Introducing Applied Cryptography

The use of cryptography to build secure systems
- Symmetric encryption gives us tools
  - hash functions and message digests
  - encryption and decryption ciphers
- ...but suffer two fundamental problems
  - the key distribution problem
  - the key management problem

Confidentiality
- limitation of access to information to authorized parties
  - symmetric encryption

Integrity
- prevention of unauthorized alteration of information
  - message authentication codes

Authentication
- identity and information origin verification
  - challenge-response

Non-repudiation
- inability to deny previous commitments or actions

Point-to-Point vs End-to-End Security

Asymmetric Encryption

- Commonly referred to as public key encryption
- Messages encrypted using key pairs (public & private)
- One key used for encryption, the other for decryption
- Public key distributed as much as possible
- Private key kept secret
- Versatile and more secure than symmetric algorithms
- Comparatively slow
Asymmetric Encryption

- Based on mathematical one-way functions
  - (relatively) easy to compute with known parameters
  - computationally infeasible to recover parameters
- Three main classes of problems
  - integer (prime) factorization
  - discrete logarithm
  - elliptic curve
- Comparatively slow, often combined with other tools
  - exchange of session keys in transport protocols
  - sign hashes of messages rather than messages

Asymmetric Encryption Operations

- Encryption of messages
  - encrypt message using recipient's public key
- Decryption of messages
  - decrypt message using recipient's private key
- Signing of messages
  - encrypt (hash of) message using sender's private key
- Combinations: signing and encryption of messages
  - sign message using sender's private key
  - encrypt message using receiver's public key

Asymmetric Encryption Trust

- As long as the keys can be trusted
  - messages can be kept secret (only receiver can decrypt)
  - senders and receivers can be authenticated
  - message content can be trusted
  - (acknowledged) messages cannot be repudiated
- Can be used to address the key distribution problem
  (does not solve the key management problem)

Public Key Infrastructures (PKI)

- Virtual infrastructures consisting of clients, servers and Certification Authorities (CAs)
- CAs are trusted third parties which provide signed certificates (i.e., signs public keys)
- CA certificates are distributed in browsers and similar tools (trusted and considered known by all)
- Since CA public keys are known, (signed) certificates can be validated offline (without connecting to the CA)
- Secure connections are established between parties using certificates and encryption algorithms
- Network traffic tunneled through encrypted channels

Certificates

- Certificate = signed tuple of public key & identity
- Certificates can be self-signed or signed by others
- Certificates signed by trusted parties can be used for encryption, authentication and message integrity checks
- Self-signed certificates can only be used for encryption (they suffer from the key distribution problem)

Certification Authorities

- Trusted third party organizations
- Provides certificate issuing services
- CA public keys assumed to be well known (distributed with, e.g., browsers and operating systems)
- Commercial operators (2014)
  - Symantec (owns VeriSign, Thawte, Geotrust)
  - Comodo SSL
  - Go Daddy
  - GlobalSign
Secure Socket Layer (SSL)

- A protocol for establishing secure connections using certificates and cryptography algorithms
- Transport Level Security (TLS) ≈ SSL v3.0
- Clients use server certificate to authenticate server
- Servers use client certificate to authenticate client (optional)
- Once identities are established, symmetric session keys are exchanged
- Keystores to manage certificates and keys

Application of SSL

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

Credentials Delegation

- Temporary session / delegation keys
  - generate new key pair
  - create a certificate with restricted access rights and lifetime
  - sign new certificate (public key) using old private key
- Use new certificate / key pair instead of original
- Simplifies (but does not solve) key management
  - limits exposure of the original key pair
- Improves security in distributed systems
  - limits lifetimes of authentication sessions
  - limits attack surface for broken keys
  - useful for cross-domain authentication (e.g., single-sign on systems)
- Used in distributed computing middlewares
  - The Globus Toolkit

Cryptanalysis

- (The science of) Recovering plaintext from ciphertext without knowledge of the cipher key
- attack: attempted cryptanalysis
- break: anything faster than brute force attacks
- Attack targets
  - plaintexts
  - ciphertexts
  - plaintexts-ciphertext combinations
  - protocols
  - systems
  - users

