Introduction to Symmetric and Asymmetric Cryptography

Ali E. Abdallah
Birmingham City University
Email: Ali.Abdallah@bcu.ac.uk
Lectures are part of the project:

**ConSoLiDatE**

Multi-disciplinary Cooperation for Cyber Security, Legal and Digital Forensics Education

Supported by

The Higher Education Academy

December 2014-March 2016
Objectives

- Motivate the needs for cryptography
- Explain the role of cryptography in everyday use

Symmetric Cryptography:
- Describe the main concept
- Analyse some examples
- Discuss strength and limitations

Asymmetric Cryptography
- Describe the main concept
- Analyse some examples
- Discuss strength and limitations

Questions.
Why Use Cryptography?

- To communicate secret information when other people (eavesdroppers) are listening.

- When attacker has access to the raw bits representing the information
  - Mitigation: Data encryption
The Cast of Characters

- **Alice** and **Bob** are “honest” players.

- **Eve** and **Malory** are adversaries (intruders)

- **Eve** “eavesdropper”, is a passive intruder. Sniffs messages at will

- **Malory** is an active “intruder”. Aims to view, alter, delete and inject messages into the network
Confidentiality

- **Problem**: Alice and Bob would like to exchange messages over a public network (such as Internet) in such a way that information contents are not revealed to anyone but the intended recipient.

- **Solution**: Data Encryption + clever Cryptography
How does it work?

- Two functions are needed:

- plaintext
  - encoder
  - ciphertext

- ciphertext
  - decoder
  - plaintext
Example

- **encoder** function is next letter in the alphabet

- **decoder function** is ...

“attack at midnight”

encoder

“buubdl bu njeojhiu”

decoder

“attack at midnight”
Encryption and Decryption

- **Encoding** the contents of the message (the plaintext) in such a way that hides its contents from outsiders is called **encryption**.

- The process of **retrieving** the plaintext from the **cipher-text** is called **decryption**.

- Encryption and decryption usually make use of a **key**, and the coding method is such that decryption can be performed only by knowing the proper key.
The Encryption Process

Aim: to hide a message content by making it unreadable

Plaintext → Encryption Algorithm → Scrambling data → Ciphertext: unreadable version

Material to keep secret:
- Text
- Data
- Audio
- Video
- Graphics

Key
The encryption and decryption functions take a key as an additional input.
Shared Keys

- In a symmetric cryptosystem the encryption key and the decryption key are identical.
- A longer key implies stronger encryption.
Symmetric Cryptosystems

Use the same key (the secret key) to encrypt and decrypt a message
Symmetric Encryption

Shared Key

Sender and recipient Must both know the key. This is a weakness!

Sender: Alice
Recipient: Bob

Material to keep secret: Text Data Audio Video Graphics

Encryption Algorithm

Decryption Algorithm

Confidentiality
Symmetric XOR Cipher

- $P$ encrypts to $C$ with key $K$ and $C$ decrypts $P$ to with same key $K$.

Plain: $P = 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\ 0$

Key: $K = 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0$

Cipher: $C = 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 0$

Confidentiality
One Time Pad

- The perfect encryption
- Pad: perfectly random list of letters
  - Use each letter exactly once to encrypt one letter of message and to decrypt the one letter of message
  - Discard each letter once used (hence, pad)
  - Method: Add the message letter and the key letter Mod 26. This is reversible like XOR.
- The message can never, ever, be found (unless you have the pad).
Example - one time pad

- $P$ encrypts to $C$ with key $K$ and $C$ decrypts $P$ to with same key $K$.
Symmetric Encryption

1. Agree on a Shared Key
   - Alice would like to send a confidential file to Bob
   - PASSWORD IS GREEN!

2. Encrypt using Shared Key
   - CjG5%jARGONS8*%K23##hsgdfey9826.

3. Email file

4. Decrypt using Shared Key
   - CjG5%jARGONS8*%K23##hsgdfey9826.
   - PASSWORD IS GREEN!
Emailing an encrypted message

Alice wants to send a confidential message to Bob.
Symmetric Encryption

1. Agree on a Shared Key
2. Encrypt using Shared Key
3. Email file
4. Decrypt using Shared Key

CREDIT CARD CODE IS 5206
Data Encryption Standard (DES)

- Developed in the 1970s; made a standard by the US government, was adopted by several other governments worldwide and was widely used in the financial industry until 2004.
- Block cipher with 64-bit block size.
- Uses 56-bit keys: Strong enough to keep most random hackers and individuals out, but it is easily breakable with special hardware.
- A variant of DES, Triple-DES or 3DES is based on using DES three times (normally in an encrypt-decrypt-encrypt sequence with three different, unrelated keys). Many people consider Triple-DES to be much safer than plain DES.
Advanced Encryption Standard (AES)

- Current standard.
- DES was perceived as breakable in mid 2000.
- AES was a stronger replacement to DES.
2. **RC2, RC4 and RC5** (RSA Data Security, Inc.)
   - Variable-length keys as long as 2048 bits
   - Algorithms using 40-bits or less are used in browsers to satisfy export constraints
   - The algorithm is very fast. Its security is unknown, but breaking it seems challenging. Because of its speed, it may have uses in certain applications.

