COMP8760 Lecture - 6

RSA: its Correctness and Security

Sanjay Bhattacherjee

University of Kent

Study Material

Book 1 *Cryptography Made Simple*
Author Nigel P. Smart.
Link to eBook

Section 1.1.2 Rings
Section 15.1 “Naive” RSA Encryption
Section 11.5.1 Basic notions of security
Section 11.5.2 Modern notions of security

Ring

A ring is a triplet $(R, +, \times)$ that satisfies the following properties.

- $(R, +)$ is a an abelian group, satisfying the following properties:
  - Closure
  - Associativity
  - Identity
  - Inverse
  - Commutative
- $(R, \times)$ satisfies the following properties:
  - Closure
  - Associativity
  - Identity

Ring: Examples

$(\mathbb{Z}_{10}, + \pmod{10}, \times \pmod{10})$

- $(\mathbb{Z}_{10}, + \pmod{10})$ is a an abelian group.
- $(\mathbb{Z}_{10}, \times \pmod{10})$ satisfies
  - Closure
  - Associativity
  - Identity

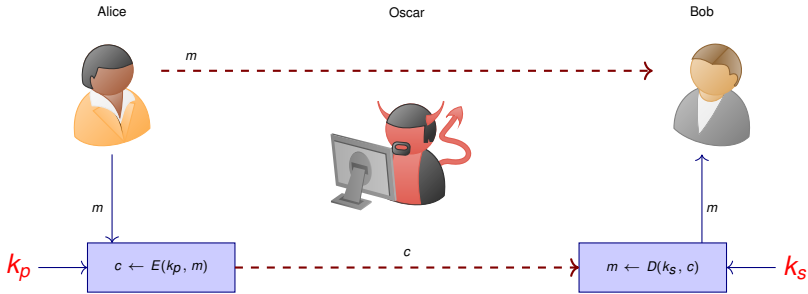
$(\mathbb{Z}_2, + \pmod{2}, \times \pmod{2})$

- $(\mathbb{Z}_2, + \pmod{2})$ is a an abelian group.
- $(\mathbb{Z}_2, \times \pmod{2})$ satisfies
  - Closure
  - Associativity
  - Identity

Rings used in Cryptography

$\mathbb{Z}_N$

Asymmetric Key Encryption System



Confidentiality with
two different keys (k_s and k_p) of Bob (recipient)!

“Naive” RSA Encryption System

Setup

Input: The security parameter ν (typically, $\nu = 2048$)

1. Generates two $\nu/2$-bit prime numbers p and q
2. Assigns $N \leftarrow p \cdot q$

$\mathbb{Z}_N$ and $\mathbb{Z}_N^*$

The Ring: $\mathbb{Z}_N$

1. $N = p \cdot q$ is composite
2. $\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}$ is a ring
   with respect to $+ \pmod N$ and $\cdot \pmod N$.

The Multiplicative Group: $\mathbb{Z}_N^*$

1. $\mathbb{Z}_N^*$ is the set of all non-zero $m \in \mathbb{Z}_N$, such that $\gcd(m, N) = 1$
2. It is a group with respect to $\cdot \pmod N$ with

$$|\mathbb{Z}_N^*| = \phi(N).$$

$$\phi(N) = N\left(1 - \tfrac{1}{p}\right)\left(1 - \tfrac{1}{q}\right) = \underbrace{(p-1)}_{\phi(p)}\underbrace{(q-1)}_{\phi(q)} = N - (p+q) + 1.$$

$\mathbb{Z}_N$ and $\mathbb{Z}_N^*$

The Multiplicative Group: $\mathbb{Z}_N^*$

1. $\mathbb{Z}_N^*$ has all non-zero $m \in \mathbb{Z}_N$, such that $\gcd(m, N) = 1$
2. It is a group with respect to $\cdot \pmod N$ with

$$|\mathbb{Z}_N^*| = \phi(N).$$

$$\phi(N) = N\left(1 - \tfrac{1}{p}\right)\left(1 - \tfrac{1}{q}\right) = \underbrace{(p-1)}_{\phi(p)}\underbrace{(q-1)}_{\phi(q)} = N - (p+q) + 1$$

   We also use $M = \phi(N)$.
3. Multiples of q in $\{0, N-1\}$: $|\{0 \cdot q, 1 \cdot q, 2 \cdot q, \ldots, (p-1) \cdot q\}| = p$.
   Multiples of p in $\{0, N-1\}$: $|\{0 \cdot p, 1 \cdot p, 2 \cdot p, \ldots, (q-1) \cdot p\}| = q$.
   0 is a multiple of both p and q.
   The total number of integers $\in \mathbb{Z}_N$ but $\notin \mathbb{Z}_N^*$ is $(p + q - 1)$.
4. For large values of p and q, the fraction

$$\frac{M}{N} = \frac{\phi(N)}{N} \approx 1.$$

   In other words, the set $\mathbb{Z}_N^*$ is almost as large a set as $\mathbb{Z}_N$.

