Security +
Encryption Standards

Author: Joseph Lee
Email: joseph@ripplesoftware.ca
Mobile: 778-725-3206

General Concepts

Forward secrecy / perfect forward secrecy
- Using a key exchange to **provide a new key for each session** provides improved forward secrecy because if keys are found out by an attacker, **past data cannot be compromised with the keys**

Confusion
- Cipher-text is **significantly different** than the **original plaintext data**
- The property of confusion hides the relationship between the cipher-text and the key

Diffusion
- Is the principle that **small changes in message** plaintext **results in large changes in the cipher-text**
- The idea of diffusion is to **hide the relationship between the cipher-text and the plaintext**

Secret-algorithm
- A proprietary algorithm that is not publicly disclosed
- This is discouraged because it cannot be reviewed

Weak / depreciated algorithms
- An algorithm that can be **easily cracked** or defeated by an attacker

High-resiliency
- Refers to the strength of the encryption key if an attacker discovers part of the key

Data-in-transit
- Data sent over a network

Data-at-rest
- Data stored on a medium

Data-in-use
- Data being used by an application / computer system

Out-of-band KEX

- Using a medium/channel for key-exchange other than the medium the data transfer is taking place (phone, email, snail mail, web-portal)

In-band KEX
- Using the same medium / channel for key-exchange that the data transfer is taking place
- DH is an example

Integrity
- Ability to determine the message has not been altered
- Hashing algorithms manage

Authenticity
- Ability to determine that the sender is the intended sender of the message

Confidentiality
- Ability to ensure ensure the message cannot be read by others

Digital Signature
- Provides **authentication**, **integrity**, **non-repudiation**
- A digital signature is created by encrypting the message hash with the sender's private key
- Digital signatures are also used to authenticate software applications and updates
- When digital signatures are used to authenticate servers and clients to each other, the server and client public keys must not be shared with anyone else
- Unique keys must be generated for each client

Message digest
- One-way cryptographic hash of a message

Codes
- Simple secret codes such as slang words

Transposition
- Changing the order of the characters

Obfuscation algorithms
- Just another name for encryption that is used for confidentiality

SDP Software Defined Perimeter
- Also called a **Black Cloud**, is an approach to computer security which evolved from the work done at the Defence Information Systems Agency (DISA) under the **Global Information Grid (GIG) Black Core** Network initiative around 2007
- Connectivity in a Software Defined Perimeter is based on a need-to-know model, in which device posture and identity are verified before access to

application infrastructure is granted
- See: https://en.wikipedia.org/wiki/Software_Defined_Perimeter

Initialization vector (IV) / salt
- A nonce used to pad the message data
- Deters brute force attempts to decrypt the message since is has been padded with an IV (salt) and is therefore less predictable

CRC Cyclic Redundancy Check
- An **error-detecting** code commonly used in **digital networks** and **storage devices** to **detect accidental** changes to raw data
- Called CRC because the check (data verification) value is a **redundancy** (it expands the message without adding information
- Detecting errors in the integrity of data
- CRC was invented **in 1961** and is used in ethernet as well as **across all communication** channels and **many protocols**

Pre-shared Key (PSK)
- An encryption key or key pair that has already been shared and therefore does not need to be shared at runtime / during connection initialization

PMK Pairwise master key
- After the **PSK** or **802.1X authentication**, a shared secret key is generated, called the **Pairwise Master Key (PMK)**
- The **PMK** is derived from a password that is put through **PBKDF2-SHA1** as the cryptographic hash function
- In a pre-shared-key network (such as WPA2 personal), the PMK is actually the PSK

Steganography
- Hide information within other information (such as data in an image file)

PKCS Public Key Cryptography Standards
- Group of public-key cryptography standards devised and published by **RSA Security LLC**, starting in the early 1990s

OTP One time pad / Vernam cipher
- Is an encryption technique that cannot be cracked, but requires the use of a **one-time pre-shared key the same** size as, or longer than, the message being sent

Key Management
- The ability of an organization to manage cryptographic keys

Decentralized / Distributed key management
- Users create their own keys for use
- Bitcoin allows any address to have its own keys

Centralized key management

- All valid keys are managed by a central server

Multilevel key management

- Used in systems with conflicting security requirements / different levels of security

