Asset

Anything within the organization that is worth protecting:

Information, Systems, Devices, Facilities, Personnel, Intellectual Property

Confidentiality:

Military / Government confidentiality:

- Top Secret: drastic effects / grave damage to national security
- Secret: significant effects / critical damage to national security
- Confidential: noticeable effects / serious damage to national security
- Sensitive but unclassified: internal use
- Unclassified: public data

Commercial / private confidentiality

- Coonfidential / private: drastic effects on the competitiveness
- Sensitive
- Public

In general classified is used as a term to describe anything but public data.

Deleting Data

There are multiple ways to delete data with massive difference in effectiveness:

- Erasing: removes only the link to the storage point, actual data remains
  Simple tools can recover this data!
- Clearing: Overwrites the data to delete it, usually with a single character
  Can not be recovered using simple tools
- Purging: Clearing, but overwrites multiple times to make recovery harder
  This method might also include others such as degaussing. It is Irrecoverable
- Degaussing: Strong magnetic field that can wipe data from an HDD
  (SSD, CD, etc not affected!)
- Destruction: Destroy the physical hardware of said data. Irrecoverable.

Tracing and Hiding sensitive Data

- Steganography: Embedding a Message within a file
- Watermarking: unique identifier, that usually can't copied, or mutated.
  Often used in Documents, movies, etc to stop counterfeits.

STRIDE Model

- Spoofing: Using a false identity to gain access to a system.
- Tampering: unauthorized changes/manipulation of data
- Repudation: The ability to deny having performed an attack,
  to others being blamed.
- Information Disclosure: unauthorized revelation of classified
  private information.
- Denial of Service: Prevent or restrict access to a service by flooding it.
- Elevation of Privilege: Gaining unauthorized privileges on a system
  For example: becoming root as regular user

Copyright

Protects these works: Literay, musical, dramatic, choreographic, graphical,sculptural,audiovisual, recordings, architectural works

Computer software is in the literary works category

Note only the source code is protected, not the idea itself.

Implicit Copyright is automatically granted if you are the creator of said work.

The explicit copyright can be obtained from the government and lets you use the symbol: ©

This copyright lasts for 70 years after the death of the last copyright holder.

Trademark

Protects: Slogans, Logos, words used to identify something

Implicit Trademark is automatically granted if you use the ™ symbol

Explicit Trademark has to be obtained from the government and lets you use the ® symbol.

Patents

Patents protect intellectual Property for 20 years.

For a patent a product must be useful,new,not be obvious

After the 20 years the patents enters into the public domain.

Patents are the worst thing ever.

Trade Secrets

Copyright, Patents and Trademarks require you to disclose what is protected

Because of this, many companies simply hide this information from the public

This means, while you are not protected by the law, as long as this information stays within your company, it will be protected forever.

Likely the "best way" to protect computer software, aka proprietary trash

CIA triad

The primary goal of Security infrastructure:

- Confidentiality: prevention of unauthorized access to data
  no matter if the data is in transit, in storage or in process
  Usually done with encryption and access control
- Integrity: prevention of unauthorized alteration of data
  Data integrity: data is complete, consistent and accurate
  System integrity: System only does what it was intended to do
  Usually done with hash verification, intrusion detection
- Availability: Systems should be accessible at all times
  Usuall done with Redundancy, backups

Nonrepudation & Accountability

Nonrepudation ensures that every action can be traced to the actual source

This prevents attackers from covering their actions.

Accountability ensures that every being is responsible for their actions.

E.G. Nonrepudation ensures Accountability.

Usually done with certificates, session identifiers, logs, etc

Data Classification

- Data in Use, Transit, Rest
  used by application, tranit via x, rest = storage, HDD, SSD
- Personally Identifiable Information PII
  information to identify a person, name, social security number etc
- Protected Health Information PHI
  Health Information recorded in any form.
- Proprietary(Trash) Data
  microtroll windoof, appul

Threats

- Threat: A potential danger to an asset
- Threat intelligence: Knowledge about emerging or existing threats
- Thrat event: Accidental or malicious exploitation of a vulnerability

Vulnerability

- Common Vulnerabilities and Exposures (CVE)
  Industry wide standard identification number for Vulnerabilities
- Common Vulnerability Scoring System (CVSS)
  Uses the CIA triad to score vulnerabilities on severity

Risk

Assessment of the possibility that a threat will exploit a vulnerability

Realized Risk

A threat actor has now taken advantage of a vulnerability

The whole point of security is to stop exactly this.

Risk Management

- Identifying vulnerabilities
- evaluating importance of data and countermeasure cost
- Implementing cost effective countermeasures

Risk management is the balance of threat/risk and usability

In other words, only implement measures that are necessary

Some data is not worth protecting, and some systems can be replaced easily.

Others are sensitive, or fundamental, these must be preserved at all costs.

Risk Analysis

- Evaluation, assessment, assigment of value to assets
- Examining environment for risks
- Evaluating the likelihood of a threat event occurring
- Assessing the cost of countermeasures
- present cost/benefit report to upper management

Asset Valuation

The representation of an asset in currency

This can include things such as repair costs, maintenance costs,etc

Exposure

Possibility of an asset loss due to a threat

exposure factor (EF) checks how serious that loss would be

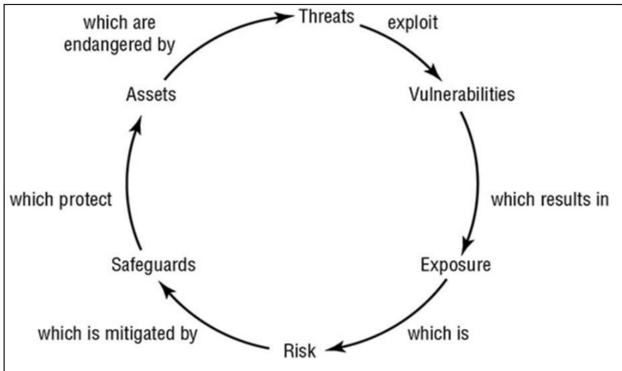
Attack

The intentional try to exploit a vulnerability

Breach

The circumvention of security measures by a threat actor.

