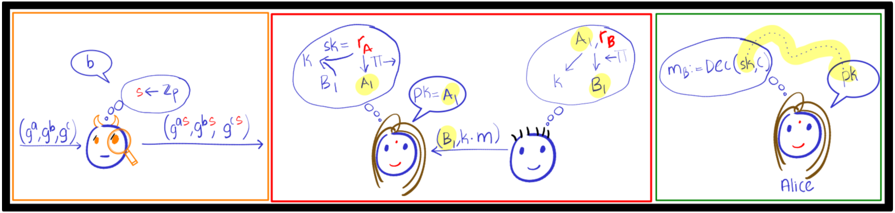
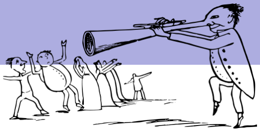


CS409m: Introduction to Cryptography

Lecture 13 (26/Sep/25)

Instructor: Chethan Kamath

Announcements



© svgsilh.com

Changes to mid-sem crib session

- View your answer sheet 12:30-14:30 on Monday (29/Sep) in CC305
- Submit cribs online by Wednesday (01/Oct, 23:59)

Bounty on Problem 7.3:

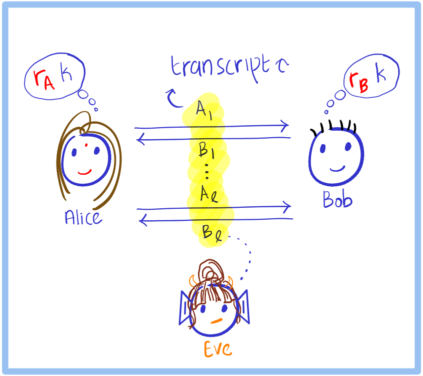
- Come up with a *simple* construction of MAC from weak PRF
- Construction provided in solution set is too complex!

- Quiz 2 on 08/Oct, 08:25-09:25

Recall from Last Lecture...

- Task: key exchange
- Threat model: computational secrecy against eavesdroppers

Key Exchange



Basic Group Theory

Definition 3 (Lecture 11)

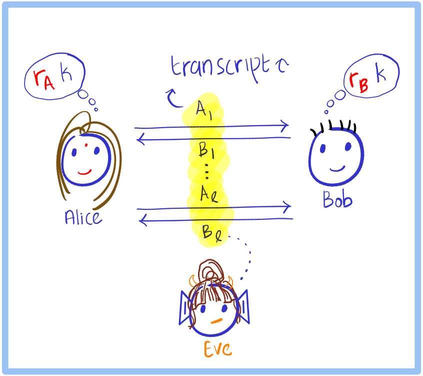
An Abelian group $\mathbb{G}$ is a set $\mathcal{G}$ with a binary op. $\cdot$ satisfying:

1. Closure
2. Associativity
3. Existence of identity
4. Existence of inverse
5. Commutativity

Recall from Last Lecture...

- Task: key exchange
- Threat model: computational secrecy against eavesdroppers

Key Exchange



Basic Group Theory

Definition 3 (Lecture 11)

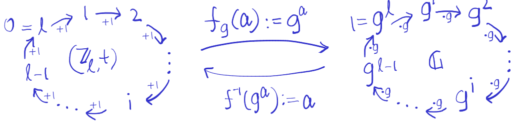
An Abelian group $\mathbb{G}$ is a set $\mathcal{G}$ with a binary op. $\cdot$ satisfying:

1. Closure
2. Associativity
3. Existence of identity
4. Existence of inverse
5. Commutativity

Motivation: need richer algebraic *structure* to construct key exchange

Recall from Last Lecture...

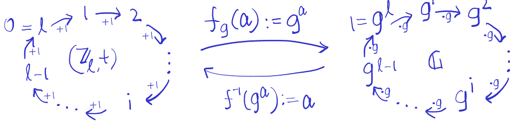
- Cyclic group: there exists a "generator" $g \in \mathcal{G}$ with order $\ell = |\mathcal{G}|$
  - That is $\{g^1 = g, g^2, \ldots, g^{\ell-1}, g^\ell = 1\} = \mathcal{G}$



  - "Isomorphism" between $(\mathbb{Z}_\ell, +)$ and $\mathbb{G}$

Recall from Last Lecture...

- Cyclic group: there exists a "generator" $g \in \mathcal{G}$ with order $\ell = |\mathcal{G}|$
  - That is $\{g^1 = g, g^2, \ldots, g^{\ell-1}, g^\ell = 1\} = \mathcal{G}$

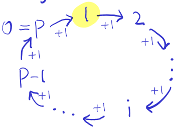


  - "Isomorphism" between $(\mathbb{Z}_\ell, +)$ and $\mathbb{G}$
- Examples:

Addition modulo prime p

◆ order p ◆ cyclic

$(\mathbb{Z}_p, +)$

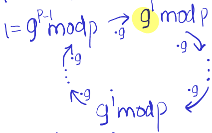


$\{0, \ldots p-1\}$ $\quad g_1 + g_2 := g_1 + g_2 \pmod p$

Multiplication modulo prime p

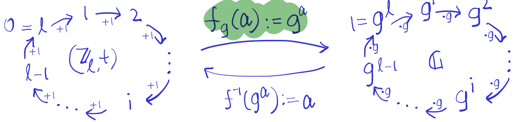
◆ order $p-1$ ◆ cyclic

$(\mathbb{Z}_p^\times, \cdot)$

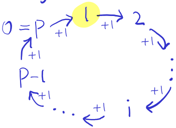
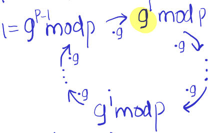


$\{1, \ldots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

- Cyclic group: there exists a "generator" $g \in \mathcal{G}$ with order $\ell = |\mathcal{G}|$
  - That is $\{g^1 = g, g^2, \ldots, g^{\ell-1}, g^{\ell} = 1\} = \mathcal{G}$



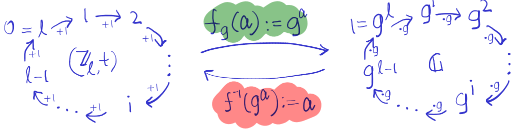
  - "Isomorphism" between $(\mathbb{Z}_\ell, +)$ and $\mathbb{G}$
- Examples:

Addition modulo prime p	Multiplication modulo prime p
◆ order p ◆ cyclic	◆ order p-1 ◆ cyclic
	
$(\mathbb{Z}_p,+)$	$(\mathbb{Z}_p^\times,\cdot)$
$\{0,\ldots p-1\}$ $g_1+g_2 := g_1+g_2 (\text{mod } p)$	$\{1,\ldots p-1\}$ $g_1\cdot g_2 := g_1\cdot g_2 (\text{mod } p)$

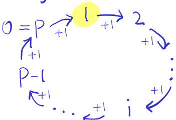
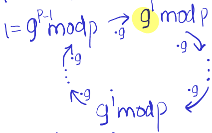
- Easy to compute: Group operation, exponentiation, inverse etc.

Recall from Last Lecture...

- Cyclic group: there exists a "generator" $g \in \mathcal{G}$ with order $\ell = |\mathcal{G}|$
  - That is $\{g^1 = g, g^2, \ldots, g^{\ell-1}, g^\ell = 1\} = \mathcal{G}$



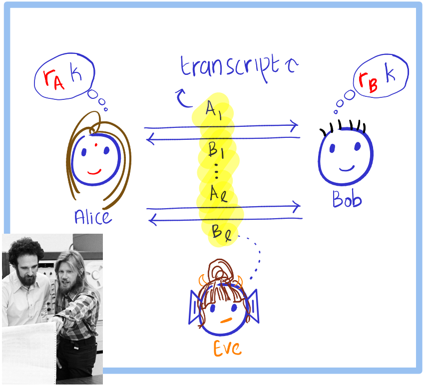
  - "Isomorphism" between $(\mathbb{Z}_\ell, +)$ and $\mathbb{G}$
- Examples:

⚠ Addition modulo prime p	Multiplication modulo prime p
◆ order p ◆ cyclic	◆ order $p-1$ ◆ cyclic
	
$(\mathbb{Z}_p, +)$: $\{0, \ldots, p-1\}$, $g_1 + g_2 := g_1 + g_2 \pmod p$	$(\mathbb{Z}_p^\times, \cdot)$: $\{1, \ldots, p-1\}$, $g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

- Easy to compute: Group operation, exponentiation, inverse etc.
- What is possibly hard to compute? Discrete logarithm (DLP)

Plan for Today's Lecture

NEW Diffie-Hellman Key Exchange

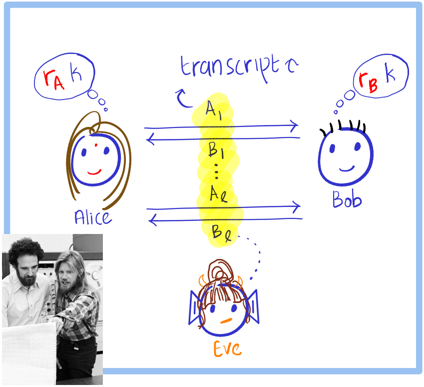


© icour.fr

Plan for Today's Lecture

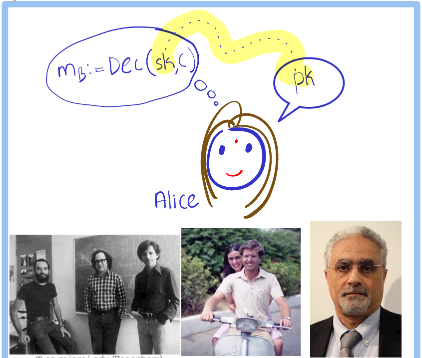
- Task: public-key encryption
- Threat model: IND-CPA

NEW Diffie-Hellman Key Exchange



© icour.fr

NEW Public-Key Encryption

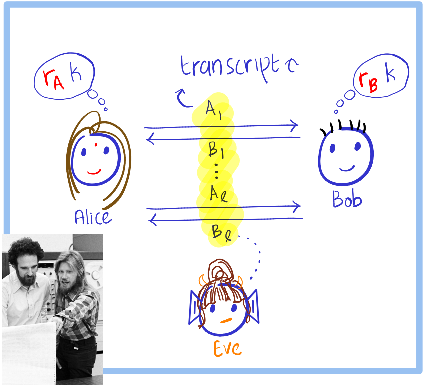


© cs.miami.edu (Rosenberg) ©Oded Goldreich ©Alexander Klink/Wikipedia

Plan for Today's Lecture

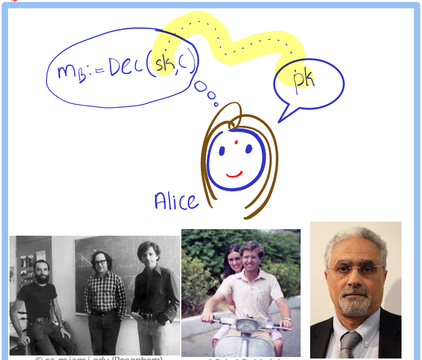
- Task: public-key encryption
- Threat model: IND-CPA

Diffie-Hellman Key Exchange



© icour.fr

Public-Key Encryption



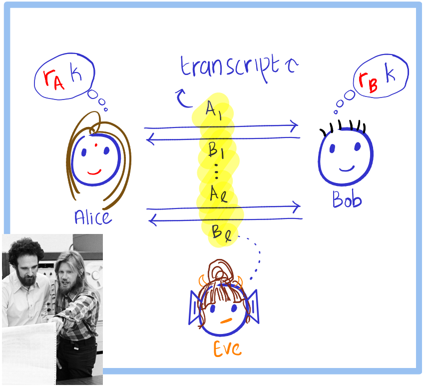
© cs.miami.edu (Rosenberg) ©Oded Goldreich ©Alexander Klink/Wikipedia

