CSEC
CYBER SECURITY
EDUCATION CONSORTIUM

0100101010010101010101010011001010101010000101010101010101010100010100100

Encryption & Decryption – 3

Sujeet Shenoi
Tandy School of Computer Science
University of Tulsa, Tulsa, OK 74104
sujeet@utulsa.edu

ADVANCED TECHNOLOGICAL EDUCATION

Secure Encryption Systems

01001010100101010101010100110010101010100001010101010101010101000101001000

- Modern techniques are based on "Hard Problems" (NP-Complete Problems)
- Involve heuristic search (2^n possibilities)
- Satisfiability
  - — Pick v_1 v_2, v_3: Boolean such that $(v_1) \wedge (v_2 \vee v_3) \wedge (\neg v_3 \vee \neg v_1)$ is True
- Knapsack
  - — Pick v_1 v_2, v_3 ε $\{0,1\}$ such that $v_1{*}a_1 + v_2{*}a_2 + v_3{*}a_3 = T$ (Target sum)

Classes P, NP and EXP

Class P

- Set of problems whose solutions run in time bounded by "polynomial functions" of the size of the problems

Class NP

- Set of problems whose solutions run in time bounded by polynomial functions of the size of the problems "assuming the ability to guess perfectly"

Class EXP

- Set of problems whose solutions run in time bounded by "exponential functions" of the size of the problems

Classes P, NP and EXP (contd.)

Fundamental Result: $P \subseteq NP \subset EXP$

Is: $P \subseteq NP$ or $P = NP$? Not known!

Some Comments

- NP-Complete problem does not guarantee that there is no solution easier than exponential
- Every NP-Complete problem has a solution that runs in time proportional to 2^n; feasible if n is small
- Non-determinism can be modeled by “threads”
- Interceptors may use other information to simplify the task of breaking the encryption

CSEC
CYBER SECURITY
EDUCATION CONSORTIUM

Secret & Public Encryption Algorithms

01001010100101010101010100110010101010100001010101010101010101000101001000

Secret Key Algorithms (Symmetric)

- One key for encryption and decryption ($K_E = K_D = K$)
- $C = \{ P \}_K$ and $P = \{ C \}_K$
- One key per channel (#keys = n*(n-1)/2)

Public Key Algorithms (Asymmetric)

- Separate keys for encryption and decryption ($K_E \neq K_D$)
- $C = \{ P \}_{K_E}$ and $P = \{ C \}_{K_D}$
- $C = \{ P \}_{K_D}$ and $P = \{ C \}_{K_E}$
- Two keys per user (#keys = 2*n)

NSF

ADVANCED
TECHNOLOGICAL
EDUCATION

Public Key Algorithms

01001010100101010101010100110010101010100001010101010101010101000101001 00

Public Key Algorithms (Asymmetric)

- Key Pair: (K_A^{priv}, K_A^{pub})
- K_A^{priv}: Private Key; K_A^{pub}: Public Key
- K_A^{priv} is kept by secret by A
- K_A^{pub} is distributed widely by A

- A $\rightarrow$ Receiver: $C = \{ P \}_{K_A^{priv}}$ (and $P = \{ C \}_{K_A^{pub}}$)

- Sender $\rightarrow$ A: $C = \{ P \}_{K_A^{pub}}$ (and $P = \{ C \}_{K_A^{priv}}$)

Merkle-Hellman Algorithm

010010101001010101010101001100101010101000010101010101010101010100010100100

Merkle-Hellman (1978)

- Encodes a binary message as a solution to the knapsack problem
- NP-complete problem
- Simple knapsack (linear time)
- Hard knapsack (exponential time)
- “One way” encryption

Merkle-Hellman Algorithm (contd.)

010010101001010101010101001100101010101000010101010101010101010100010100100

General Knapsack

- Given $S = [a_1, a_2, \ldots, a_n]$ and target sum T,
- Find $V= [v_1, v_2, \ldots, v_n]$, $v_i \varepsilon \{0,1\}$ such that $\Sigma_{i=1}^{n} a_i*v_i = T$
- E.g., if $S = [9, 5, 2, 13]$ and $T = 24$, then $V = [1, 0, 1, 1]$

Superincreasing Knapsack

- Each $a_k \varepsilon S$ satisfies the condition: $a_k > \Sigma_{j=1}^{k-1} a_j$
- E.g., $S = [1, 2, 5, 13]$ is a superincreasing knapsack

Merkle-Hellman Algorithm (contd.)

