IoT - Security

http://standardsinsight.com/iot/iot-interoperability-requires-security

Zurich University
of Applied Sciences
School of
Engineering
InES Institute of
Embedded Systems

Basic Concepts

Security Goals - CIA

- Confidentiality
  - Sensitive data must be protected from unauthorized read accesses
- Integrity
  - Data and systems must be protected from unauthorized modification
- Availability
  - Information must be available when needed

https://aadamov.wordpress.com/2015/04/02/cia-triad-fundamental-concept-of-information-security/

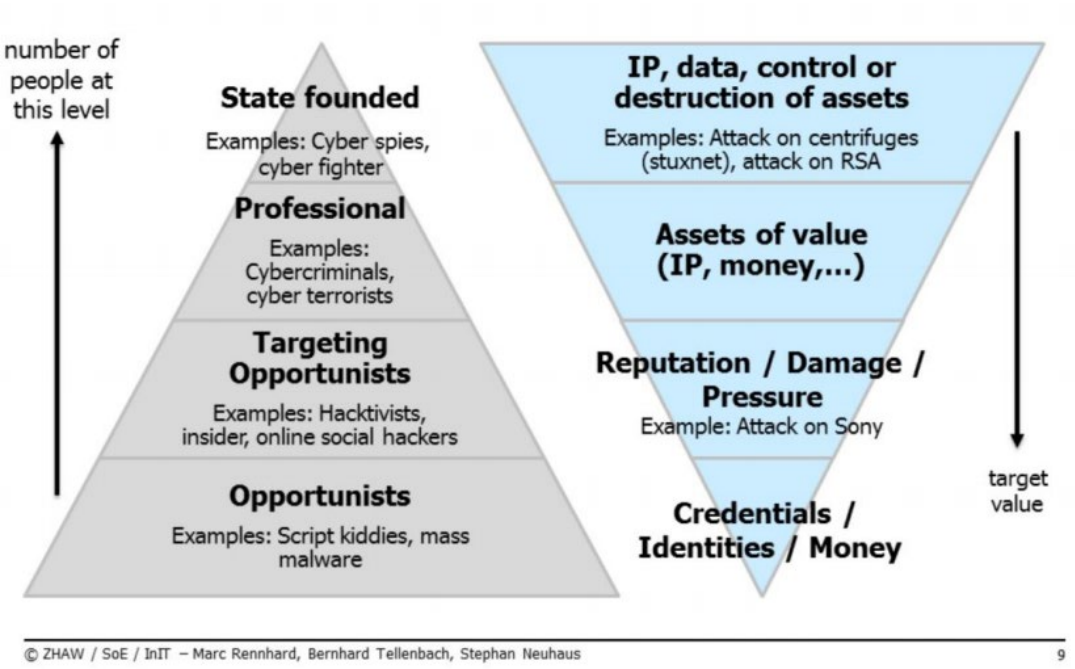
IS_IntroSoftwareSecurity.pdf

Learning Aim: The student will be able to explain all the security relevant terms defined

Properties

- Asset
  - A resource of value which is worth protecting
- Risk
  - The harm potential to assets by an attack
  - Risk = probability * impact
- Countermeasure
  - Any action device, process or technique that reduces a risk.

IS_IntroSoftwareSecurity.pdf

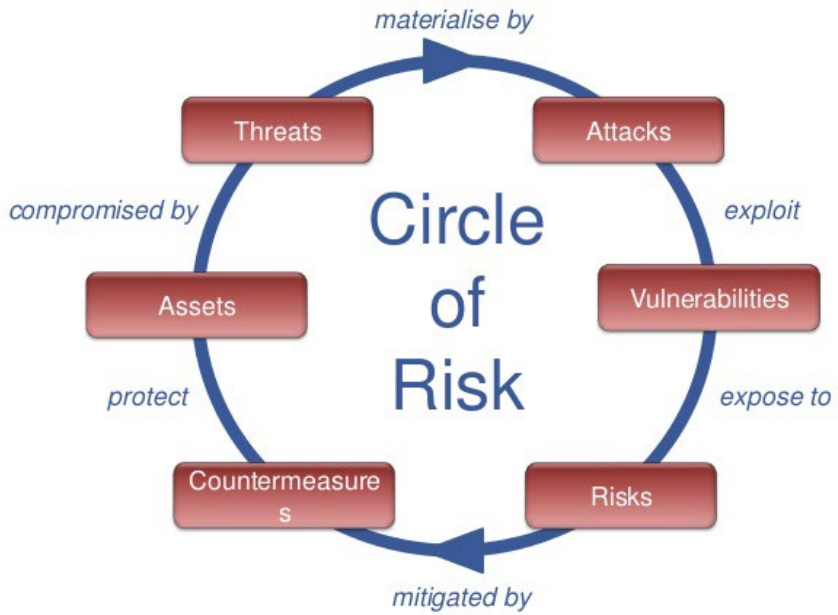


IS_Thread_Landscape.pdf

Properties

- Vulnerability
  - A defect or a flaw an attacker can exploit
- Threat
  - Possible danger that may exploit a vulnerability
- Exploit
  - The attack that takes advantage of a vulnerability

