Chapters 8 and 9

More Number Theory and RSA Algorithm

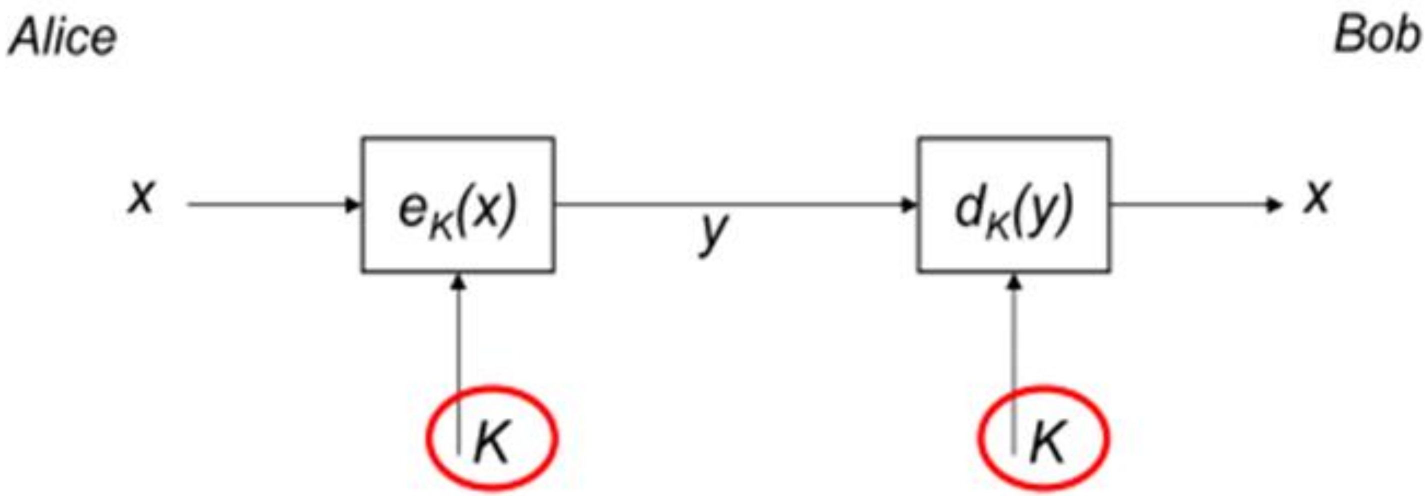
Dr. Shin-Ming Cheng

NTUST
CONNECTIVITY LAB

Introduction to Public-Key Cryptography

- Symmetric Cryptography Revisited
- Principles of Asymmetric Cryptography
- Practical Aspects of Public-Key Cryptography
- Important Public-Key Algorithms
- Essential Number Theory for Public-Key
- Algorithms

Symmetric Cryptography revisited



- Two properties of symmetric (secret-key) crypto-systems
  - The same secret key K is used for encryption and decryption
  - Encryption and Decryption are very similar (or even identical) functions

Symmetric Cryptography: Analogy

- Safe with a strong lock, only Alice and Bob have a copy of the key
  - Alice encrypts -> locks message in the safe with her key
  - Bob decrypts -> uses his copy of the key to open the safe

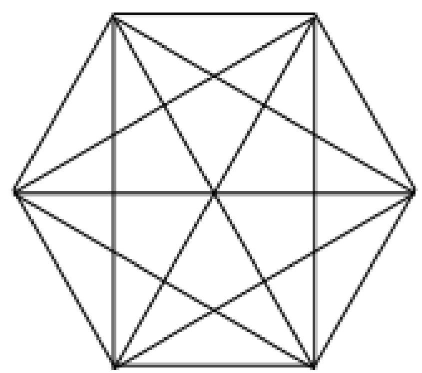
Symmetric Cryptography: Shortcomings

- Advantages: very secure, fast, widespread
- Key distribution problem: The secret key must be transported securely
- Number of keys:
  - In a network, each pair of users requires an individual key $\rightarrow n$ user in the network require $\frac{(n\times(n-1))}{2}$ keys, each user store $(n-1)$ keys

Example:

6 users (nodes)

$\frac{6\cdot 5}{2} = 15$ keys (edges)



Idea behind Asymmetric Cryptography

New Idea:

Use the "good old mailbox" principle:

Everyone can drop a letter

But: Only the owner has the correct key to open the box

1976: first publication of such an algorithm by Whitfield Diffie and Martin Hellman,and also by Ralph Merkle.