"Naive" RSA Encryption System

Setup: ... continued

3. Find an integer e such that

$$\gcd(e, M) = 1.$$

In other words, $e \in \mathbb{Z}_M^*$.

The algorithm:

- Choose random integers $z \leq M$ until

$$\gcd(z, M) = 1.$$

Use the the extended Euclidean algorithm to find $\gcd(z, M)$.

- Once we have found such $z = e$, the output of the extended Euclidean algorithm will be in the form

$$1 = x \cdot e + y \cdot M.$$

- Applying the modular operation $(\mathrm{mod}\ M)$ on both sides of this equation, we get

$$x \cdot e = 1 \pmod{M}.$$

- This element x output by the algorithm is the inverse of e in the multiplicative group $\mathbb{Z}_M^*$.
We denote this inverse element by d.
So,

$$d \cdot e = 1 \pmod{M}.$$

Setup: Summary

Input: The security parameter ν (typically, $\nu = 2048$)

1. Generates two $\nu/2$-bit prime numbers p and q
2. Assigns $N \leftarrow p \cdot q$
   We denote $M = \phi(N)$.
3. Find an integer e such that
   $$\gcd(e, M) = 1.$$
4. Find an integer d such that
   $$d \cdot e = 1 \pmod{M}.$$
5. Finally, the output of the setup phase is:
   public key $\mathfrak{pk} \leftarrow (N, e)$
   secret key $\mathfrak{sk} \leftarrow (N, d)$

Note: p, q and $\phi(N)$ are not required after the setup is complete!

Encryption and Decryption

Encryption

The user's message is mapped to an element $m \in \mathbb{Z}_N$. The encryption algorithm takes as input the message m and the public key $\mathfrak{pk} = (N, e)$, and finds the ciphertext c as follows.

$$c \leftarrow m^e \pmod{N}.$$

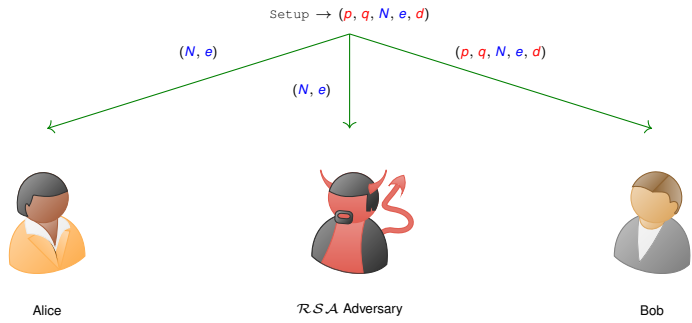
Decryption

The decryption algorithm takes as input the ciphertext c and the secret key $\mathfrak{sk} = (N, d)$, and finds the message m' as follows.

$$m' \leftarrow c^d \pmod{N}.$$

We say that the system works correctly if the message m that was encrypted is indeed the one that has been found as the output of decryption. In other words, the system works correctly if $m = m'$.

RSA Encryption



Extended Euclidean Algorithm (EEA)

Input: Integers a, b
Output: Integers g, x, y such that

$$g = \gcd(a, b) = x \cdot a + y \cdot b.$$

Input: a, b

Initialisation:

$r' \leftarrow a$, $r \leftarrow b$
$s' \leftarrow 1$, $s \leftarrow 0$
$t' \leftarrow 0$, $t \leftarrow 1$

Iteration:

```
while r ≠ 0 do:
    q ← ⌊r'/r⌋
    (r', r) ← (r, r' − q · r)
    (s', s) ← (s, s' − q · s)
    (t', t) ← (t, t' − q · t)
```

Final:

$d \leftarrow r', x \leftarrow s', y \leftarrow t'$
return g, x, y

EEA for Multiplicative Inverse d of $e \mod M$

$$\gcd(e, M) = d \cdot e + y \cdot M$$

We know that the multiplicative inverse of $e \pmod M$ exists if and only if

$$\gcd(e, M) = 1$$

Hence, we can find e (and hence the corresponding d) such that

$$1 = d \cdot e \pmod M$$

Example

- Choose $p = 7$, $q = 11$, and compute $N = p \cdot q = 77$
- Find $M = (p - 1)(q - 1) = 60$
- Choose e, such that $\gcd(e, M) = 1$
  Let $e = 13$, then using EEA we get:

$$\gcd(13, 60) = -23 \times 13 + 5 \times 60 = 1.$$

So, $d = -23 \pmod{60} = 60 - 23 \pmod{60} = 37$.

