Computer Security

"I'm sorry, ma'am, but you still have too much saliva in there. You need to spit in the bowl again."

Last time

- Multilevel and multilateral security
- Security policies
- Confidentiality Policies
  - The Bell-LaPadula Model
- Integrity Policies
  - The Biba Integrity Model
- Hybrid Policies
  - The Chinese Wall Model

Security in the Course

- Lectures
  - *Introduction*
  - *Threat analysis*
  - *Introduction to access control matrix*
  - *Security policies*
  - **Cryptography**
  - **Authentication**
  - **Key management**
  - Design principles
  - Access control mechanisms
  - Assurance
  - The future
- Literature

Today

- Cryptography
- Authentication
- Key Management
  - KDC
  - Symmetric keys
  - Asymmetric keys
  - PKI
- Security Protocols
  - Kerberos
  - (SSL/TLS)

Cryptography

Cryptography can be used to provide:

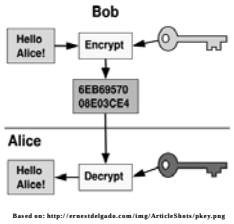
1. Confidentiality and integrity
2. Authentication of the communicators
3. Digital Signatures

Cryptography – What is it?

- A collection of complicated math
  - If you intend to use cryptography in a new way (or suggest a new technique) – study a lot!
  - SHA-3 Competition ends 31:st of October
- A core technology in cyberspace
- No the answer of any security problem
- But helps out

Cryptographic system

- Five parts
- If $K_1 == K_2$
  - Symmetric cryptography
- If $K_1 \neq K_2$
  - Asymmetric cryptography
- Stream or block
- Crypto analysis
- Digital signatures
- Hash functions
- Random number generation



Based on: http://ernestdelgado.com/img/ArticleShots/pkey.png

Different uses

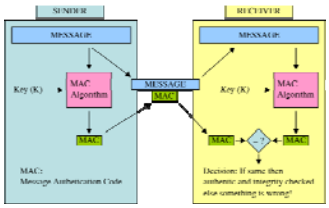
- How you use an algorithm is as important as what algorithm you use
  - *Electronic Code Book*
    - Each block is independent
  - *Cipher Block Chaining*
    - XOR, initialization vector (IV)
  - *Output Feedback*
    - Repeated encryption of IV → key stream

Different uses (2)

- *Counter Encryption*
  - K_i = {IV + i}K
  - Lower cost than *Output Feedback*
  - Less sensitive to birthday attacks (key length: 2^n)
  - Calculate *i*th bit without generating 0…*i*-1
- *Cipher Feedback*, CFB
  - Self-synchronizing *stream cipher*
  - One of the *output*-bits is added to the next clear text bit

Different uses (3)

- *Message Authentication Code*, MAC
  - Use CBC, but only save the last *output*-block = MAC
  - Gives integrity and authentication
  - Requires secure channel



Cryptography in context

- Why is cryptography not the answer to everything?
- Cryptography is math, and math is theory and logic
- The real world is not logical
  - Rules are not obeyed by software, systems or human beings

When is it secure?

- A cryptographic system is said to be computationally secure if one or both conditions are true:
  - The cost of breaking the encryption is greater than the value of the protected data
  - The time to crack the encryption is longer than the life time of the protected data

Secure key length

- The key length is most of the time not related to the security of the system
  - Just make sure it is long enough
- Two problems
  - Quality of the key
  - Quality of the algorithm
- *Entropy*

Secure key length(2)

- First problem: Source of keys
  - Random number generators are not perfect
  - Password entropy
  - Dictionary attack
  - Protecting the key with a password?
- Second problem: Quality of algorithm
  - Stick with the established technologies

Choice of algorithm

- Hard, there is no absolute truth
- Only because you haven't cracked the algorithm, doesn't make it secure
- Anyone who comes up with a new cryptographic algorithm is either a genius or an idiot
- This doesn't mean that everything new is bad, only that everything new is suspicious

Is cryptography practical?

