Course Notes

(COMS3000)
Version 1.2

(Kellie Lutze)

13 November 2017

This page is intentionally left blank

Contents

Glossary

AES
Advanced Encryption Standard 20
ALE
Annualised Loss Expectancy;
$(ALE = ARO \times SLE)$ 2
ARO
Annualised Rate of Occurrence 2

DCS
Distributed Control System 34
DES
Data Encryption Standard 20

FAR
False Acceptance Rate 10
FRR
False Reject Rate 10

IaaS
Infrastructre as a Service 36

OTP
One-Time Pad 23

PaaS
Platform as a Service 36
PCI DSS
Payment Card Industry Data Security Standard 33
PCN
Process Control System (has PLCs) 34
PCS
Process control System (Typically a DCS) 34

RSA
Rivest–Shamir–Adleman cipher 25

SaaS
Software as a Service 36
SLE
Single Loss Expectancy 2
SSL
Secure Socket Layer 25

TLS
Trasport Layer Security 25

1 Introduction

1.1 Security

What is security?

Webster online Dictionary:

- *the quality or state of being secure*
- *free from risk or loss*

Security is about dealing with potential loss of or damage to **assets**. Very much a business concept.

How to Deal with Risk?

- Prevention
- Insurance (passing consequences of risk onto others)

Things that can be done with a risk

- Accept it
- Transfer it (insurance)
- Reduce (mitigation)

This course will focus on the third option, that is the reduction of risk.

That is, the act of *Reducing the risk of 'damage'to information assets by means of Protective Measures.*

Information Security

Data can be *damaged* in a couple of different ways,

- Confidentially - Breach of confidentiality (privacy and secrecy),
- Integrity - Unauthorised modification of data
- Availability - Denial of service, ransom ware
- (Not really separate from above) Authenticity - Ensuring the author of a message is who they claim to be.
- (Not really separate from above) Non-repudiation - Prevention of being being able to repudiated something they have already done or agreed to.

2 Risk Management

2.1 What is Risk Management?

The process concerned with identification, controlling, and minimising or eliminating risks.

A **threat** is an actor that uses a **vulnerability** to

A **vulnerability** is a weakness in a system that can be **exploited** by a **threat**.

Risk management helps information systems management strike an informed economic balance between the impact of risks and the cost of protective measures.

Risk - A Definition *"The likelihood that a particular threat using a specific attack, will exploit a particular vulnerability of a system that results in an undesirable consequence."*

Threat - A Definition *"Any circumstance or event with the potential to cause harm to an information system in the form of destruction, disclosure, adverse modification of data, and/or the denial of service."*

Vulnerability - A Definition *"Weakness in an information system, cryptographic system, or other components (e.g... , system security procedures, hardware design, internal controls) that could be exploited by a threat."*

2.1.1 Risk Assessment (Risk Analysis/Evaluation)

"A process of analysing THREATS to, and VULNERABILITIES of, an information system and the POTENTIAL IMPACT the loss of information or capabilities of a system would have. The resulting analysis is used as a basis for identifying appropriate and cost-effective counter-measures."

The **Residual Risk** is the risk that remains after risk mitigation has occurred.

Risk management is dealing with potential outcomes vs the cost of preventing or mitigating the *risk* of these outcomes eventuating. This is important as there is often limited funds.

2.1.2 Quantitative Risk Analysis

$$RISK = \text{Expected cost of damage} = Impact \times Likihood$$

There are two main questions that are asked,

1. What is it the probability of a loss event occurring?
2. What is the impact in terms of $?

Using these questions to get out *quantitative risk analysis parameters* we can derive/assign some values.

AROs (Annualised Rate of Occurrence) $\Rightarrow$ The number of times a loss event is expected to occur within a year.

SLEs (Single Loss Expectancy) $\Rightarrow$ The impact (loss) of a single loss event occurring in $

ALEs (Annualised Loss Expectancy) $\Rightarrow$ The expected average loss per year due to a risk.

$$ALE = ARO \times SLE$$

ALE can serve as a measure of risk *exposure*.

2.1.3 Qualitative vs. Quantitative Risk Analysis

It can be difficult to assign probabilities to loss events and as such qualitative measures are used such as,

- Likelihood: high, medium, low
- Impact Rating: Very high, high, medium, low, very low

That is in the case where risks cannot be quantified they are ranked from highest to lowest.

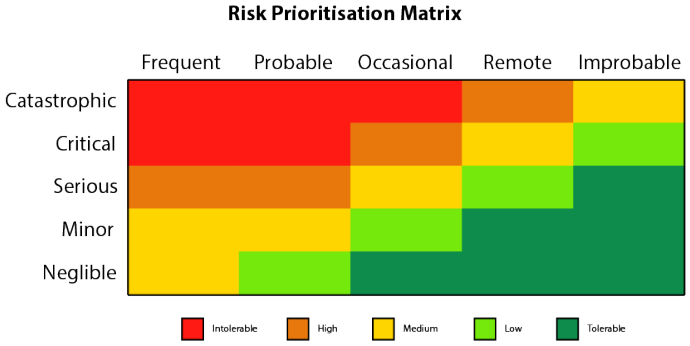


Figure 1: Risk Matrix Example

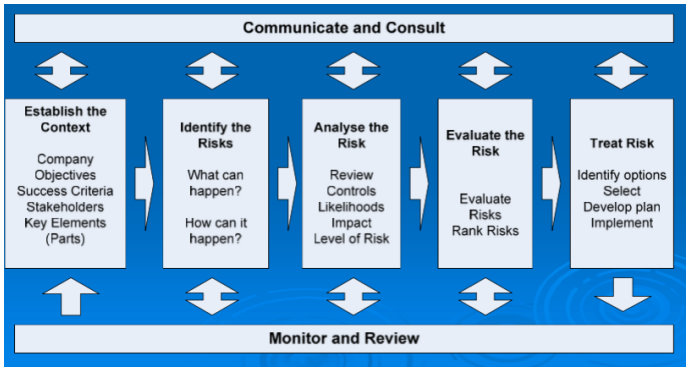


Figure 2: Risk Management Process

3 Certifications

There are various organisations that are responsible for the certification process.

3.1 (ISC)^2

International Information Certification Consortium. It is an international non-profit association focused on inspiring a safe and secure cyber world.

It offers Certified Information Systems Security Professional (CISSP) certification.

3.1.1 CISSP

Requires,

- Five years experience in infosec
- Experience needs to be verified by another CISSP.
- 6 Hour exam
- Costs $800 AUD

3.1.2 CBK

Common body of knowledge. This is where the CISSP domains are drawn from with various security topics existing within the CBK.

3.1.3 CISSP Domains

Consists of eight domains,

1. Security and Risk Management
2. Asset Security
3. Security Engineering
4. Communication and Network Security
5. Identify and Access Management
6. Security Assessment and Testing
7. Security Operations
8. Software Development Security

3.1.4 ISACA

Previously known as the Information Systems Audit and Control Association. Now it is just ISACA.

It is a nonprofit, independent association that advocates for professionals involved in information security, assurance,risk management and governance.

3.2 Digital Certificates

Certificates are used to establish secure connections to TLS/SSL or HTTPS. The certificate path shows the authentication chain.

3.2.1 X.509 Certificates

A public key certificate standard issued by ITU-T that defines a format of certificates. It is the most commonly used format today.

Structure

- Certificate
  - Version
  - Serial Number
  - Algorithm ID
  - Issuer
  - Validity
    - Not before
    - Not after
  - Subject
  - Subject Public Key information
    - Public key algorithm
    - Subject Public key (e.g. n and e for RSA)
  - Issuer Unique Identifier (Optional)
  - Subject Unique Identifier (Optional)
  - Extensions (Optional)
- Certificate Signature Algorithm
- Certificate Signature

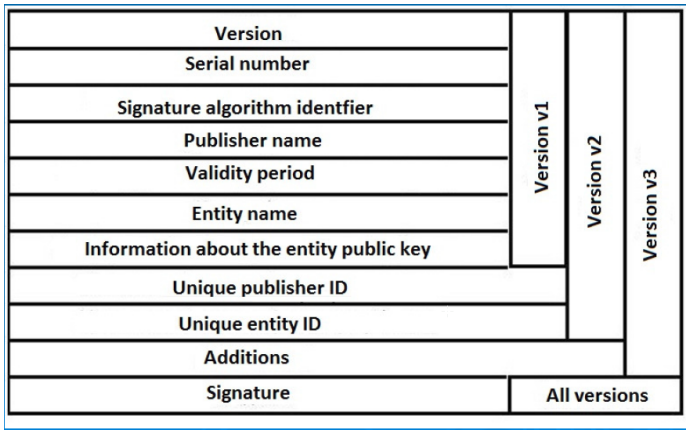


Figure 3: X.509 Certificate Structure

It should be noted that the value of the version field inside of a version two certificate is actually '1', not the expected '2'. This is due to the version one certificates having a version value of '0'.

3.3 Certificate Authorities (CAs)

Public key certificates can be purchased from CAs or self-signed (in the case of self signing, since no CAs can vouch for it, it will not be trusted by everyone else by default).

Details of the identify verification process are usually published in a document called the ***Certification Practice Statement (CPS)***.

CAs often have different levels of identity checking, such as responding to emails from a given address all the way to presenting ID such as passports etc.

Extended Validation (EV) Certificates require a more rigorous verification and validation process through the CA. As a result they are more expensive but are also trusted more as indicated by the green address bar in browsers.

3.3.1 Creating A Self Signed Certificate

Use OpenSSL which is an open source implementation of SSL/TLS as well as providing a generic crypo library it is often installed on most systems already.

1. To start with, a public key and private key pair are needed.

   ```
   openssl genrsa -des3 -out privkey.pem 2048
   ```

2. Now generate the self-signed certificate

   ```
   openssl req -new -x509 -key privkey.pem -out cacert.der -days
   ```

3. Display the contents of your new certificate

   ```
   openssl x509 -in cacert.der -noout -text
   ```

4. The new certificate can bow be installed in the Windows Certificate Store (or Linux/Mac equivalent)
5. (Optional) You can generate a Certificate Signing Request (CSR) which can send the certificate to a CA for signing

   ```
   openssl req -new -key privkey.pem -out cert.csr
   ```

3.3.2 Untrusted Certificates

In the case that a certificate is untrusted (no one verifies it or the CA that is verifying it is not trusted) i.e. Broken chain of trust.

This could indicate a man-in-the-middle attack and will branded as untrusted by the browser.

4 Access Control

4.1 The Concept of Trust

Fundamental to Information Security

4.1.1 Trust vs. Trustworthiness

Bod is trustworthy when he actually returns something that he was *trusted* to return

4.1.1.1 Direct and Indirect Trust

Direct: Based on Experience
Indirect: You don't know the person, the level of trust can be gained by a trusted third party

Concrete Example: E-Commerce Credit cards - Using one online is indirect trust on the sellers side in the sense that the trust that the payment will be made without any reversals etc.

Another example of indirect trust is the use of SSL where the certificate authority is the trusted third party.

4.2 Access Control

We need to control who has access. This access can allow creation, reading, modification and deletion of data. **Lack of Access Control** can lead to confidentiality, integrity and availability being compromised (see Section 1.1 for definitions on these three risk areas). A door with a lock is an example of access control where the key is required to gain access.

Another example is the use of age limitations on entering bars in Australia. Access control can be anonymous or identified.

4.2.1 Access Control Decisions

Access control normally deals with identities in information security.

There are two separate functions that are key to access control.

1. Establish identity,
   a) Identification (determine who they are/computer, person or machine etc)
   b) Authentication (see Section 5)(verification of id) - Proves the claimed identity.
2. Authorisation (see Section 6)
   - Once the identity of a person is known and verified, a decision can be made concerning what they are allowed to access. This is often associated with a role → Role Based Access Control (RBAC)

The decision can be attribute based (i.e. Age for entering a pub) or time based (i.e. Traffic lights) or anonymous (e.g. Movie ticket where the possession of a ticket gives access regardless of the identity of the person).

Access control is usually by Id as it allows the provision of an audit trail to exist.

Example:
Student ID card: Id is the name, Photo is the authentication and room access is the authorisation.
Parking Ticket: Ticket is the id, registration is the Authentication

5 Authentication

The process of verifying a claimed identity.

This can be provided through the use of three general types of mechanisms.

1. With something you **know**. i.e. Password, signature, PIN etc.
2. With something you **have**. i.e. Student Id card, time based/event based tokens (think 2FA)
3. With something you **are** (or do). i.e. Think biometrics

5.1 Passwords

The most common form of authentication. They are something you **know**.

5.1.1 Problems with Passwords

- Hard to remember
  - Same password used on multiple systems - One system gets compromised, all are compromised.
  - Users forget them
  - Users write them down
  - Failure to reset default passwords
- Attacks on passwords
  - Eavesdropping i.e. Network sniffing
  - Should surfing - Someone looking over your shoulder
  - Social engineering
  - Brute force (common passwords first significantly reduces the cracking time)
  - Key loggers
  - Attacks on the audit trail - Passwords being added to audit logs are failed logins
  - Attacks against storage - Passwords used to be stored in Plain text.They are now commonly hashed. (see Section 8.9)
  - Fake interfaces i.e. Card skimmers etc

The *Carna Botnet* is a great example of issues regarding default passwords,

$\Rightarrow$ 500 000 devices compromised via services such as telnet.

NOTE: A useful tool for scanning ports or *pen testing* is nmap.

5.1.2 Key Issues with Insecure Communication with Password Authentication

There are three ways an attacker could gain access to someones passwords,

1. By gaining access to the information stored on the server (i.e password file),
2. Interception of the password in transit
3. User disclosure i.e. Picking an easily guessed password, writing it down etc.

The final issue cannot be solved with the traditional methods as most systems do not have an ability to distinguish between the people entering the same password (i.e. If the attacker has the same password the system cannot differentiate between them and the original user) [1].

Solving the first issues can be achieved through the use of a one-way hash function to encode the password (see Section 8.9). This ensures that the information that an attacker can gain access to is in actual fact not the password. Therefore it is not automatically compromised (see Section 5.1.5 for ways in which it still may be compromised).

The second issue is not solved using this hash. An attacker can still eaves drop on the connection and perform a replay attack. A solution to this issue is the use of a sequence of "passwords"in which, the ith time a user authenticates with the ith password. These passwords would need to be some sort of hash of the actual password. i.e $y_i = F(x_i)$

According to [1], there are two obvious schemes.

1. All passwords within the sequence are computed prior to the connection and the server is aware of them all.
2. The user sends the y_{i+1} after logging in with x_i

This is also relevant to One time passwords which can be found in Section 5.1.3.

These two issues have the inherent issues with the first one requiring large numbers of values to be stored and the problem of the second one not being robust.

A solution that combines features from both is as follows.

Let the ith password x_i equal $F^{1000-i}(x)$ for some fixed word x. So our sequence is

$$F^{999}(x), ..., F(F(F(x))), F(F(x)), F(x), x$$

Since it is feasible to compute F^n, we can use this method to compute the next password in the sequence based on the previous one thus requiring us to only remember the last password. This is robust as the method F is unknown to an attacker, even if they have access to the previous passwords, they cannot compute the next one and since each password is distinct a replay attack would not succeed.

Now in the case that the client and the server get out of sync, they can repeatedly apply $F(x_i)$ until a match is found. This provides a means in which a repeat of an previously used password is not possible.

Now, since we have a ceiling of how many permutations of the original x we want (namely 1000). When we reach this ceiling a new value of x should be chosen and the processes started again. This provides an additional level of security.

Additionally, the above depends on the assumption that the calculation of $F(x)$ is not only feasible but can be achieved quickly.

5.1.3 One Time Passwords

The idea of single use passwords is to defend against eavesdroppers by using passwords that cannot be used again. They are predominantly used to counter replay attacks.

The S/Key One-Time password system is as described in [2] is one scheme for one time passwords.

A single use password is the only password to ever cross the network. The users secret pass-phase is never transmitted.

While the S/Key system does protect against replay attacks. It does not protect against inside jobs or an attacker from gaining access to private information (as it is authentication only). It also does not protect against attacks where the intruder can intercept and modify the packet stream.

There are two sides to this scheme,

1. Client - Client must generate the one time password,
2. Host - Verification of the one time password and allow the user to change their secret pass-phrase.

The client passes the users secret pass phrase through multiple layers of a secure hash function to create the onetime password. On each use the number of applications used is decremented by one allowing for a unique sequence of passwords.

The S/Key system uses the MD4 message digest algorithm for its secure hash function. The one-time passwords are 65 bits in length.

Since the MD4 Algorithm creates a digest of 128 bits regardless of the input size. The S/Key system folds this output using XOR to produce a 64bit password.

When a challenge is raised by the host, the client generates/looks up

their one time password using the seed and their secret pass phrase. The way this work for the verification is the same as detailed in Section 5.1.2.

5.1.4 Unix Passwords

Traditionally, Unix stored encrypted passwords in the a file found at `/etc/passwd` which could be read by any one.

The contents of this file appear as follows.

```
root:fi3sED95ibqR6:0:1:System Operator:/:/bin/ksh
uucp:OORoMN9FyZfNE:4:4::/var/spool/uu:/usr/lib/uucp/uucico
rachel:eH5/.mj7NB3dx:181:100:Rachel
Cohen:/u/rachel:/bin/ksh
arlin:f8fk3j1OIf34.:182:100:Arlin
Steinberg:/u/arlin:/bin/csh
```

The issues that this method introduced was that attackers could launch an *offline* attach with pretty much no limitations on the number of attempts that they could make.

To solve this issues, the encrypted or hashed passwords need to be hidden. i.e. in a file such as `/etc/shadow` which is not accessible by anyone.

5.1.4.1 Attacks Against Password Storage

The issue is in the fact that passwords need to be stored somewhere accessible to the system during login. This creates a vulnerability.

Originally passwords were stored in plaintext.

5.1.4.2 UNIX Password File

As mentioned in the parent section, encrypting the file does not completly solve the issue.

For example, in the case where two users use the same password. The resulting hash would be the same and would allow both users to login as each other (since they know the password). Unix originally used 12 random bits in a modified DES algorithm to prevent this. This method is called *SALT*.

5.1.5 Password Cracking

If an attacker has access to the password file they can perform the following attacks,

- Dictionary attack $\Rightarrow$ For *common* words p, calculate $h(p)$ and see if it matches any entry in the password file.
- Brute-force attack $\Rightarrow$ Try all the possible password combinations (See Section 5.1.5.1 for more details).
- Hybid $\Rightarrow$ Use of a dictionary of words combined with numbers etc.

These are *offline* attacks and are only successful when the attacker has plenty of time and access to the file of hashed passwords.

These types of attacks do not need to be engineered from scratch with tools existing that perform the above functionality. Kali Linux is a penetration distribution that comes pre-packaged with over 30 of these tools.

5.1.5.1 Brute Force Attack

Brute forcing a password is the process of trying every possible password combination. When brute forcing a password, the following assumptions are made.

- Password length of 8 characters.
- Randomly chosen password (if this is not the case, a dictionary attack would be more successful.
- A guess throughput of 10, 0000, 000 guesses per second.

The time such an attack would take depends on the complexity of the password.

Case One
Only lowercase alphanumerical letters (i.e. a-z).

