Network Security Contd

Internet Technologies
COMP90007

RSA: An Asymmetric Key Algorithm

- **RSA - Rivest, Shamir, Adleman**
- Famous and robust algorithm
- Key generation:
  - Choose two large primes, p and q
  - Compute n = p × q and z = (p − 1) × (q − 1).
  - Choose ***d*** to be relatively prime to ***z***, i.e., no common factors
  - Find ***e*** such that
    - **(d x e) mod z = 1**
  - Public key is (e, n), and private key is (d, n)
- Encryption:
  - Cipher = Plain^{e} (mod n)
- Decryption:
  - Plain = Cipher^{d} (mod n)

RSA Example

- Let p=3, q=11: then ***z*** is (3 – 1) x (11 – 1) = 20
- What is a potential d?
- If d = 7 then they, z, 20, has no common factors
- What is an e?
- If e = 3, then (d x e) is 1 in mod z
- What are the two key tuples then?
- Enc: 3, 33 Dec: 7, 33 (as n=3 x 11=33 and d=7 and e=3)

S is the 19th character in the alphabet…

Plaintext (P)		Ciphertext (C)		After decryption		
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Encryption: $C = P^3 \bmod 33$

Decryption: $P = C^7 \bmod 33$

Another Use of Cryptography: Digital Signatures

- Cryptographic approaches can also be used to ensure authenticity and allow for non-repudiation
- Requirements
  - Receiver can ***verify the claimed identity of the sender***
  - ***Sender cannot deny she created*** contents of the message
  - ***Receiver cannot have derived the message themselves***

Digital Signatures

- Approaches
  - Using symmetric keys via an intermediary
    - You need a BIG BROTHER to do all the messaging, not preferred..!
  - Using **public keys** as individuals…

Using Public Keys

- Sender Alice uses ***private key on P***
- ***Receiver Bob uses her public key to undo and get P***
- RSA can do this as well, as ***E(D(P)) = P in RSA***

- Alice cannot deny signing as she only knows her private key

Signatures with Message Digests

- Why E(D(P)) when P is large for just signing; if contents is not secret
- Basic concept of a **message digest is to use a one-way hash function** for an arbitrary length of plaintext, so that it becomes a ***"unique" small fixed-length bit string***
- Thus **no need to deal with huge message text and encryption just for authentication** purposes and hashing is generally fast!
- A message digest (MD) has four important properties:
  - 1 Given P, it is easy to compute MD(P)
  - 2 Given MD(P) it is effectively impossible to find P
  - 3 Given P, no one can find P' such that MD(P') = MD(P)
  - 4 A change in even a single bit of input produces a very different output

Famous Message Digest Algorithms

- MD5
- SHA-1
- Outputs
  - Given ”this is a test“ (text could have been longer)
  - MD5:

    e19c1283c925b3206685522acfe3e6
  - SHA-1:

    6476df3aac780622368173fe6e768a2edc3932c8

Public Key Management

- There is **specific PK infrastructure** to avoid compromising the security of PK's **during the initial distribution process**
- Certification Authority (CA)
  - A trusted intermediary who uses non-electronic identification to identify users prior to certifying keys and issuing certificates
- X.509 became the standard for certificates
  - An international format for certificate expression
- Then: PKI (Public Key Infrastructure) is a
  - **Hierarchically structured Certificate Authorities** allowing for the establishment of a chain of trust and thus certification paths
  - *Verisign* was a famous company in this domain for example

Certificate Issuing

- A Certificate authority (CA) give the following certificate:

I hereby certify that the public key
19836A8B03030CF83737E3837837FC3s87092827262643FFA82710382828282A
belongs to
Robert John Smith
12345 University Avenue
Berkeley, CA 94702
Birthday: July 4, 1958
Email: bob@superdupernet.com

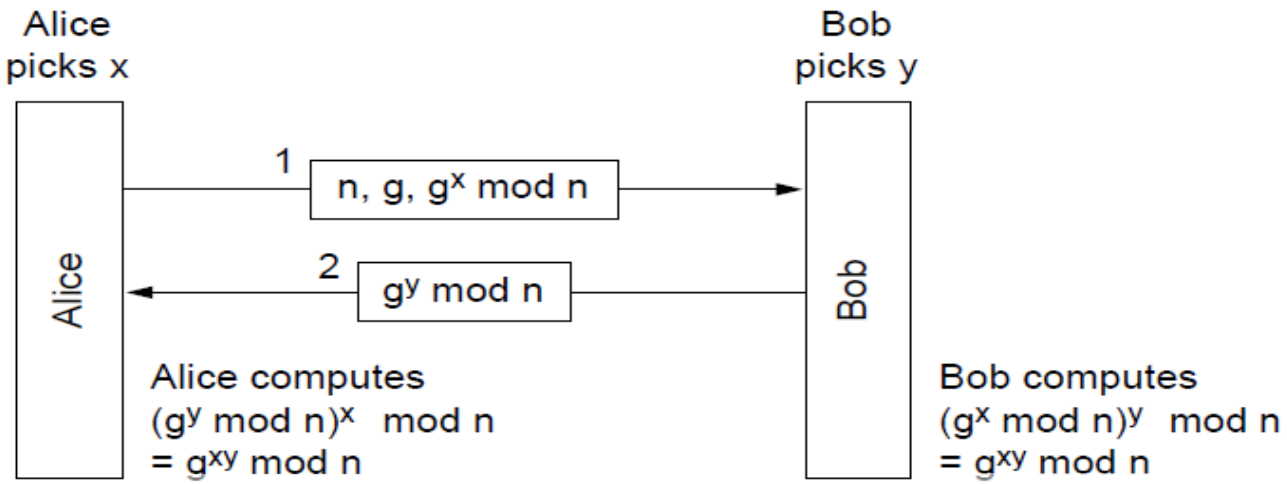
SHA-1 hash of the above certificate signed with the CA's private key

Authentication

- **Authentication is a primary tenet** of network security
- However, **authentication process itself needs to be secure** also
- A fundamental principle: **minimise the use of permanent keys in establishment of secure connections** (the less packets are exchanged using such keys, the less exposure to potential attackers)
- Four methods in common use:
  - Shared keys
  - Key distribution
  - Kerberos
  - Public keys

Authentication Based on a Shared Secret Key

- How to create a key with Diffie-Hellman key exchange:



Shared secret

Shared secret

Is there a way to break this: YES

Still open to man-in-the-middle attack!