- Must be efficient for the "good guys"
- The (computational) cost to protect something is linear to the key length
- The cost to break something is exponential to the key length

=> Increased computational speed is profitable for the defenders

Authentication

- User identification
  - Something you know
  - Something you have
  - Something you are
- Even safer
  - Combine the above
- Also usable
  - Where you are

Safer

Something you know

- This usually means passwords
  - Security is often based on this
  - There tend to be a lot of passwords...
- Psychological problems
- Social Engineering
- Operational issues

Table 1. Frequency of passwords / PINs, 15 participants

Area	Frequency	Average
Homepages, game logins, computer applications	54	3.60
Email	22	1.47
Computer login	16	1.07
ATM / credit card	15	1.00
Mobile PIN	12	0.80
Online bank device	12	0.80
Alarm or access code	7	0.47
Gas card	6	0.40
Grocery store card	5	0.33
Copy code	2	0.13
Total:	151	10.07

Passwords

- System related
  - If a manner to use passwords is OK, depends on what kind of attack it is supposed to protect against
    - A specific account in a system
    - Any account in a system
    - Any account in any system
    - DoS
  - Multilateral security
  - Can users be taught and be disciplined?
  - Password reuse?

Password attacks

- Shoulder peeking
- Eavesdropping
- Fake log-on application
- Logs
- Theft of the password database
- On-line guessing
- Off-line guessing

Password guessing

- "Humans are incapable of securely storing high-quality cryptographic keys, and they have unacceptable speed and accuracy when performing cryptographic operations. They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed, but they are sufficiently pervasive that we must design our protocols around their limitations"
  - Network Security: Private Communication in a Public World

Something you have

- Passive
  - Regular key
  - Magnetic card
- Smart cards
  - PIN activated memory
  - Special purpose reader
    - Encrypted cards
    - The secret never has to leave the card

Something you are

- Biometrics
  - Signature verifier
  - Face scanner
  - Fingerprint reader
  - Eye scanner
  - Voice recognition

Biometrics

- Problems
  - *Noise, collusion, false repudiation*, statistics, individual differences, religion, …
- Limitations
  - Expensive
  - Not appreciated by users
  - Not usable for network authentication
- Most suited as complementary mechanisms (often manned) due to assumptions
- Useful as a discouragement

Logging in without password

- How to log in without sending the password
- On the whiteboard…

Key distribution

- What if there is millions of users and thousands of servers
- n^2 symmetric keys
- Better to use a centralized service
  - KDC - Key Distribution Center
  - Everyone knows the key of the KDC
  - KDC knows everybody
  - KDC supplies a key to each pair that wants to communicate

Key distribution - KDC realms

- KDCs scales to hundreds of users, not millions
- There is no common entity trusted by everybody
- KDCs can be arranged in hierarchies to ensure that the trust is local

Key distribution

- Protocol
  - Symmetric keys
  - Asymmetric keys
  - On the whiteboard…

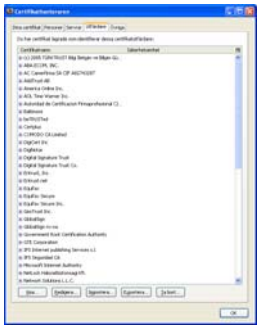
Digital Certificates

- Certification Authority (CA) signs certificates
- Certificate = a signed message saying “I, the CA, guarantee that 123OST is Daniels public key”
- If everyone has a certificate, the corresponding private key and the public key of a CA, authentication is possible

CA

- What is a CA?
  - A “trusted” third part
  - This could be governmental or financial institutions, or specialized companies such as VeriSign
- Important that users acquire the public key of the CA in a secure manner
- Chains of CAs
  - PKI – Public Key Infrastructure

Whom do you trust?



Contents of certificates

Subject	Distinguished Name, Public Key
Issuer	Distinguished Name, Signature
Period of validity	Not Before Date, Not After Date
Administrative information	Version, Serial Number
Extended Information	

- (Above list is simplified)
- All certificates has a period of validity
- Each CA has a revocation list

PKI - Public Key Infrastructure

- Public (Key Infrastructure) or (Public Key) Infrastructure
- Problem
  - Revocation?
  - Name
  - Can you trust all embedded certificates?
  - How does the root-CA obtain its keys?
  - Who generates new keys and how are they propagated?
    - Server to Client
    - Client to Server
- Solves some problems, but often impractical

Schneier on PKI

- Secrets and Lies, p239
  - "As it is used, with the average user not bothering to verify the certificates and no revocation mechanism, SSL is just simply a (very slow) Diffie-Hellman key-exchange method. Digital certificates provide no actual security for electronic commerce; it's a complete sham."