Correctness

To show that the scheme is correct, we have to show that

$$\begin{aligned} m = m' = c^d \pmod N \\ = (m^e)^d \pmod N \\ = (m)^{e \cdot d} \pmod N \end{aligned} \tag{2}$$

From (1) we have

$$e \cdot d = 1 \pmod M.$$

So, we may write $e \cdot d = 1 + s \cdot M$ for some integer s.
Substituting this value of $e \cdot d$ in (2), the new statement to be proved is

$$\begin{aligned} & m = m^{1+s \cdot M} \pmod N \\ \implies & m = m \cdot m^{s \cdot \phi(N)} \pmod N \\ \implies & 0 = m \cdot \left(m^{s \cdot \phi(N)} - 1 \right) \pmod N. \end{aligned} \tag{3}$$

To prove (3), we will have to show that

$m \cdot \left(m^{s \cdot \phi(N)} - 1 \right)$ *is a multiple of* N.

Correctness

To prove correctness, we will have to show that

$$m \cdot \left(m^{s \cdot \phi(N)} - 1 \right) \textit{ is a multiple of } N.$$

We recollect at this point that $m \in \mathbb{Z}_N$. So,

- either $m \in \mathbb{Z}_N^*$,
- or $m \notin \mathbb{Z}_N^*$.

There are four possible scenarios as listed below.

	divisibility by p		divisibility by q
Case 1:	$\gcd(m, p) = 1$	and	$\gcd(m, q) = 1$
Case 2:	$m = 0 \pmod{p}$	and	$\gcd(m, q) = 1$
Case 3:	$\gcd(m, p) = 1$,	and	$m = 0 \pmod{q}$
Case 4:	$m = 0 \pmod{p}$,	and	$m = 0 \pmod{q}$

Correctness

To prove correctness, we will have to show that

$m \cdot \left(m^{s \cdot \phi(N)} - 1\right)$ *is a multiple of* N.

Case 1: $\gcd(m, p) = 1$ and $\gcd(m, q) = 1$

This is the most common case for

$$m \in \mathbb{Z}_N^*.$$

By Euler's theorem, we have

$$\begin{aligned} m^{\phi(N)} &= 1 \pmod{N} \\ \implies \left(m^{\phi(N)}\right)^s &= 1 \pmod{N} \\ \implies m^{s \cdot \phi(N)} &= 1 \pmod{N}. \end{aligned} \tag{4}$$

Hence, N divides $m \cdot \left(m^{s \cdot \phi(N)} - 1\right)$ and hence (3) is proved to be correct.

Correctness

To prove correctness, we will have to show that

$m \cdot \left(m^{s \cdot \phi(N)} - 1\right)$ *is a multiple of* N.

Case 2: $m = 0 \pmod{p}$ and $\gcd(m, q) = 1$

Since $\gcd(m, q) = 1$, we use Fermat's little theorem to note that

$$\begin{aligned} m^{(q-1)} &= 1 \pmod{q} \\ \implies \left(m^{(q-1)}\right)^{(p-1)} &= 1 \pmod{q} \\ \implies m^{\phi(N)} &= 1 \pmod{q} \\ \implies \left(m^{\phi(N)}\right)^{s} &= 1 \pmod{q} \\ \implies m^{s \cdot \phi(N)} &= 1 \pmod{q}. \end{aligned} \tag{5}$$

From (5), we get that q divides $\left(m^{s \cdot \phi(N)} - 1\right)$.
We already know that in this case, p divides m.
Combining, we get that N divides $m \cdot \left(m^{s \cdot \phi(N)} - 1\right)$.

Correctness

To prove correctness, we will have to show that

$$m \cdot \left(m^{s \cdot \phi(N)} - 1 \right) \textit{ is a multiple of } N.$$

Case 3: $\gcd(m, p) = 1$ and $m = 0 \pmod{q}$

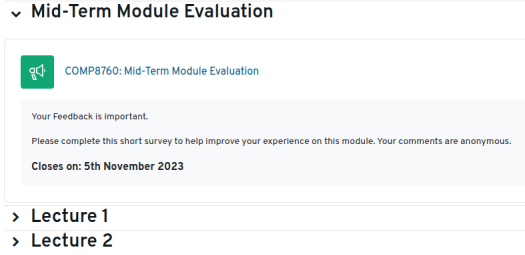
The arguments in this case are the same as Case 1, by interchanging the prime q with p.

