CSL759: Cryptography and Computer Security

Ragesh Jaiswal

CSE, IIT Delhi

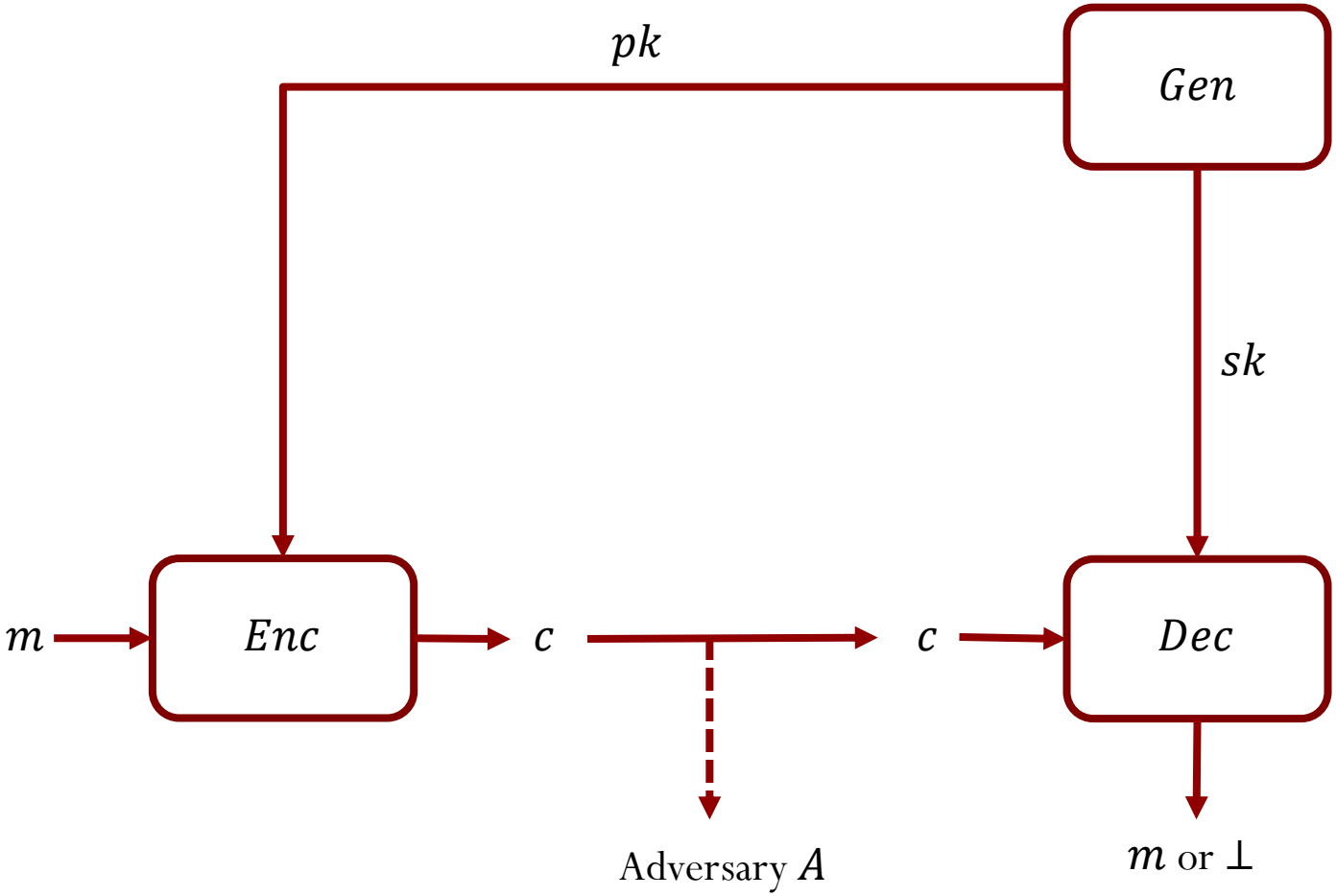
Public-key Encryption

Public-key Encryption

- Definition: A public-key encryption scheme $PKE = (Gen, Enc, Dec)$ consists of three algorithms such that:
  1. The key generation algorithm Gen takes as input 1^n (n is the security parameter) and outputs a pair of keys (pk, sk). pk is known as the public key and sk is known as the secret key.
  2. The encryption algorithm Enc takes as input the public key pk and a message (from appropriate space) and outputs the ciphertext $c \leftarrow Enc_{pk}(m)$.
  3. The decryption algorithm takes as input the private key sk and a ciphertext and outputs a message m or a special symbol $\perp$ (denoting failure). This is denoted by $m \leftarrow Dec_{sk}(c)$.

