

**ECE646 Lecture 11**

**RSA Implementation:  
Efficient encryption, decryption  
& key generation**

1

---

---

---

---

**Efficient encryption  
and decryption**

2

---

---

---

---

**Number of bits vs. number of decimal digits**

$$10^{\#digits} = 2^{\#bits}$$

$$\#digits = (\log_{10} 2) \cdot \#bits \approx 0.30 \cdot \#bits$$

256 bits = 77 D  
384 bits = 116 D  
512 bits = 154 D  
768 bits = 231 D  
1024 bits = 308 D  
2048 bits = 616 D

3

---

---

---

---

### How to perform exponentiation efficiently?

$$Y = X^E \bmod N = \underbrace{X \cdot X \cdot X \cdot X \cdot X \dots \cdot X \cdot X}_{E\text{-times}} \bmod N$$

E may be in the range of  $2^{1024} \approx 10^{308}$

#### Problems:

1. huge storage necessary to store  $X^E$  before reduction
2. amount of computations infeasible to perform

#### Solutions:

1. modulo reduction after each multiplication
2. clever algorithms  
200 BC, India, "Chandah-Sâtra"

---

---

---

---

---

---

### Right-to-left binary exponentiation

$$Y = X^E \bmod N$$

$$E = (e_{L-1}, e_{L-2}, \dots, e_1, e_0)_2$$

$$S: X \quad X^2 \bmod N \quad X^4 \bmod N \quad X^8 \bmod N \quad \dots \quad X^{2^{L-1}} \bmod N$$

$$E: e_0 \quad e_1 \quad e_2 \quad e_3 \quad \dots \quad e_{L-1}$$

$$Y = X^{e_0} \cdot (X^2 \bmod N)^{e_1} \cdot (X^4 \bmod N)^{e_2} \cdot (X^8 \bmod N)^{e_3} \cdot \dots \cdot (X^{2^{L-1}} \bmod N)^{e_{L-1}}$$

$$\left| \quad (X^a)^b = X^{ab} \quad X^a \cdot X^b = X^{a+b} \quad \right|$$

$$Y = X^{e_0 + 2 \cdot e_1 + 4 \cdot e_2 + 8 \cdot e_3 + \dots + 2^{L-1} \cdot e_{L-1}} \bmod N =$$

$$= X^{\sum_{i=0}^{L-1} e_i \cdot 2^i} = X^E \bmod N$$

5

---

---

---

---

---

---

### Right-to-left binary exponentiation: Example

$$Y = 3^{19} \bmod 11$$

$$E = 19 = 16 + 2 + 1 = (10011)_2$$

$$S: X \quad X^2 \bmod N \quad X^4 \bmod N \quad X^8 \bmod N \quad X^{16} \bmod N$$

$$3 \quad 3^2 \bmod 11 = 9 \quad 9^2 \bmod 11 = 4 \quad 4^2 \bmod 11 = 5 \quad 5^2 \bmod 11 = 3$$

$$E: e_0 \quad e_1 \quad e_2 \quad e_3 \quad e_4$$

$$1 \quad 1 \quad 0 \quad 0 \quad 1$$

$$Y = X \cdot X^2 \bmod N \cdot 1 \cdot 1 \cdot X^{16} \bmod N =$$

$$3 \cdot 9 \cdot 1 \cdot 1 \cdot 3 \bmod 11$$

$$= X^{19} \bmod N$$

$$(27 \bmod 11) \cdot 3 \bmod 11 = 5 \cdot 3 \bmod 11 = 4$$

6

---

---

---

---

---

---

**Left-to-right binary exponentiation**  
 $Y = X^E \text{ mod } N$

$E = (e_{L-1}, e_{L-2}, \dots, e_1, e_0)_2$

E:       $e_{L-1}$      $e_{L-2}$      $e_{L-3}$     ...     $e_1$      $e_0$

$$Y = (((((1^2 \cdot X^{e_{L-1}})^2 \cdot X^{e_{L-2}})^2 \cdot X^{e_{L-3}})^2 \dots)^2 \cdot X^{e_1})^2 \cdot X^{e_0} \text{ mod } N$$

