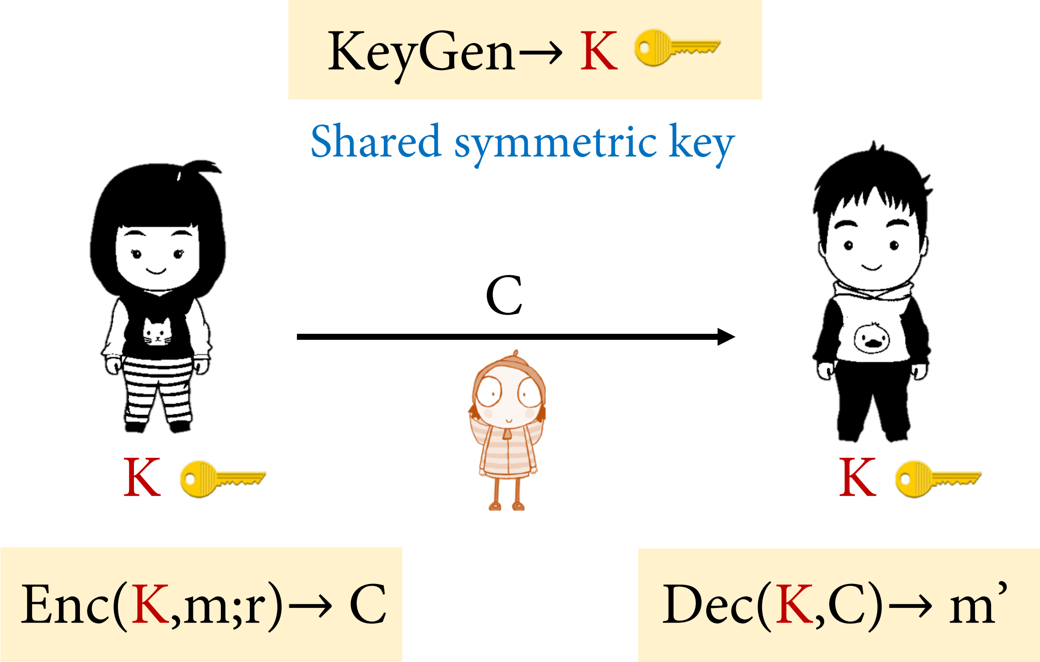
PUBLIC KEY ENCRYPTION

Lecture I

ACM Summer School 2024

Symmetric Key Encryption (SKE) : Recall

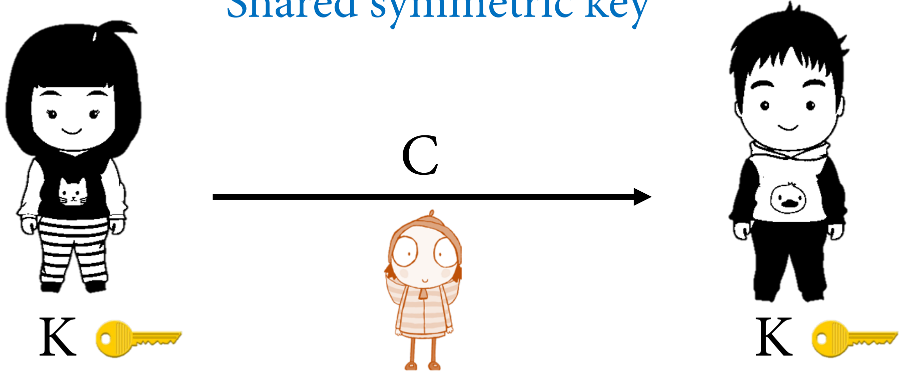


- Correctness: ∀ K ∈ Range(KeyGen), m’ = m
- IND-CPA Security
  (against eavesdropping adversary Eve)

Asymmetric/Public Key Encryption (PKE)

