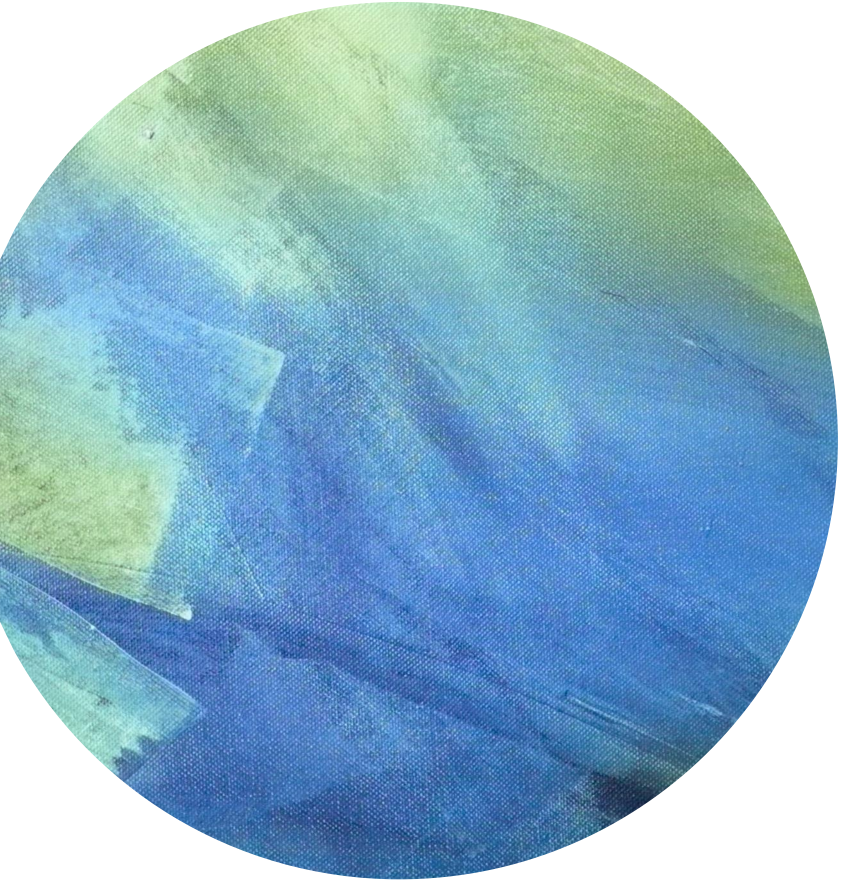
Lecture 14

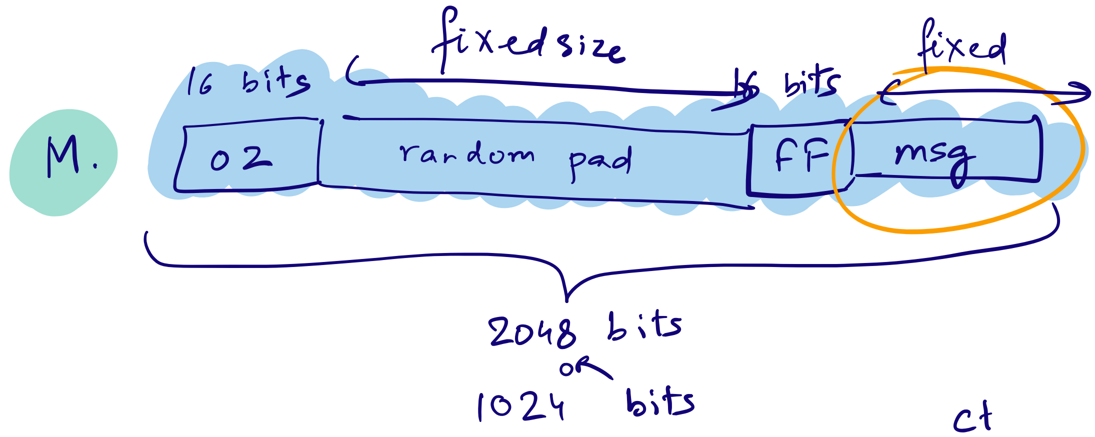
Scribe : Eshrit



Outline

CCA attacks

CCA security



$$\text{Textbook-RSA-Enc}_{e,N}(M) = \overset{ct}{M^e} \bmod N$$

$$\text{Dec}_{d,N}(ct) = ct^d \bmod N$$

$$= (M^e)^d \bmod N$$

$$= M^{(k\varphi(N)+1)} \bmod N$$

[Because $e \cdot d \equiv 1 \bmod \varphi(N)$]