|     $(X^a)^b = X^{ab}$      $X^a \cdot X^b = X^{a+b}$     |

$$Y = X^{(e_{L-1} \cdot 2 + e_{L-2}) \cdot 2 + e_{L-3} \cdot 2 + \dots + e_1 \cdot 2 + e_0} \text{ mod } N =$$

$$= X^{2^{L-1} \cdot e_{L-1} + 2^{L-2} \cdot e_{L-2} + 2^{L-3} \cdot e_{L-3} + \dots + 2 \cdot e_1 + e_0} \text{ mod } N = X^{\sum_{i=0}^{L-1} e_i \cdot 2^i} \text{ mod } N =$$

7

---

---

---

---

---

---

---

---

**Left-to-right binary exponentiation: Example**  
 $Y = 3^{19} \text{ mod } 11$

$E = 19 = 16 + 2 + 1 = (10011)_2$

E:       $e_4$      $e_3$      $e_2$      $e_1$      $e_0$

1    0    0    1    1

$$Y = (((((1^2 \cdot X)^2 \cdot 1)^2 \cdot 1)^2 \cdot X)^2 \cdot X \text{ mod } N$$

$$= (((3^2 \text{ mod } 11)^2 \text{ mod } 11)^2 \text{ mod } 11 \cdot 3^2 \text{ mod } 11 \cdot 3 \text{ mod } 11$$

$$= (81 \text{ mod } 11)^2 \text{ mod } 11 \cdot 3^2 \text{ mod } 11 \cdot 3 \text{ mod } 11 =$$

$$= (5 \cdot 3)^2 \text{ mod } 11 \cdot 3 \text{ mod } 11 =$$

$$= 4^2 \text{ mod } 11 \cdot 3 \text{ mod } 11 =$$

$$= 5 \cdot 3 \text{ mod } 11 = 4$$

$Y = (X^8 \cdot X)^2 \cdot X \text{ mod } N = X^{19} \text{ mod } N$       8

---

---

---

---

---

---

---

---

**Exponentiation:  $Y = X^E \text{ mod } N$**

<p><b>Right-to-left binary exponentiation</b></p> <pre> Y = 1; S = X; for i=0 to L-1 {   if (e_i == 1)     Y = Y · S mod N;   S = S<sup>2</sup> mod N; } </pre>	<p><b>Left-to-right binary exponentiation</b></p> <pre> Y = 1; for i=L-1 downto 0 {   Y = Y<sup>2</sup> mod N;   if (e_i == 1)     Y = Y · X mod N; } </pre>
---	--

$E = (e_{L-1}, e_{L-2}, \dots, e_1, e_0)_2$

9

---

---

---

---

---

---

---

---

### Exponentiation Example: $Y = 7^{12} \bmod 11$

Right-to-left binary exponentiation

Left-to-right binary exponentiation

$$12 = (1100)_2$$

i		0	1	2	3
$e_i$		0	0	1	1
$S_{\text{before}}$		7	5	3	9
$Y_{\text{after}}$		1	1	3	5
$S_{\text{after}}$		7	5	3	4

