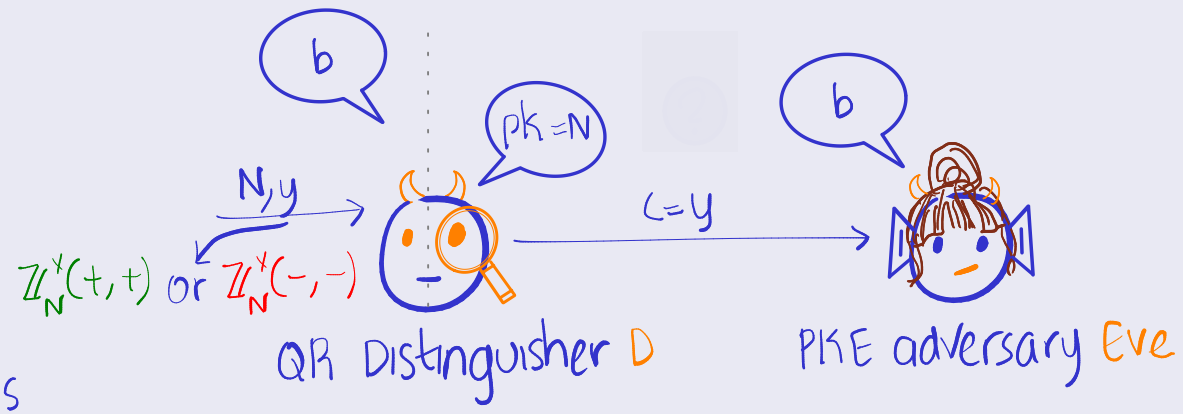


Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.



Analysis

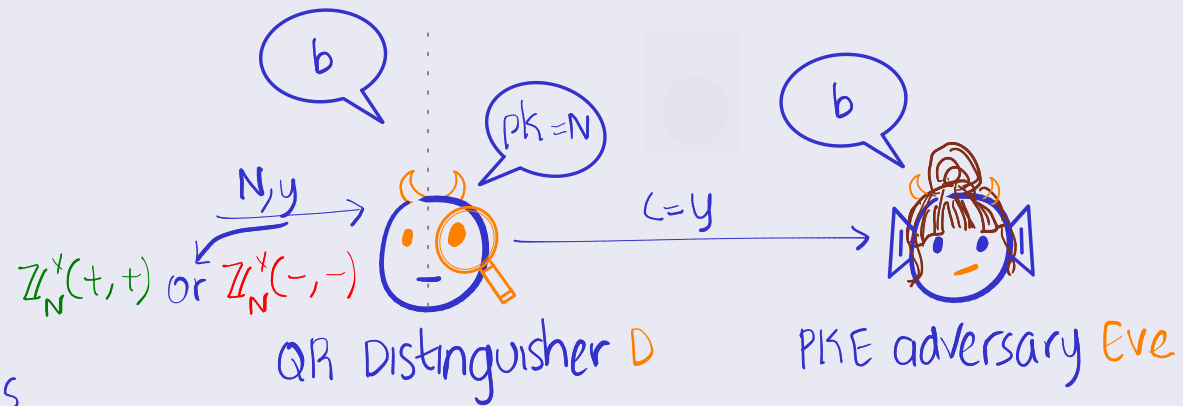
$$\left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [D(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [D(N,y)=0] \right|$$

Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.