Key Escrow

- Is the process of keeping a private key in a safe stored environment. If the original key is lost, the organization can retrieve the copy of the key
- A **key recovery agent** is a designated individual who can recover keys from an escrow

Basic Text Ciphers

ROT / Caesar cipher / substitution cipher / shift cipher (ROT13)

Vignere

- ROT determined by the key (repeating code)

XOR

- A comparison cipher using and exclusive or algorithm

Key Stretching

- The technique used to encrypt plaintext passwords into a hash so if the data is stolen the passwords are more difficult to determine
- Additional **unique random bits** are added to **each password** before hashing **(salting)** which makes the output less predictable
- The **salt** is stored appended to the password, or in another accessible location such as in the same database
- Salting the password before hashing it produces **different output result** for the same password input
- **bcrypt**
  - Is a widely used key-stretching algorithm
  - Is based on the **Blowfish** block cipher and is used on many Unix / Linux bistros and uses salt
  - Bycrypt results contain $2$1
- **PBKDF2 Password-based key derivation function**
  - Is a widely used key-stretching algorithm
  - Uses **64-bit salts** and pseudo random function to protect passwords
  - **WPA2**, and Apple iOS, Cisco OS
  - Some applications use the function 1000’s of times or more
  - Key sizes of 128 bits, 256 bits, 512 bits are most common

Cipher Modes

ECB Electronic Code Book

- 64-bit blocks
- Predictable because always produces same output (does not provide diffusion)
- Not recommended to use ECB

CBC Cipher Block Chaining

- A block cipher mode
- It's strength lies in using an **initialization vector (IV)** on the first block to ensure unique output and XORing the output of the previous block with the next block
- This increases the **diffusion** by creating different output for the same input
- CBC mode sometimes suffers from **pipeline delays** because of the dependency on the previous block being encrypted before the next block can be encrypted

CFB Cipher Feedback

- Uses an IV to increase diffusion
- Uses sub-block sizes of the main block size
- After the each sub-block encrypted, the first half the block is XOR'd with the IV and combined with the second half the IV which is then used encrypt the next block of plaintext with the same process
- This creates an IV that is always changing and partially depends on the previous IV
- This process is related until the entire message is encrypted

OFB Output Feedback

- Very similar to CFB mode, except that OFB does not XOR the output with the original input before breaking it in half and using the first half

CTR Counter Mode

- Converts a block cipher to a stream cipher
- CTR mode uses a random 64-bit block as the first IV then increments a specified number (the counter) for every subsequent block of plaintext
- The encrypted IV + counter is XOR's with the original message to get the cipher-text of each block
- CTR offers best performance since each block does not depend on the previous block to be encrypted

GCM Galois Counter Mode

- Used by many block ciphers
- It uses counter mode for operation and the **Galois mode** for **message integrity** and authentication using **message digests** to ensure the data has not been modified in transit
- It is widely used due to it's good performance
- 96-bit IV added to 32-bit counter starting with 1
- IV and counter are encrypted with the key and then used to XOR against the plaintext

Hashing Algorithms

MD5 Message Digest

MIC Message Integrity Check

ICV Integrity Check Value

SHA Secure Hashing Algorithm

- Provides message integrity
- Does not provide message authenticity
- **SHA-0**
  - Not in use
- **SHA-1**
  - 160-bit algorithm, 40 character hash (flawed)
- **SHA-2**
  - SHA-256 and SHA-512 are the major versions of SHA-2
  - SHA-224 and SHA-384 are the minor truncated versions of the major SHA-2 functions mentioned before
- **SHA-3**
  - Uses **Keccak**
  - Keccak (aka Sha3) was created outside of the US National Security Agency (NSA)

HMAC Hash Message Authentication Code

- **Primary Function:** to provide message integrity and authenticity by providing a message hash with the message that is tamper resistant
- Uses a combination of Hashing algorithm and symmetric key to verify integrity (message has not been changed in transit) of a digital message
- It can be used with block ciphers or streaming ciphers. HMAC can use MD5, and SHA series algorithm as the hashing function
- The general equation of HMAC is: HMAC = h(K2 + h (K1 + M))
- HMAC prevents message tampering by ensuring that the private key is hashed with the message in two rounds, there by making it more difficult to substitute a message and re-compute the HMAC without knowing the private key
- MACs in general can be build into the cipher or appended to the message
- RCF 2104 - https://www.ietf.org/rfc/rfc2104.txt
- The **steps of using an HMAC to provide integrity checks** to a message are:
  1. The sender/server encrypts the message/data with the clients public key
  2. The sender/server also hashes the message/data

