Overview of Security and Symmetric-key Encryption

Dawn Song

dawnsong@cs.berkeley.edu

Outline

- What is security about?
- How to evaluate security of systems?
- Introduction to crypto (I): symmetric key encryption

What is Computer Security about?

- **Computing in the presence of an adversary!**
  - An *adversary* is the security field's defining characteristic
- **Reliability, robustness, and fault tolerance**
  - Dealing with Mother Nature (random failures)
- **Security**
  - Dealing with actions of a knowledgeable attacker dedicated to causing harm
  - Surviving malice, and not just mischance
- **Wherever there is an adversary, there is a computer security problem!**

Computer Security History

- **Early history interwoven with military apps**
  - **First big users of computers**
  - **First to worry seriously about the potential for misuse**
- **Terminology has military connotations:**
  - ***Attacker* who is trying to *attack* computer systems**
  - ***Defenders* working to protect their system from these *threats***

Analyze to Learn!

- **We're going spend a lot of time studying attackers and thinking about how to break into systems**
  - **Why spread knowledge that will help bad guys be more effective?**
- **To protect a system, you have to learn how it can be attacked**
  - **Civil engineers learn what makes bridges fall down so they can build bridges that last**
  - **Software engineering is similar**
- **Security is the same and different!**
  - **Why?**

Challenges in Securing Systems

- **Similar:**
  - **Analyze previous successful attacks**
- **But, deploy a new defense, they respond, you build a better defense, they respond, you…**
  - **Need to find ways to anticipate kinds of attacks**
- **Different:**
  - **Attackers are intelligent (or some of them are)**
  - **Attacks will change and get better with time**
  - **Have to anticipate future attacks**
- **Security is like a game of chess**
  - **Except the attackers often get the last move!**

Need to Secure System before Depolyment

- **A deployed system is very hard to change**
  - **Serious consequences if attackers find a security hole in a widely deployed system**
- **Goal: Predict *in advance* what attackers might do and eliminate all security holes**
- **Reality: Have to think like an attacker**
- **Thinking like an attacker is not always easy**
  - **Can be fun to try to outwit the system**
  - **Or can be disconcerting to think about what could go wrong and who could get hurt**
- **What if you don't anticipate attacks?**
  - **Analog cellular phones in the 80's and 90's**

Real-World Example: Analog Cellular

- **1970's: analog cellular had no security**
  - **Phones transmit ID/billing info in the clear**
  - **Assumption: attackers wouldn't bother to assemble equipment to intercept info…**
- **Attackers built "black boxes" to intercept and clone phones for fraudulent calling**
  - **Where's the best place to intercept?**
  - **Cellular operators completely unprepared**
- **Early 90's, US carriers losing >$1B/yr**
  - **70% of LD cellular calls placed from downtown Oakland on Fri nights fraudulent**
- **Problems: huge capital investment/debt, 5–10 yrs & huge replacement cost**

Lesson Learned

- **Failing to anticipate types of attacks, or underestimating the threat, can be costly**
- **Security design requires studying attacks**
  - **Security experts spend a lot of time trying to come up with new attacks**
  - **Sounds counter-productive (why help the attackers?), but it is better to learn about vulnerabilities before the system is deployed than after**
- **If you know about the possible attacks in advance, you can design a system to resist those attacks**
  - **But, anything else is a toss of the dice…**

A Process for Security Evaluation

- **How to evaluate the security of a system?**
  - **A three-step process**
- **Step I: *security goals***
  - **What properties do we want the system to have, even when it is under attack?**
  - **What are we trying to protect from the attacker?**
  - **Or, to look at it the other way around, what are we trying to prevent?**

Some Common Security Goals

- ***Confidentiality*:**
  - **Private information that we want to keep secret from an adversary (password, bank acct balance, diary entry, …)**
  - **Anything we want to prevent adversary from learning**
- ***Integrity*:**
  - **Want to prevent adversary from tampering with or modifying information**
- ***Availability*:**
  - **System should be operational when needed**
  - **Must prevent adversary from taking the system out of service at inconvenient times**

Example: CS161 Grades Database?

- **One obvious goal is protecting its integrity**
  - **Don't want you to be able to give yourself an A+ merely by tampering with grade database**
- **Federal law and university rules require us to protect its confidentiality**
  - **No one else can learn what grade you are getting**
- **We probably also want some level of availability**
  - **So you can check your grades to date and we can calculate grades at the end of the semester**

Security Goals

- **How to identify security goals?**
  - **Highly application-dependent**
  - **If someone figures out how to violate this goal, would it be a security breach?**
    - **If yes, you've found a security goal!**

Step 2: Threat Model and Assessment

- **What kind of threats might we face?**
- **What kind of capabilities might we expect the adversaries to have?**
- **What are the limits on what the adversary might be able to do to us?**
- **What are their motivations and incentives?**