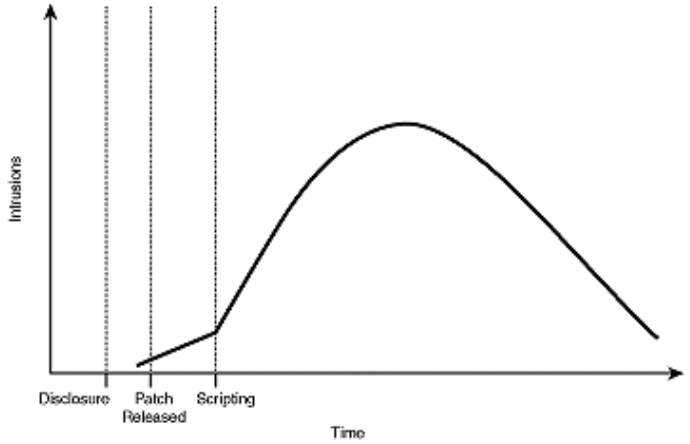
IS_IntroSoftwareSecurity.pdf



https://espincorp.wordpress.com/tag/vulnerability/

Approaches

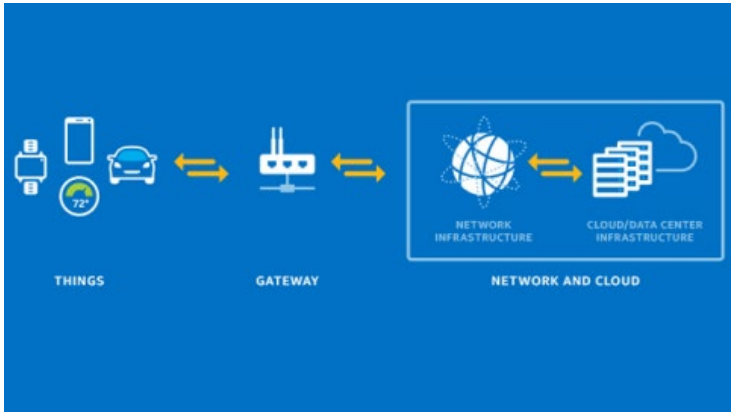
- Penetrate and Patch
  - A defect or a flaw an attacker can exploit
  - Who finds vulnerability?
  - Is patch correctly tested/proven not to introduce new vulnerabilities
  - Will user install patch?



http://www.informit.com/articles/article.aspx?p=23950&seqNum=7

Learning Aim: The student will be able to explain the pros and cons of the three security approaches

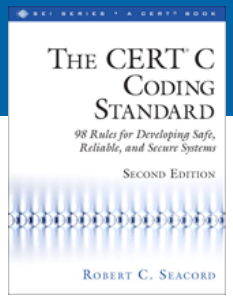
- Network Security Devices
  - So many attack types impossible to recognise them all
  - Also need patches
  - Configuration an issue