i	3	2	1	0
$e_i$	1	1	0	0
Y	1	7	2	5

$S_{\text{before}}$  - S before round i is computed

$S_{\text{after}}$  - S after round i is computed

10

---



---

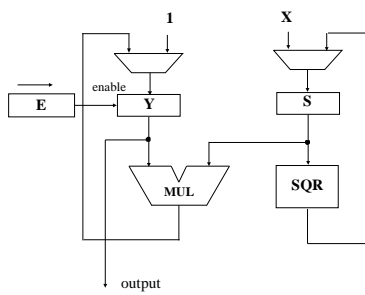


---



---

### Right-to-Left Binary Exponentiation in Hardware



11

---



---

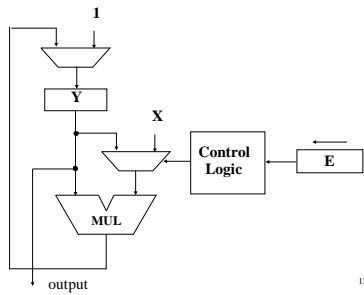


---



---

### Left-to-Right Binary Exponentiation in Hardware



12

---



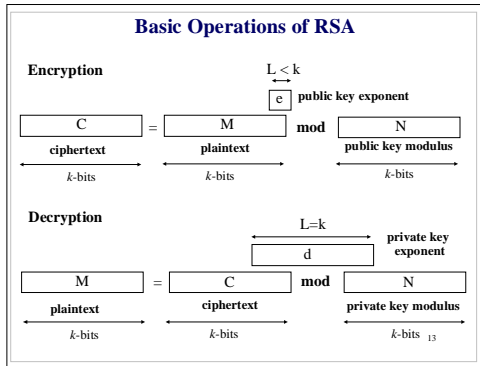
---



---



---




---

---

---

---

---

---

### Time of exponentiation

$t_{\text{EXP}}(e, L, k) = \# \text{modular\_multiplications}(e, L) \cdot t_{\text{MULMOD}}(k)$

e, L	#modular_multiplications
e=3	2
$e = F_2 = 2^{2^1} + 1$	17
large random L-bit e	$L + \# \text{ones}(e) \approx \frac{3}{2} \cdot L$

$t_{\text{MULMOD}}(k)$  - time of a single modular multiplication of two k-bit numbers modulo a k-bit number

<b>SOFTWARE</b>	<b>HARDWARE</b>
$t_{\text{MULMOD}}(k) = c_{\text{sm}} \cdot k^2$	$t_{\text{MULMOD}}(k) = c_{\text{hm}} \cdot k$ <sup>14</sup>

---

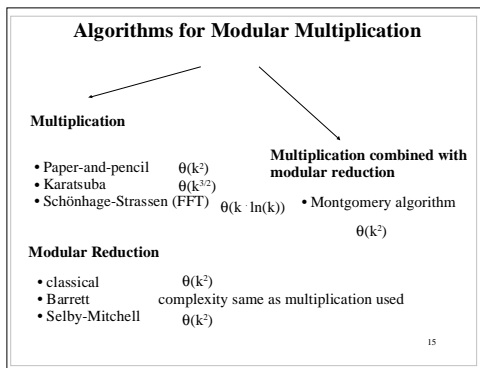
---

---

---

---

---




---

---

---

---

---

---



**Time of the RSA operations  
as a function of the key size k**

	SOFTWARE	HARDWARE
Encryption/ Signature verification with a small exponent e	$c_{se} \cdot k^2$	$c_{he} \cdot k$
Decryption / Signature generation	$c_{sd} \cdot k^3$	$c_{hd} \cdot k^2$
Key Generation	$c_{sk} \cdot k^4 / \log_2 k$	$c_{hk} \cdot k^3 / \log_2 k$
Factorization (breaking RSA)	$\exp(c_{sf} \cdot k^{1/3} \cdot (\ln k)^{2/3})$	19

---



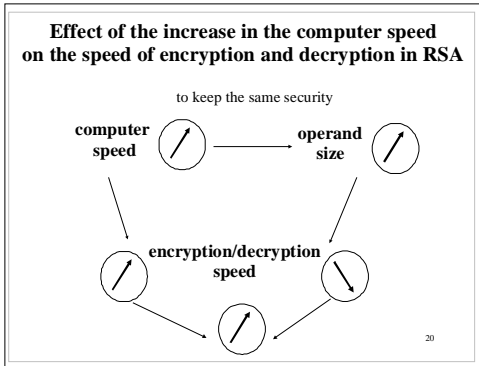
---



---



---




---



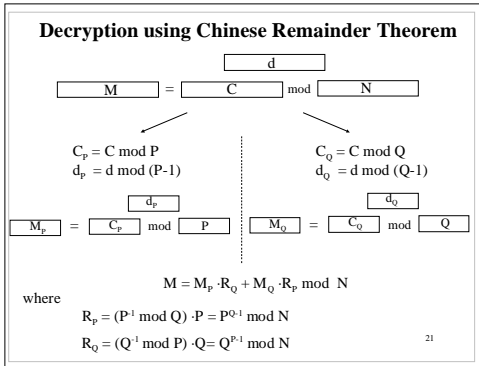
---



---



---




---



---



---



---

**Time of decryption  
without and with Chinese Remainder Theorem**

**SOFTWARE**

Without CRT  $t_{\text{DEC}}(k) = t_{\text{EXP}}(\text{random } e, k, L=k) = c_e \cdot k^3$

