Course: Modern Cryptography

DL and Factorisation based Public-Key Encryption Schemes

Shashank Singh

IISER Bhopal

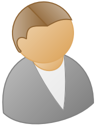
October 30, 2025

ElGamal Encryption Scheme

m

alice

$\langle G, q, g, g^x, x \rangle \leftarrow \mathrm{GEN}(n)$



$sk := x$

$pk := g^x$

bob

ELGAMAL ENCRYPTION SCHEME

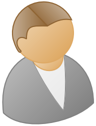
$\langle G, q, g, g^x, x\rangle \leftarrow \mathtt{GEN}(n)$

m

$y \leftarrow \mathbb{Z}/q\mathbb{Z}$

alice

$(c_1, c_2) := (g^y, mg^{xy})$



$sk := x$

$pk := g^x$

bob

Dec: $m = c_2\,(c_1{}^x)^{-1}$

- Let $\mathcal{G}$ be a polynomial-time algorithm that takes as input n and (except possibly with negligible probability) outputs a description of a cyclic group G, its order q (with $\|q\| \approx n$), and a generator g.

- Now we formally describe ElGamal encryption scheme as $(\texttt{GEN}, \texttt{ENC}, \texttt{DEC})$.

Syntax of The ElGamal Encryption Scheme

- GEN: on input 1^n, run $\mathcal{G}(1^n)$ to obtain (G, q, g). Then choose a uniform $x \in \mathbb{Z}/q\mathbb{Z}$ and compute $h := g^x$. The public key is $\langle G, q, g, h \rangle$ and the private key is $\langle G, q, g, x \rangle$. The message space is G.
- ENC: on input a public key $pk = \langle G, q, g, h \rangle$ and a message $m \in G$, choose a uniform $y \in \mathbb{Z}/q\mathbb{Z}$ and output the ciphertext $\langle g^y, m \cdot h^y \rangle$.
- DEC: on input a private key $pk = \langle G, q, g, x \rangle$ and a ciphertext (c_1, c_2), output $\hat{m} = c_2 \cdot ({c_1}^x)^{-1}$.

Theorem

If the DDH (Decisional Diffie Hellman) problem is hard relative to G, then the ElGamal encryption scheme is CPA-secure.

DDH-Based Key Encapsulation

Let $\mathcal{G}$ be as defined. Define a KEM as follows:

- `GEN`: on input n run $\mathcal{G}(n)$ to obtain (G, q, g). Choose a uniform $x \in \mathbb{Z}/q\mathbb{Z}$ and set $h := g^x$. Also specify a function $H : G \mapsto \{0,1\}^{\ell n}$ for some function ℓ. Set $pk := \langle G, q, g, h, H \rangle$ and $sk := \langle G, q, g, x \rangle$.
- `ENCAPS`: on input a public key $pk := \langle G, q, g, h, H \rangle$, choose a uniform $y \in \mathbb{Z}/q\mathbb{Z}$ and output the ciphertext g^y and the key $H(h^y)$.
- `DECAPS`: on input a private key $sk := \langle G, q, g, x \rangle$ and a ciphertext $c \in G$, output the key $H(c^x)$.

Theorem

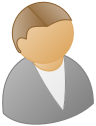
If the DDH problem is hard relative to G, and H is modeled as a random oracle, then the above Construction is a CPA-secure KEM.

Plain RSA Encryption Scheme

- choose primes p and q s.t., $p \approx q \approx 2^{2048}$.
- Let $N := p \cdot q;\ \phi(N) = (p-1) \cdot (q-1)$.
- Choose $e > 1$ s.t $\gcd(e, \phi(N)) = 1$.
- Compute $d := \left[e^{-1} \mod \phi(N)\right]$.

m

alice



bob

$sk := \langle N, d \rangle$

$pk := \langle N, e \rangle$

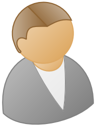
Plain RSA Encryption Scheme

- choose primes p and q s.t., $p \approx q \approx 2^{2048}$.
- Let $N := p \cdot q$; $\phi(N) = (p-1) \cdot (q-1)$.
- Choose $e > 1$ s.t $\gcd(e, \phi(N)) = 1$.
- Compute $d := \left[e^{-1} \mod \phi(N)\right]$.

m

alice

ENC $c := m^e \pmod N$



bob

$sk := \langle N, d \rangle$
$pk := \langle N, e \rangle$

DEC $m = c^d \pmod N$

Formal description of plain RSA algorithm is given below

Algorithm 1: RSA key generation `genRSA`(n)

Input: Security Parameter n

Output: N, e, d

$N, p, q \leftarrow$ `genModulus`(n)

`/* It is a PPT algorithm which outputs` (N, p, q) `where` $N = pq$`, and` p `and` q `are` n`-bit primes except with probability negligible in` n`. */`

$\phi(N) := (p-1) \cdot (q-1)$

Choose $e > 1$ such that $\gcd(e, \phi(N)) = 1$.

