10.1 Public-Key Encryption

- In private key encryption, two parties agree on a secret key K which is used for encryption and decryption

- In comparison, public key encryption schemes, the receiver generates a pair of keys, (PK, SK), called public and private key respectively.

- Public key is used by sender to encrypt a message to receiver
- Private key is used by receiver to decrypt a message from sender.

- Generally public keys are public knowledge. They can be posted to a public repository. Anyone with the public key is considered a legitimate party.

- Public key encryption is called asymmetric encryption b/c sender and receiver are not interchangeable. Public key encryption only allows communication in one direction.

Public Key Encryption

1. Solves Key distribution Problem
   - No need to store a Key in advance of their communication.

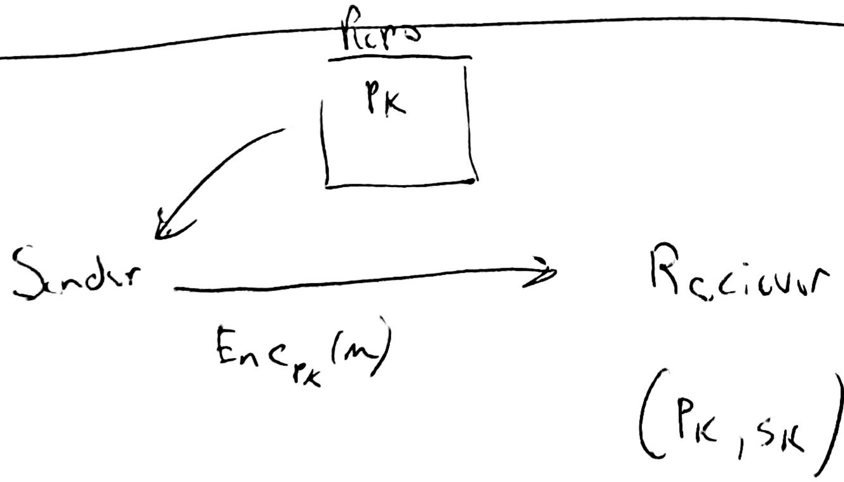
2. 2x-3x orders of Magnitude slower than Private Key encryption.

3. Used mostly in Online transcations where advance communication has not occured. Example: Credit Card Transactions.

Assumption:

1. We assume Adversaries do not alter Key distributions. (However this is a solvable problem.)

2. We assume senders have a legitimate copy of receivers public Key.



<u>Def.</u> A public key encryption scheme Π is a tuple of probabilistic, PT algorithms (Gen, Enc, Dec) where

① $Gen(\,)$

- input: 1^n
- output: (pk, sk) where $||pk||, ||sk|| \geq n$

② $Enc_k(m)$

- Input: pk, message m
- output: ciphertext c

③ Dec

- input: sk, ciphertext c
- output: m

<u>Correctness</u>

For all n, every (pk, sk) output by $Gen(1^n)$ and every message m, it holds that

$$Dec_{sk}(Enc_{pk}(m)) = m$$

<u>Assume.</u> We want message space to be $\{0,1\}^n$. However some message spaces will be missing some strings.

Security against CPA

A public-key encryption $\Pi = (Gen, Enc, Dec)$ has indistinguishable encryptions in the presence of an eavesdropper if for all PPT adversaries A, there exists a negl func negl s.t.

$$\Pr\left[PubK^{eav}_{A,\Pi}(n) = 1 \right] \leq \frac{1}{2} + negl(n)$$

Where $Pub^{eav}_{A,\Pi}(n)$:

1. $Gen(1^n)$ outputs (pk, sk)
2. A is given pk and outputs a pair of messages m_0, m_1 with $|m_0| = |m_1|$
3. $b \leftarrow \{0,1\}$ and ciphertext $c \leftarrow Enc_{pk}(m_b)$ is computed and given to A.
4. A outputs a bit b'
5. Output is 1 if $b' = b$, OW 0 is output

Where probability is taken over random coins used by A, gen, and b.

Notes

1. Public Key encryption schemes are never perfectly secret.

   Example:

   Given challenge cipher C, an adversary could encrypt every message in $\mathcal{M}$ to find cipher C, since time is unbounded.

2. No deterministic public key encryption scheme has indistinguishable encryptions in presence of eavesdropper.

   Example

   Given challenge cipher C, A could encrypt (m_0, m_1) and determine which is encrypted to be C.