Underlying hard problem: Decisional Diffie-Hellman (DDH)

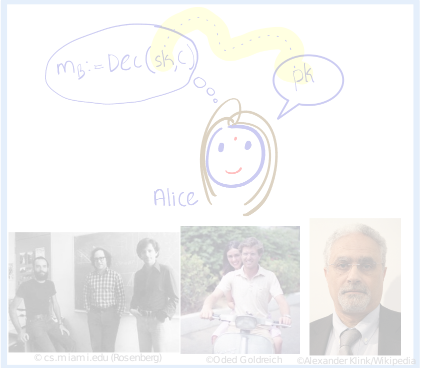
Plan for Today's Lecture

- Task: public-key encryption
- Threat model: IND-CPA

NEW Diffie-Hellman Key Exchange



NEW Public-Key Encryption

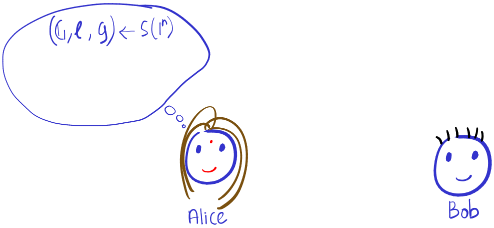


Underlying hard problem: Decisional Diffie-Hellman (DDH)

Diffie-Hellman Key-Exchange Protocol

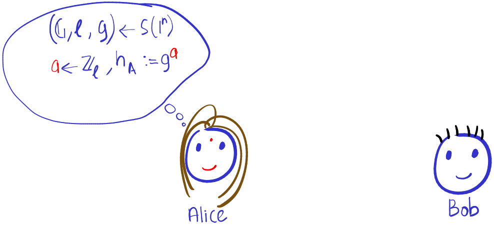
Protocol 1

Diffie-Hellman Key-Exchange Protocol



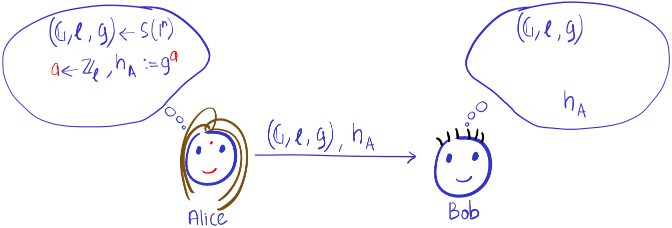
Protocol 1

Diffie-Hellman Key-Exchange Protocol



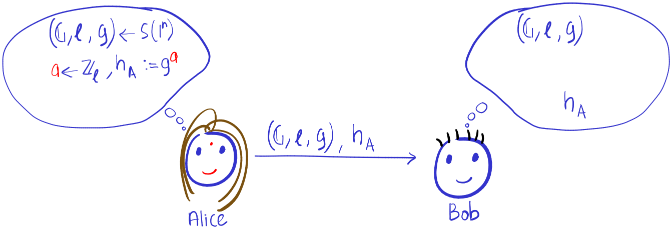
Protocol 1

Diffie-Hellman Key-Exchange Protocol



Protocol 1

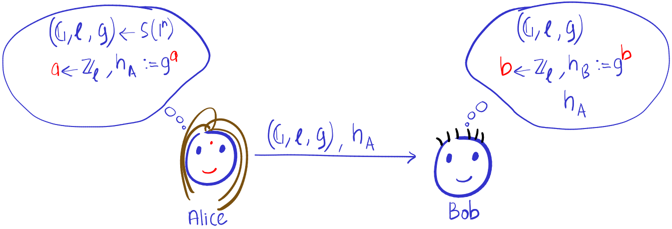
Diffie-Hellman Key-Exchange Protocol



Protocol 1

1. Alice→Bob: Send $((\mathbb{G},\ell,g), h_A := g^a)$, where $(\mathbb{G},\ell,g) \leftarrow S(1^n)$ and $a \leftarrow \mathbb{Z}_\ell$

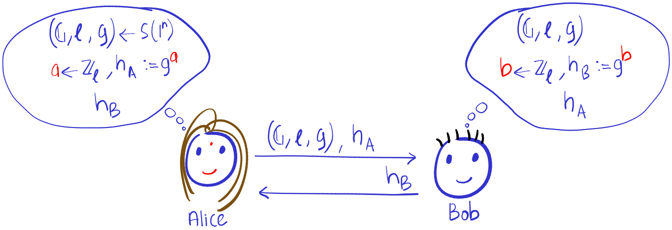
Diffie-Hellman Key-Exchange Protocol



Protocol 1

1. Alice→Bob: Send $((\mathbb{G}, \ell, g), h_A := g^a)$, where $(\mathbb{G}, \ell, g) \leftarrow S(1^n)$ and $a \leftarrow \mathbb{Z}_\ell$

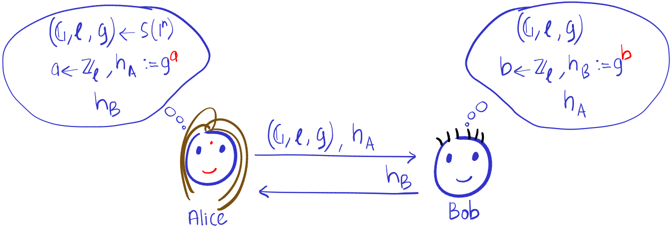
Diffie-Hellman Key-Exchange Protocol



Protocol 1

1. Alice→Bob: Send $((\mathbb{G}, \ell, g), h_A := g^a)$, where $(\mathbb{G}, \ell, g) \leftarrow S(1^n)$ and $a \leftarrow \mathbb{Z}_\ell$

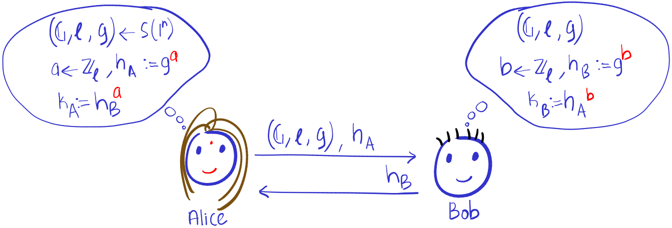
Diffie-Hellman Key-Exchange Protocol



Protocol 1

1. Alice→Bob: Send $((\mathbb{G}, \ell, g), h_A := g^a)$, where $(\mathbb{G}, \ell, g) \leftarrow \mathsf{S}(1^n)$ and $a \leftarrow \mathbb{Z}_\ell$
2. Alice←Bob: Send $h_B := g^b$ for $b \leftarrow \mathbb{Z}_\ell$

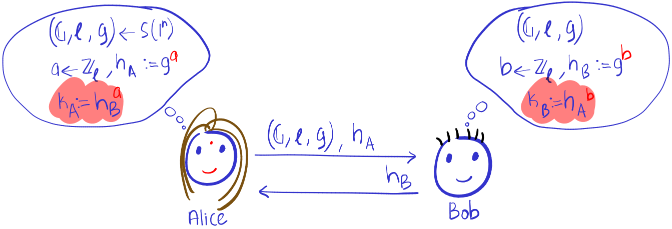
Diffie-Hellman Key-Exchange Protocol



Protocol 1

1. Alice$\rightarrow$Bob: Send $((\mathbb{G}, \ell, g), h_A := g^a)$, where $(\mathbb{G}, \ell, g) \leftarrow \mathsf{S}(1^n)$ and $a \leftarrow \mathbb{Z}_\ell$
2. Alice$\leftarrow$Bob: Send $h_B := g^b$ for $b \leftarrow \mathbb{Z}_\ell$

Diffie-Hellman Key-Exchange Protocol



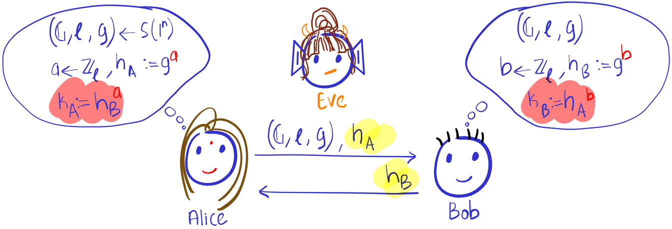
Protocol 1

1. Alice→Bob: Send $((\mathbb{G}, \ell, g), h_A := g^a)$, where $(\mathbb{G}, \ell, g) \leftarrow S(1^n)$ and $a \leftarrow \mathbb{Z}_\ell$
2. Alice←Bob: Send $h_B := g^b$ for $b \leftarrow \mathbb{Z}_\ell$
3. Alice outputs $k_A := (h_B)^a$; Bob outputs $k_B := (h_A)^b$

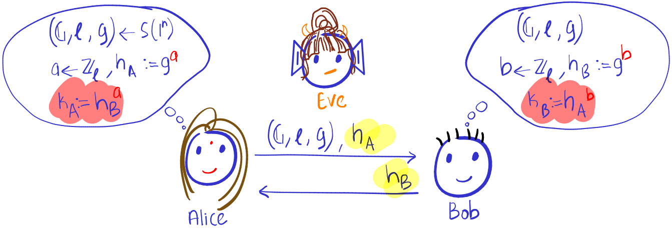
- Correctness of key generation (by Exercise 4, Lecture 12):

$$k_A = h_B^a = (g^b)^a = g^{ab} = (g^a)^b = h_A^b = k_B$$

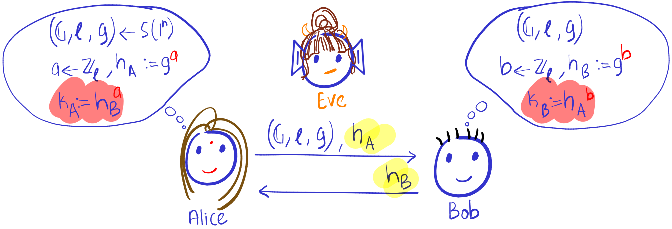
When is it Secret Against Eavesdroppers?



- What does Eve see? The transcript is $(h_A := g^a, h_B := g^b)$

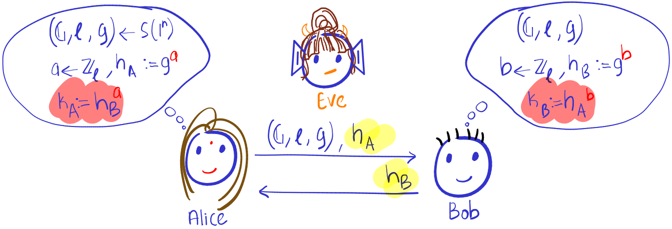


- What does Eve see? The transcript is $(h_A := g^a, h_B := g^b)$
- ? What if DLog problem is easy over $\mathbb{G}$?

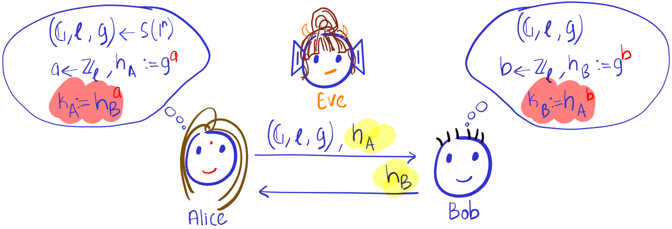


- What does Eve see? The transcript is $(h_A := g^a, h_B := g^b)$
- What if DLog problem is easy over $\mathbb{G}$?
  - ⚠ Then Eve can invert h_A to get a and compute $k = h_B^a$

When is it Secret Against Eavesdroppers?