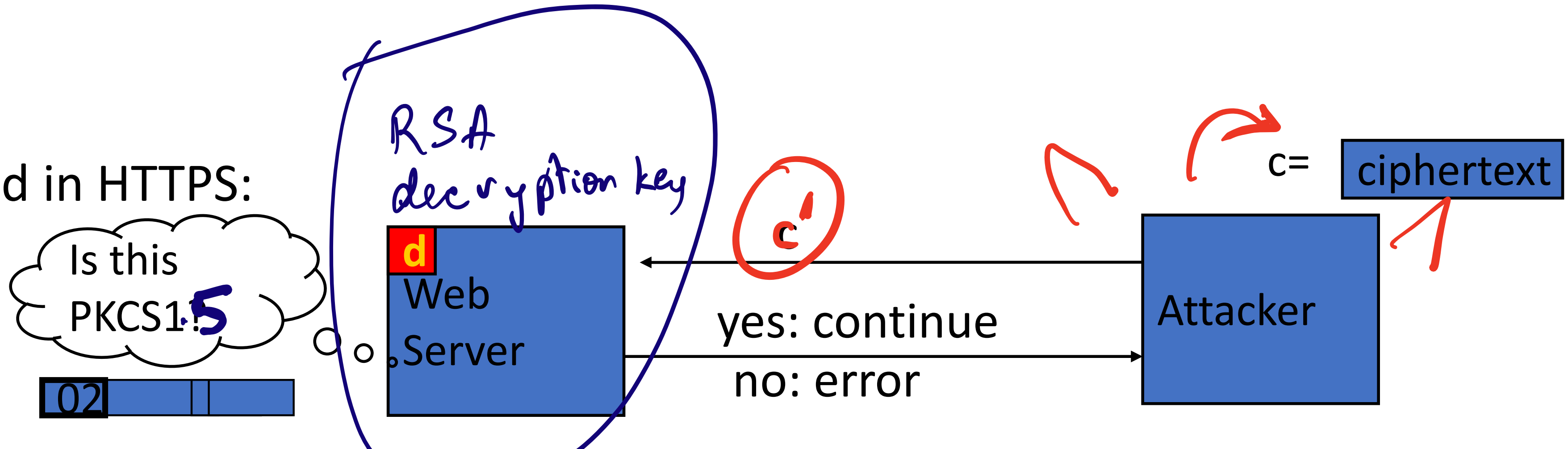
$$= \underbrace{\left(M^{\varphi(N)}\right)^k}_{=1} \cdot M \bmod N$$

(Euler's Thm)

$M \ll N$.

CCA Attack on PKCS1 v1.5 (Bleichenbacher 1998)

PKCS1 used in HTTPS:



⇒ attacker can test if 16 MSBs of plaintext = '02'

Does $(c')^d$ have msb = 02

Chosen-ciphertext attack: to decrypt a given ciphertext **c** do:

- Choose $r \in Z_N$. Compute $c' \leftarrow r^e \cdot c = \left(r \cdot \text{PKCS1}(m)\right)^e$
- Send c' to web server and use response

