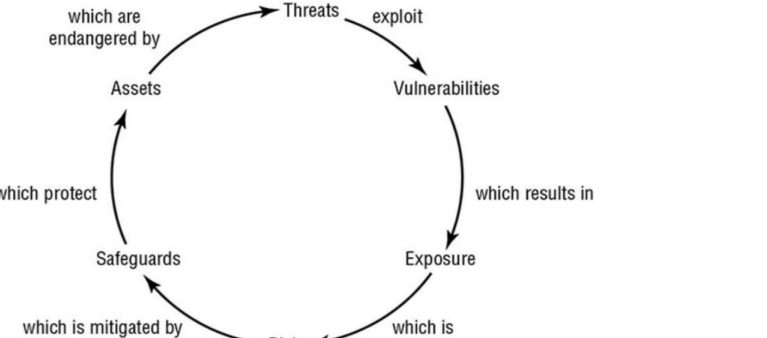
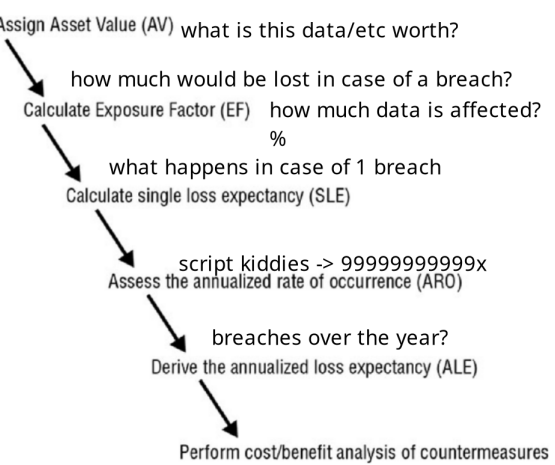


Asset Anything within the organization that is worth protecting: Information, Systems, Devices, Facilities, Personnel, Intellectual Property	CIA triad The primary goal of Security infrastructure: – Confidentiality: prevention of unauthorized access to data no matter if the data is in transit, in storage 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 Usual done with Redundancy, backups
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.	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
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.	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
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.	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
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	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.
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.	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.
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.	Risk Analysis – Evaluation, assessment, assignment 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
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.	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
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	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

 <p>which are endangered by</p> <p>Assets</p> <p>Threats</p> <p>exploit</p> <p>Vulnerabilities</p> <p>which results in</p> <p>Exposure</p> <p>which is</p> <p>Risk</p> <p>which is mitigated by</p> <p>Safeguards</p> <p>which protect</p>	<h3>Privacy</h3> <ul style="list-style-type: none"> - GDPR <ul style="list-style-type: none"> - 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 <ul style="list-style-type: none"> - 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
<h3>Quantitative and Qualitative Risk Analysis</h3> <ul style="list-style-type: none"> - 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 <p>Both methodologies are necessary for complete risk analysys!</p>	<h3>Access Control</h3> <ul style="list-style-type: none"> - 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
<h3>Quantitative Risk Analysis</h3>  <ul style="list-style-type: none"> - 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 $ALE = SLE * ARO$ <p>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</p> <ul style="list-style-type: none"> - Safeguard Costs <p>First, compile a list of safeguards against each threat. Assign each safeguard a deployment value -> Annual Cost of Safeguard (ACS)</p> <ul style="list-style-type: none"> - Safeguard Cost/Benefit <p>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!</p>	<h3>Steps of Access Control</h3> <ol style="list-style-type: none"> 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!
<h3>Dealing with Risk</h3> <ul style="list-style-type: none"> - 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. 	<h3>Authentication Factors</h3> <ul style="list-style-type: none"> - 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. <p>Type 1 is the weakest form of authentication and type 3 the strongest!</p>