3. **IDEA (International Data Encryption Algorithm)**
   - Developed at ETH Zurich in Switzerland.
   - Uses a 128 bit key, and it is generally considered to be very secure.
   - Patented in the United States and in most of the European countries. The patent is held by Ascom-Tech. Non-commercial use of IDEA is free. Commercial licenses can be obtained by contacting idea@ascom.ch.
   - Used in email encryption software such as PGP and RSA
4. **Blowfish**

- Developed by Bruce Schneider.
- Block cipher with 64-bit block size and variable length keys (up to 448 bits). It has gained a fair amount of acceptance in a number of applications. No attacks are known against it.
- Blowfish is used in a number of popular software packages, including Nautilus and PGPfone.

5. **SAFER**

- Developed by J. L. Massey (one of the developers of IDEA). It is claimed to provide secure encryption with fast software implementation even on 8-bit processors.
- Two variants are available, one for 64 bit keys and the other for 128 bit keys. An implementation is in ftp://ftp.funet.fi/pub/crypt/cryptography/symmetric/SAFER.
Limitations

- Parties that have not previously met cannot communicate securely
- Many people need to communicate with a server (many-to-one communications)
  - cannot keep server key secret for long
- Once the secret key is compromised, the security of all subsequent messages is suspect and a new key has to be generated
- Authentication service must know private key
  - privacy implications---someone else knows your key
  - two possible points of attack
  - changing authentication service requires a new key
- Digital signatures are difficult
- Cross-realm authentication
  - accessing services outside the domain or realm of your authentication server is problematic
  - requires agreement and trust between authentication services
  - introduces another potential point of attack
Private or symmetric key systems rely on symmetric encryption algorithms where information encrypted with a key $K$ can only be decrypted with $K$.

Secret key is exchanged via some other secure means (hand-delivery, over secured lines, pre-established convention).

Time to crack known symmetric encryption algorithms

| KEY LENGTH | SPEND $$T$HOUSANDS | SPEND $$MILLIONS | SPEND $100 \text{ MILLION}
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>40 bits</td>
<td>seconds</td>
<td>&lt; 1 second</td>
<td>&lt; .01 second</td>
</tr>
<tr>
<td>56 bits</td>
<td>hours</td>
<td>minutes</td>
<td>1 second</td>
</tr>
<tr>
<td>64 bits</td>
<td>days</td>
<td>hours</td>
<td>minutes</td>
</tr>
<tr>
<td>80 bits</td>
<td>years</td>
<td>months</td>
<td>days</td>
</tr>
<tr>
<td>128 bits</td>
<td>&gt; million years</td>
<td>&gt; million years</td>
<td>&gt; centuries</td>
</tr>
</tbody>
</table>
Symmetric Cryptosystems Problems

- How to transport the secret key from the sender to the recipient securely and in a tamperproof fashion?

- If you could send the secret key securely, then, in theory, you wouldn't need the symmetric cryptosystem in the first place -- because you would simply use that secure channel to send your message.

- Frequently, trusted couriers are used as a solution to this problem.
Asymmetric Cryptosystems

In asymmetric-key cryptography, users do not need to know a symmetric shared key; everyone

• shields a private key
• advertises a public key
Alice and Bob don’t already share a key and can’t meet to do so. How can they make their future communications confidential?

- The main protocol we study is the celebrated Diffie-Hellmann Key Exchange (DHKE) protocol.
Diffie-Hellman Key Exchange Idea

Alice fills up another 1/3 of key using her part (x) and sends the mix to Bob

Bob fills up another 1/3 of key using his part (y) and sends the mix to Alice

Alice completes the key by adding her secret part (x)

Two keys are the same: it doesn't matter if x if filled first or y.

Bob completes the key by adding his secret part (y)
An alternative interpretation

- Alice & Bob each think of a secret color (known only to them)
- They mix their color with yellow (agreed upon openly ahead of time) and exchange.
- They mix their color with what they’ve received.
- Both have the same color but observer cannot duplicate.
Asymmetric Encryption

Sender knows **public key**
Recipient knows **private key**.
Signing and verification

Sender knows **private** key
Recipient knows **public** key.
Properties

- These algorithms are based on computationally intensive problems such as finding the prime factors of large numbers.
  - Longer the length of the key pair, the more time it takes to crack the private key
  - Keys used in today’s internet will take millions of years to crack using today’s technologies
Slow ...

