L07: Cryptography

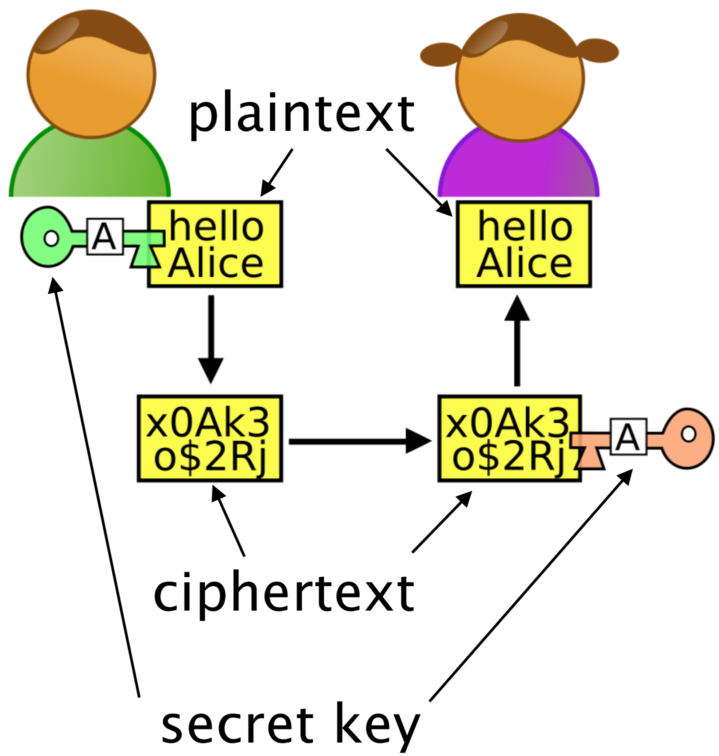
- Cryptography is the study of methods for sending and receiving secret messages through insecure channels

- Outline:
  - **Secret Key Cryptography**
  - Key Exchange
  - Public Key Cryptography and RSA

- Reading: Rosen 4.6

Secret Key Cryptography

- In secret key cryptography, the sender (Bob) and the receiver (Alice) first agree on a common secret key in advance



Caesar Cipher (Shift Cypher)

- Encryption
  - The secret key k is a number from $\mathbf{Z}_{26}$
  - Replace each letter by an integer from $\mathbf{Z}_{26}$
  - The encryption function is $f(p) = (p + k) \bmod 26$. It replaces each integer p by $f(p)$.
  - Replace each integer by the corresponding letter
- Decryption
  - Just replace $f(p)$ with $f^{-1}(p) = (p - k) \bmod 26$ in the procedure above.

Caesar Cipher: Example

- **Example**
  Encrypt the message “MEET YOU IN THE PARK” using $k = 3$
- **Solution**
  - Replace letters by numbers:
    $12\ 4\ 4\ 19 \quad 24\ 14\ 20 \quad 8\ 13 \quad 19\ 7\ 4 \quad 15\ 0\ 17\ 10.$
  - Replace each of these numbers p by $f(p)$:
    $15\ 7\ 7\ 22 \quad 1\ 17\ 23 \quad 11\ 16 \quad 22\ 10\ 7 \quad 18\ 3\ 20\ 13.$
  - Translating the numbers back to letters
    “PHHW BRX LQ WKH SDUN.”