Network security - Levels

HTTP	FTP	SMTP
TCP		
IP/IPSec		

(a) Network level

HTTP	FTP	SMTP
SSL or TLS		
TCP		
IP		

(b) Transport level

	S/MIME	PGP	SET
Kerberos	SMTP		HTTP
UDP	TCP		
IP			

(c) Application level

Kerberos

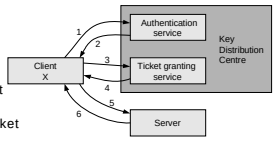
- Kerberos is system for identification
- Based on Needham-Schroeders key distribution for symmetric keys
- Created at MIT in the 80's
  - web.mit.edu/kerberos/www
- Open source
- Used in many commercial products

Kerberos - Questions

- How can a computer ensure that it is communication with a certain computer?
- How can a computer ensure that it is communicating with a certain user at another computer?
- How does the user know that it is communicating with the correct computer?

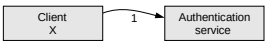
Kerberos - Distributed auth.

Distributed authentication à la Kerberos:



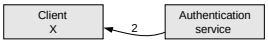
1. Request for TGS ticket
2. Ticket for TGS
3. Request for Server ticket
4. Ticket for Server
5. Request for service
6. Authenticate Server (optional)

Kerberos - Ticket request



- Client sends
  - Who it is (userID)
  - Who it wants to acquire a ticket to
  - Nouce (time stamp and replay protection)

Kerberos - Ticket response

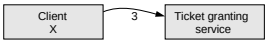


- Session key $K_{C,TGS}$ to communicate with TGS + Nouce, everything encrypted with $K_{C,AS}$, a key the client knows through its password
- A ticket $T_{C,TGS}$ used to prove to TGS that the client is whom it claims to be. Encrypted $K_{AS,TGS}$, a key that AS and TGS knows

Kerberos – A ticket

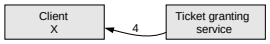
- A ticket contains:
  - The name of the server
  - The name of the client
  - The address of the client
  - A time stamp
  - Period of validity
  - Session key

Kerberos - Server ticket request



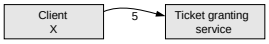
- Client sends
  - An “authenticator” (name, address, time) encrypted with $K_{C,TGS}$
  - Whom it wants to obtain a ticket to (S)
  - The ticket, $T_{C,TGS}$, encrypted with $K_{AS,TGS}$
  - Nonce

Kerberos - Server ticket



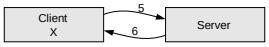
- TGS returns, if the authenticator is correct
  - Session key $K_{C,S}$ to communicate with S + Nonce, everything encrypted with $K_{C,TGS}$
  - A ticket $T_{C,S}$ used to prove to S that the client is whom it claims to be. Encrypted with $K_{TGS,S}$, a key known to TGS and S

Kerberos - Request for service



- Client sends
  - Authenticator encrypted with $K_{C,S}$
  - The ticket, $T_{C,S}$, encrypted with $K_{TGS,S}$
- Now, S can be sure that C is actually C

Kerberos - Mutual authentication



- Client sends
  - Authenticator encrypted with $K_{C,S}$
  - The ticket, $T_{C,S}$, encrypted with $K_{TGS,S}$
  - Time stamp
- The server replies with timestamp+1, encrypted with med $K_{C,S}$
- Now, C can be certain that S actually is S

Kerberos - Advantages

- No passwords are sent!
- Cryptographic protection against spoofing
- Ticket-system – Time limited access
- Time present in messages, defense against replay attacks
- Bilateral authentication

Kerberos - Disadvantages

- Requires that the TGS always is available
- Requires that servers trust each other
- The time system has flaws
- You can crack the password off-line
- Limited scalability
- All-or-nothing solution

Summary

- Cryptography
- Authentication
- Key management
  - KDC
  - Symmetric keys
  - Asymmetric keys
  - PKI
- Security protocol
  - Kerberos
  - (SSL/TLS)

Next Time

- Design principles
- Access control
- Assurance