IS_IntroSoftwareSecurity.pdf

http://www.intel.com/content/www/us/en/internet-of-things/gateway-solutions.html

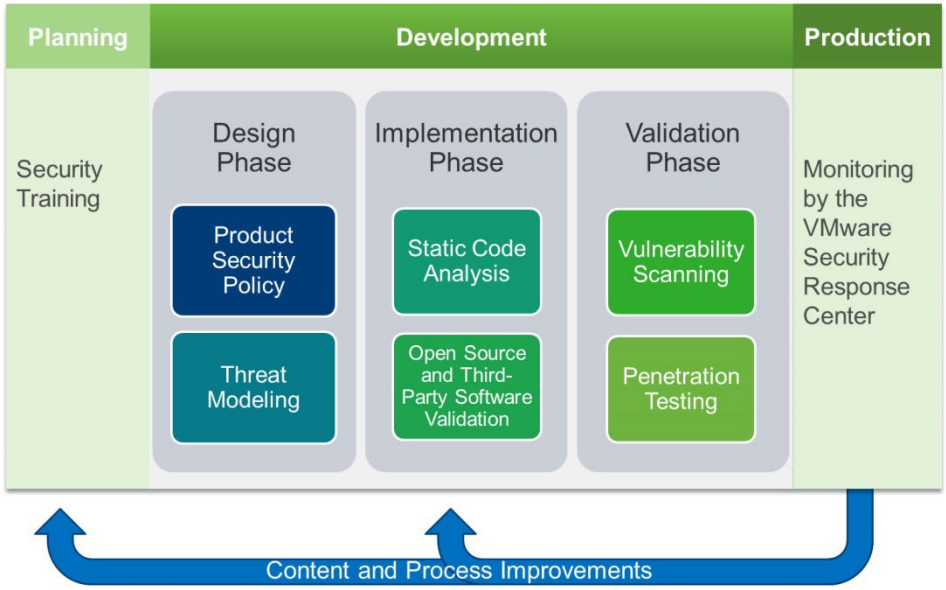
Approaches



http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=85158

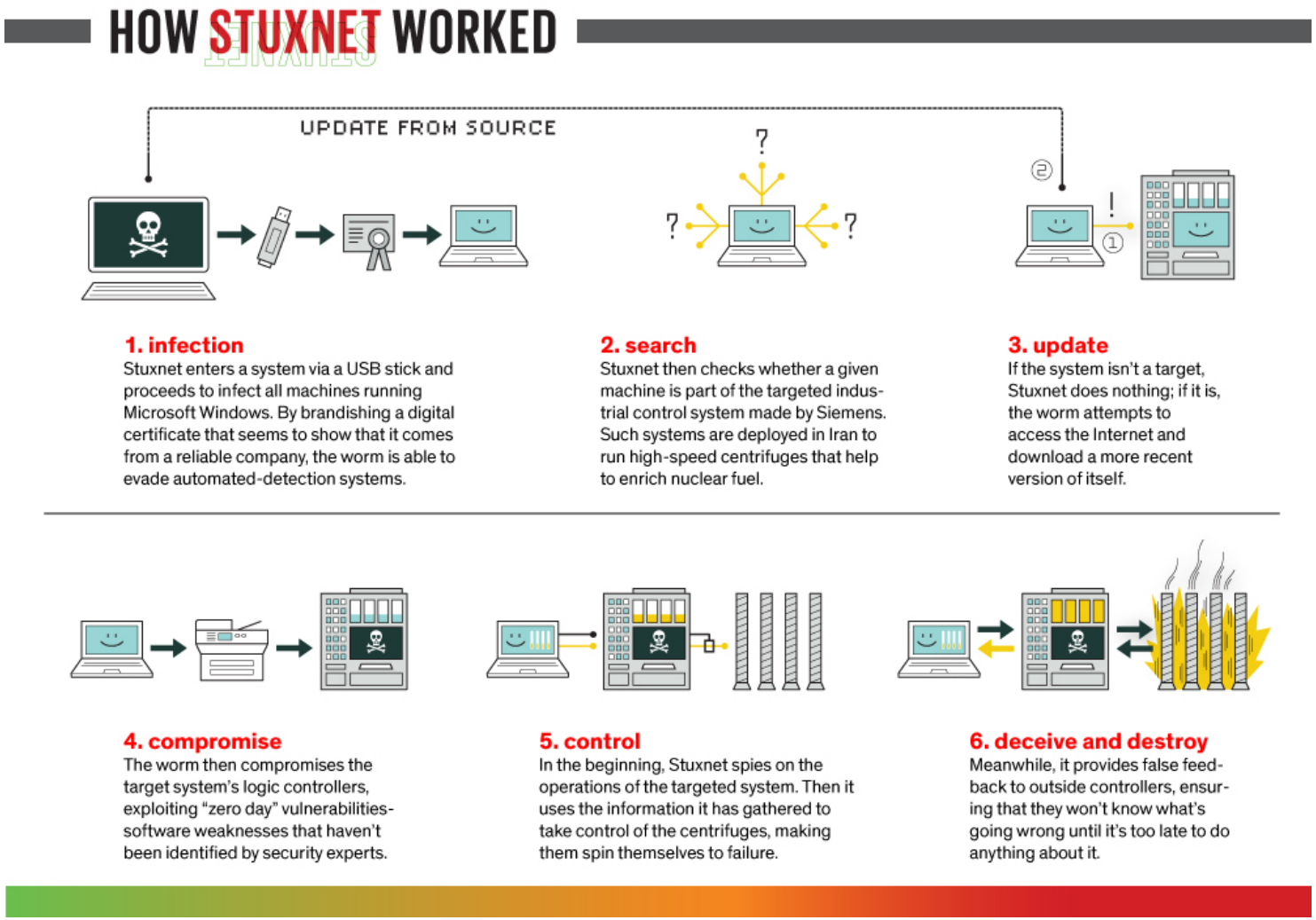
- Secure Development Lifecycle
  - Security Activities applied during the phases of SW development
  - Waterfall or agile development methodologies
  - Requires additional know-how in modelling, vulnerability scanning methods and tools and penetration testing
  - **Learn to think like an attacker**

IS_IntroSoftwareSecurity.pdf



http://www.vmware.com/security/sdl.html

A famous case

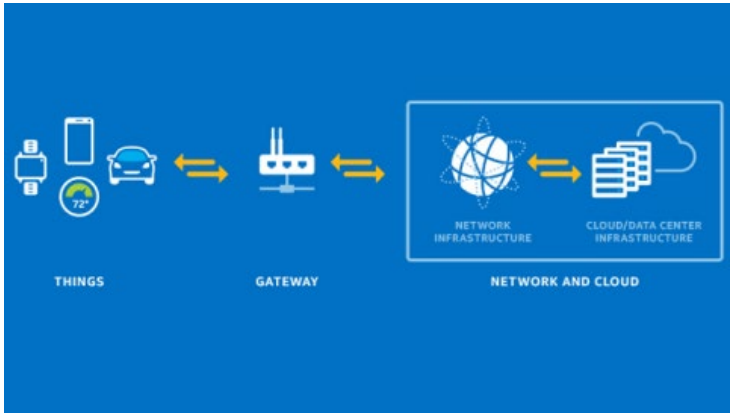


https://sharkscale.wordpress.com/2016/02/06/defending-against-stuxnet/

IoT Approaches

Zurich University of Applied Sciences
School of Engineering
InES Institute of Embedded Systems

- Fundamental IoT Architecture
  - Current architectures all assume gateway from IoT devices to cloud.



