RSA cryptography

EE577A

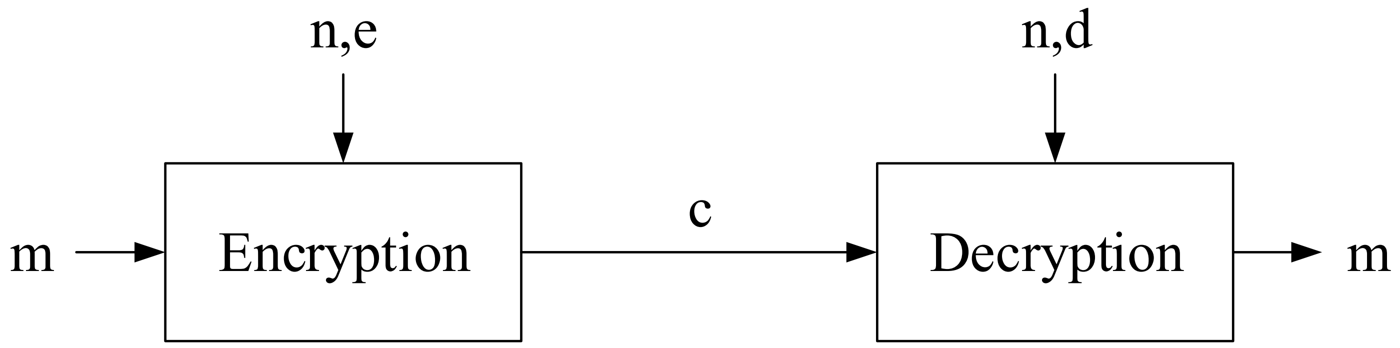
Fall 2019

RSA Outline

- Plain text: m
- Cipher text: c

- Public key: n, e
- Encryption: $c \equiv m^e \ mod \ n$

- Private key: n, d
- Decryption: $m \equiv c^d \ mod \ n$



Euler’s Totient Function and Theorem

- Euler’s totient function $\varphi(n)$:
- For a positive integer n, $\varphi(n)$ is the number of integer $k \in [1, n]$ where $GCD(k, n) = 1$.
- If n is a prime, $\varphi(n) = n - 1$.

- $\varphi(7) = 6$ since $k \in \{1,2,3,4,5,6\}$
- $\varphi(8) = 4$ since $k \in \{1,3,5,7\}$

- Euler’s Theorem:
- If n and a are coprime positive integers, we have

$$\mathrm{a}^{\varphi(n)} \equiv 1 \bmod \mathrm{n} \rightarrow \exists k_1 \in \mathbb{Z}, \mathrm{a}^{\varphi(n)} = 1 + k_1 n$$

Generation of Keys

- p, q: random large primes and $p \neq q$. (They are usually similar in magnitude.)
- $n = pq$ and $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$.
- e: an integer in $(1, \varphi(n))$ and $GCD\big(e, \varphi(n)\big) = 1$.
- $d \equiv e^{-1}\ mod\ \varphi(n) \rightarrow ed \equiv 1\ mod\ \varphi(n) \rightarrow \exists k_2 \in \mathbb{Z}, ed = 1 + k_2\varphi(n)$.

RSA Proof

- $m \equiv c^d \ mod\ n \rightarrow c \equiv m^e \ mod\ n$

- $c^d \ \equiv (m^e \ mod\ n)^d \ \equiv m^{ed} \ \equiv m^{1+k_2\varphi(n)} \ mod\ n$
- $c^d \ \equiv (m\ mod\ n) * \left(m^{\varphi(n)} \ mod\ n\right)^{k_2} \equiv m\ mod\ n$

- We usually choose $m, c \in [0, n)$.
- Modular exponentiation is a very complex operation.

Chinese Remainder Theorem (CRT)

- Assume $b_1, \dots, b_k$ are mutually coprime positive integers.
- We want to find a unique congruent solution x, so that

$$x \equiv a_i \ mod \ b_i, 1 \leq i \leq k$$

- for positive integers $a_1, \dots, a_k$.