KeyGen→ K

Shared symmetric key



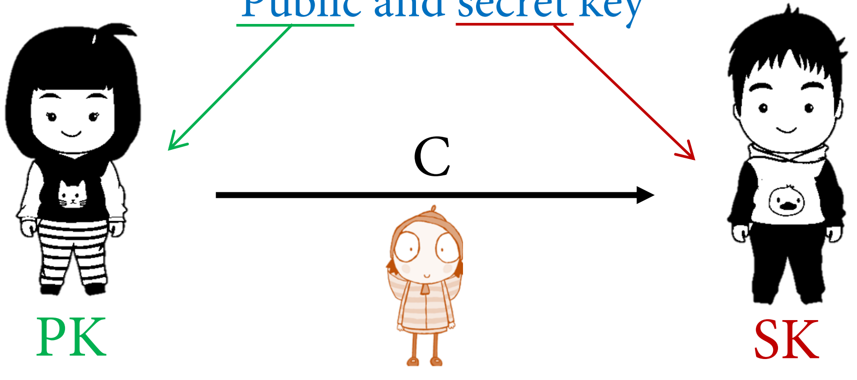
Enc(K,m;r)→ C

Dec(K,C)→ m’

- Correctness: ∀ K ∈ Range(KeyGen), m’ = m
- IND-CPA Security
(against eavesdropping adversary Eve)

KeyGen→ (PK,SK)

Public and secret key



Enc(PK,m;r)→ C

Dec(SK,C)→ m’

- Correctness: ∀ PK,SK ∈ Range(KeyGen), m’ = m
- IND-CPA Security for PKE
(against eavesdropping adversary Eve)

IND-CPA Security (SKE): Recall

Challenger

Adversary

KeyGen(λ) $\rightarrow$ K

$\mathrm{Enc}_K(.)$ $\longleftrightarrow$ Oracle access $\longleftrightarrow$ As many ciphertexts as it wants

$b \leftarrow \{0,1\}$ $\xleftarrow{m_0, m_1}$ $m_0 \neq m_1$

$\mathrm{Enc}(K, m_b) \rightarrow C_b$ $\xrightarrow{C_b}$

Output b'

IND-CPA Security (Chosen Plaintext Attack)

$\forall$ PPT adversaries, $\Pr[b' = b] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$

IND-CPA Security for PKE

Challenger | Adversary 

$\text{KeyGen}(\lambda) \rightarrow K$

$\text{Enc}_K(.)$ ~~Oracle access~~ As many ciphertexts as it wants

m_0, m_1 ($m_0 \neq m_1$)

$b \leftarrow \{0,1\}$

$\text{Enc}(K, m_b) \rightarrow C_b$ $\quad C_b$

Output b'

IND-CPA Security (Chosen Plaintext Attack)

$\forall$ PPT adversaries, $\Pr[b' = b] \leq \frac{1}{2} + \text{negl}(\lambda)$

Challenger | Adversary 

$\text{KeyGen}(\lambda) \rightarrow PK, SK$

PK — Can itself generate ciphertexts

m_0, m_1 ($m_0 \neq m_1$)

$b \leftarrow \{0,1\}$

$\text{Enc}(PK, m_b) \rightarrow C_b$ $\quad C_b$

Output b'

IND-CPA Security (Chosen Plaintext Attack)

$\forall$ PPT adversaries, $\Pr[b' = b] \leq \frac{1}{2} + \text{negl}(\lambda)$

Perfect secrecy for PKE?

$\forall$ ~~PPT~~ adversaries, $\Pr[b' = b] = \frac{1}{2}$

Impossible to get perfectly secret PKE

- Any unbounded adversary, given PK and a ciphertext $C \leftarrow \mathrm{Enc}(PK, m)$, can determine m with probability 1.

Challenger

Adversary

$\mathrm{KeyGen}(\lambda) \rightarrow PK, SK$

$\xrightarrow{PK}$

Can itself generate ciphertexts

$\xleftarrow{m_0, m_1}$

$m_0 \neq m_1$

$b \leftarrow \{0,1\}$

$\mathrm{Enc}(PK, m_b) \rightarrow C_b$

$\xrightarrow{C_b}$

Output b'

EXERCISE 1

Deterministic PKE?
(where Enc is not a randomized algorithm)

EXAMPLE OF AN IND-CPA SECURE PKE

Groups

- Group $(\mathbb{G},*)$ consists of set $\mathbb{G}$ and operation $*$ that is:
  Associative, has an identity, is invertible, and additionally (for us) commutative.

abelian

- Order of a group $|\mathbb{G}|$: number of elements in $\mathbb{G}$.