Cryptanalysis Attacks

- Ciphertext-only attack
  - recover plaintexts (or key) using ciphertexts encrypted with the same key
- Known-plaintext attack
  - recover key using ciphertexts and corresponding plaintexts
- Chosen-plaintext attack
  - recover key using ciphertexts from selected plaintexts
- Adaptive-chosen-plaintext attack
  - chosen-plaintext attack where plaintexts are updated during analysis

- Chosen-ciphertext attack
  - recover key using ciphertexts from selected ciphertexts (primarily used to attack public key systems)
- Chosen-key attack
  - analyse relationship between (unknown) keys
- Rubber-hose cryptanalysis
  - recover key by attacking (bribe, trick, extort, etc.) users
- Combinations
**Brute-Force Attacks**
- Exhaustive key search ("try every combination")
- Can be used to attack (almost) any encrypted data (exception: information-theoretically secure ciphers)
- Feasibility primarily determined by two factors: key length and algorithm complexity
  - key length determines search space size
  - algorithm complexity determines cost of test
- Cost of test grows linearly with key length
  - 512 bit key $2^1$ (twice) as complex as 256 bit key
- Search space grows exponentially with key length
  - 512 bit key $2^{256}$ times as complex as 256 bit key

**Hash Collisions**
- Checksums and hashes are designed to minimize the likelihood of collisions
  - approaching rectangular distribution (uniform probability)
- Pigeonhole principle
  - $n$ items stored in $m$ containers: $n > m$ gives at least one container contains > 1 item
  - not possible to uniquely hash any large data set to a smaller representation
- Collisions are (unlikely but) unavoidable

**Birthday Paradox**
- The probability that randomly selected independent values in a finite set will have the same value grows much faster than the probability of selecting a specific value from the set
  - 30 people $\Rightarrow$ 7.9% chance for a specific birthday
  - 30 people $\Rightarrow$ 70% chance for the same birthday
  - the likelihood of a hash collision is around 50% when the number of data items hashed approaches the square root of the number of unique data representations

**Birthday Attacks**
- Force hash collisions using the (mathematics of) the birthday paradox
  - evaluate value permutations that produce a certain hash
  - non-rectangular value distributions reduce search space
- Spoofing signatures
  - A sends a signed message to B (hash used in signature)
  - C intercepts the message in transit
  - C finds a permutation of a different (altered) message with the same hash
  - C sends the altered message to B (pretending to be A)
- Naive brute force attack requires (at most) $2^{256}$ tests
- Hash attacks require (at most) $2^{256}$ tests
Approximate Hash Strength

- Approximate hash strength calculations
  - assume rectangular distribution and probabilities ≤ 0.5
  - m: #possible hash values
  - n: #non-colliding data values
  - \( P(n) \): probability of a value
  - \( P(n) \approx n^{1/m} \)
  - \( \frac{m}{n} \approx \frac{P(\infty)}{P(n)} \)

- Given a 32-bit hash, what is the maximum number of values we can store if we want to keep the chance of a hash collision ≤ one in a million (\( P \approx 2^{-20} \))?
  - \( n \approx \sqrt{\frac{m}{2^{32}}} \approx 2^{10} \)
  - \( P(\infty) \approx 2^{-32} \)
  - approximate #atoms in our galaxy: \( 2^{226} \)

SHA-512 (512-bit hash, 2^{512} possible values), \( P \approx 2^{-20} \)
- \( n \approx \sqrt[3]{\frac{m}{2^{512}}} \approx 2^{166} \)
- approximate #atoms in our galaxy: \( 2^{226} \)

Hash Attack Example: Passwords

- 2013: Ars Technica gave 3 experts a file with 16k passwords (MD5 hash)
  - 90% recovered in 20 hours
  - commodity system with AMD Radeon 7970 graphics card

