Introduction to Cryptology

Eugene Davis
UAH Information Security Club
January 31, 2013

What is Cryptology?

- Cryptology is the study of cryptography and cryptanalysis
  - Cryptography is literally the study of “secret writing”
    - Ciphers (Encryption)
    - Codes
    - Not Steganography
  - Cryptanalysis is the study of how to break cryptograms
- We will be focusing on encryption and only touching on the other areas

What are Ciphers?

- Ciphers are algorithms for performing encryption or decryption
- There are three basic categories of encryption
  1. Symmetric Encryption – a single key is used to encrypt or decrypt data
  2. Asymmetric Encryption – two different keys, one to encrypt and one to decrypt
  3. Cryptographic Hashes – one way functions which are hard to forge

What is Cryptanalysis?

- Cryptanalysis is the process of finding weaknesses in encryption algorithms
- By attacking an encryption algorithm, security researchers can determine some potential weaknesses
- Cryptanalysis often is stumped over the short term, but tends to beat encryption over the long term

A Brief History of Cryptology

Classical Cryptography

- Transposition ciphers and monoalphabetic substitution ciphers have been in use for thousands of years
- Poly-alphabetic ciphers (use multiple cipher alphabets) were developed in the 15th Century
- In the 19th century cryptographers realized that the secrecy of an algorithm does not help its security
  - In more modern terms, security through obscurity doesn't work

A Brief History of Cryptology

The Computer Era

- The Enigma Machine started the use of machines in cryptography
  - The British developed the Colossus as part of their efforts to break the German's ciphers – the first fully electronic, digital and programmable computer
- Developments during and after WWII advanced the state of the art in cryptography
- In 1945 Claude Shannon proved that One-Time Pads were perfectly secure
- Symmetric key cryptography became more fully developed, resulting in the Data Encryption Standard (DES) in 1976

A Brief History of Cryptology

The Computer Era (Cont.)

- In 1976 the Diffe-Hellman protocol opened a new area for cryptography by allowing the exchange of keys over public channel
- In 1978 RSA showed that static keys could be created rather than generating keys for every session
- In 1998 the Electronic Frontier Foundation demonstrated the weakness of DES
- As a result of the weakness of DES, AES became the new standard in 2001

Symmetric Cryptography

Basics

- Symmetric cryptography uses a single key, shared by all parties involved
- Good keys are composed of random data
- Monoalphabetic ciphers use two alphabets, cipher and plaintext
- Polyalphabetic ciphers use a plaintext alphabet and multiple ciphertext alphabets

Symmetric Cryptography

One-Time Pads

- One-Time Pads - XOR a key composed of random data with the message
  - Key cannot repeat
  - Other ciphers imitate this with key expansion
- Perfect Security – provided by OTP, no info is revealed to attacker
- Not a practical for most uses

Symmetric Cryptography
Block Ciphers

- A block cipher operates on blocks of data at a time
- Very common in modern cryptography
- Come in two basic forms
  - Iterated Block Ciphers – a function (round) is applied repeatedly to the blocks to encrypt them
  - Feistel Ciphers – a block is split in half, the round applied to it, then XOR with the other half, then repeated
- In effect, key data and message data are mixed together repeatedly

Symmetric Cryptography

Block Ciphers (Cont.)

- Initialization Vectors (IV) – unique values that "seed" certain modes of operation
- Modes of Operation – the security of a block cipher depends in part on how it is used
  - Electronic Codebook (ECB) – blocks are encrypted independently. Same plaintext results in same ciphertext, so avoid at all cost
  - Cipher Block Chaining (CBC) – plaintext blocks are XORed with preceding cipher block. The IV supplants cipher block for first plaintext block
  - Other less common modes exist as well

Symmetric Cryptography

Stream Ciphers

- Stream Ciphers work on by combining a stream of bits with a pseudo-random keystream
- Types of Stream Ciphers
  - Synchronous Stream Ciphers – a stream of pseudo-random bits is generated independently of the plaintext and XORed with the message
  - Self-Synchronizing Stream Ciphers - uses previous ciphertext digits to compute the keystream
    - A block cipher can use previous cipher blocks to generate a keystream for self-synchronizing

Symmetric Ciphers
DES and Triple-DES

- DES is an outdated block cipher that once was the U.S. government's standard
  - DES is based on a Feistel network
  - Using DES should be avoided at all costs
- Triple-DES uses three keys and applies DES three times time triple the effective key size
  - Encrypt with the following: $E_{k3}(D_{k2}(E_{k1}(m)))$. Decryption is just the reverse.
  - This is secure, but slow. AES should be prefered.

Symmetric Ciphers
AES

- AES is a block cipher.
- Can use key lengths of 128, 192, or 256 bits with a block size of 128 bits
- Government and industry standard
- AES should be the go-to block cipher
- Export of some strengths of AES is restricted with in the U.S.

Asymmetric Encryption

Basics

- Asymmetric Encryption uses two separate keys, a public key and a private key
- The security of asymmetric encryption depends upon computationally intensive operations
- Asymmetric Encryption also allows for signing since a public key only decrypts for the associated private key
- Asymmetric cryptography is usually used to transmit a key for symmetric cryptography

Asymmetric Encryption
Diffie-Hellman

- Alice and Bob wish to exchange a secret message without knowing each other's keys in advance. The value **p** is prime, and **g** is a primitive root mod **p**

  1. Alice has a secret value **a** and public values **p** and **g**
  2. Alice calculates $\mathbf{g^a \bmod p = A}$, sending **A** to Bob
  3. Bob now has **p**, **g**, **B**, and a secret value **b**
  4. Bob calculates $\mathbf{g^b \bmod p = B}$, sending it to Alice
  5. Both Alice and Bob now have the same values: **p**, **g**, **A**, **B**
  6. Alice calculates $\mathbf{B^a \bmod p = s}$, meanwhile Bob calculates $\mathbf{A^b \bmod p = s}$
  7. Since s is the same value for both, Alice and Bob now have a shared secret! This can be used as a key for a symmetric encryption algorithm.