$26^8 \approx 2 \times 10^1 1 \rightarrow$ An attack would take on average 2.75 hours if there is only one hashed password. Within 2.75 hours we have also attempted on average half the possible passwords. (i.e. Not very efficient).

Case Two
All printable characters (95 chars available).

$95^8 \approx 6.6 \times 10^1 5 \Rightarrow$ An attack would take 1 year on average.

Now, the above two cases make some assumptions on the guess throughput that could be considered conservative. In 2015, there was a 25 GPU cluster that achieved a speed of 350 billion guesses per second which resulted in it being able to brute force an 8 character password in less than 6 hours given case two.

More information about this cluster can be found at https://arstechnica.com/.

Given this above cluster, a password can still be considered secure if it is long enough and has enough entropy.

Time-Memory Tradeoff
To save time an attacker could store previously calculated hashes of possible passwords. However, to do this for all possible hashes given case two $95^8 \approx 6.6 \times 10^1 5$hashes $\approx$ 105600TB making it impractical to do it for all passwords. Some pre-computed passwords can be stored to save time and this is where the tradoff occurs.

A table for storing these pre-computed hashes is called a rainbow table. These are known to take more memory and CPU time but increase the speed in which a password can be cracked. RainbowCrack is a tool that makes use of rainbow tables in order to crack passwords.

Additionally, it is possible to download large rainbow tables for use to save time in computing your own.

5.1.6 Password Selection

It is important when selecting a password to ensure that it is secure. This is not as easy as it may seem as even seemingly random passwords can be surprisingly easy to crack.

Tweaking dictionaries of possible passwords allows for high levels of success against many passwords making a brute force attack rarely needed. A key in making a password secure with regards to dictionary attacks is creating one that is unlikely to exist in anyones dictionary in any form.

This is explained in more detail here.

It is can be difficult to select a password that is both secure and usable and a trade off arises from this problem. Truly secure passwords are not easy to remember and therefore causes issues with usability. A password can only be secure until it gets written down, which difficult to remember passwords encourage.

5.1.7 Online Password Attack

The Brute force attack (see Section 5.1.5.1) and dictionary attackes mentioned before are both *offline* attacks. An *online* password attack can be represented as,

- attempting to guess someones PIN from a stolen card,
- someone attempting to access your online accounts (if they know your login/username) by guessing your password.

The key difference between an online and offline attack is the time and

access limitations that dealing with a live system entails. It can be much harder to attack online passwords and much easy to defend against attacks with features such as limited number of incorrect attempts.

5.1.8 Security of Passwords

The level of security that a password offers is controlled by the UI. If a user interface includes features such as limit number attempts before being 'locked'out or having your card eaten by the machine increase the security offered by the password.

It can also be hard to automate some online passwords such as ATM PINs making it slow.

5.1.9 Password Managers

A password manager is a tool that stores passwords for a user to assist them in dealing with large numbers of passwords for different services. The use of sucha tool generally will only require a user to remember a single *master* password which allows access to their password store of site specific passwords.

Features that such tools commonly offer include,

- Option to encrypt site specific passwords chosen by user
- Option to automatically generate a password for the user (secure but hard to remember but since it is managed by the password manager the user does not need to remember it providing the best of both worlds).
- Online and offline modes,

Popular tools include Dashlane, LastPass, RoboForm, 1Password, KeePass etc.

5.2 Authentication Protocols

Authentication is the process in which a claimed identify is verified. There are various protocols that are used to achieve this.

5.2.1 Password Based Authentication Protocols

Commonly involve authentication over a network. i.e. Remote login, authentication to web server etc.

Protocols in this space include PAP(Password authentically protocol, RFC 1334).

Password authentication introduces a couple of potential problems as since the the users credentials are passed over the wire, Eve can eavesdrop and learn the password. In the case that the password is hashed, Eve can still resubmit the hashed password in what is called a *replay attack*.

To address this problem, the **Challenge Response Protocol** was established.

5.2.1.1 Challenge Response Protocol

Server gives the client a 'challenge'(c), often called a *nonce*. The client calculates a response (r), as a cryptography one-way hash of the nonce and password (p). i.e $r = h(c \parallel p)$[1] or $r = (c \text{ XOR } p)$. Client sends r back to the server. This allows the server to validate the response whereas the eavesdroppers can only see c and r but cannot calculate p. This ensures that a replay attack does not occur.

Choice of Challenge

Consider the server in the context of this protocol where the challenge is chosen deterministically and predictably, i.e. C1 = 1, C2 = 2 etc. The response would be $r = h(c_i \parallel p)$. Most would believe that the challenge should be random and unpredictable and in the case that it is not a problem occurs.

This is not the case, the important part is that c is not reused as while an attacker may be able to predict the challenge, they still do not gain advantage (a replay attack would be possible if the challenge was repeated).

Practical Example: HTTP Authentication

HTTP (Hypertext transfer protocol) is a simple request/response protocol between a web server and a client. It provides a simple access control and authentication mechanism (i.e. Limit certain web pages to certain users).

It is defined in the RFC 2617 and involves two basic types of authentication.

- Basic,
- Digest

A http `401 (Unauthorized)` is the challenge response that is sent by the server to the client. This response **MUST** include the `WWW-AUthenticate` header field as can be seen in the below sections.

The http `407 (Proxy Authentication Required)` response is used as the challenge for the authentication for a proxy. It also **MUST** include a `Proxy-Authenticate` header containing at least one challenge.

Basic Authentication

1. Client sends `GET` request to the server.
2. Server responds with
   - `Not authorised for realm XYZ` [2]
   - Requires `Basic Authentication`
3. Client asks the user for a username and password for the given realm ⇒ usually through a pop-up.
4. Client sends a new `GET` request
   - "Please give me this file. Authentication details are:"
     - Basic Authentication, username:password
   - The server checks the dteails and responds.

The problem with the above scenario is that the password has been sent in plain text and can be eavesdropped upon. Basic HTTP authentication is **NOT** secure!

Digest Authentication Key benefit of digest authentication is the ability to not send the password in the clear.

1. Client sends `GET` request to the server.
2. Server responds with

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
realm="Demo Content"
nonce="1011565422321"
```

   The server reply includes the nonce.
3. Client asks the user for a username and password for the given realm ⇒ usually through a pop-up.
4. Client computes response r as a hash (or a *digest* of the username, password, realm, nonce and url.
5. This hash is sent back to the server which verifies it.

This solves the issue found in basic authentication with the password never being sent in plain text which prevents a replay attack. However, it is still password based and the plain text password is still stored on the server. It also makes use of MD5 for its hashes which is considered broken.

It is vulnerable to **man-in-the-middle attacks**.

[1] It should be noted that $\parallel$ means concatenation in this case.

[2] A realm is a set of protected files or directories.

While not perfect, it is ok for low security applications. To gain a higher level of security other security features can be added on top such as certificate based authentication i.e. HTTPS and TLS.

5.2.2 SSH Authentication

SSH (Secure shell) authentication protocol[3]

SSH Supports several methods such as password, public key etc.

Password Authentication

1. SSH Establishes an encrypted channel between client and server
2. Password (not hash) is sent accross this channel.

Limitations include,

- Vulnerable to man-in-the-middle attacks
- There is a lack of authentication when establishing the secure channel.

Public Key Authentication

- SSH1 is Vulnerable to man-in-the-middle attacks
- SSH2 uses stronger key exchange
  - Diffie-Hellman group exchange for the Transport Layer protocol
  - RSA key exchange for the Transport Layer protocol
  - Generic Security Service Application Program Interface (GSS-API) authentication and key exchange for the SSH protocol.
  - Elliptic Curve Algorithm Integration in the SSH Transport Layer
  - SHA-2 Data Integrity Verification for the SSH Transport Layer Protocol
- Supports several authentication methods,
  - Password,
  - Public key
  - Keyboard-interactive (RFC 4256): e.g. S/Key or SecurID
  - GSSAPI

1. SSH Establishes an encrypted channel between client and server
2. Password (not hash) is sent accross this channel.

Limitations include,

- Vulnerable to man-in-the-middle attacks
- There is a lack of authentication when establishing the secure channel.

5.2.3 Lamport's Hashed Password Scheme

1. Client selects initial secret password p_0 (seed value)
2. Compute the sequence of the hash values
   - $p_1 = h(p_0)$
   - $p_2 = h(p_1) - h(h(p_0))$
   - $p_3 = h(p_2) - h(h(h((p_0)))$
   - ...
   - $p_n = h(p_{n-1})$
   - $p_{n+1} = h(p_n)$
3. Store p_{n-1} at server with counter n
4. When the user logs in, server sends n as a challenge.
5. Users sends p_n and server calculates $p_{n+1} = h(p_n)$ and compares it with the stored values of $p_{n+1} \Rightarrow$ if correct, user is authenticated
6. Server decrements n by one.
7. Next time, users needs to send password higher up in the list

[3]SSH v1 is vulnerable to man-in-the-middle attacks.

Eavesdroppers observing p_n cannot compute p_{n-1} which is required for the next authentication (the one-way nature of the hash function.

This leaves no need to store the password on the server!

Possible attacks is the Man-in-the-Middle Attack where the attacker impersonates the server and receives the password to be used during the next authentication.

The action of decrementing n results in the need to recalculate and redistribute the hash onces we reach index 0.

5.3 Multi-Factor Authentication

When higher level of security is required, two or three mechanisms can be used together. The most common example of this is two factor authentication (2FA).

Examples of this include,

- Something you **know** and **are** $\Rightarrow$ i.e. Password and fingerprint
- Something you **know** and **have** $\Rightarrow$ i.e. Password and physical key
- Something you **are** and **are** $\Rightarrow$ i.e. Fingerprint and physical key

A good example of this is the need of a PIN and your credit card. Other examples are Google's 2FA, Twitters Login verification and Paypals securekey.

SecureID

SecureID runs a pseudo random number generator to create a new token every few minutes (sometimes seconds). The use of a secure algorithm guarantees that even after observing for a long time (many numbers in the sequence), the numbers that are generated are still unpredictable.

The same algorithm runs on the server (loosly time synced).

5.4 Biometrics

Biometrics are a form of authentication that is based on *something you are*. It is based on aspects of your body and behaviour.

The word itself comes from the Greek words *bio* (life) and *metrikos* (measure).

Biometrics is the set of automated methods to recognize a person based on physiological or behavioural characteristics - Biometric Consortium.

Biometrics are becoming increasing relevant with passports, personal computing devices etc all making use of them.

Put simply, a biometric system is simply a pattern matching tool [3].

Features that can be used for biometrics include,

- Signature,
- Fingerprints,
- Voice,
- Iris,
- Retina,
- Hand-geometry,
- Gait,
- Face,
- DNA,
- Odour (Olfactory Biometrics),
- Keystroke dynamics,
- Hand vein,
- ear,
- and many more.

A biometric system is made up of the following four modules [3],

1. Sensor - The tool used to capture the biometric data from the user.

2. Feature extraction - The tool used to extract a set of "*salient or discriminatory*"[3] features from the data sourced from the sensor.
3. Matcher - This is where comparisons between the captured data and stored templates occur. Encapsulated within is also a decision making module where a threshold exists which decides how close does the template have to match the captured data to be considered a valid match.
4. Database - This is where the template data is stored for enrolled users. The enrollment module is where the first template data is captured. It is also important to include a validation stage where the data quality can be verified.

What Makes a Good Biometric?

1. Universality ⇒ Each person should have the characteristic i.e. Bald people don't have hair colour.
2. Distinctiveness ⇒ Any two people should be sufficiently different in terms of this characteristic. i.e. Shoe size would not be considered distinctive.
3. Permanence ⇒ The characteristic should be sufficiently invariant over a period of time. i.e. Hair colour might change.
4. Performance ⇒ Recognition accuracy (and speed)
5. Acceptability ⇒ The extent to which people are willing to accept the biometric in their daily lives. i.e Retina scans may seem intrusive, hand geometry may be considered unhygienic.
6. Circumvention ⇒ How easily the system can be faked or tricked. e.g. Gummy fingers

These characteristics can be broken into two main groups,

1. Identification,
2. Practicality

With identification taking, universality, distinctiveness and permanence. Practicality takes performance, acceptability and circumvention [3].

Biometric characteristic	Universality	Unicity	Persistence	Colectability	Performance	Acceptability	Circumvention
Face	high	low	medium	high	low	high	low
Fingerprint	medium	high	high	medium	high	medium	high
Hand Geometry	medium	medium	medium	high	medium	medium	medium
Iris	high	high	high	medium	high	low	high
Retinal Scan	high	high	medium	low	high	low	high
Signature	low	low	low	high	low	high	low
Voice	medium	low	low	medium	low	high	low
Thermogram	high	high	low	high	medium	high	high

Figure 4: Comparison of Biometric Methods

The success of various biometrics can be found in [4].

5.4.1 Types of Biometrics

Biometrics can be sorted into two groups,

1. Physiological
   - Fingerprint,
   - Retina,
   - etc
2. Behavioural
   - Signature,
   - Voice,
   - Gait,
   - Keystroke dynamics (good for continuous authentication)
   - etc.

5.4.2 Enrollment

Biometrics require a user to enrol where their data is *scanned* for later comparison. This can include the initial scanning of a fingerprint for fingerprint recognition systems. This data is then stored with the users identity.

5.4.3 Biometrics Modes

A biometric system can operate in two modes,

1. Verification,
2. Identification.

5.4.3.1 Verification Mode

1. User Identifies themselves. i.e. With a PIN
2. Biometric is scanned and compared with the users template in database (matched with PIN)
3. System answers the question. *Does this Biometric belong to the user?*.
4. One-to-one match.

Formal Description

X_Q: Input feature vector (this is extracted from the scanned data).

I: Claimed Identify. This is from the PIN for example.

The problem: Determine if (X_Q, I) belongs in the class w1 or w2 where,

- w1: Genuine user (Good)
- w2: Imposter (Bad!)

X_Q is matched against X_I (the biometric template of user I).

$S(X_Q, X_I)$ is the function that measures the similarities between the feature vectors (the template and the new feature vector).

$S()$ is called the matching score, this is usually a single value that indicates the level of match. The larger $S()$ the better the match. To find match cutoffs (at what point is the match not good enough?) we can simply use the threshold as calculated based on the biometric used.

5.4.3.2 Identification Mode

The user is not required to claim identity.

The system attempts to answer the question *Whose biometric is this?*. This of it as usering the biometric as a username.

This achieves a 1-N match and therefore is more difficult and more prone to error. The larger N the higher the probability of an error.

Formal Description
When given an input vector X_Q, we need to determine the corresponding identity I_k in the set of templates (template database) where $k \in 1, 2, 3, ..., N, N+1$ where $I_1, I_2, ..., I_N$ is our set of enrolled users.

if I_{N+1}, there is no match and the identification is rejected.

In the same way that we match against the given templates for Verification, a match threshold is also used.

5.4.4 Problems with Biometrics

Errors with biometrics can commonly be organised into two types, False Accept and False Reject.

Additional groups of errors also exist such as Failure to enrol.

5.4.4.1 False Accept (False Match)

This is when Alices biometric vector is wrongly matched with Bob's. This has the implication of Alice potentially gaining access to Bob's stuff. FARs FAR: False acceptance rate (or probability), or FMR: False match rate (or probability). These two terms are used to describe the proportion of impostors granted access.

5.4.4.2 False Reject (False, Non-Match)

This is a failure to match a valid biometric vector. This is more secure as it causes a lack of access rather than accidental access. However, the implications can be problematic as if Bob cannot gain access to his account/things/stuff, the value of the biometric system is decreased.

FRRs FRR: False rejection rate (or probability) or FNMR: False non-match rate (or probability) iare terms used to describe the proportion of genuine users rejected.

5.4.4.3 Failure to Enrol (FTE)

This can be caused by a low quality of input that prevents the successful creation of a template.

5.4.4.4 Causes

The above errors can be caused by a number of things such as,

- Sensor noise,
- Different characteristics of scanners. (i.e. Differing resolution between enrolment and verification/validation scanners)
- Different biomentric state (i.e. Dry fingers, bruises, cuts etc)
- Ambient conditions such as humidity or temperature,
- User interaction (i.e. Finger placement)

5.4.5 Matching Score

When attempting to match a scanned vector to a template of the same person it is important to be aware of possible variations that can occur due to the causes outlined in Section 5.4.4.4.

When modelling the match score $S()$ of measured vector and a given template can be modelled as a random variable with a PDF (probability density function). The PDF can be determined through measurements. The key idea being that the PDF of the matching score for a genuine user will differ from that of an imposter.

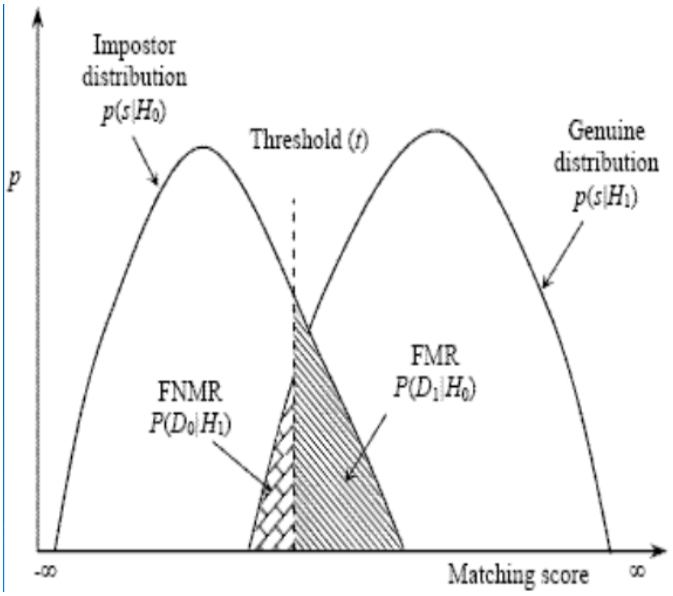


Figure 5: PDF of a Matching Score for a Genuine User Vs that of an Imposter

5.4.6 Performance of Biometrics

When considering biomentric systems, performance is important.

A False match can occur when the $S()$ of the imposter is over the threshold. A False Non-Match can occur when the $S()$ of a genuine user is under the threshold.

To reduce these two errors, it is possible to tune the system by adjusting the threshold.

Lowering t can have the effect of FMR increase and FNMR decrease.

Increasing t has the effect of FMR decrease and FNMR increase. This idea leads into what is known as the FMR-FNMR tradeoff (see Section 5.4.7).

5.4.7 FMR-FNMR Tradeoff

Since the threshold is proportional to the FNMR and inversely proportional to FMR a tradeoff exists between the two. As such, different applications choose different operating points for different tradeoff levels.

High security applications attempt to minimise the FMR as if a false match were to occur it would be a disaster. As such, a user is more likely to encounter a false non-match in these applications.

Forensic applications, i.e fingerprint matching etc are less strict and make the occurrence of a false match acceptable. As such, they will have a low threshold to reduce the likely hood for a false non-match.

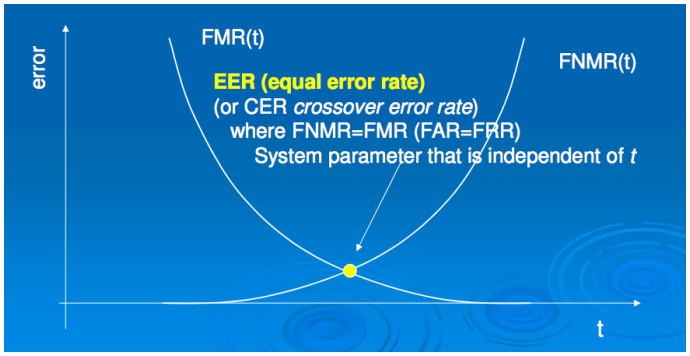


Figure 6: FNR-FNMR Trade Off

FMR or FNMR alone do not give any information about the performance of a system. Rather, both values are required, or the cross over accuracy (see Figure 6).

The equal error rate (EER) or the CER (Crossover error rate) is where FNMR==FMR (FAR == FRR).

5.4.8 Selection of a Biometric System

No system is perfect, as such it is often a trade off between characteristics such as universality, distinctiveness, performance, accuracy etc (see Section 5.4).

To maximise the benefits of a system, it is possible to combine multiple systems such as fingerprint readers with face recognition etc.

5.4.9 Attacks

Same as any other system, attacks are possible.

For facial recognition, a replay attack is possible with the recording of an image and replaying it. A spoofing attack is the act of simply holding up an image.

A solution to such attacks is something called a liveness test.

5.4.9.1 Liveness Test

Currently an ongoing research topic. It involves the verification that the image being used is of a real live person.

Attempts have been made by checking for blinking, checking reflection etc. While these seem good a solution for protecting against all forms of attacks is difficult to come by. As the blink test will only prevent spoofing attacks, not replay.

5.4.10 Limitations

Limitations include,

- Error rates
- Circumvention
- Revocation - Leaking of biometric information and the inability to *reset* this information like you can with passwords.

The inability to reset biometric information in the case that it leaks brings into question the viability of it as a form of verification alone but rather as a part of a pair (i.e. Password and fingerprint). It is an effect form of identification though.

6 Authorisation

6.1 Computer Based Access Control

When managing authorisation, the assumption is made that the user has been authenticated (i.e. They are who they say they are).

More access control information can be found in Section 4.

The fundamental access control model includes,

- Subject: Active party (user, process, etc)
- Object: Passive party (files etc) - It should be noted that the roles of subject and object are heavily dependant on the context and can be reversed.
- Reference Monitor: Grants of denies access.

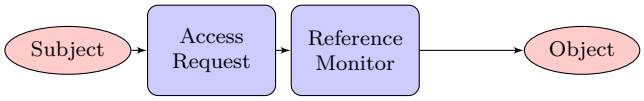


Figure 7: Fundamental Access Control Model

6.2 Access Control Policy

An access control policy specifies what subjects can do with objects. That is, what are the basic operations that can be applied to objects such as files. Some of these operations may include,

- Read,
- Write,
- Update,
- Execute,
- Append (excluded read access) - Think log file.
- Delete,
- etc.

An access control policy is simply who can do what, with which objects?

We define the following:

- set S of *subjects*
- set O of *objects*
- set A of access operations

Access rights can be defined as an *Access Control Matrix* where each entry M_{SO} specifies the set of access operations subjects s can perform on an object o

$$M = (M_{SO})_{s \in S, o \in O} \text{ with } M_{SO} \subset A$$

	Bill.doc	Edit.exe	Fun.exe
Alice	-	{execute}	{execute, read}
Bob	{read, write}	{execute}	{execute, read, write}

Table 1: Example Access Control Matrix

6.2.1 Access Control Matrix

Access control policies are rarely implemented directly as access control matrices as they are difficult to maintain for large numbers of objects, especially within dynamic environments.

An alternative solution is to store the access rights with the objects themselves. This is the concept of access control lists (see Section 6.2.2). We could also store the access rights with the subjects (this is capabilities).

6.2.2 Access Control Lists (ACL)

Access Rights are stored with the objects themselves. Columns of the access control matrix represent this information (see Table 1).

The advantage of this solution is that is makes it easy to see who has access to specific objects.

Unfortunately, it also makes it expensive to revoke access of individual users as you need to search through all ACLs.

Managing access rights of a large number of individual users (subjects) via ACLs can be tedious. A solution of this is the introduction of groups or roles.

Users can be grouped together with others that have the same permissions. This allows the same policies to be applied and revoked easier. To revoke rights of an individual, simply remove them from the group.

6.2.3 Unix Access Control

Traditional Unix systems provide a limited for of ACLs.

The owner of each file (and root) can specify

- Access rights for the owner,
- Access rights for people in the same group
- Access rights for everyone

Each access right is a combination of

- R - Read access (allowed to read the file)
- W - Write access (allowed to write to the file)
- X - Execute access (Allowed to execute the file)

This can look like the following (9 bits),