Affine Ciphers

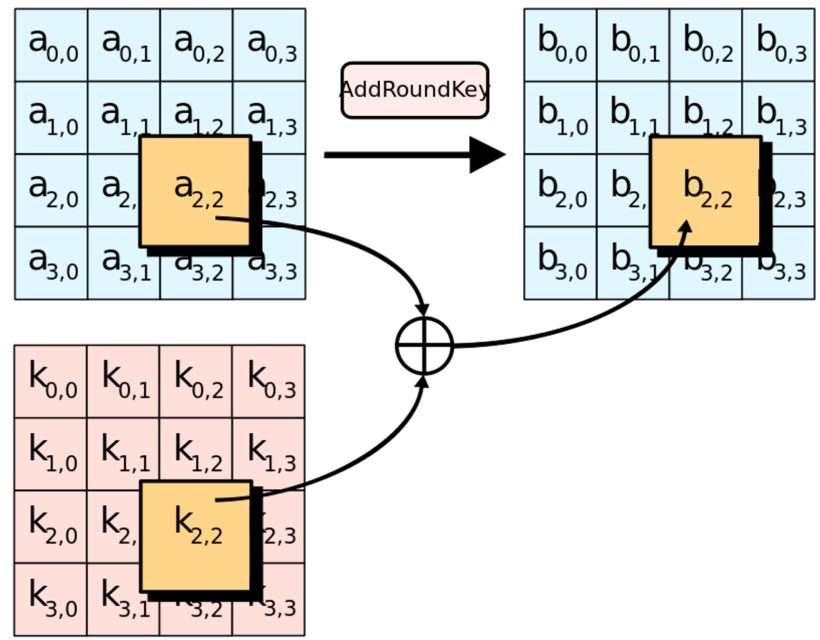
- The shift cipher is easy to break:
  - Just try all 26 possible keys!
- Affine ciphers make it (a little bit) safer by using both additions and multiplications
- Use the function $f(p) = (ap + b) \bmod 26$
  - The (a, b) pair is the secret key
  - Now there are $26^2 = 676$ possible secret keys
- However, suppose $a = 13, b = 1$
  - $f(1) = f(3) = f(5) = f(7) = \cdots = 14$
  - If we receive a 14, which number does it decrypt to?
- How to fix?
  - Choose a such that $\gcd(a, 26) = 1$, e.g., $a = 7$
  - Then $ax + b \equiv y \pmod{p}$ has a unique solution

Block Ciphers

- Each character is a number between 0 and 255
  - A byte = 8 bits
- Partition the message into blocks of k characters
  - Treat each block as a big number of $8k$ bits
  - Use arithmetic modulo 2^{8k}
- Example
  - Choose $k = 10$
  - Encryption: $f(x) = (ax + b) \bmod 2^{80}$
  - Decryption: $f^{-1}(y) = a^{-1}(y - b) \bmod 2^{80}$
  - Now there are $\frac{2^{80}}{2} \times 2^{80} = 2^{159}$ different keys
- There are libraries on arbitrary-precision arithmetic
- But affine ciphers are subject to known-plaintext attacks!

Advanced Encryption Standard (AES)

- Used in Transport Layer Security (TLS)
  - Previously known as Secure Sockets Layer (SSL)
  - Provides security for https, email, etc.
- A block cipher
  - Block size 128 bits
  - Key lengths: 128, 192, 256 bits
- Complicated operations that make it very difficult to break

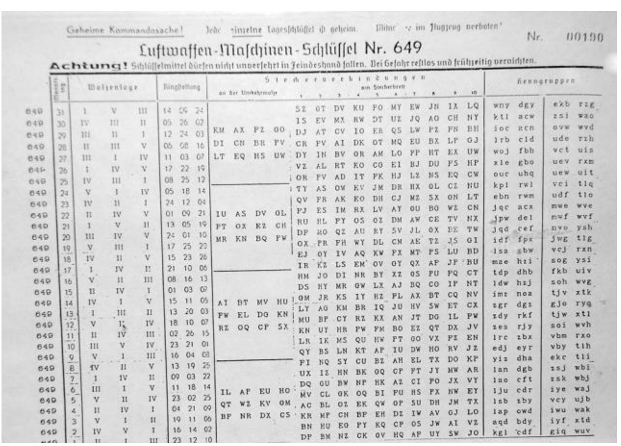


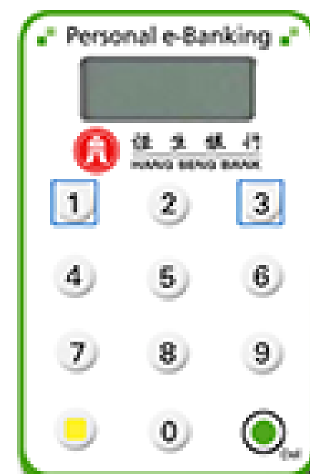
Outline

- Secret Key Cryptography
- **Key Exchange**
- Public Key Cryptography and RSA

Problems with secret-key cryptography

- How to send the secret key?
  - Keys had to be transmitted in physical form in World War II
- Electronic keys in modern days, but still have to be delivered physically for maximum security