3. If Π has indistinguishable encryptions in the presence of a eavesdropper, then Π has indistinguishable multiple encryptions in the presence of an eavesdropper.

   Ex. Π can securely encrypt vectors of messages.

Notes.

(1) No Oracle is needed. Attacker has P_K so can be encrypted by attacker.

(2.) This definition is equivalent to CPA Security

Def. Public-Key encryption $\pi = (Gen, Enc, Dec)$ has indistinguishable encryptions under CPA if for all PPT adversaries A, there exists a negl s.t:

$$Pr\left[PubK^{CPA}_{A,\pi}(n) = 1 \right] \leq \frac{1}{2} + negl(n)$$

Where $PubK^{CPA}_{A,\pi}(n)$ is the same as $PubK_{A,\pi}(n)$ except A has oracle access.

Proposition If public key encryption π has indistinguishable encryptions in the presence of an eavesdropper then it is also CPA Secure.

Encrypting Arbitrary-Length Messages (Note continued

(II) Given a fixed length message scheme that is Secure we can obtain a public key encryption for arbitrary-length messages.

Suppose $\Pi = (Gen, Enc, Dec)$ is an encryption scheme where the message space is $\{0,1\}$.

Construct $\Pi' = (Gen, Enc', Dec')$ with message space $\{0,1\}^*$ and

$$Enc'_{pk}(m) = Enc_{pk}(m_1) \cdots Enc_{pk}(m_t).$$

where $m = m_1 \cdots m_t$.

Hybrid encryption

- To improve the efficiency of public key encryption we can combine private key encryption with public key.