Case 4: $m = 0 \pmod{p}$ and $m = 0 \pmod{q}$

For $m = 0$, (3) is trivially correct.
However, note that $c = m^e \pmod{N} = 0$ which is unchanged from m.

Mid-Term Evaluations - Moodle Survey

Please provide your feedback!



Resources on Moodle

Handout Note on “Naive” RSA - on Moodle, with Lecture 6

Naive RSA Encryption System

with proof of correctness

Sanjay Bhattacherjee

Setup. The setup algorithm takes as input the security parameter ν and generates two $\nu/2$-bit prime numbers p and q. Let $N = p \cdot q$. Since N is composite, $\mathbb{Z}/N\mathbb{Z} = \{0, 1, 2, \ldots, N-1\}$ is a ring with respect to addition and multiplication modulo N. We also have $(\mathbb{Z}/N\mathbb{Z})^*$ as the set of all non-zero elements of $\mathbb{Z}/N\mathbb{Z}$ that are mutually prime to N. It forms a group with respect to multiplication modulo N. The number of elements in this group is given by

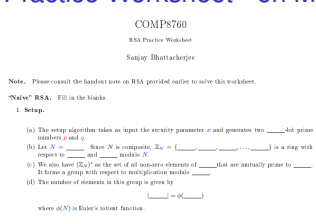
$$|(\mathbb{Z}/N\mathbb{Z})^*| = \phi(N)$$

where $\phi(N)$ is Euler's totient function. In particular, we have $\phi(N) = \phi(p)\phi(q) = (p-1)(q-1) = N-(p+q)+1$. Note that there are p multiples of q between 0 and $N-1$ namely those in the set

$$\{0 \cdot q, 1 \cdot q, \ldots, (p-1) \cdot q\}.$$

Similarly, there are q multiples of p between 0 and $N-1$ namely those in the set

Practice Worksheet - on Moodle, with Lecture 8



COMPS8760

RSA Practice Worksheet

Sanjay Bhattacherjee

Note. Please consult the handout note on RSA provided earlier to solve this worksheet.

“Naive” RSA. Fill in the blanks.

1. **Setup.**

   (a) The setup algorithm takes as input the security parameter ν and generates two _____-bit prime numbers p and q.
   (b) Let $N =$ _____. Since N is composite, $\mathbb{Z}_N = \{$_____, _____, _____, $\cdots$, _____$\}$ is a ring with respect to _____ and _____ modulo N.
   (c) We also have $(\mathbb{Z}_N)^*$ as the set of all non-zero elements of _____ that are mutually prime to _____. It forms a group with respect to multiplication modulo _____.
   (d) The number of elements in this group is given by

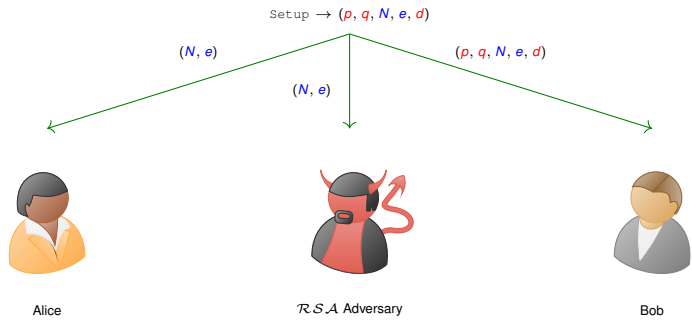
   $$|$_____$| = \phi($_____$)$$

   where $\phi(N)$ is Euler's totient function.

wdd25s

University of Kent

RSA Encryption



Security of "Naive" RSA

How can the adversary decrypt messages?

1. By knowing d (the decryption exponent)
   the secret part of $\mathfrak{sk} = (N, d)$, and compute:

$$m' = c^d \pmod{N}.$$

2. By knowing $M = \phi(N)$ and using the extended Euclidean algorithm to compute

$$\gcd(e, M).$$

   The output of the algorithm will have d such that

$$e \cdot d = 1 \pmod{M}.$$

   Then, as above in 1.
3. By knowing the primes p, q to compute

$$M = \phi(N) = \phi(p)\phi(q) = (p-1)(q-1).$$

   Then, as above in 2.

This amounts to knowing the factorisation $N = p \cdot q$.