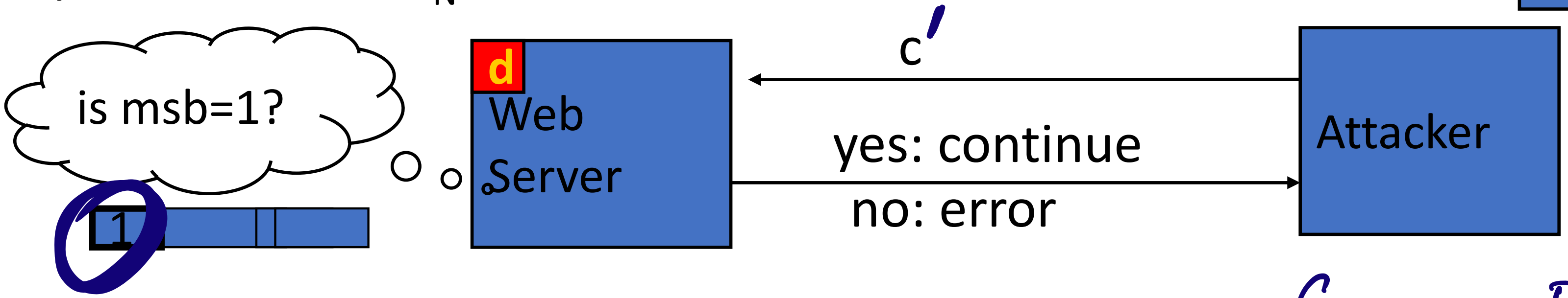
Baby Bleichenbacher

$c = M^e \bmod N$

$c' = c \cdot r^e \bmod N = (M r)^e \bmod N = \text{Enc}(M \cdot r)$

compute $x \leftarrow c^d$ in Z_N

c= ciphertext



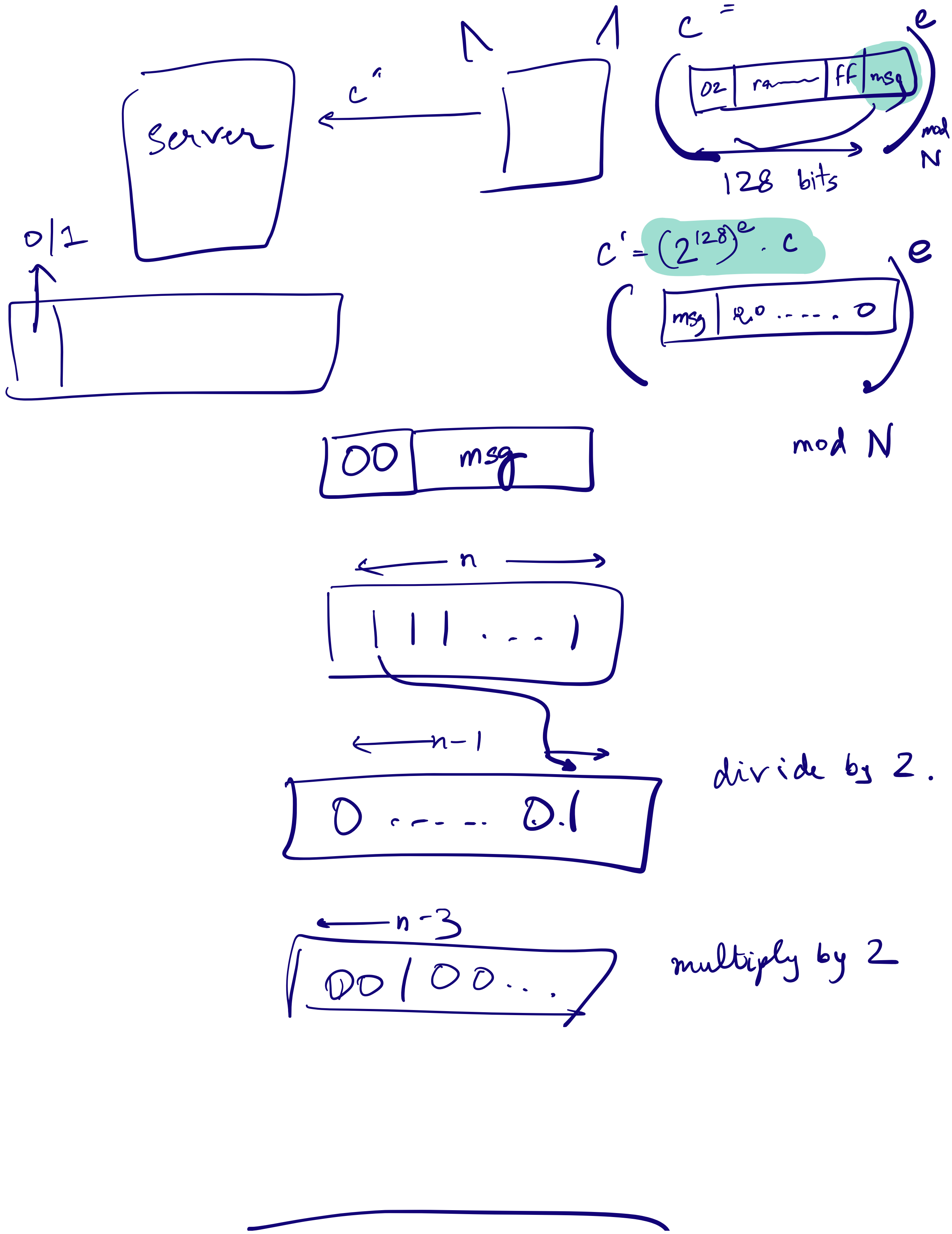
Given $c = \text{Enc}(M)$

$2^e c = \text{Enc}(2M)$

$4^e c = \text{Enc}(4M)$

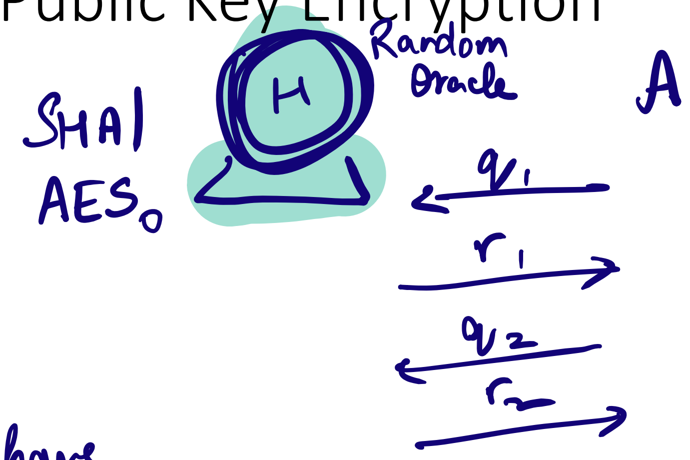
Suppose N is $N = 2^n$ (an invalid RSA modulus). Then:

- Sending c reveals msb(x)
- Sending $\mathbf{2^e \cdot c = (2x)^e}$ in Z_N reveals msb(2x mod N) = $\mathbf{msb_2(x)}$
- Sending $\mathbf{4^e \cdot c = (4x)^e}$ in Z_N reveals msb(4x mod N) = $\mathbf{msb_3(x)}$
- … and so on to reveal all of x


server
c'
0/1
c =
02
ff
msg
e
mod N
128 bits
c' = (2^128)^e . c
msg
0 0
e
mod N
00
msg
n
1 1 1 . . . 1
n-1
0 ..... 0.1
divide by 2.
n-3
00 1 0 0 . . .
multiply by 2

RANDOM ORACLE

Chosen Ciphertext (CCA) security for Public Key Encryption

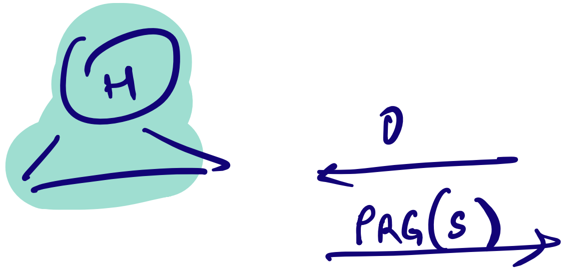


- H: Truly random function

- "Observable"

(PPT)

To obtain H(q), A must have computed H on q.



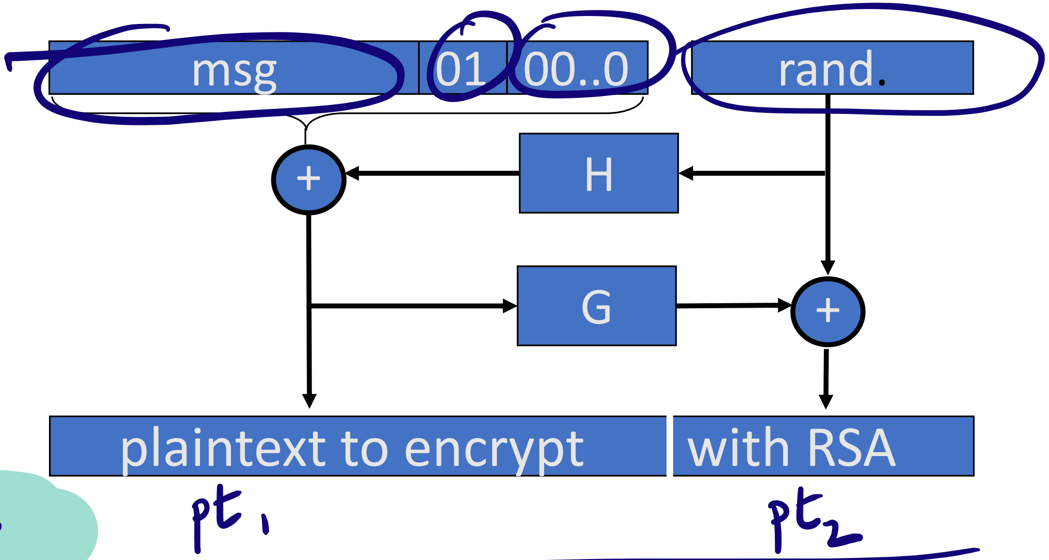
- "Programmable"

PKCS1 v2.0: OAEP

Bellare - Rogaway.

New preprocessing function: OAEP [BR94]

check pad
on decryption.
reject CT if invalid.



$pt_1 = (msg, 01, 0..0) \oplus H(rand)$

$pt_2 = G(pt_1) \oplus rand.$

$\rightarrow pt^e \bmod N$

Thm [FOPS'01] : RSA is a trap-door permutation $\Rightarrow$

RSA-OAEP is CCA secure when H,G are *random oracles*

in practice: use SHA-256 for H and G

What is a random oracle?

- H: Truly random function
- “Observable”
- “Programmable”

pt_1, pt_2

$rand = G(pt_1) \oplus pt_2$

$(msg, 01, 0\ldots0) = pt_1 \oplus H(rand)$

The factoring problem

Gauss (1805): *"The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic."*

Best known alg. (NFS): run time exp($\tilde{O}(\sqrt[3]{n})$) for n-bit integer

Current world record: **RSA-768** (232 digits)

- Work: two years on hundreds of machines
- Factoring a 1024-bit integer: about 1000 times harder

$\Rightarrow$ likely possible this decade

Summary

- Key concepts in number theory
- Hardness of discrete logarithm, factoring
- Diffie-Hellman key exchange from hardness of DDH
- Public key encryption => shared key derivation (called key exchange)