- Public key cryptosystems are slow, really slow!
  - three orders of magnitude (1000 times) slower than AES
  - mainly used as key exchange tool
- Scientists are supposed to be real “smart” and love to solve difficult problems
  - but even they hope to never solve factoring
  - if you can find a quick solution, 
    - fame, dollars and danger lurk!
Problems

Keys are usually very long and encryption is expensive

- RSA encryption is a 1000 times slower than typical symmetric algorithms
- hard to remember secret key - where do you store it?
- typically only used for authentication, then a random key and a symmetric encryption algorithm is used for subsequent communication

Multicast is problematic

- Better to authenticate using public key algorithm, then use random key with symmetric algorithm

How do you know you have the right public key for a principal?

- Public key is usually distributed as a document ``signed'' by a well-known and trusted certification authority (e.g. Verisign). This is called a certificate. How do you determine if signature is up-to-date? What if the key has been compromised?
Analysis

- **Private (Symmetric) key:**
  + encryption is fast
  - identity is not easily portable across authentication services
  - secret key *must* be held by server
  + good for structured, organizational security

- **Public (Asymmetric) key:**
  - encryption is slow
  + identity is inherently portable
  + secret key need not *ever* be revealed
  + provides digital signatures
  + good for individuals in loosely structured networks
Digital Envelopes

1. Agree on a Session Key

2. Use Session Key to Encrypt /Decrypt transmitted messages

SKY IS Blue!
Digital Envelope

- Combination of public-key (asymmetric) cryptography and symmetric systems

- Sender:
  - Generate a secret key at random called the session key (which is discarded after the communication session is done)
  - Encrypt the message using the session key and the symmetric algorithm of your choice
  - Encrypt the session key with the recipient’s public key. This becomes the “digital envelope”
  - Send the encrypted message and the digital envelope to the recipient
Digital Envelope

- **Recipient**
  - Receive the envelope, uses private key to decrypt it recovering the session key.
  - The message is secure since it is encrypted using a symmetric session key that only the sender and recipient know.
  - The session key is also secure since only the recipient can decrypt it.
  - Can even act like a one time pad
Summary

Cryptosystems: Symmetric & Asymmetric
- Symmetric: Use the same key (the secret key) to encrypt and decrypt a message
- Asymmetric: Use one key (the public key) to encrypt a message and a different key (the private key) to decrypt it.

Symmetric Cryptosystems Problems
- How to transport the secret key from the sender to the recipient securely and in a tamperproof fashion? If you could send the secret key securely, then, in theory, you wouldn't need the symmetric cryptosystem in the first place -- because you would simply use that secure channel to send your message.
- Frequently, trusted couriers are used as a solution to this problem.

Modern solutions combine features from both Symmetric & asymmetric cryptosystems.
Questions?
Summary

- Cryptography enables parties to communicate on open networks without fear of being eavesdropped
  - all cryptographic schemes have their limitations
- Symmetric schemes use a common key for encryption and decryption.
- Asymmetric (public key) schemes use a public-private key pair where the public key is used by senders to encrypt and only the recipient with the private key can decrypt the message.
- Trade-offs between symmetric and asymmetric schemes.
- Digest functions (Hash-functions) can be used to maintain integrity of a message and make it tamper-proof.
- Digital envelopes combine the security of asymmetric schemes with the efficiency of symmetric schemes.
- Certification authorities allow authenticated access to public keys.
- A hierarchy of certification authorities (hierarchy of trust) can be used.
- Certification Revocation Lists maintain a list of invalid certificates.
Digital Envelope

- Combination of public-key (asymmetric) cryptography and symmetric systems
- Sender:
  - Generate a secret key at random called the session key (which is discarded after the communication session is done)
  - Encrypt the message using the session key and the symmetric algorithm of your choice
  - Encrypt the session key with the recipient’s public key. This becomes the “digital envelope”
  - Send the encrypted message and the digital envelope to the recipient
Digital Envelope

- **Recipient**
  - Receive the envelope, uses private key to decrypt it recovering the session key.
  - The message is secure since it is encrypted using a symmetric session key that only the sender and recipient know.
  - The session key is also secure since only the recipient can decrypt it.
  - Can even act like a one time pad

*Source: Bob Thibadeau http://dollar.ecom.cmu.edu/sec/lec02.ppt*