New Security Device

The Key Exchange Puzzle

- Alice wants to send a valuable item to Bob, but the postman cannot be trusted
  - Alice can put an (unbreakable) lock on the box, but Bob cannot open it without the key
- Solution
  - Alice puts her lock on the box, and sends its to Bob.
  - Bob, after receiving the box, puts his lock on the box as well, and returns to Alice.
  - Alice, after receiving the box, takes off her lock, and sends it back to Bob.
  - Bob takes off his lock and opens the box.
- "Locks" in cryptography are one-way functions: easy to compute $f(x)$ from x, but given y, it's hard to find an x such that $f(x) = y$

Modular exponentiation

Lemma:

For any $a \in \mathbf{Z}_n$ and any nonnegative integers i, j

- $a^i \bmod n = \big(\underbrace{(a \bmod n)(a \bmod n)\dots(a \bmod n)}_{i \text{ factors}}\big) \bmod n$
- $a^{i+j} \bmod n = \left(\left(a^i \bmod n\right)\left(a^j \bmod n\right)\right) \bmod n$
- $a^{ij} \bmod n = \left(\left(a^i \bmod n\right)^j\right) \bmod n$

Proof: Directly from theorem in L05

Examples

$$13^4 \equiv 6^4 \pmod 7$$

$$3^{2+4} \equiv 3^2 \cdot 3^4 \equiv 2 \cdot 4 \equiv 1 \pmod 7$$

$$3^{4(2)} \equiv (3^4)^2 \equiv 4^2 \equiv 2 \pmod 7$$

A One-Way Function: Modular Exponentiation and Logarithm

- How to compute
$$a^n \bmod m$$
efficiently for large n?
- Repeated squaring method
  - Compute
    $a^2 \bmod m$
    $a^{2^2} \bmod m = a^4 \bmod m = (a^2 \bmod m)^2 \bmod m$
    $a^{2^3} \bmod m = a^8 \bmod m = (a^4 \bmod m)^2 \bmod m$
    …
  - Write n in binary $n = (b_k \dots b_1 b_0)_2$
  - $a^n \equiv a^{b_0 \cdot 1} \cdot a^{b_1 \cdot 2} \cdot a^{b_2 \cdot 2^2} \cdots \pmod m$
- Example: $n = 50 = (110010)_2$
  - $a^{50} \equiv a^{2^1} a^{2^4} a^{2^5} \pmod m$

A Hard Problem: Discrete Logarithm

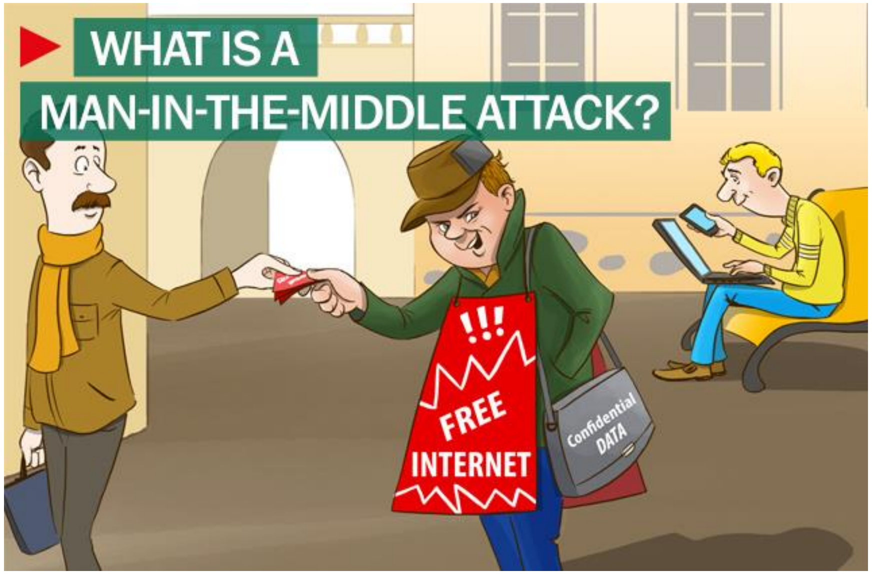
- Discrete logarithm is one such problem
  - Inverse of modular exponentiation
  - Given a prime p (potentially very large) and $r, a \in \mathbf{Z}_p$, find x such that
$$a^x \equiv r \pmod{p}$$
- Modular exponential and discrete logarithm is believed to be a one-way function
  - Yes, if you can solve discrete logarithm, you can break current crypto systems
- In 2015, it was reported that for 512-bit primes, the problem can be solved with a few thousands of CPUs in a week
  - Estimated cost to break 1024 bits: US$100 million.