Choice of m

1. In practice, for any choice of $N = p \cdot q$, there will be $(p + q - 1)$ elements $m \in \mathbb{Z}_N$ for which $\gcd(m, N) > 1$.

$$\text{Multiples of } p: \quad \{0 \cdot p, 1 \cdot p, \ldots, (q-1) \cdot p\}$$

$$\text{Multiples of } q: \quad \{0 \cdot q, 1 \cdot q, \ldots, (p-1) \cdot q\}$$

   These are in $m \notin \mathbb{Z}_N^*$ and hence $\gcd(m, N) \neq 1$.
2. If an attacker (only knowing $\mathfrak{pk} = (N, e)$) finds an $m \notin \mathbb{Z}_N^*$, then they can compute

$$\gcd(m, N)$$

   This is one of the two primes p or q.
3. They would then know factorisation of N and as a result,

   the cryptosystem will be broken!

Choice of m

4. If all messages $m \in \mathbb{Z}_N$ are equally likely to occur, the probability that the attacker finds an m for which $\gcd(m, N) > 1$ is
$$\frac{p + q - 1}{N}.$$

5. p and q are $\nu/2$-bit (around 1024-bit) primes.
So, $p + q - 1$ would be of around that size as well.
6. While N will be approximately of size ν (around 2048) bits.
7. So, the probability of finding such an element m is around
$$\frac{1}{2^{\nu/2}}.$$

Summary:

- The cases 2 and 3 (of the correctness proof) where a non-zero $m \notin \mathbb{Z}_N^*$ breaks the system, are extremely unlikely to occur.
- However, if they occur, the encryption and decryption operation will still be correct.

Homomorphic Encryption

Multiplicative Homomorphism

An encryption scheme has the multiplicative homomorphic property if given

$$\mathtt{Enc}_{\mathtt{pt}}(m_1) \text{ and } \mathtt{Enc}_{\mathtt{pt}}(m_2),$$

we can efficiently compute $\mathtt{Enc}_{\mathtt{pt}}(m_1 \cdot m_2)$, without knowing m_1 or m_2.

Application:

The multiplication computation can be delegated to untrusted devices without revealing the data.

Homomorphism of “Naive” RSA

$$\underbrace{\mathtt{Enc}_{\mathtt{pk}}(m_0)}_{\in \mathbb{Z}_N} \cdot \underbrace{\mathtt{Enc}_{\mathtt{pk}}(m_1)}_{\in \mathbb{Z}_N} \pmod N$$

$$= \underbrace{(m_0^e \pmod N)}_{\in \mathbb{Z}_N} \cdot \underbrace{(m_1^e \pmod N)}_{\in \mathbb{Z}_N} \pmod N$$

$$= \underbrace{(m_0^e \cdot m_1^e) \pmod N}_{\in \mathbb{Z}_N}$$

$$= (m_0 \cdot m_1)^e \pmod N$$

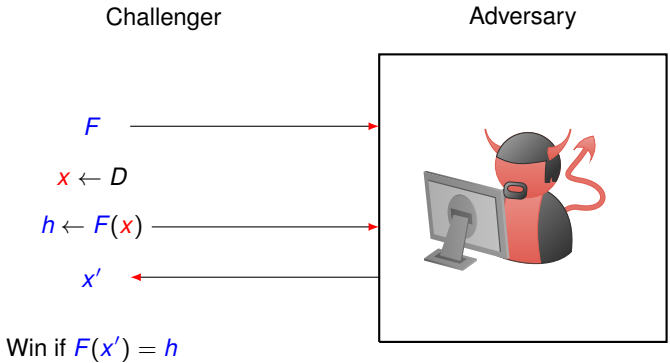
$$= \mathtt{Enc}_{\mathtt{pk}}(m_0 \cdot m_1).$$

One-Way Function (OWF)

Let $F : D \to C$. Given $h \leftarrow F(x)$, find a pre-image x' such that

$$h = F(x) = F(x').$$

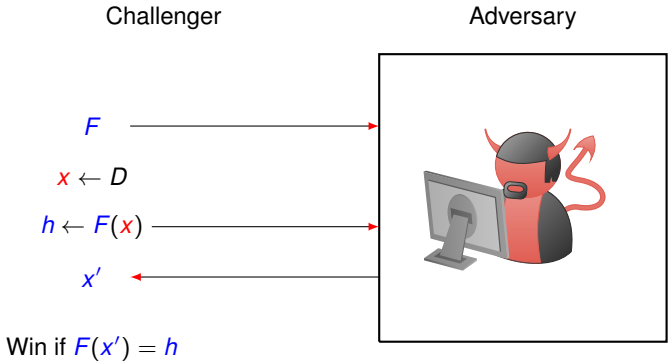
OWF Game



Advantage of Adversary = Pr[Adversary wins the game]

Computationally Unbounded Adversary

OWF Game



Advantage of Adversary = Pr[Adversary wins the game]

- A computationally unbounded adversary can run exhaustive search.
  It will check for every $x \in D$ until it finds an x' for which $h = F(x')$.
- Such an adversary will always win the above OWF game with probability 1.

