Public Key Cryptography

EECE 412

What is it?

- Two keys
  - Sender uses recipient's **public key** to encrypt
  - Receiver uses his **private key** to decrypt
- Based on **trap door, one way function**
  - Easy to compute in one direction
  - Hard to compute in other direction
  - "Trap door" used to create keys
  - Example: Given p and q, product $N=pq$ is easy to compute, but given N, it is hard to find p and q

How is it used?

- Encryption
  - Suppose we encrypt M with Bob's public key
  - Only Bob's private key can decrypt to find M
- Digital Signature
  - **Sign** by "encrypting" with private key
  - Anyone can **verify** signature by "decrypting" with public key
  - But only private key holder could have signed
  - Like a handwritten signature

Topic Outline

- The Random Oracle model for Public Key Cryptosystems
  - Public key encryption and trapdoor one-way permutations
  - Digital signatures
- Looking under the hood
  - Knapsack
  - RSA
- Uses of Public Crypto
- The order of sign and encrypt

Public Key Encryption and Trap-door One-Way Permutation as Random Oracle

- Public Key Encryption Scheme:
  - Key pair (KR, KR^{-1}) generation function from random string R
    - $KR \rightarrow KR^{-1}$ is infeasible
  - $C = \{M\}_{KR}$
  - $M = \{C\}_{KR^{-1}}$

- In:
  - fixed size short string (plaintext) M,
  - Key KR
- Out: fixed size short string (ciphertext) C

Digital Signature as Random Oracle

- Public Key Signature Scheme:
  - Key pair (σR, VR) generation function
    - VR → σR is infeasible
  - $S = Sig_{\sigma R}(M)$
  - $\{True, False\} = Ver_{VR}(S)$

	Signing	Verifying
Input	Any string M + σR	S + VR
Output	S = hash(M) | cipher block	"True" or "False"

Looking Under the Hood

Knapsack Cryptosystem

Knapsack Problem

- Given a set of n weights $W_0, W_1, \ldots, W_{n-1}$ and a sum S, is it possible to find $a_i \in \{0,1\}$ so that

$$S = a_0W_0 + a_1W_1 + \ldots + a_{n-1}W_{n-1}$$

(technically, this is "subset sum" problem)

- **Example**
  - Weights (62,93,26,52,166,48,91,141)
  - Problem: Find subset that sums to $S=302$
  - Answer: 62+26+166+48=302
- The (general) knapsack is NP-complete

Knapsack Problem

- General knapsack (GK) is hard to solve
- But **super-increasing knapsack** (SIK) is easy
- SIK: each weight greater than the sum of all previous weights
- **SIK Example**
  - Weights (2,3,7,14,30,57,120,251)
  - Problem: Find subset that sums to S=186
  - Work from largest to smallest weight
  - Answer: 120+57+7+2=186

Knapsack Cryptosystem

1. Generate super-increasing knapsack (SIK)
2. Convert SIK into "general" knapsack (GK)
3. **Public Key:** GK
4. **Private Key:** SIK plus conversion factors

- Easy to encrypt with GK
- With private key, easy to decrypt (convert ciphertext to SIK)
- Without private key, must solve GK (???)

Knapsack Cryptosystem

- Let $(2,3,7,14,30,57,120,251)$ be the SIK
- Choose $m = 41$ and $n = 491$ with m, n relatively prime and n greater than sum of elements of SIK
- General knapsack

$$(2 \cdot 41) \bmod 491 = 82$$

$$3 \cdot 41 \bmod 491 = 123$$

$$7 \cdot 41 \bmod 491 = 287$$

$$14 \cdot 41 \bmod 491 = 83$$

$$30 \cdot 41 \bmod 491 = 248$$

$$57 \cdot 41 \bmod 491 = 373$$

$$120 \cdot 41 \bmod 491 = 10$$

$$251 \cdot 41 \bmod 491 = 471$$

- General knapsack: $(82,123,287,83,248,373,10,471)$

Knapsack Example

- **Private key:** (2,3,7,14,30,57,120,251), n = 491, m^{-1}=12
  - m^{-1} mod n = 41^{-1} mod 491 = 12
  - (x^{-1} x) mod n = 1 mod n
- **Public key:** (82,123,287,83,248,373,10,471)
- **Throw away:** m = 41
- Example: Encrypt 150 = 10010110

  82 + 83 + 373 + 10 = 548 = C
- To decrypt,
  - (C m^{-1}) mod n = (548 · 12) mod 491 = 193 mod 491
  - Solve (easy) SIK with S = 193
  - Obtain plaintext 10010110 = 150