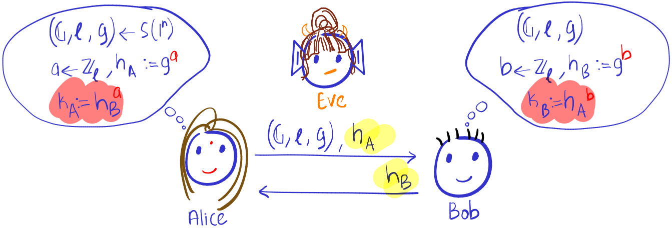


- What does Eve see? The transcript is $(h_A := g^a, h_B := g^b)$
- What if DLog problem is easy over $\mathbb{G}$?
  - Then Eve can invert h_A to get a and compute $k = h_B^a$
- Is DLog problem being hard sufficient?



- What does Eve see? The transcript is $(h_A := g^a, h_B := g^b)$
- What if DLog problem is easy over $\mathbb{G}$?
  - Then Eve can invert h_A to get a and compute $k = h_B^a$
- Is DLog problem being hard sufficient?
  - No, what if Eve can compute g^{ab} given g^a and g^b?
    - This is the "computational Diffie-Hellman" (CDH) problem

When is it Secret Against Eavesdroppers?



- What does Eve see? The transcript is $(h_A := g^a, h_B := g^b)$

What if DLog problem is easy over $\mathbb{G}$?

Then Eve can invert h_A to get a and compute $k = h_B^a$

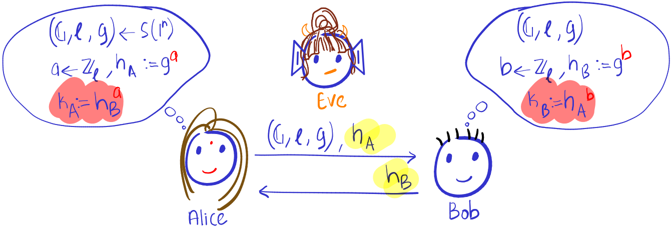
Is DLog problem being hard sufficient?

No, what if Eve can compute g^{ab} given g^a and g^b?

- This is the "computational Diffie-Hellman" (CDH) problem

Is CDH problem being hard sufficient?

When is it Secret Against Eavesdroppers?



- What does Eve see? The transcript is $(h_A := g^a, h_B := g^b)$
- What if DLog problem is easy over $\mathbb{G}$?
  - Then Eve can invert h_A to get a and compute $k = h_B^a$
- Is DLog problem being hard sufficient?
  - No, what if Eve can compute g^{ab} given g^a and g^b?
    - This is the "computational Diffie-Hellman" (CDH) problem
- Is CDH problem being hard sufficient?
  - What if Eve can distinguish g^{ab} from random group elements?
    - There exist such groups!

When is it Secret Against Eavesdroppers?...

Assumption 1 (Decisional DH (DDH) assumption in $\mathbb{G}$ w.r.to $\mathsf{S}\cdots$)

$\cdots$ holds if for all PPT distinguishers D, the following is negligible:

$$\Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow \mathsf{S}(1^n)\\ a,b\leftarrow \mathbb{Z}_\ell}}[\mathsf{D}(g^a, g^b, g^{ab}) = 0] - \Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow \mathsf{S}(1^n)\\ a,b,r\leftarrow \mathbb{Z}_\ell}}[\mathsf{D}(g^a, g^b, g^{r}) = 0]$$

"Real world" "Random world"

When is it Secret Against Eavesdroppers?...

Assumption 1 (Decisional DH (DDH) assumption in $\mathbb{G}$ w.r.to $\mathsf{S} \cdots$)

$\cdots$ holds if for all PPT distinguishers D, the following is negligible:

$$\Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow \mathsf{S}(1^n)\\ a,b\leftarrow \mathbb{Z}_\ell}}[D(g^a, g^b, g^{ab}) = 0] - \Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow \mathsf{S}(1^n)\\ a,b,r\leftarrow \mathbb{Z}_\ell}}[D(g^a, g^b, g^{r}) = 0]$$

"Real world" "Random world"

Theorem 1

Diffie-Hellman key-exchange is computationally secret against eavesdroppers under the DDH assumption in $\mathbb{G}$ w.r.to S.

Proof.

Secrecy requirement is same as the assumption! □

When is it Secret Against Eavesdroppers?...

Assumption 1 (Decisional DH (DDH) assumption in $\mathbb{G}$ w.r.to S $\cdots$)

$\cdots$ holds if for all PPT distinguishers D, the following is negligible:

$$\Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow \mathsf{S}(1^n)\\ a,b\leftarrow\mathbb{Z}_\ell}}[\mathsf{D}(g^a,g^b,g^{ab})=0] - \Pr_{\substack{(\mathbb{G},\ell,g)\leftarrow \mathsf{S}(1^n)\\ a,b,r\leftarrow\mathbb{Z}_\ell}}[\mathsf{D}(g^a,g^b,g^{r})=0]$$

"Real world" "Random world"

Theorem 1

Diffie-Hellman key-exchange is computationally secret against eavesdroppers under the DDH assumption in $\mathbb{G}$ w.r.to S.

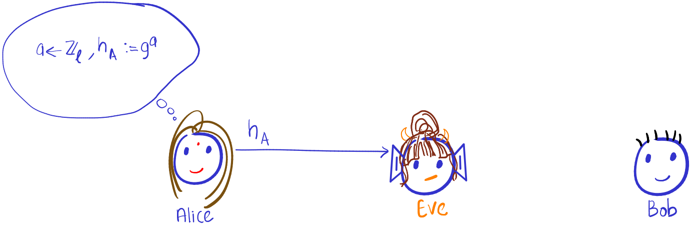
Proof.

Secrecy requirement is same as the assumption! □

Exercise 1

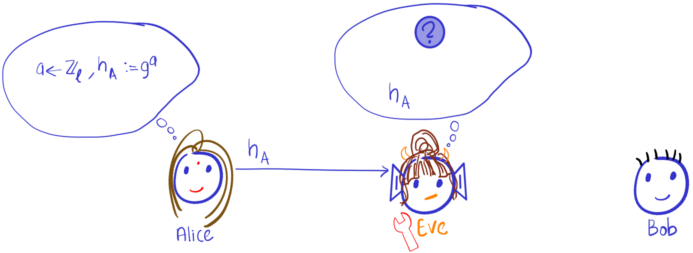
But I did slightly cheat! Figure out where.

What About Secrecy Against Active Eve?



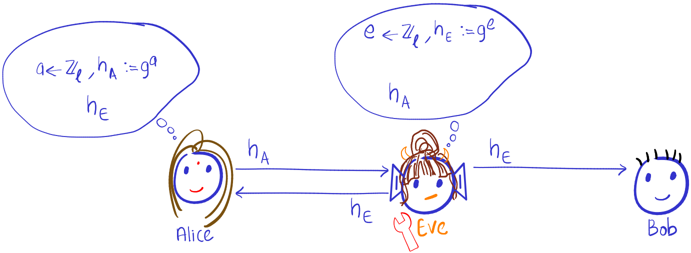
- What if Eve is an *active* adversary?
    - Recall that active Eve can intercept/tamper messages

What About Secrecy Against Active Eve?



- What if Eve is an *active* adversary?
  - Recall that active Eve can intercept/tamper messages

What About Secrecy Against Active Eve?

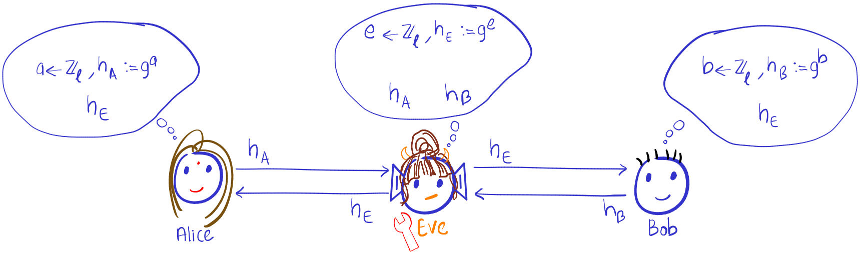


- What if Eve is an *active* adversary?
  - Recall that active Eve can intercept/tamper messages

⚠ There is a person-in-the-middle attack!

  - Pretends to be Alice to Bob and Bob to Alice
  - Eve sets up two separate key exchanges with Alice and Bob

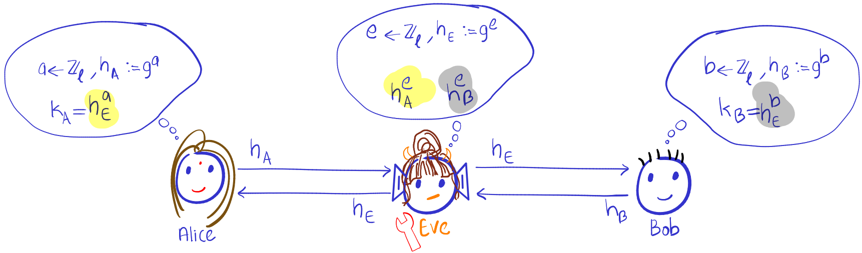
What About Secrecy Against Active Eve?



- What if Eve is an *active* adversary?
  - Recall that active Eve can intercept/tamper messages

⚠ There is a person-in-the-middle attack!
  - Pretends to be Alice to Bob and Bob to Alice
  - Eve sets up two separate key exchanges with Alice and Bob

What About Secrecy Against Active Eve?

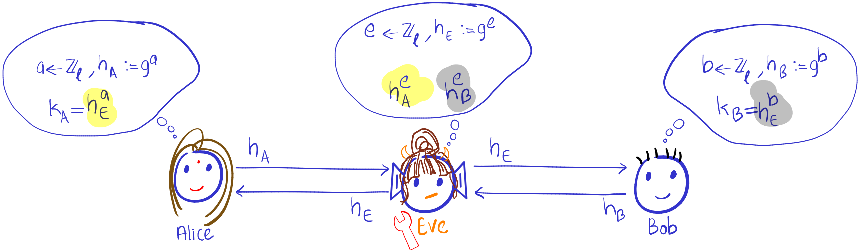


- What if Eve is an *active* adversary?
  - Recall that active Eve can intercept/tamper messages

⚠ There is a person-in-the-middle attack!

  - Pretends to be Alice to Bob and Bob to Alice
  - Eve sets up two separate key exchanges with Alice and Bob

What About Secrecy Against Active Eve?



- What if Eve is an *active* adversary?
  - Recall that active Eve can intercept/tamper messages

⚠ There is a person-in-the-middle attack!

  - Pretends to be Alice to Bob and Bob to Alice
  - Eve sets up two separate key exchanges with Alice and Bob

⚠ Insecure against active adversary

Where is DDH Assumption Known to Hold?

Addition modulo prime p

$0 = p \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} \cdots \xrightarrow{+1} i \xrightarrow{+1} \cdots \xrightarrow{+1} p-1 \xrightarrow{+1} 0$

$(\mathbb{Z}_p, +)$

$\{0, \dots, p-1\}$ $\qquad g_1 + g_2 := g_1 + g_2 \pmod p$

- order p
- cyclic

Multiplication modulo prime p

$1 = g^{p-1} \bmod p \xrightarrow{\cdot g} g^1 \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} g^i \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} 1$

$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \dots, p-1\}$ $\qquad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

- order $p-1$
- cyclic

Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

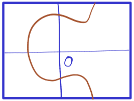
$\{1, \dots, p-1, p+1, \dots, 2p-1, \dots, pq-1\}$ $\qquad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

- order $(p-1)(q-1)$
- not cyclic

Elliptic curves modulo prime p

$(E, +)$ "curve points addition"

solutions to $y^2 = x^3 + Ax + B \pmod p$



- $|p+1 - \text{order}| \leq 2\sqrt{p}$
- cyclic

Where is DDH Assumption Known to Hold?