Sending an Encrypted Message

- Receiver picks a simple (superincreasing) knapsack (S), multiplier (w) and modulus (n) (w and n are co-prime)
- S = [1, 2, 6]; w = 11; n = 13 (n prime; larger than 9)
- Receiver computes "hard knapsack" ($H = [h_1, h_2, h_3]$)
- $h_i = w * s_i \bmod n$
- $h_1 = w * s_1 \bmod n = 11 * 1 \bmod 13 = 11 \bmod 13 = 11$
- $h_2 = w * s_2 \bmod n = 11 * 2 \bmod 13 = 22 \bmod 13 = 9$
- $h_3 = w * s_3 \bmod n = 11 * 6 \bmod 13 = 66 \bmod 13 = 1$

Merkle-Hellman Algorithm (contd.)

Sending an Encrypted Message (contd.)

- Receiver sends H = [11, 9, 1] to sender
- Receiver keeps S, w and n secret
- Suppose sender wishes to transmit P = 101 010 011
- P: 1 0 1 0 1 0 0 1 1

  11 9 1 11 9 1 11 9 1
- C: 12 9 10

Merkle-Hellman Algorithm (contd.)

Decrypting an Encrypted Message

- $H = w * S \bmod n$
- $C = H * P = w * S * P \bmod n$
- $w^{-1} * C = w^{-1} * H * P = w^{-1} * w * S * P = S * P \bmod n$
- $w^{-1} * C_i = S * P_i \bmod n$ (note: $S = [1, 2, 6]$
- $6 * 12 = 72 \bmod 13 = 7 = 101$
- $6 * 9 = 54 \bmod 13 = 2 = 010$
- $6 * 10 = 60 \bmod 13 = 8 = 011$

Merkle-Hellman Algorithm (contd.)

0100101010010101010101010011001010101010000101010101010101010100010100100

Cryptanalysis

- Modulus n: 200 bits long
- s_i are chosen to be approx. 2^{200} apart!
- Knapsack has approx. 200 terms ($m = 200$)
  - —Choose m random numbers between 0 and 2^{200}
  - —$s_i = 2^{200*i-1} + r_i$
- Each term is 200 to 400 bits long
- 1 opn/μs: 10^{47} years to try 2^{200} choices for each s_i
- Hard to break for large values of n & m

Merkle-Hellman Algorithm (contd.)

0100101010010101010101010011001010101010000101010101010101010100010100100

Weaknesses

- Shamir (1980): If modulus n is known, it is possible to determine simple knapsack S in polynomial time

RSA Algorithm

0100101010010101010101010011001010101010000101010101010101010100010100100

Rivest-Shamir-Adelman (1978)

- Based on factoring large numbers (200 digits)
- Best factorization algorithm is exponential
- No known weaknesses
  - Choose large $n = p*q$ (p, q: prime numbers)
  - Choose e relatively prime to $\varphi(n) = (p-1)*(q-1)$
  - $d = e^{-1} \bmod \varphi(n)$
- Encryption key: (e, n); Decryption key: (d, n)
- $C = P^e \bmod n$; $P = C^d \bmod n$
- $C = P^d \bmod n$; $P = C^e \bmod n$

RSA Algorithm (contd.)

01001010100101010101010100110010101010100001010101010101010101000101001000

RSA Mathematics

- Euler totient function ($\varphi(n)$): number of positive integers less than n that are relatively prime to n
- If p: prime, then $\varphi(p) = p - 1$
- If $n = p * q$ and p, q: prime, then

  $$\varphi(n) = \varphi(p) * \varphi(q) = (p - 1) * (q - 1)$$

Euler-Fermat Result

- For any integer x, if n and x are rel. prime, then

  $$x^{\varphi(n)} \equiv 1 \bmod n$$

RSA Algorithm (contd.)

