


Encryption

A Brief Overview


CSE870: Advanced Software Engineering: Cheng 1



Encryption

- Encryption:
 - Definition: mechanisms to disguise the message so that if the information is intercepted/diverted, the content of the message will not be understood.
 - Impact: foundational building block to security-based computing


CSE870: Advanced Software Engineering: Cheng 2



Terminology

- Scenario:
 - S wants to send the message T to R, where an outsider, O, wants the message and tries to access it.
 - S: Sender
 - R: Receiver
 - T: Transmission Medium
 - O: Interceptor or Intruder.
- 4 ways O might try to access message.
 - Block it: prevent T from reaching R (availability)
 - Intercept it: read or listen to message (secrecy)
 - Modify it: obtaining message and changing it
 - Fabricate: generate an authentic-looking message to be delivered to R appearing to come from S


CSE870: Advanced Software Engineering: Cheng 3



Terminology

- **Encryption**: process of encoding a message so that its meaning is not obvious
- **Decryption**: transforming encrypted message back to its normal form
- **Encode/decode**: translating **phrases** to other words or phrases
- **Encipher/decipher**: translating **letters or symbols** individually.
- **Plaintext**: original form of message: $P = (p_1, p_2, \dots, p_n)$
- **Ciphertext**: encrypted form of message: $C = (c_1, c_2, \dots, c_n)$
- Encryption/decryption relationships:
 - $C = E(P)$; $P = D(C)$; $P = D(E(P))$


CSE870: Advanced Software Engineering: Cheng 4



Encryption Algorithms

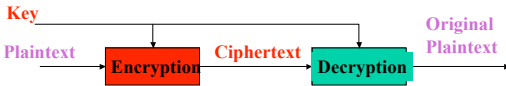
- Some encryption algs use a key K
 - $C = E(K, P)$
 - E is a SET of encryption algs
 - Key K selects specific one
- Symmetric Encryption: $P = D(K, E(K, P))$
 - encryption/decryption keys are the same
- Asymmetric Encryption: $P = D(K_D, E(K_E, P))$

CSE870: Advanced Software Engineering: Cheng 5

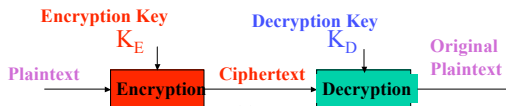


Pictorial Representation


Symmetric Encryption:



Asymmetric Encryption:




CSE870: Advanced Software Engineering: Cheng 6



More Terms

- **Cryptography:** (hidden writing)
 - Practice of using encryption to conceal text
- **Cryptanalyst:**
 - Person who studies encryption and encrypted messages
 - Intent: find hidden meaning
- **Cryptographer and Cryptanalyst:**
 - Both attempt to translate coded material to original form
 - **Cryptographer:** works on behalf of legitimate sender or receiver.
 - **Cryptanalyst:** Works on behalf of unauthorized interceptor
- **Cryptology:** research/study into encryption/decryption
 - Includes cryptography and cryptanalysis.


CSE870: Advanced Software Engineering: Cheng 7



Cryptanalysis

- **Objective: Break an encryption**
 - Deduce the meaning of a ciphertext msg
 - Determine decrypting algorithm that matches an encrypting algorithm
- **Possible techniques:**
 - break single message
 - Recognize patterns in encrypted msgs
 - break subsequent msgs with straightforward decryption alg
 - Find general weaknesses in encryption alg
 - Without necessarily intercepting any msgs
- **Tools:**
 - Encrypted msgs, known encryption algs, intercepted plaintext, data elements known/suspected of being in ciphertext, mathematical/statistical techniques, props of languages, computers, and luck


CSE870: Advanced Software Engineering: Cheng 8



Breakable Encryption

- **Encryption algorithm is BREAKABLE:**
 - Given enough time/data, an analyst could determine alg.
 - Practicality is issue
 - For given cipher scheme, may have 10^{30} possible decipherments
 - Select one from 10^{30}
 - Current technology: perform 10^{10} ops/sec
 - Require 10^{20} secs == 10^{12} years
- **Reality Check:**
 - Cryptanalyst won't just try the "hard" ways
 - Ex: more clever approach, might only take 10^{15} ops
 - 10^{10} ops/sec, 10^{15} ops will take about one day
 - Breakability estimates are based on CURRENT technology

CSE870: Advanced Software Engineering: Cheng 9




Character Representations

- Study ways to encrypt any computer material:
 - ASCII/EBCDIC chars
 - Binary data or Object code
 - Control stream