⚠ Easy! Since DLog is easy

Addition modulo prime p

$0 = p \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} \cdots \xrightarrow{+1} i \xrightarrow{+1} \cdots \xrightarrow{+1} p-1 \xrightarrow{+1} 0$

$(\mathbb{Z}_p, +)$

$\{0, \dots p-1\}$ $\quad g_1 + g_2 := g_1 + g_2 \pmod p$

◆ order p ◆ cyclic

Multiplication modulo prime p

$1 = g^{p-1} \bmod p \xrightarrow{\cdot g} g^1 \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} g^i \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} 1$

$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \dots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

◆ order $p-1$ ◆ cyclic

Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

$\{1, \dots p-1, p+1, \dots, 2p-1, \dots pq-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

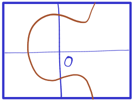
◆ order $(p-1)(q-1)$ ◆ not cyclic

Elliptic curves modulo prime p

$(E, +)$ "curve points addition"

solutions to

$y^2 = x^3 + Ax + B \pmod p$



◆ $|p+1 - \text{order}| \leq 2\sqrt{p}$ ◆ cyclic

Where is DDH Assumption Known to Hold?

Easy! Since DLog is easy

Addition modulo prime p

$0 = p \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} \cdots \xrightarrow{+1} i \xrightarrow{+1} \cdots \xrightarrow{+1} p-1 \xrightarrow{+1} 0$

$(\mathbb{Z}_p, +)$

$\{0, \dots, p-1\}$ $\quad g_1 + g_2 := g_1 + g_2 \pmod p$

- order p
- cyclic

Multiplication modulo prime p

$1 = g^{p-1} \bmod p \xrightarrow{\cdot g} g^1 \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} g^i \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} 1$

$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \dots, p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

- order $p-1$
- cyclic

Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

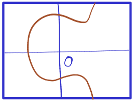
$\{1, \dots, p-1, p+1, \dots, 2p-1, \dots pq-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

- order $(p-1)(q-1)$
- not cyclic

Elliptic curves modulo prime p

$(E, +)$ "curve points addition"

solutions to $y^2 = x^3 + Ax + B \pmod p$



- $|p+1 - \text{order}| \leq 2\sqrt{p}$
- cyclic

Where is DDH Assumption Known to Hold?

⚠ Easy! Since DLog is easy

Addition modulo prime p

$0 = p \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} \cdots \xrightarrow{+1} i \xrightarrow{+1} \cdots \xrightarrow{+1} p-1 \xrightarrow{+1} 0$

$(\mathbb{Z}_p, +)$

$\{0, \ldots p-1\}$ $\quad g_1 + g_2 := g_1 + g_2 \pmod p$

◆ order p ◆ cyclic

⚠ Easy! See Assign. 4.

Multiplication modulo prime p

$1 = g^{p-1} \bmod p \xrightarrow{\cdot g} g^1 \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} g^i \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} 1$

$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \ldots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

◆ order $p-1$ ◆ cyclic ✓ Hard in subgroup

Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

$\{1, \ldots p-1,$
$p+1, \ldots, 2p-1,$
$\ldots pq-1\}$

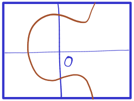
$g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

◆ order $(p-1)(q-1)$ ◆ not cyclic

Elliptic curves modulo prime p

$(E, +)$ "curve points addition"

solutions to
$y^2 = x^3 + Ax + B \pmod p$



◆ $|p+1 - \text{order}| \leq 2\sqrt{p}$ ◆ cyclic

Where is DDH Assumption Known to Hold?

⚠ Easy! Since DLog is easy

Addition modulo prime p

$0 = p \xrightarrow{+1} 1 \xrightarrow{+1} 2 \xrightarrow{+1} \cdots \xrightarrow{+1} i \xrightarrow{+1} \cdots \xrightarrow{+1} p-1 \xrightarrow{+1} 0$

$(\mathbb{Z}_p, +)$

$\{0, \dots p-1\}$ $\quad g_1 + g_2 := g_1 + g_2 \pmod p$

◆ order p ◆ cyclic

⚠ Easy! See Assign. 4.

Multiplication modulo prime p

$1 = g^{p-1} \bmod p \xrightarrow{\cdot g} g^1 \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} g^i \bmod p \xrightarrow{\cdot g} \cdots \xrightarrow{\cdot g} 1$

$(\mathbb{Z}_p^\times, \cdot)$

$\{1, \dots p-1\}$ $\quad g_1 \cdot g_2 := g_1 \cdot g_2 \pmod p$

◆ order $p-1$ ◆ cyclic ✓

Hard in subgroup

Multiplication modulo $N = pq$

primes

$(\mathbb{Z}_N^\times, \cdot)$

$\{1, \dots p-1,$
$p+1, \dots, 2p-1,$
$\dots pq-1\}$

$g_1 \cdot g_2 := g_1 \cdot g_2 \pmod N$

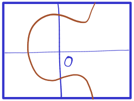
◆ order $(p-1)(q-1)$ ◆ not cyclic

Hard in its cyclic subgroup

Elliptic curves modulo prime p

$(E, +)$ "curve points addition"

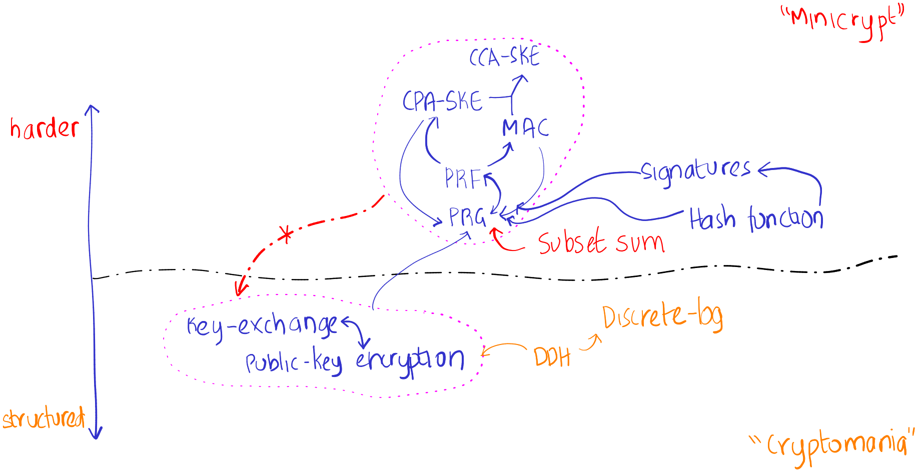
solutions to
$y^2 = x^3 + Ax + B \pmod p$



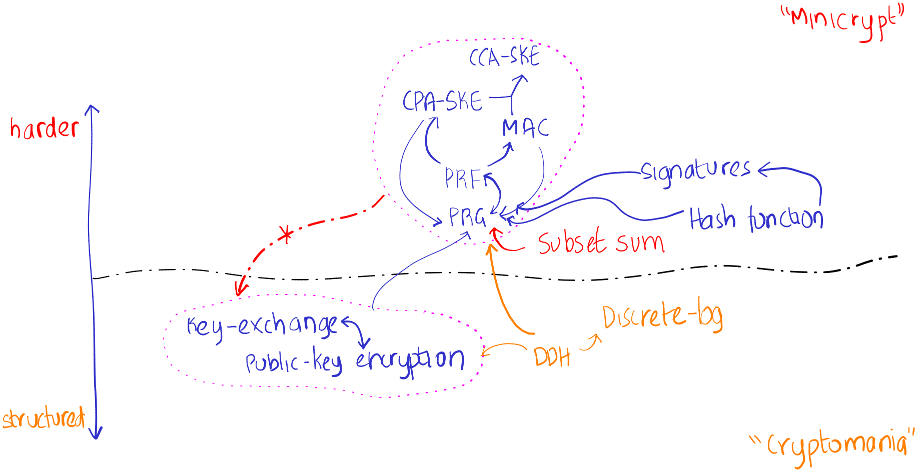
◆ $|p+1 - \text{order}| \leq 2\sqrt{p}$ ◆ cyclic

Believed hard

What Else Can be Built from DDH?



What Else Can be Built from DDH?



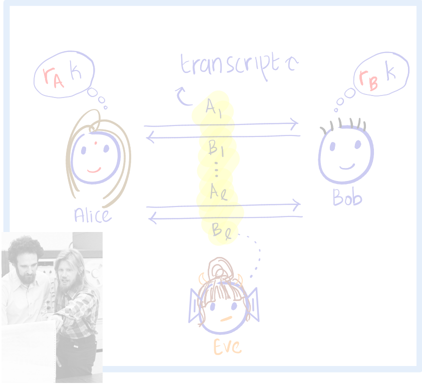
Exercise 2

Construct a PRG from DDH

Plan for Today's Lecture

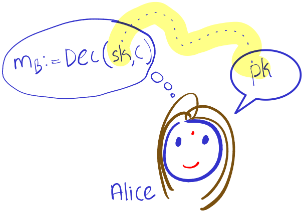
- Task: public-key encryption
- Threat model: IND-CPA

NEW Diffie-Hellman Key Exchange



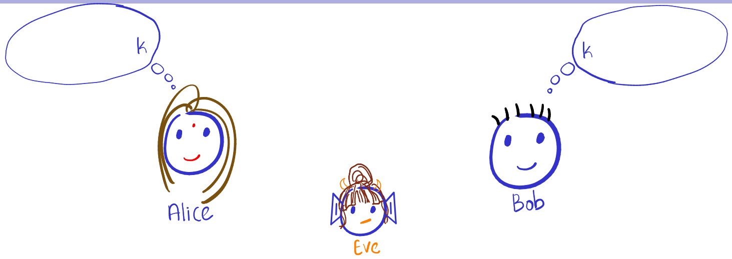
© icour.fr

NEW Public-Key Encryption

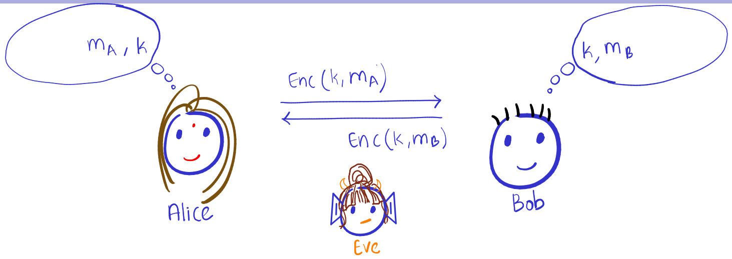


© cs.miami.edu (Rosenberg) ©Oded Goldreich ©Alexander Klink/Wikipedia

Underlying hard problem: Decisional Diffie-Hellman (DDH)

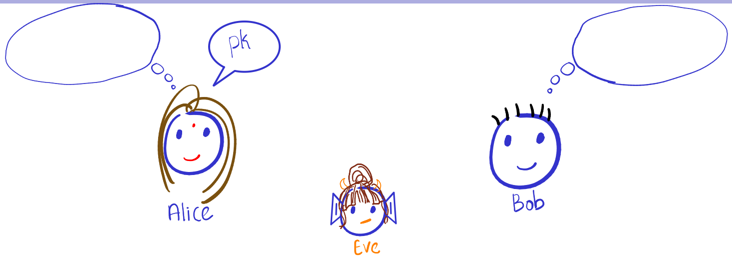


- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve



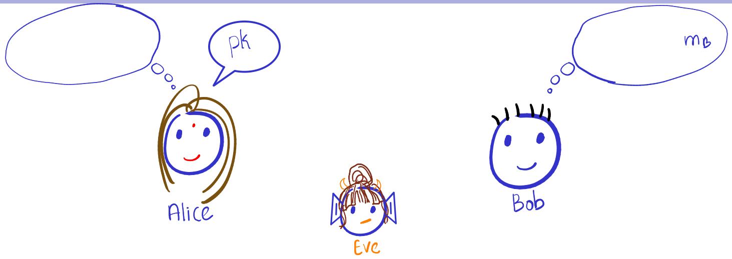
- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve

The Setting: Shared (Private) Keys vs Public Keys...