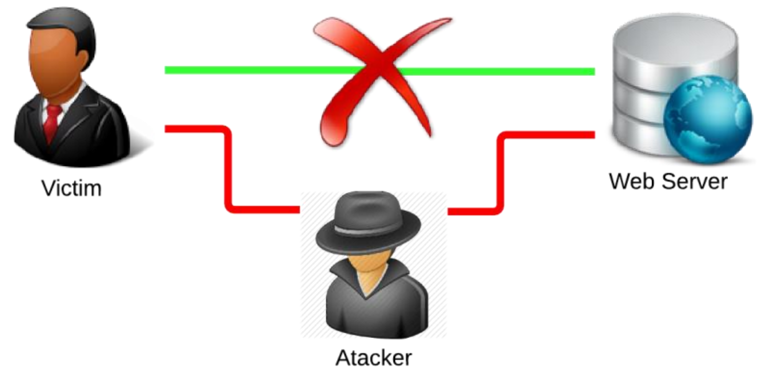
Diffe-Hellman Key Exchange

- Fix p and a
  - E.g., hardcoded in the TLS library
- The protocol
  1) Alice chooses a secret integer k_1 and Bob chooses k_2
  2) Alice sends $a^{k_1} \bmod p$ to Bob.
     Secure to eavesdropping: Even this value is known to attackers, they cannot compute k_1
  3) Bob computes $(a^{k_1})^{k_2} \bmod p$.
  4) Bob sends $a^{k_2} \bmod p$ to Alice.
  5) Alice computes $(a^{k_2})^{k_1} \bmod p$.
- The shared key is

$$\left(a^{k_1}\right)^{k_2} \bmod p = \left(a^{k_2}\right)^{k_1} \bmod p = a^{k_1 k_2} \bmod p$$

Man-in-the-middle Attack





- If attacker intercepts all traffic between two parties
- Diffe-Hellman protocol can be compromised
- Attacker
  - communicates with Alice pretending as Bob
  - communicates with Bob pretending as Alice

Outline

- Secret Key Cryptography
- Key Exchange
- **Public Key Cryptography and RSA**

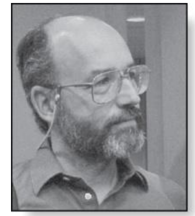
Public Key Cryptography

- All previous ciphers need a common secret key
- Encryption and decryption are symmetric
- The key has to be
  - communicated physically in secret
  - using the DH protocol (secure to eavesdropping but not man-in-the-middle)
- Public key cryptography
  - Encryption and decryption are asymmetric
  - Everyone has
    - a public key: shared with everyone else
    - a private key: kept secret

The RSA Cryptosystem

Ronald Rivest
(Born 1948)

Adi Shamir
(Born 1952)



Leonard
Adelman
(Born 1945)