Step 3: Security Analysis

- **Is there an attack within the threat model that can violate the security goals?**
  - **We'll talk about this a lot in class**

Summary: Security Evaluation

- Step 1: Identify security goals
- Step 2: Perform a threat assessment
- Step 3: Security analysis

16

Administravia

- Staff shortage
  - No reader
  - Pls be considerate of the under-staffed situation
- If you plan to drop the course, pls do so soon
  - We'll try to let seniors on the waitlist in
  - Others can take it next time
- How many have taken 170, 162, 122?
  - Students have diverse background
  - Pls be understanding: no one-size fits all

17

3-min Stretch Break

18

Cryptology

- Cryptology is the study of Cryptography & Cryptanalysis
- Cryptography
  - Literally:
    Crypt: secret, graphia: writing---Cryptography: the study of how to send secret messages
  - Formally:
    The study of mathematical techniques to enforce security properties: Confidentiality, integrity, etc.
- Cryptanalysis is the study of how to break cryptographic systems

Brief History of Cryptography (I)

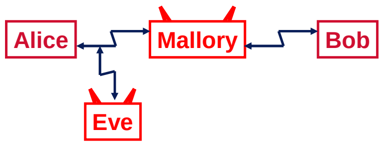
- First phase: manual
  - Caesar cypher (Romans)
    - » Permute the alphabet by shifting each letter forward by a fixed amount
    - » Caesar cipher with a shift by 3:
      - What's the original message for "fubswrjudskb"?
  - Clearly not very secure
- Second phase: mechanical era
  - Enigma machine: a German project to create a mechanical encryption/decryption device
  - British effort to break the code
    - » Important for WWII, estimate shortening war by 1 year

Brief History of Cryptography (II)

- Third phase: Modern Cryptography
  - Relying on mathematics and electronic computers
  - Early roots by Claude Shannon
    - » E.g., One-time pad
  - DES by NIST (1970's)
  - …

Basic Communication Setting

- Introducing security protocol participants
  - Alice (usually the protocol initiator)
  - Bob, Alice's friend
  - Eve the eavesdropper (passive attacker)
  - Mallory the malicious attacker (active attacker)
- Basic setting

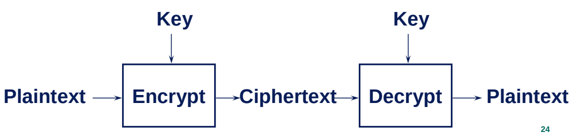


Security Goal

- Confidentiality
  - Attacker cannot learn the content of the message
- Integrity
  - Attacker cannot alter the content of the message

Symmetric-key Model

- Solution I for confidentiality
  - Symmetric key encryption
- Encryption key = decryption key
- Encryption: E_K(plaintext) = ciphertext
- Decryption: D_K(ciphertext) = plaintext
- We write $\{\text{plaintext}\}_K$ for E_K(plaintext)



Threat Model

- **Known ciphertext (ciphertext only)**
  - **Attacker only has a copy of some ciphertext**
- **Known plaintext**
  - **Attacker obtains ciphertext and corresponding plaintext**
- **Chosen plaintext attack (CPA)**
  - **Attacker can choose plaintext that is going to be encrypted and obtains ciphertext**
- **Chosen Ciphertext attack (CCA)**
  - **In addition to chosen plaintext attack, attacker can choose ciphertext and obtains corresponding plaintext**

One-time Pad

- **Alice & Bob share an n-bit secret key K = K1…Kn, where bits K1,…,Kn chosen randomly**
- **Alice wishes to send n-bit msg M = M1…Mn**
- **Desired properties of the encryption scheme:**
  - **Can encrypt: map M to C = C1…Cn**
  - **Given knowledge of K, easy to decrypt: get M from C**
  - **Eve, who doesn't know K, should learn no info about M**
- **Encryption scheme: $C = M \oplus K$**
  - **$C_j = M_j \oplus K_j$**

XOR Properties

- **XOR truth table**

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

- **Some XOR properties**
  - **$a \oplus a = 0$**
  - **$a \oplus b \oplus b = a$**

How Secure is One-time Pad?

- What may Eve learn about M by seeing C?
- What if Eve knew something about M apriori?
  - Does Eve learn anything in addition?
- One-time pad is secure
  - Eve learns no additional info about M by seeing C
  - No matter what M is, C is a uniformly random n-bit string
- Proof
  - For a given M, any C is possible by picking the unique K: $K = M \oplus C$
  - Each such K is equally likely
  - Thus C is equally likely to be any n-bit string

Advantage of One-time Pad

- No other assumptions required for security
  - Attacker without computation limitation

Disadvantage of One-time Pad

- K needs be the same length as the message & can't be reused
- What happens if reuse K?
  - $C = M \oplus K$
  - $C' = M' \oplus K$
  - Eve learns $M \oplus M'$