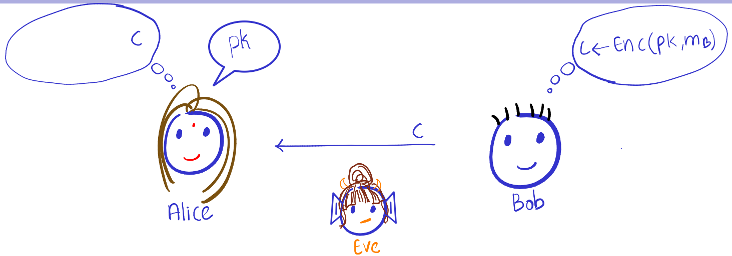


- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key pk*; known to *everyone*!
  2. Bob wants to use *pk* to secretly send a message *to* Alice in presence of Eve

The Setting: Shared (Private) Keys vs Public Keys...

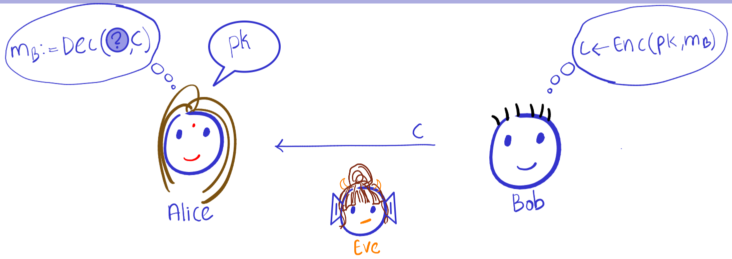


- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key pk*; known to *everyone*!
  2. Bob wants to use *pk* to secretly send a message *to* Alice in presence of Eve

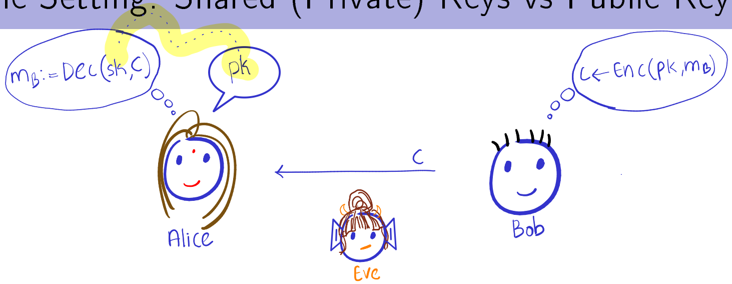


- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key pk*; known to *everyone*!
  2. Bob wants to use *pk* to secretly send a message *to* Alice in presence of Eve

The Setting: Shared (Private) Keys vs Public Keys...



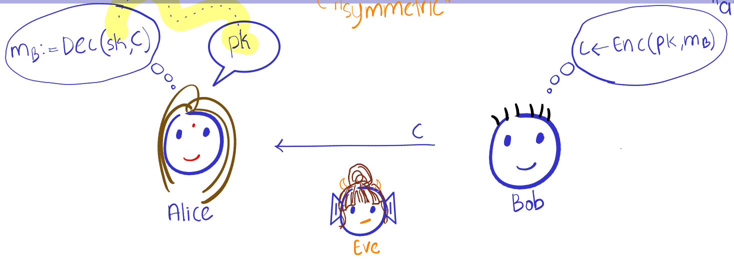
- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key pk*; known to *everyone*!
  2. Bob wants to use *pk* to secretly send a message *to* Alice in presence of Eve



- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key pk*; known to *everyone*!
  2. Bob wants to use *pk* to secretly send a message *to* Alice in presence of Eve
  3. Alice decrypts using her *secret key* *sk* (related to *pk*)

The Setting: Shared (Private) Keys vs Public Keys

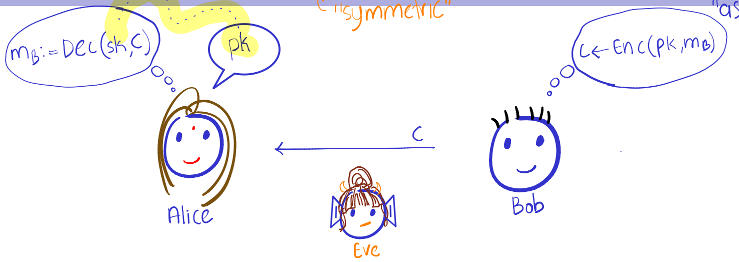
"symmetric" "asymmetric"



- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key pk*; known to *everyone*!
  2. Bob wants to use *pk* to secretly send a message *to* Alice in presence of Eve
  3. Alice decrypts using her *secret key* *sk* (related to *pk*)

The Setting: Shared (Private) Keys vs Public Keys

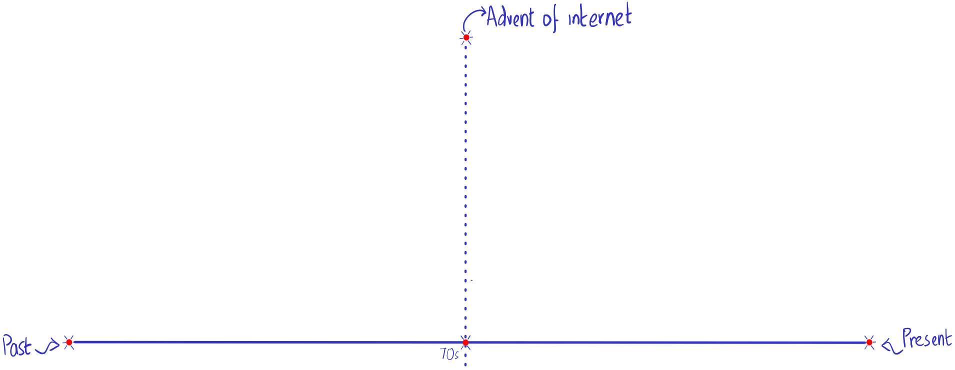
"symmetric" "asymmetric"



- Recall the SKE setting: Alice and Bob *share* $k \in \{0,1\}^n$ and want to securely communicate in presence of *eavesdropper* Eve
- The *public-key* setting:
  1. Alice announces a *public key pk*; known to *everyone*!
  2. Bob wants to use *pk* to secretly send a message *to* Alice in presence of Eve
  3. Alice decrypts using her *secret key* *sk* (related to *pk*)

+ Advantage: scalability! It suffices to have one "key" per user

The Setting: Shared (Private) Keys vs Public Keys...



The Setting: Shared (Private) Keys vs Public Keys...

Advent of internet

Programming Techniques — S. L. Graham, R. L. Rivest, Editors

Secure Communications Over Insecure Channels

Ralph C. Merkle
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

DHKE

©icour.fr

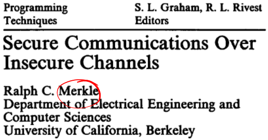
RSA cryptosystem

©cs.miami.edu/Rosenberg

Past 70s 74 76 78 Present

The Setting: Shared (Private) Keys vs Public Keys...

Advent of internet


Programming Techniques — S. L. Graham, R. L. Rivest, Editors

Secure Communications Over Insecure Channels

Ralph C. Merkle
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

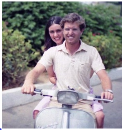
DHKE


©icour.fr

RSA cryptosystem


©cs.miami.edu/Rosenberg

Goldwasser–Micali PKE


©Oded Goldreich

Past — 70s — 74 — 76 — 78 — 82 — Present

The Setting: Shared (Private) Keys vs Public Keys...

Programming Techniques — S. L. Graham, R. L. Rivest, Editors

Secure Communications Over Insecure Channels

Ralph C. Merkle
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

THE POSSIBILITY OF SECURE NON-SECRET DIGITAL ENCRYPTION
J. H. Ellis, January 1970

DHKE

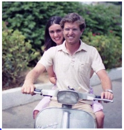
©icour.fr

Advent of internet

RSA cryptosystem

©cs.miami.edu/Rosenberg

Goldwasser–Micali PKE



©Oded Goldreich

Past → 70s 74 76 78 82 ← Present

Advent of internet

Programming Techniques — S. L. Graham, R. L. Rivest, Editors

Secure Communications Over Insecure Channels

Ralph C. Merkle
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

THE POSSIBILITY OF SECURE NON-SECRET DIGITAL ENCRYPTION
J. H. Ellis, January 1970

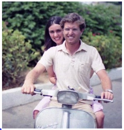
DHKE

©icour.fr

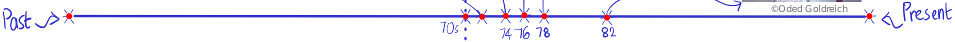
RSA cryptosystem

©cs.miami.edu/Rosenberg

Goldwasser–Micali PKE



©Oded Goldreich

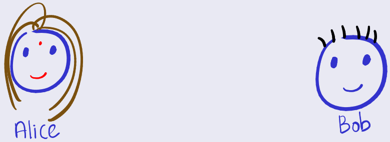


- PKE IRL: PGP, hybrid encryption

Syntax of Public-Key Encryption

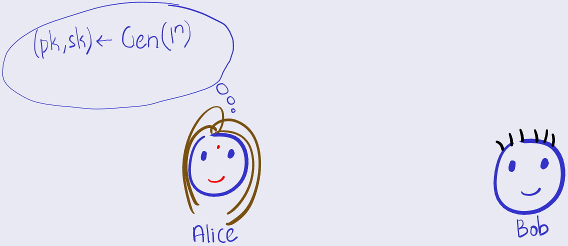
Definition 4 (Public-Key Encryption (PKE))

A PKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



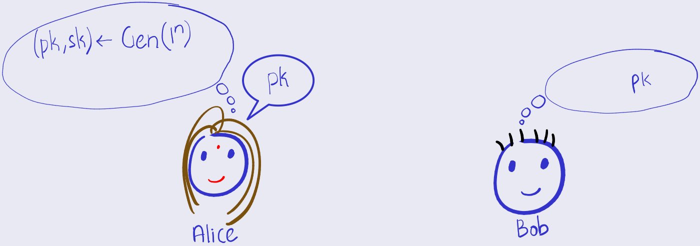
Definition 4 (Public-Key Encryption (PKE))

A PKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



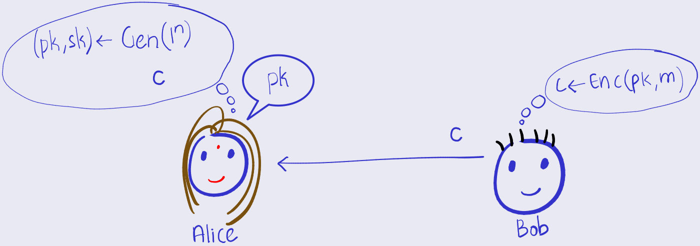
Definition 4 (Public-Key Encryption (PKE))

A PKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



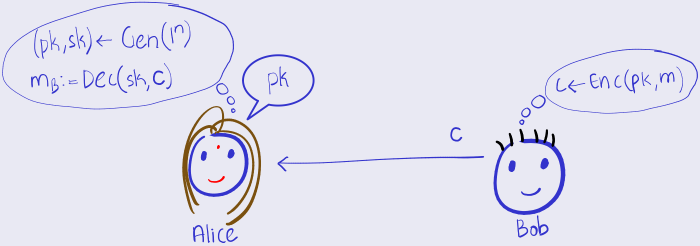
Definition 4 (Public-Key Encryption (PKE))

A PKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



Definition 4 (Public-Key Encryption (PKE))