With CRT  $t_{\text{DEC-CRT}}(k) = 2 \cdot t_{\text{EXP}}(\text{random } e, k/2, L=k/2) = 2 \cdot c_e \cdot \left(\frac{k}{2}\right)^3 = \frac{1}{4} t_{\text{DEC}}(k)$

**HARDWARE**

Without CRT  $t_{\text{DEC}}(k) = t_{\text{EXP}}(\text{random } e, k, L=k) = c_h \cdot k^2$

With CRT  $t_{\text{DEC-CRT}}(k) = t_{\text{EXP}}(\text{random } e, k/2, L=k/2) = c_h \cdot \left(\frac{k}{2}\right)^2 = \frac{1}{4} t_{\text{DEC}}(k)$

---

---

---

---

---

---

**Chinese Remainder Theorem**

Let  $N = n_1 \cdot n_2 \cdot n_3 \cdot \dots \cdot n_M$

and for any  $i, j$   $\text{gcd}(n_i, n_j) = 1$

Then, any number  $0 \leq A \leq N-1$   
can be represented uniquely by

$A \leftrightarrow (a_1 = A \bmod n_1, a_2 = A \bmod n_2, \dots, a_M = A \bmod n_M)$

A can be reconstructed from  $(a_1, a_2, \dots, a_M)$  using equation

$$A = \sum_{i=1}^M (a_i \cdot N_i \cdot N_i^{-1} \bmod n_i) \bmod N \quad \text{where } N_i = \frac{N}{n_i} = n_1 \cdot n_2 \cdot \dots \cdot n_{i-1} \cdot n_{i+1} \cdot \dots \cdot n_M$$


---

---

---

---

---

---

**Chinese Remainder Theorem  
for  $N=P \cdot Q$**

$N = P \cdot Q \quad \text{gcd}(P, Q) = 1$

$M \leftrightarrow (M_p = M \bmod P, M_q = M \bmod Q)$

$$M = M_p \cdot \frac{N}{P} \cdot \left[ \left[ \frac{N}{P} \right]^{-1} \bmod P \right] + M_q \cdot \frac{N}{Q} \cdot \left[ \left[ \frac{N}{Q} \right]^{-1} \bmod Q \right] \bmod N$$

$$= M_p \cdot Q \cdot ((Q^{-1}) \bmod P) + M_q \cdot P \cdot ((P^{-1}) \bmod Q) \bmod N =$$

$$= M_p \cdot R_q + M_q \cdot R_p \bmod N$$


---

---

---

---

---

---

### Concealment of messages in the RSA cryptosystem

*Blakley, Borosh, 1979*

**There exist messages that are not changed by the RSA encryption!**

For example:

$$\begin{array}{ll} M=1 & C = 1^e \bmod N = 1 \\ M=0 & C = 0^e \bmod N = 0 \\ M=N-1 \equiv -1 \bmod N & C = (-1)^e \bmod N = -1 \end{array}$$