- The solution is $x \equiv \left(\sum_{i=1}^{k} B_i B_i^{-1} a_i\right) mod \ b$

- $b = \prod_{i=1}^{k} b_i$, $B_i = \prod_{j=1, j\neq i}^{k} b_j = \frac{b}{b_i}$, and $B_i B_i^{-1} \equiv 1 \ mod \ b_i$

Chinese Remainder Theorem (CRT)

- If we want to find $x \equiv A \, mod \, b$ and $b = \prod_{i=1}^{k} b_i$.

- We can calculate $a_i \equiv A \, mod \, b_i$ so that

$$x \, mod \, b_i \equiv (A \, mod \, b) \, mod \, b_i \equiv A \, mod \, b_i \equiv a_i$$

- Therefore, $x \equiv \left(\sum_{i=1}^{k} B_i B_i^{-1} a_i\right) mod \, b$ is the solution of $x \equiv A \, mod \, b$.

RSA-CRT

- In RSA, we have $b_1 = p$, $b_2 = q$, $b = b_1 b_2 = pq = n$, and $A = c^d$.
- $M_p = a_1 \equiv c^d \, mod \, p$ and $M_q = a_2 \equiv c^d \, mod \, q$
- $B_1 = \frac{b}{b_1} = \frac{pq}{p} = q$ and $B_2 = p$
- $B_1 B_1^{-1} \equiv 1 \, mod \, b_1 \rightarrow q q^{-1} \equiv 1 \, mod \, p$
- $m \equiv c^d \equiv (B_1 B_1^{-1} a_1 + B_2 B_2^{-1} a_2) \equiv \left(q q^{-1} M_p + p p^{-1} M_q \right) mod \, n$

RSA-CRT

- $d = d_p + k_{dp}(p-1) = d_p + k_{dp}\varphi(p)$
- $M_p \equiv c^d \equiv c^{d_p + k_{dp}\varphi(p)} \equiv c^{d_p} mod\ p$
- $c = c_p + k_{cp}p$
- $M_p \equiv c^{d_p} \equiv \left(c_p + k_{cp}p\right)^{d_p} \equiv c_p^{d_p}\ mod\ p$
- $M_q \equiv c_q^{d_q}\ mod\ q$
- $m \equiv \left(qq^{-1}M_p + pp^{-1}M_q\right) mod\ n$

RSA-CRT

- Bezout's identity: $pp^{-1} + qq^{-1} \equiv 1\ mod\ n.$
- $m \equiv qq^{-1}M_p + pp^{-1}M_q \equiv qq^{-1}M_p + (1 - qq^{-1})M_q\ mod\ n$
- $m \equiv M_q + qq^{-1}\left(M_p - M_q\right) mod\ n$
- $h \equiv q^{-1}\left(M_p - M_q\right) \equiv q^{-1}\left(\left(M_p - M_q\right) mod\ p\right) mod\ p$
- $m \equiv \left(M_q + qh\right) mod\ n$

RSA-CRT

- If we choose remainders of all congruent classes.

