CS6846 – Quantum Algorithms and Cryptography

RSA Encryption

Instructor: Shweta Agrawal, IIT Madras
Email: shweta@cse.iitm.ac.in

Recap: RSA Assumption

GenRSA

Input: Security parameter 1^n
Output: N, e, d as described in the text

$(N, p, q) \leftarrow \mathsf{GenModulus}(1^n)$
$\phi(N) := (p-1)(q-1)$
find e such that $\gcd(e, \phi(N)) = 1$
compute $d := [e^{-1} \bmod \phi(N)]$
return N, e, d

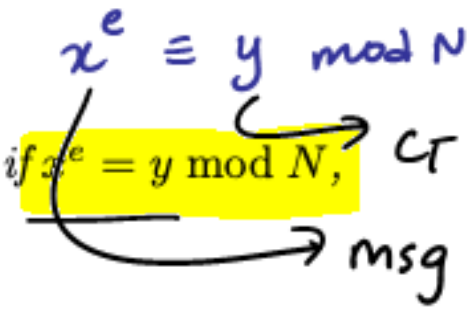
$N = p \cdot q$

Recap: RSA Assumption

Given $y \in \mathbb{Z}_N^*$, together with (N, e), hard to compute x s.t. $x^e \equiv y \bmod N$.

The RSA experiment $\mathsf{RSA\text{-}inv}_{\mathcal{A},\mathsf{GenRSA}}(n)$

1. *Run* $\mathsf{GenRSA}(1^n)$ *to obtain* (N, e, d).
2. *Choose* $y \leftarrow \mathbb{Z}_N^*$.
3. $\mathcal{A}$ *is given* N, e, y, *and outputs* $x \in \mathbb{Z}_N^*$.
4. *The output of the experiment is defined to be 1 if* $x^e = y \bmod N$, *and 0 otherwise.*



DEFINITION 7.46 *We say* the RSA problem is hard relative to GenRSA *if for all probabilistic, polynomial-time algorithms* $\mathcal{A}$ *there exists a negligible function* negl *such that*

$$\Pr[\mathsf{RSA\text{-}inv}_{\mathcal{A},\mathsf{GenRSA}}(n) = 1] \leq \mathsf{negl}(n).$$

RSA Assumption : Is GenRSA s.t. RSA problem is hard relative to it.

"TextBook" RSA

$\text{Gen}(1^\lambda)$: Run $\text{GenRSA}(1^\lambda) \to N, e, d$

Msg space $= \mathbb{Z}_N^*$

$\text{Enc}(PK = (N, e), m)$:

Idea: compute $m^e \bmod N \triangleq CT$

$\text{Dec}(CT, d)$: $CT^d \bmod N$

$= m^{e \cdot d} = m^1 \bmod N = m$

Recap:IND-CPA Security

The CPA indistinguishability experiment $\mathsf{PubK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$**:**

1. $\mathsf{Gen}(1^n)$ *is run to obtain keys* (pk, sk).
2. *Adversary* $\mathcal{A}$ *is given* pk *as well as oracle access to* $\mathsf{Enc}_{pk}(\cdot)$. *The adversary outputs a pair of messages* m_0, m_1 *with* $|m_0| = |m_1|$. *(These messages must be in the plaintext space associated with* pk*.)*
3. *A random bit* $b \in \{0,1\}$ *is chosen, and then the ciphertext* $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ *is computed and given to* $\mathcal{A}$. *We call* c *the challenge ciphertext.* $\mathcal{A}$ *continues to have access to* $\mathsf{Enc}_{pk}(\cdot)$.
4. $\mathcal{A}$ *outputs a bit* b'.
5. *The output of the experiment is defined to be 1 if* $b' = b$*, and 0 otherwise.*

DEFINITION 10.4 *Public-key encryption scheme* $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has* indistinguishable encryptions under chosen-plaintext attacks *(or is* CPA secure*) if for all probabilistic, polynomial-time adversaries* $\mathcal{A}$*, there exists a negligible function* negl *such that:*

$$\Pr[\mathsf{PubK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n).$$

What goes wrong?

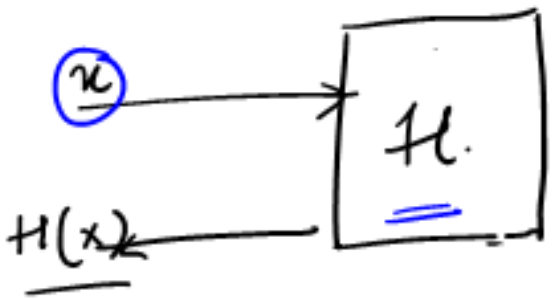
Enc is deterministic

IND-CPA cannot be satisfied.

$\mathcal{A}$ can itself encrypt m_0 and m_1,
& test which is equal to CT.

Randomizing Encryption

Random Oracle.



SHA-3
hash function

$$H : \{0,1\}^{l_1} \longrightarrow \{0,1\}^{l_2}.$$

Random Oracle Model

1. Oracle is a box that takes some binary string as input and returns binary string as output

Random Oracle Model

1. Oracle is a box that takes some binary string as input and returns binary string as output
2. Internal workings unknown and inscrutable

Random Oracle Model

1. Oracle is a box that takes some binary string as input and returns binary string as output
2. Internal workings unknown and inscrutable
3. Box is consistent: same input, same output

Random Oracle Model

1. Oracle is a box that takes some binary string as input and returns binary string as output
2. Internal workings unknown and inscrutable
3. Box is consistent: same input, same output
4. Anyone can interact (honest or adversary) by *querying* oracle

Random Oracle Model

1. Oracle is a box that takes some binary string as input and returns binary string as output
2. Internal workings unknown and inscrutable
3. Box is consistent: same input, same output
4. Anyone can interact (honest or adversary) by *querying* oracle
5. *Random* oracle mimics random function

Random Oracle Model

1. Oracle is a box that takes some binary string as input and returns binary string as output
2. Internal workings unknown and inscrutable
3. Box is consistent: same input, same output
4. Anyone can interact (honest or adversary) by *querying* oracle
5. *Random* oracle mimics random function
6. Hard to invert by definition

RSA Encryption in ROM

$H : \{0,1\}^{2n} \to \{0,1\}^{\ell}$ be hash fn

(modelled as a random oracle)

Key Generation (Gen) : As before

output N, e, d.

$\underbrace{N, e}_{\text{Public}}$ ↪ Private.

Enc (m, N, e): $m \in \{0,1\}^{\ell}$.

- Pick $r \leftarrow \mathbb{Z}_N^*$
- Compute $r^e \bmod N \triangleq c_1$
- Compute $\underbrace{H(r)}_{G} \oplus m \triangleq c_2$.

Dec:

- $c_1^d \bmod N$ to get r.
- $H(r) \oplus c_2$ to get m.

Is this Secure: Intuition