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

CSE870: Advanced Software Engineering: Cheng 10




Substitution-based Encryption

- Monoalphabetic Ciphers
 - Caesar Cipher:** $c_i = E(p_i) = p_i + 3$
 - wuhdwb lpsrvleoh,
 - wklv phvvdjh lv qrw wrr kdug wr euhdn
 - Easy to perform in field (no written instructions)
 - Permutation:** reordering of the elements
 - $c_i = \pi(p_i)$; $\pi(\lambda) = 25 - \lambda$
 - Use a key:**

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
K	E	Y	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
S	P	E	C	T	A	U	L	R	B	D	F	G	H	I	J	K	N	O	Q	V	W	X	Y	Z		

– **Weakness:** study frequency distribution

CSE870: Advanced Software Engineering: Cheng 11



Polyalphabetic Substitution Ciphers

- Desire flat distribution
- Combine distributions that are high with low ones
 - Encipher **T** as **a** and sometimes as **b**
 - Also encipher **X** as **a** and sometimes as **b**
- Use two separate encryption alphabets
 - Tables for odd and even positions
 - $\pi_1(\lambda) = (3 * \lambda) \bmod 26$
 - $\pi_2(\lambda) = (5 * \lambda) + 13 \bmod 26$
 - TREAT YIMPO SSIBL E
 - Fumnf dyvtf czysh h

CSE870: Advanced Software Engineering: Cheng 12



Substitution Discussion

- Major weakness:
 - frequency distribution
 - (index of coincidence: measure of variation between frequencies in a distribution)
 - Some letters are just used more frequently than others
 - Numerous enciphering techniques still can make it difficult to hide these patterns
 - **Kasiski Method:** find number of alphabets used
 - Identify repeated patterns of 3 or more chars
 - For each pattern, write down position at which each instance of pattern begins
 - Compute difference between start points of success instances
 - Determine all factors of each difference
 - If polyalphabetic subst used, key length will be one of the factors that appears often in previous step



Transpositions (Permutations)

- Definition: encryption where letters are rearranged.
- Goal: diffusion, spread info from message or key out widely across the ciphertext.
- Try to break established patterns.



Transposition Techniques

- Columnar Transpositions:
 - Rearrangement of chars of plaintext into cols

C1	C2	C3	C4	C5
C6	C7	C8	C9	C10
C11	C12	Etc.		

T	H	I	S	I
S	A	M	E	S
S	A	G	E	T
O	S	H	O	W
H	O	W	A	C
O	L	U	M	N
A	R	T	R	A
N	S	P	O	S
I	T	I	O	N
W	O	R	K	S

tssoh oaniw haaso lrsto imghw utpir seeoa mrook istwc nasns



Transpositions

- Digram: patterns of adjacent letters.
 - Study 2 and 3 letter combinations of adj letters
- Double Transposition Alg:
 - Involves 2 columnar transpositions
 - With different number of columns, applied sequentially.
- Fractionated Morse:
 - keyed monoalphabetic cipher
 - Result is subsequently blocked (clustered)
 - Morse code is used as its basis



Secure Encryption Systems

- Previous algs could be completed manually, although tedious
 - Decryption could also be done manually
- New technology requires more “hard” encryption algs to hinder cryptanalysts
- Review 3 key, important encryption algs
- Look at recent developments.



Important Encryption Algs

- Merkle-Hellman knapsack:
 - Alg based on “hard” problems (NP-complete)
- Rivest-Shamir-Adelman (RSA):
 - More resilient to attacks than Merkle alg
- Data Encryption Standard (DES):
 - Developed with support from NIST
 - Provide secure encryption for commercial applications
- Clipper program:
 - Skipjack: cryptographic alg – maintain secrecy



Some "Hard" theories

- NP-complete:
 - Encryption algs that would require NP-complete alg to decrypt
- Number theory:
 - Inverses
 - Primes
 - Modular Arithmetic
 - Euclidean alg: procedure for computing gcd of 2 numbers.




Public Key Encryption

- Traditional key system:
 - Need a key for every pair of users
 - $N*(N-1)/2$ keys, grows exponentially with users
 - Each user has to keep track of many keys
- Public key (asymmetric encryption system)
 - Each user has 2 keys: public and private key
 - May publish the public key freely, inverses
 - $P=D(k_{PRIV}, E(k_{PUB}, P))$
 - Only 2 keys are needed per user
 - B, C, and D can ally encrypt mesgs for A with A's public key



Merkle-Hellman Knapsacks


- Knapsack problem:
 - Set of positive integers
 - Target sum
 - Find subset of integers that equal the target
 - NP-complete alg.
- Encode binary msg as soln to knapsack problem
 - Reduce ciphertext to target sum
 - By adding terms corresponding to 1s in plaintext
 - Convert blocks of plaintext to knapsack sum by adding into sum the terms that match with 1 bits in plaintext.