Analysis

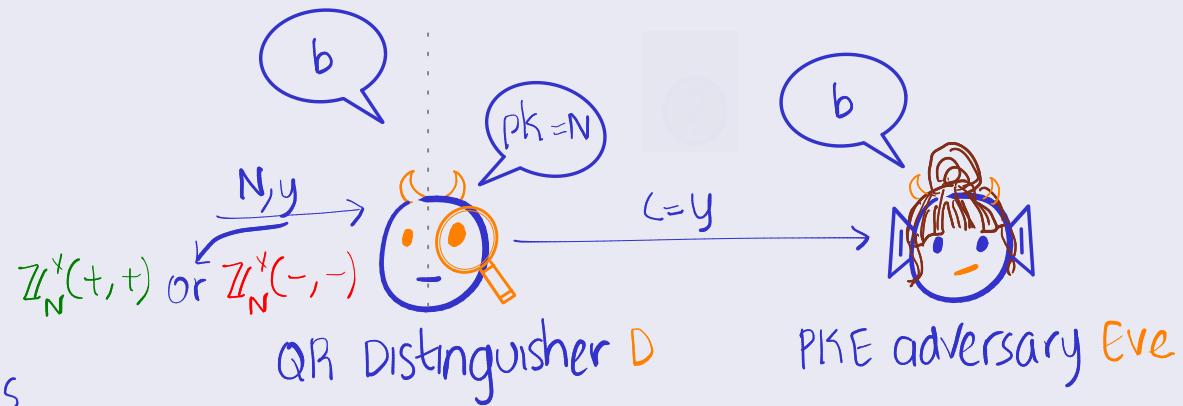
$$\left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [D(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [D(N,y)=0] \right| = \left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [Eve(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [Eve(N,y)=0] \right|$$

Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.



Analysis

$$\left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [D(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [D(N,y)=0] \right| = \left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [Eve(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [Eve(N,y)=0] \right|$$

distributed identically to Enc(pk, 0)

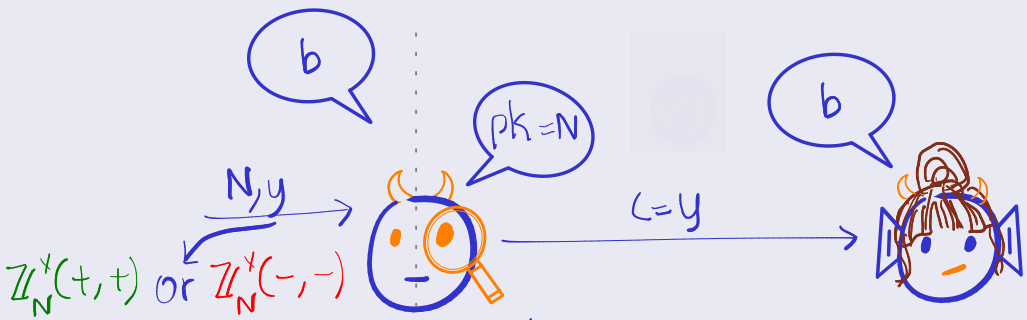
distributed identically to Enc(pk, 1)

Goldwasser-Micali PKE is CPA-secret

Theorem 1 (QR $\rightarrow$ IND-CPA security1)

Goldwasser-Micali PKE is CPA-secret under QR assumption.

Proof. $\exists$D against QR $\Leftarrow$ $\exists$Eve against PKE.



$\Rightarrow$ non-negligible

Analysis

$$\left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [D(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [D(N,y)=0] \right| = \left| \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(+,+)}} [Eve(N,y)=0] - \Pr_{\substack{N \leftarrow S(1^n) \\ y \leftarrow \mathbb{Z}_N^{\times}(-,-)}} [Eve(N,y)=0] \right|$$

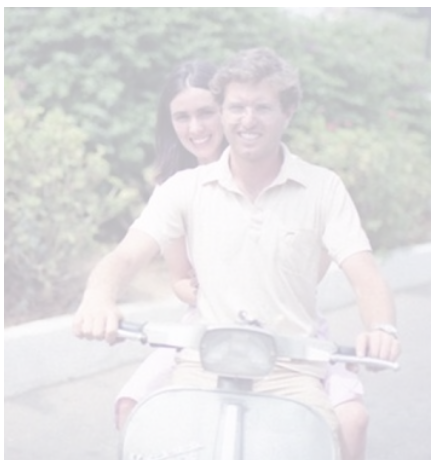
distributed identically to Enc(pk, 0)

distributed identically to Enc(pk, 1)

