

CSE331: Introduction to Networks and Security

Lecture 24
Fall 2006



Announcements

- Midterm 2 is Friday, November 10th
- Project 3 has been assigned:
 - It's due November 20th
 - It covers basic cryptography / DES



Advanced Encryption Standard (AES)

- National Institute of Standards & Technology
NIST
 - Computer Security Research Center (CSRC)
 - <http://csrc.nist.gov/>
 - <http://www.esat.kuleuven.ac.be/~rijmen/rijndael/>
- Uses the Rijndael algorithm
 - Invented by Belgium researchers
Dr. Joan Daemen & Dr. Vincent Rijmen
 - Adopted May 26, 2002
 - Key length: 128, 192, or 256 bits
 - Block size: 128, 192, or 256 bits

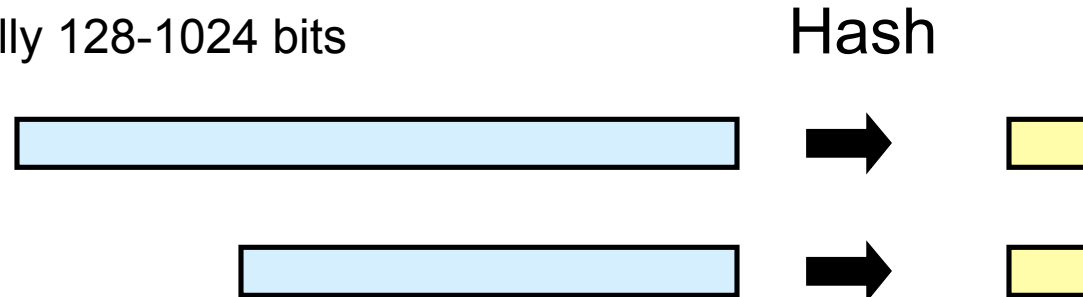


Problems with Shared Key Crypto

- Compromised key means interceptors can decrypt any ciphertext they've acquired.
 - Change keys frequently to limit damage
- Distribution of keys is problematic
 - Keys must be transmitted securely
 - Use couriers?
 - Distribute in pieces over separate channels?
- Number of keys is $O(n^2)$ where n is # of participants
- Potentially easier to break?

Hash Algorithms

- Take a variable length string
- Produce a fixed length digest
 - Typically 128-1024 bits



- (Noncryptographic) Examples:
 - Parity (or byte-wise XOR)
 - CRC
- Realistic Example
 - The NIST Secure Hash Algorithm (SHA) takes a message of less than 2^{64} bits and produces a digest of 160 bits

Cryptographic Hashes

- Create a hard-to-invert summary of input data
- Useful for integrity properties
 - Sender computes the hash of the data, transmits data and hash
 - Receiver uses the same hash algorithm, checks the result
- Like a check-sum or error detection code
 - Uses a cryptographic algorithm internally
 - More expensive to compute
- Sometimes called a Message Digest
- Examples:
 - Secure Hash Algorithm (SHA)
 - Message Digest (MD4, MD5)

Uses of Hash Algorithms

- Hashes are used to protect *integrity* of data
 - Virus Scanners
 - Program fingerprinting in general
 - Modification Detection Codes (MDC)
- Message Authenticity Code (MAC)
 - Includes a cryptographic component
 - Send (msg, hash(msg, key))
 - Attacker who doesn't know the key can't modify msg (or the hash)
 - Receiver who knows key can verify origin of message
- Make digital signatures more efficient

Desirable Properties

- The probability that a randomly chosen message maps to an n -bit hash should ideally be $(\frac{1}{2})^n$.
 - Attacker must spend a lot of effort to be able to modify the source message without altering the hash value
- Hash functions h for cryptographic use as MDC's fall in one or both of the following classes.
 - *Collision Resistant Hash Function*: It should be computationally infeasible to find two distinct inputs that hash to a common value (ie. $h(x) = h(y)$).
 - *One Way Hash Function*: Given a specific hash value y , it should be computationally infeasible to find an input x such that $h(x)=y$.

Secure Hash Algorithm (SHA)

- Pad message so it can be divided into 512-bit blocks, including a 64 bit value giving the length of the original message.
- Process each block as 16 32-bit words called $W(t)$ for t from 0 to 15.
- Expand from these 16 words to 80 words by defining as follows for each t from 16 to 79:
 - $W(t) := W(t-3) \oplus W(t-8) \oplus W(t-14) \oplus W(t-16)$
- Constants H_0, \dots, H_5 are initialized to special constants
- Result is final contents of H_0, \dots, H_5

for each 16-word block begin

A := H0; B := H1; C := H2; D := H3; E := H4

for I := 0 to 19 begin

TEMP := S(5,A) + ((B ∧ C) ∨ (¬B ∧ D)) + E + W(I) + 5A827999;

E := D; D := C; C := S(30,B); B := A; A := TEMP

end

Chaining Variables

for I := 20 to 39 begin

TEMP := S(5,A) + (B ⊕ C ⊕ D) + E + W(I) + 6ED9EBA1;

E := D; D := C; C := S(30,B); B := A; A := TEMP

end

for I := 40 to 59 begin

TEMP := S(5,A) + ((B ∧ C) ∨ (B ∧ D) ∨ (C ∧ D)) + E + W(I) + 8F1BBCDC;

E := D; D := C; C := S(30,B); B := A; A := TEMP

end

for I := 60 to 79 begin

Shift A left 5 bits

TEMP := S(5,A) + (B ⊕ C ⊕ D) + E + W(I) + CA62C1D6;