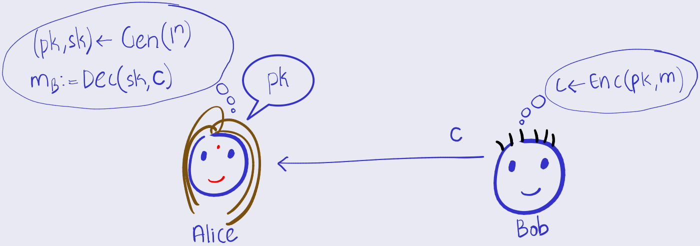
A PKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



Syntax of Public-Key Encryption

Definition 4 (Public-Key Encryption (PKE))

A PKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



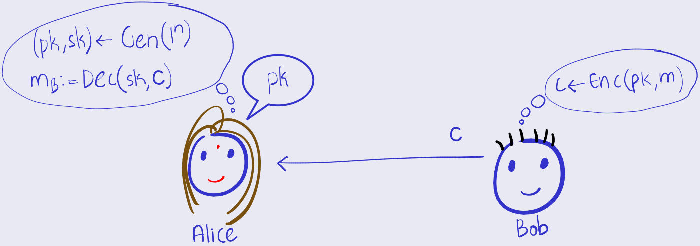
- Correctness of decryption: for every $n \in \mathbb{N}$, message $m \in \mathcal{M}_n$,

$$\Pr_{(pk,sk)\leftarrow\mathsf{Gen}(1^n),c\leftarrow\mathsf{Enc}(pk,m)}[\mathsf{Dec}(sk,c)=m]=1$$

Syntax of Public-Key Encryption

Definition 4 (Public-Key Encryption (PKE))

A PKE Π is a triple of efficient algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax:



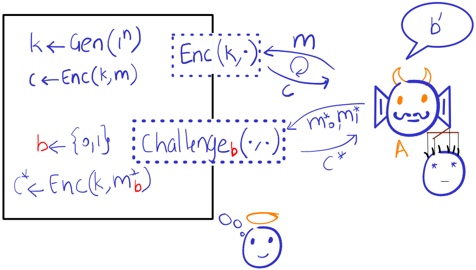
- Correctness of decryption: for every $n \in \mathbb{N}$, message $m \in \mathcal{M}_n$,

$$\Pr_{(pk,sk)\leftarrow \mathsf{Gen}(1^n),\, c\leftarrow \mathsf{Enc}(pk,m)}[\mathsf{Dec}(sk,c) = m] = 1$$

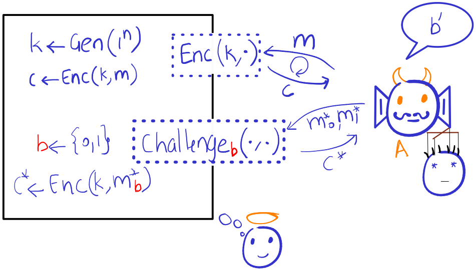
? How can an unbounded eavesdropper Eve break PKE?

How to Define Security?

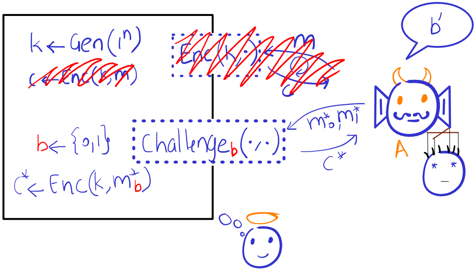
- Recall CPA-secrecy requirement in the SKE setting



- Recall CPA-secrecy requirement in the SKE setting
- What is different in the PKE setting?



- Recall CPA-secrecy requirement in the SKE setting
- ? What is different in the PKE setting?
  - The public key known to Eve ⇒ encryption oracle "redundant"



How to Define Security? . . .

- Recall CPA-secrecy requirement in the SKE setting
- What is different in the PKE setting?
  - The public key known to Eve ⇒ encryption oracle "redundant"
  - Eavesdropper=chosen-plaintext attacker!

Definition 5 (CPA Secrecy for PKE)

A PKE $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is CPA-secret if for *every* PPT (stateful) eavesdropper *Eve*, the following is negligible:

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow \mathrm{Gen}(1^n)\\(m_0,m_1)\leftarrow \mathrm{Eve}(pk)\\c\leftarrow \mathrm{Enc}(pk,m_0)}} [\mathrm{Eve}(c)=0] - \Pr_{\substack{(pk,sk)\leftarrow \mathrm{Gen}(1^n)\\(m_0,m_1)\leftarrow \mathrm{Eve}(pk)\\c\leftarrow \mathrm{Enc}(pk,m_1)}} [\mathrm{Eve}(c)=0] \right|$$

How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting
- What is different in the PKE setting?
  - The public key known to Eve ⇒ encryption oracle "redundant"
  - Eavesdropper=chosen-plaintext attacker!

Definition 5 (CPA Secrecy for PKE)

A PKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA-secret if for *every* PPT (stateful) eavesdropper *Eve*, the following is negligible:

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n)\\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk)\\ c\leftarrow \mathsf{Enc}(pk,m_0)}} [\mathsf{Eve}(c)=0] - \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n)\\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk)\\ c\leftarrow \mathsf{Enc}(pk,m_1)}} [\mathsf{Eve}(c)=0] \right|$$

"Left world" "Right world"

How to Define Security?...

- Recall CPA-secrecy requirement in the SKE setting
- What is different in the PKE setting?
  - The public key known to Eve ⇒ encryption oracle "redundant"
  - Eavesdropper=chosen-plaintext attacker!

Definition 5 (CPA Secrecy for PKE)

A PKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA-secret if for *every* PPT (stateful) eavesdropper *Eve*, the following is negligible:

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow\mathsf{Gen}(1^n)\\(m_0,m_1)\leftarrow\mathsf{Eve}(pk)\\c\leftarrow\mathsf{Enc}(pk,m_0)}}[\mathsf{Eve}(c)=0] - \Pr_{\substack{(pk,sk)\leftarrow\mathsf{Gen}(1^n)\\(m_0,m_1)\leftarrow\mathsf{Eve}(pk)\\c\leftarrow\mathsf{Enc}(pk,m_1)}}[\mathsf{Eve}(c)=0] \right|$$

"Left world" "Right world"

- Alternative, equivalent notion: semantic security
  - Ciphertext doesn't leak (non-trivial) information about plaintext

How to Define Security?

- Recall CPA-secrecy requirement in the SKE setting
- What is different in the PKE setting?
  - The public key known to Eve ⇒ encryption oracle "redundant"
  - Eavesdropper=chosen-plaintext attacker!

Definition 5 (CPA Secrecy for PKE)

A PKE $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA-secret if for *every* PPT (stateful) eavesdropper *Eve*, the following is negligible:

$$\delta(n) := \left| \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n)\\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk)\\ c\leftarrow \mathsf{Enc}(pk,m_0)}}[\mathsf{Eve}(c)=0] - \Pr_{\substack{(pk,sk)\leftarrow \mathsf{Gen}(1^n)\\ (m_0,m_1)\leftarrow \mathsf{Eve}(pk)\\ c\leftarrow \mathsf{Enc}(pk,m_1)}}[\mathsf{Eve}(c)=0] \right|$$

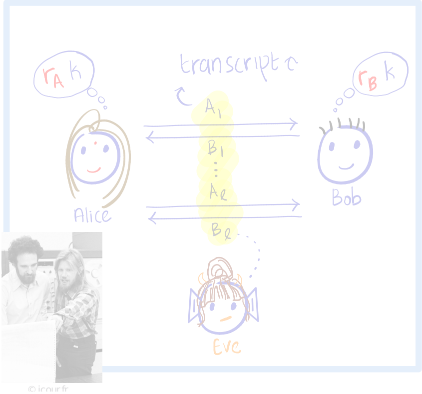
"Left world" "Right world"

- Alternative, equivalent notion: semantic security
  - Ciphertext doesn't leak (non-trivial) information about plaintext
- Stronger notion: ind. against chosen-ciphertext attack (CCA)

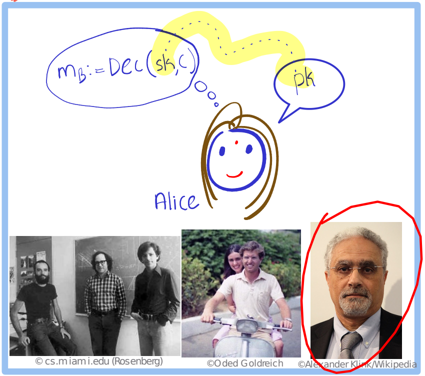
Plan for Today's Lecture

- Task: public-key encryption
- Threat model: IND-CPA

NEW Diffie-Hellman Key Exchange

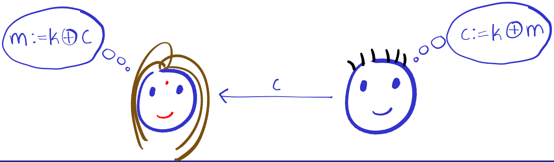


NEW Public-Key Encryption



Underlying hard problem: Decisional Diffie-Hellman (DDH)

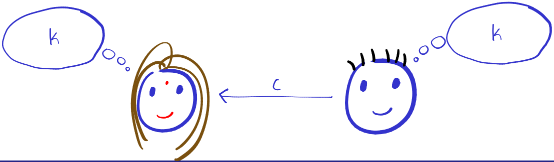
One-Time Pad Can be Generalised Over Groups



Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)

- Key generation Gen: output $k \leftarrow \{0,1\}^n$
- Encryption $\mathrm{Enc}(k,m)$: output $c := k \oplus m$
- Decryption $\mathrm{Dec}(k,c)$: output $m := k \oplus c$

One-Time Pad Can be Generalised Over Groups



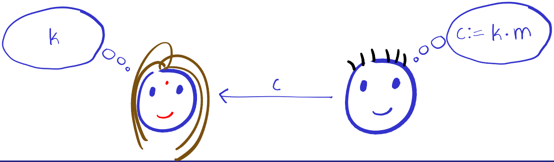
Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)

- Key generation Gen: output $k \leftarrow \{0,1\}^n$
- Encryption $\mathsf{Enc}(k, m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k, c)$: output $m := k \oplus c$

Pseudocode 2 (OTP over group $\mathbb{G} := (\mathcal{G}, \cdot)$ with message space $\mathcal{G}$)

- Key generation Gen: output $k \leftarrow \mathcal{G}$

One-Time Pad Can be Generalised Over Groups



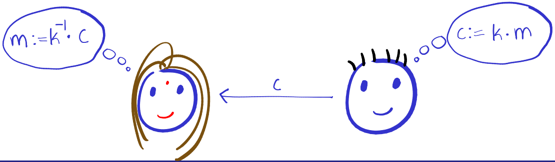
Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)

- Key generation Gen: output $k \leftarrow \{0,1\}^n$
- Encryption $\mathsf{Enc}(k, m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k, c)$: output $m := k \oplus c$

Pseudocode 2 (OTP over group $\mathbb{G} := (\mathcal{G}, \cdot)$ with message space $\mathcal{G}$)

- Key generation Gen: output $k \leftarrow \mathcal{G}$
- Encryption $\mathsf{Enc}(k, m)$: output $c := k \cdot m$

One-Time Pad Can be Generalised Over Groups



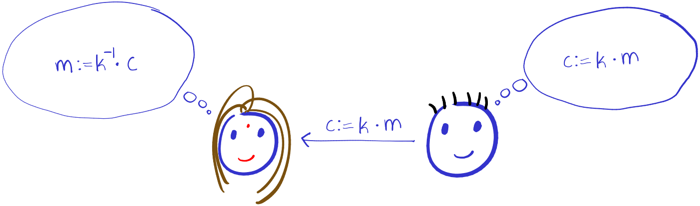
Pseudocode 1 (OTP over $(\{0,1\}^n, \oplus)$ with message space $\{0,1\}^n$)

- Key generation Gen: output $k \leftarrow \{0,1\}^n$
- Encryption $\mathsf{Enc}(k, m)$: output $c := k \oplus m$
- Decryption $\mathsf{Dec}(k, c)$: output $m := k \oplus c$