- Attack surface
  - broken hash (MD5)
    - fast hashes (MD5) \( \rightarrow \) 8k tests/s \( \approx \) 20 hours
    - slow hashes (SHA512) \( \rightarrow \) 2k tests/s \( \approx \) 2.5 years
  - policy limitations
    - reduce search space based on patterns (8 chars, 3 digits)
    - dictionaries
    - reduce search space based on commonalities

http://arstechnica.com/security/2013/05/how-crackers-make-minced-meat-cut-off-your-passwords/

Rainbow Tables

- Precomputed tables for reversing cryptographic hashes
- Employs space-time trade-off
  - tables expensive to generate
  - lookup faster than computation
  - data set value space comparatively small
- Search space database fits in RAM (or on disc)
  - 8 chars alphanumeric MD5 = 576GB
- Can be avoided using key derivation functions with salts
  - salt = random data used as additional input to hash
  - hash(hash(password),salt)

Password Strength

- Password attacks are heuristics brute-force based
  - dictionary and combinator attacks
  - targeting recovered password files or online systems
- Complexity = \( \#\text{values}\times\text{functions} \)
  - 8 lowercase letters: \( 26^8 \approx 2^{38} \)
  - 8 upper/lowercase letters or numbers: \( 62^8 \approx 2^{46} \)
  - 4 words from dictionary of 1000: \( 1000^4 \approx 2^{30} \)
  - 4 words from dictionary of 100000: \( 100000^4 \approx 2^{67} \)
- Probability
  - assuming rectangular value distributions
  - difficulty to guess vs difficulty to use (remember)

Password Selection

- Not enough information to decipher messages without key
- Perfect security (special case)
  - an encryption algorithm is perfectly secure if a ciphertext produced using it provides no information about the plaintext without knowledge of the key.
- One-time pad (1882) (bit / char level)
  - stream cipher with perfectly random (secret) keystream
  - lifetime limited by the amount of tokens on the pad
- Even with perfect ciphers, key problems remain
  - key distribution
  - key management

Information-Theoretic Security

https://xkcd.com/936/
WS-Security

- A standard for secure SOAP point-to-point messaging
- XML Security protects XML
  - XML Encryption
  - XML Signature
- WS-Security protects Web Services by securing SOAP
  - links XML-Security with tokens
  (X.509, Kerberos, SAML)
  - provides SOAP bindings and defines SOAP security headers
- Extensible
- Customizable

Cryptanalysis

Today
Introduction
Asymmetric
Encryption
Public Key Infrastructure (PKI)

Keystores: encrypted databases for keys and certificates
- Usually stored in a single file called .keystore
- Applications must provide database decryption key (username & password) to access keystore content
- Keystores only containing public keys and certificates are commonly referred to as truststores
- Keystores can be shared between SSL applications (usually only done for truststores)

Keystores can be shared between SSL applications
- Keystores only containing public keys and certificates are commonly referred to as truststores
- Keystores can be shared between SSL applications (usually only done for truststores)

WS-Security Message Structure

- WS-Security protects Web Services by securing SOAP
- links XML-Security with tokens
  (X.509, Kerberos, SAML)
  - provides SOAP bindings and defines SOAP security headers

Transport-Level Security

- Provides a secure channel for SOAP message transport via HTTP over TLS (HTTPS)
  - all communication is encrypted
- Hybrid encryption scheme (good performance)
- Point-to-point security
  - protection over a single network connection
  - does not work across intermediaries

Apache Rampart

- WS-Security implementation for Axis2
- Supported specs (Rampart 1.4)
  - WS-Security
  - WS-SecureConversation
  - WS-Trust (partial support)
- No need to modify service code
  - security callbacks in message handler chains
  - configured in services.xml (service-side) and in client-side configuration file
  - advanced stub generators can generate security code
- Installation guide and examples on course web page
Today
Introduction
Asymmetric Encryption
Public Key Infrastructure (PKI)
Cryptanalysis
An Example: WS-Security

Rampart Message Processing

Rampart Sample 3

Rampart Sample 9