Superincreasing Knapsack

- Superincreasing sequence:
 - Each integer is greater than sum of all preceding integers
 - $a_k > \sum_{j=1}^{k-1} a_j$
 - Solution of superincreasing knapsack (e.g., simple knapsack) is easy to find
- Convert simple knapsack into Hard knapsack
 - Pick superincreasing sequence S of m integers
 - $S = [s_1, s_2, \dots, s_m]$
 - Choose multiplier w and modulus $n, n > \sum_{j=1}^{m-1} s_j$
 - Choose n to be prime
 - Replace every s_j in simple knapsack with term:
 - $h_j = w * s_j \text{ mod } n$
 - Hard knapsack: $H = [h_1, h_2, \dots, h_m]$


CSE870: Advanced Software Engineering: Cheng 22



Merkle-Hellman (cont'd)

- Merkle-Hellman is Public key cryptosystem
 - Each user has **public key**:
 - Set of integers of a (simple) knapsack problem
 - Each user has **private key**
 - Set of integers for corresponding superincreasing knapsack
- *Contribution: design of technique to convert superincreasing knapsack into a regular one.*
 - Change numbers in nonobvious, reversible way.


CSE870: Advanced Software Engineering: Cheng 23



Merkle-Hellman (cont'd)

- Encryption alg starts with binary message
 - $P = [p_1, p_2, \dots, p_k]$
- Divide message into blocks of m bits,
 - $P_0 = [p_1, p_2, \dots, p_m], P_1 = [p_1, p_2, \dots, p_{2m}]$,
 - Value of m is number of terms in simple or hard knapsack
- Encipherment of message P is sequence of targets
 - Each target is sum of some of the terms of the hard knapsack H
 - Terms selected correspond to 1 bits in P_i ,
 - P_i serves as selection vector for elts of H
 - Each term of ciphertext is $P_i * H$


CSE870: Advanced Software Engineering: Cheng 24



Merkle-Hellman (cont'd)

- Decryption:
 - Legitimate recipient knows simple knapsack and values of w and n
 - $H = w * S \text{ mod } n$
 - $C = H * P - w * S * P \text{ mod } n$
- To decipher, multiply C by w_{-1}
 - $w_{-1} * C = w_{-1} * H * P = w_{-1} * w * S * P = S * P \text{ mod } n$
- Weaknesses:
 - How easy is it to determine w or n from H ?

CSE870: Advanced Software Engineering: Cheng 25




Example

- $S = [1,2,4,9]$; $H = [15,13,9,16]$,
- $w = 15, n = 17, m = 4$; $h_i = w * s_i \text{ mod } n$

- $P = 0100101110100101$
- Encode with H as follows:
 - $P = 0100\ 1011\ 1010\ 0101$
 - $[0,1,0,0] * [15,13,9,16] = 13$
 - $[1,0,1,1] * [15,13,9,16] = 40$
 - $[1,0,1,0] * [15,13,9,16] = 24$
 - $[0,1,0,1] * [15,13,9,16] = 29$
- Encrypted message as integers: 13,40,24,29,
 - Public knapsack $H = [15,13,9,16]$


CSE870: Advanced Software Engineering: Cheng 26



RSA: Rivest-Shamir-Adelman

- Superficially looks similar to Merkle-Hellman:
- Exploits number theory and finding prime factors of a target:
 - $C = P^e \text{ mod } n$; $P = C^d \text{ mod } n$
- Symmetry in modular arithmetic
 - encryption/decryption are mutual inverses and commutative.
 - $P = C^d \text{ mod } n = (P^e)^d \text{ mod } n = (P^{ed}) \text{ mod } n$
- Choosing keys: (e, n) and (d, n)
 - Select value for n
 - should be quite large: a product of two large primes p and q (100 digits ea)
 - Select value for e : relatively prime to $(p-1) * (q-1)$
 - E has no common factors with above product.
 - Choose e as prime larger than both $(p-1)$ and $(q-1)$
 - Select value for d : $e * d \equiv 1 \text{ mod } (p-1) * (q-1)$
- How to use: user distributes e and n , keeps d secret
- To encrypt, need to find large prime numbers.

CSE870: Advanced Software Engineering: Cheng 27

 **DES: Data Encryption Standard**

- Developed for US govt for general public use.
- Repeats 16 cycles of substitution and transposition
 - Shannon's theory of information secrecy
 - **Confusion**: info is changed so that output bits have no obvious relation to input bits
 - **Diffusion**: spread the effect of one plaintext bits to other ciphertext bits.
 - Splits data block into 2 pieces:
 - Scrambles each half independently
 - Combines key with one half
 - (key is transformed during each cycle)
 - Swap 2 halves
 - Repeat 16 times.

CSE870: Advanced Software Engineering: Cheng 28