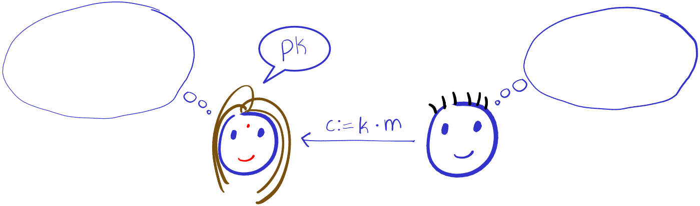
Pseudocode 2 (OTP over group $\mathbb{G} := (\mathcal{G}, \cdot)$ with message space $\mathcal{G}$)

- Key generation Gen: output $k \leftarrow \mathcal{G}$
- Encryption $\mathsf{Enc}(k, m)$: output $c := k \cdot m$
- Decryption $\mathsf{Dec}(k, c)$: output $m := k^{-1} \cdot c$

Let's Build on Group-Based OTP to Construct a PKE

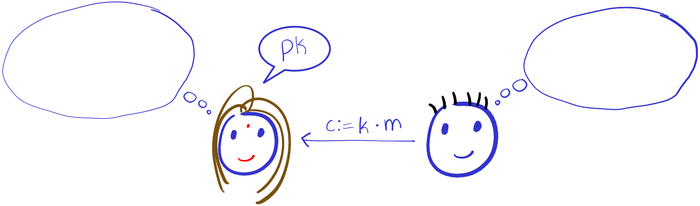


Let's Build on Group-Based OTP to Construct a PKE



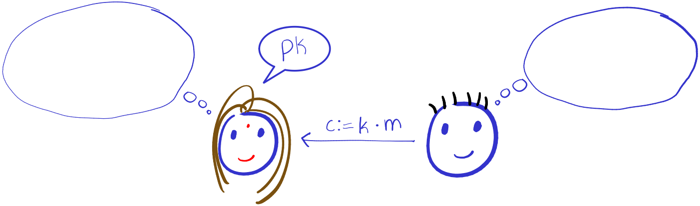
- Our ciphertexts will be of form $c := k \cdot m$

Let's Build on Group-Based OTP to Construct a PKE



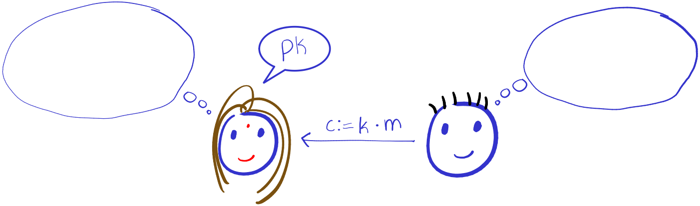
- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c

Let's Build on Group-Based OTP to Construct a PKE



- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c
- Any ideas on
  1. What can the public key pk be?
  2. How to generate k?

Let's Build on Group-Based OTP to Construct a PKE

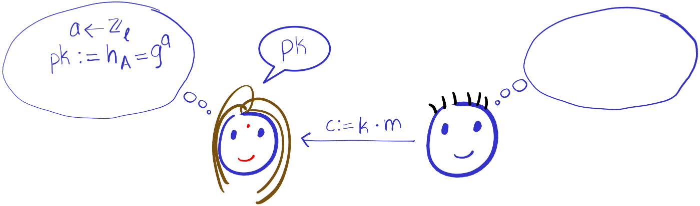


- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c

Any ideas on
  1. What can the public key pk be?
  2. How to generate k?

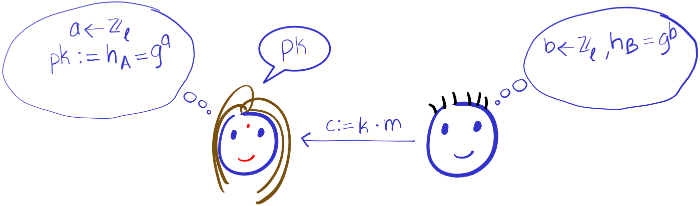
Hint: we have already exploited this "structure" in DHKE

Let's Build on Group-Based OTP to Construct a PKE



- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c
- Any ideas on
  1. What can the public key pk be?
  2. How to generate k?
- Hint: we have already exploited this "structure" in DHKE

Let's Build on Group-Based OTP to Construct a PKE

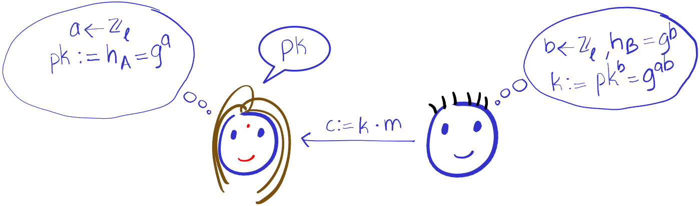


- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c

? Any ideas on
  1. What can the public key pk be?
  2. How to generate k?

Hint: we have already exploited this "structure" in DHKE

Let's Build on Group-Based OTP to Construct a PKE

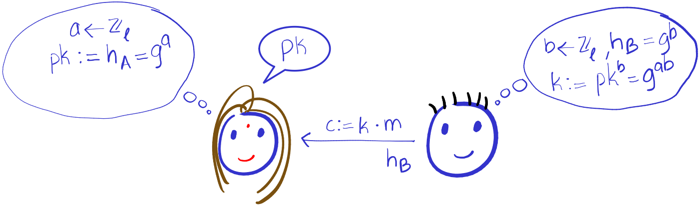


- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c

Any ideas on
1. What can the public key pk be?
2. How to generate k?

Hint: we have already exploited this "structure" in DHKE

Let's Build on Group-Based OTP to Construct a PKE

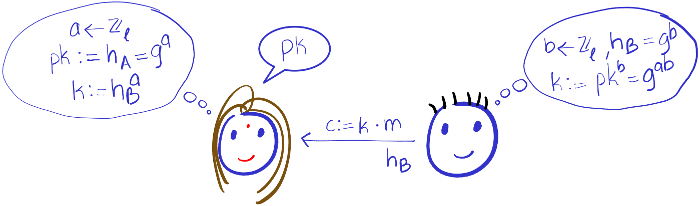


- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c

? Any ideas on
  1. What can the public key pk be?
  2. How to generate k?

Hint: we have already exploited this "structure" in DHKE

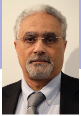
Let's Build on Group-Based OTP to Construct a PKE



- Our ciphertexts will be of form $c := k \cdot m$
- We need:
  1. Structure: two ways to generate the OTP k
  2. Eve mustn't be able to generate this k from pk and ciphertext c

Any ideas on
  1. What can the public key pk be?
  2. How to generate k?

Hint: we have already exploited this "structure" in DHKE

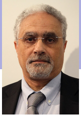


©Alexander Klink/Wikipedia

ElGamal PKE over Group $\mathbb{G}$...

Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

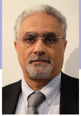
- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample group $(\mathbb{G}, \ell, g) \leftarrow \mathsf{S}(1^n)$
  2. Sample random index $a \leftarrow \mathbb{Z}_\ell$
  3. Output $(pk := g^a, sk := a)$



ElGamal PKE over Group $\mathbb{G}$ ...

Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample group $(\mathbb{G}, \ell, g) \leftarrow \mathsf{S}(1^n)$
  2. Sample random index $a \leftarrow \mathbb{Z}_\ell$
  3. Output $(pk := g^a, sk := a)$
- Encryption $\mathsf{Enc}(pk, m)$:
  1. Sample random index $b \leftarrow \mathbb{Z}_\ell$, and set $k := pk^b$
  2. Output $c := (c_1, c_2) := (k \cdot m, g^b)$

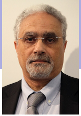


©Alexander Klink/Wikipedia

ElGamal PKE over Group $\mathbb{G}$...

Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample group $(\mathbb{G}, \ell, g) \leftarrow \mathsf{S}(1^n)$
  2. Sample random index $a \leftarrow \mathbb{Z}_\ell$
  3. Output $(pk := g^a, sk := a)$
- Encryption $\mathsf{Enc}(pk, m)$:
  1. Sample random index $b \leftarrow \mathbb{Z}_\ell$, and set $k := pk^b$
  2. Output $c := (c_1, c_2) := (k \cdot m, g^b)$
- Decryption $\mathsf{Dec}(sk, c =: (c_1, c_2))$: output $m := (c_2^{sk})^{-1} \cdot c_1$

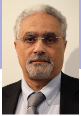


Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample group $(\mathbb{G}, \ell, g) \leftarrow \mathsf{S}(1^n)$
  2. Sample random index $a \leftarrow \mathbb{Z}_\ell$
  3. Output $(pk := g^a, sk := a)$
- Encryption $\mathsf{Enc}(pk, m)$:
  1. Sample random index $b \leftarrow \mathbb{Z}_\ell$, and set $k := pk^b = (g^a)^b = g^{ab}$
  2. Output $c := (c_1, c_2) := (k \cdot m, g^b)$
- Decryption $\mathsf{Dec}(sk, c =: (c_1, c_2))$: output $m := (c_2^{sk})^{-1} \cdot c_1$

- Correctness of decryption:

ElGamal PKE over Group $\mathbb{G}$...



Pseudocode 3 (ElGamal PKE over group $\mathbb{G} = (\mathcal{G}, \cdot)$)

- Key generation $\mathsf{Gen}(1^n)$:
  1. Sample group $(\mathbb{G}, \ell, g) \leftarrow \mathsf{S}(1^n)$
  2. Sample random index $a \leftarrow \mathbb{Z}_\ell$
  3. Output $(pk := g^a, sk := a)$
- Encryption $\mathsf{Enc}(pk, m)$:
  1. Sample random index $b \leftarrow \mathbb{Z}_\ell$, and set $k := pk^b = (g^a)^b = g^{ab}$
  2. Output $c := (c_1, c_2) := (k \cdot m, g^b)$
- Decryption $\mathsf{Dec}(sk, c =: (c_1, c_2))$: output $m := (c_2^{sk})^{-1} \cdot c_1$

  $(g^b)^a = \cancel{g^{ab}}$ $\qquad$ $\cancel{g^{ab}} \cdot m$
- Correctness of decryption:

ElGamal PKE is CPA-secret

Theorem 2 (DDH $\rightarrow$ CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

ElGamal PKE is CPA-secret

Theorem 2 (DDH → CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

ElGamal PKE is CPA-secret

Theorem 2 (DDH → CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

Left world H_0

$(pk, Enc(pk, m_0))$

Right world H_1

$(pk, Enc(pk, m_1))$

ElGamal PKE is CPA-secret

Theorem 2 (DDH $\rightarrow$ CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

Left world H_0

$(g^a, (g^b, g^{ab} \cdot m_0))$

Right world H_1

$(g^a, (g^b, g^{ab} \cdot m_1))$

ElGamal PKE is CPA-secret

Theorem 2 (DDH $\rightarrow$ CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

Left world H_0

$(g^a, (g^b, g^{ab} \cdot m_0))$

Right world H_1

$(g^a, (g^b, g^{ab} \cdot m_1))$

Hybrid world H_0'

$(g^a, (g^b, g^r \cdot m_0))$

Hybrid world H_1'

$(g^a, (g^b, g^r \cdot m_1))$

Theorem 2 (DDH → CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

Left world H_0
$(g^a, (g^b, g^{ab} \cdot m_0))$

Right world H_1
$(g^a, (g^b, g^{ab} \cdot m_1))$

Hybrid world H_0'
$(g^a, (g^b, g^r \cdot m_0))$

Hybrid world H_1'
$(g^a, (g^b, g^r \cdot m_1))$

? Why is H_0 indistinguishable from H_0' ?