  We have $\Pr\left[Dec_{sk}\left(Enc_{pk}(m)\right) = m\right] \approx 1$.

Public-key Encryption



Public-key Encryption

- What are the advantages of Public-key encryption over private-key encryption?
  - Key distribution is *simpler*.
  - *Open systems*: The identity of the person is not required to be known before secure communication.

- What are the disadvantages of PKE?
  - PKE schemes tend to be *slow* because it involves more complex arithmetic operations (e.g. computing exponentiations, inverses modulo N etc.).

- We will first see two PKE schemes before discussing security for PKE.

RSA and El-Gamal

Textbook RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod \ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod \ N)]$.

- GenRSA(1^n)
  - Run GenModulus(1^n) to obtain (N, p, q).
  - Let $\phi(N) = (p-1) \cdot (q-1)$.
  - Find e such that $\gcd(e, \phi(N)) = 1$.
  - Compute $d = [e^{-1} \ (mod \ \phi(N))]$.
  - Return (N, e, d)

- GenModulus(1^n)
  - Run GRP(1^n) to obtain p, q.
  - Let $N = p \cdot q$.
  - Return (N, p, q).

- GRP(1^n)
  - For $i = 1$ to t
    - Randomly pick $p' \in \{0,1\}^{n-1}$
    - $p \leftarrow 1|p'$
    - If (p is prime) then output p
  - Output "fail"

Textbook RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- What are the issues with Textbook RSA as a PKE?

Textbook RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- What are the issues with Textbook RSA as a PKE?
  1. Message m should be a member of Z_N^*.

Textbook RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- What are the issues with Textbook RSA as a PKE?
  1. Message m should be a member of Z_N^*.
  2. Can this PKE scheme be IND-CPA secure?

Textbook RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- What are the issues with Textbook RSA as a PKE?
  1. Message m should be a member of Z_N^*.
  2. Can this PKE scheme be IND-CPA secure?
     - No, since this is a deterministic encryption scheme.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 1: How do we encode a typical message m which is a k-bit string?
  - If $k \geq N$, then we will have to break the message into bit chunks and then do appropriate padding. This will allow us to get $m_1', m_2', \ldots \in Z_N$ which we can individually encrypt using textbook RSA.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 1: How do we encode a typical message m which is a k-bit string?
  - If $k \geq N$, then we will have to break the message into bit chunks and then do appropriate padding. This will allow us to get $m_1', m_2', \ldots \in Z_N$ which we can individually encrypt using textbook RSA.
- Problem 2: What do we do if $m \notin Z_N^*$ (i.e., $\gcd(m, N) \neq 1$)?

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 1: How do we encode a typical message m which is a k-bit string?
  - If $k \geq N$, then we will have to break the message into bit chunks and then do appropriate padding. This will allow us to get $m_1', m_2', \ldots \in Z_N$ which we can individually encrypt using textbook RSA.
- Problem 2: What do we do if $m \notin Z_N^*$ (i.e., $\gcd(m, N) \neq 1$)?
  - Let $m \notin Z_N^*$, what is the value of $[(m^e)^d \ (mod\ N)]$?

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 1: How do we encode a typical message m which is a k-bit string?
  - If $k \geq N$, then we will have to break the message into bit chunks and then do appropriate padding. This will allow us to get $m_1', m_2', \ldots \in Z_N$ which we can individually encrypt using textbook RSA.
- Problem 2: What do we do if $m \notin Z_N^*$ (i.e., $\gcd(m, N) \neq 1$)?
  - Let $m \notin Z_N^*$, what is the value of $[(m^e)^d \ (mod\ N)]$?
    - Answer: m

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 1: How do we encode a typical message m which is a k-bit string?
- Problem 2: What do we do if $m \notin Z_N^*$ (i.e., $\gcd(m, N) \neq 1$)?
- Claim 3: Given that the messages are random strings, the probability that the encrypted message $m \notin Z_N^*$ is very small.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 1: How do we encode a typical message m which is a k-bit string?
- Problem 2: What do we do if $m \notin Z_N^*$ (i.e., $\gcd(m, N) \neq 1$)?
- Claim 3: Given that the messages are random strings, the probability that the encrypted message $m \notin Z_N^*$ is very small.
- Claim 4: It is computationally hard to find a message m such that $m \notin Z_N^*$.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod\ N)]$?