<ul style="list-style-type: none">– Location <p>This is a form that is used in combination with others.</p> <p>Ex: A certain IP might be a requirement to log into one of your systems</p> <p>Or your bank might block a transaction if it isn't from your country of residence.</p>	<ul style="list-style-type: none">– Access Control Matrix <p>This writes Objects,Subjects and privileges into a table</p> <p>If Subject tries to access an object the table for said object is checked</p> <ul style="list-style-type: none">– Capability Table <p>This is the same as the Access Control Matrix but with a subject focus</p> <p>In this table the subject and all accessible Objects are written down</p> <ul style="list-style-type: none">– Content-Dependent Control <p>Constrict Access to the data within an Object</p> <p>In a database a user might be able to check table 1 but not table 2.</p> <p>While the object is the entire database!</p> <ul style="list-style-type: none">– Context-Dependent Control <p>Give a subject access depending on what the subject does</p> <p>Ex. The checkout button in an online shop only works, if you have something in the shopping cart.</p> <ul style="list-style-type: none">– Need to Know <p>Subjects should only have access to Objects they need to do their job.</p> <ul style="list-style-type: none">– Least Privilege <p>Subjects should only have the privileges they need to do their job.</p> <ul style="list-style-type: none">– Separation of Duties and Responsibilities <p>No single person should have total control over the entire System!</p>
<ul style="list-style-type: none">– Multifactor Authentication <p>Multifactor Authentication simply uses more than one type to authenticate</p> <p>This might be a password(Type1) and a token(Type2)</p> <p>The strongest form of authentication if therefore all 3 types together!</p> <ul style="list-style-type: none">– Passwords <p>Passwords are never stored in plain text, they might be stolen easily with 1 breach</p> <p>Instead, they are saved with a hash-algorithm, makes them nearly useless to threat actors</p> <p>Unless they also have the access to said algorithm.</p> <p>Types of passwords:</p> <ul style="list-style-type: none">– Plain Password <p>use numbers, characters and special symbols with a length of at least 10.</p> <p>10 characters -> 928 years of cracking!!</p> <p>Don't use personal information such as names etc.</p> <ul style="list-style-type: none">– PassPhrases <p>Passphrases are chained words that are easy to remember</p> <p>It is important to have a longer passphrase than a regular password!</p> <p>Still use special symbols. Also try to include random upper-lower case spelling</p> <p>Or leadspeak slnc3 that ls quite 3f3ctlv3!!</p> <ul style="list-style-type: none">– Cognitive Passwords <p>A series of personal questions. -> what is the name of your first pet?</p> <p>Best way to do this is letting users create both the question and the answer!</p> <ul style="list-style-type: none">– SmartCards <p>Card used for identification / authentication. Often integrates key encryption</p> <p>Usually temper resistant</p> <p>One downside, loss of card might give a threat a window of exploitation!</p> <ul style="list-style-type: none">– Tokens <p>Generated by a special Device.</p> <p>Regular password generators: any device can take that place. ex. smartphone</p> <ul style="list-style-type: none">– Synchronous Dynamic Token <p>Time-based One-Time Password (TOTP)</p> <p>Device generates Token every x seconds.</p> <ul style="list-style-type: none">– Asynchronous Dynamic Token <p>HMAC-based One-Time Password (HOTP)</p> <p>Device generates one time token based on algorithm. Stays until used!</p>	<ul style="list-style-type: none">– Common Access Control Attacks <ul style="list-style-type: none">– Access Aggregation Attacks (Passive Attacks) <p>This is the collection of nonsensitive data, that combined could give a threat actor the opportunity to launch a proper attack.</p> <p>Ex. IP address, open ports, Operating System -> specific exploit</p> <ul style="list-style-type: none">– Password Attacks (Brute Force) <p>Spam random sequences until you get the right one</p> <ul style="list-style-type: none">– Dictionary Attacks (Brute Force) <p>Try passwords from a list of passwords, example leaked password list.</p> <p>Can also be done with list of common passwords, or slightly changed previous passwords (One-Upped-Passwords -> 1 character changed)</p> <ul style="list-style-type: none">– Birthday Attack (Brute Force) <p>Try to get the same hash as the password with a different sequence</p> <p>Can be mitigated by using better hashing algorithms. SHA-3 instead of md5</p> <p>Note the attacker needs access to the hash in order for this to work!</p> <ul style="list-style-type: none">– Rainbow Table Attacks <p>Combines the Birthday attack with a table of precomputed hashes.</p> <p>This is then used to compare to a password hash list.</p> <ul style="list-style-type: none">– Sniffer Attacks <p>Threat actor analyzes data sent over network with a sniffer tool.</p> <p>A good example for this is wireshark</p> <p>Can be mitigated by using encryption and One-Time passwords</p> <p>encryption makes the data useless and One-Time passwords are as well</p> <ul style="list-style-type: none">– Spoofing Attacks <p>Pretending to be something/someone else. Ex. pretending to be router.</p> <ul style="list-style-type: none">– Social Engineering Attacks <p>Gaining and then misusing trust of someone.</p> <p>*Indian accent* you get refund if you buy me 2 cards from target</p> <ul style="list-style-type: none">– Shoulder Surfing (Social Engineering) <p>Reading information on a screen from a persons back.</p> <ul style="list-style-type: none">– Phishing (Social Engineering) <p>Trick a Person to click on a fake link to log in, giving the attacker all the credentials to log-in</p> <ul style="list-style-type: none">– Spear Phishing (Social Engineering) <p>Targeted Phishing at a group. Ex. Employees at company x.</p> <ul style="list-style-type: none">– Whaling (Social Engineering) <p>"Phishing für grosse Fisch" -> CEOs etc</p> <ul style="list-style-type: none">– Vishing (Social Engineering) <p>Phishing via VOIP or instant messaging</p>
<ul style="list-style-type: none">– Access Control Models (Authorization) <ul style="list-style-type: none">– Discretionary Access Control (DAC) <p>Every object has an owner, and that owner can to grant or deny permissions</p> <p>NTFS from windoof uses this</p> <ul style="list-style-type: none">– Role Based Access Control <p>Permissions are assigned to roles not users. Users are assigned to roles.</p> <p>Users who are in a role with said privileges can use them.</p> <ul style="list-style-type: none">– Rule Based Access Control <p>Global rules that are applied to all subjects</p> <p>Good example is a firewall, which applies said rules equally to all subjects</p> <ul style="list-style-type: none">– Attribute Based Access Control <p>Similar to Rule based Control but with additional attributes</p> <p>This could give one subject more rights than another</p> <ul style="list-style-type: none">– Mandatory Access Control <p>Use of labels applied to both subject and Object</p> <p>If user has the same label as a file, then user has access to it.</p>	<ul style="list-style-type: none">– Protection Mechanism <ul style="list-style-type: none">– Layering (defense in depth) <p>Multiple Controls in Layers, if one fails, there are still the other ones</p> <ul style="list-style-type: none">– Abstraction <p>Combining Objects into groups in order to simplify permission management</p>