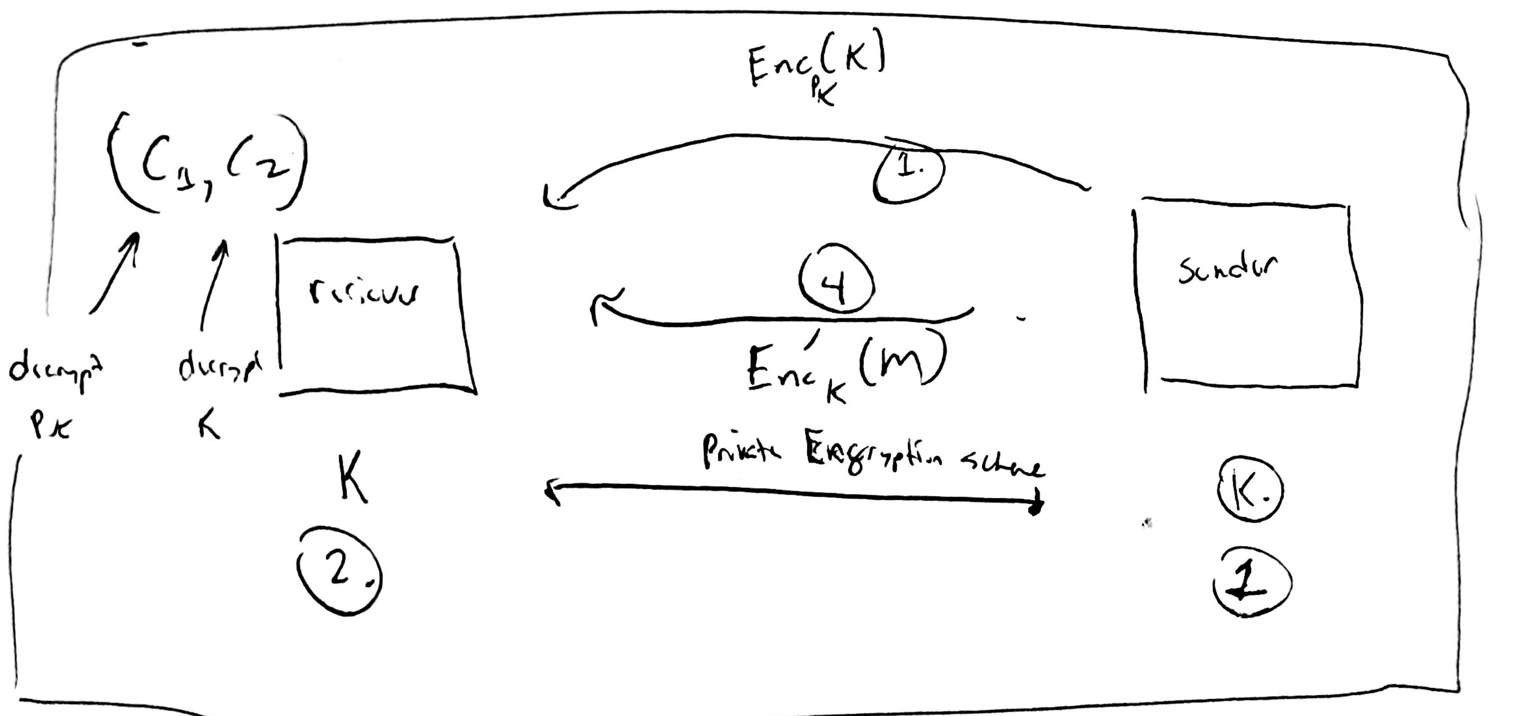
Intuition

1. Sender chooses a random secret key K, and encrypts K using the public key of the reciever. Call it C_1

2. reciever will decrypt K using their secret key

3. Sender and reciever now share a private key K

4. Sender and reciever can now use a private key encryption with key K. Sender sends message m_2 and reciever obtains ciphertext C_2.



Construction.

Let $\Pi = (Gen, Enc, Dec)$ be a public key-encryption scheme and let $\Pi' = (Enc', D')$ be a private key-encryption scheme.

Then $\Pi^H = (Gen^H, Enc^H, Dec^H)$ is:

- $Gen\ H = Gen$
- $Enc^{hy}_{PK}(m)$:
  1. $K \leftarrow \{0,1\}^n$ where n is determined by $|PK|$
  2. Compute $c_1 \leftarrow Enc_{PK}(k)$ and $c_2 \leftarrow Enc'_K(m)$
  3. output $\langle c_1, c_2 \rangle$
- $Dec^H_{SK}(\langle c_1, c_2 \rangle)$:
  1. Compute $K := Dec_{SK}(c_1)$
  2. Output $m := Dec'_K(c_2)$

Why do we care about hybridization

- Allows us to achieve functionality of public key encryption at the efficiency of private key encryption

Example

1. have public key encryption deal with

2. Have

- There two schemes

  - Outer scheme - Public key - Keys invariant

  - Inner scheme - Private key - Key changes with each message

- If Π is a CPA secure scheme and Π' is a private key scheme that has indistinguishable encryptions in the presence of a eavesdropper, then Π^{hy} is a CPA-secure encryption scheme

EL Gamal Encryption Scheme

Our first and last public key encryption scheme!

1. Security is based on hardness of DDH problem.
2. We will start by proving a helpful lemma.

Lemma 10.18

$|G| < \infty$ and $m \in G$. Then choosing $g \leftarrow G$ uniformly and setting $g' = m \cdot g$ gives the same distribution for g' as choosing $g \leftarrow G$.

$$\Pr[m \cdot g = g'] = 1/|G|$$

Where the probability is taken over a random choice of G.

Proof. $\hat{g} \in G$. Then $\Pr[m \cdot g = \hat{g}] = \Pr[g = m^{-1} \cdot \hat{g}]$

g is chosen uniformly, the prob that g is equal to a fixed element $m^{-1} \cdot g$ is exactly $1/|G|$

The intuition of El Gamal follows from the Lemma

Intuition of El Gamal

$$\text{Sender} \xrightarrow[g \cdot m]{} \text{Reciever}$$

g $\qquad\qquad\qquad\qquad\qquad$ g

$$\frac{g \cdot m}{g}$$

1. Both sender and reciever share a random element
2. Encryption is done by multiplying the message with m
3. Decryption is done by reciever using inverse of group elements
4. This is equivalent one time pad!

Let $\mathcal{G}$ be a polynomial-time Algorithm. Lets construct the El Gamal construction!

Input: 1^n

Output: (G, q, g)

El Gamal Construction

- $Gen(1^n)$ runs $\mathcal{G}(1^n)$ to obtain (G, q, g) and chooses $x \leftarrow \mathbb{Z}_q$. Public Key is $\underline{\langle G, q, g, g^x \rangle}$ and the private Key is $\underline{\langle G, q, g, x \rangle}$.