Every M such that

$$M_p = M \bmod P \in \{1, 0, -1\}$$

$$M_q = M \bmod Q \in \{1, 0, -1\}$$

$$C_p = C \bmod P = (M^e \bmod N) \bmod P = M^e \bmod P = M_p^e \bmod P = M_p$$

$$C_q = C \bmod Q = (M^e \bmod N) \bmod Q = M^e \bmod Q = M_q^e \bmod Q = M_q$$

---

---

---

---

---

---

### Concealment of messages in the RSA cryptosystem

*Blakley, Borosh, 1979*

**At least 9 messages not concealed by RSA!**

**Number of messages not concealed by RSA:**

$$\sigma = (1 + \gcd(e-1, P-1)) \cdot (1 + \gcd(e-1, Q-1))$$

A.  $e=3$        $\sigma = 9$

B.  $\gcd(e-1, P-1) = 2$  and  $\gcd(e-1, Q-1) = 2$        $\sigma = 9$

C.  $\gcd(e-1, P-1) = P-1$  and  $\gcd(e-1, Q-1) = Q-1$        $\sigma = P \cdot Q = N$

**It is possible that all messages remain unconcealed by RSA!**

26

---

---

---

---

---

---

### Efficient key generation

27

---

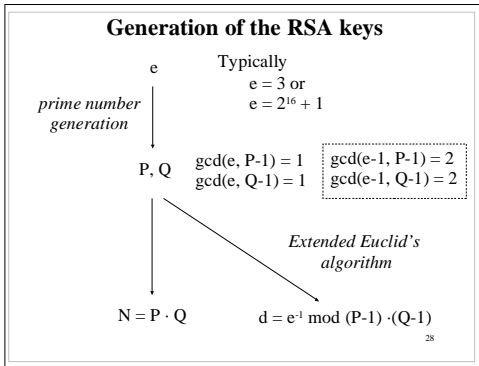
---

---

---

---

---




---

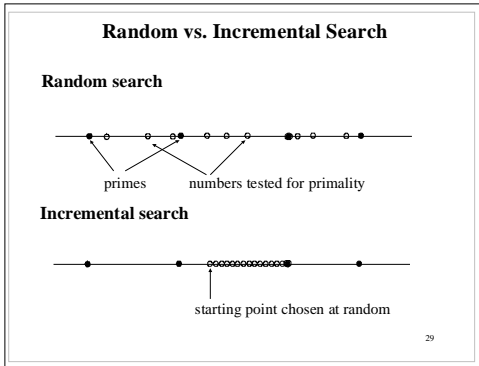
---

---

---

---

---




---

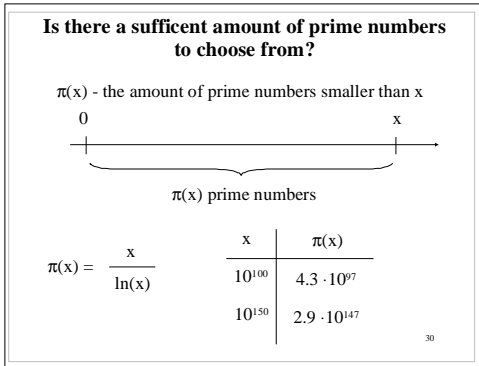
---

---

---

---

---




---

---

---

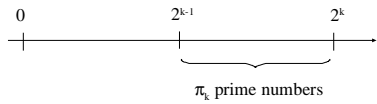
---

---

---

**Is there a sufficient amount of prime numbers of the given bit length to choose from?**

$\pi_k$  - the amount of prime numbers of the size of k-bits



$$\begin{aligned} \pi_k &= \pi(2^k) - \pi(2^{k-1}) \approx \\ &\approx 0.5 \cdot \pi(2^k) \approx \\ &\approx \pi(2^{k-1}) \end{aligned}$$

k	$\pi_k$
384	$7 \cdot 10^{112}$
512	$4 \cdot 10^{151}$
1024	$2.5 \cdot 10^{305}$

31

---

---

---

---

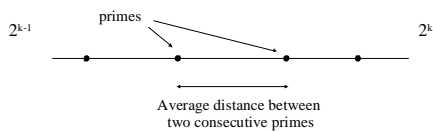
---

---

---

---

**Average distance between primes of the given bit length (1)**



$$\begin{aligned} \text{Average distance (k)} &\approx \frac{2^k - 2^{k-1}}{\pi_k} \approx \frac{2^{k-1}}{\pi(2^{k-1})} \approx \ln 2^{k-1} \approx \\ &\approx 0.69 \cdot (k-1) \end{aligned}$$

32

---

---

---

---

---

---

---

---

**Average distance between primes of the given bit length (2)**

Number of bits k	Average distance between primes	Average amount of odd numbers to test
256	177	88
384	265	132
512	354	177
1024	709	355

33

---

---

---

---

---

---

---

---

### Fermat's primality test

**n composite**

The diagram shows a large outer oval labeled  $\{1..n-1\}$ . Inside it are two smaller ovals:  $L(n)$  on the left and  $W(n)$  on the right.  $L(n)$  is labeled "Liars to the compositeness of n".  $W(n)$  is labeled "Witnesses to the compositeness of n".