Computationally Bounded Adversary

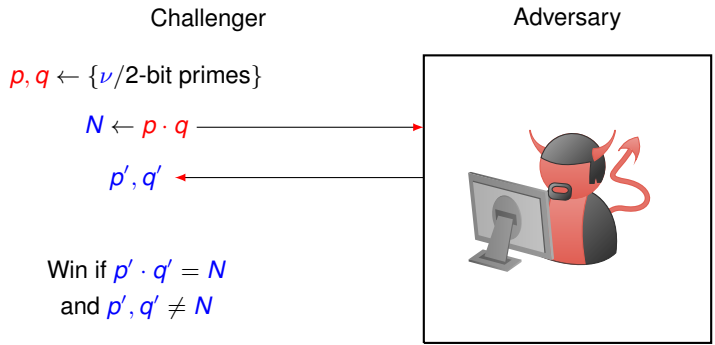
Adversary

So, we assume that our adversary is computationally bounded.
In particular, it does not have the resources to run an exhaustive search.

- With this restriction in place, we will assume the adversary has several other powers like encryption and decryption capabilities.
- We would want our adversary to have negligible probability of winning the security games.

Factoring: The Hardness Assumption

Given $N \leftarrow p \cdot q$, find p and q.

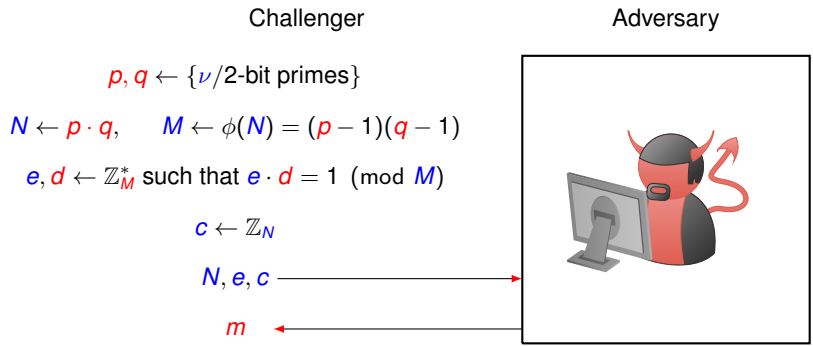


Advantage of Adversary = Pr[Adversary wins the game]

RSA: The Hardness Assumption

Given N, e and c, find m such that

$$m^e = c \pmod{N}.$$



Win if $m^e = c \pmod{N}$

Advantage of Adversary = Pr[Adversary wins the game]

An Abstract Cryptographic System (Encryption)

Consider an encryption system defined by the following 3 algorithms:

- Key generation algorithm: $(\mathfrak{pk}, \mathfrak{sk}) \leftarrow \texttt{KeyGen}()$
  This is typically the $\texttt{Setup}()$ algorithm or a part of it.

- Encryption algorithm: $c \leftarrow \texttt{Enc}_{\mathfrak{pk}}(m)$
  Here, $\texttt{Enc}_{\mathfrak{pk}}(m)$ is a short form of $\texttt{Enc}(m, \mathfrak{pk})$

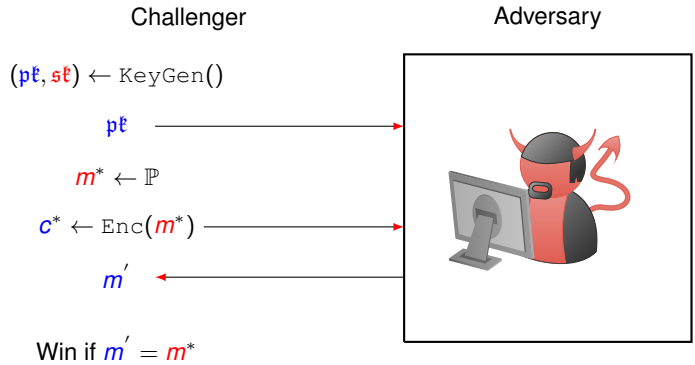
- Decryption algorithm: $m \leftarrow \texttt{Dec}_{\mathfrak{sk}}(c)$
  Here, $\texttt{Dec}_{\mathfrak{sk}}(c)$ is a short form of $\texttt{Dec}(c, \mathfrak{sk})$

Note:

- $\mathbb{P}$ is the set of all plaintexts - also called the Plaintext space
  So $m \in \mathbb{P}$
- $\mathbb{C}$ is the set of all ciphertexts - also called the Ciphertext space
  So $c \in \mathbb{C}$