EXAMPLES

$\mathbb{Z}$ =(Integers, +)	Identity 0	Inverse of x is $-$x	Infinite order
$\mathbb{Z}_n$ =(Integers modulo n, + mod n)	Identity 0	Inverse of x is (n − x)	Order n
$\mathbb{Z}_5^*$ =({1,2,3,4}, × mod 5)	Identity 1	Inverses ?	Order 4

x mod 5 such that gcd(x, 5) = 1

For general $\mathbb{Z}_n^*$: Extended Euclidean Algorithm

EXERCISE 2

Groups

- Group $(\mathbb{G},*)$ consists of set $\mathbb{G}$ and operation $*$ that is:
  Associative, has an identity, is invertible, and additionally (for us) commutative.

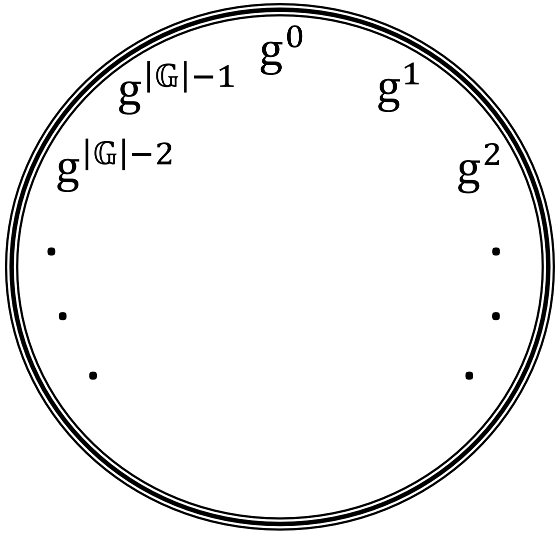
  abelian

- Order of a group $|\mathbb{G}|$: number of elements in $\mathbb{G}$.
- Lagrange's Theorem:

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

- Finite Cyclic Group (in multiplicative notation):

  $\exists\, g \in \mathbb{G}$ such that $\mathbb{G} = \{g^0, g^1, \ldots, g^{|\mathbb{G}|-1}\}$



$\mathbb{Z}_n$(additive group): generator $g = 1$

$\mathbb{Z}_5^*$(multiplicative group): generator?

EXERCISE 3

Computing on Groups

Need efficient algorithms to generate and operate on groups:

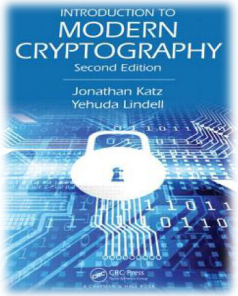
Need an ensemble of groups indexed by λ

1. Generating group: Need an efficient algorithm that, given λ, outputs a description of a cyclic group $\mathbb{G}$, along with its order $|\mathbb{G}|$ and its generator g.

2. Description of a group: This specifies how elements of $\mathbb{G}$ are represented as bit-strings, with each group element having a unique bit representation.

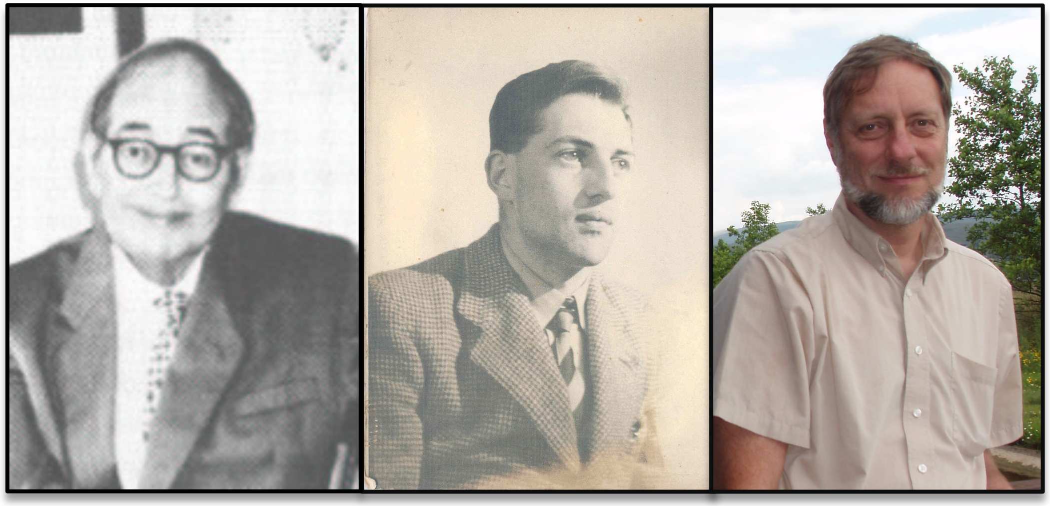
3. Efficient operations on group elements: There must be a polynomial time algorithm for adding, inverting, and randomly sampling a group element. Given generator g, there must be an efficient exponentiation algorithm to compute g^x.