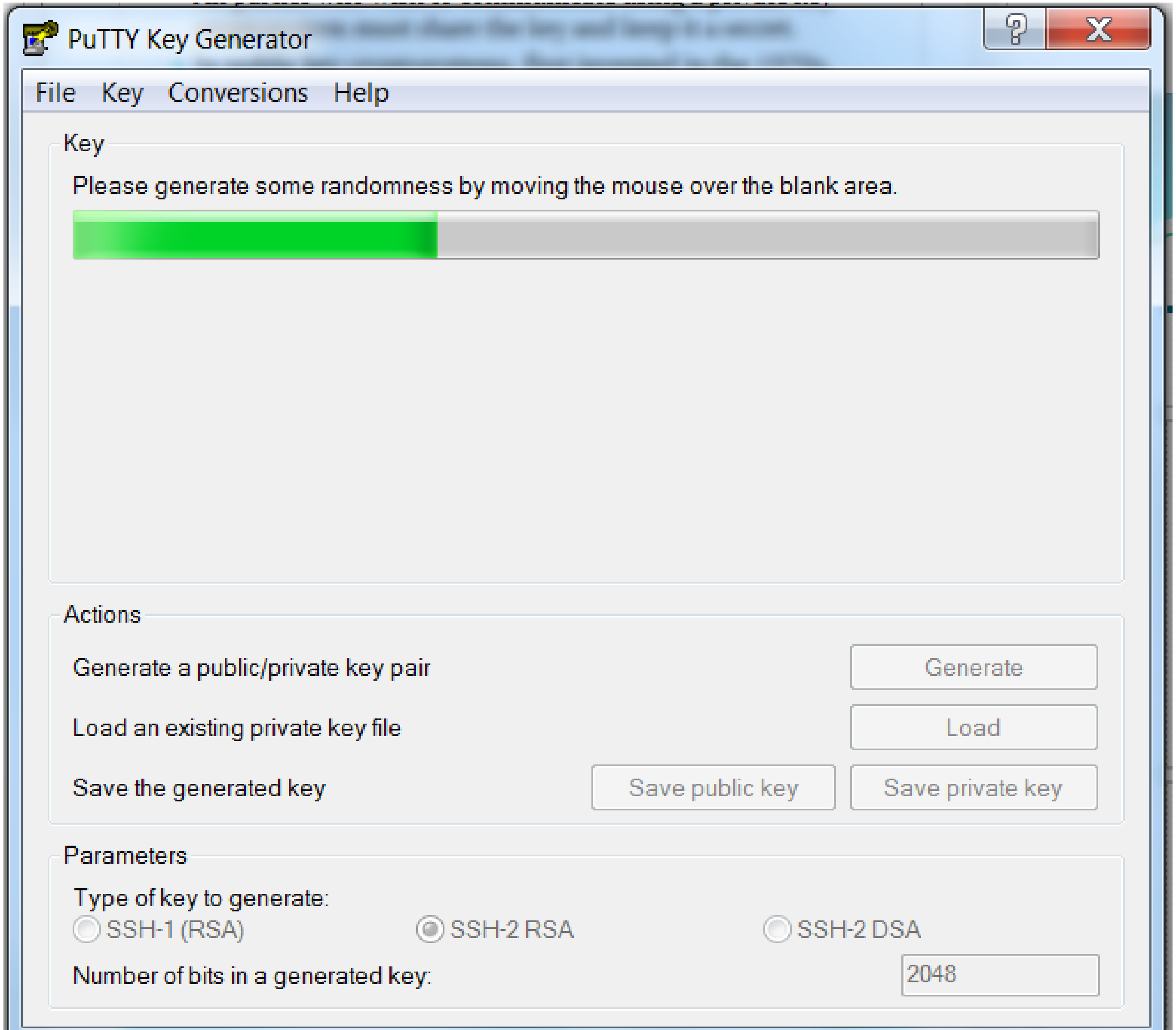
Clifford Cocks
(Born 1950)

- RSA was introduced in 1976 by RSA.
- In fact, Clifford Cocks, working secretly for the UK government, discovered it 3 years earlier.
    - Made known to public in 1997.