- $Enc_{pk}(m)$ is as follows:

  Choose a random element $y \leftarrow \mathbb{Z}_q$ and output the cipher text

  $$\underline{\langle g^y, h^y \cdot m \rangle} = \langle c_1, c_2 \rangle, \quad h^y = (g^x)^y$$

- $Dec_{SK}(c)$ is as follows:

  Use $SK = \langle G, q, g, x \rangle$ to compute

  $$m := c_2 / c_1^x$$

Thm. If the DDH problem is hard relative to $\mathcal{G}$, then EL Gamal Encryption is CPA-Secure.

Proof Let Π denote the El Gamal encryption scheme.

1. Let A be a PPT adversary.
2. Define $\varepsilon(n) = \Pr[\mathrm{PubK}^{cav}_{A,\Pi}(n) = 1]$
3. Consider the following scheme $\tilde{\Pi}$.

$\widetilde{Gen} = Gen$

$\widetilde{Enc}$ of a message m w/r to pk $\langle G, q, g, h \rangle$ is done by choosing $y \leftarrow \mathbb{Z}_q$ and $z \leftarrow \mathbb{Z}_q$ and outputting

$$\langle g^y, g^z \cdot m \rangle$$

Not really an encryption scheme. But exp is still defined

4. By corollary, $g^z m$ is a uniformly-distributed group element. The ~~encryption~~ element is independent of m being encrypted,
5. g^y is also independent of m.
6. Hence, the ciphertext is independent of m.
7. Therefore $\Pr[\mathrm{PubK}^{cav}_{A,\tilde{\Pi}}(n) = 1] = 1/2$ since $\widetilde{Enc}$ is equivalent to one time pad.

Now consider D that tries to solve DDH relative to $\mathcal{G}$

Algorithm D: Algorithm is given G, q, g, g_1, g_2, g_3

(1) set $pk = \langle G, q, g, g_1 \rangle$ run $A\langle pk \rangle$ and obtain m_0, m_1.

(2) Choose a random bit b, and set $c_1 = g_2$, $c_2 = g_3 m_b$

(3) Give $\langle c_1, c_2 \rangle$ to A and obtain b'.

(4) If $b' = b$ output 1 OW output 0.

Case 1

$$\boxed{\begin{array}{l}\text{Suppose D is run and obtains } (G, q, g). \\ \text{Then chooses uniformly } x, y, z \in \mathbb{Z}_q \text{, and sets} \\ g_1 = g^x,\ g_2 = g^y,\ g_3 = g^z.\end{array}}$$ 1+2

Then D runs A on a public key $pk = \langle G, q, g, g^x \rangle$ and a ciphertext is

$$\langle c_1, c_2 \rangle = \langle g^y, g^z \cdot m_b \rangle$$ 3

Therefore

$$\Pr[D(G,q,g,g^x,g^y,g^z=1))] = \Pr(PubK^{cov}_{A,\tilde{\Pi}}(n)=1) = 1/2$$

$\boxed{\text{Since } D \text{ is running } A.}$

Case 2

Suppose D is run and generates (G,q,g).

Chooses a random element $x,y \in \mathbb{Z}_q$ and sets

$negl(n) \geq -\frac{1}{2} + \varepsilon(n)$

$g_1 = g^x$, $g_2 = g^y$, and $g_3 = g^{xy}$. Then D runs

$negl(n) + 1/2 \geq \varepsilon(n)$

A on a public key. $pk = \langle G,q,g,g^x \rangle$ and

a ciphertext $\langle c_1, c_2 \rangle = \langle g^y, g^{xy} \cdot m_b \rangle = \langle g^y, (g^x)^y m_b \rangle$. is given to A

$$\Pr[D(G,q,g,g^x,g^y,g^{xy})=1] = \Pr(PubK^{cov}_{A,\Pi}(n)=1] = \varepsilon(n)$$

Since DDH is hard relative to $\mathcal{G}$, there $\exists$ a negl function s.t.

$$negl(n) = |\Pr(D(G,q,g,g^x,g^y,g^z)=1] - \Pr[D(G,q,g,g^x,g^y,g^{xy})]|$$

$$= |1/2 - \varepsilon(n)| \implies \varepsilon(n) \leq 1/2 + negl(n)$$