

Fundamentals of Information & Network Security

ECE 471/571



Lecture #19: RSA (continued)

Instructor: Ming Li

Dept of Electrical and Computer Engineering
University of Arizona

Chinese Remainder Theorem (CRT)

- Solving systems of congruences
 - Suppose m_1, m_2, \dots, m_r are pairwise relatively prime positive integers
 - a_1, a_2, \dots, a_r are integers,

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

....

$$x \equiv a_r \pmod{m_r}$$

The CRT asserts that this system has a unique solution mod $M = m_1 * m_2 * \dots * m_r$.

Example?

$$x = \sum_{i=1}^r a_i M_i y_i \pmod{M},$$

where $M_i = M / m_i$ and $y_i = M_i^{-1} \pmod{m_i}$, for $1 \leq i \leq r$

Decryption Optimization using CRT

Suppose that $Dec_K(y) = y^d \bmod n$ and $n = pq$. Define $d_p = d \bmod (p - 1)$ and $d_q = d \bmod (q - 1)$; and let $M_p = q^{-1} \bmod p$ and $M_q = p^{-1} \bmod q$.

Algorithm 1: CRT-OPTIMIZED RSA DECRYPTION

Input: $(n, d_p, d_q, M_p, M_q, y)$

$$x_p \leftarrow y^{d_p} \bmod p$$

$$x_q \leftarrow y^{d_q} \bmod q$$

$$x \leftarrow M_p q x_p + M_q p x_q \bmod n$$

Return (x)

Correctness: use Fermat's Little Theorem and CRT.

Result in a computational saving of 75%.

Finding Big Primes

- How many primes are there?
- The probability of a random chosen number n being prime is $1/\ln n$.
 - for a hundred-digit number, the chance is 1 in 230.
- Test whether a random number n is a prime.
 - Fermat's Theorem: if p is a prime and $0 < a < p$, then $a^{p-1} = 1 \pmod p$
 - For a non-prime n of a hundred digits, the chance of $a^{n-1} = 1 \pmod n$ is about 1 in 10^{13}
 - Unfortunately, there are Carmichael numbers (very rare) that show $a^{n-1} = 1 \pmod n$ for all a 's
 - Miller-Rabin algorithm

Finding Big Primes

- Prior to 2002, there was no known method of efficiently proving the primality of very large numbers. All of the algorithms in use, including the most popular (Miller-Rabin), produced a probabilistic result.
- In 2002, Agrawal, Kayal, and Saxena developed a relatively simple deterministic algorithm that efficiently determines whether a given large number is a prime. The algorithm, known as the AKS algorithm, does not appear to be as efficient as the Miller-Rabin algorithm. Thus far, it has not supplanted this older, probabilistic technique.

Finding e and d

- e: public key, can be randomly chosen, relatively prime to $\phi(n)$
- d: private key, is calculated by Euclid's algorithm, $ed=1 \bmod \phi(n)$
- Small e makes public key operations (e.g., encryption, signature verification) faster, while leaving private key operations (e.g., decryption, signature signing) unchanged
- d should not be small (Why?)

Two popular values of e

- 3 and 65537 ($2^{16}+1$)
- Advantage: efficient computation
 - 3 : 2 multiplies
 - 65537 : 17 multiplies

Problems of $e=3$

- #1:
 - if $m < n^{1/3}$, then $m = c^{1/3}$
 - Solution: Pad m to be larger than $n^{1/3}$
- #2:
 - If one message m is encrypted with three public keys $\langle 3, n_1 \rangle, \langle 3, n_2 \rangle, \langle 3, n_3 \rangle$. By Chinese Remainder theorem, one can compute $c = m^3 \bmod n_1 n_2 n_3$ from c_1, c_2, c_3 . Since $m < n_1, m < n_2, m < n_3$, then $m = c^{1/3}$
 - Solution: pad m with different numbers when generating c_1, c_2, c_3
- #3:
 - picking p and q such that 3 is relatively prime to $(p-1)(q-1)$
 - It is easier to choose eligible p and q for 65537.

Attacks on RSA

- Brute-force attacks: trying all possible private keys
- Mathematical attacks: trying to factor the product of two primes
- Chosen ciphertext attacks: exploit properties of the RSA algorithm
- Timing attacks: depend on the running time of the decryption algorithm (one type of side channel attacks)

Countermeasures

- Brute-force attacks: use a large key space
- Mathematical attacks: use large enough n (1024-2048 bits), select p and q with constraints
- Timing attacks: constant exponentiation time, random delay, blinding the ciphertext
- Chosen ciphertext attacks: randomly pad the plaintext before encryption, e.g., optimal asymmetric encryption padding (OAEP)

Required Key Length

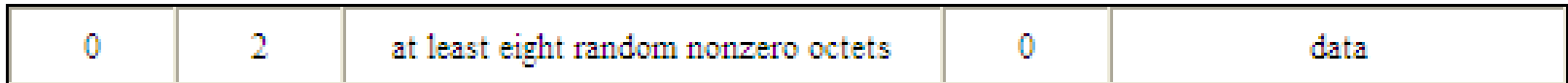
- Comparable key sizes in terms of computational effort for cryptanalysis

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521
Table 1: NIST Recommended Key Sizes		

Source: NSA website

PKCS—Public Key Cryptography Standard: Encryption

- Standard for the encoding of information that will be signed or encrypted through RSA
- A suite of standards PKCS #1—15
- PKCS #1 for formatting a message to be encrypted:



- The encoding addresses several RSA threats:
 - guessable message
 - sending same encrypted message to ≥ 3 recipients ($e=3$)
 - Encrypting messages $< 1/3$ length of n ($e=3$)