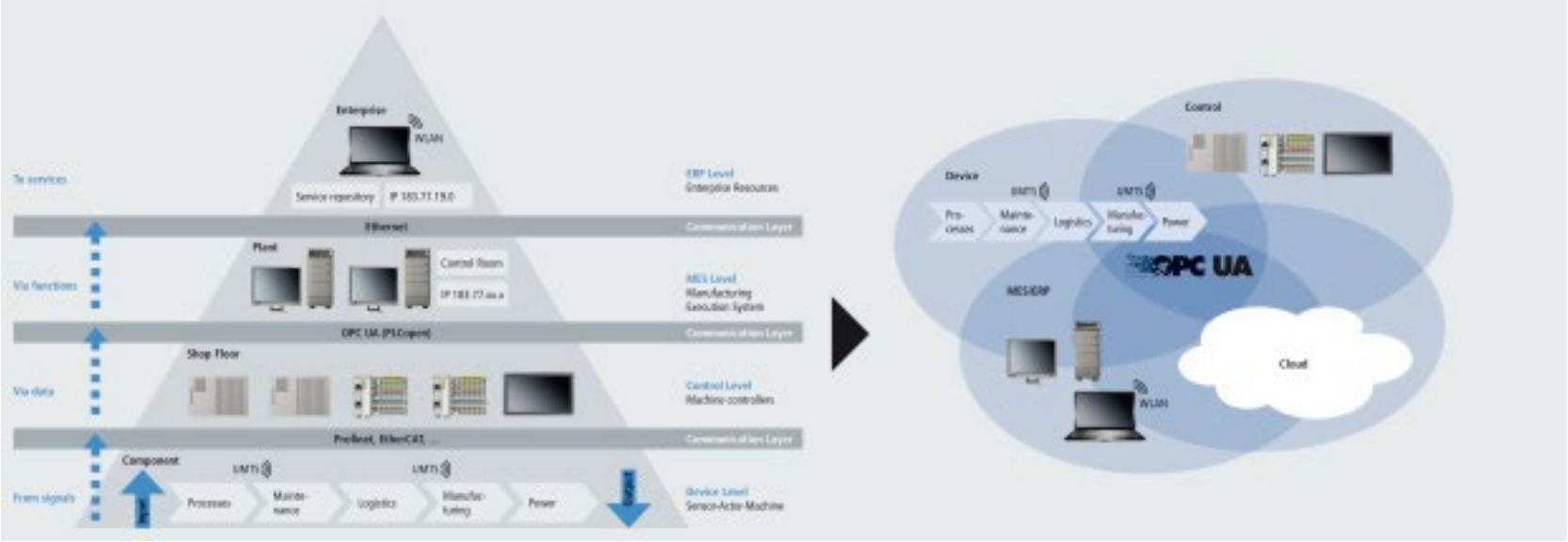
http://www.intel.com/content/www/us/en/internet-of-things/gateway-solutions.html

- Gateways
  - Need multi-protocol gateways

http://hackerboards.com/multiprotocol-iot-gateway-has-linux-and-android-bsps/

IoT Approaches

- Industrial IoT Architecture
  - Here the assumption is that the traditional automation pyramid morphs to an extended cloud based on the premise of a unified communication protocol and device/system modelling such as OPC-UA
  - Hidden assumption is that more system devices are connected to the Internet rather than being hidden by a gateway.
  - OPC-UA has been certified «sicher» https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/OPCUA/OPCUA.pdf?__blob=publicationFile&v=2



http://www.ee.co.za/article/plc-forerunner-industry-4-0-internet-things.html

Case Study

Analysis of a Gateway solution for RT communication

- Due to RT constraints on decryption the system MUST be hidden by a gateway

Learning Aim: The student will be able to explain the methodology used in this case-study

STRIDE Threat Model

- STRIDE – Threat model developed by Microsoft

STRIDE Threat List		
Type	**Examples**	**Security Control**
Spoofing	Threat action aimed to illegally access and use another user's credentials, such as username and password	Authentication
Tampering	Threat action aimed to maliciously change/modify persistent data, such as persistent data in a database, and the alteration of data in transit between two computers over an open network, such as the Internet	Integrity
Repudiation	Threat action aimed to perform illegal operation in a system that lacks the ability to trace the prohibited operations.	Non-Repudiation
Information disclosure.	Threat action to read a file that they were not granted access to, or to read data in transit.	Confidentiality
Denial of service.	Threat aimed to deny access to valid users such as by making a web server temporarily unavailable or unusable.	Availability
Elevation of privilege.	Threat aimed to gain privileged access to resources for gaining unauthorized access to information or to compromise a system.	Authorization

Analysis by STRIDE (1)

ID	SubID	STRIDE Categrov	Threat Description	Procedure	Effect	Prerequisites
1	1	Spoofing	Attacker pretends to be an IO-Controller.	Spoofing Destination MAC address (change forwarding-table in Switch).	Attacker disrupts switch and its bridging	- MAC address of IO-Controller must be known - access to PROFINET subnet required
		Spoofing	Attacker pretends to be an IO-Controller.	Spoofer transmits I/O frame to device	Device recieves frames with spurious timing: Dependent on I/O device	Address of I/O device must be known - Cycle counter must be known and replicated

Topology	Protocol	System Phase	Exploitability	Technical Impact
star	any	Anytime	Average	Severe (Low - with
any	RTC1..3	Running	Easy	Severe

- Protocol by Protocol / Function by Function / Frame by Frame analysis
  - So far 62 distinct threats classified by STRIDE
  - Also classified by exploitability and technical impact
  - Results in “risk” matrix

Exploitability \ Impact	Low	Moderate	Severe
Easy	4	3	4
Average	4	18	7
Difficult	0	10	12

Figure 1: *Risk estimation matrix: Each field contains number of threats with appropriate rating.*

Analysis by STRIDE (2)