E := D; D := C; C := S(30,B); B := A; A := TEMP

end

H0 := H0+A; H1 := H1+B; H2 := H2+C; H3 := H3+D; H4 := H4+E

end

Public Key Cryptography

- Sender encrypts using a *public* key
- Receiver decrypts using a *private* key
- Only the private key must be kept secret
 - Public key can be distributed at will
- Also called *asymmetric* cryptography
- Can be used for digital signatures
- Examples: RSA, El Gamal, DSA

Public Key Notation

- Encryption algorithm
 $E : \text{keyPub} \times \text{plain} \rightarrow \text{cipher}$
Notation: $K\{\text{msg}\} = E(K, \text{msg})$
- Decryption algorithm
 $D : \text{keyPriv} \times \text{cipher} \rightarrow \text{plain}$
Notation: $k\{\text{msg}\} = D(k, \text{msg})$
- D inverts E
 $D(k, E(K, \text{msg})) = \text{msg}$
- Use capital “K” for public keys
- Use lower case “k” for private keys
- Sometimes E is the same algorithm as D

Secure Channel: Private Key

Alice



K_A, K_B
 k_A

Bart



K_A, K_B
 k_B

$K_B\{\text{Hello!}\}$

$K_A\{\text{Hi!}\}$

Trade-offs for Public Key Crypto

- More computationally expensive than shared key crypto
 - Algorithms are harder to implement
 - Require more complex machinery
- More formal justification of difficulty
 - Hardness based on complexity-theoretic results
- A principal needs one private key and one public key
 - Number of keys for pair-wise communication is $O(n)$



RSA Algorithm

- Ron Rivest, Adi Shamir, Leonard Adleman
 - Proposed in 1979
 - They won the 2002 Turing award for this work
- Has withstood years of cryptanalysis
 - Not a guarantee of security!
 - But a strong vote of confidence.
- Hardware implementations: 1000 x slower than DES

RSA at a High Level

- Public and private key are derived from secret prime numbers
 - Keys are typically ≥ 256 bits (512, or 1024)
- Plaintext message (a sequence of bits)
 - Treated as a (large!) binary number
- Encryption is modular exponentiation
- To break the encryption, conjectured that one must be able to factor large numbers
 - Not known to be in P (polynomial time algorithms)

Number Theory: Modular Arithmetic

- Examples:
 - $10 \bmod 12 = 10$
 - $13 \bmod 12 = 1$
 - $(10 + 13) \bmod 12 = 23 \bmod 12 = 11 \bmod 12$
 - $23 \equiv 11 \pmod{12}$
 - “23 is congruent to 11 (mod 12)”
- $a \equiv b \pmod{n}$ iff $a = b + kn$ (for some integer k)
- The *residue* of a number modulo n is a number in the range $0 \dots n-1$

Modular Arithmetic Properties

- Commutative, Associative, Distributive
- Reduce intermediate results mod n :
 - $(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$
 - $(a - b) \bmod n = ((a \bmod n) - (b \bmod n)) \bmod n$
 - $(a * b) \bmod n = ((a \bmod n) * (b \bmod n)) \bmod n$
 - $(a * (b + c)) \bmod n =$
 $((a * b) \bmod n + (a * c) \bmod n) \bmod n$

Number Theory: Prime Numbers

- A *prime number* is an integer > 1 whose only factors are 1 and itself.
- Two integers are *relatively prime* if their only common factor is 1
 - gcd = greatest common divisor
 - $\text{gcd}(a,b) = 1$
 - $\text{gcd}(15,12) = 3$, so they're not relatively prime
 - $\text{gcd}(15,8) = 1$, so they are relatively prime

Finite Fields (Galois Fields)

- For a prime p , the set of integers mod p forms a *finite field*
- Addition $+$ Additive unit 0
- Multiplication $*$ Multiplicative unit 1
- Inverses: $n * n^{-1} = 1$ for $n \neq 0$
 - Suppose $p = 5$, then the finite field is $\{0, 1, 2, 3, 4\}$
 - $2^{-1} = 3$ because $2 * 3 \equiv 1 \pmod{5}$
 - $4^{-1} = 4$ because $4 * 4 \equiv 1 \pmod{5}$
- Usual laws of arithmetic hold for modular arithmetic:
 - Commutativity, associativity, distributivity of $*$ over $+$

RSA Key Generation

- Choose large primes p and q .
 - Should be roughly equal length (in bits)
- Let $n = p * q$
- Choose a random encryption exponent e
 - With requirement: e and $(p-1)*(q-1)$ are relatively prime.
- Derive the decryption exponent d
 - $d = e^{-1} \text{ mod } ((p-1)*(q-1))$
 - d is e 's inverse mod $((p-1)*(q-1))$
- Public key: $K = (e, n)$ pair of e and n
- Private key: $k = (d, n)$
- Discard primes p and q (they're not needed anymore)

RSA Encryption and Decryption

- Message: m
- Assume $m < n$
 - If not, break up message into smaller chunks
 - Good choice: largest power of 2 smaller than n
- Encryption: $E((e,n), m) = m^e \bmod n$
- Decryption: $D((d,n), c) = c^d \bmod n$

Example RSA

- Choose $p = 47$, $q = 71$
- $n = p * q = 3337$
- $(p-1)*(q-1) = 3220$
- Choose e relatively prime with 3220: $e = 79$
 - Public key is $(79, 3337)$
- Find $d = 79^{-1} \bmod 3220 = 1019$
 - Private key is $(1019, 3337)$
- To encrypt $m = 688232687966683$
 - Break into chunks < 3337
 - 688 232 687 966 683
- Encrypt: $E((79, 3337), 688) = 688^{79} \bmod 3337 = 1570$
- Decrypt: $D((1019, 3337), 1570) = 1570^{1019} \bmod 3337 = 688$