One-Way Passive Attack (OW-PASS)

Also called One-Way Chosen Plaintext Attack (OW-CPA)



Advantage of Adversary = Pr[Adversary wins the game]

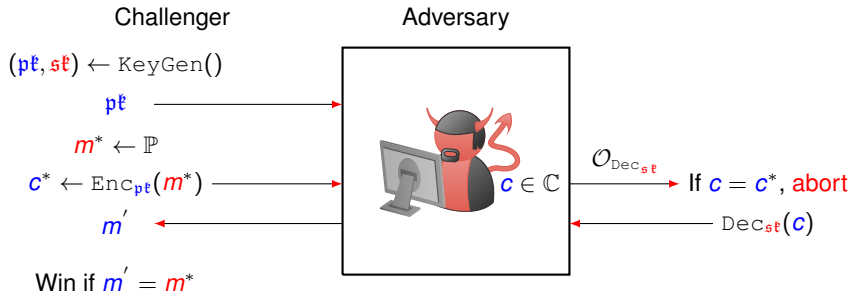
One-Way Passive Attack (OW-PASS)

- No oracles are provided to the adversary, and hence
  the adversary has very limited powers.
- The adversary has the public key $\mathfrak{pk}$
  to encrypt a limited number of messages.

  Hence, the security model is also called
  Chosen Plaintext Attack.
- Hence, $\mathtt{OW} - \mathtt{PASS}$ is a very weak form of security

One-Way Chosen Ciphertext Attack (OW-CCA)

The adversary has access to a decryption oracle.



Advantage of Adversary = Pr[Adversary wins the game]

Modern Notions of Security

- The simple notion of "breaking" an encryption algorithm (recovering the whole message as in $\mathtt{OW-CPA}$ and $\mathtt{OW-CCA}$) is expecting too much work from the adversary.
  What if the adversary can break a part of a message?
- The adversary should not be allowed to obtain any information about the plaintext.

Essentially three notions of security.

1. Perfect security
2. Semantic security
3. $\mathtt{IND}$ security / $\mathtt{IND}$istinguishability of encryptions / polynomial security

Perfect Security

- also called information-theoretic security

An adversary with infinite computing power
can learn nothing about the plaintext given the ciphertext.

- Achieved only if,
  1. the key is as long as the message, and
  2. the same key is never used twice.

- This is the strongest notion of security.
  It is secure against an all-powerful adversary.
- Only achievable in symmetric key schemes.

Semantic Security

- is like perfect security but,

the adversary is assumed to have
only polynomially bounded computational power.

- Formally,
  1. for all probability distributions on the message space,
     whatever an adversary can compute (in polynomial time) about the plaintext given the ciphertext,
     they should also be able to compute without the ciphertext.
  2. having the ciphertext does not help in finding out anything about the message.

- For a single bit message,

$$\mathit{Advantage}(A) = 2 \cdot \left| \underbrace{\Pr[A(c) = \mathtt{Dec}_{sk}(c)]}_{\text{Probability that the Adversary guesses the bit correctly}} - \underbrace{\frac{1}{2}}_{\text{Probability that the Adversary guesses the bit randomly}} \right| .$$

IND Security

Semantic Security is difficult to prove for practical schemes.

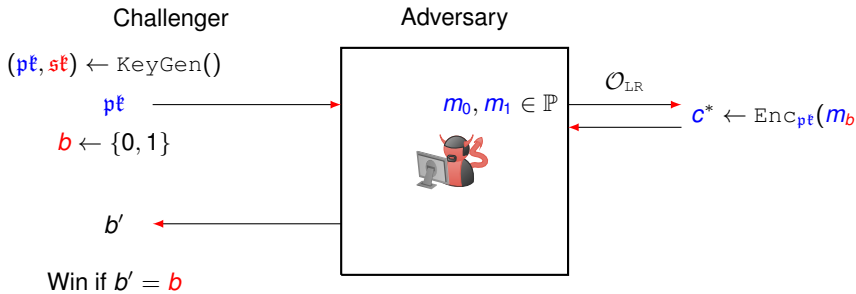
If a system has IND security then it also has semantic security.

The IND Security Game

1. **Find stage:** The adversary produces two plaintext messages m_0 and m_1 of equal length.
2. **Challenge stage:** The challenger chooses a random bit b and encrypts
$$c^* \leftarrow \mathrm{Enc}(m_b).$$
3. **Guess stage:** Given c, the adversary has to guess b with probability (significantly) greater than $\frac{1}{2}$.