Asymmetric (Public-Key) Cryptography

Principle: "Split up" the key

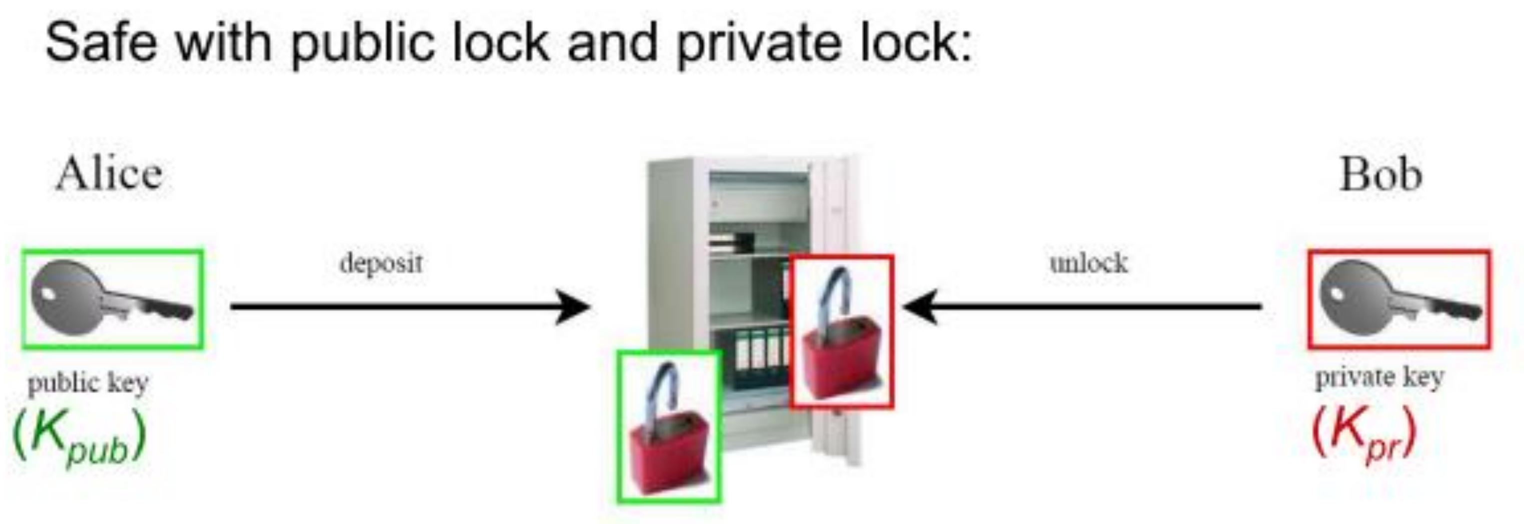
K

Public Key (K_{pub})
(Encrypt)

Secret Key (K_{pr})
(Decrypt)

→ During the key generation, a key pair K_{pub} and K_{pr} is computed

Asymmetric Cryptography: Analogy



- Alice deposits (encrypts) a message with the - not secret - public key K_{pub}
- Only Bob has the - secret - private key K_{pr} to retrieve (decrypt) the message

Basic Protocol for Public-Key Encryption

Alice		Bob
		$(K_{pubB}, K_{prB}) = K$
	$\xleftarrow{\quad K_{pubB} \quad}$	
x		
$y = e_{K_{pubB}}(x)$	$\xrightarrow{\quad y \quad}$	
		$x = d_{K_{prB}}(y)$

Security Mechanisms of Public-Key Cryptography

- Key Distribution without a pre-shared secret
  - Diffie-Hellman key exchange, RSA
- Nonrepudiation and Digital Signatures to provide message integrity
  - RSA, DSA or ECDSA
- Identification, using challenge-response protocols with digital signatures
- Encryption
  - RSA / ElGamal
  - Disadvantage: Computationally very intensive
    - (1000 times slower than symmetric Algorithms!)

