RSA Encryption

Slides by Prof. Jonathan Katz.
Lightly edited by me.

Recall…

- Let p, q be random, equal-length primes
- Compute modulus N=pq
- Choose e, d such that $e \cdot d = 1 \bmod \phi(N)$

- The e^{th} root of x modulo N is $[x^d \bmod N]$
  - I.e., easy to compute given p, q (or d)
- *RSA assumption*: given N, e only, it is hard to compute the e^{th} root of a uniform $c \in \mathbb{Z}_N^*$

- This suggests a public-key encryption scheme!

“Plain” RSA encryption

N, e

c

$(N, e, d) \leftarrow \text{RSAGen}(1^n)$

$pk = (N, e)$

$sk = d$

$m = [c^d \bmod N]$

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

Is this scheme secure?

- This scheme is *deterministic*
  - Cannot be CPA-secure!
- RSA assumption only refers to hardness of computing the e^{th} root of a *uniform* c
  - c is not uniform unless m is
  - W
  - Easy to compute e^{th} root of $c = [m^e \bmod N]$ when m is small

Plain RSA should never be used!

- RSA assumption only refers to hardness of computing the e^{th} root of c *in its entirety*
  - *Partial* information about the e^{th} root may be leaked
  - (In fact, this is the case)

Chosen-ciphertext attacks

- Of course, plain RSA cannot be CCA-secure since it is not even CPA-secure
  - … but these ciphertexts are completely malleable.
- Given ciphertext c for unknown message m, can compute c' = $[\alpha^e \cdot c \bmod N]$
  - What does this decrypt to?

How to fix plain RSA?

- One approach: use a *randomized* encoding
- I.e., to encrypt m
  - First compute some reversible, randomized mapping M = E(m)
  - Then set c := [M^e mod N]
- To decrypt c
  - Compute M := [c^d mod N]
  - Recover m from M

PKCS #1 v1.5

- Standard issued by RSA labs in 1993
- Idea: introduce *random padding*
  - E(m) = r|m

- I.e., to encrypt m
  - Choose random r
  - Compute the ciphertext c := [$(r|m)^e$ mod N]

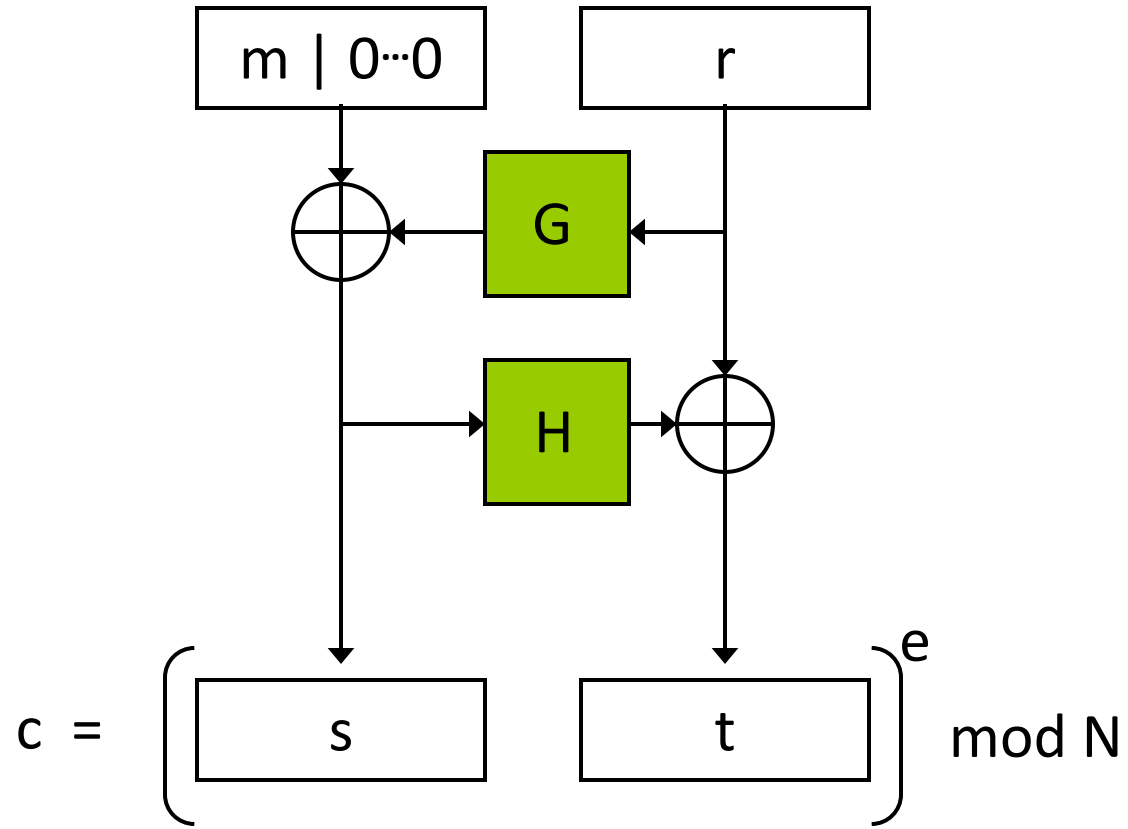
- Issues:
  - No proof of CPA-security (unless m is very short)
  - Chosen-plaintext attacks are known if r is too short
  - Chosen-ciphertext attacks possible

PKCS #1 v2.0

- *Optimal asymmetric encryption padding* (OAEP) applied to message first

- This padding introduces *redundancy*, so that not every $c \in \mathbb{Z}^*_N$ is a valid ciphertext
  - Need to check for proper format upon decryption
  - Return error if not properly formatted

OAEP



$H(s) \oplus t = r$ $\quad$ $G(r) \oplus s = m|0\ldots0$

Security?

- RSA-OAEP can be proven CCA-secure under the RSA assumption, if G and H are modeled as random oracles

- Widely used in practice…