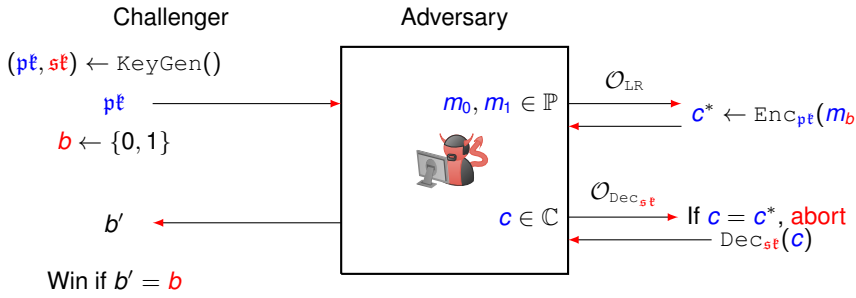
A system is said to have INDistinguishability of encryptions if no adversary can win the following game with probability (significantly) greater than $\frac{1}{2}$.

IND − CPA Security



Advantage of Adversary = $2 \cdot \left| \Pr[\text{Adversary wins the game}] - \frac{1}{2} \right|$.

IND − CCA Security



Advantage of Adversary = $2 \cdot \left| \Pr[\text{Adversary wins the game}] - \frac{1}{2} \right|$.

IND − CPA adversary does not have access to the decryption oracle $\mathcal{O}_{\mathtt{Dec}_{\mathfrak{sk}}}$.

Theorem 15.2

Naive RSA encryption is not IND − CPA secure.

Proof idea: The adversary knows the encryption of any known message.

- The adversary chooses m_0 and m_1
- It has the public key $\mathfrak{pk} = (N, e)$
- It computes

$$c_0 \leftarrow {m_0}^e \pmod{N}.$$

- Upon receiving the challenge ciphertext c^*, check if

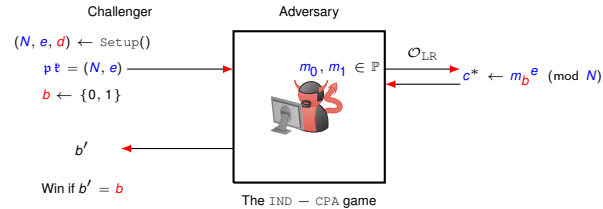
$$c_0 = c^*.$$

  - If yes, the message m_b encrypted by the challenger was m_0
  - If not, the message m_b encrypted by the challenger was m_1

Theorem 15.2

Naive RSA encryption is not IND − CPA secure.

Proof idea: The adversary knows the encryption of any known message.



- Compute $c_0 \leftarrow m_0{}^e \pmod N$.
- If $c^* = c_0$, then $m_b = m_0$; else $m_b = m_1$.

Theorem 15.2

Naive RSA encryption is not IND − CCA secure.

Proof idea: The encryption is malleable.

- For the challenge ciphertext c^*, the adversary uses the homomorphic property to compute

$$c' = \mathrm{Enc}_{\mathrm{pk}}(2) \cdot c^* \pmod{N}$$

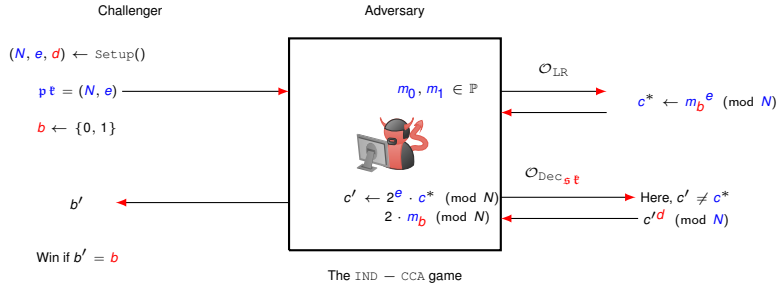
- Uses $\mathcal{O}_{\mathrm{Dec}_{\mathrm{sk}}}$ to decrypt c' to get $2 \cdot m_b \pmod{N}$
- Find $2^{-1} \cdot (2 \cdot m_b) \pmod{N} = m_b$
- Any $a \in \mathbb{Z}_N$ works in place of 2

IND − CCA Security of "Naive" RSA

Theorem 15.2

Naive RSA encryption is not IND − CCA secure.

Proof idea: The encryption is malleable.



- Compute "related ciphertext" $c' \leftarrow 2^e \cdot c^* \pmod N$ and query $\mathcal{O}_{\text{Dec}_{\mathfrak{sk}}}(c')$
- Response: $c'^d = (2^e \cdot c^*)^d = (2^e \cdot m_b{}^e)^d = (2 \cdot m_b)^{e \cdot d} = 2 \cdot m_b \pmod N$