Another one-way function: Multiplication and factoring

- Let $n = pq$, where p and q are large primes (e.g. 1024 bits or longer)
- The factoring problem: Given n, find p and q
- On the other hand, it is known how to find random large primes efficiently
- Public key in RSA: n and e, such that e is relatively prime to $(p-1)(q-1)$
  - In practice: Pick e randomly and check
- Private key in RSA: p and q
  - Actually, only d (defined later) is needed
- Everyone uses a different set of keys

Key Generator



This is My Public Key

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "rsa-key-20161118"
AAAAB3NzaC1yc2EAAAABJQAAAQEAkrwKeUwwz0jThhh2NSS8EJhEDl8VDzyCh8Rw
y2NJ6nHymOwyCWicUhjiY7wPOMljt6XFlmnAHACz0JhAg/hAHHYF8bdJJZ4slZrM
kNRQ0ZUDVDvacygKjeXDjneCvFrS+78ancE7gGGkZMaxWf4NsQVCoX3wRMuk6cHs
mrwGINYWGCHshjLAnzYwPvLegvlPszh1zhgzziMGNU08wf/q8WOrZmrtHB4epWhI
aSEjNIZmDlbkyy8SwW4y/7GjVKNLpnObUHh7qqBDnmWd5HnMWAEuHxbAhMXqIWIS
UKe8cwnFBWHpHCXMCyoCI1uJNhfjtj2hq7QKkejH/jCJ5U26pQ==
---- END SSH2 PUBLIC KEY ----
```

RSA Encryption

- Let $x < n$ be the message to be encrypted
- Alice encrypts it as
$$C = x^e \bmod n$$
  - (n, e) is Bob's public key
  - Sends C to Bob
  - C may be eavesdropped
- Security
  - Exponentiation can be computed efficiently
    - Proportional to the length of the key
  - Computing x from C, n, e is believed to be difficult
    - Known as the RSA problem

RSA Decryption

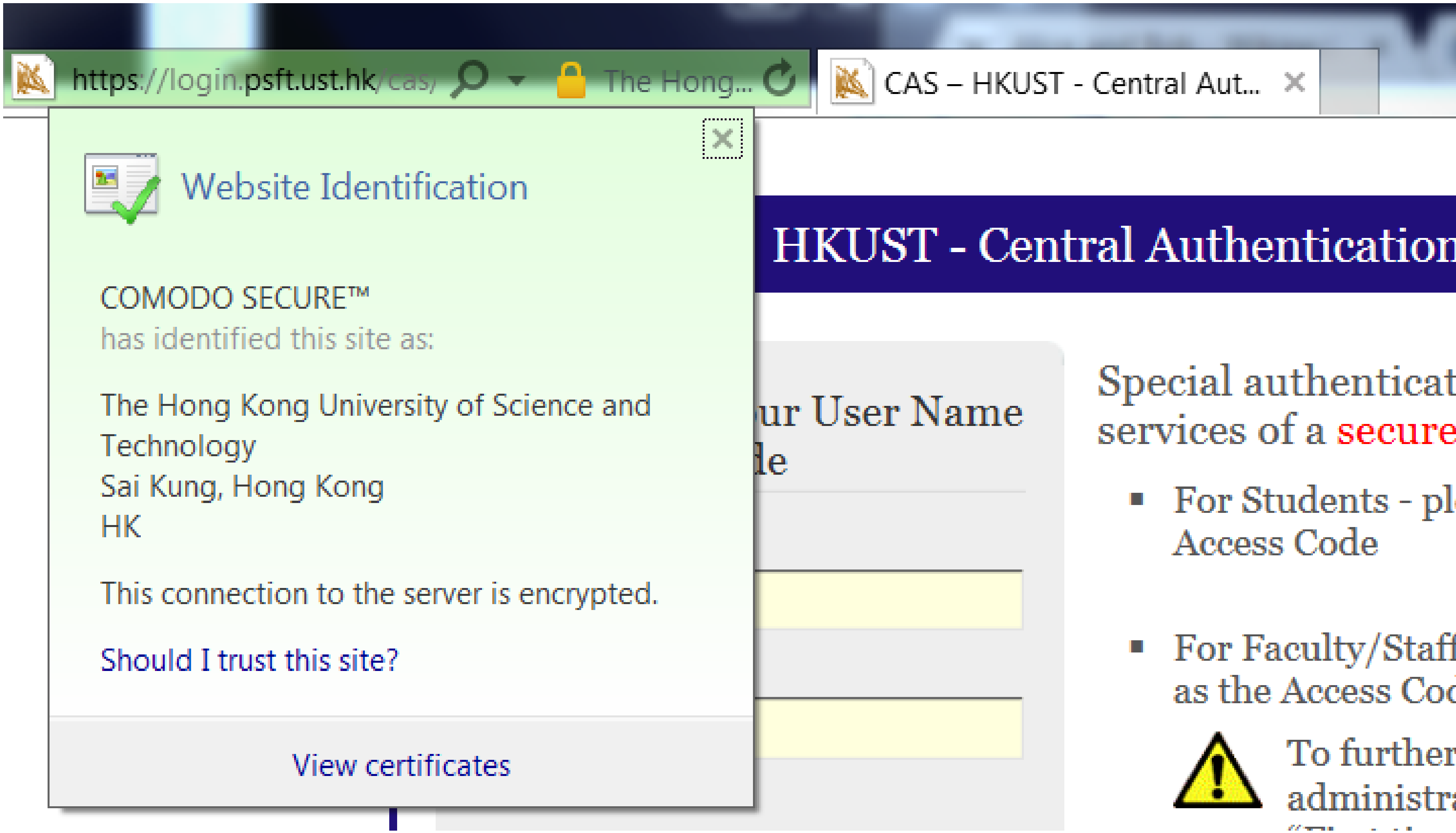
- Bob receives C
- Bob decrypts x from C using his private key (p, q)
  - Find d, the inverse of e modulo $(p-1)(q-1)$, i.e.,
    $$de \equiv 1 \pmod{(p-1)(q-1)}$$
  - Compute
    $$C^d \bmod n$$
- Will show later that $C^d \equiv (x^e)^d \equiv x^{de} \equiv x \pmod{n}$
- Security:
  - It's hard to find d without knowing (p, q)

RSA in Use

- Sending secrete keys (e.g., for use in AES)
  - Alice encodes the secret key using Bob's public key
  - Bob decodes the secret key using his private key
- Digital signatures (authentication)
  - Alice encodes her message using her private key d and her public key n
    - Computes $C = x^d \bmod n$