Plan for Today's Lecture

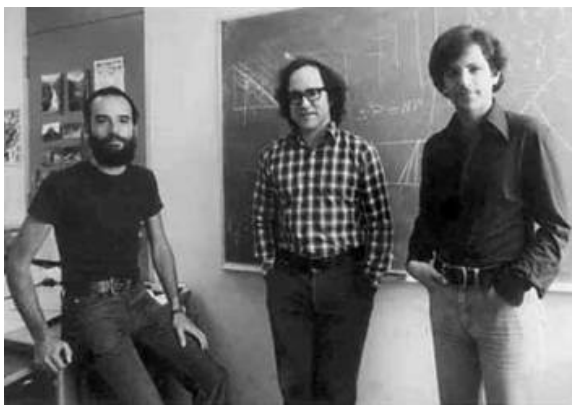
- Task: public-key encryption (PKE)
- Threat model: IND-CPA

Goldwasser-Micali PKE



©Oded Goldreich

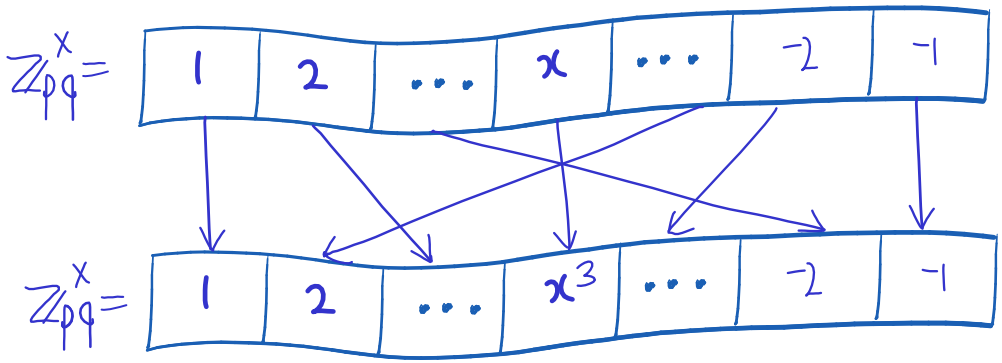
RSA PKE



©cs.miami.edu (Rosenberg)

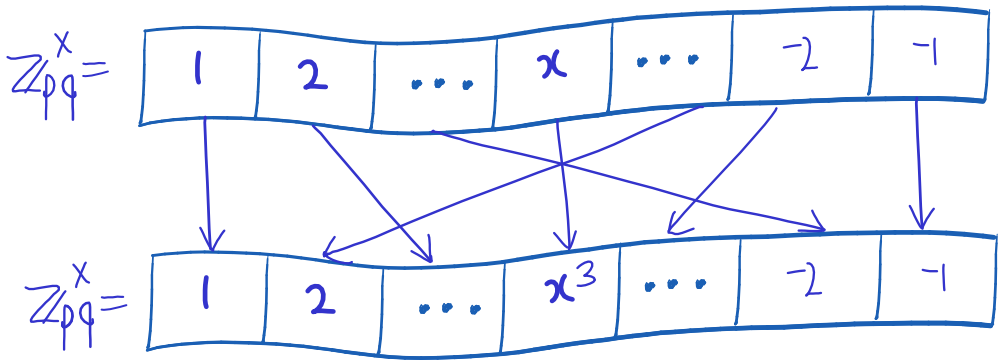
Algebraic setting: multiplication modulo semiprime (RSA group)

Powering Map $x \mapsto x^e \bmod pq$



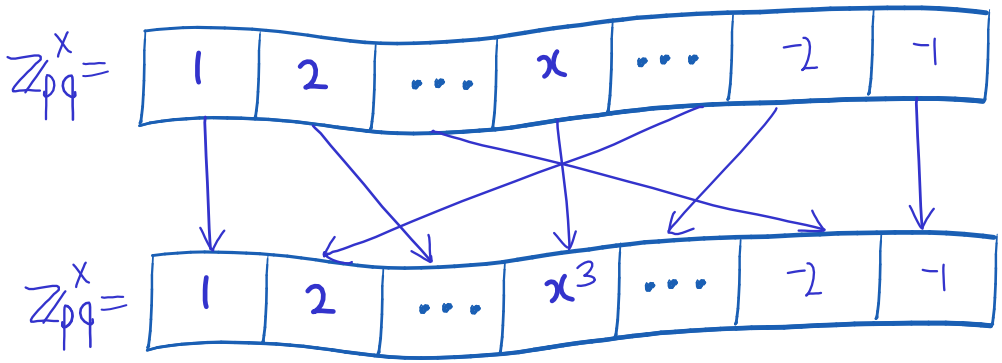
- Consider $f_{N,e}(x) := x^e \bmod N$ for $3 \leq e \leq \phi(N)$
  - $f_{N,e}$ is a permutation if e is coprime to $\phi(N)$
  - Efficiently computable via square-and-multiply