$a \in W(n)$  iff  $a^{n-1} \neq 1 \pmod n$

34

---



---



---



---

### Fermat's test

**n composite Carmichael number**

The diagram shows a large outer oval labeled  $\{1..n-1\}$ . Inside it are two smaller ovals:  $L(n)$  on the left and  $W(n)$  on the right.  $L(n)$  is labeled "Liars to the compositeness of n".  $W(n)$  is labeled "Witnesses to the compositeness of n".

$W(n) = \{a: 1 \leq a \leq n, \gcd(a, n) > 1\}$   
 $|L(n)| = \phi(n)$   
 $|W(n)| = n - \phi(n)$

35

---



---



---



---

### Carmichael numbers

A composite integer is a Carmichael number iff

$$n = p_1 \cdot p_2 \cdot p_3 \cdot \dots \cdot p_k$$

$k \geq 3$

$p_i$  are distinct primes,  $p_i \neq p_j$  for  $i \neq j$

$$\forall p_i \quad (p_i - 1) \mid (n - 1)$$


---

Smallest Carmichael number  
 $n = 561 = 3 \cdot 11 \cdot 17$

---

Among all numbers smaller or equal to  $10^{15}$   
 There are about  $3 \cdot 10^{13}$  prime numbers  
 $10^5$  Carmichael numbers<sup>6</sup>

---



---



---



---

### Good probabilistic primality test

**n composite**

{1..n-1}

$\forall n \text{ composite} \quad |W(n)| \geq |L(n)|$

If  $a \in W(n)$  test returns "n composite"  
 else test returns "n probably prime"  
 or "n pseudoprime to the base a"<sup>37</sup>

---



---



---



---

### Miller-Rabin test

**n composite**

{1..n-1}

$\forall n \text{ composite} \quad |L(n)| \leq \phi(n)/4 < (n-1)/4$

38

---



---



---



---

### Miller-Rabin test

**n composite**

{1..n-1}

For certain composite numbers, such as  
 $n = 3 \cdot 5 \cdot 7 \cdot \dots \cdot (2k+1)$   
 there are only two strong liars: 1 and n-1

39

---



---



---



---

**Miller-Rabin test**  
**Mathematical Basis**

If  $n$  is **prime** then

$1$  has **only two** square roots modulo  $n$   
 i.e., there are only two numbers,  $y_1$  and  $y_2$ , such that  
 $y_1^2 \bmod n = 1$  and  $y_2^2 \bmod n = 1$   
 $y_1 = 1$  and  $y_2 = n-1 \equiv -1 \pmod n$

If  $n$  is **composite** then

$1$  has **at least four** square roots modulo  $n$   
 i.e., there exist numbers,  $y_1, y_2, y_3, y_4$ , such that  
 $y_1^2 \bmod n = 1, y_2^2 \bmod n = 1, y_3^2 \bmod n = 1, y_4^2 \bmod n = 1,$   
 $y_1 = 1, y_2 = n-1 \equiv -1 \pmod n, y_3 \not\equiv \pm 1 \pmod n, y_4 \not\equiv \pm 1 \pmod n$

