CS408
Cryptography & Internet Security

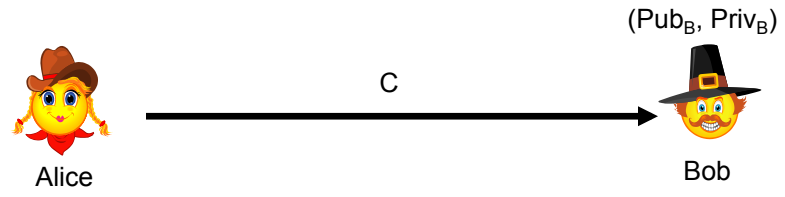
Lecture 15:

Public-key Cryptography,
RSA

Public key encryption

Each entity has:

- a public key (Pub), which is made public
- a private key (Priv), which is kept secret



generate ciphertext
$C = E_{PubB}(M)$

retrieve plaintext
$M = D_{PrivB}(C)$

Public key encryption

- Entities don't need to establish a secret key or a trust relationship ahead of time
- A public key encryption scheme is a collection of three algorithms (G, E, D)
  - Key generation algorithm G: generates a pair of keys (Pub, Priv)
  - Encryption algorithm E: C = $E_{Pub}(M)$
  - Decryption algorithm D: M = $E_{Priv}(C)$
- The following should always hold true:
  - $D_{Priv}(E_{Pub}(M)) = M$

- It is infeasible to derive the private key from the public key
- The public keys may be made publicly available, e.g., in a publicly available directory
- Many can encrypt, only one can decrypt
- Provides: confidentiality
- Does not provide: authentication, non-repudiation
- It is a keyed cryptographic primitive
- Example: RSA encryption, El-Gamal encryption

Public-key Cryptography

- Public-key cryptography (a.k.a. asymmetric-key cryptography)
  - encryption key different from decryption key
  - cannot derive decryption key from encryption key
  - higher cost than symmetric cryptography
  - simplifies key distribution

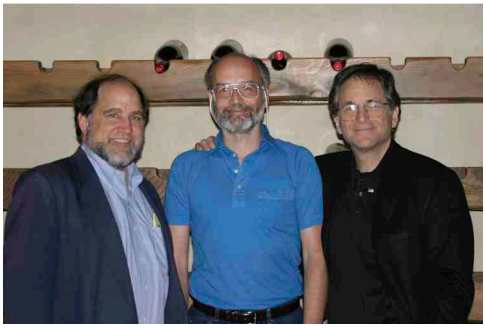
  - Came into existence in 1976
  - Analogy with a mailbox

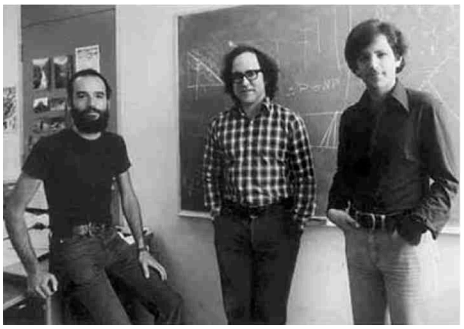
Miscellaneous: Turing Award

- The Turing Award is the equivalent of the Nobel Prize for Computer Science
  - $1,000,000 prize (used to be $250K until 2014)
  - Named after Alan Turing (British mathematician who was part of the team which cracked the Enigma machine; also had many contributions in theoretical Computer Science and a significant role in the creation of the modern computer)

Miscellaneous: Turing Award

- Past winners:
  Rivest, Shamir, Adleman (in 2002) for "their ingenious contribution to making public-key cryptography useful in practice"





Miscellaneous: Turing Award

- Past winners:

  Shafi Goldwasser and Silvio Micali (in 2012):

  for "transformative work that laid the complexity-theoretic foundations for the science of cryptography, and in the process pioneered new methods for efficient verification of mathematical proofs in complexity theory."
  - Press release:

    http://www.acm.org/press-room/awards/turing-award-12
  - "their work helped to establish the tone and character of modern cryptographic research."

Public-key Cryptography

- Alice has pair of keys Pub_A, $Priv_A$
- $\mathbf{E}_{PubA}$ must be a one-way function:

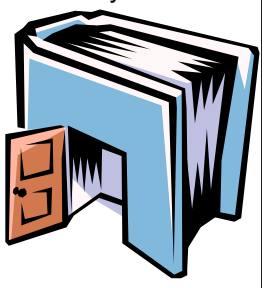
  knowing C= $\mathbf{E}_{PubA}$[M], it should be infeasible to find M
- However, $\mathbf{E}_{PubA}$ should not be one-way from Alice's perspective. The function $\mathbf{E}_{PubA}$ must have a trapdoor such that knowledge of the trapdoor enables Alice to invert it (you can think of the private key $Priv_A$ as the trapdoor)

One-way Trapdoor Functions

Definition:

A function f: {0,1}* → {0,1}* is a trapdoor one-way function iff f(x) is a one-way function; however, given some extra information it becomes feasible to compute f^{-1}: given y, find x s.t. y = f(x)

- Example: $f(x) = x^3 \bmod n$, n = pq (based on the integer factorization problem)
  - If factorization of n is unknown, then f is a one-way function
  - However, if one knows p and q s.t. n=pq, then it is easy to invert the function



RSA Algorithm

- Invented in **1978** by Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman
  - Published as R. L. Rivest, A. Shamir, L. Adleman, "*On Digital Signatures and Public Key Cryptosystems*", Communications of the ACM, vol. 21 no. 2, pp120-126, Feb 1978
- Security relies on the difficulty of factoring large composite numbers

Z^*_n (multiplicative group of integers mod n)

- Let n>1 be an integer and let $Z^*_n = \{a \in Z_n \mid \gcd(a,n)=1\}$. $(Z^*_n, \bullet)$ is a group, where $\bullet$ is multiplication modulo n. $(Z^*_n, \bullet)$ is called the multiplicative group of Z_n.