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod\ N)]$?
  - Answer: $2n \cdot (3 \cdot (2n)^2) = 24n^3$.
  - Suppose we use a small value of e (say $e = 3$). Then encryption is faster. $O(n^2)$ instead of $O(n^3)$.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod\ N)]$?
  - Answer: $2n \cdot (3 \cdot (2n)^2) = 24n^3$.
  - Suppose we use a small value of e (say $e = 3$). Then encryption is faster. $O(n^2)$ instead of $O(n^3)$.
- Suppose we want decryption to be fast. Can we pick a small d such that $e \cdot d \equiv 1 \ (mod\ \phi(N))$?

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod\ N)]$?
  - Answer: $2n \cdot (3 \cdot (2n)^2) = 24n^3$.
  - Suppose we use a small value of e (say $e = 3$). Then encryption is faster. $O(n^2)$ instead of $O(n^3)$.
- Suppose we want decryption to be fast. Can we pick a small d such that $e \cdot d \equiv 1 \ (mod\ \phi(N))$?
  - No. Since then an adversary can try all possible values of d to decrypt.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod \ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod \ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod \ N)]$?
- Suppose we want decryption to be fast. Can we pick a small d such that $e \cdot d \equiv 1 \ (mod \ \phi(N))$?
- Is there a way to make decryption faster?

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod\ N)]$?
- Suppose we want decryption to be fast. Can we pick a small d such that $e \cdot d \equiv 1 \ (mod\ \phi(N))$?
- Is there a way to make decryption faster?
  - CRT-RSA: Instead of computing $[c^d \ (mod\ p)]$ and $[c^d \ (mod\ q)]$ and then use CRT to compute $[c^d \ (mod\ N)]$.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod \ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod \ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod \ N)]$?
- Suppose we want decryption to be fast. Can we pick a small d such that $e \cdot d \equiv 1 \ (mod \ \phi(N))$?
- Is there a way to make decryption faster?
  - CRT-RSA: Instead of computing $[c^d \ (mod \ p)]$ and $[c^d \ (mod \ q)]$ and then use CRT to compute $[c^d \ (mod \ N)]$.
  - Note that $[c^d \ (mod \ p)] = [c^{dp} \ (mod \ p)]$ and $[c^{dq} \ (mod \ q)] = [c^{dq} \ (mod \ q)]$, where
    - $dp = [d \ (mod \ (p-1))]$ and
    - $dq = [d \ (mod \ (q-1))]$

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod \ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod \ N)]$.

- Problem 5: What is the running time for encrypting a message $m \in Z_N$? That is, computing $[m^e \ (mod \ N)]$?
- Suppose we want decryption to be fast. Can we pick a small d such that $e \cdot d \equiv 1 \ (mod \ \phi(N))$?
- Is there a way to make decryption faster?
  - CRT-RSA: Instead of computing $[c^d \ (mod \ p)]$ and $[c^d \ (mod \ q)]$ and then use CRT to compute $[c^d \ (mod \ N)]$.
  - Note that $[c^d \ (mod \ p)] = [c^{dp} \ (mod \ p)]$ and $[c^{dq} \ (mod \ q)] = [c^{dq} \ (mod \ q)]$, where
    - $dp = [d \ (mod \ (p-1))]$ and
    - $dq = [d \ (mod \ (q-1))]$
  - This reduces the running time by a constant factor.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Attacks on Textbook-RSA:
  - Encrypting short messages using small e:
    - Suppose $e = 3$ and $m < N^{1/3}$, is it possible to find m from $c = [m^3 \ (mod\ N)]$?

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Attacks on Textbook-RSA:
  - Encrypting short messages using small e:
    - Suppose $e = 3$ and $m < N^{1/3}$, is it possible to find m from $c = [m^3 \ (mod\ N)]$?
      - Yes. Simply compute $c^{1/3}$.
  - Encrypting same message under different keys using small e:
    - Suppose m is encrypted using the following public keys $< N_1, 3 >, < N_2, 3 >, < N_3, 3 >$, is it possible to find m?