Introduction to Modern Cryptography, Katz and Lindell
(Appendix B)



Advent of Public Key Cryptography

Non-secret encryption

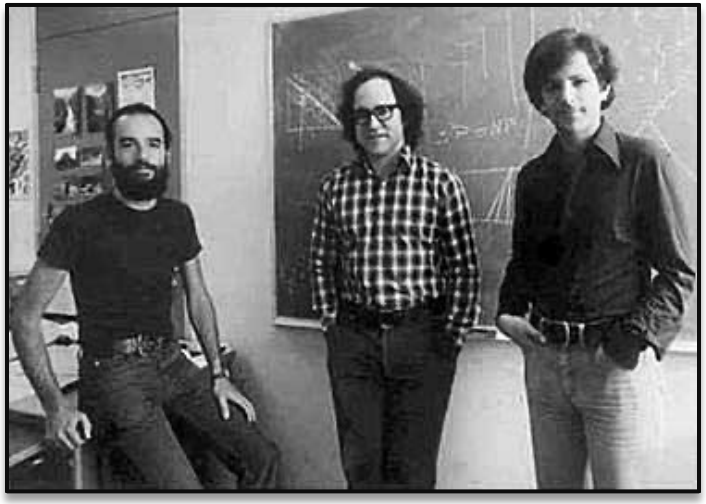


James Ellis

Clifford Cocks

Malcolm Williamson

RSA

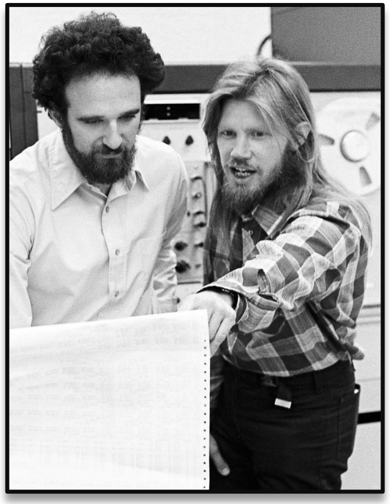


Adi Shamir

Ronald Rivest

Leonard Adleman

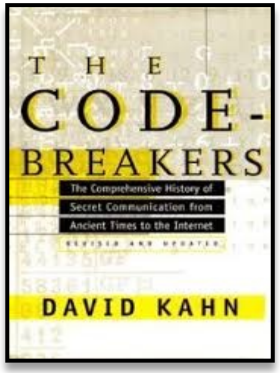
Diffie-Hellman



Martin E. Hellman

Whitfield Diffie

The collaborative work of Clifford Cocks, James Ellis, and Malcolm Williamson at GCHQ resulted in the discovery of public key cryptography (PKC) in the early 1970s. Even though outside researchers subsequently made similar discoveries, the UK's GCHQ did not make it public until 1997. –*National Security Agency (NSA)*



Diffie-Hellman Key-exchange [1976]

How can Alice and Bob communicate via an *insecure channel* and *generate a shared secret key*?