Powering Map $x \mapsto x^e \bmod pq$



- Consider $f_{N,e}(x) := x^e \bmod N$ for $3 \leq e \leq \phi(N)$
  - $f_{N,e}$ is a permutation if e is coprime to $\phi(N)$
  - Efficiently computable via square-and-multiply
- What about the inverse map $f_{N,e}^{-1}(x) := x^{1/e} \bmod N$?
  - Taking e-th root believed to be hard

Powering Map $x \mapsto x^e \bmod pq$



- Consider $f_{N,e}(x) := x^e \bmod N$ for $3 \leq e \leq \phi(N)$
  - $f_{N,e}$ is a permutation if e is coprime to $\phi(N)$
  - Efficiently computable via square-and-multiply
- What about the inverse map $f_{N,e}^{-1}(x) := x^{1/e} \bmod N$?
  - Taking e-th root believed to be hard

Assumption 3 (RSA assumption w.r.to S...)

... holds if for all PPT A, the following is negligible:

$$\delta(n) := \Pr_{\substack{(N,(p,q)) \leftarrow S(1^n) \\ e \leftarrow [1,\phi(N)] \\ x \leftarrow [1,N]}} [A(N, x^e) = x]$$

(Textbook) RSA PKE


Key idea: apply the power map to encrypt

Pseudocode 4 (RSA PKE for $\mathcal{M}_n := \mathbb{Z}_N^\times$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors: $(N, (p, q)) \leftarrow \mathsf{S}(1^n)$
  2. Sample $e \leftarrow [1, \phi(N)]$ such that $gcd(e, \phi(N)) = 1$
  3. Compute d such that $ed = 1 \bmod \phi(N)$
  4. Output $(pk := (N, e), sk := (N, d))$

(Textbook) RSA PKE


Key idea: apply the power map to encrypt

Pseudocode 4 (RSA PKE for $\mathcal{M}_n := \mathbb{Z}_N^\times$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors: $(N, (p, q)) \leftarrow \mathsf{S}(1^n)$
  2. Sample $e \leftarrow [1, \phi(N)]$ such that $gcd(e, \phi(N)) = 1$
  3. Compute d such that $ed = 1 \bmod \phi(N)$
  4. Output $(pk := (N, e), sk := (N, d))$
- Encryption $\mathsf{Enc}(pk, m)$: Output $c := m^e \bmod N$

(Textbook) RSA PKE


Key idea: apply the power map to encrypt

Pseudocode 4 (RSA PKE for $\mathcal{M}_n := \mathbb{Z}_N^\times$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors: $(N, (p, q)) \leftarrow \mathsf{S}(1^n)$
  2. Sample $e \leftarrow [1, \phi(N)]$ such that $gcd(e, \phi(N)) = 1$
  3. Compute d such that $ed = 1 \bmod \phi(N)$
  4. Output $(pk := (N, e), sk := (N, d))$
- Encryption $\mathsf{Enc}(pk, m)$: Output $c := m^e \bmod N$
- Decryption $\mathsf{Dec}(sk, c)$: Output $m := c^d \bmod N$

(Textbook) RSA PKE


Key idea: apply the power map to encrypt

Pseudocode 4 (RSA PKE for $\mathcal{M}_n := \mathbb{Z}_N^\times$)