---

---

---

---

---

---

**Miller-Rabin test**  
**Algorithm (1)**

Find  $s$  and  $r$ , such that

$$n - 1 = 2^s \cdot r, \text{ where } r \text{ is odd}$$

For example:

$$n = 49 \\ n - 1 = 48 = 2^4 \cdot 3 \quad s=4, r=3$$

$$n = 61 \\ n - 1 = 60 = 2^2 \cdot 15 \quad s=2, r=15$$

---

---

---

---

---

---

**Miller-Rabin test**  
**Algorithm (2)**

Compute

$$a^{n-1} \bmod n = \underbrace{(\dots((a^r \bmod n)^2 \bmod n)^2 \bmod n \dots)^2 \bmod n = 1}_{s \text{ squarings}}$$

$$\begin{array}{c} \xrightarrow{\text{square mod } n} \\ a^r \quad (a^r)^2 \quad (a^r)^{2^2} \quad (a^r)^{2^3} \dots (a^r)^{2^{s-1}} \quad (a^r)^{2^s} \quad \bmod n \\ \xleftarrow{\text{square root mod } n} \end{array}$$

---

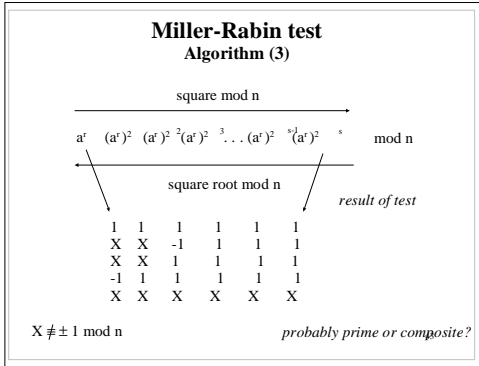
---

---

---

---

---




---

---

---

---

---

---

**$-\log_2$  of the bound on the error probability of declaring a  $k$ -bit composite number a prime after  $t$  iterations of the Miller-Rabin test**

$k$ = number of bits of $n$	$t$ - number of iterations of the Miller-Rabin test									
	1	2	3	4	5	6	7	8	9	10
100	5	14	20	25	29	33	36	39	41	44
150	8	20	28	34	39	43	47	51	54	57
200	11	25	34	41	47	52	57	61	65	69
250	14	29	39	47	54	60	65	70	75	79
300	19	33	44	53	60	67	73	78	83	88
400	37	46	55	63	72	80	87	93	99	105
500	56	63	70	78	85	92	99	106	113	119
600	75	82	88	95	102	108	115	121	127	133

44

---

---

---

---

---

---

**Minimal number of the Miller-Rabin tests  $t$ , necessary to obtain the probability of error  $< 2^{-100}$  for a  $k$ -bit number  $n$**

$k$	$t$	$k$	$t$	$k$	$t$
160	34	202-208	23	335-360	12
161-163	33	209-215	22	<b>361-392</b>	<b>11</b>
164-166	32	216-222	21	393-430	10
167-169	31	223-231	20	431-479	9
170-173	30	232-241	19	<b>480-542</b>	<b>8</b>
174-177	29	242-252	18	543-626	7
178-181	28	253-264	17	627-746	6
182-185	27	265-278	16	747-926	5
186-190	26	279-294	15	<b>927-</b>	<b>4</b>
				<b>1232</b>	
191-195	25	295-313	14	1233-1853	3
196-201	24	314-334	13	over 1853	2

45

---

---

---

---

---

---

### Random vs. Incremental Search

**Random search**

**Incremental search**

46

---



---



---



---

### Using division by small primes

**D** – Division by small primes  
**R<sub>2</sub>** – Miller-Rabin test with base 2  
**R** – Miller-Rabin test with the random base a

47

---



---



---



---

### Merten's Theorem

The proportion of candidate odd integers NOT ruled out by the trial division by all primes  $\leq B$

$$\alpha(B) = (1-1/3) \cdot (1-1/5) \cdot (1-1/7) \cdot \dots \cdot (1-1/B)$$

$$\alpha(B) \approx 1.12 / \ln B$$

For  $B=256$ ,  $\alpha(B) \approx 0.2$   
 80% of tested numbers discarded by the trial division