Compute $d := \left[e^{-1} \mod \phi(N)\right]$.

return N, e, d

Plain RSA public-key encryption scheme

- `GEN`: on input n run `genRSA`(n) to obtain N, e, and d. The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$
- `ENC`: on input a public key $pk = \langle N, e\rangle$ and a message $m \in \mathbb{Z}/N\mathbb{Z}$, compute the ciphertext

$$c := [m^e \bmod N]$$

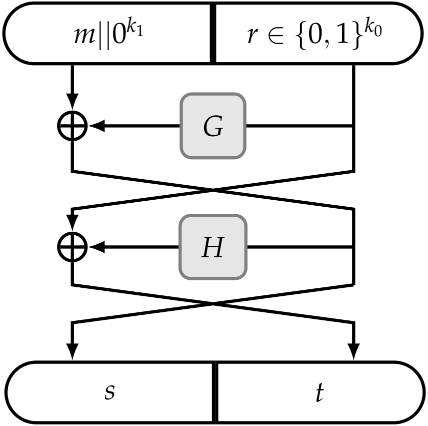
- `DEC`: on input a private key $sk = \langle N, d\rangle$ and a ciphertext $c \in \mathbb{Z}/N\mathbb{Z}$, compute the message

$$m := \left[c^d \bmod N\right]$$

- Plain RSA is not even CPA-secure.

RSA-OAEP

OAEP- Optimal Asymmetric Encryption Padding



- The OAEP transformation is a two-round Feistel network with G and H as round functions.
- First set $m' := m||0^{k_1}$ and choose a uniform $r \in \{0,1\}^{k_0}$. Then compute

  $$s := m' \oplus G(r) \text{ and } t := r \oplus H(s)$$

  and set $\hat{m} = s||t$.

- Let $\ell(n), k_0(n), k_1(n)$ be integer-valued functions with $k_0(n), k_1(n) = \Theta(n)$ and such that $\ell(n) + k_0(n) + k_1(n)$ is less than the minimum bit-length of moduli output by $\texttt{genRSA}(1^n)$.

- Let $G : \{0,1\}^{k_0} \mapsto \{0,1\}^{\ell+k_1}$ and $H : \{0,1\}^{\ell+k_1} \mapsto \{0,1\}^{k_0}$. be functions.

THE RSA-OAEP ENCRYPTION SCHEME

- `GEN`: on input n run `genRSA`(n) to obtain N, e, and d. The public key is $\langle N, e\rangle$ and the private key is $\langle N, d\rangle$
- `ENC`: on input a public key $pk = \langle N, e\rangle$ and a message $m \in \mathbb{Z}/N\mathbb{Z}$, first set $m' := m||0^{k_1}$ and choose a uniform $r \in \{0,1\}^{k_0}$. Then compute $s := m' \oplus G(r)$ and $t := r \oplus H(s)$ and set $\hat{m} = s||t$. Compute the ciphertext

$$c := [\hat{m}^e \bmod N]$$

- `DEC`: on input a private key $sk = \langle N, d\rangle$ and a ciphertext $c \in \mathbb{Z}/N\mathbb{Z}$, compute the message $\hat{m} := \left[c^d \bmod N\right]$. If $\|\hat{m}\| > \ell + k_0 + k_1$, output $\perp$. Otherwise, parse $\hat{m}$ as $\langle s, t\rangle$. Compute $r := H(s) \oplus t$ and $m' := G(r) \oplus s$. If the k_1 lsbs of m' are not all 0, output $\perp$. Otherwise, output the ℓ msb's of m'.

RSA based CCA-secure KEM

Let `genRSA` be as usual, and construct a KEM as follows:

- `GEN`: $(N, e, d) \leftarrow \texttt{genRSA}(1^n)$. Set $pk = \langle N, e \rangle$ and $sk = \langle N, d \rangle$. A function $H : \mathbb{Z}/N\mathbb{Z}^* \mapsto \{0, 1\}^n$ is also specified, but we leave this implicit.
- `ENCAPS`: on input $pk = \langle N, e \rangle$ and 1^n, choose a uniform $r \in \mathbb{Z}/N\mathbb{Z}^*$ and output $c := r^e \bmod N$ and $k := H(r)$.
- `DECAPS`: on input $sk = \langle N, d \rangle$ and a ciphertext $c \in \mathbb{Z}/N\mathbb{Z}^*$, compute $r := c^d \bmod N$ and output the key $k := H(r)$.

Theorem

If the RSA problem is hard relative to `genRSA` *and H is modeled as a random oracle, then above Construction is CCA-secure.*