- Let p and q be two large primes
- Denote their product n=pq
- $Z^*_n = Z^*_{pq}$ contains all integers in the range [1, n-1] that are relatively prime to both p and q
- The size of Z^*_n is $\phi(pq) = (p-1)(q-1)$
- For every $x \in Z_n^*$, $x^{(p-1)(q-1)} \equiv 1 \bmod n$

RSA Public Key Cryptosystem

Key generation:

- Select 2 large prime numbers of about the same size, p and q
- Compute n = pq, and $\phi(n) = (q-1)(p-1)$
- Select a random integer e, $1 < e < \phi(n)$, s.t. $\gcd(e, \phi(n)) = 1$
- Compute d, $1< d< \phi(n)$ s.t. $ed \equiv 1 \bmod \phi(n)$

Public key: (n, e)
Private key: d

Note: p and q must remain secret

RSA Public Key Cryptosystem

Encryption

- Obtain the recipient's public key (n,e)
- Represent the message as an integer M , $0 \le M < n$
- Compute $C = M^e \bmod n$
- Send ciphertext C to recipient

Decryption

- Given a ciphertext C, use private key d to recover M:
  $M = C^d \bmod n$

RSA Example

- p = 11, q = 7, n = 77, $\phi(n) = 60$
- d = 13, e = 37 (ed = 481; ed mod 60 = 1)

- Let M = 15. Then $C \equiv M^e \bmod n$
  $C \equiv 15^{37} \pmod{77} = 71$

- $M \equiv C^d \bmod n$
  $M \equiv 71^{13} \pmod{77} = 15$

Why does RSA decryption work?

- Need to show that $(M^e)^d \equiv M \bmod n$, where $n = pq$
- Since $ed \equiv 1 \bmod \phi(n)$, there exists k s.t. $ed = 1 + k\,\phi(n)$
- If we can show that $M^{ed} \equiv M \bmod p$ and $M^{ed} \equiv M \bmod q$, then $M^{ed} \equiv M \bmod n$ (since gcd(p,q)=1)

- Show that $M^{ed} \equiv M \bmod p$:
  - If $M \equiv 0 \bmod p$, then certainly $M^{ed} \equiv M \bmod p$
  - If $M \neq 0 \bmod p$, then by Fermat's Little Theorem: $M^{p-1} \equiv 1 \bmod p$
    Thus: $M^{ed} \equiv M^{1 + k\,\phi(n)} \equiv M^{1 + k(p-1)(q-1)} \equiv M\,(M^{p-1})^{k(q-1)} \equiv M\,1^{k(q-1)} \equiv M \bmod p$

- Similarly, we can show that $M^{ed} \equiv M \bmod q$

RSA Implementation

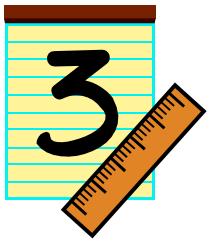
n, p, q

- The security of RSA depends on how large n is, which is often measured in the number of bits for n. Current NIST recommendation is at least 2048 bits for n.
- p and q should have the same bit length, so for 2048-bit RSA, p and q should each be about 1024 bits.
- p - q should not be small

RSA Implementation

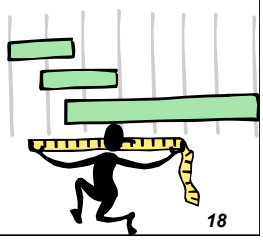
Value for e ?

- e is usually chosen to be 3 or $2^{16} + 1 = 65537$
- In order to speed up the encryption
  - the smaller the number of 1 bits, the better
  - why?



RSA on Long Messages

- RSA requires that the message M is at most n-1 where n is the RSA modulus.
- What about longer messages?
  - They are broken into blocks with value at most n-1.
  - Smaller messages are padded.
  - CBC is used to prevent attacks regarding the blocks.
- In practice RSA is used to encrypt symmetric keys, so the message is not very long.
  - To encrypt a long message M for Bob, Alice generates a random symmetric key K, then computes: $AES_K(M)$ and $RSA_{Bob\_pub\_key}(K)$, and sends them to Bob.



Public Key Cryptography

- Advantages over symmetric key crypto
  - Key management
    - Key establishment: does not require secure channel to transmit secret keys
    - Key distribution: does not require $O(n^2)$ keys to be managed to communicate with n entities
- Disadvantages over symmetric key crypto
  - Slower (orders of magnitude)
- Is not meant to completely replace symmetric key cryptography, but to supplement it
  - $E1_{Pub}(K)$, $E2_K(M)$

    (where E1 is symmetric-key encryption and E2 is public-key encryption)