Generate $(\mathbb{G}, q, \mathrm{g})$

Group Order Generator

$\mathrm{x} \leftarrow \mathbb{Z}_q$

$\mathrm{X} = \mathrm{g}^{\mathrm{x}}$

$(\mathbb{G}, q, \mathrm{g}), \mathrm{X} \longrightarrow$

$\longleftarrow \mathrm{Y}$

$\mathrm{y} \leftarrow \mathbb{Z}_q$

$\mathrm{Y} = \mathrm{g}^{\mathrm{y}}$

$\mathrm{Key}_{\mathrm{Alice}} = \mathrm{Y}^{\mathrm{x}}$

$\mathrm{Key}_{\mathrm{Bob}} = \mathrm{X}^{\mathrm{y}}$

Depends on the group!

Given g^{x} and g^{y} for random x and y, g^{xy} should be hidden from Eve, i.e. $(\mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{g}^{\mathrm{xy}}) \approx_c (\mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}, \mathrm{R})$

Discrete Log Assumption

Discrete Log (w.r.t. g): $DL_g(X) \coloneqq$ unique x such that $X = g^x$

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

Group Order Generator

$x, g \overset{\text{Efficient}}{\underset{\text{exponentiation}}{\Longrightarrow}} g^x$

EXERCISE 4

$g^x, g \overset{?}{\Longrightarrow} x$

Challenger

Adversary

$\mathrm{Gen}(1^\lambda) \longrightarrow (\mathbb{G}, q, g)$

$X \leftarrow \mathbb{G}$

$\xrightarrow{(\mathbb{G}, q, g), X}$

$\xleftarrow{x}$

Output: 1, if $g^x = X$
0, otherwise

Discrete Log Assumption

$\forall$ PPT Adversaries, $\Pr[\text{Output} = 1] \leq \mathrm{negl}(\lambda)$

Diffie-Hellman Key-exchange [1976]

Discrete Log Broken $\Longrightarrow$ DH Key-exchange broken

If Discrete log assumption is broken for $(\mathbb{G}, q, \mathrm{g})$

Eve gets x, y from $\mathrm{g}^{\mathrm{x}}, \mathrm{g}^{\mathrm{y}}$ and hence can compute g^{xy}

Group Order Generator

$\mathrm{x} \leftarrow \mathbb{Z}_q$

$\mathrm{X} = \mathrm{g}^{\mathrm{x}}$

$\xrightarrow{(\mathbb{G}, q, \mathrm{g}), \mathrm{X}}$

$\xleftarrow{\mathrm{Y}}$

$\mathrm{y} \leftarrow \mathbb{Z}_q$

$\mathrm{Y} = \mathrm{g}^{\mathrm{y}}$

$\mathrm{Key}_{\mathrm{Alice}} = \mathrm{Y}^{\mathrm{x}}$

$\mathrm{Key}_{\mathrm{Bob}} = \mathrm{X}^{\mathrm{y}}$

Decisional Diffie-Hellman Assumption

$$\{(g^x, g^y, g^{xy})\}_{(\mathbb{G},q,g)\leftarrow \mathrm{Gen}(1^\lambda),\, x,y\leftarrow \mathbb{Z}_q} \approx_c \{(g^x, g^y, g^r)\}_{(\mathbb{G},q,g)\leftarrow \mathrm{Gen}(1^\lambda),\, x,y,r\leftarrow \mathbb{Z}_q}$$

EXERCISE 5: Proof by reduction

Claim: Decisional Diffie-Hellman (DDH) assumption $\Longrightarrow$ Discrete Log (DLog) assumption

Dlog assumption $\Longrightarrow$ DDH assumption?

No! E.g.: In $\mathbb{Z}_p^*$ for prime p,
DLog assumption is believed to hold but DDH assumption doesn't hold!

Diffie-Hellman Key-exchange [1976]

DDH Assumption $\Longleftrightarrow$ DH Key-exchange secure against Eve

(Eve's transcript = $((\mathbb{G}, q, g), X, Y)$, Key = g^{xy}) $\approx_c$ (Eve's transcript, Random key)

Group Order Generator

$x \leftarrow \mathbb{Z}_q$

$X = g^x$

$\xrightarrow{(\mathbb{G}, q, g), X}$

$\xleftarrow{Y}$

$y \leftarrow \mathbb{Z}_q$

$Y = g^y$

$\text{Key}_{\text{Alice}} = Y^x$

$\text{Key}_{\text{Bob}} = X^y$

ElGamal Encryption [Taher ElGamal 1985]