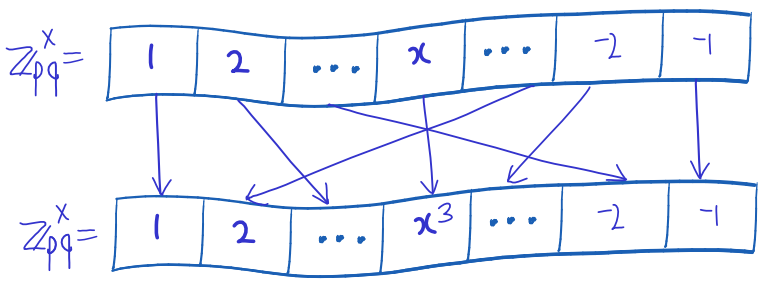
- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample semiprime with factors: $(N, (p, q)) \leftarrow \mathsf{S}(1^n)$
  2. Sample $e \leftarrow [1, \phi(N)]$ such that $gcd(e, \phi(N)) = 1$
  3. Compute d such that $ed = 1 \bmod \phi(N)$
  4. Output $(pk := (N, e), sk := (N, d))$
- Encryption $\mathsf{Enc}(pk, m)$: Output $c := m^e \bmod N$
- Decryption $\mathsf{Dec}(sk, c)$: Output $m := c^d \bmod N$
- Correctness of decryption : $\forall m \in \mathbb{Z}_N^\times : (m^e)^d = m^{ed} = m \bmod N$

Exercise 3

- Show that RSA PKE is not IND-CPA secure
- Show that RSA PKE is OW-CPA secure (see Assignment 4)

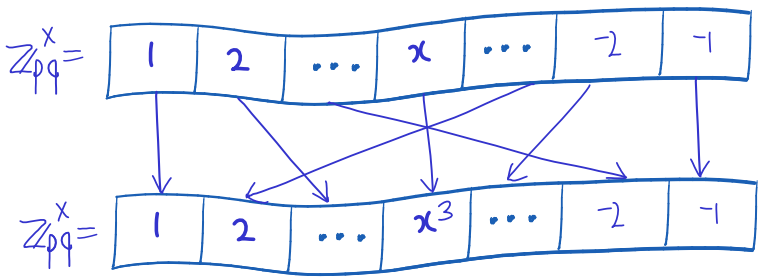
Recap/Next Lecture

- Group-based functions (easy) vs. their inverse (hard):
  - $\mathbb{Z}_p^\times$: exponentiation ($x \mapsto g^x$) vs discrete-log
  - $\mathbb{Z}_N^\times$: squaring ($x \mapsto x^2 \bmod pq$) vs square root
  - $\mathbb{Z}_N^\times$: e-th power ($x \mapsto x^e \bmod pq$) vs e-th root
- Built PKE/KEx based on these hard problems
- Takeaway: structure, structure, structure



Recap/Next Lecture

- Group-based functions (easy) vs. their inverse (hard):
  - $\mathbb{Z}_p^\times$: exponentiation ($x \mapsto g^x$) vs discrete-log
  - $\mathbb{Z}_N^\times$: squaring ($x \mapsto x^2 \bmod pq$) vs square root
  - $\mathbb{Z}_N^\times$: e-th power ($x \mapsto x^e \bmod pq$) vs e-th root
- Built PKE/KEx based on these hard problems
- Takeaway: structure, structure, structure



- Next Lecture(s): how to deal with active adversary?
  - Digital signatures: public-key version of MAC
  - How to construct digital signature

References

1. [KL14, Chapter 9.2] for more on the number theory used in this lecture
2. Goldwasser-Micali PKE was proposed in [GM82]. That paper is considered to be the first paper on "provable security"
3. A description of the RSA PKE can be found in [KL14, Chapter 12.5]

Shafi Goldwasser and Silvio Micali.

Probabilistic encryption and how to play mental poker keeping secret all partial information.

In *14th ACM STOC*, pages 365–377. ACM Press, May 1982.

Jonathan Katz and Yehuda Lindell.

Introduction to Modern Cryptography (3rd ed.).

Chapman and Hall/CRC, 2014.