```
-rwxr-x--- bill students thefile
```

- The first character indicates the directory (d) or normal file (-)
- Characters 2-4 indicate the **owner** (bill) can read, write and execute,
- Characters 5-7 indicate the **group** (students) can read and execute but not write
- Characters 8-10 indicate that no permissions are granted to **everyone**(world)

Windows uses a similar slightly more fine grained system.

In Unix everything is treated as a file. More recent versions of Unix implement full ACL.

With everything being a file, another difference is that files are not accesses by users, but by processes started by users, this process will have an ID and the access rights of the user that started it.

There is some exceptions to the above. **SUID** is one. It stands for set user id (SGID → set group ID) which allows processes to be run under the owners privileges rather than the user. This opens quite the potential security issues.

6.2.3.1 SUID Vulnerability

Often SUID programs are owned by root, i.e. Have unrestricted access.

An attacker could exploit this by gaining control of such a program and using it to execute arbitrary code, with root access.

Pre-eminent attack → "Buffer overflow attack" is one such attack method.

6.2.4 Principle of Least Privilege

> *"Every program and every privileged user of the system should operate using the least amount of privilege necessary to complete the job" - Jerome Saltzer*

6.2.5 Capabilities

Alternative to ACLs for managing access rights.

- A capability is an (object - rights) pair that is used like a movie ticket.
- Rights are stored with Subjects → Rows of access control matrix
- Advantages include,
  - Easy to see permissions of individual users
  - Easy to revoke access for a particular user
- Disadvantages include,
  - Hard to see who has access to a particulate object
  - Harder to revoke access to a particular object → must find all tickets.
- It is rarely used in practical systems

6.2.6 Types of Access Control - Ownership

- The owner typically is in control of defining access rights "Discretionary Access Control" (DAC) - At the discretion of the owner. There is an exception with root.
- The alternative is a system wide policy that defines access to objects "Mandatory Access Control" (MAC), this is often used in the military and other high security contexts.

6.2.6.1 Problems with Discretionary Access Control (DAC)

A third party can ask or trick the owner for access.

One method of tricking the owner would be to do the following,

1. Create a new file f2
2. Grant the original owner write access to f2
3. Grant the third party read access to f2
4. Copy contents of f to f2.

This program would have to be executed by the original owner.

This highlights the key issue being that programs and users cannot be trusted. Malicious programs of the type mentioned above are called Trojan horses, they provide additional functionality to what is expected with the additional functionality being malicious in intent.

Using DAC is insufficient as programs cannot be tested for all types of unexpected behaviour.

6.2.6.2 Mandatory Access Control (MAC) - Multi-level Security

- Every file (object) has a **classification**. i.e. a tag to indicate its sensitivity level. e.g. Secret, top-secret etc.
- Each user (subject) has a **clearance**, which indicates the sensitivity level of objects they can access.
- A user can only access objects that match or is lower than their clearance. Clearance dominated the classification.
- Classifications and clearances are assigned by the system/management with no user discretion allow.

Categories - Compartments

Now, not everyone with an appropriate clearance can access all the documents as it is often not necessary.

Categories exist within classification levels and are known as **compartments.**

These can be organised by topic, i.e Nuclear, Crypto etc. A user has access to their level of classification within their assigned compartment.

This extra levelling allows for more fine grained control.

Lattice-based Access Control Model

Denning, Dorothy E. "A lattice model of secure information flow."Communications of the ACM 19.5 (1976): 236-243.

Elements of a lattice model include,

- An ordered sequence of sensitivity **levels**
- A set of **components** (sometimes called codewords)
- a "dominated" relationship written $\geq$

Nodes in a lattice consist of sensitivity label and a set of compartments.
i.e {secret, Nuclear}, {secret, Crypto}, {secret}, {top-secret, Nuclear}

Some of these are comparable and others are not.

- {top-secret} $\geq$ {secret}
- {top-secret, Nuclear} $\geq$ {secret, Nuclear}
- {top-secret, Crypto} ?? {secret, Nuclear} **No dominance relationship**

To have access to an object, a subject needs to have the necessary **clearance** AND **compartment**.

Properties include,

- Transitivity: $A \geq B$ and $B \geq C \rightarrow A \geq C$
- Reflexivity: $A \geq A$
- Antisymmetry: $A \geq B$ and $B \geq A \Rightarrow A = B$
- Partial ordering $\rightarrow$
  - Not every two elements need to be comparable,
  - For every A and B there exists a greatest lower bound
  - For every A and B there exists a least upper bound

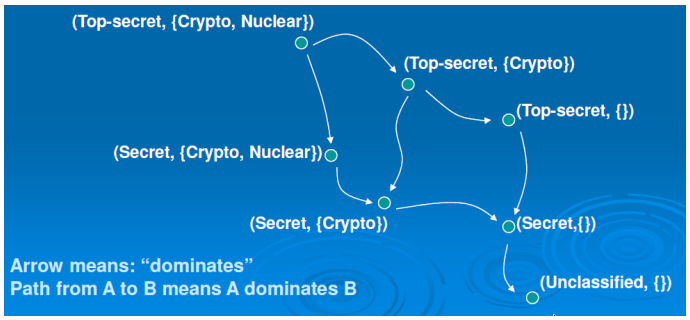


Figure 8: Lattice Example

6.2.7 Bell-LaPadula (BLP) Model

The goal was to create a formal (mathematical) model of a security policy for Government and Military applications.

It was developed in the early 1970s and was funded by the US department of defence.

It was supposed to support multi-user systems (alternative, separate machine for each level of classification). Each user has a clearance.

Each process operating on behalf of the user is a subject, with each processes clearance level being able to be lower than that of the owner (at the owners discretion). Each file is an object and therefore has a classification. BLP is concerned about the confidentiality and as such it attempts to prevent information from leaking from a higher level to a lower one.

Two rules exists to prevent information from flowing downwards.

- "Simple Security Property" - A subject can only read an object if its clearance dominates the objects classification (is equal or higher). **It should be noted that this does not guarantee that no information flows downwards as a user or Trojan with a high level security clearance could copy files to lower level classification.**
- *(start)-property- - A subject at a given security level must not write to any object at a lower security level.

6.2.7.1 High Water Mark

Consider a subject with a secret clearance. This subject has not accessed any files yet

- No need to be at the highest level,
- Its current security level is unclassified.

This subject can write to an unclassified file, then they read from a classified file. This updates the current level to classified.

That is they can no longer write to an unclassified file.

The prevention of allowing the security level from being decreased would make this concept redundant.

6.2.7.2 Discussion

It is a significant security model and as such is important for the design of new secure operating systems.

It has been implemented in Multics

- OS developed by MIT in 1960
- "predecessor" to Unix.

Limitations do exist,

- it only addresses confidentiality (i.e. Not integrity etc)
- Does not address the problem of managing classifications.
  - How are classifications assigned, changed etc?
  - How are clearances assigned, changed etc?
- It is not very practical for standard applications, perhaps too complex.
- Does not address the problem of convert channels.
  - Unintended communication channel. i.e Users can communicate through the modulation of the CPU load.

6.2.8 Security Evaluation/Certification - "Common Criteria" (CC)

The Common Criteria for Information Technology Security Evaluation. It is the international standard (ISO/IEC 15408) for evaluation and certification of security of computer systems.

It is functional and in terms of assurance levels - Evaluation assurance level (EAL) - 1 being the lowest and 7 being the highest.

It allows for specification of security requirements of a system and rigorous evaluation against them.

It is largely used by Government Agencies.

Getting certified is expensive in terms of money and time. Windows 7 is EAL4 certified.

6.3 SE Linux

> *"NSA Security-enhanced Linux is a set of patches to the Linux kernel and some utilities to incorporate a strong, flexible mandatory access control (MAC) architecture into the major subsystems of the kernel ..."*

It has been integrated into the Linux kernel since version 2.6 2003.

7 Information Theory

7.1 What is Information?

We want to be able to mathematically model and define information.

Shannon's Definition of Information

See Section 7.1.1 for more information on Shannon.

> *"Information is reduction of uncertainty"*

Information is gained from observing the outcome of a random experiment.

7.1.1 Claude Shannon

1926 - 2001

The father of information theory and the father of the digital/information age. He is responsible for the first publication of the term *bit*. He provided a mathematical foundation of electronic communication and cryptography.

His two most significant publications while at Bell Labs were,

- A Mathematical Theory of Communication (1948)
- Communication Theory of Secrecy Systems (1949)

7.1.2 Measure of Information

The unit of Information is bits (Introduced by Shannon is his 1948 paper).

$I(X)$ is the average amount of information gained from observing the outcome of a random variable X. It is also the minimum average number of bits needed to encode all possible outcomes of X

Now the question of how much information do we get on average by observing the outcome of a random experiment with N possible outcomes?

Firstly, let's assume that all outcomes are equally as likely.

Experiment	**Outcome**
Tossing a coin ($N = 2$)	$I(X) = 1$ bit.
Rolling a four sided dice ($N = 4$)	$I(X) = 2$ bits.
Rolling an eight sided dice ($N = 8$)	$I(X) = 3$ bits.
Rolling an N sided dice	$I(X) = log_2 N$ bits.

Table 2: Average Amount of Information Examples

From Table 2, the base of the logarithm is the unit we are currently using. We could change this to get a different unit but this is rarely done.

Base 10 is known as the Hart (Hartley) or decimal unit with base e being knonw as the natural unit.

It should be noted that information theory is not concerned with the value of information, only the quantity.

The equation seen in Table 2 is as follows,

$$I(X) = log_2 N$$

Where,

- X is the random variable being observed.
- N is the number of possible outcomes.

It assumes that the outcomes N are all equally likely which is not always true.

Given a probability p for each of the outcomes N as

- $p = \frac{1}{N}$
- $N = \frac{1}{p}$

We can rewrite out definition of information to be,

$$I(X) = log_2 N = log_2(\frac{1}{p}) = -log_2(p)$$

7.1.3 Information - Entropy

Often Shannon information is also called "*Entropy*".

> "*My greatest concern was what to call it. I thought of calling it 'information', but the word was overly used, so I decided to call it 'uncertainty'. When I discussed it with John von Neumann, he had a better idea. Von Neumann told me, 'You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, nobody knows what entropy really is, so in a debate you will always have the advantage.*" - Claude Shannon, as quoted in M. Tribus, E.C. McIrvine, "Energy and information", Scientific American, 224, 1971

7.1.4 Information Vs. Encoding

Information is the minimal number of (lower bound) of bits we need to encode information.

For example, take a coin toss, we can encode the outcome as follows,

- Heads, Tails,
- H, T
- 0, 1
- the result is heads, the result is tails,
- 000000000000, 111111111111
- etc.

From this example the amount of information is 1 bit regardless of how we encode it.

7.1.5 Coding

Take a random variable X with $N = 8$ equally likely outcomes,

$$I(X) = log_2 8 = 3\text{bits}$$

From observing the outcome of X we gain 3 bits of information. That is to say that a minimum of 3 bits it used to encode all possible outcomes.

Take our random variable X and set $Nk = 3$ where the outcomes are taken from the set {A, B, C}

$$I(X) = log_2 3 = 1.58\text{bits}$$

To encode this we can use the following approaches.

First approach,

- A $\rightarrow$ 00
- B $\rightarrow$ 01
- C $\rightarrow$ 10
- 11 unused

So we get 2 bits per outcome or symbol.

According to Shannon we should be able to do better.

Second approach, let us use variable length codes (note we don't have a separator to delimit the different code words)

- A $\rightarrow$ 0
- B $\rightarrow$ 1
- C $\rightarrow$ 10

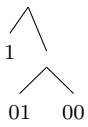
If we encode A,B,C we get 0110. Since we don't have a delimiter we cannot separate words from each other make decoding this sequence ambiguous.

To remove this ambiguity. Lets added a so-called prefix free code. i.e No code word can be the prefix of another code word.

7.1.5.1 Prefix-free Codes

Phone numbers are prefix-free codes.

We can create a prefix-free binary code by selecting code words are leaves of a binary tree.



So using this idea we can make a second attempt at encoding

- A $\rightarrow$ 1
- B $\rightarrow$ 01
- C $\rightarrow$ 00

Now we get 10100 from the A,B,C $\Rightarrow$ No Ambiguity!!!

The average code word length (assuming all symbols are equally likely) = 1.66...

Huffman coding is a mechanism to find optimal prefix-free codes. It operates on the principle of making short code words for frequent symbols and longer ones for less frequent. This is used for lossless compression.

The theoretical minimal average code length (bits) is the Entropy $I(X)$.

7.2 Hartley's Definition of Information

$$I(X) = -log_2(p)$$

The issue with this definition is that it makes the assumption that all outcomes are equally likely with the constant probability $p = \frac{1}{N}$. This is not always the case.

A solution to this problem is the use of the weighted average of 'Harteley's Information'. That is for each outcome multiply the information gained by its probability. Then add up all the outcomes.

i.e. $I(X) - p_1 \times (-log_2(p_1)) + p_2 \times (-log_2(p_2))$

7.3 Shannon's Measure of Information

Shannon's Information (Entropy) is the weighted average of Hartley's Information

$$H(X) = -\sum_{i=1}^{N} p_i \times log_2 pi$$

In the case where all the outcomes are equally as likely, both Shannon's and Hartleys definitions are the same.

7.4 Entropy in Practice

The entropy contained within 64 random hexadecimal characters is 264 bits.

As the entropy contained within a single hexadecimal character is 4 bits.

i.e. $4 \times 64 = 256$.

Now for average entropy.

8 Printable ASCII Characters

Printable characters have a character code 32 - 127 excluding delete (127) that is 95 characters. Given that they are truly random we cannot use compression. Therefore we have to assume that each character is equally likely.

$$I(X) = log_2 95 \approx 6.57 \text{bits/character}$$

.

So for eight characters we get $8 \times 6.57 \approx 52.56$ bits.

7.4.1 Binary Entropy Function

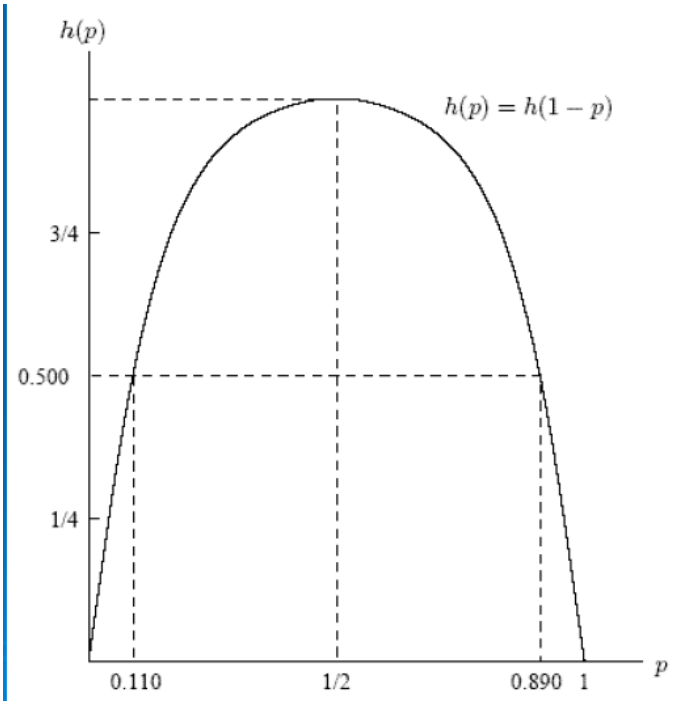


Figure 9: The Binary Entropy Function

Given a random variable X with two outcomes x_1 and x_2, with probabilities p and $1 - p$. We managed to achieve the maximum Entropy/Uncertainty when both outcomes are equality as likely.

7.5 Redundancy

Using more bits for messages than their Entropy is called redundancy. Reducing redundancy gives us compression.

Redundancy is not always bad, it can be used to detect and correct errors and as such having less redundancy results in less error tolerance. Using this idea we can see that the English language has a huge amount of redundancy.

7.6 Entropy in the English Language

$H(X) = 3.9$bits per letter. Looking at combination of letters gives us digrams (two letters) such as TH and EN, trigrams (three letters) and the other N-grams (N-letters).

Doing the same type of analysis for N-grams we get a convergence at about ≈ 1.5 bits per letter. Using ASCII we get 8 bits per letter (truly random ASCII printable ASCII we can get 6.57 bits per character).

A very good compression algorithm should be able to compress English by a factor of ≈ 5.

7.7 Password Entropy

Entropy is often used as a measure of password quality with higher entropy resulting is higher quality and lower entropy meaning passwords as easier to guess.

As can be seen by Shannon's definition of information, the quality of a password does not depend on just the total number of possible passwords (N) but also their frequency (probability).