Main Result: $(P^e)^d \equiv (P^d)^e \equiv P \bmod n$

- $e * d \equiv 1 \bmod \varphi(n)$ where $n = p*q$
- $e * d \equiv k * \varphi(n) + 1$ for some integer k
- $P^{p-1} \equiv 1 \bmod p$ E-F result: P, p rel prime
- $P^{k\varphi(n)} \equiv 1 \bmod p$ (p-1) is a factor of $\varphi(n)$
- $P^{k\varphi(n)+1} \equiv P \bmod p$ multiplying by P
- $P^{k\varphi(n)+1} \equiv P \bmod q$ same result for q
- $(P^e)^d = P^{ed} = P^{k\varphi(n)+1} = P \bmod p = P \bmod q$
- $(P^e)^d = P \bmod n$ $n = p*q$

RSA Algorithm (contd.)

01001010100101010101010100110010101010100001010101010101010101010001010010

RSA Encryption

- Suppose p = 3 and q = 11
- n = 33 and $\varphi(n) = (p-1)*(q-1) = 20$
- Choose e = 13 (relatively prime to 20)
- Find d such that $e * d \equiv 1 \bmod 20 \Rightarrow d = 17$
- Public key (e, n) = (13, 33)
- Private key (d, n) = (17, 33)
- Plaintext P = 7
- $C = 7^{13} \bmod 33 = 13$
- $P = 13^{17} \bmod 33 = 7$

RSA Algorithm (contd.)

Using RSA

- Choose primes p, q (100 digits each)
- Calculate $n = p*q$ (200 digits/512 bits; 1024bits recommended for secure applications)
- Choose large e relatively prime to $\varphi(n)$
- Compute d such that $e * d \equiv 1 \bmod \varphi(n)$
- Public key (e, n)
- Private key (d, n)
- Can discard p, q, $\varphi(n)$
- Primality Test (iteration k: Prob(p is not prime) = $1/2^k$)
  - —$\gcd(p, r) = 1$
  - —Jacobi Function: $J(r, p) \equiv r^{(p-1)/2} \bmod p$

Digital Signature Algorithms

0100101010010101010101010011001010101010000101010101010101010100010100100

- El Gamal Algorithm (1984)
  - Pick p: prime; $a < p$ and $x < p$; $(p-1)$ has a large prime factor: q
  - Compute: $y = a^x \bmod p$
  - Private key: x; Public key: y (and p, a)
- Message Signing (m: message)
  - Pick k: $0 < k < p-1$ (relatively prime to p-1)
  - Compute: $r = a^k \bmod p$
  - Compute: $s = k^{-1}*(m - x*r) \bmod (p-1)$ $(k*k^{-1} \equiv 1 \bmod (p-1))$
  - Message Signature: r & s
- Signature Verification
  - Compute: $y^r r^s \bmod p$
  - Compute: $a^m \bmod p$
  - Check: $y^r r^s \bmod p \equiv a^m \bmod p$

Digital Signature Algorithms (contd.)

- U.S. Digital Signature Algorithm (1994)
  - DSS (Digital Signature Standard)
  - El Gamal Algorithm with restrictions
    - p: 170 digits long $(2^{511} < p < 2^{512})$
    - q: prime factor of p-1 $(2^{159} < q < 2^{160})$
    - Hash value of m: H(m) used instead of m
    - Computations of r and s taken mod q
    - Changes simplify the algorithm
    - Changes weaken encryption

Cryptographic Hash Algorithms

0100101010010101010101010011001010101010000101010101010101010100010100100

- Simpler than Digital Signature Algorithms
- Hash function (f) produces "digest" of data/message
- $S \rightarrow R$: m, f(m)
- R: computes new f(m) & compares with old f(m)
- Difficult to "invert," i.e., change m <u>and</u> f(m)

- XOR bits: 10101010 00101111 $\rightarrow$ 1
  (Prob = 1/2)
- XOR bytes: 10101010 00101111 $\rightarrow$ 10000101
  (Prob = $1/2^8$)
- Most digests are between 100 to 1,000 bits

Secure Hash Algorithm (SHA)

- Designed for Digital Signature Standard (DSS)
- NIST (1992-1995)
- Input: $\leq 2^{64}$ bits; Digest: 160 bits
- Operations: XOR, + mod 2^{32}, left circular shift(n,v)