A breach combined with an attack, can result in penetration



Quantitative and Qualitative Risk Analysis

– Quantitative Risk Analysis:

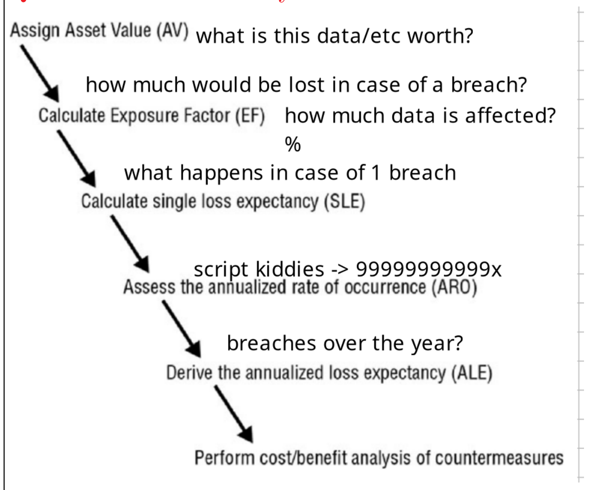
assignment of real dollar figures to loss of assets

– Qualitative Risk Analysis

The subjective / intangible worth value to the loss of assets

Both methodologies are necessary for complete risk analisys!

Quantitative Risk Analysis


Assign Asset Value (AV)
what is this data/etc worth?
how much would be lost in case of a breach?
Calculate Exposure Factor (EF)
how much data is affected?
%
what happens in case of 1 breach
Calculate single loss expectancy (SLE)
script kiddies -> 99999999999x
Assess the annualized rate of occurrence (ARO)
breaches over the year?
Derive the annualized loss expectancy (ALE)
Perform cost/benefit analysis of countermeasures

- Exposure Factor (EF): percentage of loss by realized risk
- Single Loss Expectancy (SLE): Cost of single realized risk

  SLE = EF * Asset Value (AV)
- Annualized Rate of Occurrence (ARO):

  expected frequency of a risk within a year
- Annualized Loss Expectancy (ALE):

  Expected yearly cost of Losses due to realized risks

  ALE = SLE * ARO

If you implemented a safeguard, you have to recalculate the ARO.
The entire idea of of security is to reduce the ARO!!
The EF usually remains the same

– Safeguard Costs

First, compile a list of safeguards against each threat.
Assign each safeguard a deployment value -> Annual Cost of Safeguard (ACS)

– Safeguard Cost/Benefit

ALE without safeguard - ALE with safeguard - ACS = Value of safeguard
if the value of safeguard is below 0, then it is financially irresponsible
Note that this only takes in the financial damage since this is Quantitative!

Dealing with Risk

- Risk Mitigation Implementation of safeguards in order to eliminate vulnerabilities or block threats
- Risk Assignment / Risk Transferring

  purchasing insurance or outsourcing. Transfer the cost of risk to other entity
- Risk Acceptance Doing nothing as cost/benefit would be low
- Risk Deterrence auditing, cameras, security guards, warnings, etc
- Risk Avoidance Not using the system associated with the risk
- Risk Rejection Simply ignore risk without cost/benefit analysis!
- Residual Risk A countermeasure might not fully eliminate a risk

  This is the remaining risk that we have decided to accept.

Privacy

- GDPR
  - Companies have inform authorities in case of serious data breaches
  - Individuals have the right to demand their data from companies
  - Individuals have the right to be forgotten (deletion of data)
  - EU tries to enforce this globally
  - enforces pseudonomization
- Patriot act
  - blanked authorization of surveillance of an individual with 1 warrant
  - ISP have to provide data
  - easier wiretapping
- pseudonomization replacement of data with aliases

  This makes it harder to identify a person from said data
- anonymization Complete obfuscation of an identity

Access Control

- Subject The Requesting party

  Note, this can be a person, a program, a process, etc.
- Object The requested "object"

  Databases, programs, files, etc
- Access Control The management of the relationship

  Between Subject and Object
- Preventive Access Control

  Stops unwanted access to Objects
- Detective Access Control

  Detects unwanted access to Objects after it has occurred
- Corrective Access Control

  Restores the last "correct" state after unwanted access
- Deterrent Access Control

  discourages unwanted access to Objects
- Directive Access Control

  Directive issued by company. -> Don't click on links.
- Compensating Access Control

  Controls used in addition/ as a replacement. Can be any control measure
- Recovery Access Control

  More advanced version of corrective control
- Physical Control

  Control of items that one can physically touch
- Technical / logical Control

  Hardware or Software mechanism for access control
- Administrative Control

  Policies and Procedures created by a Company

Steps of Access Control

1. Identification

   Providing an identity to enforce Nonrepudation.
   is usually public information.
   username, tokens, fingerprints, facial recognition
2. Authentication

   Often used in combination with Identification.
   Verifies the identity. Usually a password, or token.
3. Authorization

   Controls what a user is allowed to do and what they aren't
   There are different ways of controlling this.(Check next page)
4. Auditing

   Record all actions by all Subjects in order to hold them accountable
5. Accounting (Accountability)

   This would mean taking actions against a person,
   that has acted in bad faith/maliciously
   Keep in mind that you need a strong system of identification,
   and auditing
   in order to actually win in a court of law against the bad actor!