3. The sender/server then encrypts the hash with it's private key
4. Sends the message/data do the recipient/client

RIPEMD RACE Integrity Primitives Evaluation Message Digest

- Not often seen
- Open-standard type development
- 128, 160, 256, 320-bit versions
- Relatively stable

HAVAL

- A cryptographic **hash function**
- Weaknesses have been found in HVAL
- Unlike MD5, but like most modern cryptographic **hash functions**, **HAVAL** can produce **hashes** of different lengths – 128 bits, 160 bits, 192 bits, 224 bits, and 256 bits
- **HAVAL** also allows users to specify the number of rounds (3, 4, or 5) to be used to generate the **hash**

Symmetric Encryption

DES Data Encryption Standard

- **16 Rounds**
- **Key size** 56 bit
- **Block size** 64 bits
- Allows use of different modes
- Can be broken more easily with brute force attacks
- NOT recommended for use today

3DES Triple Data Encryption Standard

- **48 rounds**
- **Key sizes** is 168 bits (3 x 56-bit keys)
- **Block size** 64 bits
- Basically using DES three times (hence rounds = DES x 3)

AES Advanced Encryption Standard

- **10 rounds** for 128-bit keys, **12 rounds** for 192-bit keys, **14 rounds** for 256-bit keys
- **Key sizes** of 128-bit, 192-bit, and 256-bit
- **Block size** 128-bit
- Allows use of various modes.
- *** most attacks are theoretical (see side-channel attacks)
- NIST adopted AES from the Rijndel encryption algorithm
- AES is **fast, efficient, and strong**.

BlowFish

- **16 rounds** of encryption
- **Key sizes** 32-bit to 448-bit (wide range of variable key sizes)
- **Block size** 64-bit

- *** considered good for strong encryption
- public domain algorithm

TwoFish
- **Key sizes** 128-bit, 192-bit, and 256-bit
- **Block size** 128-bit
- public domain algorithm

Stream Ciphers

RC4 Rivest Cipher
- *** Not a block cipher ***
- *** Is a streaming cipher ***
- **key sizes:** 40-bit - 2048-bit
- Very fast protocol
- Uses streaming pseudo-random (keystream) bits XOR'd into plaintext
- Especially Insecure when the beginning of the keystream is not discarded
- Prohibited for use if TLS by IETF (RFC 7465)
- 1 round of encryption
- most popularly used in wireless with the now obsolete WEP
- IETF RFC 7465 prohibits use of RC4 in TLS connections

IDEA International Data Encryption Algorithm
- **8 rounds** and a final half-round modification
- **Block size** 64-bit
- **Key size** 128-bit
- Intended as a replacement for DES
- A minor revision of an earlier cipher **PES Proposed Encryption Standard**
- Patent expired in 2012
- Was used in **PGP v2.0** and **BassOmatic**
- Also included as an optional algorithm in **OpenPGP**
- Uses XOR

Asymmetric Encryption (Public Key Cryptography)

- Used primarily for digital signatures, authentication, and to setup a shared key to encrypt communication channels
- Key Pairs (public / private) can be pre-shared or generated at run-time / during connection initialization
- **Key-pair**
  - A matching public and private key
- **Public key**
  - The public key can be shared publicly. It is used to send information to the party that holds the private key
- **Private key**
  - The private key is used to decrypt the message send that have been encrypted with the matching public key

    - However, the public key can also be used to encrypt messages that can be encrypted with the public key
- **Static and Ephemeral Keys**
    - Static keys are semipermanent (stay the same), while ephemeral keys are re-issued for each session or after a given amount of time
    - Ephemeral keys allow increased perfect forward secrecy since any previous data that has been passively monitored cannot be decrypted with the keys of the current session