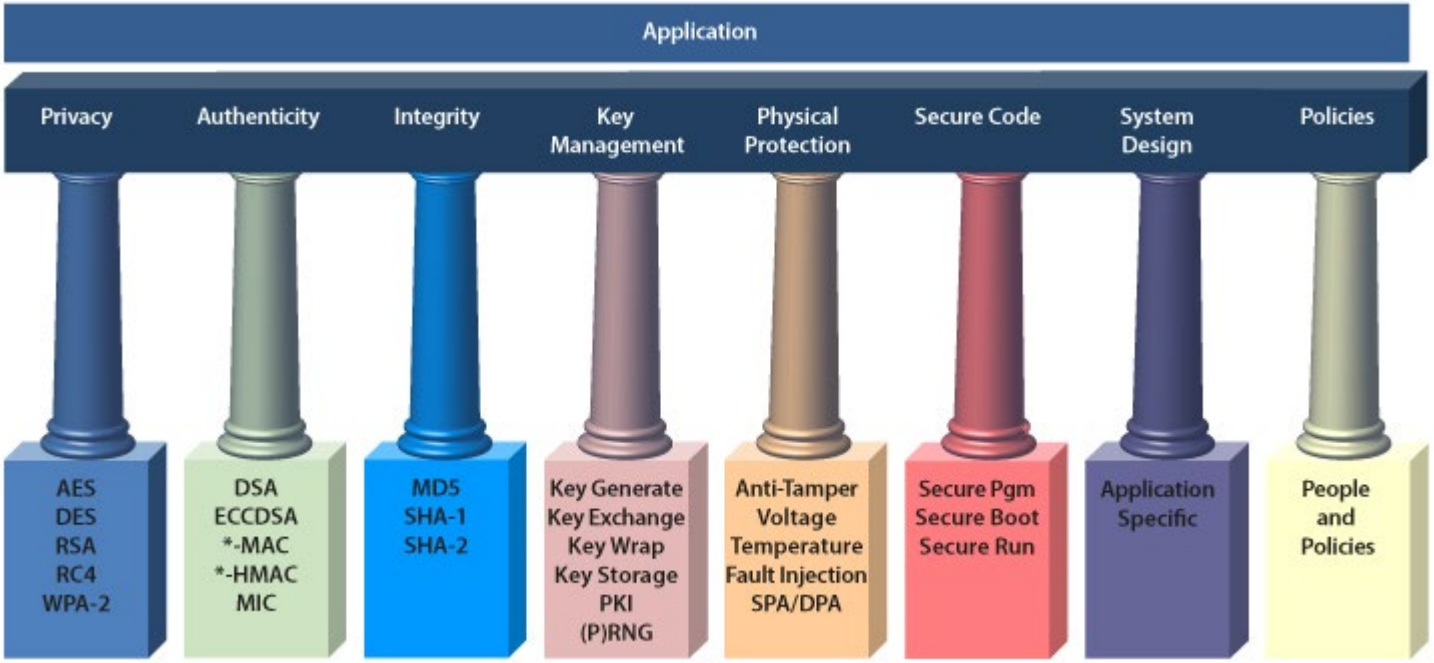
- By applying countermeasures (i.e Gateway)

ID	Countermeasure	Description	No. Threats Mitigated
1	Strong Authetication	Only maintenance engineers and other authorized personnel has access to PROFINET subnet.	48
2	Encryption	Data link between engineering station and gateway will be encrypted.	2
3	MAC Address Blacklisting	Frames from outside the PROFINET subnet containing as source MAC address one of the IO-Devices or the IO-Controllers MAC address will be blocked / dropped.	7
4	Layer 2 Frames Blocking	Layer 2 Frames (ISO/OSI) from outside into the PROFINET subnet will be blocked / dropped. (whitelisting of allowed requests possible)	27
5	RPC Blocking	RPC requests from outside into PROFINET subnet will be blocked. (whitelisting of allowed calls possible)	8
6	Logging	Traffic through gateway will be logged (non-volatile storage) and appropriate actions performed.	8
7	Limited Login Attempts	After several login attempts, IP will be blocked (temporary, in case of recurrence permanently).	2
8	Sealed Devices	PROFINET cables shall be sealed on both ends to be able to detect physical manipulation.	5
9	Controlled Site / Restricted Access	Access only with company badge and site equipped with surveillance cameras.	5

- can reduce to:

Exploitability \ Impact	Low	Moderate	Severe
Easy	0	0	0
Average	3	1	0
Difficult	5	31	22

Embedded Security



https://aadamov.wordpress.com/2015/04/02/cia-triad-fundamental-concept-of-information-security/

TLS

MQTT uses TLS
CoAP uses DTLS

Learning Aim: The student will be able to explain the basic functionality of TLS

TLS (1)

- Secure Sockets Layer (SSL)
  - Developed by Netscape

- Name-changed to TLS during standardisation by IETF
  - Transport Layer Security
  - Sits on TCP layer
  - TLS V1.2 – RFC5246
  - TLS V1.3 in draft
  - Offers encryption, authentication, integrity