Authentication Factors

- Type 1: Something you know
- Type 2: Something you have, ex. a Device, smartcard, etc
- Type 3: Something you are/do, ex. Fingerprint, face, etc.

Type 1 is the weakest form of authentication and type 3 the strongest!

– Location

This is a form that is used in combination with others.
Ex: A certain IP might be a requirement to log into one of your systems
Or your bank might block a transaction if it isn't from your country of residence.

Multifactor Authentication

Multifactor Authentication simply uses more than one type to authenticate
This might be a password(Type1) and a token(Type2)
The strongest form of authentication if therefore all 3 types together!

Passwords

Passwords are never stored in plain text,
they might be stolen easily with 1 breach
Instead, they are saved with a hash-algorithm,
makes them nearly useless to threat actors
Unless they also have the access to said algorithm.

Types of passwords:

– Plain Password

use numbers, characters and special symbols with a length of at least 10.
10 characters -> 928 years of cracking!!
Don't use personal information such as names etc.

– PassPhrases

Passphrases are chained words that are easy to remember
It is important to have a longer passphrase than a regular password!
Still use special symbols. Also try to include random upper-lower case spelling
Or leadspeak s1nc3 that 1s qu1te 3ff3ct1v3!!

– Cognitive Passwords

A series of personal questions. -> what is the name of your first pet?
Best way to do this is letting users create both the question and the answer!

– SmartCards

Card used for identification / authentication. Often integrates key encryption
Usually temper resistant
One downside, loss of card might give a threat a window of exploitation!

– Tokens

Generated by a special Device.
Regular password generators: any device can take that place. ex. smartphone

– Synchronous Dynamic Token
Time-based One-Time Password (TOTP)

Device generates Token every x seconds.

– Asynchronous Dynamic Token
HMAC-based One-Time Password (HOTP)

Device generates one time token based on algorithm. Stays until used!

Access Control Models (Authorization)

– Discretionary Access Control (DAC)

Every object has an owner, and that owner can to grant or deny permissions
NTFS from windoof uses this

– Role Based Access Control

Permissions are assigned to roles not users. Users are assigned to roles.
Users who are in a role with said privileges can use them.

– Rule Based Access Control

Global rules that are applied to all subjects
Good example is a firewall, which applies said rules equally to all subjects

– Attribute Based Access Control

Similar to Rule based Control but with additional attributes
This could give one subject more rights than another

– Mandatory Access Control

Use of labels applied to both subject and Object
If user has the same label as a file, then user has access to it.

Authorization Mechanism

– Implicit Deny

Deny everything that hasn't been specifically allowed
Most used!

– Constrained Interference

Applications might hide functionality based on the privileges of a user

– Access Control Matrix

This writes Objects,Subjects and privileges into a table
If Subject tries to access an object the table for said object is checked

– Capability Table

This is the same as the Access Control Matrix but with a subject focus
In this table the subject and all accessible Objects are written down

– Content-Dependent Control

Constrict Access to the data within an Object
In a database a user might be able to check table 1 but not table 2.
While the object is the entire database!

– Context-Dependent Control

Give a subject access depending on what the subject does
Ex. The checkout button in an online shop only works, if you have something in the shopping cart.

– Need to Know

Subjects should only have access to Objects they need to do their job.

– Least Privilege

Subjects should only have the privileges they need to do their job.

– Separation of Duties and Responsibilities

No single person should have total control over the entire System!

Common Access Control Attacks

– Access Aggregation Attacks (Passive Attacks)

This is the collection of nonsensitive data, that combined could give
a threat actor the opportunity to launch a proper attack.
Ex. IP address, open ports, Operating System -> specific exploit

– Password Attacks (Brute Force)

Spam random sequences until you get the right one

– Dictionary Attacks (Brute Force)

Try passwords from a list of passwords, example leaked password list.
Can also be done with list of common passwords, or slightly changed
previous passwords (One-Upped-Passwords -> 1 character changed)

– Birthday Attack (Brute Force)

Try to get the same hash as the password with a different sequence
Can be mitigated by using better hashing algorithms. SHA-3 instead of md5
Note the attacker needs access to the hash in order for this to work!

– Rainbow Table Attacks

Combines the Birthday attack with a table of precomputed hashes.
This is then used to compare to a password hash list.

– Sniffer Attacks

Threat actor analyzes data sent over network with a sniffer tool.
A good example for this is wireshark
Can be mitigated by using encryption and One-Time passwords
encryption makes the data useless and One-Time passwords are as well

– Spoofing Attacks

Pretending to be something/someone else. Ex. pretending to be router.

– Social Engineering Attacks

Gaining and then misusing trust of someone.
Indian accent you get refund if you buy me 2 cards from target

– Shoulder Surfing (Social Engineering)

Reading information on a screen from a persons back.

– Phishing (Social Engineering)

Trick a Person to click on a fake link to log in, giving the attacker
all the credentials to log-in

– Spear Phishing (Social Engineering)

Targeted Phishing at a group. Ex. Employees at company x.

– Whaling (Social Engineering)

"Phishing für grosse Fisch" -> CEOs etc

– Vishing (Social Engineering)

Phishing via VOIP or instant messaging

Protection Mechanism

– Layering (defense in depth)

Multiple Controls in Layers, if one fails, there are still the other ones

– Abstraction

Combining Objects into groups in order to simplify permission management

- Data Hiding
  Storing Objects in compartments that can't be seen / accessed by an unauthorized subject
- Security through Obscurity
  Not informing a subject about an object, and hoping it will stay hidden
- Encryption
  Turning data into gibberish via algorithms.