<ul style="list-style-type: none"> - 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. 	<p>Manually set IP address (abando): <code>edit /etc/network/interfaces</code></p> <p>Configure SSH: <code>edit /etc/ssh/sshd_config</code> For birbs sake, use a nonstandard port for ssh :)</p> <p>Check current shells: <code>ps</code></p> <p>Check current processes: <code>htop</code></p> <p><code>grep</code> read lines <code>grep 'Warning' /var/log/rkhunter.log</code></p> <p>Read Lines: <code>awk 'sshd.*invalid user/ {print \$11}' auth.log</code></p> <p>bit-by-bit copy: <code>dd if=<media/partition> of<image_files></code> <code>mount - o ro,noexec,loop evidence_01 /mnt/investigation</code> mount with read only and no execution</p> <p>Check recently changed files <code>ls -lasrt</code></p>
<p>Red-Team Offensive Cyber-Security: simulate attacks</p> <ul style="list-style-type: none"> - 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 	<p>PID The unique identifier the kernel gives each process This shows both background and foreground applications</p>
<p>Blue-Team Defensive Cyber-Security: prevent attacks</p> <ul style="list-style-type: none"> - 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 	<p>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.</p> <p>notable logging systems/files: <code>syslog, rsyslog, var/log ,auth.log</code></p> <p>The shred command with -f and -n force deletes log files. Mainly used like this: <code>shred -f -n 15 /var/log/auth.log*</code> This shreds 15 lines from every log file with the name auth.log(something)</p> <p>other things to look out for:</p> <ul style="list-style-type: none"> - set-UID Rogue Files - Directories with .something Hidden... - Regular files in the /dev directory - Recently modified files <code>ls -lasrt</code>
<p>Linux Adding user: <code>usermod -m username -s /path/to/shell</code> Change shell: <code>chsh -s /path/to/shell</code> Change password: <code>passwd username</code> Add user to group: <code>usermod -a -G groupname username</code> Change file permission: <code>chmod permission file</code> Change file owner: <code>chown file user/group</code> Check IP address: <code>ip addr / ip -c -brrie a</code> DNS query: <code>dig domain</code> (dig shitgaem.online)</p> <p>- File System Permissions r = read, w = write , x = execute</p> <p>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:</p>	<p>Schedules Tasks Schedules tasks can be written either in cron.d or with systemd</p>
	<p>IP-Tables name one reason not to use ufw...</p> <p>Flush all rules: <code>iptables -F</code> Block Input: <code>iptables -P INPUT DROP</code> Block Output: <code>iptables -P OUTPUT DROP</code> Block Forward: <code>iptables -P FORWARD DROP</code> Allow Port: <code>iptables -A INPUT/OUTPUT -p port ...other shit...</code> Show rules: <code>iptables -L -n -v - line-numbers</code></p>
	<p>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!</p>
	<p>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 <code>chmod 777</code>, but a confidential process still can't open it. This is called MLS in SELinux and is related to Multi Category Security (MCS)</p>
	<p>Snort Detection software like rkhunter</p>
	<p>NetCat This can be used for anything dealing with TCP and UDP. You can also use it to control compromised systems...</p>
	<p>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.</p>
	<p>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</p>