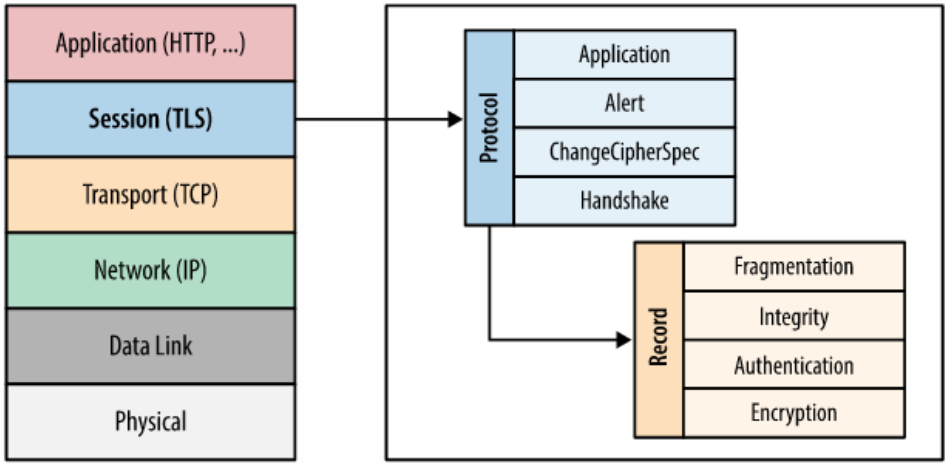


Figure 4-1. Transport Layer Security (TLS)

https://hpbn.co/transport-layer-security-tls/

TLS (2)

0 ms: TLS runs over a reliable transport (TCP), which means that we must first complete the TCP three-way handshake,

56 ms: With the TCP connection in place, the client sends a number of specifications in plain text, such as the version of the TLS protocol it is running, the list of supported cipher suites, and other TLS options it may want to use.

84 ms: The server picks the TLS protocol version for further communication, decides on a cipher suite from the list provided by the client, attaches its certificate, and sends the response back to the client. Optionally, the server can also send a request for the client's certificate and parameters for other TLS extensions.

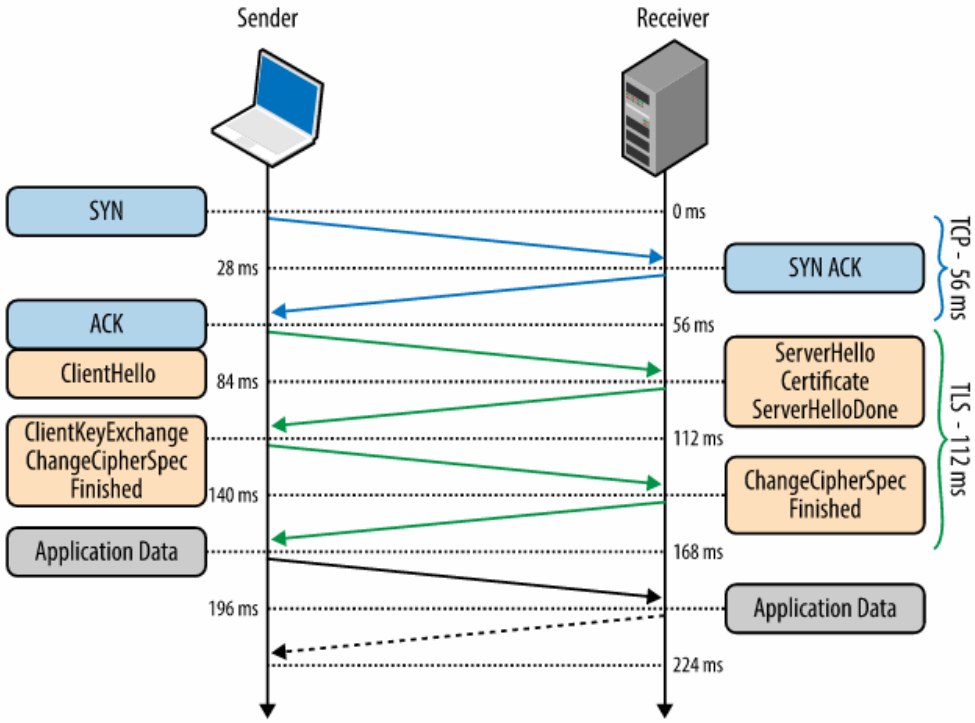


Figure 4-2. TLS handshake protocol

https://hpbn.co/transport-layer-security-tls/

TLS (3)

112 ms: Assuming both sides are able to negotiate a common version and cipher, and the client is happy with the certificate provided by the server, the client initiates either the RSA or the Diffie-Hellman key exchange, which is used to establish the symmetric key for the ensuing session.

140 ms: The server processes the key exchange parameters sent by the client, checks message integrity by verifying the MAC, and returns an encrypted Finished message back to the client.

168 ms: The client decrypts the message with the negotiated symmetric key, verifies the MAC, and if all is well, then the tunnel is established and application data can now be sent.