7.8 Number Conversions

7.8.1 Hexadecimal

1. Divide number by 16
2. Record the integer value as our result
3. If there is a remainder, multiply by 16, record this remainder.
4. Divide our result by 16, and repeat steps 2 and 3.
5. Continue until the integer division results in zero.
6. Convert the numbers that we recorded as remainders as hex values based on the mapping shown in Table 10.

HEXADECIMAL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DECIMAL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 10: Hexadecimal/Decimal

Example

Decimal numbers 10 and 33.

For 10.

$$10/16 = 0.625$$
$$0.625 \times 16 = 10$$
$$10 \text{ decimal} = A \text{ hexadecimal}$$

As can be seen here, anything less than 16 can be converted straight away.

For 33.

$$33/16 = 2.0625$$
$$0.0625 \times 16 = 1$$

$$33 \text{ decimal} = 21 \text{ hexadecimal}$$

7.9 Fundamental Theorem of Arithmetic

Every positive integer greater than one can be expressed uniquely as a product of primes, apart from the rearrangement of terms.

8 Cryptography

8.1 What is Cryptography?

The primary aim of cryptography if confidentiality, with secondary aims of authenticity, integrity and non-repudiation.

Terminology

- **Crypography** - Literally *secret writing*. Science and art (practice) of keeping messages secure.
- **Cryptanalysis** - Science and art (practice) of breaking message security.
- **Cryptology** - *Secret words* - Science of secret communications (theory and math) associated with crypography and cryptanalysis.

8.2 Cryptographic Algorithms "Ciphers"

The central component of cryptography.

The basic model involves *decryption* and *encryption*.

Encryption of the plaintext P is the ciphertext C

$$C = E(P)$$

Decryption of the ciphertext C is the plaintext P

$$P = D(C)$$

This gives us the general identity

$$P = D(E(P))$$

.

Ciphers allow us to ensure that only the expected and authorised people can encrypt and decrypt the information. A simple way of achieving this is restrict knowledge of the algorithms to the authorised individuals. That is, keep the algorithm secret.

Issues with this is the non-scalability as there would need to be a different algorithm for every conversation.

A better approach is the use of **keys**. Modern algorithms are actually large classes of encryption and decryption functions (not a single one). Sender and received both need to use the correct function and which function is used defined by the index or the **key**. That is the function is parameterised by the key.

From Section 8.3 we have $C = E_{Ke}(P)$ and $C = E_{Kd}(P)$ and the identity

$$P = D_{Kd}(E_{Ke}(P))$$

In the case of Symmetric cryptography $Ke = Kd$ (see Section 8.14), in the case of asymmetric cryptography $Ke \neq Kd$ (see Section 8.15), these are also called public key ciphers.

8.3 Kerchhoffs' Principle

Auguste Kerckhoff

> *"The security of a cipher should rely on the secrecy of the key only"* - Auguste Kerckhoffs, "La Cryptographie militaire", 1883

The above quote works with the assumption that an attacker know everything there is to know about a cryptographic algorithm, that is as stated by Shannon "the enemy knows the system".

There is the alternative of *"security through obscurity"* which is widely considered to be bad practice.

8.4 Types of Attacks

The goal of the attacker is to recover the plaintext, or deduce the key (even better).

Attacks can be classified based on what information is available to the attacker.

- **Ciphertext only attack** - The attacker knows the cipertext of several messages encrypted with the same or several keys.
- **Known-Plaintext attack** - Known ciphertext/plaintext pair of several messages.
- **Chosen-Plaintext attack** - Attacker can choose the plaintext that gets encrypted thereby potentially getting information about the key.
- **Adaptive Chosen-Plaintext attack** - Attacker can choose a series of plaintext, basing the choice on the result of previous encryption → differential analysis.

The is also the **brute force attack** where an attacker attempts every possible key. If this is the best that an attacker can do against a cipher it is considered to be secure and strong. Security can be increased by increasing the key length.

8.5 Simple Encryption

Assuming all you have is pen and paper, there are several ways to encrypt a message.

Let the message be *"attack at dawn"*.

8.5.1 Caesar Cipher

This is a good example of a *substitution cipher* where one letter is replaced by another.

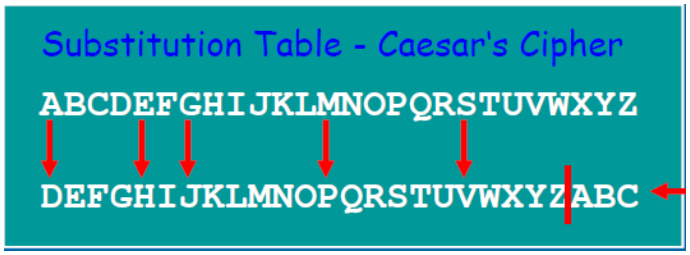


Figure 11: Caesar Cipher Substitution Table

As can be seen in Figure 11 there is three cyclic shifts. That is the list of characters have been shifted three spaces to the left.

Using this method, we can generalise and shift it any number of times k for $0 < k < 26$. It should be noted that this cipher is not very secure at all and since there are only 25 different keys, brute forcing it is easy.

8.5.1.1 Monoalphabetic Substitution Cipher

The Caesar cipher is an example of a monoalphabetic substitution cipher.

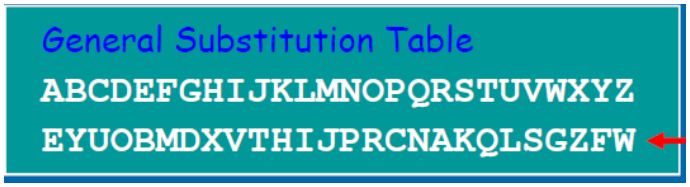


Figure 12: Monoalphabetic Cipher Substitution Table

Unlike the method shown in the previous parent section, this method does not reply on simple shifting but rather it allows the use of any permutation.

With the alphabet, that gives us $26! \approx 4 \times 10^26$ possible keys.

This is much more secure against brute force due to the sheer size of the key space. However, sometime such as frequency analysis (see Section 8.8) would work much better.

Substitution methods are not used as a stand alone technique in modern ciphers. However, the principles do form the basis of them.

8.5.1.2 Vigenere Cipher

This method improves on the monoalphabetic substitution cipher method detailed in Section 8.5.1.1.

Improvements include the use of multiple alphabets (polyalphabetic cipher) where each plaintext key is mapped to a different ciphertext letter. This changes the frequency between the plaintext and the ciphertext.

The Vigenere cipher is attributed to Blaise de Vigenere, a French diplomat. It was actually invented by Giovan Battista Bellaso in 1553. It was thought to be unbreakable for > 250 years.

It involved the use of 26 different Caesar ciphers in what is known as a Vigenere square. Each row corresponded to a Caesar cipher.

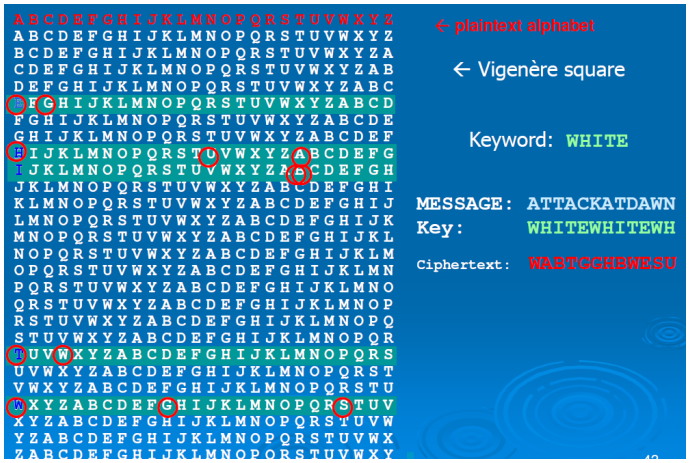


Figure 13: Vigenere Polyalphabetic Substitution Cipher

This cipher can be broken by looking for repeated words (N-grams) in the cipher text with $N > 2$. These words can occur by coincidence. Although this occurs less frequently for larger words. It can also occur when the same key lines up with the same plaintext word resulting in the same word being encrypted with the same key.

If we can determine the key length n we know that every n-th letter is encrypted with the same Caesar cipher. We can then used frequency analysis to break each of the n Caesar ciphers individually.

To determine the key length, we know that a distance that repetitions must have a distance that is a multiple of the key length.

Kasiski Test

First invented by Charles Babbage, it was later independently invented by Friedrich Wilhelm Kasiski.

It is an attack method for polyalphabetic substitution ciphers.

It works as follows,

1. Find repetitions of N-grams, N>2
2. Write down the distances of the repetitions
3. Key length is likely to be the greatest common divisor of these distances.

8.5.1.3 Enigma Cipher

Famous example of example of a polyalphabetic substitution cipher. It had a key space of $\approx 180 \times 10^{18}$.

It was first broken by the Polish Cipher Bureau pre WWII, and then by the British during.

8.6 Transposition Ciphers

Instead of substituting letters, transposition ciphers reorder them.

This is performed by,

1. Writing down the plaintext on a piece of paper in horizontal rows of c characters each.
2. The ciphertext is the vertical reading of this message. Column after column,

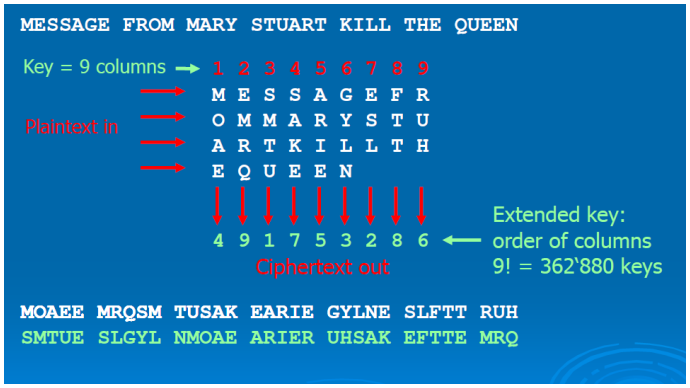


Figure 14: Transposition Cipher

Blocks of cipher text can be identified as being encrypted using a transposition cipher as it will have the same relative frequency as normal English text.

Ciphertext created using a transposition cipher can be broken using *Anagramming*.

This method is not used as a stand alone technique in modern ciphers. However, the principles do form the basis of them.

8.7 Perfect Security

A cipher is considered to be strong if the best attack against it is brute force. This leaves the security to be dependant on the key length, the resources the attacker has and how long the information needs to be protected for. This is the basis of the term *computational security*.

While this is not perfect security, such a thing does exist. A theoretical unbreakable cipher given an attacker has infinite resources and time.

A One-time Pad or a Vernam cipher are both theoretically unbreakable. This is proven by Shannon.

8.7.1 One-time Pad (OTP)

Given a plaintext message of n bits, choose n random bits (one time pad, only to be used once).

Each bit is XOR'ed with the corresponding bit of the random bit string to create the cipher text.

i.e. $C_i = M_i \oplus K_i$.

Decryption is achieved through the XORing of the key with the ciphertext.

- $C_i \oplus K_i = (M_i \oplus K_i) \oplus K_i$
- $C_i \oplus K_i = (K_i \oplus K_i) \oplus M_i = M_i$

$\oplus$ is commutative (i.e. $A \oplus B = B \oplus A$) and associative (i.e $A \oplus (B \oplus C) = (A \oplus B) \oplus C$).

The OPT is unbreakable due to,

- The ciphertext C not revealing any information about the plaintext M in a information theoretical sense.
- If knowledge of the ciphertext does not change, the probability that a message M was sent is $P(M|C) = P(M)$, M and C are independent random variables. No correlation exists between the two.
- An attacker has the same change of guessing the message with or without knowing the ciphertext.
- Shannon proves that if $P(M|C) = P(M)$ an attacker gains 0 bits of information from observing the ciphertext.

If an attacker attempts to brute force the OTP (try every possible combination of the key) it is impossible to tell which one is the real one. In normal ciphers, there is typically only one. For the OTP, given a ciphertext of n bits C and generating all possible plaintexts, there is not way to prove which one is real.

The reason we use others (even though OPT is perfectly secure and fast) is that for a key of n bits, the Entropy needed to be n bits (no redundancy) as well as the difficulty in generating truly random bit strings.

The key needs to be the same size as the message and can never be reused. It is hugely expensive. Encryption of a one gigabyte file will require a one gigabyte key. This makes key distribution difficult. Compare this with AES which a 256 bit key can be used to encrypt a message of any size.

8.8 Frequency Analysis

This is an approach to breaking encryption where statistics are used.

For monoalphabetic Substitution ciphers (see Section 8.5.1.1), since the letters are just replaced the frequency of each letter won't changes.

That is the letter with the highest frequency most likely responds to the letter e which has highest frequency in English. This can be done for each letter.

Statistics of N-grams can also be used within this method.

Since monoalphabetic substitution ciphers can be broken easily using this method, it shows that a large key spaces is not enough.

8.9 Hash Functions

A hash function[4] $h()$ is a function with the following two properties.

1. Compression: $h()$ maps an input x of arbitrary finite bit-length to an output of $h(x)$ of fixed, typically short bit-length.
2. Ease of computation: Given $h()$ and an input x, it is easy to calculate $h(x)$

A *one way* has function has an additional one-way property

- For essentially all output of $y = h(x)$, it is *computationally unfeasible* (practically impossible) to find x.
- This is also called 'pre-image resistance'(See Section 8.9.3 for details on a Pre-Image Attack).

[4] Useful online hashing tool can be found at http://onlinemd5.com /

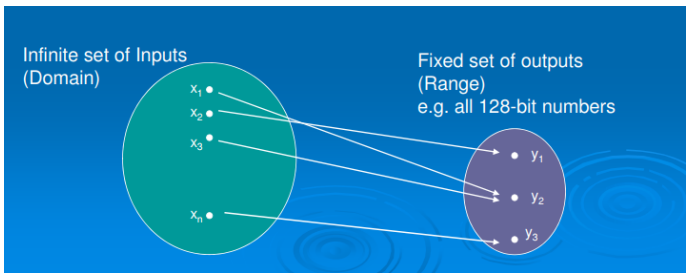


Figure 15: Hashing Problem Space

8.9.1 Collision Resistance

A *cryptographic one-way hash function* is a one-way hash function with the additional property of 'lq Collision Resistance''.

A collision is when two distinct inputs result in the same hash. It is important to avoid collisions as they create opportunities for the integrity, availability and the confidentiality of a given system to be compromised.

Two primary types of collision resistance.

1. Strong
   - It is computationally unfeasible to find any two distinct inputs x_1 and x_2 so that $h(x_1) = h(x_2)$
2. Weak (also known as 2nd pre-image resistance) See Section 8.9.3 for details on the Pre-image Attack.
   - For a given output $y = h(x_1)$, it is unfeasible to find another input x_2 so that $h(x_2) = h(x_1)$
   - It should be noted that this is more difficult than the above case therefore it is classified as the weak resistance.

Collision resistance is crucial for the security of algorithms and protocols.

8.9.2 An Ideal Model of a Cryptographic One-way Hash Function

Picture a machine with an elf inside where the h elf has

- An unbiased coin,
- A pen,
- A really long piece of paper

When an input arrives,

1. If it is known, use the same output as before
2. If it is unknown, toss the coin for a new output (one toss per bit), record the input and the output for later reference

Assumptions:

- Everyone has access to the machine, but cannot access the internal data. i.e It is impossible to access the previously recorded input output values that are stored within the machine.

This is known as a **Random Oracle Model** and it has the important property that if even a single bit differs on the input, the output is completely different.

8.9.3 Pre-Image Attack

The *work factor* to find a *pre-image* for an n-bit hash function (using the random oracle model) is the amount of the work to find a given output $h(x_2)$ such that $h(x_1) = h(x_2)$. Simply put, two inputs that result in the same hash.

The best method for finding a *pre-image* is to brute force it. Now since there are 2^n (where n is the size of our output in bits) outputs therefore this is our work factor.

Now if we consider a 128 bit outside (like that which is used in MD5), and the availability of 1 billion computers that can calculate 1 billion has values a second it will still take about 20 000 billion years to find a *pre-image*.

8.9.4 Collision Attack

Finding a collision between any two inputs.

- It is easier than the pre-image attack
- The work factor is $2^{\frac{n}{2}}$ (Birthday paradox) - This can be achieved in 10 seconds which is significantly faster then when compared to the 20 000 billion years that it would take a pre-image attack.

8.9.4.1 Birthday Paradox

The probability that in a set of n randomly selected people, that two people share a birthday reaches 99.9% when $n = 70$ and 50% when $n = 23$. These probabilities work with the assumption that each day of the year is equally probably as a birthday.

8.9.5 Cryptographic One-Way Hash Function

Also called *digest functions* or *digital finger prints*.

Examples

- MD5 - 128 Bit digest - **Broken!**
- SHA-1 - 160 bits **Broken!**

A cryptographic hash function is considered **Broken** if it collisions can be found significantly faster than brute force would allow.

Hash functions that are still considered (practically) secure at this point in time are,

- SHA-2 - Different bit lengths 512 → SHA-512
- SHA-3 - Announced the Keccak Algorithm as the winner of the 5 year NIST (National Institute for Standards and Technology) competition, October 2012

8.9.6 Finding Hash Collisions or Pre-Images

Bitcoin Mining

The process of finding an input with the first n bits of the hash output. Requires a 'proof of work'

8.9.7 Digital Signing

From the RFC1319 the MD2 Message-Digest algorithm takes an input as a message of arbitrary length and produces an output of 128bit that is used as a '' fingerprint''. This *fingerprint* is also called a digest.

It is conjectured that it is computationally unfeasible to produce two messages that have the same message digest or produce the message given the digest.

The MD2 algorithm is intended for digital signature applications. It achieves in cases where the large file needs to be *compressed* in a secure manner before being signed with a private key under a public-key cyptosystem such as RSA.

8.10 Risk Involved with Leaked Private Key

- Corresponding public key should not be used any more. Should be revoked.