Red-Team

Offensive Cyber-Security: simulate attacks

- Think outside the box
  Find new ways and tools and attack systems to show the flaws
- Deep Knowledge of Systems
  Deep Knowledge about systems, flaws, exploits, methodologies,etc always up-to-date with technology
- Software Development
  Learn how to develop your own tools.
- Penetration testing
  Identify vulnerabilities and potential threats
- Social Engineering

Blue-Team

Defensive Cyber-Security: prevent attacks

- Organized and detail-oriented
  Prevent gaps by thinking about EVERYTHING
- Cybersecurity Analysis and threat profile
  Assess the security of an organization. Create Risk/Threat Profiles.
- Hardening Techniques
  Reduce the attack surface hackers might exploit
- Knowledge about detection Software
  Be familiar with software that recognizes unauthorized actions low skill application would be rkhunter.
- Security Information, Event Management (SIEM)
  Software that allows real-time analysis of security events

Linux

Adding user: usermod -m username -s /path/to/shell

Change shell: chsh -s /path/to/shell

Change password: passwd username

Add user to group: usermod -a -G groupname username

Change file permission: chmod permission file

Change file owner: chown file user/group

Check IP address: ip addr / ip -c -brie a

DNS query: dig domain (dig shitgaem.online)

- File System Permissions

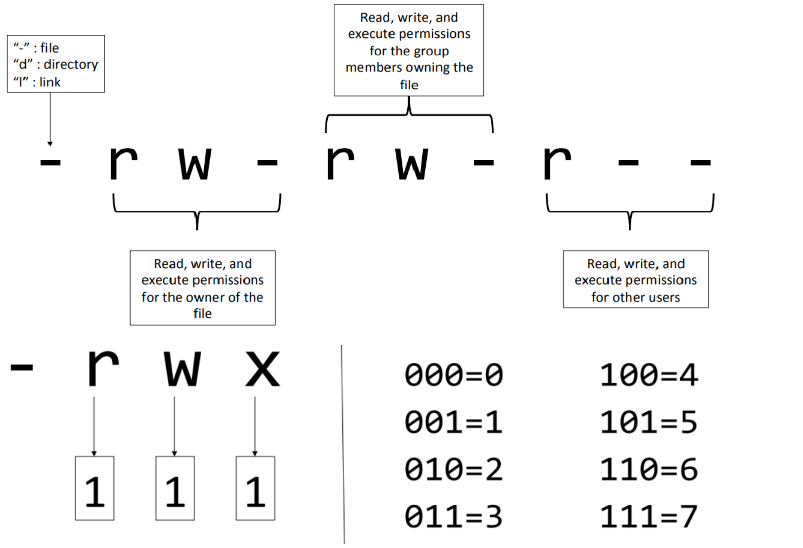
r = read, w = write , x = execute

Every single file has these attributes.

These attributes are also duplicated for 3 different types of users.

1. owner, 2. group of owner, 3. other

This means the actual permission would look like this:


"-" : file
"d" : directory
"l" : link
Read, write, and execute permissions for the group members owning the file
- r w - r w - r - -
Read, write, and execute permissions for the owner of the file
Read, write, and execute permissions for other users
- r w x
1 1 1
000=0
001=1
010=2
011=3
100=4
101=5
110=6
111=7

Manually set IP address (abando): edit /etc/network/interfaces

Configure SSH: edit /etc/ssh/sshd_config

For birbs sake, use a nonstandard port for ssh :)

Check current shells: ps

Check current processes: htop

grep read lines

grep 'Warning' /var/log/rkhunter.log

Read Lines: awk 'sshd.*invalid user/ {print $11}' auth.log

bit-by-bit copy: dd if=<media/partition> of<image_files>

mount - o ro,noexec,loop evidence_01 /mnt/investigation

mount with read only and no execution

Check recently changed files ls -lasrt

PID

The unique identifier the kernel gives each process

This shows both background and foreground applications

logs

Logs capture every single action on linux, this can be used to detect bad actors. However, logs can also be spoofed, which means you always have to be sure, that everything is being logged, and that no logs have been tampered with.

notable logging systems/files: syslog, rsyslog,var/log ,auth.log

The shred command with -f and -n force deletes log files.

Mainly used like this: shred -f -n 15 /var/log/auth.log*

This shreds 15 lines from every log file with the name auth.log(something)

other things to look out for:

- set-UID Rogue Files
- Directories with .something Hidden...
- Regular files in the /dev directory
- Recently modified files ls -lasrt

Schedules Tasks

Schedules tasks can be written either in cron.d or with systemd

IP-Tables

name one reason not to use ufw...

Flush all rules: iptables -F

Block Input: iptables -P INPUT DROP

Block Output: iptables -P OUTPUT DROP

Block Forward: iptables -P FORWARD DROP

Allow Port: iptables -A INPUT/OUTPUT -p port ...other shit...

Show rules: iptables -L -n -v – line-numbers

Sticky Bit

This is a single bit in front of rwx. -> 1777 sticky set, 0777 sticky not set

The interpretation of this bit depends on the file type

For directories, it means that any files within that folder

May only be renamed or deleted by the owner.

For files this bit is deprecated!

Security Enhanced Linux (SELinux)

This is a module created by the NSA that implements types, which mark files based on the type of a subject.

Ex. a top-secret process can create a file with chmod 777, but a confidential process still can't open it.

This is called MLS in SELinux and is related to Multi Category Security (MCS)

Snort

Detection software like rkhunter

NetCat

This can be used for anything dealing with TCP and UDP.

You can also use it to control compromised systems...