Knapsack Weakness

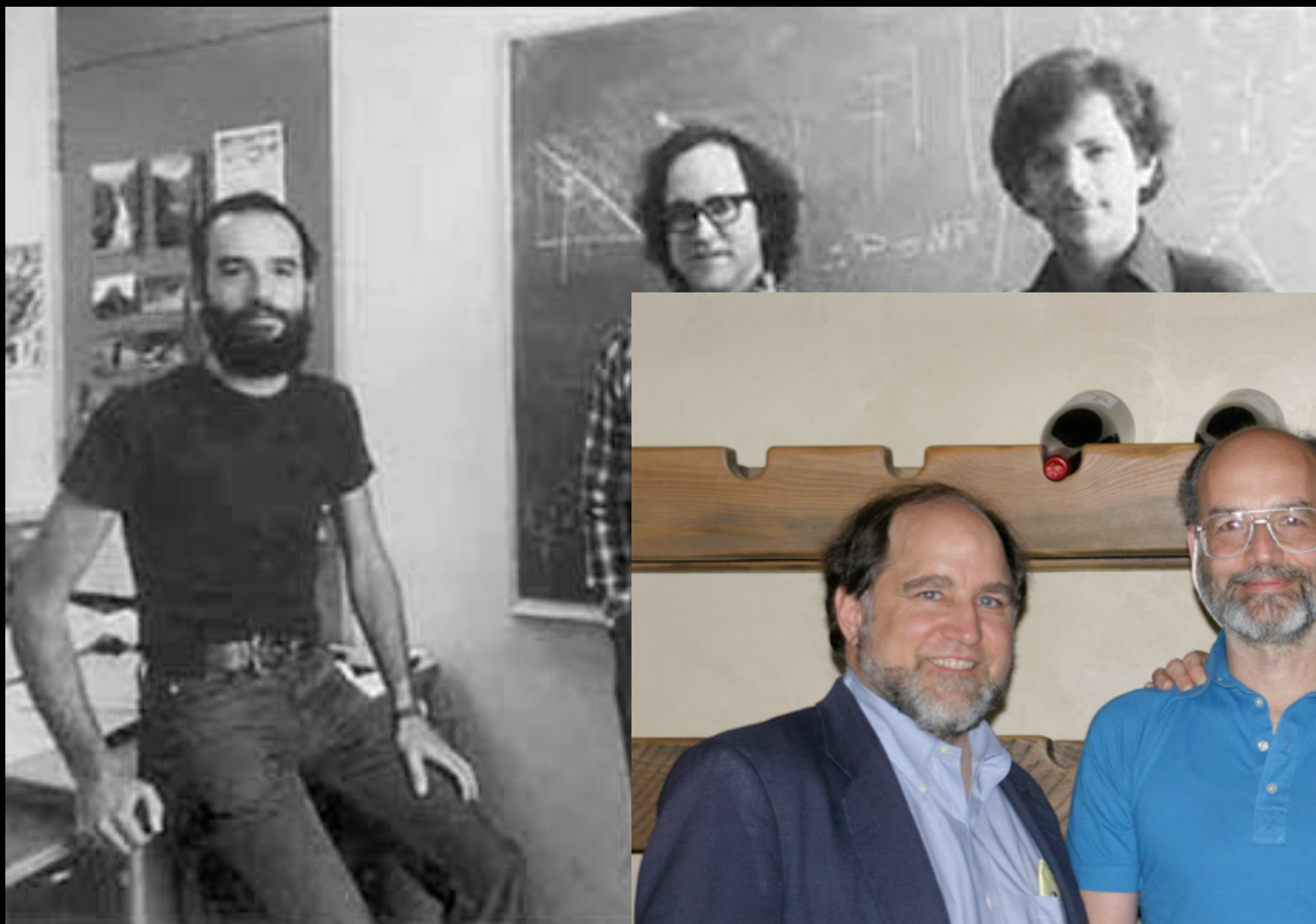
- **Trapdoor:** Convert SIK into "general" knapsack using modular arithmetic
- **One-way:** General knapsack easy to encrypt, hard to solve; SIK easy to solve
- This knapsack cryptosystem is **insecure**
  - Broken by Shamir in 1983 with Apple II computer
  - The attack uses **lattice reduction**
- "General knapsack" is not general enough!
- This special knapsack is easy to solve!

RSA

Cocks (GCHQ), independently, by

Rivest, Shamir and Adleman (MIT)

Rivest, Shamir, and Adleman



1978

2003

basics

- Let p and q be two large (e.g., 200 digits) prime numbers
  - use probabilistic primality tests to find p & q quickly
- Let $n = p \times q$ be the modulus
  - Factoring n is supposed to be hard (i.e., billions of years)
- e relatively prime to $(p-1)(q-1)$ -- encryption exponent
- $d = e^{-1} \bmod (p-1)(q-1)$ -- decryption exponent
- **Throw Away:** p, q
- **Public key:** (n, e)
- **Private key:** d
- Notation: public is in cyan, secret is in red

encrypting & decrypting

- To encrypt message M compute
  - $C = M^e \bmod n$ -- fast with modular exponentiation
- To decrypt C compute
  - $M = C^d \bmod n$
- Recall that e and n are public
- If attacker can factor n, he can use e to easily find d since $ed = 1 \bmod (p-1)(q-1)$
- Factoring the modulus breaks RSA
- It is not known whether factoring is the only way to break RSA