ElGamal PKE is CPA-secret

Theorem 2 (DDH → CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

Left world H_0

$(g^a, (g^b, g^{ab} \cdot m_0))$

Right world H_1

$(g^a, (g^b, g^{ab} \cdot m_1))$

DDH

Hybrid world H_0'

$(g^a, (g^b, g^r \cdot m_0))$

Hybrid world H_1'

$(g^a, (g^b, g^r \cdot m_1))$

? Why is H_0 indistinguishable from H_0' ? DDH assumption

ElGamal PKE is CPA-secret

Theorem 2 (DDH → CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

Left world H_0

$(g^a, (g^b, g^{ab} \cdot m_0))$

DDH

Hybrid world H_0'

$(g^a, (g^b, g^r \cdot m_0))$

Right world H_1

$(g^a, (g^b, g^{ab} \cdot m_1))$

DDH

Hybrid world H_1'

$(g^a, (g^b, g^r \cdot m_1))$

? Why is H_0/H_1 indistinguishable from H_0'/H_1'? DDH assumption

ElGamal PKE is CPA-secret

Theorem 2 (DDH → CPA-PKE)

ElGamal PKE is CPA-secret under DDH assumption in $\mathbb{G}$ w.r.to S.

Proof sketch. Hybrid argument.

Four hybrids H_0, H_0', H_1', H_1

Left world H_0

$(g^a, (g^b, g^{ab} \cdot m_0))$

Right world H_1

$(g^a, (g^b, g^{ab} \cdot m_1))$

DDH

DDH

Hybrid world H_0'

$(g^a, (g^b, g^r \cdot m_0))$

OTP

Hybrid world H_1'

$(g^a, (g^b, g^r \cdot m_1))$

? Why is H_0/H_1 indistinguishable from H_0'/H_1'? DDH assumption

? Why is H_0' indistinguishable from H_1'? OTP over group

Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE $\rightarrow$ CPA-PKE)

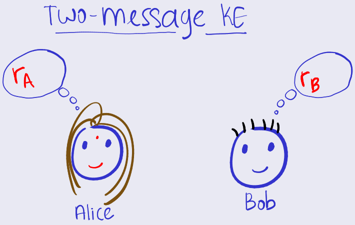
If two-message key exchange protocol Π exists then so does PKE.

Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE $\rightarrow$ CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1

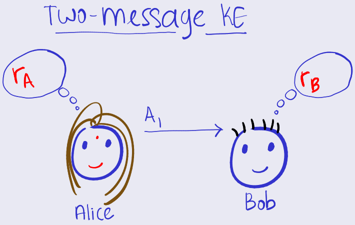


Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE $\rightarrow$ CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1

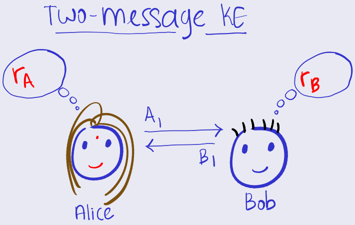


Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE → CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1

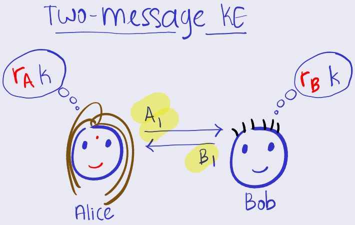


Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE $\rightarrow$ CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1

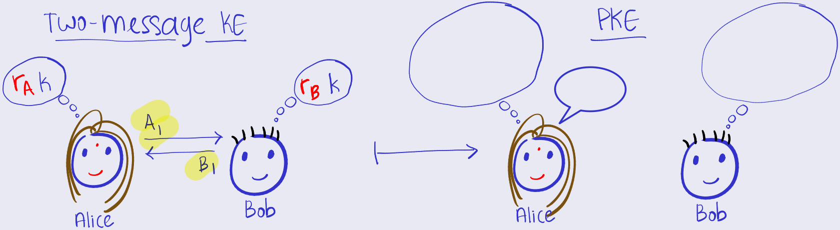


Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE → CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1

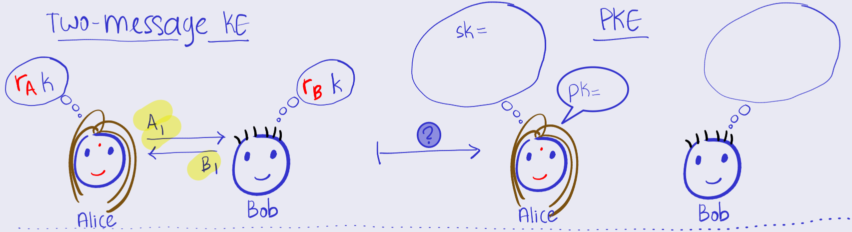


Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE → CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1



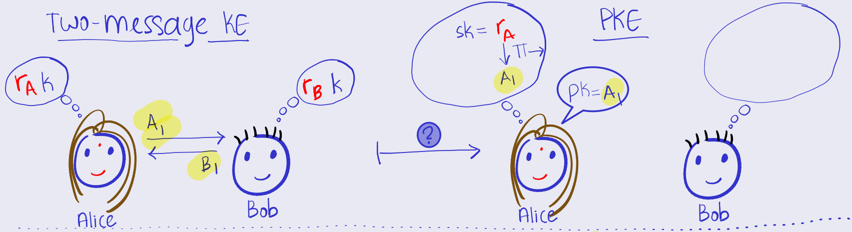
? How does Alice generate (pk,sk)?

Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE → CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1



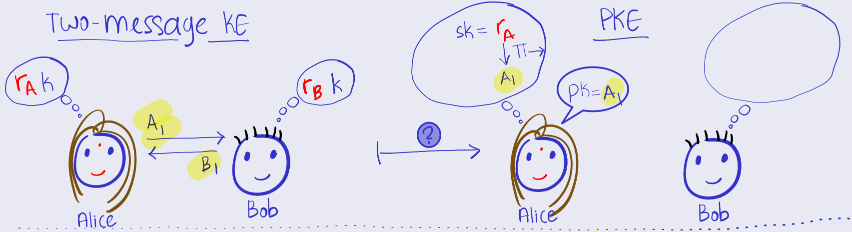
? How does Alice generate (pk, sk)? Generate first message A_1 using coins r_A

Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE → CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1



? How does Alice generate (pk, sk)? Generate first message A_1 using coins r_A

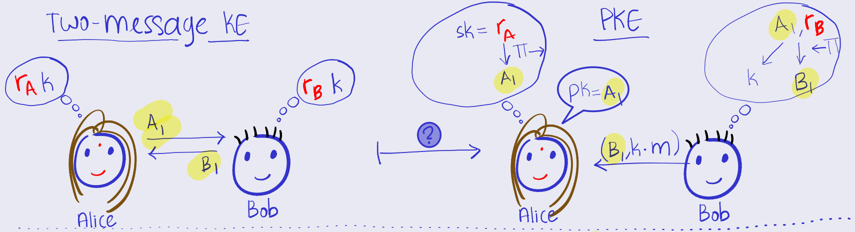
? How does Bob encrypt?

Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE → CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1



? How does Alice generate (pk, sk)? Generate first message A_1 using coins r_A

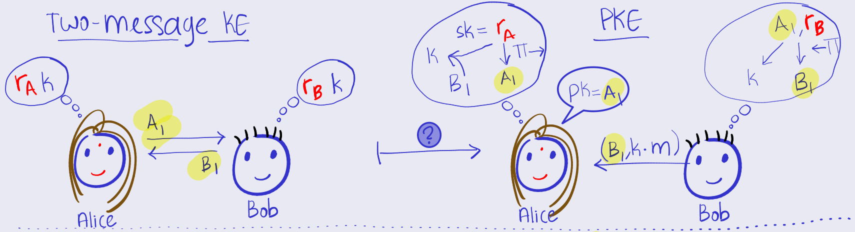
? How does Bob encrypt pt? Generate 2nd message B_1 and shared key k using coins r_B and A_1. Use k as OTP; send B_1 to help decryption.

Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE → CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1



? How does Alice generate (pk, sk)? Generate first message A_1 using coins r_A

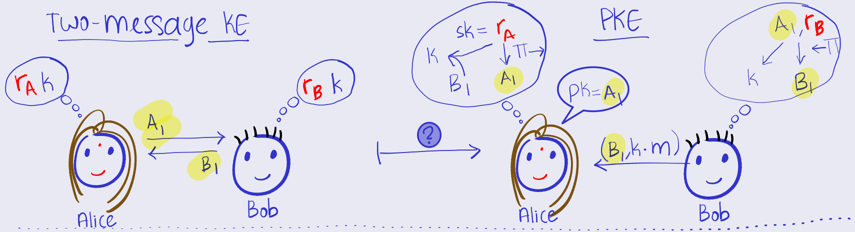
? How does Bob encrypt? Generate 2^{nd} message B_1 and shared key k using coins r_B and A_1. Use k as OTP; send B_1 to help decryption.

Is it a Coincidence that ElGamal is Similar to DHKE? No!

Claim 1 (Two-message KE $\rightarrow$ CPA-PKE)

If two-message key exchange protocol Π exists then so does PKE.

Construction 1



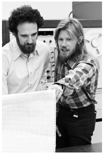
? How does Alice generate (pk, sk)? Generate first message A_1 using coins r_A

? How does Bob encrypt? Generate 2^{nd} message B_1 and shared key k using coins r_B and A_1. Use k as OTP; send B_1 to help decryption.

Exercise 3 (Converse to Claim 1: two-message KE $\leftarrow$ CPA-PKE)

If PKE exists then so does two-message key exchange.

Recap/Next Lecture

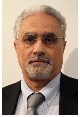


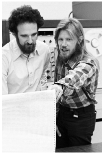
© icour.fr

- Diffie-Hellman key exchange (DHKE)
  - Based on DDH assumption in cyclic groups
  - Algebraic *structure* exploited: $(g^a)^b = g^{ab} = (g^b)^a$

Recap/Next Lecture

©Alexander Klink/Wikipedia



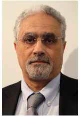


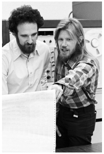
© icour.fr

© icour.f

- Diffie-Hellman key exchange (DHKE)
  - Based on DDH assumption in cyclic groups
  - Algebraic *structure* exploited: $(g^a)^b = g^{ab} = (g^b)^a$

- Public-key encryption (PKE)
  - Equivalent to two-round KE
  - Derived Elgamal PKE from DHKE





- Diffie-Hellman key exchange (DHKE)
  - Based on DDH assumption in cyclic groups
  - Algebraic *structure* exploited: $(g^a)^b = g^{ab} = (g^b)^a$
- Public-key encryption (PKE)
  - Equivalent to two-round KE
  - Derived Elgamal PKE from DHKE
- Next lecture:
  - Factoring and related hardness assumptions
  - RSA group: multiplicative group modulo $N := pq$
  - Goldwasser-Micali encryption
  - RSA encryption

$a \leftarrow \mathbb{Z}_\ell$
$pk := h_A = g^a$
$k := h_B^a$

pk

$c := k \cdot m$

h_B

Key-exchange

Public-key encryption

DDH

Factoring

QR

References

1. [KL14, Chapter 11] for more details on key exchange
2. Read the seminal paper by Diffie and Hellman [DH76] for a description of the namesake key-exchange. In general this paper is a very insightful read.
3. Boneh's survey [Bon98] is an excellent source on the DDH problem.

Dan Boneh.
The decision diffie-hellman problem.
In *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 1998.

Whitfield Diffie and Martin E. Hellman.
New directions in cryptography.
IEEE Trans. Inf. Theory, 22(6):644–654, 1976.

Jonathan Katz and Yehuda Lindell.
Introduction to Modern Cryptography (3rd ed.).
Chapman and Hall/CRC, 2014.