  - ID of certificate is added to a Certificate Revocation List (CRL) which is published by the responsible CA.
- Applications/Protocols should always check current CRLs before accepting a certificate.

8.11 Public Key Infrastructure (PKI)

A PKI is the infrastructure required to make public key cryptography work in a secure manner.

> *"A public-key infrastructure (PKI) is a set of hardware, software, people policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates. ..." - Wikipedia*

The components includes,

- Certificates - Binds identity to public key, typically X.509 format
- Certification Authorities (CAs) - Responsible for the issuing of certificates and the signing of them with digital signatures.
- Certificate Revocation Lists (CRLs) - List of certificates (serial numbers) that should no longer be trusted.

Certificates are used in TLS/SSL to authenticate the server.

8.12 Designing a Cipher

This is as much art as it is science. That is that there is no guarantee when designing a secure safe cipher. Some trial and error is required.

The goal of designing a new cipher is computational security such that the best attack is brute force and no shortcuts can be taken.

We want the length of the key to be the sole determining factor when considering security.

Sadly, there is no possible proof that a cipher is secure against all types of attacks. As such the best bet is to show that it is secure against all *known* attack types and that is can withstand scrutiny of the worlds best cryptographers with no weakness being found.

8.13 Side Channel Attacks

These are attacks on the implementation of the cipher, not on the algorithm itself. They use information gained from the physical implementation of a cryptosystem, such as timing information, power consumption, etc to break the system.

AES implementations have been broken using Side channel attacks

- Based on timing for cache access to look-up tables.

These types of attacks have been successful against a range of crytographic systems.

8.14 Symmetric

The same key is used for encryption and decryption. The security is based in the secrecy of the key.

These are also called secret key ciphers.

8.14.1 Modern Ciphers

Often product ciphers. A product cipher is a composition of ciphers (functions) where each function may be a substitution of transposition operation. Iteration is used to increase security.

Feistal Ciphers are a common class of product ciphers with a very specific structure. They include multiple rounds of the same operation.

Feistal ciphers are named after Horst Feistal. More information can be found in Section 8.14.2.

8.14.2 Feistal Ciphers

These ciphers combine substitution and transposition.

Substitution involves the swapping of the left half with something else (Round function F) which is called an S box. The choice of F determine the security of the cipher (F should be highly non-linear).

Transposition involves the swapping of the halves.

For Feistal ciphers, decryption is the same as encryption but with a reverse order of operations. It should be noted that the reversibility comes from the structure. The function F does not have to be reversible.

8.14.3 DESs

One of the most widely used ciphers for a long time (NIST[5]standard since 1977). It is based on IBMs (Horst Feistal's) Lucifer cipher.

The NSA made some changes, namely the key length of the SBoxes.

DES is a Feistal Cipher (See Section 8.14.2 for more details). It has,

- 16 rounds,
- 64 bits blocks,
- 56 bit key (64 bit key less parity)

The round function $F(R, K)$,

- Expands from 32 bit to 48.
- S-Box: 6-bit input $\rightarrow$ 4 bit output
- Implemented via look-up tables.

No major weaknesses found with brute force being the best option of attack. The issue is however, that the 56 bit key is too short.

This was demonstrated by a series of challenges organised by RSA Security. Brute forcing was achieved in 7 days in 2006 for a cost of $10 000 using COPACOBANA and was also achieved as early as 1997 in 3 months, 1998 in 3 days, 1999 in 22 hours.

For comparison, 128bit key brute forcing has an estimated success time of 90 billion years.

8.14.3.1 Extending the Life of DES

Due to the only issue being with the key length, we can still use it but through passing the plaintext through two levels of DES each with different keys.

However, DES is vulnerable to the "meet-in-the-middle" attack which makes this method not much more secure than single level encryption.

8.14.3.2 Meet-in-the-Middle Attack

$$x = E_{K1}(p) = D_{K2}(c)$$

For any plaintext-ciphertext pair (p, c)

- Calculate $E_{K1}(p)$ for all 2^256 values of $K1$ (and store in a table)
- Calculate $D_{K2}(p)$ for all 2^256 values of $K1$ (and store in a table)
- if $E_{K1}(p) = D_{K2}(c)$ we have a match and we are likely to have found $K1$ and $K2$ for a computational cost of only 2^57 instead of 2^112.
- Trades off computational cost with storage. i.e. Time space tradeoff.

This is known as a plaintext attack.

8.14.4 AESs

Due to the weakness of the key length present in DES, it was supposed to replaced in 1989 and 1994 but was re-certified both times.

In 1998 NST announced the AES development, in 2000 NIST chose AES to replace DES.

[5]National Institute of Standards and Technology

8.14.4.1 Public Selection Process

Criteria,

- Security,
- Cost,
- flexibility,
- patent-free
- efficiency (on a wide range of platforms including 8 bit CPUs)

There were 15 candidates with 5 broken, 5 less good than others, and 5 finalists.

The finalists were,

- Serpent (Ross Anderson, Eli Biham, Lars Knudsen)
- Rijndael (Joan Daemen, Vincent Rijmen)
- Twofish (Counterpane) - Feistel structure
- Mars (IBM) - Feistel structure
- RC6™(RSA Data Security Inc.)- Feistel structure

The winner was Rijndael with a non-Feistel, symmetric block cipher.

- 128-bit blocks
- Key lengths of 128, 192 and 256 bits
- It was also designed to handle additional block sizes and key lengths, however, they were not adopted in the standard.

8.14.4.2 How it Works

Substitution-Permutation (Transposition) Network.

It has different number of rounds for each key length,

- 128 bit key: 10 rounds,
- 193 bit key: 12 rounds,
- 256 bit key: 14 rounds

The operations are performed on a matrix of 4x4 bytes (128 bits) called teh *state*.

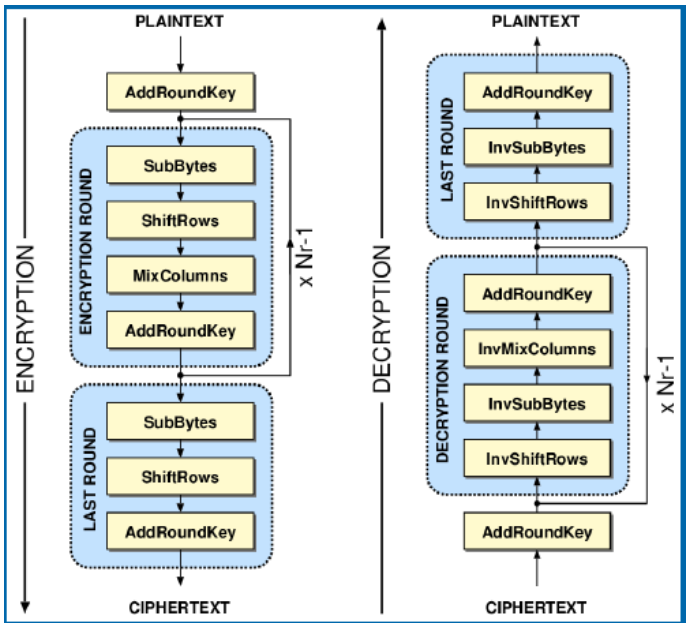


Figure 16: AES Process

The steps,

1. The initial round,
   - AddRoundKey - Every byte of *state* combined with the round key using a bitwise XOR - The round key is derived from the Key according to the Key schedule.
   - Normal rounds,
     - SubBytes - a non-linear substitution step - Each byte of the state is replaced with another, based on a look-up table
     - ShiftRows - a transposition step - Each row of the state is shifted cyclically a certain number of steps.
     - MixColumn - A linear transformation on columns
     - AddRoundKey - Combine with the round key - this is the same as the initial round
   - Final round - This is the same as a normal round but without the MixColumns step.

8.14.4.3 Operations

The choice of transformations such as S-boxes and permutations in AES are carefully chosen based on mathematical properties of computation in Finite Fields or Galois fields.

8.14.4.4 Security

In 2003, US Government state that AES can be used to protect classified information.

- AES-128 for up to SECRET
- AES-192 or AES-256 for TOP SECRET

To date there has been no practical attack against AES found. It has been known to be vulnerable to side channel attacks same as all ciphers (see Section 8.13 for more details).

8.14.4.5 Performance

Newer generation CPUs have specific AES instructions (AMD and Intel).

8.14.5 Message Authenticate Codes (MAC)

8.14.5.1 Efficient Authentication/Integrity

We assume we want to provide authentication and integrity for packets in a secure network protocol i.e. TLS.

We can achieve this using a digital signature i.e. SHA-2 + RSA. **Problems** exist with this method as public key cryptography is relatively expensive. Especially if it has to be done per packet.

We can do this more **efficiently** using a secret-key algorithm.

Authentication/Integrity with Secret-key Cryptography

We can achieve this through the use of a one-way hash function and a secret key.

For example.

1. Assume Alice and Bob share a secret key K
2. Alice sends a message to m to Bob. (Note: we want authentication and integrity)
3. Basic idea $\rightarrow$ Alice computes a cryptographic checksum or Message Authentication Code (MAC). MAC $= h(K||m)$
4. If Trudy alters m, she cannot compute a valid MAC without knowing $K \Rightarrow$ This is how we achieve Integrity.
5. Knowing K, Bob can verify the MAC. Only someone knowing K i.e. Alice. would have been able to compute a valid MAC. $\Rightarrow$ This is how we achieve Authentication.

8.14.5.2 HMAC

Now a simple MAC $= h(k||m)$ is not secure for a hash functions such as SHA-1, SHA-2 or md5 based on the so-called *Merkle-Damgard* constructing. Hashes created with those algorithms are susceptible to 'length extension attacks'[6].

[6] New SHA-3 (Keccak) is NOT vulnerable to the length extension attack. Therefore a simple MAC = SHA-3($K||m$) is secure.

A more complex, nested version of MAC is used to fix this vulnerability. It is called HMAC.

It used Keyed hashing for Message Authentication and is the most widely used MAC in the Internet. IETF Standard RFC2104.

$$\text{HMAC}(K, m) = H((K' \oplus \text{opad})||H((K' \oplus \text{ipad})||m))$$

- K': Keyed hashed or padded to blocksize
- opad: Outer padding, constant 0x5c
- ipad: Inner padding, constant 0x36
- $\oplus$: XOR
- $||$: Concatenation.

Unlike standard MAC, HMAC does not rely on the collision resistance of the hash algorithm, so it is secure even with weak hash functions such as md5 or SHA-1.

8.14.6 Block Ciphers vs. Stream Ciphers

Most ciphers are block ciphers (with the exception of the one time pad).

A block cipher works on fixed-sized plaintext blocks (64 bits, 128 bits, etc) and produces blocks of ciphertext of the same size).

A stream cipher works on smaller units of plaintext (bits or bytes). It generates a pseudorandom key stream (difference to OTP, where the stream is random).

Encryption is typically done by XORing the key stream with the plaintext.

The key stream is typically generated via a feedback mechanism using shift-registers.

The key stream can be denoted with S where S is a function of a secret key k.

Encryption: $C = P \oplus S$ (bitwise XOR)

Decryption: $P = C \oplus S$

Stream based ciphers are not perfect security since the key stream is not truly random (unlike that of the OTP) and will eventually be repeated.

8.14.6.1 Stream Cipher RC4

The R stands for RSA, it is the most widely used stream cipher (i.e MS Word, Excel, WEP etc).

It is both simple and elegant in that it can be implemented in a few dozen lines of code. It is not completely secure as some vulnerabilities have been discovered in the past few years. AES is a better choice.

RC4 based cryptosystems,

- WEP
- TKIP (WPA/WPA2 option)
- BitTorrent protocol encryption
- Microsoft P2P encryption
- SSL (option)
- SSH (option)
- RDP
- PDF

8.14.6.2 Encryption Modes of Block Ciphers

Streams of data can be encrypted using different Block cipher encryption modes.

Electronic Code Book Mode (ECB)

The same plaintext blocks will always result in the same ciphertext block (no dependencies between blocks).

This makes it vulnerable to replay attacks. Additionally, ciphertext can be reordered and dictionary attacks can also be used (note: this is not feasible for large block sizes).

It has the advantages of being simple, fast to implement, limited error propagation.

It is not considered secure due to its susceptibility to insertion, replay, reordering, dictionary attacks.

These issues can be solved through the use of a feedback mechanisms. That is where the result of a previous block is fed back into the encryption of the current block.

This allow the introduction of dependencies between blocks where the ciphertext blocks depend not just on the plaintext blocks but also on the previous ciphertext blocks.

The different ways of using feedback exist in CBC, CFB, and OFB.

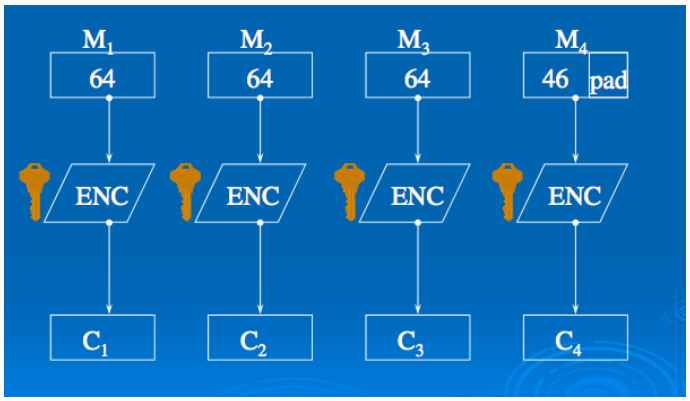


Figure 17: ECB Cipher Mode

Cipher Block Chaining Mode (CBC)

The same input results in different output with high probability.

The initialisation vector (IV) does not need to be secret - it is just a dummy ciphertext block to start with. This should be changed frequently as the encryption of the same plaintext with the same key and IV still results in the same ciphertext.

Each block of the ciphertext depends on the current plaintext block, previous ciphertext block and the IV.

It prevents reordering attack (cannot reorder ciphertext), insertion attack, replay attack (the same plaintext can result in different ciphertext) and dictionary attacks (same as replay attacks).

Transmission errors are handled differently to ECB. An error in a bit results in the loss of a complete ciphertext block. That is, a single plaintext block is lost. Synchronisation occurs after that.

When a bit error occurs, the block that it belongs to is lost, the next block has a single bit error in the same position. This could allow an attacker to predict bit changes in plaintext blocks. Further plaintext blocks are not effected.

CBC is self recovering from bit errors.

Encryption of multiple blocks cannot be done in parallel due to the dependencies.

It is the most commonly used cipher mode and is supported by all IPsec and TLS/SSL implementations.

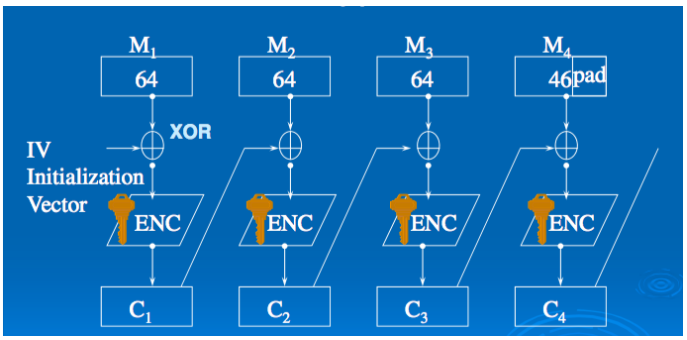


Figure 18: CBC Cipher Mode Encryption

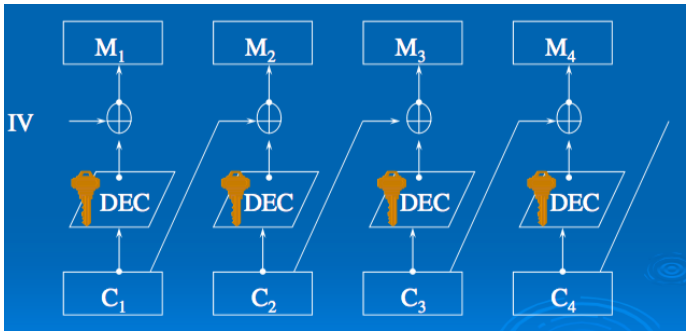


Figure 19: CBC Cipher Mode Decryption

Cipher Feedback Mode (CFB)

Error propagation is again different. A single bit error would result multiple blocks being lost. The number of blocks lost is dependant on the block size and key size. i.e

$$\text{blocks lost} = \frac{\text{block size}}{\text{key size}}$$

Synchronisation allows CFB to recover from whole blocks from being deleted from the ciphertext stream (thank goodness). This is done in the same way that is handles bit errors. It is self synchronising at the block level similar to CBC.

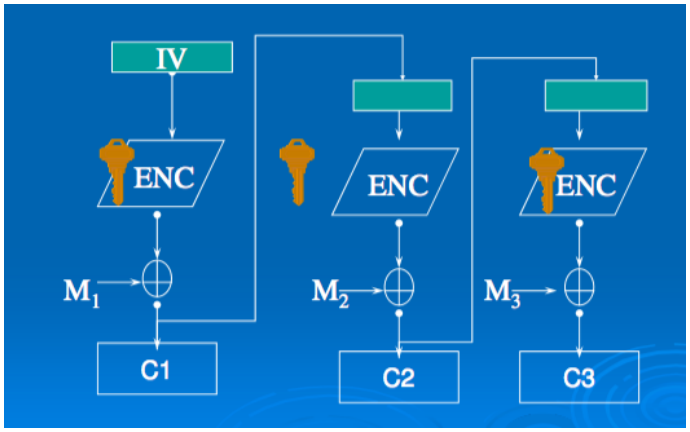


Figure 20: CFB Cipher Mode

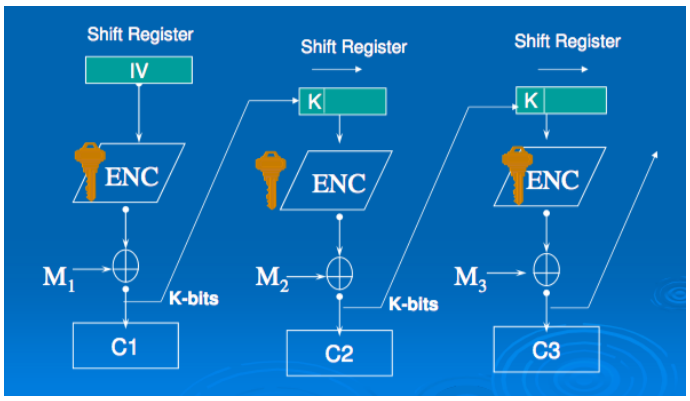


Figure 21: CFB Cipher Mode General K Bit

Output Feedback Mode (OFB)

For OFB, the IV needs to be unique but does not need to be kept secret. This mode uses a block cipher to in stream cipher mode. It is similar to OTP[7]OTPs in this aspect.

Advantages of this mode include, no error propagation (or error extension) as in CBC. That is a single bit error in plaintext will result in a single bit error in the ciphertext. This property makes it useful for video etc. Additionally, it allows for the pre-computing of pseudo random streams. The use of XOR can be implemented very efficiently (works like a cipher stream).

It does not need full blocks to start encryption making it useful for terminal applications.

Issues do exist such as its lack of self synchronisation. It cannot recover from the loss of entire cipherblocks.

While a single bit error in the plaintext input of a single block only results in a single bit error in the ciphertext, a single bit error in the input data to a block cipher will result in the complete output being affected. This is a desirable property as it means a single bit change will result in a different output.

More formally, a function is said to satisfy the strict avalanche criterion if, whenever a single input bit is flipped, each of the output bits should change with a probability of one half.

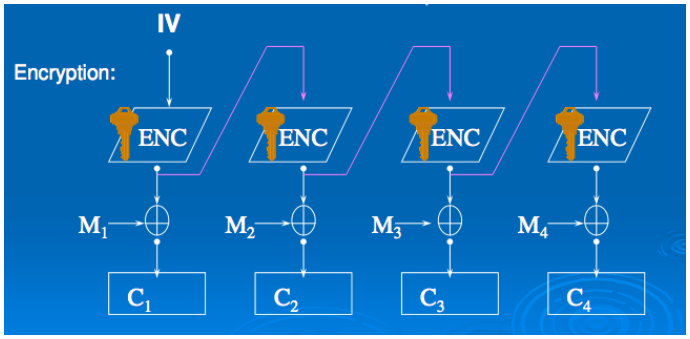


Figure 22: OFB Cipher Mode

[7] One-time pad.

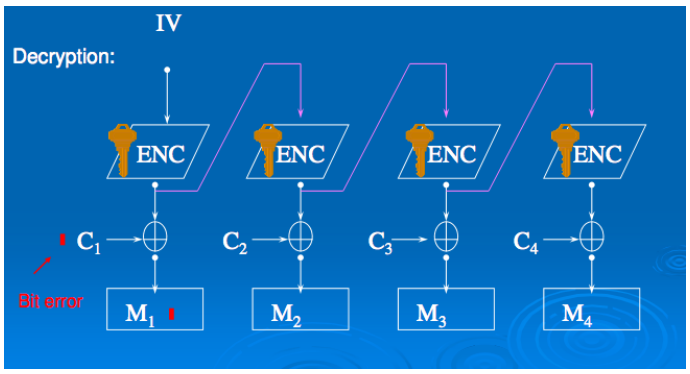


Figure 23: OFB Error Propagation Decryption

8.15 Asymmetric

8.15.1 Public Key Cryptography

Invented in the 1970s it was the major breakthrough in cryptography in the last 2000 years or so. The basic idea was outlined by Diffie and Hellman with the following definitions being used.

One-way function Easy to compute and hard to inverse the reverse of. Not a new idea (think one-way hash function)

Trapdoor One-way Function A new concept (mathematical padlock), the function is one-way except if the secret trapdoor is known.

It is important to note that Diffie and Hellman did not find a Trapdoor one-way function, they did however, find a one-way function and showed how it can be used in establishing a secret key (symmetric key encryption).

The main idea is that the encryption key $\neq$ the decryption key.