    - Sends (C, x) to Bob
  - Bob decodes the message using Alice's public key (n, e)
    - Computes $C^e \bmod n = x^{de} \bmod n$
    - He will know the message indeed came from Alice if $C^e \bmod n = x$
  - The scheme above transmits x in plaintext. How to also keep it secret? (See textbook for the answer.)

RSA in Use

- How to prevent man-in-the-middle attacks?
- How to make sure that Alice's public key indeed belongs Alice?
- Certificate authority (CA)
  - A small number of trusted third parties: Comodo, Symantex, GoDaddy, GlobalSign, …
- How to make sure that a CA's public key indeed belongs to that CA?
- Built into Internet browsers
- How can I trust my browser and the CAs?
- Well, you have to …


https://login.psft.ust.hk/cas/
The Hong...
CAS – HKUST - Central Aut...
Website Identification
COMODO SECURE™
has identified this site as:
The Hong Kong University of Science and Technology
Sai Kung, Hong Kong
HK
This connection to the server is encrypted.
Should I trust this site?
View certificates
HKUST - Central Authentication
ur User Name
le
Special authenticati
services of a securec
For Students - ple
Access Code
For Faculty/Staff
as the Access Cod
To further
administra

RSA: Correctness

- **Proof plan**

  We want to show

$$C^d = x^{de} \equiv x \pmod{n}.$$

  Step 1: Show that

$$x^{de} \equiv x \pmod{p}$$
$$x^{de} \equiv x \pmod{q}$$

  Step 2: Show that

$$x^{de} \equiv x \pmod{pq}$$

Fermat Little Theorem

- **Lemma:** Let p be a prime number. For any non-zero $a \in \mathbf{Z}_p$, $1 \cdot_p a,\ \ 2 \cdot_p a,\ \ \ldots,\ \ \ (p-1) \cdot_p a$ are a permutation of the set $\{1, 2, \ldots, p-1\}$
- **Proof sketch**
  - None of them is 0
  - If $x \cdot a \equiv y \cdot a \pmod{p}$, then $x \cdot a \cdot a^{-1} \equiv y \cdot a \cdot a^{-1}, x \equiv y$
- **Example**

$\cdot_7$	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

Fermat Little Theorem

- **Theorem**

  Let p be a prime number. Then for any non-zero $a \in \mathbf{Z}_p$

$$a^{p-1} \equiv 1 \pmod{p}$$

- **Proof**

  From the lemma,