- $m \equiv \left(M_q + qh\right) \, mod \, n$

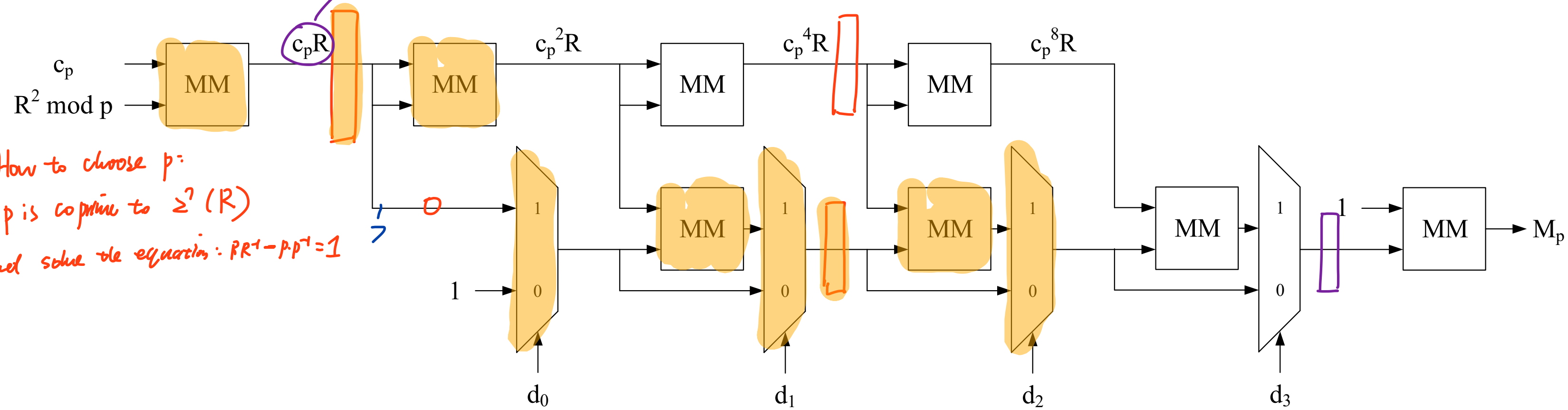
- $0 \leq h \leq p - 1 \rightarrow 0 \leq hq \leq pq - q = n - q$
- $0 \leq M_q \leq q - 1 \rightarrow 0 \leq M_q + qh \leq n - q + q - 1 = n - 1$
- $m \equiv \left(M_q + qh\right) \, mod \, n \rightarrow m = M_q + qh$

Modular Exponentiation

- $M_p \equiv c_p^{d_p} \ mod \ p$ and $\boxed{d_p = \sum_{i=0}^{n-1} d_i 2^i}$ so d_p has n bits
- $M_p \equiv \prod_{i=0}^{n-1} c_P^{d_i 2^i} \equiv \prod_{i=0}^{n-1} \left(c_p^{2^i}\right)^{d_i} \equiv \prod_{i=0}^{n-1} \left(c_p^{2^i} \ mod \ p\right)^{d_i} \ mod \ p$
- Example: n=4, MM is Montgomery modular multiplication.

n = 7

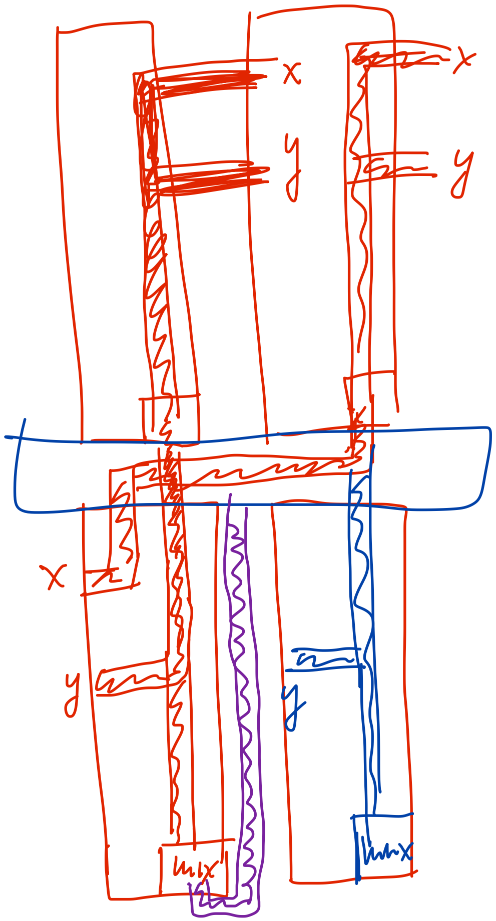
$C_p \cdot R^2 \ mod \ p \cdot R^{-1} \ mod \ p = C_p R \ mod \ p$.