Public Key Authentication

- Can be achieved using asymmetric key pairs between a **client** and **server**
- If the two parties each have a public / private key pair, and have already shared their public keys with each other:
    - The client and server (A & B) can authenticate to each other by encrypting a message:
        - First with their own private key
        - Second with the other party's public key
    - The client and server can then decrypt the exchanged messages using:
        - First the other party's public key
        - Second with their own private key
    - If the exchanged messages are able to be decrypted, the only possible sender of the message would be the party with which each party has shared their public key
    - However, this relies on keeping the public key secret to only each other
    - During the first connection, if the client does not have the server's public key, but the server does have the clients public key, then the server can send its public key encrypted with only the client's public key

RSA Rivest Shamir Adleman

- Suitable for encryption / decryption, digital signatures, and key exchange
- One round of encryption
- Uses a subset of primes (safe-primes) as key-space in the key generation process
- **1024-bit to 4096-bit** are standard key sizes
- Max key size is theoretically unlimited
- NIST recommends minimum 2048-bit keys
- NIST key management guidelines suggest that 15360-bit RSA key is equivalent in strength to using 256-bit symmetric key
- Some smaller key sizes have been broken in published attacks
- Very common standard

PGP/GPG Pretty Good Privacy / GNU Privacy Guard

- Bulk encryption, data-at-rest encryption (full-disk encryption, file encryption)

- Data-in-transit encryption similar to S/MIME for email
- PGP has been owned by Symantec Corporation since June 2010
- GPG is an open standard protocol
- Key pair generation
- Key exchange
- Web-of-trust rather than public key infrastructure

ECC Elliptic Curve Cryptography

- Suitable for encryption / decryption, digital signature, and key exchange
- Because Elliptic curve cryptography uses less computation and memory, it is suited for hardware applications
- Used for confidentiality encryption and digital signatures
- Especially due to low computational power and memory usage so implemented in smartphones and other mobile devices
- Elliptic curve cryptography uses less computation and memory, it is suited for mobile applications
- Smaller keys are more sure against brute force

El-Gamal

- Published by IEEE in 1985 - A Public-key crypto-system and signature scheme based on discrete logarithms (1985)
- Digital signatures and general encryption
- Partially based on Diffie-Hellman algos
- DSA is based on El-Gamal
- used in PGP/GPG
- There are 3 phrases in the protocol:
  - Setup Phase
    - Done only once
    - The receiving party
    - 1024 bit large prime number
    - Generate a public key from the prime number
  - Encryption Phase
    - The sending party creates an ephemeral key
    - Alice computes the shared key
    - Ephemeral key and cipher text is sent to recipient
  - Decryption Phase
    - The receiving party computes the shared key using

DH Diffie-Hellmen

- Diffie-Hellman is a **key-exchange algorithm (KEX algo)** designed to securely generate a symmetric key shared between two parties
- DH is not suitable for encryption / decryption or digital signatures
- The key-exchange enables more efficient encryption of data between to parties than using asymmetric keys
- **DHE / EDH Diffie-Hellman Ephemeral**
  - Uses new keys for each session
  - This allows forward secrecy since no two sessions use the same encryption key, if a key is breached, the amount of information that can

be decrypted is minimal
- **ECDHE Elliptical Curve Diffie-Hellman Ephemeral -** New keys are generated using ECC

DSS Digital Signature Standard
- Developed by **NSA National Security Agency**
- Specified as **NIST FIPS 186** in 1994
- Suite of algorithms used to create digital signatures
- Defines the **DSA Digital Signature Algorithm**
- Contains a definition of RSA signatures
- Uses certificate standards **PKCS #1 v 2.1** and American National Standard **X9.31**
- Contains a definition of **ECDSA - Elliptical Curve Digital Signature Algorithm**
- Required for US Government communications for sensitive unclassified information
- One of the most preferred digital signature algorithms used today

DSA Digital Signature Algorithm
- Adopted into NIST FIPS 186 DSS in 1994
- A variant of El-Gamal and Schnorr algorithms
- Originally used SHA-1, but current standards use SHA-2
- Key length can be multiple of 64 with minimum keys-pace of 512bits
- NIST SP 800-57 recommends minimum of 2048 bits

S/MIME Secure Multipurpose Internet Mail Extensions
- Used to encrypt email attachments by converting to text and specifying filenames in headers
- Used to digitally sign and encrypt email
- Used by most email applications
- Uses RSA for asymmetric encryption and AES for symmetric encryption
- Can encrypt email data-at-rest, and also data-in-transit
- Requires a PKI (public key infrastructure) to distribute and manage certificates