Basic Key Transport Protocol (1/2)

- In practice: Hybrid systems
  - incorporating asymmetric and symmetric algorithms
- Key exchange (for symmetric schemes) and digital signatures are performed with (slow) asymmetric algorithms
- Encryption of data is done using (fast) symmetric ciphers
  - block ciphers or stream ciphers

Basic Key Transport Protocol (2/2)

Example: Hybrid protocol with AES as the symmetric cipher

Alice		Bob
		$(K_{pubB}, K_{prB}) = K$
	$\xleftarrow{K_{pubB}}$	
Choose random symmetric key K		
$y_1 = e_{K_{pubB}}(K)$	$\xrightarrow{y_1}$	
		$K = d_{K_{prB}}(y_1)$
message x		
$y_2 = AES_K(x)$	$\xrightarrow{y_2}$	$x = AES^{-1}_K(y_2)$

Key Exchange (asymmetric)

Data Encryption (symmetric)

How to build Public-Key Algorithms (1/2)

- Asymmetric schemes are based on a "one-way function" $f(\cdot)$:
  - Computing $y = f(x)$ is computationally easy
  - Computing $x = f^{-1}(y)$ is computationally infeasible
- One way functions are based on mathematically hard problems
  - The problems are considered mathematically hard, but no proof exists (so far)

How to build Public-Key Algorithms (2/2)

- Factoring integers
  - RSA
  - Given a composite integer n, find its prime factors
    - Multiply two primes: easy
- Discrete Logarithm
  - Diffie-Hellman, Elgamal, DSA
  - Given a, y and m, find x such that $a^x = y \bmod m$
    - Exponentiation a^x : easy
- Elliptic Curves (EC):
  - ECDH, ECDSA
  - Generalization of discrete logarithm

Key Lengths and Security Levels

Symmetric	*ECC*	*RSA, DL*	*Remark*
64 Bit	128 Bit	≈ 700 Bit	Only short term security (a few hours or days)
80 Bit	160 Bit	≈ 1024 Bit	Medium security (except attacks from big governmental institutions etc.)
128 Bit	256 Bit	≈ 3072 Bit	Long term security (without quantum computers)

Leonhard Euler

- Leonhard Euler (1707 – 1783)
  - Swiss mathematician and physicist
  - Made important discoveries in fields as diverse as calculus, number theory and topology, and introduced much of the modern mathematical terminology and notation
  - Also renowned for his work in mechanics, optics and astronomy
  - Considered to be the preeminent mathematician of the 18th century and one of the greatest of all time

Euler's Phi Function $\phi(n)$

- **Definition** The Euler phi function (or Euler totient function) is defined by $\phi(n) = |\{x | 1 \leq x \leq n, x \perp n\}|$
- **Remark**
  - We will derive the following properties
    - $\phi(p) = p - 1$ for every prime p
    - $\phi(p^k) = p^{k-1}(p - 1)$
    - $\phi(mn) = \phi(m)\phi(n)$ for $m \perp n$

Euler's Phi Function $\phi(n)$

- $\{0,1,2,3,4,5\}(m=6)$
  - $\gcd(0,6)=6$
  - $\gcd(1,6)=1$
  - $\gcd(2,6)=2$
  - $\gcd(3,6)=3$
  - $\gcd(4,6)=2$
  - $\gcd(5,6)=1$
- 1 and 5 relatively prime to $m = 6$, hence $\phi(6)=2$
- $\{0,1,2,3,4\}(m=5)$
  - $\gcd(0,5)=5$
  - $\gcd(1,5)=1$
  - $\gcd(2,5)=1$
  - $\gcd(3,5)=1$
  - $\gcd(4,5)=1$
- $\phi(5)=4$

Euler's Phi Function $\phi(n)$

- **Proposition** $p > 0$ is prime iff $\phi(p) = p - 1$
  - ($\Longrightarrow$) p is prime $\Longrightarrow$ $a \perp p$ for each a with $1 \leq a \leq p - 1$, and there are $p - 1$ of them
  - ($\Leftarrow$) p is not a prime
    - (i) $p = 1$, $\phi(1) = 1 \neq 1 - 1$
    - (ii) p is a composite with a proper divisor d, then $1 < d < p$ and $\gcd(p, d) = d > 1$,hence $\phi(p) \leq p - 2$
- **Proposition** $\phi\left(p^k\right) = p^{k-1}(p - 1)$
  - $S = \{p, 2p, 3p, \ldots, (p^{k-1} - 1) \cdot p, p^{k-1} \cdot p\}$ list all integers between 1 and p^k which are not $\perp p^k$, hence there are $p^k - p^{k-1}$ integers $\perp p^k$