48

---



---



---



---

**Incremental search for a prime**  
Efficient implementation of division by small primes

Set of small primes	3	5	7	11
$n_0 = 91$	$n_0 \bmod 3 = 1$	$n_0 \bmod 5 = 1$	$n_0 \bmod 7 = \underline{0}$	$n_0 \bmod 11 = 3$
$n = 93$	$1+2 \bmod 3 = \underline{0}$	$1+2 \bmod 5 = 3$	$0+2 \bmod 7 = 2$	$3+2 \bmod 11 = 5$
$n = 95$	$0+2 \bmod 3 = 2$	$3+2 \bmod 5 = \underline{0}$	$2+2 \bmod 7 = 4$	$5+2 \bmod 11 = 7$
$n = 97$	$2+2 \bmod 3 = 1$	$0+2 \bmod 5 = 2$	$4+2 \bmod 7 = 6$	$7+2 \bmod 11 = 9$
$n = 99$	$1+2 \bmod 3 = \underline{0}$	$2+2 \bmod 5 = 4$	$6+2 \bmod 7 = 1$	$9+2 \bmod 11 = \underline{0}$
$n = 101$	$0+2 \bmod 3 = 2$	$4+2 \bmod 5 = 1$	$1+2 \bmod 7 = 3$	$0+2 \bmod 11 = 2$

---

---

---

---

---

---

---

---

**Division by small primes – Practical implementation (2)**

	3	5	7	11	S[k]
91			1		1
93	1				1
95		1			1
97	3				0
99	1		7		1
101		5		1	0
103	3				0
105	1	1			1
107			1		0
109	3				0
111		5			0
113	1			11	1
115	3		7		0
117		1			1
119	1				1
121	3		1		1
				1	1

80

---

---

---

---

---

---

---

---

**Optimum number of small primes**

$\{ 3, 5, 7, \dots, B_{opt} \}$

$$B_{opt} \approx \frac{R_2}{D \cdot \ln(R_2/D)}$$

$R_2$  = time of the Miller-Rabin test with base 2  
 $D$  = time spent on test dividing one number by one small prime

81

---

---

---

---

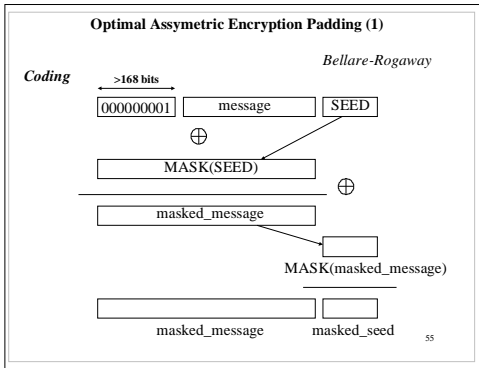
---

---

---

---






---



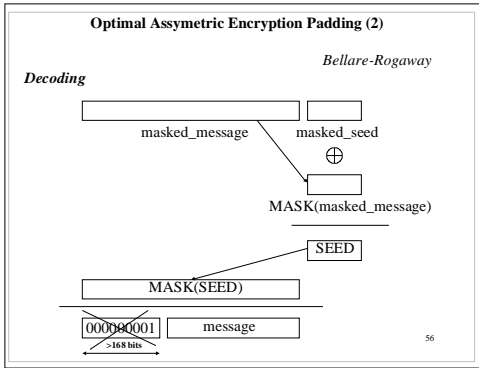
---



---



---




---



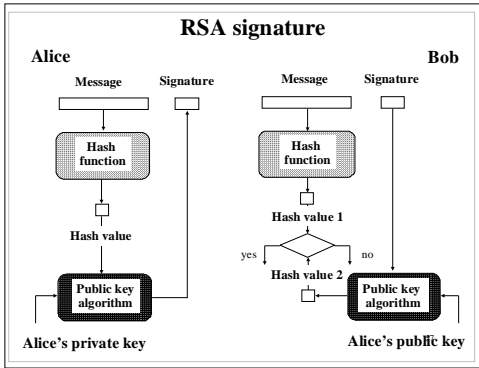
---



---



---




---



---



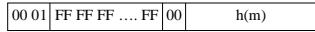
---



---

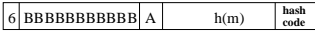
### Padding for signatures with appendix

PKCS #1 for signatures



at least 8 bytes  
←-----→

ISO-14888



33CC for SHA-1  
31CC for RIPEMD-160

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---