- The graphic demonstrates how this works: the common paint is the values p and g, the secret colors are a and b, the public transport paints are A and B, and the Common secret is the resulting value s

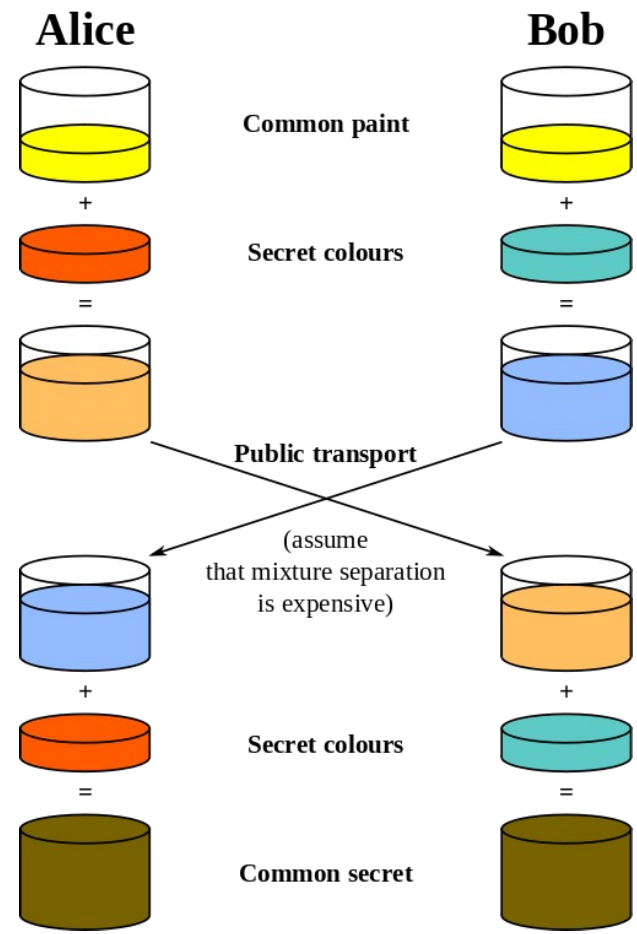

Image from Wikipedia's Diffie-Hellman page

Asymmetric Encryption
RSA

- Keys are generated by combining prime numbers
- Performing encryption/decryption
  - Alice transmits her public key, (n,e) to Bob
  - Bob encrypts a message "m"
    $c = m^e \bmod n$
  - Alice decrypts "c" by doing:
    $m = c^d \bmod n$
- Security depends on the hardness of factoring to find the primes used to make the key
- There are many other asymmetric algorithms that we have not covered

Public Key Infrastructure (PKI)

- A PKI provides a mechanism to distribute public keys
  - A PKI can provide identification of users or servers, as with SSL certificates
  - The fundamental challenge is ensuring key distribution is not co-opted by and attacker
- Probably the most used PKI is the one which handles SSL certificates
  - Unfortunately, mistakes on the part of an issuing authority undermine the trust of the entire system

Hashes

- Hash functions are algorithms that take an arbitrary data input and return a fixed-sized bit string
  - Properties:
    - Pre-image resistance – given h it should be difficult to find a message m such that h = hash(m)
    - Second pre-image resistance – given an input m_1 it should be different to find a another input m_2 where $m_1 \neq m_2$ but $hash(m_1) = hash(m_2)$
    - Collision resistance – it should be difficult to find two messages $m_1 \neq m_2$ such that $hash(m_1) = hash(m_2)$.
- Hashes can be used to verify that data hasn't been changed

Using Encryption
As End Users

- Most of us see encryption on a daily basis
  - Websites – e-Commerce, banking, email, social networking all depend upon SSL (secure socket layer) and its associated PKI
  - Hard drive encryption – products like Bitlocker and Truecrypt let you protect your data
  - Cell Phones – encrypt the texts and calls you make as they go to the towers
- As an end user you must evaluate the security of the products and ensure that you protect your keys and passwords

Using Encryption
As Developers

- As more applications move to the web, cryptography becomes more important
- Many software companies encrypt data and software to protect their intellectual property
- As a developer you must understand best practices for implementing encryption
  - Never use an algorithm you invented yourself! It is probably just XOR encryption anyway...
  - Use existing libraries and be cautious even then
  - This talk does not qualify you to deal with encryption

And Another Thing

Encryption allows you to convert a secure channel problem into a key management problem, much like a lever converts distance into force.
Encryption is not a magic wand which secures systems, but a useful tool which comes with trade-offs.

Conclusions

- Cryptology lets us protect our data and test our protections
- AES is the symmetric cipher to use, but not in ECB mode
- Asymmetric cryptography avoids the cost of exchanging keys in person
- Hashes can verify data is unchanged
- Encryption is important in many areas of modern computing
- Encryption is not a one-size fits all solution for security, each system needs to be evaluated and have its solutions for security

References

- The Code Breakers by David Khan
- Coursera's Cryptography Course (Lecturer Dan Boneh) https://www.coursera.org/course/crypto
- Computer Security by Dieter Gollman
- Applied Cryptography by Bruce Schneier

License