The benefits include the ability to make the encryption key public which solves the issue with key distribution. At least partially.

The ability to have a one-way operation removes the need for a secure channel (confidentiality) as only the owner of the corresponding key can decrypt it.

The first publicly known practical public key cryptography cryptosystem was RSA (see Section 8.15.2 for details).

Public key crypography solves the key distribution problem however, it is orders of magnitude slower than symmetric key cryptography so hybrid systems exist.

These include systems that use Public key cryptography to establish a symmetric secret or session keys. This is found it SSL/TLS.

8.15.1.1 Modular Arithmetic

Most public key algorithms use modular integer arithmetic. That is numbers that wrap around.

Examples include,

- $(9 + 8) \bmod 13 = 4$
- $(3 \times 6) \bmod 13 = 5$

Useful properties include,

- $(a + b) \bmod n = (a \bmod n + b \bmod n) \bmod n$
- $(a \times b) \bmod n = (a \bmod n \times b \mod n) \bmod n$
- $(a + c \times n) \bmod n = (a \bmod n + b \bmod n) \mod n$ for all $c \Rightarrow$ These numbers are congruent.

$\oplus$ **is addition and subtraction mod 2.**

8.15.1.2 Modular Exponentiation

Normal Exponentiation rules apply i.e $(g^x)^y = (g^y)^x = g^{xy}$ and $g^x \times g^y = g^{x+y}$

Modular exponentiation can be very efficiently implemented via the "square and multiply" algorithm with a running time of $O(log_2 n)$ with n being the size of the exponent.

8.15.1.3 Discrete Logarithm

The inverse of modular exponentiation is the "Discrete Logarithm".

i.e $7^? \bmod 9 = 7 \Rightarrow log_7 7 \bmod 9 = 4$

It is guaranteed that the discrete logarithm $log_a c mod n$ has a unique solution for all c where n is a prime number and a is a generator (of the Cyclic Group of integers modulo n)

- a is a generator if by computing ab for b = {1, .. n-1}, all elements 1,2..n-1 are *generated* .(This means that for all c the log exists.)

Number theory tells us that a is a generator if $gcd(a, n-1) = 1$, this allows us to directly check this without trial and error.

The computation is not currently able to do it efficiently (there is no known efficient algorithm for solving discrete logarithms for large n).

Since there is know currently known efficient function it is assumed that none exists (we cannot know for sure). This is the Diffie-Hellman-Phlig conjecture (not a theorem as there is no proof).

Modular exponentiation is a one-way function. Not a trapdoor.

8.15.1.4 Diffie-Hellman Key Agreement Protocol

Say there are two parties. Both initially agree on a large prime number p and a generator g (all calculations are done to modulo p).

The first person (P_1) chooses a random number x, they then computer $X = g^x$ then sends it to the second person (P_2). The other person chooses a random number y and computes $Y = g^y$ and sends it back to the first person.

P_1 then computes $K_{AB} = Y^x = ((g^y)^x \bmod p = g^{xy} \bmod p$. P_2 then computes $K_{AB} = X^y = ((g^x)^y \bmod p = g^{xy} \bmod p$.

If a third person (P_3) is listening in on the insecure channel they will see g^x and g^y but since they won't be able to calculate the discrete logarithms they won't be able to find out the values for x or y and therefore won't be able to calculate the secret K_{AB}.

The issue apparent with this method is that it does not contain any authentication, therefore anyone can be pretend to be P_1 or P_2.

The Diffie-Hellman protocol allows for the establishment of shared secret keys but is not a public key encryption algorithm.

The "discrete logarithm" problem provides us a means of creating a one-way function but not a **trapdoor one-way function**. This functionality is provided with RSA (see Section 8.15.2).

8.15.1.5 Public Key Crypto Systems

Most of Public Key systems rely on either factoring or discrete logarithm problems being hard.

For example, the El Gamal public key encryption system is based on the discrete logarithm problem.

8.15.1.6 Public Key Certificates

In the case that an attacker pretends to be someone else, they can trick receivers into thinking a key created by them belong to someone else. This will allow them access to all information sent as it would

be sent to them directly under the assumption that it was being sent to someone else.

This is an attack on privacy and authenticity and brings up the concept of trust with regards to the key - identity pair.

This is done through the use of digital certificates.

If a third party verifies the authenticity of a certificate and we trust them we can assume that the identity-key pair that has been verified is valid. However, we also need the ability to verify the third parties identity, this is solved through **certificate chains**.

Since a chain has to start somewhere **Certificate Authorities (CA)**. These exist as part of the chain as well with top level CA's being called **Root CAs** and act as a trust anchor. This trust is bootstrapped through other means such as manual (by user) or automatic (installed by default) installation in the web browsers certificate store.

More details can be found in Section 3.

8.15.2 RSAs

The first known public key cipher.

8.15.2.1 How it Works

We make use of modular arithmetic. That is mod n. (P_1 and P_1 represent person one and person two respectively).

1. P_1 chooses two large (> 100 digit) primes p and q,
2. P_1 then computes $n = p * q$.
3. P_1 computes $z = (p-1) * (q-1)$ - *Euler's Totient function*
4. P_1 select an exponent e that has no common factors with z (g and z are relatively prime [8] or $gcd(e, z) = 1$.
5. P_1 keeps p and q secret but sends P_2 (n, e) i.e P_1s public key.
6. P_2 can encrypt a message m as follows, $c = m^e \mod n$ - Modular exponentiation is easy!
7. P_1 computes d such that $c^d = m \mod n$, that is $c^d = (m^e)^d \mod n = m^{ed} \mod n = m$
8. d is the secret key that allows decryption (trapdoor).

Decryption - Finding d

How will P_1 find the secret d that allows for decryption? That is $(m^e)^d \mod n = m$.

d can be found if the prime factors p and q of n are known. That is find d such that $e * d \mod z = 1$, $z = (p-1)(q-1)$. An efficient algorithm for solving this is the "*Extended Euclid's Algorithm*".

This is secure as there is now known (currently) efficient way to find d for large values of n without also knowing both p and q, the prime factors of n. There is no known efficient way to invert the encryption $m^e \mod n$ without the trapdoor d.

Now that P_1 has solved for d, p and q can be discarded as long as they are never revealed.

The exponentiation of $m^e \mod n$ is a trapdoor one way function with d being the trapdoor.

This is a result of factoring being considered a difficult problem in the same way that computing Discrete Logarithms is considered difficult.

8.15.2.2 Security

It has been proven that breaking RSA is the equivalent of solving the age old factoring problem. While not impossible, it is unlikely given that both cryptographers and mathematicians have both tried unsuccessfully for centuries to solve this problem.

Key Length

According to RSA Security (2003) a 1024 bit RSA key is the equivalent in strength to an 80 bit key of a symmetric cipher. That is that there is no need to brute force a 1024 bit key space as in AES, you would need to factor a 1024 bit integer. While this is still hard, it is not as hard as brute forcing a 128 bit key.

768 bit RSA keys have been broken. Currently there is a 100000 reward for breaking a 1024 bit RSA key.

Other equivalents include,

- 2048 bit RSA $\approx$ 112 bit symmetric key
- 3072 bit RSA $\approx$ 128 bit symmetric key

The security of RSA is based around the idea that factoring is hard.

This does not bode well for RSA with emerging technologies as using Shor's algorithm and a quantum computer, integers can be factored easily.

8.15.2.3 RSA Signatures

Encryption and decryption are essentially the same operation and can be applied in any order(*commutativity*). i.e

$$(m^e)^d \mod n = (m^d)^e \mod n = x^{ed} \mod n$$

That is to say that a person can encrypt a message using their private key and that same message could be decrypted by the public key. i.e $c = m^d \mod n$ (encryption) $\rightarrow m = c^e \mod n$ (encryption).

This is used in **Digital Signatures** as anybody can decrypt the message using the public key and verify that it is in fact a valid message (this is misses out some redundancy). This is due to the requirement of someone knowing the private key to be able to encrypt it in the first place.

This has the benefit of providing authentication and integrity.

The assumption is that only the authorised person has access to the private key. That is that the private key is still private.

Providing Signatures for Large Files

Let the file be f. Since the process of modular exponentiation is slow with large numbers we cannot realistically perform standard encryption on the whole file as it would take too long with the resulting signature being too large.

What we do instead is make use of a hash function $h()$ to compute $h(f)$ into a 128 bit value as hashing is cheap. We can now use this hash to sign (encrypt with the private key) in stead of the entire file f.

$$s = (h(f))^d \mod n$$

To verify a signature created in this manner S_R we first decrypt the hash using the public key $\rightarrow S_R^e \mod n = h_R(f)$.

We then compute the hash of the file $h(f)$ and compare the two values, if they match we have successfully verified the signature. This allows for authentication (sent by the person we think it was) and non-repudiation as well as integrity (no tampering).

This method obviously relies on several factors. Beyond the difficulty of factorisation that has already been mentioned with RSA, it also relies on the fact that an attacker cannot find a hash collision.

8.16 TLS/SSL

TLSs: Transport layer Security
SSLs: Secure Socket Layer

[8] Greatest common divisor.

8.16.1 History

SSL developed by Netscape in 1994 with versions 1.0 - 3.0. The goal was to provide authentication, integrity and confidentiality of communication through a web browser and server. The design is generic, SSL/TLS can be used with any TCP-based application.

SSL was adopted by the IETF (Internet Engineering Task Force) as a standard with minor modifications (TLS).

TLS,

- V1.0 Based on SSL 3.0 (but not interoperable) RFC2246, in 1999
- V1.1 RFC4346 2006 - Fixes to a few weaknesses,
- V1.2 RFC5246b 2008 - Added more secure hash functions i.e SHA-256

The names TLS and SSL are often used interchangeably with TLS being the most widely used security protocol.

8.16.2 TLS

It exists between the application and the reliable (TCP) transport layer. While TLS is most commonly used for web traffic (HTTP ⇒ HTTPS) it can be used for any other applications.

Most programming languages provided support for TLS/SSL.

TLS consists of the following parts,

- Handshake protocol - Establishes a shared secret key, negotiates cipher suite[9].
- Cipher change protocol - Enables a cipher change.
- Alert protocol - Reports errors
- Record Protocol - Main part, it provides secure transport.

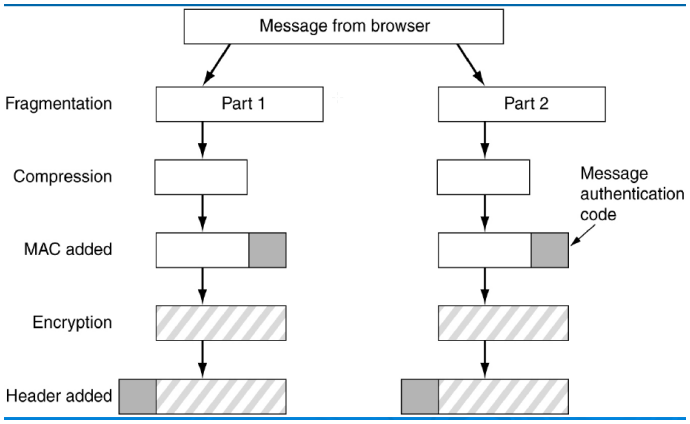


Figure 24: TLS Record Protocol

The padlock and https are both indicators that TLS is active in the browser. HTTPS is commonly server on port 443 vs the regular HTTP port 80.

It provides,

- Key establishment,
- Authentication
- Confidentiality
- Integrity.

TLS makes use of cryptographic hash functions, secret-key ciphers and public key ciphers.

8.16.2.1 TLS Handshake

The initial phase of a TLS session which has the purpose of negotiation of the cipher suite to be used. Mutual authentication of server (and client) is achieved in this phase.

Authentication of the server if required whereas the authentication of the client is optional. In most cases the negotiation is done through the use of public key cryptography and via the exchange of X509 certificates or certificate chains.

During this stage, shared secret keys for encryption and authentication and MAC are established. After the handshake all data that is sent via the TLS connection is encrypted as such integrity is provided through HMAC.

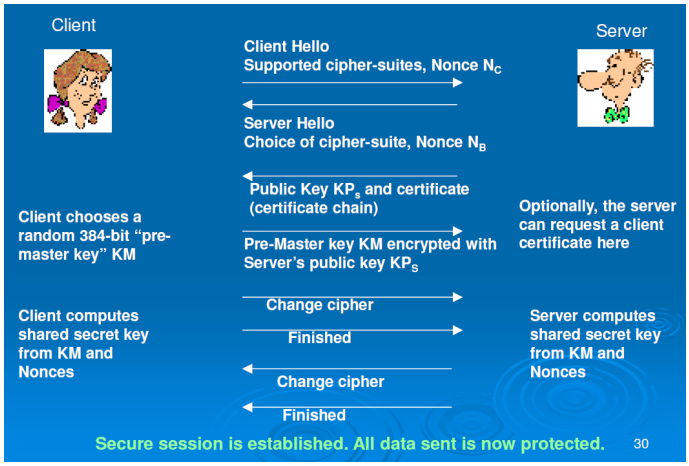


Figure 25: SSl/TLS Handshake

The `ChangeCipherSpec` message in TLS v1.2 consists of a single byte of value 1. The clients finished message contains SHA256 HMAC of the master_secret and "client finished" concatenated with the hash of the handshake so far.

In the server finished message, A SHA256 HMAC of the master_secret and "server finished" concatenated with the hash of all the handshake messages so far.

The `ChangeCipherSpec` and the finished messages are always sent together to allow for the verification of the successful completion of the key exchange and the authentication process.

8.16.2.2 Client Authentication

The server authenticates to the client using a server certificate. The client/user can authenticate to the server (remember this is not always required) using a certificate the same as the server or using a username and password. This is all sent over a secure encrypted TLS connection.

8.16.2.3 Man-in-the-Middle Attack (MITM)

We can assume the attacker can redirect a browser request to a proxy that they control. This can be done through DNS poising or ARP spoofing.

In the case of a MITM attack against a TLS secured connection, it is possible if DH_anon (plain Diffie-Hellman) is used for key Establishment.

[9] A combination of specific algorithms to be used in a TLS session

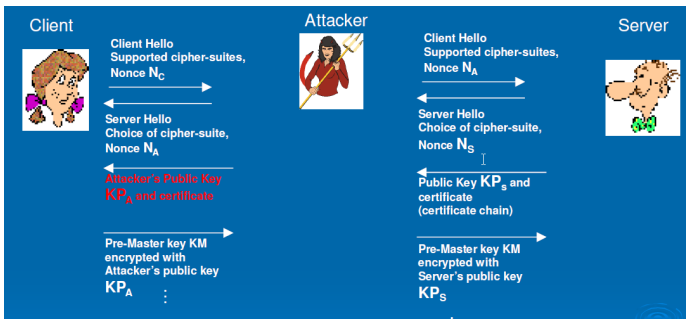


Figure 26: SSL Man-in-the-Middle Attack

The MITM attack shown in Figure 26 will be successful if the attacker can find a way to redirect the clients request with either DNS poisoning or ARP spoofing.

8.16.2.4 Recent Vulnerabilities in SSL/TLS

- HeartBleed - Implementation bug in OpenSSL - Missing bounds check in TLS heart bleed extension.
- POODLE - Design bug in SSL3.0. It is a problem with padding used CBC cipher mode. Even though not many browsers use SSL3.0, they happily downgrade during handshake (MITM rollback attack).

9 Network Security

9.1 (ISC)$^{2®}$ CISSP® Domains

Consists of eight domains (used to be 10).

- Security and Risk Management
- Asset Security
- Security Engineering (Physical + crypto)
- Communication and network security
- Identity and access management
- Security operations
- Security assessment and testing
- Software development security

9.2 OSI and TCP/IP Models

9.2.1 OSI

Layer	Description
Application	May add data encryption
Presentation	Defines ASCII, EBCDIC, MIDI, MPEG, PICT and GIF
Session	SQL, NFS, RPC, NetBIOS are session layer protocols
Transport	Provided flow control and error recovery (TCP/UDP)
Network	Logical addressing and end-to-end delivery of packets
Data Link	Translates data into frames and adds a CRC
Physical	Encodes and transmits data bits.

Table 3: ISO's OSI Model

9.2.2 TCP/IP

Invented by Dr. Robert E. Kahn and Dr. Vinton Cerf.

Layer	Mapping (NOTE: Not exact!)
Application	Application, presentation, session
Transport (host-to-host)	Transport
Network (Internet)	Network
Link	Data link (Network access)
Physical	Physical (Network access)

Table 4: TCP/IP Model and How it Maps to OSI

9.2.3 Physical Layer Security

Two primary groups of mediums.

1. Wired,
2. Wireless

Wired	Wireless
Limit access to, • jacks, • wires, • patch panels.	All access is in the “air”.

Table 5: Physical Security Wired vs. Wireless

9.2.3.1 Confidentiality

Access to the medium (see Table 5). Medium access is just volts on a wire and gives us promiscuous mode interfaces [10] and allows packet sniffing.

9.2.3.2 Integrity

Signal can be delayed or reflected.

9.2.4 Data Link Layer Security

Circuit switched networks - PSTN [11] - Not any more!

Packet switched networks - IEEE 802 standards are divided the data link layer into two sub layers.

1. Logical Link control (LLC) layer
2. Media Access Control (MAC) layer.

9.2.4.1 Media Access Control (MAC)

- Provides carrier sense multiple access.
- CSMA/Collision avoidance (CSMA/CA)
- CSMA/Collision Detectio (CSMA/CD)

9.3 IEEE 802.11 WLAN Security

9.3.1 Attraction

- Allows devices to move about with freedom
- Greater convenience than cables
- Significant reduce time and resources required to setup new networks.
- Easy modification of networks
- Allows for networks in difficult locations
- Allows for ad-hoc networks, easily created, modified, torn-down

9.3.2 Characteristics

- Electromagnetic broadcast technology
- Transmission is to the universe
- Compromise confidentiality.
- Fragile availability
- Threat to integrity.