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

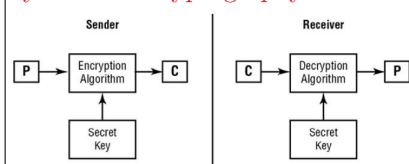
This Cipher is UNBREAKABLE!

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

No proper way to transmit, store a key

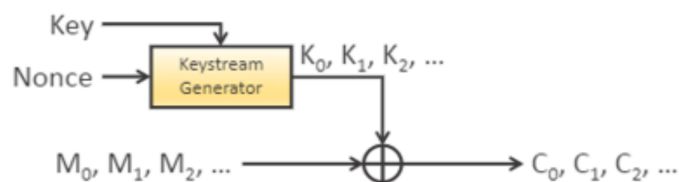
Using a key twice == Cipher broken

Symmetric Cryptography



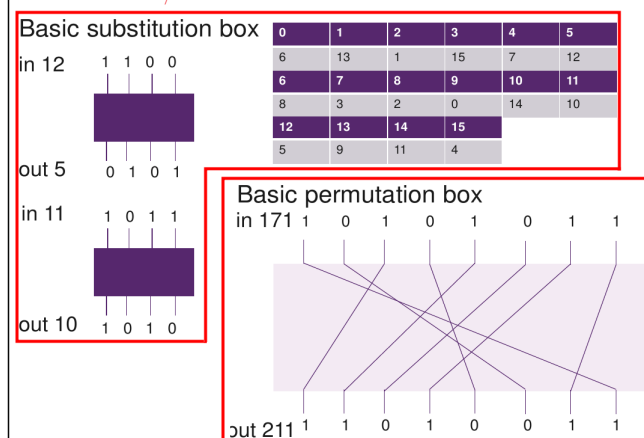
- **Same key for encrypting and decrypting**
- **Shared key for all parties involved!**
If one leaks the key, the cipher is broken!
- **Doesn't confirm identity!**
Anyone who has the key can pretend to be another

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



Block Cipher

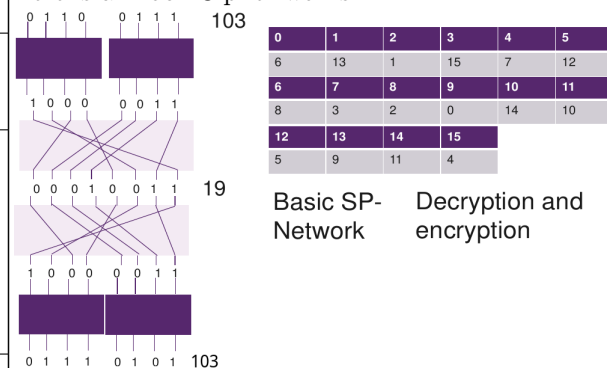
Takes in an input of a fixed size and returns an output of the same size

– **Diffusion and Confusion**

– **SP-Network**

Advanced Eryption Standard (AES) is a Block Cipher

Here is a Block Cipher works:



AES

Built around the Rijndael algorithm

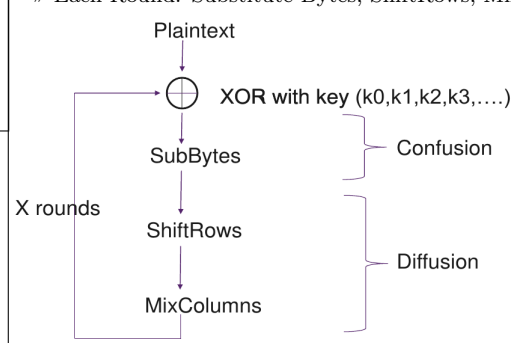
Superceedes the DES as a standard

– **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)

Plaintext

Subbytes

ShiftRows

MixColumns

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)
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)

No changes

1 to the left

2 to the left

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)

Plaintext

Subbytes

ShiftRows

MixColumns