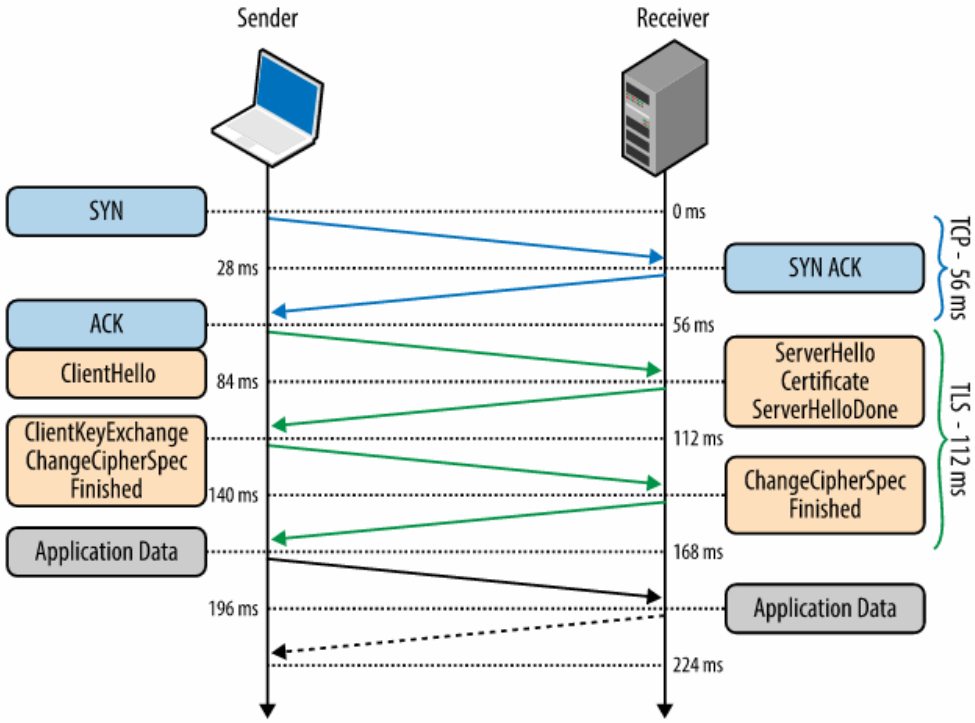


Figure 4-2. TLS handshake protocol

https://hpbn.co/transport-layer-security-tls/

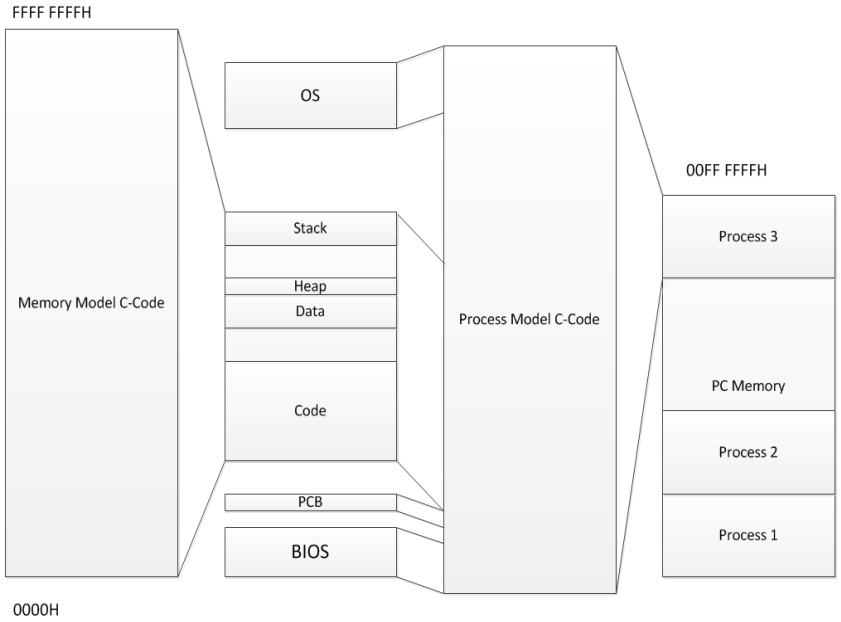
Securing Computer Systems

Try to derive the concept of a hypervisor and separation kernel by examining the basics of computer systems

Learning Aim: The student will be able to explain the hypervisor and related terms, the HW requirements to implement one and the secure boot process

Computer System Issues

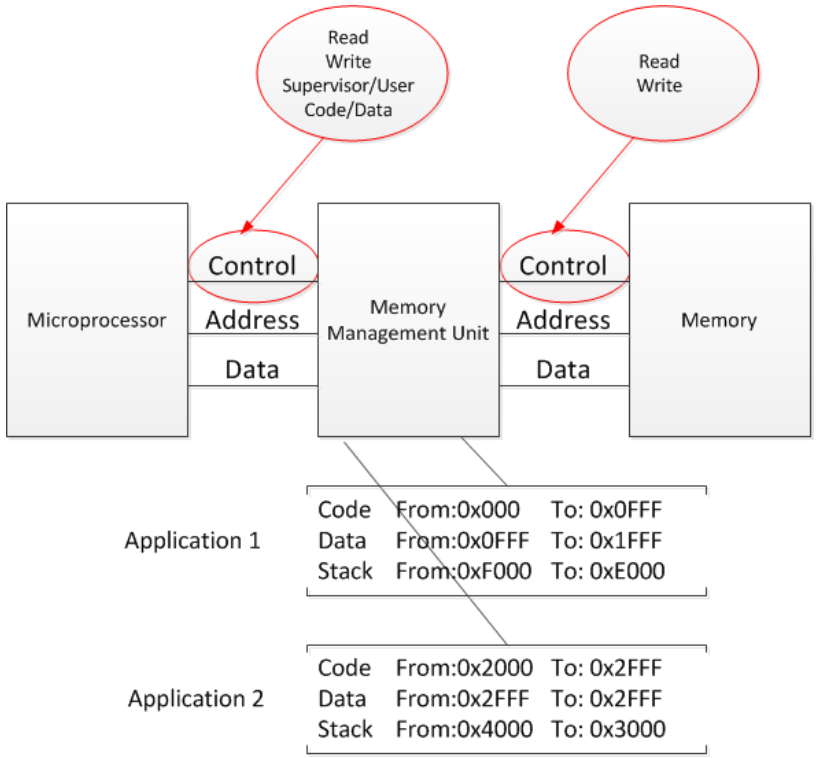
- Application
  - Assumes that the entire size_t address range is available
- Application (process) requires
  - Data, code and stack segements
  - Process memory model include OS
- Multiple Applications on one processor
  - OS manages memory
- Application 1 may not r/w into memory of application 2
- Achieved by the use of a Memory Management Unit