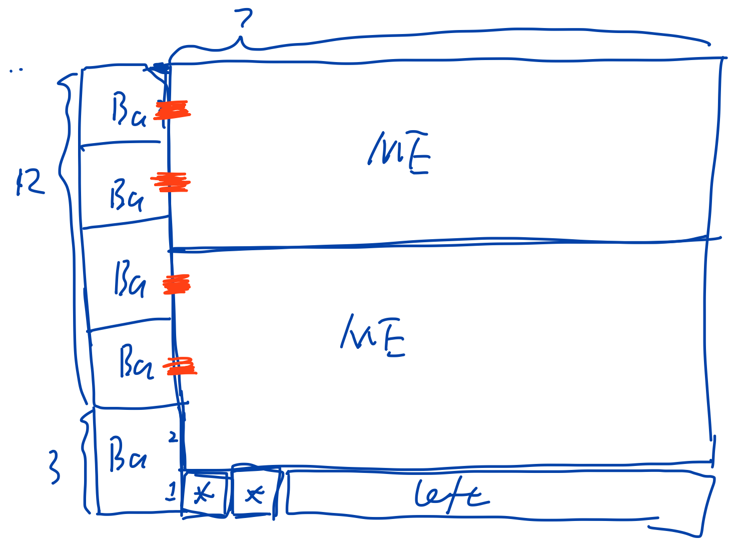


How to choose p:

p is coprime to $2^? (R)$

and solve the equation: $RR^{-1} - pp^{-1} = 1$

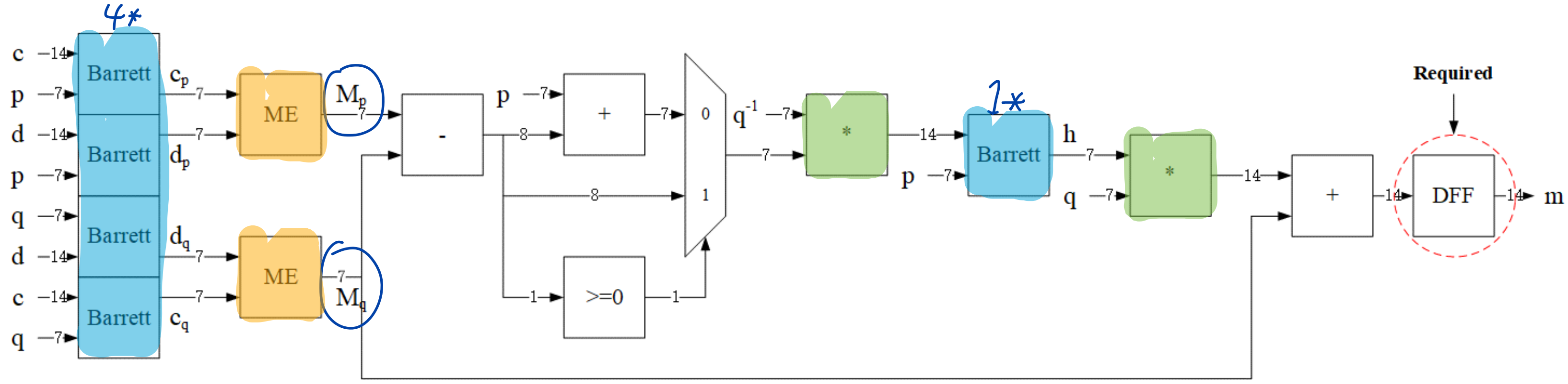

x
y
x
y
x
y
y
mix
mix


?
12
Ba
Ba
Ba
Ba
ME
ME
3
Ba
2
1
left

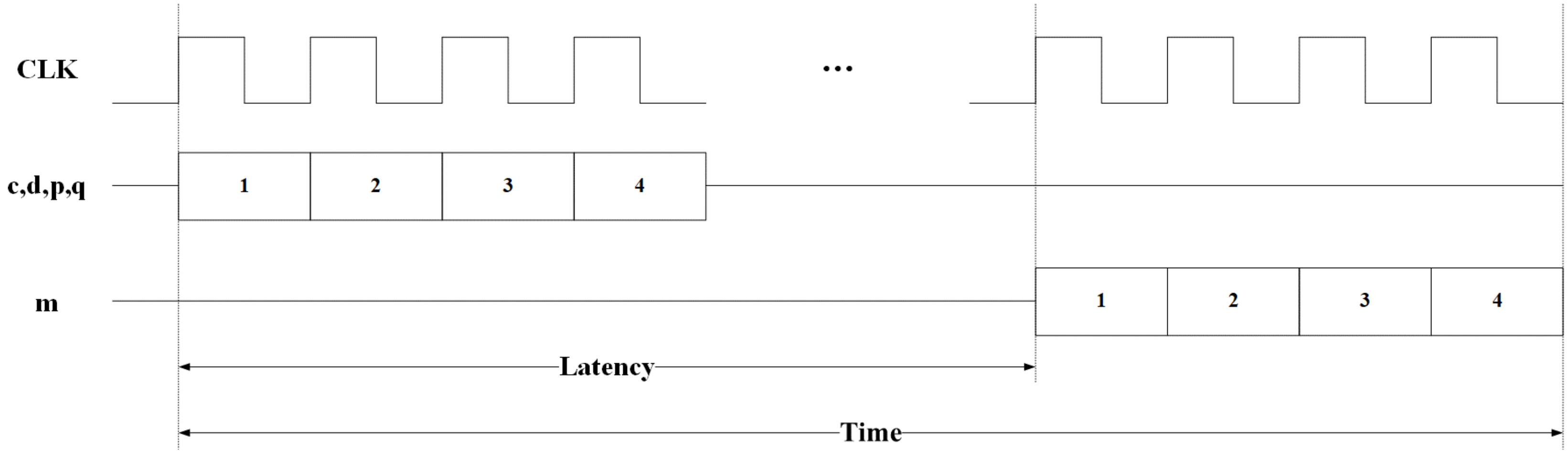
RSA Implementation

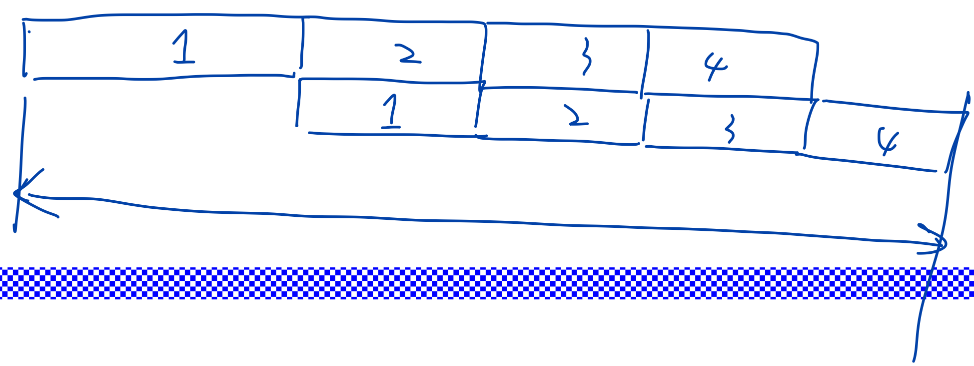
- Assume: $p > q$, 7-bit p, q, 14-bit c, d, m

d_p is 7-bit



Time





RSA Example: 10 bits

- $p = 1{,}009,\ q = 1{,}021,\ m = 2{,}003$
- $p^{-1} = 85,\ q^{-1} = 925,\ n = pq = 1{,}030{,}189,\ \varphi(n) = 1{,}008 * 1{,}020 = 1{,}028{,}160$
- $d = 939577 \rightarrow e = 1033,\ c = m^e \ mod\ n = 1{,}016{,}820$
- $c_p = c\ mod\ p = 757,\ c_q = c\ mod\ q = 925$

$p-1$ $q-1$

- $d_p = d\ mod\ (p-1) = 121,\ d_q = d\ mod\ (q-1) = 157$
- $M_p = c_p^{d_p}\ mod\ p = 994,\ M_q = c_q^{d_q}\ mod\ q = 982$
- $M_p - M_q\ mod\ p = 12,\ h = q^{-1}\left(M_p - M_q\right) mod\ p = 1$
- $m = M_q + qh = 2003$

RSA Project

Function								
Inputs								Outputs
Given				Python				Hardware
c	d	p	q	r_p	p^{-1}	r_q	q^{-1}	m
831	2971	127	113					
4624	9833	107	97					
4058	2831	113	109					
6757	6593	103	89					

Metric		
Time (ns)	Area (mm^2)	Area*Time