Has two physical interfaces

1. Infra-red - up to 2Mbps in the THz range.
2. Radio - Up to 2 Mbps FHSS in 2.4GHz band, industrial, scientific and medical bands (ISM) 2.400 - 2.500 GHz.

Standards	Description
802.11	Up to 2 Mbps FHSS in 2.4GHz band
802.11a	Up to 54Mbps OFDM in 5GHz band
802.11b	Up to 11 Mbps DSSS in 2.4GHz band
TGc	Provided required information for bridge operations
802.11d	Additional regulatory domains (roaming)
802.11e	Quality of Service (QoS)
802.11F	**WITHDRAWN!** - RP for AP Interoperability
802.11g	Up to 54 Mbps in the 2.4GHz band - OFDM above 20Mbps, DSSS below 20Mbps
802.11h	Spectrum and power management in 5GHz band
802.11i	MAC Security Enhancements
802.11j	4.9 GHz - 5GHz operation in Japan
802.11 - 2007	New Release (ma) of the standard with amendments a, b, d, e, g, h, i and j (July 2007)
802.11k	Radio Resource Measurement (2008)
802.11m, ma, mb	Maintenance of IEE Std 802.11
802.11n	Higher throughput using MINO (multiple input, multiple out antennas) September 2009.
802.11p	WAVE: WLess Access for Veh Env (July 2010)
802.11r	Fast BSS Transmission for VoFi (2008)
802.11s	Mesh Networking, Extended service set

Table 6: 802.11 Standards

9.3.2.1 DSS - Direct Sequence Spread Spectrum

DSS is a spread spectrum modulation technique that is used to reduce signal interference.

[10] Allow the network interface cards (NIC) to accept all traffic even if it is not addressed to the given NIC.

[11] Public Switched Telephone Network, our normal telephone network.

9.3.2.2 WLAN Channels

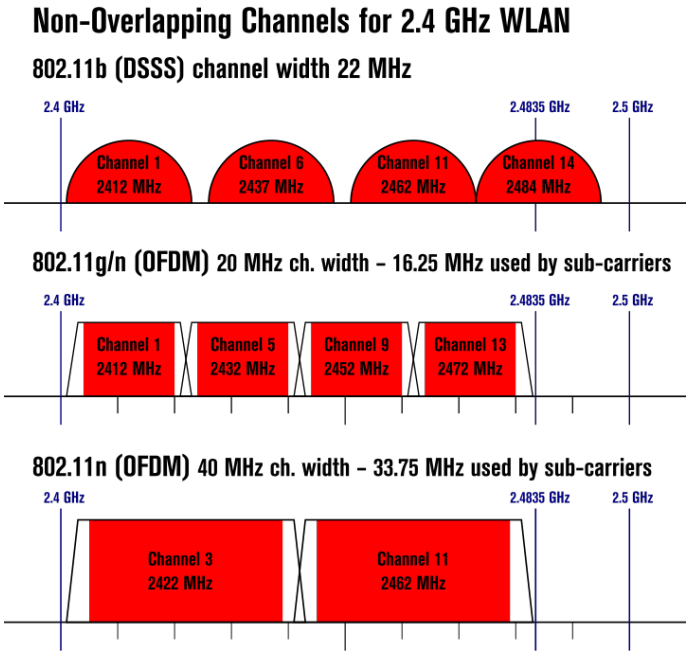


Figure 27: Non-Overlapping Channels for Most Countries

It should be noted that Channel 14 is not allowed in Australia.

See Wikipedia for details.

To minimise interferance, pick channels that do not overlap.

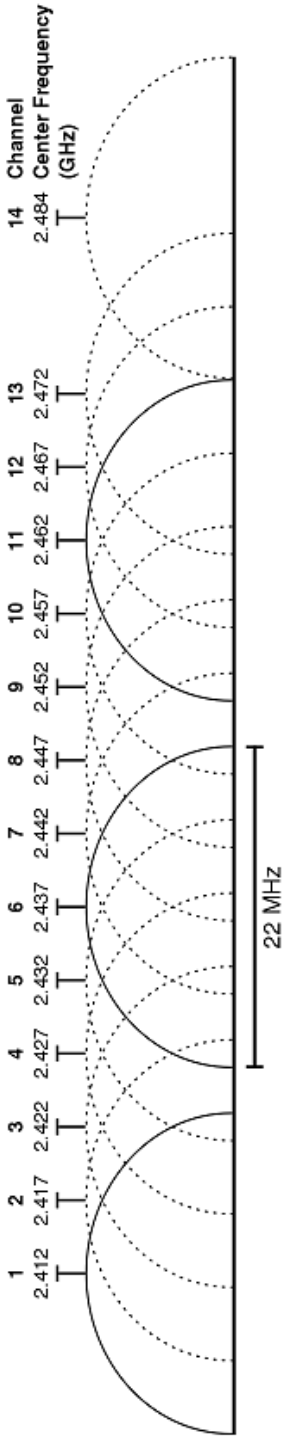


Figure 28: Overlapping Channels for Most Countries

9.3.2.3 IEEE 802.11a OFDM Channels

- Three bands, but only one is internal ISM [12]
- Two lower bands are USA specific UNII (Unlicensed National Information Infrastructure).
- 5.2 GHz band (5150 - 5350 MHz)
- In Australia these can be used under the Low Interference Potential Devices Class license.
- Upper band is 5.8 GHz (5725 - 5825 MHz) ISM.
- USA UNII Lower band - Channels 36, 40, 44, 48
- USA UNII Middle band - Channels 52, 56, 60, 64
- USA UNI Upper band (ISM 5.8 GHz band) - Channels 149, 153, 157, 161.

[12] International Relations and Security Network?

9.3.2.4 IEEE 802.11n MIMO

t x r : c

t: The number of transmit antennae
r: The number of receive antennae
c: The number of spatial channels

The current gear is 2x2:2, 2x3:2, 3x3:2 - all the same

Its' range is from 100Mbps to a max of 300Mbps depending on the band size and Guard interval. Full certification is 3x3:3 up to 450Mbps whereas the actual standard goes up to 4x4:4 up to 600 Mbps.

9.3.2.5 IEEE 802.11i MAC Security

This standard took a long time to be ratified with parts starting to be implemented as early as the draft of IEEE 802.11i/D3. It resolves the issues with confidentiality and integrity including those of authentication of all the data frames. Sadly management frames still remain unprotected.

It uses IEEE802.1X for authentication, specifies key management algorithms and two data encapsulation methods.

IEEE 802.1X use in IEEE 802.11i

- Supplicant, authenticator, authentication server
- IEEE 802.1X is used between client NIC and AP - The protocol between AP and the AS is typically the Remote Authentication Dial-In User Service (RADIUS)
- Two way authentication
- Protocol negotiation and key exchanges - the authentication and negotiation parameters verified again as part of the key exchanges.

9.3.2.6 IEEE 802.11w

- Provides some protected management frames
- Provides mechanism that enable, data integrity, data origin authenticity, replay protection, data confidentiality for some selected management frames.
- Provides protection against injection attacks, disassociation attacks, Fake APs.
- Ratified September 2009 but not advertised. Except for CISCO

9.3.2.7 Robust Security Network (RSN)

- Allows the creation of robust security network associations (RSNAs) only.
- An associated exists only if the procedure to establish authentication or association between them includes the 4-way handshake.

The Four-Way Handshake

Messages,

1. Authenticator nonce, seq, PMK-id → PTK= Hash(PMKm ANonce, SNonce, supplicants MAC address, AP's MAC address)
2. Supplicant nonce, RSN IE, seq, MIC → PTK = Hash(PMK, ANonce, SNonce, supplicants MAC address, APs MAC address)
3. ANonce, RSN IE, GTK, Seq+1, MICE
4. Seq+1, MIC

RSNA Requirements

- Protocol information statement conformance statement (PICS)
- RSNA optional but if implemented requires,
  - RSN Information Element (IE)
  - Implementation of CCMP
  - A RSN only allows RSNAs
  - RSNAs can use either TKIP or CCMP.

9.3.2.8 RSN and TSN

- AN AP in a RSN must not associate with any pre-RSNA STAs (no RSA IE in the Association request)
- An AP must include the RSN IE in the Beacon frames showing group cipher CCMP of TKIP but not WEP
- A network that allows creation of pre-RSNAs as well as RSNAs is a *transition security network (TSN)*, identified by RSN IE showing the group cipher WEP

9.3.2.9 Cipher Suite Combinations in RSN IE

Pairwise	Groupwise	RSN/TSN
CCMP	CCMP	RSN
CCMP	TKIP	RSN
CCMP, TKIP	TKIP	RSN
TKIP	TKIP	RSN
CCMP	WEP-40 or WEP-104	TSN
CCMP, TKIP	WEP-40 or WEP-104	TSN
TKIP	WEP-40 or WEP-104	TSN
UseGroup	WEP-40 or WEP-104	TSN

Table 7: Cipher Suite Combinations in RSN IE

9.3.2.10 Wi-Fi Protected Access (WPA)

- Snapshot of the IEEE 802.11i/D3 draft - Addressed infrastructure mode WEP vulnerabilities - "WPA is not available in ad-hoc mode"
- Uses the temporal Key Integrity Protocol (TKIP)
- WPA Personal Mode offer only pre-shared key
- WPA Enterprise Mode Offer both PSK and IEEE 802.1X/EAP authentication
- IEEE 802.11i Transition Security Network (TSN)
- It doesn't use TKIP for both pairwise and groupwise - It forms a TSN
- It does not have to implement the mandatory CCMP
- It does not use the RSN IE (WPA IE is different)
- Does not use RSNAs (WPA four-way handshake is different to IEEE 802.11i)
- Although it is not supposed to service a mixture of WEP and WPA, many vendors do as an added bonus.

WPA "Four-Way Handshake"

- WPA cannot establish with just the four-way handshake
- Must also perform an immediate GTK handshake [13]

Messages,

1. Authenticator nonce, seq, PMK-id → PTK= Hash(PMKm ANonce, SNonce, supplicants MAC address, AP's MAC address)
2. Supplicant nonce, IE ***(NOTE: No RSN IE)***, seq, MIC → PTK = Hash(PMK, ANonce, SNonce, supplicants MAC address, APs MAC address)
3. ANonce, IE ***(NOTE: No RSN IE)***, GTK, Seq+1, MIC
4. Seq+1, MIC

WPA2

[13] Group Temporal Key.

- Implements IEEE 802.11i but differs to allow for interoperability with WPA
- Available in both infrastructure mode and ad-hoc mode
- WPA2 provides both TKIP and AES-CCMP
- An access point and client card running only CCMP in WPA2 will be running an IEEE 80211i RSN.
- An AP that allows WPA clients will be running a TSN.

WPA2 Implementation of TSN

- TKIP+CCMP often WPA compatible
- Wi-Fi certifies in default configuration
- WPA2 mode doesn't allow simultaneous WEP
- Each Wi-Fi Certified device must support WEP
- Many vendors permit a mixture of WPA2/WPA or WPA/WEP or even all three at once.

There are **risks with using WPA2/WPA with TSN**. TSN is weaker than RSN. Since WPA-PSK (or WPA2-PSK) PMK is a hash of the PSK passphrase - the use of weak passphrases weaken the key strength considerably.

PSK dictionary passwords are all secure.

There are also **risks** with using **WPA2/WEP TSN**. Network strength reduced to that of WEP where,

- Packets can be passively captured in hours for attack,
- Packets can be actively captured for attacks in minutes,
- FMS attacks can be done in hours or days
- ARP packets can be actively captured in minutes
- PTW attacks can be achieved in seconds,

Mixing in WEP destroys all of WPA/WPA2 security.

Transition Risks

Running of mixed modes introduces risks. To fix this we should require the last of the legacy equipment, ensure all devices use CCMP.

Even then we do not have a RSN as we must reconfigure to refuse pre-RSNAs. This fails if an old computer fires up with WEP and allows an attacker to capture ARPs recover the key.

WPA	WPA2
• Only certified for infrastructure, not ad-hoc, • Only TKIP certified, but many use AES • No use of PMK caching or pre-authorisation • 4-way handshake is different	• Must provide both TKIP and AES-CCMP • IE and RSN (WPA2) RSN IE are different

Table 8: WPA vs. WPA2

WPA2-PSK

- 64 hex characters = 256 bits
- 8-63 printable ASCII characters = 64-504 bits (these are hashed to 256 bits)
- Many say at least 20 characters required to be secure.

9.3.3 KRACK

KRACK is a wireless security vulnerability that affects WPA and WPA2.

It is one of the 10 CVE's (common vulnerabilities and exposures) that have been released. (9 of these are client based and one is infrastructure based).

We are lucky that all 10 are implementation issues meaning that patches can be released to fix the issues.

The four-way handshake is prone to replay attacks (man-in-the-middle (MITM)) in the case where a user is de-authenticated from the network allowing them to connect to the MITM AP.

If a replay of message three occurs, the same key will be set back in place but all the counters will be reinitialised, therefore no nonces can be reused. Known plaintext can be used to determine the key stream and thus decrypt the ciphertext.

9.3.4 Management Frames

Working groups considerably extending the functionality of management frames to include sensistive information such as

- radio resource data,
- location based IDs
- fast-roaming information
- wireless network management

Security in wireless networks neet to be extended to management frames as well as dataframes.

9.3.5 Threats

- Eavesdropping (loss of confidentiality)
- Masquerading and resource theft
- Traffic redirection - eavesdropping (confidentiality), tampering (integrity)
- Denial of service (DDoS) - Stealth or general.

9.3.6 Security

- Service Set Identifier (SSID)
- MAC Address authentication
- Wired Equivalent Privacy (WEP)
  - One-way authentication
  - Static WEP keys
  - Key Size
  - Initialisation Vector (IV)

9.3.6.1 Wired Equivalent Privacy (WEP)

- Part of IEEE 802.11
- Short keys - 40 bit (64 bit RC4) 40 bit secret, 24 bit cleartext IV, 104 bit (128 bit RC4) 104 bit secret, 24 bit cleartext IV
- Static Keys + short IVs = repeated keystream
- Weak IVs

FMS WEP Attacks

- Was quickly incorporated into tools like AirSnort
- 5 - 10 million encrypted packets
- Can be achieved in < 1 second
- Vendors removed weakest IVs from new implementations.
- Various enhanced FMS attacks.

Statistical WEP Attacks

- Korek posted new WEP statistical cryptanalysis attack code to the NetStumbler forums 08/08/2004
- The attacks do not require millions of packets like FMS attacks,
- The number of weak IVs does not matter unlike the FMS attacks.
- Need hundreds of thousands of unique IVs to defend against it.
- Incorporated into tools like Aircrack and WepLab
- Shows that WEP is thoroughly broken.

PTW WEP Attacks

- Can be achieved in < 60 seconds including capture time
- Requires whole ARP reply packets

Primary Failings of WEP

- Client based only meaning the client must prove their identity to the AP but not vice versa. This makes it vulnerable to attacks by a rogue AP.
- Static WEP keys - There is no concept of dynamic or per-session WEP keys in IEEE 802.11b. The same key has to be manually entered at all the stations in the WLAN causing key management issues. This could result in keys not being changed often enough.
- Small key size - making bigger keys does not fig this issues as it only makes more work for the attacker rather than preventing them from gaining access.
- Weak Initialisation Vector (IV) - Known and weak as well as small IVs are used allowing a high chance of the same secret key and key stream from being used by two different frames.

9.4 CCMP and TKIP

Both are encryption protocols.

CCMP - The Advanced Encryption Standard Algorithm in counter mode with cipher block chaining with message authentication code protocol.

CCMP - [AES] in CTR mode with CBC with MAC (CCM) Protocol (CCMP)

TKIP - Temporal Key integrity protocol.

TKIP	CCMP
• Legacy hardware for WEP • Per frame keying	• New hardware, • Stronger than TKIP • AES in counter mode with 128 bit block and 128 bit key • Authentication and integrity vis cipher block chaining message authentication code.

Table 9: TKIP vs. CCMP

9.5 Physical Security

This deals with the restricting of physical access to the data or to systems that store or can access the data.

This will commonly deal with staff and personnel. How to achieve? (the following has been taken from PCI DSS V3.2.

- Use appropriate facility entry controls to limit and monitor physical access to the systems - This can include cameras as well as access control mechanisms. This data should be reviewed and compared with other entries. According to PCI DSS V3.2 this data should be stored for at least 3 months. Restriction of access refers to wireless access points as well.
- Develop procedures to identify and authorise visitors (with visitors being escorted). Visitors should be access to give up any authorisation before leaving. Use of a physical audit trail such as a visitors log is good.
- Control physical access for on site personnel to sensitive areas with authorised only access when necessary for job requirements. Access should be revoked as soon as not required anymore (termination, change of job etc)
- Physically secure all media. Backups to be stored in secure location (preferable off site with annual security reviews).
- Strict control to be placed on any internal and external distribution of data.
- Strict control is to be maintained over the storage and accessibility of media.
- Destruction of media once it is no longer required.
- Protect data collection points from tampering

9.6 Splunk

- Intrusion Detection System
- Intrusion Prevention System
  - Stopping people from getting what you want to keep private/secret

9.7 Microsoft

3 Levels of Separation

1. Physical
   - Physical Key to unlock
   - Automated and digital record of who accessed it
   - Once broken disks are destroyed and turned to concrete as a preventative measure
   - RAID is used to protect from disk failure
   - Data is fragmented across all disks, so you require all of the disks in order to get information
     - Encryption
     - Disks are locked to the rack hardware
     - No USB ports to allow people to plug into the racks
2. Customer data
   - Data is linked to the customer's tenant ID
   - Trust & Confidence
   - When an USB is given to a support officer
     - Is it encrypted?
     - What's the password? - Email me the password
   - Plug USB into laptop and get unencrypted data, however you encrypt the data yourself later - Remote wiping of devices
   - Policies from in tune that prevents sharing of private data
3. Back end data
   - The "back-end" data is the service provider data (configurations, logs, etc) all the data needed to run the service (networking, firewalls, hypervisors, etc), as well as all the data generated by running the service (logs, audit trails, alarms, reports, work tickets) - this service provider data was referred

9.8 Malware

- Malware – any sort of software designed with malicious intent.
- Virus – spread between files and to other machines around it.
- Worm – like virus, but network based.
- Spyware – steal information using keyloggers, controls microphones, etc.
- Trojan – says it's one thing but does something else.
- RAT – remote access trojan.
- Exploit Kits - a package of exploits of a system
- Ransomware – encrypt files and request payment for a key (wannacry)

10 Payment Card Industry (PCI) Security

Payment card data is a target.

Important details to remember,

- PAN - Primary account number = card number
- SAD - Sensitive Authentication Data
- CHD - Cardholder Data
- PED - PIN entry device
- CVV - CVV2, CVC, etc - Card verification value
- QSA - Qualified security Assessor
- ASV - Approved Scanning Vendor

10.1 Payment Card Data

Card Holder Data - Data Inclusions

- PAN (Primary account number)
- Card holder name
- Expiration Date
- Service Code

Sensitive Authentication Data (SAD) - Data inclusions

- Full magnetic stripe data or equivalent on chip
- CAV2/CVC2/CVV2/CID
- PINs/PIN blocks

Track Data - Data inclusions

- Payment cards typically use two tracks of data on the magnetic stripe (or EMV chip).