KeyGen: Uses Gen to get $(\mathbb{G}, q, \mathrm{g})$, $\mathrm{x} \leftarrow \mathbb{Z}_q$

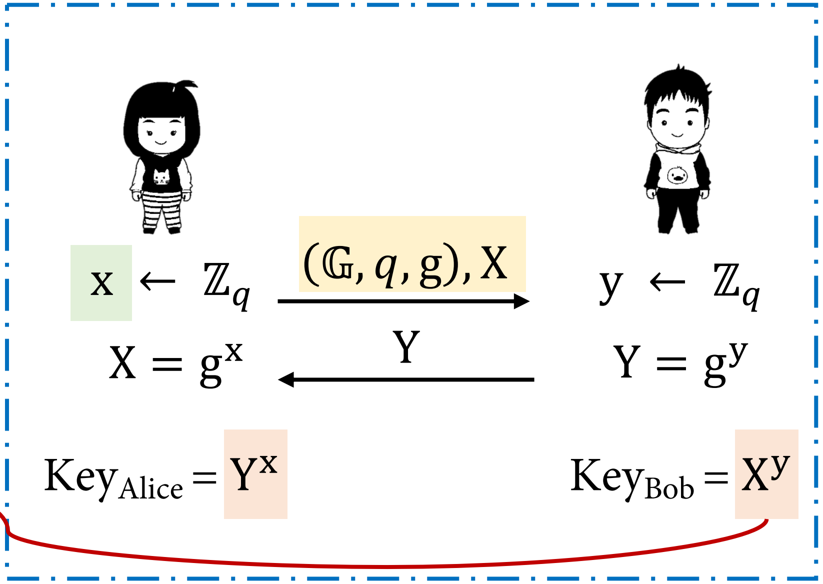
$\mathrm{PK} = (\mathbb{G}, \mathrm{g}, \mathrm{X} = \mathrm{g}^{\mathrm{x}})$, $\mathrm{SK} = (\mathbb{G}, \mathrm{g}, \mathrm{x})$

Enc(PK, m): $\mathrm{y} \leftarrow \mathbb{Z}_q$

$(\mathrm{Y} = \mathrm{g}^{\mathrm{y}}, \mathrm{C} = \mathrm{m}\mathrm{X}^{\mathrm{y}})$

Dec(SK, C): $\mathrm{C}\mathrm{Y}^{-\mathrm{x}}$

One-time pad for messages in the group



- Alice's message X in the key exchange becomes her public key.

- Bob's message Y in the key exchange and the ciphertext of the one-time pad C form the final ciphertext of the encryption.

IND-CPA Security of ElGamal Encryption

THEOREM:

If DDH Assumption holds for the collection of groups used, then ElGamal is IND-CPA Secure.

PROOF:

DDH Adversary A^* (acts as IND-CPA Challenger)

IND-CPA Adversary A

Gets challenge $(\mathbb{G}, q, g), g^x, g^y, g^z$
where $(\mathbb{G}, q, g) \leftarrow \mathrm{Gen}(1^\lambda), x, y \leftarrow \mathbb{Z}_q$
and $z = xy$ or $z \leftarrow \mathbb{Z}_q$

- Set $PK = (\mathbb{G}, q, g), g^x$
- $b \leftarrow \{0,1\}$
- $C_b = (g^y, m_b g^z)$

$A^* \xrightarrow{PK} A$

$A \xrightarrow{m_0, m_1} A^*$

$A^* \xrightarrow{C_b} A$

$A \xrightarrow{b'} A^*$

If $b' = b$ output 1, else 0

When $z \leftarrow \mathbb{Z}_q$
A^* outputs 1 w.p. $\frac{1}{2}$

When $z = xy$
(Exactly IND-CPA experiment)
A^* outputs 1 w.p. $\frac{1}{2} + \mathrm{Advantage}_A$

∎

Coming up in Part II

- Security against Chosen ciphertext attacks (CCA) in PKE.
- Elgamal and CCA Security
- Random Oracle model
- Hybrid Encryption
- Example of CCA Secure PKE