Reverse Shell

The idea is, since the starting connection comes from the victim,

Not only do we not have NAT and firewall problems, the connection also looks more legit than when we connect.

This gives a hacker some sort of legitimacy on that system.

Scapy

Tool used to send, sniff, dissect and forge IP packets.

You can probe, scan and attack networks

You can attack signature for IDS/IPS systems

Encryption Terms

– Plain Text: unencrypted message

– Ciphertext: encrypted message

– Cipher: Algorithm used to encrypt

– Cryptographic key: Just a number to decrypt a message

The range is defined by the algorithm. 0 to 2^n

A key with 128 bits would have a range of: 0 to 2^{128}

It is critical to keep the keys secret!

– One-Way Function:

mathematical function that produces output in a way that input can't be retrieved.

There is no TRUE One-Way-Function

Cryptography works on the believe that it can't be broken RIGHT NOW

However, this does not mean it will stay so forever, see already broken ciphers

– Reversability: The option of encryption....

– Nonce: A Public,unique One-Time-Use Number

Makes sure a key is not re-used twice!

– Initialization Vector (IV): A random bit string

Same length as the block size and is 'XORed with the message'

IVs are used to create a unique ciphertext with the same key

– Confusion:

This is the case when encryption is so complicated,

that merely reforming the string doesn't reveal the message

Aka bruteforce doesn't work anymore.

– Diffusion:

A change in the plaintext will result in multiple changes in the ciphertext.

– The Kerckhoff's Principle

This means everything about the system is public but the key

It therefore requires the system to be secure even under these circumstances

The idea is that public algorithms may hasten the improvements on them

– Permutation Swapping Bytes around

– Byte Substitution Replacing bytes with others

– SP-Networks algorithm that uses repeated Permutations and Substitutions

Permutations and Substitutions are combined to a round

Rounds are then repeated many times

Caesar Cipher or ROT3

One of the earliest encryption systems

Simply shifts a chracter by 3 A to D, B to E...

One-Time-Pad

Create a key with the same length as the message

XOR each message bit with each key bit

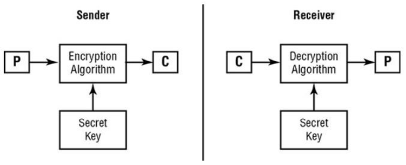
This Cipher is UNBREAKABLE!

However it is not practical.. 1GB file 1GB key...

No proper way to transmit, store a key

Using a key twice == Cipher broken

Symmetric Cryptography



– Same key for encrypting and decrypting

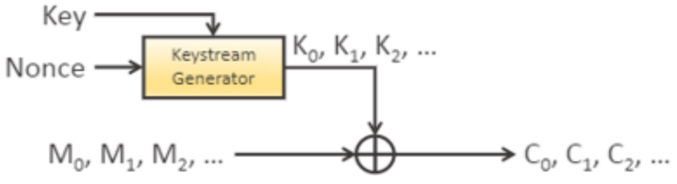
– Shared key for all parties involved!

If one leaks the key, the cipher is broken!

– Doesn't confirm identity!

Anyone who has the key can pretend do be another

Stream Ciphers



+ Encryption of long continuous streams, of possible unknown length

+ Extremely fast with low memory footprint, ideal for low power devices

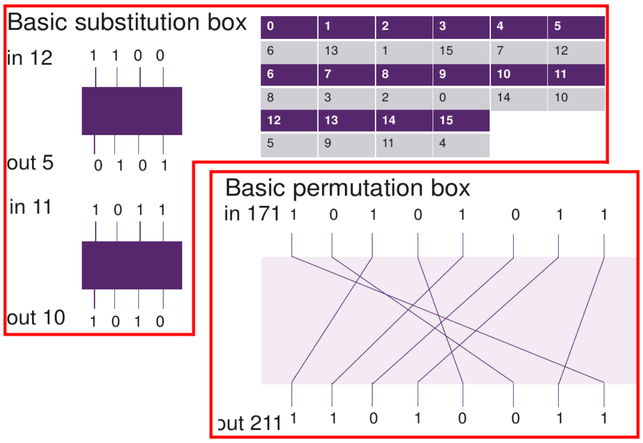
+ If designed well, it can seek to any location in the stream

– The keystream must appear statistically random

– You must never reuse a key + nonce

– Stream ciphers do not protect the ciphertext (no guaranteed integrity)

Substitution / Permutation box



0	1	2	3	4	5
6	13	1	15	7	12
6	7	8	9	10	11
8	3	2	0	14	10
12	13	14	15		
5	9	11	4		

Block Cipher

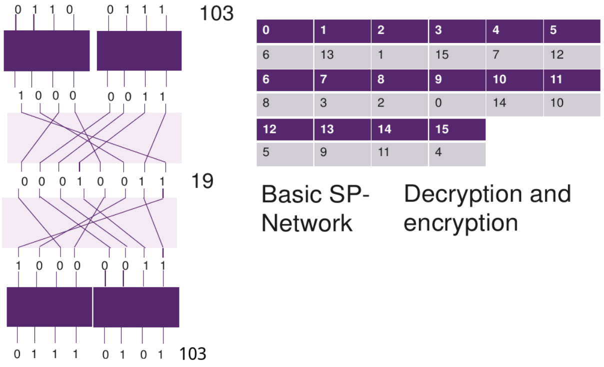
Takes in an input of a fixed size and returns an output of the same size

– Diffusion and Confusion

– SP-Network

Advanced Ecryption Standard (AES) is a Block Cipher

Here is a Block Cipher works:



0	1	2	3	4	5
6	13	1	15	7	12
6	7	8	9	10	11
8	3	2	0	14	10
12	13	14	15		
5	9	11	4		

Basic SP-Network

Decryption and encryption

AES

Built around the Rijndael algorithm

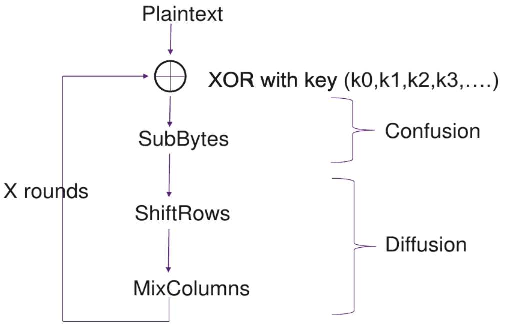
Superceedes the DES as a standard

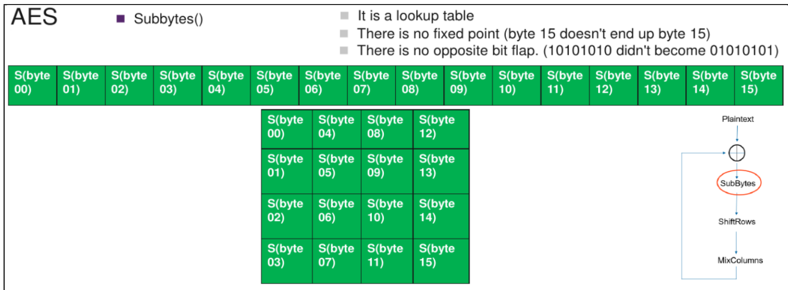
– SP-Network with 128-bit block size

» Key length 128,192,256 bit

» 10, 12 or 14 rounds

» Each Round: Substitute Bytes, ShiftRows, MixColumns, KeyAddition



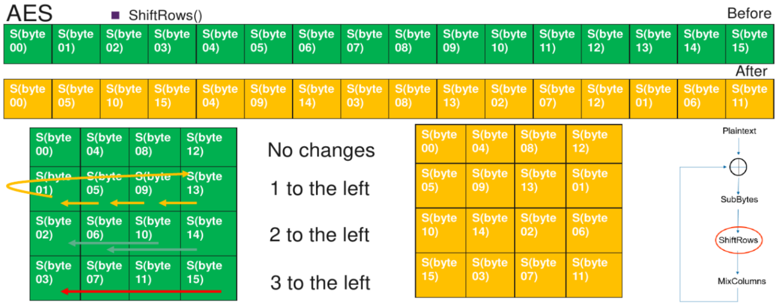


AES

- Subbytes()
  - It is a lookup table
  - There is no fixed point (byte 15 doesn't end up byte 15)
  - There is no opposite bit flap. (10101010 didn't become 01010101)

S(byte 00)	S(byte 01)	S(byte 02)	S(byte 03)	S(byte 04)	S(byte 05)	S(byte 06)	S(byte 07)	S(byte 08)	S(byte 09)	S(byte 10)	S(byte 11)	S(byte 12)	S(byte 13)	S(byte 14)	S(byte 15)

S(byte 00)	S(byte 04)	S(byte 08)	S(byte 12)
S(byte 01)	S(byte 05)	S(byte 09)	S(byte 13)
S(byte 02)	S(byte 06)	S(byte 10)	S(byte 14)
S(byte 03)	S(byte 07)	S(byte 11)	S(byte 15)



AES

- ShiftRows()

Before

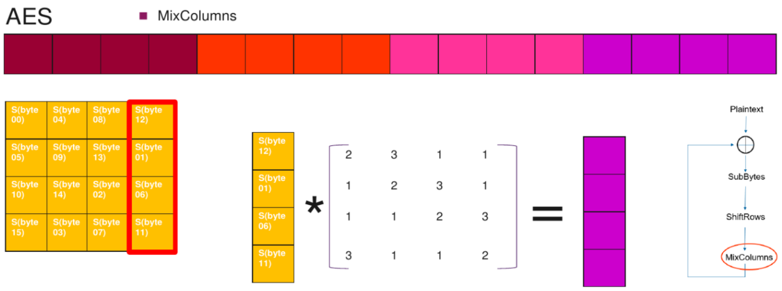
S(byte 00)	S(byte 01)	S(byte 02)	S(byte 03)	S(byte 04)	S(byte 05)	S(byte 06)	S(byte 07)	S(byte 08)	S(byte 09)	S(byte 10)	S(byte 11)	S(byte 12)	S(byte 13)	S(byte 14)	S(byte 15)

After

S(byte 00)	S(byte 05)	S(byte 10)	S(byte 15)	S(byte 04)	S(byte 09)	S(byte 14)	S(byte 03)	S(byte 08)	S(byte 13)	S(byte 02)	S(byte 07)	S(byte 12)	S(byte 01)	S(byte 06)	S(byte 11)

S(byte 00)	S(byte 04)	S(byte 08)	S(byte 12)	No changes
S(byte 01)	S(byte 05)	S(byte 09)	S(byte 13)	1 to the left
S(byte 02)	S(byte 06)	S(byte 10)	S(byte 14)	2 to the left
S(byte 03)	S(byte 07)	S(byte 11)	S(byte 15)	3 to the left

S(byte 00)	S(byte 04)	S(byte 08)	S(byte 12)
S(byte 05)	S(byte 09)	S(byte 13)	S(byte 01)
S(byte 10)	S(byte 14)	S(byte 02)	S(byte 06)
S(byte 15)	S(byte 03)	S(byte 07)	S(byte 11)



AES

- MixColumns

S(byte 00)	S(byte 04)	S(byte 08)	S(byte 12)
S(byte 05)	S(byte 09)	S(byte 13)	S(byte 01)
S(byte 10)	S(byte 14)	S(byte 02)	S(byte 06)
S(byte 15)	S(byte 03)	S(byte 07)	S(byte 11)

$$\begin{bmatrix}\text{S(byte 12)}\\\text{S(byte 01)}\\\text{S(byte 06)}\\\text{S(byte 11)}\end{bmatrix} * \begin{bmatrix}2&3&1&1\\1&2&3&1\\1&1&2&3\\3&1&1&2\end{bmatrix} =$$

Block Cypher with random input length

Obviously we want to encrypt more than just one block

How do we do that?