Stream Cipher

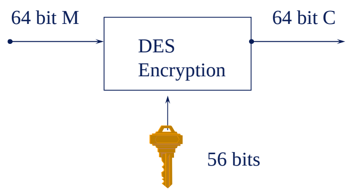
- **Pseudo-random generator**
  - **F(k,i) = r_i**
  - **k is secret**
  - **Attacker cannot distinguish r_1, r_2, … r_i, from a sequence of random numbers**
  - **Stream ciphers can be constructed using block ciphers**
    - » **See later**
- **Encrypt using stream ciphers**
  - **Alice and Bob share k**
  - **Alice wishes to send n-bit msg M = M1…Mn**
  - **Ci = Mi ⊕ F(k,i)**
  - **Practical "one-time pad"**

Block Cipher

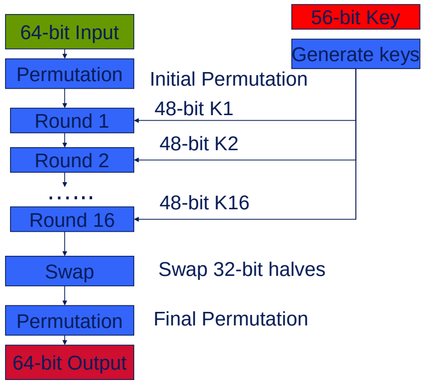
- **Alice & Bob share a k-bit random key K**
- **Encrypt an n-bit msg M into n-bit ciphertext C**
- **Encryption function E:**
  - **C = E(K, M)**
- **Decryption function D:**
  - **M = D(K, C)**

DES

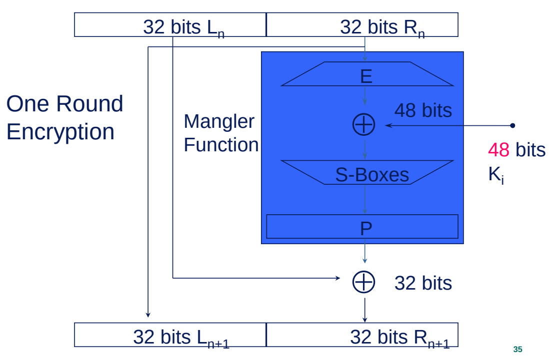
- **Data Encryption Standard (DES)**
  - **An example of a block cipher**
  - **Designed by IBM in 1974 responding to NIST request**
  - **Standardized in 1979**
- **Designed for fast VLSI implementation**
- **Key length 56, block length 64**



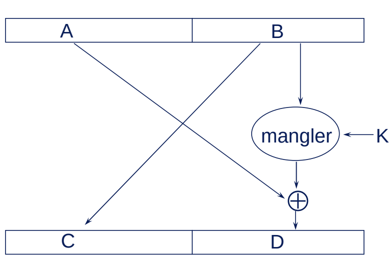
DES Overview



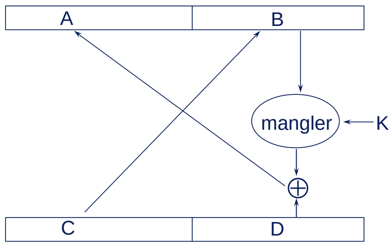
A DES Round



Feistel Cipher Encryption



Feistel Cipher Decryption



37

Why Feistel?

- So mangler function f doesn't need to be reversible
  - enc(A,B): C=B, D=A ⊕ f(B)
  - dec(C,D): B=C,
    A=D ⊕ f(C), because A ⊕f(B) ⊕f(B) = A
- DES is Feistel

38

How Secure is DES?

- Best practical attack known is exhaustive key search
  - 2^{55} (due to symmetry in key structure)
- 1977: Diffie & Hellman: $20,000,000 machine that breaks DES key in 1 day
- 1993: Wiener: $100,000 machine that breaks DES key in 1.5 days
- 1998: EFF's DES Cracker
  - EFF spent $250,000 to build it
  - Tests $88*10^9$ keys per second
  - Solved DES Challenge II-2 in 56 hours
- 1999: DES Cracker + distributed.net (100,000 computers)
  - Tests $254*10^9$ keys per second
  - Solved DES Challenge III in 22 hours

39

Advanced Encryption Standard AES

- 1998 NIST announced a competition for a new cipher
  - DES block length is too short
- Winning cipher was Rijndael (pronounced Rhine-doll)
  - Belgian designers: Joan Daemen & Vincent Rijmen
  - Adopted by NIST as Advanced Encryption Standard (AES), Nov 2001
- Officially adopted for US government work, but voluntarily adopted by private sector
- Block length 128, Key size: 128, 192, or 256
- AES is not Feistel
  - All functions are reversible
- High-speed cipher
  - About 16 clock cycles/byte on modern 32-bit CPUs
  - That's 200 MByte/s on a 3.2 GHz P4!