$$x = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (p-1) \equiv \left(1 \cdot_p a\right)\left(2 \cdot_p a\right) \cdots \left((p-1) \cdot_p a\right) \pmod{p}$$

$$\equiv 1 \cdot 2 \cdot 3 \cdot \ldots \cdot (p-1) \cdot a^{p-1} \pmod{p}$$

x^{-1} exists in $\mathbf{Z}_p$, multiplying both sides by x^{-1} gives:

$$1 \equiv a^{p-1} \pmod{p}$$

Fermat Little Theorem

- **Example:** $\mathbf{Z}_7$

a	a^0	a^1	a^2	a^3	a^4	a^5	a^6
1	1	1	1	1	1	1	1
2	1	2	4	1	2	4	1
3	1	3	2	6	4	5	1
4	1	4	2	1	4	2	1
5	1	5	4	6	2	3	1
6	1	6	1	6	1	6	1

- The sequence in each row cycles

Fermat's Little Theorem

Pierre de Fermat
(1601-1665)

- **Corollary**
  If p is prime and a is an integer not divisible by p, then
  $$a^{p-1} \equiv 1 \pmod{p}$$
- **Examples:**
  $$9^6 \equiv 2^6 \equiv 1 \pmod{7} \qquad 14^6 \equiv 0 \pmod{7}$$
- Useful in computing the remainders of large powers
- **Example:**
  Find 7^{222} mod 11.
  By the theorem, we know that $7^{10} \equiv 1 \pmod{11}$, and so $(7^{10})^k \equiv 1 \pmod{11}$, for any positive integer k.
  Therefore,
  $$7^{222} = 7^{22 \cdot 10 + 2} = (7^{10})^{22} \cdot 7^2 \equiv 1^{22} \cdot 49 \equiv 5 \pmod{11}$$

RSA Correctness Step 1

- **Proof**

  We know d is the inverse of e modulo $(p-1)(q-1)$, so
  $$de = 1 + k(p-1)(q-1).$$
  It follows that
  $$\begin{aligned} C^d &\equiv (x^e)^d \\ &\equiv x^{de} \\ &\equiv x^{1+k(p-1)(q-1)} \pmod{p} \end{aligned}$$
  Case 1: x is not a multiple of p.
  Applying Fermat's Little Theorem:
  $$x^{k(p-1)(q-1)} \equiv 1 \pmod{p},$$
  so
  $$x^{1+k(p-1)(q-1)} \equiv x \pmod{p},$$

RSA Correctness Step 1 (cnt'd)

- Case 2: x is a multiple of p.
  Then
  $$x \equiv 0 \pmod{p}$$
  Thus in this case, we have
  $$x^{1+k(p-1)(q-1)} \equiv x \equiv 0 \pmod{p}.$$
- The proof for $x^{de} \equiv x \pmod{q}$ is symmetric.

RSA Correctness Step 2

- Proof of Step 2:
  - We already have
$$x^{de} \equiv x \pmod{p}$$
$$x^{de} \equiv x \pmod{q}$$
  - Because $\gcd(p, q) = 1$, by the Chinese Remainder Theorem (treating x^{de} as the unknown and x as given), x^{de} has a unique solution in $\boldsymbol{Z}_{pq}$:
$$x^{de} \equiv x \pmod{pq}$$