– Electronic Code Block (ECB)

– Cipher Block Chaining (CBC)

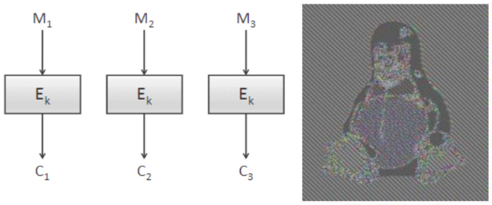
– Counter Mode (CTR)

Electronic Code Block (ECB)

Just encrypt block after block.

However, this might give away the bigger picture

Aka the pattern of the data is still visible!!

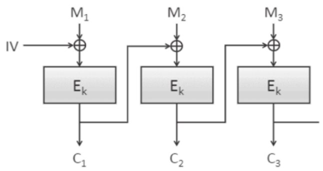


The ECB Penguin

Cipher Block Chaining (CBC)

XOR each output with the next block.

not parallelizable, but more secure than ECB



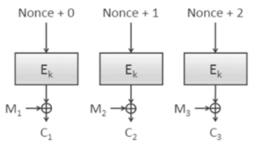
Counter Mode (CTR)

Encrypt a counter (Nonce) to produce a stream cypher

Encrypted Nonce is then XORed with the plain text

parallelizable!!

Standard for all AES ciphers!



Cons and Pros of symmetric Cryptography

– Key distribution

Keys have to be shared securely, anyone who has the key can encrypt and decrypt all messages (sent by that key)

– No Nonrepudation

Because everyone who has the key can encrypt and decrypt, there is no guarantee that this message is from a trusted source.

– No message Integrity

If the message gets damaged, then there is no recovery inbuilt.

+ Speed

often 1000 to 10000 times faster than asymmetric algorithms

Lots of processors have an AES intruction set.

Alternatives: Chacha20 cipher

Diffie-Hellman

With this method the problem of sharing a key over the internet is solved

We can now do so without any worries of giving a malicious third party access

Every TLS handshake is in some way powered by this.

We are not actually exchanging a key, only some mathematical part of it!!

Discrete Logarithm

A logarithm that is implicit when using mod.

$a^b (mod\, n) = c == b = \log_{a,n}(c)$

These are harder to calculate than regular ones!

Which is why they are used in Diffie-Hellman!

- 3^x mod 7 = 1, what is x?

This leaves us having to brute force the answer

- Brute force:
  - 3^1 (mod7) = 3 (mod 7) = 3
  - 3^2 (mod7) = 9 (mod 7) = 2
  - 3^3 (mod7) = 27(mod 7) = 6
  - 3^4 (mod7) = 81 (mod 7) = 4
  - 3^5 (mod7) = 243 (mod 7) = 5
  - 3^6 (mod7) = 729 (mod 7) = 1

What if mod 7 was mod some 2000 bit number

Primitve Root

A number g is a primitive root of p when:

$\bigvee_{k=0}^{p} g^k \; mod\, p =$ Distinct from each other

In other words, every single result from the modulo must be different!

Diffie-Hellman Example

1. Agree on Parameters

Alice and Bob agree on a large prime p and a second prime / primitive root g

p is usually at least 2048 or 4096 bits

2. Select Private Numbers

Alice picks the random number a

Bob picks the random number b

» private numbers are between 1 and p

» If p is 2048 bits, then you are guessing a number with 2048 bits, have fun :)

» They NEVER tell each other the private number

3. Alice and Bob each calculate a Public Key

» Alice calculates key:$g^a \; mod\, p$

» Bob calculates key:$g^b \; mod\, p$

Because we are using a discrete logarithm, it is mathematically infeasible to get the private numbers by calculation.

4. Alice and Bob exchange the Public Keys

These are simple the calculated versions of keys.

5. Alice and Bob calculate the shared key

for both this is: $g^{ab} \; mod\, p$

The shared key is therefore the same for both parties

6. Calculate Master Secret

The shared key is also called the Pre-Master

This is because the shared key is quite big and not often used to encrypt directly

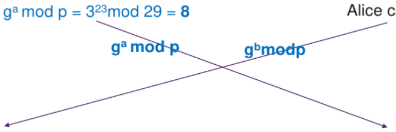
It is instead used to control sessions after it has been hashed

The hashed shared key is then called the Master Secret

Alice and Bob agree on **g = 3** and **p = 29**

Alice chooses **a = 23**
Alice calculates g^a mod p = 3^{23}mod 29 = **8**

Bob chooses **b = 12**
Alice calculates g^b mod p= 3^{12}mod 29 = **16**



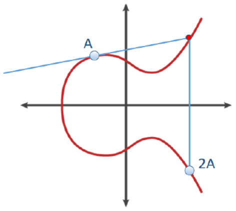
Alice calculates: $(g^b)^a$mod 29 = 1623 mod 29 = *24*
The shared secret is **24**

Bob calculates: $(g^a)^b$ mod 29 = 8^{12}mod 29 = 24
The shared secret is **24**

Public, **Private**

Eliptic Curve Cryptography

- Elliptic curves are a drop-in replacement for the mathematics underpinning regular Diffie-Hellman.
  - If you look into your browser, you see something like ECDHE
  - Elliptic curve Diffie Hellman is now becoming the standard
- Elliptic curve is a two dimension curve
  - y2=x3+ax+b
  - The private key is a number
  - The public key is composed of two numbers(X,Y) number



- Elliptic curves are much stronger than traditional public-key schemes for the same key length.