Euler's Phi Function $\phi(n)$

- Phi especially easy for $e_i = 1$, e.g.,

$$m = p \times q \rightarrow \phi(m) = (p-1)\,(q-1)$$

  - $m = 899 = 29 \times 31$:
    - $\phi(899) = (29-1)(31-1) = 28 \times 30 = 840$
- Finding $\phi(m)$ is computationally easy if factorization of m is known
  - otherwise the calculation of $\phi(m)$ becomes computationally infeasible for large numbers

Euler's Phi Function $\phi(n)$

- If canonical factorization of m known: $m = p_1^{e_1} \cdot p_2^{e_2} \cdot \cdots \cdot p_n^{e_n}$
- Calculate Phi according to the relation $\phi(m) = \prod_{i=1}^{n}\left(p_i^{e_i} - p_i^{e_i-1}\right) = \prod_{i=1}^{n}(p-1)p^{e_i-1}$
- Example
  - $\phi(360)= 2^2(2-1) \cdot 3(3-1) \cdot (5-11) = 96$
    - $360 = 2^3 \cdot 3^2 \cdot 5$

Euler's Theorem

- Generalization of Fermat's little theorem to any integer modulus
- Given two relatively prime integers a and m :
$$a^{\phi(m)} \equiv 1 \pmod{m}$$
- Example: $m = 12, a = 5$
  - Calculate Euler's Phi Function
  $\phi(12) = \phi(2^2 \cdot 2^1)(3^1 - 3^0) = (4 - 2)(3 - 1) = 4$
  - Verify Euler's Theorem
  $$5^{\phi(12)} = 5^4 = 25^2 = 625 \equiv 1 \bmod 12$$

Euler's Theorem

- Fermat's little theorem = special case of Euler's Theorem
  - for a prime p: $\phi(p) = (p^1 - p^0) = p - 1$
  - Fermat: $a^{\phi(p)} = a^{p-1} \equiv 1 \pmod{p}$

- Example: Compute $11^{2006} \bmod 21$
  - Solution $\phi(21) = 12$, so $11^{2006} = 11^{12\times 167+2} = \left(11^{\phi(21)}\right)^{167} \times 11^2 \equiv 1^{167} \times 121 \equiv 16 \pmod{21}$

Fermat's Little Theorem

- Given a prime p and an integer a: $\boxed{a^p \equiv a (\bmod p)}$
- Can be rewritten as $a^{p-1} \equiv 1 \pmod p$
  - Find modular inverse, if p is prime.
  - Rewrite to $a \cdot a^{p-2} \equiv 1\ (mod\ p)$
    - Comparing with definition of the modular inverse $a \cdot a^{-1} \equiv 1 \pmod p$

  -> $a^{-1} \equiv a^{p-2} \pmod p$ is the modular inverse modulo a prime p
- Example: $a = 2, p = 7$

$$a^{p-2} = 2^5 = 32 \equiv 4 \bmod 7$$

  - Verify $2 \cdot 4 \equiv 1 \bmod 7$

Chinese Remainder Theorem

- 孫子算經
  - 「令有物，不知其數，三三數之，剩二，五五數之，剩三，七七數之，剩二，問物幾何?」
  - 答曰:「 二十三」解曰:「三三數之剩二，置一百四十，五五數之剩三，置六十三，七七數之剩二，置三十，併之，得二百三十三，以二百一十減之，即得。凡三三數之剩一，則置七十，五五數之剩一，則置二十一，七七數之剩一，則置十五，即得」
- 韓信點兵
  - 傳當年漢高祖巡狩雲夢大澤，欲藉機擒韓信，但不知其兵數，恐有變，故問曰:「卿部下有多少兵卒？」信曰:「敬稟陛下，兵不知其數，三三數之剩二，五五數之剩三，七七數之剩二。」
  - 高組不解，問法於張良。良曰:「兵數無法算，不可數!」
  - 其後雖擒韓信，但仍不知其解。

Chinese Remainder Theorem

- Example $N = 15 = 3 \times 5$
  - Every element $a \in Z_N$ can be represented by its coordinates $(a \bmod 3, a \bmod 5)$
  - This leads to the table:

	0	1	2	3	4
0	0	6	12	3	9
1	10	1	7	13	4
2	5	11	2	8	14

  - All elements in Z_N have different coordinates
  - Given (a_1, a_2) with $0 \le a_1 < 3$ and $0 \le a_2 < 5$, we can reconstruct a

Chinese Remainder Theorem

- Example $N = 24 = 4 \times 6$
  - Every element $a \in Z_N$ can be represented by its coordinates $(a \mod 4, a \mod 6)$
  - This leads to the table:

	0	1	2	3	4	5
0	0/12		8/20		4/16	
1		1/13		9/21		5/17
2	6/18		2/14		10/22	
3		7/19		3/15		11/23

  - a and $a + 12 \pmod{24}$ map to the same coordinates
  - Given (a_1, a_2) with $0 \leq a_1 < 4$ and $0 \leq a_2 < 6$, we can not uniquely reconstruct a

Chinese Remainder Theorem

- Remark
  - If $N = m_1 m_2$ with $m_1 \perp m_2$, computation modulo N can be replaced by modulo m_1 and modulo m_2
    - i.e., $Z_N \cong Z_{m_1} \times Z_{m_2}$ iff $\gcd(m_1, m_2) = 1$
  - If $N = m_1 m_2$, it is very easy to compute the coordinates of $a \in Z_N$, since they are simply $(a \bmod m_1, a \bmod m_2)$
  - However, given the coordinates (a_1, a_2) of a with $0 \le a_1 \le m_1$ and $0 \le a_2 < m_2$, how do we compute the corresponding a?

Chinese Remainder Theorem

- Example Solve the system $x \equiv 4 \pmod 7$ and $x \equiv 3 \pmod 5$
- Solution
  - We have $x = 4 + 7u$ and $x \equiv 3 \pmod 5$ for some $u \in Z$.
  - Substituting in the 2nd equation gives $4 + 7u \equiv 3 \pmod 5$.
  - Therefore, u is given by $2u \equiv 7u \equiv 3 - 4 \equiv 4 \pmod 5$.
  - Hence we compute u as $u \equiv \frac{4}{2} \equiv 2 (\mathrm{mod}\ 5)$.
  - But then $x \equiv 4 + 7u \equiv 4 + 7 \times 2 \equiv 18 \pmod{35}$.

Chinese Remainder Theorem

- **Proposition** The system $x \equiv a_1 \pmod{m_1}$ and $x \equiv a_2 \pmod{m_2}$ has a solution if $m_1 \perp m_2$.
  - Any two solutions are congruent modulo $m_1 m_2$.
- **Proof**
  - If $t = m_1^{-1}(a_2 - a_1) \bmod m_2$,then $x = a_1 + m_1 t$ is such a solution.
  - Assume x_1 and x_2 are two solutions.

  $x_1 \equiv a_1 \equiv x_2 \pmod{m_1}$ and $x_1 \equiv a_2 \equiv x_2 \pmod{\mathrm{m}_2}$

  $\Rightarrow m_1 | (x_1 - x_2)$ and $m_2 | (x_1 - x_2)$

  $\Rightarrow m_1 m_2 | (x_1 - x_2)$ since $m_1 \perp m_2$

Chinese Remainder Theorem

- **Chinese Remainder Theorem** If $m_1, \dots, m_r$ are pairwise relatively prime, then the system $x \equiv a_i \pmod{m_i}$ $1 \le i \le r$ has a unique solution modulo $M = m_1 m_2 \dots m_r$
- **Proof 1** Induction on r (對r做數學歸納法)
- **Proof 2** $x = \sum_{i=1}^{r} a_i \times M_i \times y_i$ is a solution, where $M_i = \frac{M}{m_i}$ and $y_i \equiv M_i^{-1} \pmod{m_i}$
  - $M_i \equiv 0 \left(\mathrm{mod}\, m_j\right)$ for $j \neq i$ and $M_i \times y_i \equiv 1 \pmod{m_i}$

Chinese Remainder Theorem

- **Example** Find the unique x modulo $M = 1001 = 7 \times 11 \times 13$ such that $x \equiv 5 \pmod 7$, $x \equiv 3 \pmod{11}$, and $x \equiv 10 \pmod{13}$

- Solution
  - $M_1 = 143, y_1 = 5$; $M_2 = 91, y_2 = 4$; $M_3 = 77, y_3 = 12$.
  - $x = \sum_{i=1}^{r} a_i \times M_i \times y_i \equiv 5 \times 143 \times 5 + 3 \times 91 \times 4 + 10 \times 77 \times 12 \pmod{1001} \equiv 894 \pmod{1001}$