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Attacks on Textbook-RSA:
  - Encrypting short messages using small e:
    - Suppose $e = 3$ and $m < N^{1/3}$, is it possible to find m from $c = [m^3 \ (mod\ N)]$?
      - Yes. Simply compute $c^{1/3}$.
  - Encrypting same message under different keys using small e:
    - Suppose m is encrypted using the following public keys $< N_1, 3 >, < N_2, 3 >, < N_3, 3 >$, is it possible to find m?
      - Yes. Compute $([m^3 \ mod\ N_1], [m^3 \ mod\ N_2], [m^3 \ mod\ N_3]) \rightarrow c = [m^3 \ mod\ N_1 \cdot N_2 \cdot N_3]$.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Attacks on Textbook-RSA:
  - Encrypting short messages using small e:
  - Encrypting same message under different keys using small e:
  - Using same RSA modulus N for creating different key pairs $(< N, e_1 >, < N, d_1 >), (< N, e_2 >, < N, d_2 >), \ldots$ for different people.

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Attacks on Textbook-RSA:
  - Encrypting short messages using small e:
  - Encrypting same message under different keys using small e:
  - Using same RSA modulus N for creating different key pairs $(< N, e_1 >, < N, d_1 >), (< N, e_2 >, < N, d_2 >), \ldots$ for different people.
    1. Fact: N can be factored using (e_i, d_i).

Textbook RSA: Nuts and Bolts

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in Z_N^*$, compute the ciphertext $c = [m^e \ (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d \ (mod\ N)]$.

- Attacks on Textbook-RSA:
  - Encrypting short messages using small e:
  - Encrypting same message under different keys using small e:
  - Using same RSA modulus N for creating different key pairs $(< N, e_1 >, < N, d_1 >), (< N, e_2 >, < N, d_2 >), \ldots$ for different people.
    1. Fact: N can be factored using (e_i, d_i).
    2. If $\gcd(e_1, e_2) = 1$, then $X \cdot e_1 + Y \cdot e_2 = 1$. Let $c_1 = [m^{e_1} \ (mod\ N)]$ and $c_2 = [m^{e_2} (mod\ N)]$. Then $m = [c_1^X \cdot c_2^Y \ (mod\ N)]$.

Padded RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in \{0,1\}^{l(n)}$, choose a random string $r \leftarrow \{0,1\}^{|N|-l(n)-1}$ and compute the ciphertext $c = [(r||m)^e (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N^*$, compute the message $m = [c^d\ (mod\ N)]$. Output the $l(n)$ low order bits of m.

Padded RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in \{0,1\}^{l(n)}$, choose a random string $r \leftarrow \{0,1\}^{|N|-l(n)-1}$ and compute the ciphertext $c = [(r||m)^e (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N$, compute the message $m = [c^d\ (mod\ N)]$. Output the $l(n)$ low order bits of m.

- Can the above scheme be IND-CPA secure if $l(n)$ is large?

Padded RSA

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a message $m \in \{0,1\}^{l(n)}$, choose a random string $r \leftarrow \{0,1\}^{|N|-l(n)-1}$ and compute the ciphertext $c = [(r||m)^e (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N$, compute the message $m = [c^d\ (mod\ N)]$. Output the $l(n)$ low order bits of m.

- Can the above scheme be IND-CPA secure if $l(n)$ is large?
  - No. The adversary can try out all possibilities for the random string.
- What if $l(n)$ is small compared to $2n$?
  - Theorem: If the RSA problem is hard relative to GenRSA, then the above scheme with $l(n) = O(\log n)$ has indistinguishable encryptions under chosen-plaintext attack.

Case study:PKCS#1 v1.5

- Gen: On input 1^n run $GenRSA(1^n)$ to obtain N, e, d. Let N be k bytes long.
  - $pk =< N, e >$
  - $sk =< N, d >$
- Enc: On input a public key $pk =< N, e >$ and a D byte long message m such that $D \leq k - 11$, choose a random string $r \leftarrow \{0,1\}^{8(k-D-3)}$ and compute the ciphertext $c = [(00000000||00000010||r||00000000||m)^e (mod\ N)]$.
- Dec: On input a private key $sk =< N, d >$ and a ciphertext $c \in Z_N$, compute the message $m = [c^d\ (mod\ N)]$. Output the appropriate message.

- There are chosen ciphertext attacks on the above scheme. This attacks decrypts a target ciphertext by using just 0/1 information regarding whether a few related ciphertexts decrypt correctly or not!

End