Digital Certificates

- Digital certificates can be used for encryption and authentication
- Certificate owners share their public keys by sharing the certificate itself
- **CA - Certificate Authority**
  - Manages, validates, and revokes certificates
  - Can be public CA for internet, or private CA server for internal network, LAN or WAN
- **Root Certificate**
  - Is the first certificate created by the CA that identifies it, and the store
- **Trusted root CA store**
  - Collection of root certificates that are trusted by connecting clients
- **Details in a Certificate**

    - **Serial number**
        - Uniquely identifies the certificate
        - The CA uses the serial number to validate the certificate
    - **Issuer**
        - The CA that issued the certificate
    - **Validity dates**
        - Valid from and valid to dates
    - **Subject**
        - The owner of the certificate
    - **Public key**
        - RSA symmetric encryption uses the public key with the matching private key
    - **Usage**
        - Whether the certificate is for encryption, authentication, or multiple usages
- **Trust Models / Certificate Trust Models**
    - If two CAs want to trust each other they issue each other certificates
    - **Hierarchal Trust Model**
        - Includes root CA that issues and signs certs to intermediate CAs, who can then issue certs to child CAs
        - All certificate holders know the root CA
        - If one **trust** the CA then he automatically **trust** the certificates that CA issues
        - The end user or server only need to verify the bottom level cert and the root cert
    - **Web of trust / decentralized trust model**
    - **Mesh Trust Model**
        - There is no root CA
        - Multiple peer CAs issue certificates to each other
        - The server verifies each certificate along the chain that connected the user to the server
    - **Bridge Trust Model**
        - Allows the trust relationship to be severed by breaking the bridge CA
        - Allows organizations to work together securely
        - A user accessing a resource or server can use either the hierarchal or mech trust models when connecting over a bridge CA (AKA principal CA)
        - Bridge server does not issue certs to end users but does issue to other CAs
- **Certificate chaining**
    - Combines all the certificates from the root CA down to the certificate issued to the end user
- **CSR Certificate signing request**
    - Includes the public key
    - Specifications include PKCS
- **CRL Certificate Revocation List**
    - If you want to revoke / expire a certificate before the expiration date, the CRL

    - If the private key is exposed then you would want to revoke the public key
    - Other reasons for revocation include: key compromise, change of affiliation, superseded, cease of operation, certificate hold
- **OCSP Online Certificate Status Protocol**
    - RFC 6960
    - Another method of validating a certificate
    - An Internet protocol used for **obtaining the revocation status** of an **X.509** digital certificate
    - The client can **query the CA with the serial number** of the certificate
    - The CA responds with the status of the serial number such as "good", "revoked", "unknown" which can **indicate forgery**
    - Clients using a **OCSP** can get more **up-to-date** responses than a **cached CRL** which could be out-of-date
- **OCSP stapling**
    - Clients receive a **time-stamped** response from the web-server that has been signed by the CA
    - This removes the need for the client to query the CA directly for the OCSP status
- **Certificate Issues**
    - Expired
    - Trusted
- **RA Registration Authority**
    - Assists in the registration process of collecting information related to the certificate's owner

Public Key Infrastructure (PKI)

- Hybrid cryptography solutions for secure tunnels for data-in-transit
- Group of technologies used to request, create, manage, store, distribute, and revoke digital certificates
- Asymmetric encryption depends on the use of certificates for a variety of purposes.
- Needs to ensure high level of security in terms of authenticity, integrity, and confidentiality, and non-repudiation

Public Key Pinning HTTP Public Key Pinning (HPKP)

- Is designed to **prevent impersonation of a website**, using fraudulent certificates
- The web-server responds with an **extra header** that includes a **list of hashes derived from valid public keys used by the website** and a max-age specifying how long the client should use the provided hashes
- Each time the client connects to the website, it compares a hash of the provided public key to the stored value to **ensure the public key of the server certificate has not changed**
- HPKP presents some risk because client will expect the server present a specific certificate chain for a specific amount of time, and if the server certificates change before that time, then clients will refuse to connect