Chinese Remainder Theorem

- Algorithm Chinese remainder algorithm
  - Input: Vectors $a = (a_1, \ldots, a_r)$ and $m = (m_1, \ldots, m_r)$ with $m_1 \perp m_2$
  - Output: Integer CRA with CRA $\equiv a_i \bmod m_i$
  - Function CRA(a,m,r)

```
if    r=1   Then
            Set CRA = a_1
Else
            Set t = m_{r-1}^{-1}(a_r - a_{r-1}) mod m_r
            Set a_{r-1} = a_{r-1} + t m_{r-1}
            Set m_{r-1} = m_{r-1} m_r
            Set CRA = CRA(a, m, r - 1)
End If
End Function
```

The RSA Cryptosystem

- Martin Hellman and Whitfield Diffie published their landmark public-key paper in 1976
- Ronald Rivest, Adi Shamir and Leonard Adleman proposed the asymmetric RSA cryptosystem in 1977
- Until now, RSA is the most widely use asymmetric cryptosystem although elliptic curve cryptography (ECC) becomes increasingly popular
- RSA is mainly used for two applications
  - Transport of (i.e., symmetric) keys
  - Digital signatures

RSA

- RSA was the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography
  - Patented by MIT in 1983 as U.S. Patent 4,405,829
    - Expired on 21 September 2000
  - Publicly described in 1977 by Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman at MIT
    - Rivest, Shamir, and Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Communications of the ACM, 21(2), pp. 120-126, 1978

Encryption and Decryption

- RSA operations are done over the integer ring Z_n (i.e., arithmetic modulo n), where $n = p \times q$, with p, q being large primes
- Encryption and decryption are simply exponentiations in the ring

Definition

Given the public key $(n, e) = k_{pub}$ and the private key $d = k_{pr}$ we write

$$y = e_{k_{pub}}(x) \equiv x^e \bmod n$$

$$x = d_{k_{pr}}(y) \equiv y^d \bmod n$$

where $x, y \in \mathbf{Z}_n$.

We call $e_{k_{pub}}()$ the encryption and $d_{k_{pr}}()$ the decryption operation.

- In practice x, y, n and d are very long integer numbers (≥ 1024 bits)
- The security of the scheme relies on the fact that it is hard to derive the "private exponent" d given the public-key (n, e)

Key Generation

- Like all asymmetric schemes, RSA has set-up phase during which the private and public keys are computed

Algorithm: RSA Key Generation

Output: public key: $k_{pub} = (n, e)$ and private key $k_{pr} = d$

1. Choose two large primes p, q
2. Compute $n = p \times q$
3. Compute $\Phi(n) = (p - 1)(q - 1)$
4. Select the public exponent $e \in \{1, 2, \ldots, \Phi(n) - 1\}$ such that $\gcd(e, \Phi(n)) = 1$
5. Compute the private key d such that $d \times e \equiv 1 \bmod \Phi(n)$
6. **RETURN** $k_{pub} = (n, e)$, $k_{pr} = d$

- Remarks:
  - Choosing two large, distinct primes p, q is non-trivial
  - $\gcd(e, \Phi(n)) = 1$ ensures that e has an inverse and, thus, that there is always a private key d

Example: RSA with small numbers

ALICE

Message $x = 4$

BOB

1. Choose $p = 3$ and $q = 11$
2. Compute $n = p \times q = 33$
3. $\Phi(n) = (3 - 1)(11 - 1) = 20$
4. Choose $e = 3$
5. $d \equiv e^{-1} \equiv 7 \bmod 20$

$\xleftarrow{K_{pub} = (33,\ 3)}$

$y = x^e \equiv 4^3 \equiv 31 \bmod 33$

$\xrightarrow{y = 31}$

$y^d = 31^7 \equiv 4 = x \bmod 33$

Proof of Decryption

- There exists $k \in Z$ such that $ed = 1 + k\phi(n)$
  - If $\gcd(x, p) = 1$
    - We have $x^{p-1} \equiv 1\ (mod\ p)$ by Fermat's Little Theorem
    - Taking $k\,(q-1)$ -th power and multiplying with x yields $x^{1+k(p-1)(q-1)} \equiv x \pmod{p}$ $\quad (*)$
  - if $\gcd(x, p) = p$, then $x \equiv 0 \pmod{p}$ and $(*)$ is valid again
- Hence $x^{ed} \equiv x \pmod{p}$ in both cases, and by a similar argument we have $x^{ed} \equiv x \pmod{q}$
- Since p and q are distinct primes, the CRT leads to $y^d \equiv (x^e)^d = x^{ed} = x^{1+k(p-1)(q-1)}$ $\equiv x \pmod{N}$