Symmetric	Diffie-Hellman and RSA	Elliptic Curve
56	512	112
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Ephemeral Mode

For Diffie-Hellman this means calculating a new key for every session
This is also called Perfect Forwarding Secrecy
The reason for this is that calculating this key costs close to no power
So why not just create a new one for every session to increase security?
This is usually done every browser refresh etc.
It is also automatically done after a certain time, again to improve security

RSA Rivest-Shamir-Adleman

– Public Key cryptosystem
– widely used for secure data transmission
– Most common method for public cryptography
– Offers Nonrepudation!!
– Reverseable Keys

Both the public key and the private key can be used to encrypt or decrypt.
It just has to be the inverse at the other end.
Encrypt pub-key -> decrypt priv-key | encrypt priv-key -> decrypt pub-key

» The reverseable keys lead to 2 operating Modes

1. Encrypt so only the receiver can read

If I want to send a message to the server that only said server can read, then I can encrypt my message with the servers public key.
The server then uses its private key to decrypt. Not even you can decrypt it :P

2. Nonrepudation Mode

By encrypting with the private key, everyone who has the public key can read.
However, it guarantees that said message is from you, and no one else!

Prime Factorization

Every non-prime number has a prime factorization
This means every non-prime number can be created by multiplying x primes
Prime factorization of 30 -> 5 * 3 * 2
Calculating the prime factorization is EXTREMELY HARD for big numbers
In other words, it is not feasible to calculate it with a current Computer
Which is why it is used by the RSA algorithm

RSA Functionality

Public keys: e n
Prime factors: p1,p2
Private key: d
Message: m

e is almost always 3 or 65537
n is a random Prime factorization of 4096 bits

$$d = \frac{k * \phi(n) + 1}{e}$$

where k is a random integer
p1,p2,d must be private!!

– Encrypting:
$c = m^e \, mod \, n$

– Decrypting:
$m = c^d \, mod \, n$

– Combined
$m^{ed} \, mod \, n = m$

The PHI Function

- Euler has studied the distribution of prime numbers and invented the PHI function
  - ϕ(n)
  - It outputs how many integers are less than or equal to n that do not share any common factor with n.

ϕ(8)= 4 1 2 3 4 5 6 7 8 ϕ(7)= 6 1 2 3 4 5 6 7

In red, the numbers that are relatively prime with n.
coprime

counts for prime:
PHI = prime - 1
Φ(21377)= 21376

- Euler theorem

$\boxed{m^{\phi(n)} = 1 \text{ mod n}}$ — $\boxed{m^{e*d} = \text{m mod n}}$

$1^k = 1$
$m^{k*\phi(n)}$ = 1 mod n
$m*m^{k*\phi(n)}$ = m mod n
$m^{k*\phi(n)+1}$ = m mod n
where k is an integer

⇒ ed = (k*ϕ(n)+1)
d = (k*ϕ(n)+1)/e

- It is computationally infeasible to calculate a private from a public key.
- This is achieved through intractable mathematical problems.
- In order to calculate ϕ(n) easily, you have to know p1 and p2 where p1*p2=n

Using RSA

1. Choose two very large Primes

- Alice randomly generates a prime number p1
  - p1=

1838476548476453627840298364538745366746352928736539382279438292033874693937373798201992394739099928334411737377921100388376547382653873638268837628763887439987367922O
29874635373829288393829476555281818330900309887654337893393875648382719485736

- then a second randomly generated number p2
  - p2=

384756472888882761991003756281937265917476294619766541123456229986747562927562782928473900193847699920000292827727347776299841100093857693929385939859093002900000934 10
0092581100000945586666883322956685492858000000000000003333000556600000000000072

- Alice calculates n
  - n=p1xp2

1948475948398729829989937500022229918855587630022911134949494498739995577575799222200000000000044550000000000039998858775777775666339929922929292929292929929947765767548
856888788787300001000294857587384784839899999999285478567575837829000000111
11111844775663829476634929478758498287948880000292848587584854848584588472838573628875620948573911111009284856690019298384758549499999388574666464646462837838373783738
2887882738389783877887778282834883888388887378887748758837585858885855784986548

- Alice hides p1 and p2 (p1 and p2 are secret)
- n is public

2. Calculate PHI

$\phi(n) = \phi(a) * \phi(n)$

a and b are the prime factorization primes!

if number is prime, then: $\phi(a) = a - 1$

Ex. n = 77, a = 11, b = 7 -> $\phi(77) = \phi(11) * \phi(7) = 10 * 6 = 60$

3. Choose k and e to calculate d

$$d = \frac{k * \phi(n) + 1}{e} \text{ with k being an integer}$$

Ex. n=55, e=7, k=4, $\phi(n) = 40$ $d = \frac{4 * \phi(55) + 1}{7} = 23$

RSA quirks

– Very weak with short messages

To mitigate this, padding is added
Optimal assymetric Encryption Padding (OAEP) is used
pseudo random padding that introduces an IV then hashes it
Server has to create same padding to check if it matches up

– Not common to encrypt with RSA!

TLS used RSA before but no longer.
RSA is used more for signing! Something that Diffe can't!!

– RSA is 1000x slower than symmetric crypto systems!!