Certificate Types

- **Machine / computer**
  - Certificates that have been issued to a particular computer
  - Used to identify a computer within a domain
- **User**
  - Certificates can be issued in order to identify a user, such as smart-cards, token devices (Ubikey, etc.)
- **Email**
  - Used for encryption of email, and digital certificates
- **Code signing**
  - Used to validate the authentication of applications, scripts, executables
  - It verifies the code has not been modified
- **Embedded Certificates**
  - Contained on smart-cards
  - Holds a user or machine private key
- **Self-signed**
  - Is not issued by the trusted CA
  - Reduce costs of getting trusted CA issued certs
  - These can be used for a development environment to use HTTPS in development
- **Wildcard (*)**
  - Used for multiple sub-domains of the same root domain
- **SAN Subject Alternative Name**
  - Used for multiple root level domains that are owned by the same organization
- **DV Domain validation**
  - Validates the ownership of a domain used on the internet
- **EV Extended validation**
  - Use additional steps beyond domain validation. For example, PayPal uses an extended validation certificate, and they company name will appear before the domain name in the url-bar
  - This helps prevent phishing attacks

Certificate Formats

- Certificates are usually stored as **binary** or as **base64 ASCII** (American Standard Code for Information Interchange)
- Some certificates are **encrypted for additional security**
- Certificate file **extensions** include:
  .crt, .cer, .pem, .key, .p7b, .pc7, .pfx, .p12 although **.cer files are not always using the CER** format mentioned below
- **CER Canonical Encoding Rules**
  - **ASCII - X.690** standard format and **ASN.1 (Abstract Syntax Notation One)**
- **DER Distinguished Encoding Rules**
  - **Binary - X.690** standard format, and **ASN.1**
- **X.509v3 format**
  - For user on the Internet and many other network applications
- **X.509v2 format**

- **PEM Privacy Enhanced Mail**
  - **Most common cert format** and can be used for all certificate type
  - Can be formatted as **CER or DER (ASCII or binary)**.
- **P7B PKCS#7 -**
  - Used to **share public keys**
  - Are **CER based**
  - Never include a private key
  - Uses **base64 (ASCII) encoding**
  - Has extension **.p7b**
- **P12 PKCS#12**
  - Are often used to hold the **private key**
  - Are **DER based (binary)**
- **PFX Personal Information Exchange**
  - Precursor to P12, often used with Windows, also used to hold the **private key**

SSL Secure Socket Layer / TLS Transport Layer Security

- Commonly used to encrypt HTTP (hyper-text transfer protocol) traffic in to HTTPS (secured HTTP)
- Can also be used to encrypt other data-in-transit such as FTP
- Requires certificates
- Internal networks can have an internal CA, which is especially useful in a non-transparent proxy
- The Internet generally uses a 3rd party CA to authenticate the server to the client
- Uses both symmetric and asymmetric encryption to achieve confidentiality
- Asymmetric encryption is used to create a shared key-pair which is then used to encrypt data-in-transit
- Certificate store in browsers hold a list of trusted CA

HSTS HTTP Strict Transport Security

- IETF RFC 6797 HTTP Strict Transport Security
- Enables websites to declare themselves accessible only via secure connections which is designed to prevent downgrade attacks and cookie hijacking
- Once a client connects and read the HSTS policy in the web-servers connection headers, the client will know that this website policy is to demand clients use SSL/TLS

Single Packet Authorization

- Make exploitation of public port services more difficult
- Reduce the probability of having a zero-day exploit an exposed service
- SPA requires only a single encrypted packet in order to communicate various pieces of information including:
  - Authentication
  - Desired access through a firewall policy and/or complete commands to execute on the target system
- Without the correct authentication a standard nmap port scan will not

reveal a listening port with port knocking protection enabled.

- **Links:**
  - Single Packet Authorization with **Fwknop**: **https://www.cipherdyne.org/fwknop/docs/SPA.html**
  - Single Packet Authorization: **https://www.linuxjournal.com/article/9565**
  - fwknop Single Packet Authorization -> Port Knocking: **https://www.cipherdyne.org/fwknop/**
  - A comprehensive guide to strong service concealment: **https://www.cipherdyne.org/fwknop/docs/fwknop-tutorial.html**
  - Single Packet Authorization: **https://www.securitygeneration.com/single-packet-authorization/**

KMIP Key Management Interoperability Protocol Standard

- Extensible communication protocol that defines message formats for the manipulation of cryptographic keys on a key management server
- Allows for clients to ask a server to encrypt or decrypt data, without needing direct access to the **key**