RSA in the works

simple RSA example: initialization

- Select "large" primes $p = 43$, $q = 59$
- Then $n = p \times q = 2537$ and $(p-1)(q-1) = 2436$
- Choose $e = 13$ (relatively prime to 2436)
- Find d such that $ed = 1 \bmod (p-1)(q-1)$, we find that $d = 937$ works
  - note: d exists because $\gcd(e, (p-1)(q-1)) = 1$
- **Public key:** $(N, e) = (2537, 13)$
- **Private key:** $d = 937$

simple RSA example: encryption

- plain text: M = "STOP" = (18 19, 14 15)
- ciphertext: $C = M^e \bmod n$ =
  ($1819^{13} \bmod 2537$, $1415^{13} \bmod 2537$) =
  20 81 21 82 = "UDVE"
  - fast modular exponentiation

simple RSA example: decryption

- $ed = 1 \bmod (p-1)(q-1) \Rightarrow \exists$ k s.t. $ed = k(p-1)(q-1)+1$
- $C^d \equiv (M^e)^d = M^{de} = M^{1+ k(p-1)(q-1)} (\text{mod } n)$
- $M^{p-1} \equiv 1 \bmod p$ and $M^{q-1} \equiv 1 \bmod q$ by Fermat's Little Theorem:
  - If *p* is prime and *a* is an integer not divisible by *p*, then $a^{p-1} \equiv 1 \bmod p$. Furthermore $a^p \equiv a \bmod p$
- $C^d \equiv M \times ((M^{p-1})^{k(q-1)}) \bmod p \equiv M \times 1 \bmod p \equiv M \bmod p$
- $C^d \equiv M \times ((M^{q-1})^{k(p-1)}) \bmod p \equiv M \times 1 \bmod p \equiv M \bmod q$
- Because gcd(p,q) =1, $C^d \equiv M \bmod p \times q$ by Chinese Remainder Theorem

simple RSA example: decryption

- Decrypt message 0981 0461
- $M \equiv C^d \bmod p \times q$
- $0981^{937} \bmod 2537 = 0704$ = "HE"
- $0461^{937} \bmod 2537 = 1115$ = "LP"
- HELP

Uses for Public Key Crypto

Uses for Public Key Crypto

- Confidentiality
  - Transmitting data over insecure channel
  - Secure storage on insecure media
- Authentication
- Digital signature provides integrity and **non-repudiation**
  - No non-repudiation with symmetric keys

Non-non-repudiation

- Alice orders 100 shares of stock from Bob
- Alice computes **MAC** using symmetric key
- Stock drops, Alice claims she did not order
- Can Bob prove that Alice placed the order?
- **No!** Since Bob also knows symmetric key, he could have forged message
- **Problem:** Bob knows Alice placed the order, but he can't prove it

Non-repudiation

- Alice orders 100 shares of stock from Bob
- Alice **signs** order with her private key
- Stock drops, Alice claims she did not order
- Can Bob prove that Alice placed the order?
- **Yes!** Only someone with Alice's private key could have signed the order
- This assumes Alice's private key is not stolen (revocation problem)

Sign and Encrypt
vs
Encrypt and Sign

Public Key Notation

- **Sign** message M with Alice's **private key:** $[M]_{Alice}$
- **Encrypt** message M with Alice's **public key:** $\{M\}_{Alice}$
- Then

$$\{[M]_{Alice}\}_{Alice} = M$$

$$[\{M\}_{Alice}]_{Alice} = M$$

Confidentiality and Non-repudiation

- Suppose that we want confidentiality and non-repudiation
- Can public key crypto achieve both?
- Alice sends message to Bob
  - **Sign and encrypt** $\{[M]_{Alice}\}_{Bob}$
  - **Encrypt and sign** $[\{M\}_{Bob}]_{Alice}$
- Can the order possibly matter? (see Stamp)

Sign and Encrypt

M = “I love you”

Alice

$\{[M]_{Alice}\}_{Bob}$

Bob

$\{[M]_{Alice}\}_{Charlie}$

Charlie

Q: What is the problem?

A: Charlie misunderstands crypto!

Encrypt and Sign

M = "My theory, which is mine, is this: ...."

Alice

$[\{M\}_{Bob}]_{Alice}$ →

Charlie

$[\{M\}_{Bob}]_{Charlie}$ →

Bob

Note that Charlie cannot decrypt M

Q: What is the problem?

A: Bob misunderstands crypto!

Summary

- The Random Oracle model for Public Key Cryptosystems
  - Public key encryption and trapdoor one-way permutations
  - Digital signatures
- Looking under the hood
  - Knapsack
  - RSA
- Uses of Public Crypto
- The order of sign and encrypt