  1. Track One - Contains all the fields of both track one and two with a length of up to 79 characters.
  2. Track Two - Provides shorter processing time for older dial up transmissions with a length of up to 40 characters.

10.2 EMV Chip Cards

Track data is found on the chip differs from the track data found on the magnetic stripe. The chip data includes a Unique chip CVV/CVC code.

The purpose of this is to prevent cloning of cards magnetic stripes being possible from the chip data. This does not prevent *card-not-present* fraud from occurring in cases of e-commerce, mail order/phone order etc.

10.3 (PCI DSSs)

- Build and maintain a secure network
  1. Install and maintain a firewall configuration to protect cardholder data
  2. Do not user vendor supplied defaults for system passwords and other security parameters.
- Protect card holder data
  3. protect stored cardholder data
  4. encrypt transmission of cardholder data across open, public networks.
- Maintain a vulnerability management program
  5. Use and regularly update anti-virus software and programs
  6. Develop and maintain secure systems and applications
- Implement Strong access control measures
  7. Restrict access to cardholder data by business. Need to know.
  8. Assign a unique ID to each person with computer access
  9. Restrict physical access to cardholder data.
- Regularly monitor and test networks
  10. Track and minitor all access to network resources and cardholder data
  11. Regularly test security systems and processes.
- Maintain an information security policy
  12. Maintain a policy that addresses information security for employees and contractors.

10.3.1 Firewall

1. Documented configuration and formal processes for firewalls and routers
2. All external traffic and wireless traffic to pass through firewall,
3. Implement a DMZ[14] between internet and cardholder data environment.
4. Personal firewalls on remote access PCs.

10.3.1.1 Standards

Configuration standards and procedures will help to ensure that the organisations first line of defence in the protection of its data remains strong.

1. Establish and implement firewall and router configuration standards that include the following:
   a) *Approval* - A formal process for approving and testing all network connections and changes to the firewall and router configurations. Without formal approval and testing of changes, records of the changes might not be updated which could lead to inconsistencies between network documentation and the actual configuration.
   b) *Diagrams* - Current network diagram that identifies all connections between the cardholder data environment and other networks including any wireless networks. Without current network diagrams, devices could be overlooked and be unknowingly left out of the security controls implemented for PCI[15] DSS[16] and thus be vulnerable.
   c) *Data Flows* - Current diagram that shows all cardholder data flows across the systems and networks. Network and cardholder data-flow diagrams help an organisation to understand and keep track of the scope of their environment by showing how cardholder data flows.
   d) *Firewall DMZs* - Requirements for a firewall at each Internet connection and between any DMZ and the internal network zone. Using a firewall on every internet connection and between any DMZ and internal network allows the organisation to monitor and control access and minimises the chances of a malicious actor accessing the internal network via an unprotected connection.
   e) *Roles/Responsibilities* - Description of groups, roles, and responsibilities for management of network components. This ensures that personnel are aware of who is responsible for the security of all network components and that those assigned to manage components are aware of their responsibilities. If this is not done, devices could be left unmanaged.
   f) *Justify Ports* - Documentation of business justification and approval for use of all services and protocols, and ports allowed. This includes documentation of security features implemented for those protocols considered to be insecure. Compromises often happen due to unused or insecure services and ports. These often have known vulnerabilities and many people don't patch services they don't use.
      - Clearly documentation of these services, ports and protocols ensures that those not in use can easily be detected and disabled.
      - Approvals should be granted by others not responsible for managing the configuration.
   g) *Rule Reviews* - Require to review firewall and router rule set at least every six months. This review allows an organisation the opportunity to clean up any unneeded, outdated, or incorrect rules and ensure that all rule sets allow only authorised services and ports that match the documented business justifications.
2. Restrict Connections - Build a firewall and router configuration that restricts connections between untrusted networks and any

[14] Demilitarised zone: a physical or logical subnetwork that contains and exposes an organisations external facing services to an untrusted network

[15] a standard for connecting computers and their peripherals.

[16] decision support system

system components in the cardholder data environment.
   a) *Perimeter Firewalls* - Install perimeter firewalls between all wireless networks and the cardholder data environment. Configure these firewalls to deny all but authorised traffic between the wireless environment and the cardholder data environment.
3. Implement DMZs - Prohibit direct public access between the Internet and any system component in the cardholder environment.
4. Firewall on all Mobile Devices - Install personal firewall software or equivalent functionality on any portable computing device (including company and/or personal) that connect to the Internet when outside of the network.
5. Policies and procedures - Ensure that security policies and operational procedures for managing firewalls are documented, in use and known to all affected parties.

10.4 Operations

Term	Description
Cardholder	The owner of the card, the person making the purchase.
Merchant	The place where the purchase is being made, this is often the store.
Acquirer	Band of financial institution that processes credit or debit cards on behalf of the merchant.
Payment Brand Network	The payment network the card is part of. i.e. Visa, Mastercard, EFTPOS.
Issuer	For credit cards it is the provider of the credit. Usually a bank. For debit cards this will be a bank.

Table 10: List of Terms

10.4.1 Authorisation

1. Cardholder swipes card at merchant
2. Acquirer asks payment brand network to determine issuer
3. Payment brand network determines issuer and requires approval for purchase.
4. Issuer approves purchase.
5. Payment brand network sends approval to acquirer
6. Acquirer sends approval to merchant
7. Cardholder completes purchase and receives receipt.

10.4.2 Clearing

1. Acquirer sends purchase information to the payment brand network.
2. Payment brand network sends purchase information to issuer, which prepares data for card holders statement.
3. Payment brand network provides complete reconciliation[17] to acquirer.

10.4.3 Settlement

1. Issuer determines acquirer via the payment brand network
2. Issuer sends payment to acquirer
3. Acquirer pays merchant for card holder's purchase,
4. Issuer bills cardholder.

[17] Validates that the records are correct by comparing them against each other (usually at least 2 sets)

11 Industrial Control Systems (ICS) Security

11.1 Industrial Networking

11.1.1 Internet of Things (IoT)

11.1.1.1 Telstra Case Study

Need to consider security in,

- the dirt → in really big machines,
- the factory → MODBUS over RS482 and over Ethernet,
- transport → rail and traffic management systems,
- buildings
- security systems themselves

How to achieve security?

Understand the standards

- IEC 62443/ISA-99
- NIST SP800-53 & 82
- SANS CSC

What to do?

- *Security Architecture and Design*
  - IT-OT Interfaces, Business Zone, DMZ, Operations, PCN,
  - Enforcement zones (Purdue model)
- *Network Design and Installation*
  - Network teams,
  - Industrial grade equipment
- *Security Assessments*
  - Gap Analysis
  - Compliance Audits
- *Vulnerability and Penetration Testing*
  - Usually only in a test environment
  - Use the Red Team for detailed VPT and "Specials"
- *Security Remediation*
  - Strategy and planning
  - Review and Reporting

Things to know!

Term	Description
ICS	Industrial Control System
CI	"Critical Infrastructure" - Small subset of ICS
SCADA	Supervisory Control and Data Acquisition - Remote Monitoring and control, it is a big subset of ICS
DCSs DCS	Distributed Control system - Process control and another big subset of ICS. It is not SCADA.
PCSs PCS	Process Control System (typically DCS)
PCNs PCN	Process Control Network (has PLCs)
PLC	Programmable Logic Controller
PLC LAN	A PCN

Table 11: List of Useful Terms

11.1.1.2 Things to Know

Many domestic devices such as Smart meters, smart homes, smart appliances, devices, vehicles, buildings etc.

Security issues with IoT devices

- The limited processing power and size of the devices and limit the use of encryption
- Patching flaws in low cost devices not a high priority.

11.1.2 Industrial Networking Security

- Physical Layer availability/resiliency requirements
- Segment the physical and logical topology.
- Firewalls with strong ACLs
- Defence in depth
- Use managed industrial switches,
- Use intrusion detection services - IDS not IPS.

11.2 Critical Infrastructure

Critical infrastructure are assets that are essential for the functioning of society.

Includes,

- Airports, bridges, dams
- Electricity, fuels, water,
- Hospitals, lighthouses,
- Railways, roads, transport, post,
- Sewage, waste management,
- Telecommunications,
- National broadband network (NBN) ?

11.2.1 Advanced Persistent Threat

Usually a group or a a state with both the capability and the intent to persistently and effectively target a specific asset. These are usually long term sophisticated attacks.

11.2.1.1 Advanced

Considerable intelligence gathering abilities with access to specialised/protected knowledge. They combine multiple tools and techniques to get to and acquire the target. They are not limited to technological resources.

11.2.1.2 Persistent

This does not mean the attacks are constant. They will have a specific target and will typically not be able simple criminal gain.

The motivation is usually state political.

They will have considerable resources and will continuously monitoring with the attacks being planned.

Can involve the re-acquiring of lost targets and the maintaining of long term access to targets.

11.2.1.3 Threat

- Both capability and intent,
- Motivated and well resourced.
- Effective actions,
- Specific target,
- coordinated and managed (directed)

11.2.2 Critical Infrastructure Resilience

11.2.2.1 Australian Government Critical Infrastructure Resilience Strategy

- Effective business-government partnership
- Organisation resilience BoK
- Assist owners and operators
- Timely and high quality policy advice
- Implement the Australian Government's Cyber Security Strategy and
- Support programs by States and Territories.

11.2.3 Stuxnet

- Discovered June 2010
- Targets Siemens "Step-7" SCADA software on Microsoft Windows,
- First to include a PLC root kit
- Different variants targeted five Iranian plants
- Speculation that Israel & USA may be involved.
- Used four zero day attacks.
- Initially via infected USB drives
- Then P2P RPC to spread.
- Root kit capability with devices drivers signed with private keys of two stolen certs (Realtek and JMicron - Both located in Taiwan)
- It infects project files belonging to the Siemens WinCC/PCS 7 SCADA control
- Intercepts communications between WinCC and the Siements PLC devices.
- Covertly installs itself on PLC devices,
- hides itself from WinCC
- Used a zero-day (hard coded password) exploit in the WinCC database software.
- Specific slave variable frequency drives,
- Siemens S7-300 system
- Only attacks systems with drives from Vacon in Finland and Fararo Paya in Iran
- Only attacks between 807 Hz and 1210 Hz.
- Randomly 1410 Hz then 2Hz then 1064Hz
- Rootkit masks the rotational speed.

According to Eric Bryes (2011) you can stop Stuxnet by changing all of the default passwords when installing the Siemens Control systems however, you will also prevent the system from working correctly as the devices have hard coded passwords in its PCS7 applications.

12 Cloud Computing

> *"Clouding computing is a paradigm for allowing network access to a scalable and elastic pool of shareable physical or virtual resources with on-demand self-service provisioning and administration."*
> - ISO/IEC 17788:2014
> Information technology - Cloud computing - Overview and vocabulary

There are six key characteristics that depict a *cloud*

1. **broad network access** - physical and virtual resources are available over a network and accessed through standard mechanisms that promote use by heterogeneous[18] client platforms. It provided a increased level of convenience in that users can access resources from wherever they need to work.
2. **On-demand self service** - A cloud service customer can provision computing capabilities as needed, automatically or with minimal interaction with the cloud service provider. Offers a relative reduction in costs, time, and effort needed to take an action since it grants the user the ability to do what they need when they need.
3. **multi-tenancy** - Allows mutliple tenants through the method of resource
   allocation that allows isolation and inaccessibility by other tenants. A tenant is a group of cloud service users sharing access to a set of resources.
4. **resource pooling** - cloud service resources can be aggregated in order to serve one or more cloud service customers. Key element in multi-tenancy. It provided abstraction to hide the complexity from the customer. A customer generally has no control on how the resources are provisioned/located etc.
5. **Rapid elasticity and scalability** - Rapid and elastic provisioning of resources based on need (automatically) or customer demand (manually). Resources will often appear to be unlimited and as such customers do not need to worry about availability.
6. **measured service** - Usage can be monitored, controlled, reported and billed. Used to optimise and validate the cloud service. This allows customers to only pay for the service they use.

It should be noted that a managed service is not necessarily a cloud service.

12.1 Roles and Activities

Cloud Service Provider -
Party which makes the cloud service available.

Cloud Service Customer -
The party in which is in a business relationship for the purpose of using the cloud service.

Cloud Service User -
Natural person, or entity acting on their behalf that is associated with a cloud service customer and uses that cloud service. Can include devices and applications.

12.2 Deployment Models

1. Public cloud,
2. Private cloud,
3. Community cloud,
4. Hybrid cloud,

12.3 Service Models (NIST)

These are also known as *"capability types"*.

- Infrastructure as a service (IaaS)/Infrastructure capabilities type.
- Platform as a service (PaaS)/Platform capabilities type
- Software as a service (SaaS)/Application capabilities type

[18] diverse in character or content.

12.3.1 IaaSs

Provides,

- Infrastructure as a service,
- Network as a service,
- Data storage as a service,
- Compute as a service

12.3.2 PaaSs

Provides,

- Platform as a service
- Network as a service,
- Data storage as a service,
- Communication as a service

12.3.3 SaaSs

Provides,

- Software as a service,
- Network as a service,
- Data storage as a service,
- Communication as a service

12.4 Security

12.4.1 Cloud Security Alliance (CSA)

The lower down the stack the cloud provider stops the more more security they are responsible for implementing and managing.

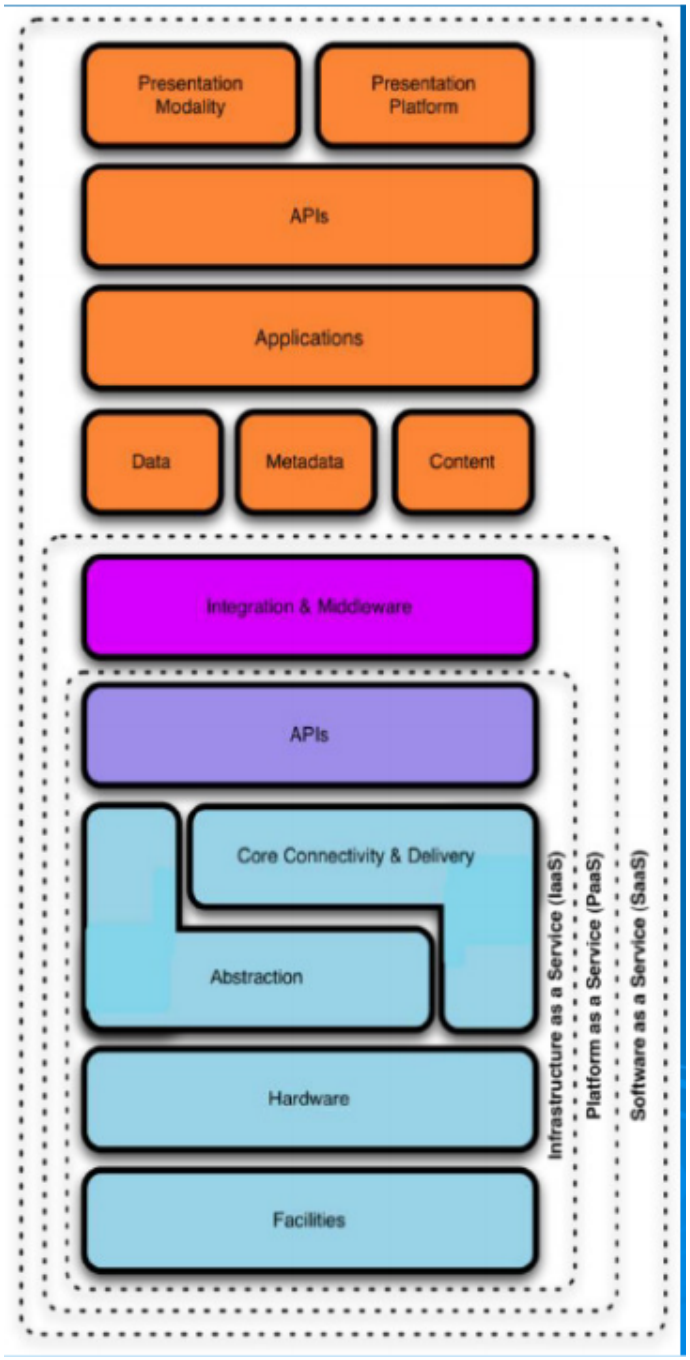


Figure 29: CSA Stack

SaaS deals with the entire stack, PaaS deals with everything below Integration and Middleware and IaaS deals with everything below and including APIs.

To secure PaaS and SaaS we need to have a secure IaaS. There is is the physical security, facilities and hardware. There is also HR security where personal are secure.

12.5 Edge Computing

When performing data processing at the edge of the network, we need,

- Distributed and decentralised architecture,
- Big data on small devices,
- Performing analytic and knowledge generation at or near the source of the data.
- Reduce the bandwidth needed to transfer data between devices and the data centre.