- Algorithm: Non-linear function; interweaves bits
  - Pad message: Multiple of 512 bits (msg 1 0…0 <64-bit length>)
    (512 bits = 16 32-bit words: $W_0 \ldots W_{15}$)
  - Expand to 80 words: $W_0 \ldots W_{79}$
  - Initialize 5 32-bit pattern constants: $H_0^0 \ldots H_4^0$
  - Perform 80-step 4-round diffusion algorithm: digest = $H_0^{80} \ldots H_4^{80}$

MD4 and MD5 Hash Algorithms

- MD4 (Rivest, 1991-92)
  - Exceptionally fast, less secure
  - 16-word block (512 bits)
  - 48-step 3-round diffusion algorithm
  - 4 pattern constants (128 bits)

- MD5 (Rivest, 1992)
  - Slower, more secure
  - 16-word block (512 bits)
  - 64-step 4-round diffusion algorithm
  - 4 pattern constants (128 bits)

Quantum Cryptography

010010101001010101010101001100101010101000010101010101010101010100010100100

Rationale

- One-Time Pad: Only provably unbreakable system
- Requires a long, unpredictable string of numbers
  - —String generation
  - —String communication
- Quantum cryptography addresses both problems
- Based on physics instead of mathematics
- Quantum Key Distribution (QKD)

Quantum Cryptography (contd.)

- Photons
  - Assume four directional orientations (–, |, <, >,)
  - Orientations – and | can be distinguished with high certainty, but < and > sometimes appear as – or |
  - Orientations < and > can be distinguished with high certainty, but – and | sometimes appear as < or >
- Polarizing Filters (+ and o)
  - \+ Rectilinear Filter: Discriminates – and |, but has 50% probability of counting < and > as – or |
  - o Circular Filter: Discriminates < and >, but has 50% probability of counting – and | as < or >

Quantum Cryptography (contd.)

- BB84: Protocol: Bennett and Brassard (1984)
  - Sender sends a stream of photons to receiver
  - Sender uses + or o filter to control each photon being sent
  - Receiver uses + or o filter and records photon orientation
  - Nobody can eavesdrop without disrupting communication
  - Using a filter to view a photon disrupts the communication
  - E.g., A rectilinear + filter allows – photons, and some < and > photons, but blocks all | photons
  - http://fredhenle.net/bb84/

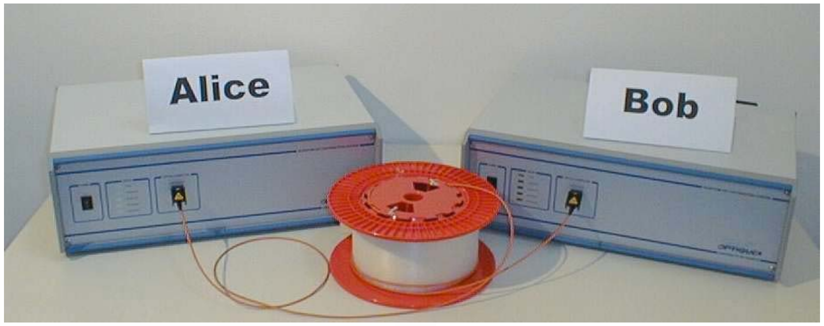
Quantum Cryptography (contd.)

- BB84 Protocol
  - S & R: – & < represent 0; | & > represent 1
  - S: Sends series of photons to R and records orientation
  - R: Uses + or o filters at random and records orientation
  - R: Sends series of filters used to S (public channel)
  - S: Sends the correct filters used to R (public channel)
  - R: Determines which photons received were correct
  - Inefficient, only half the bandwidth of the communications channel carries meaningful data
  - Problems with sending and receiving photons

Quantum Cryptography (contd.)

0100101010010101010101010011001010101010000101010101010101010100010100100

- IBM T.J. Watson Research Center (U.S.)
  - Aunt Martha's Coffin (1989): 1 foot
- Id Quantique (Switzerland)
  - Optical fiber system: 20 miles
- MagiQ Technologies (U.S.)
  - Optical fiber system: 65 miles
  - Cost: $70,000 to $100,000



Quantum Cryptography (contd.)

0100101010010101010101010011001010101010000101010101010101010100010100100

- Los Alamos/NIST (U.S.)
  - Optical fiber system (2007): 100 miles
- European Consortium (Canary Islands, Spain)
  - Air (2007): 90 miles
- Satellite Transmission (250 miles) is a possibility