Computer System Issues

- Memory Management Unit
  - Maintains a table with access rights per application
  - Is re-programed every time a context switch takes place
  - Throws an exception if code attempts to access a non-permissible area
    - Null pointer exception / segmentation fault

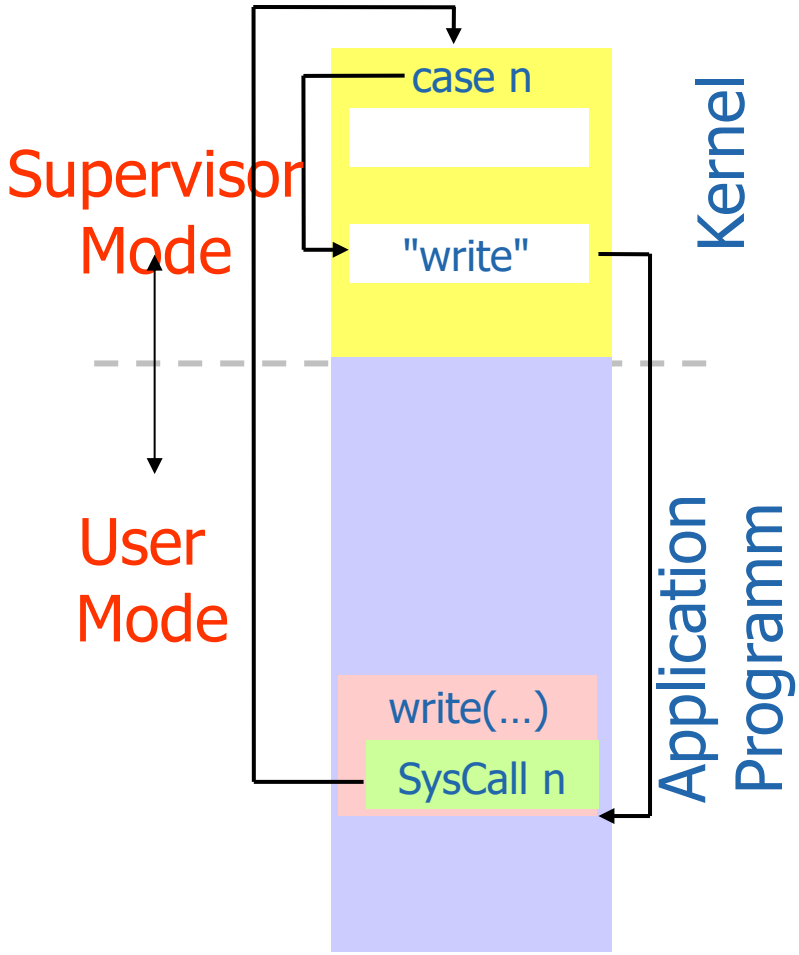
- MMU is a protected device
  - Microprocessor supports OS by implementing a Supervisor mode/bit
  - Processor asserts S/U bits and Code/Data bits to let MMU know what kind of access is being carried out



Computer System Issues

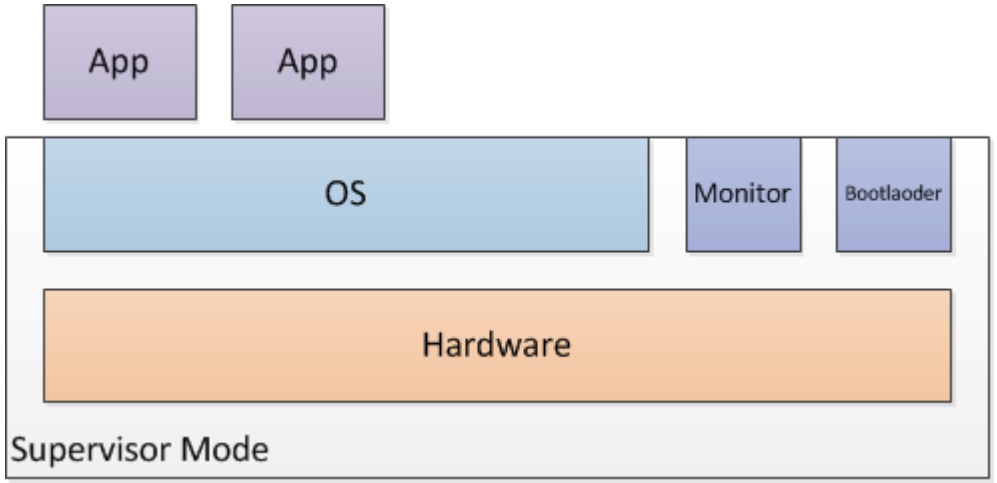
- Supervisor Mode
  - uP boots in supervisor mode
  - Carries out necessary code
  - Code will switch into User mode
  - Only Interrupts executed in Supervisor Mode

  - OS usually executed in Supervisor mode to control tasks / restrict rights of tasks

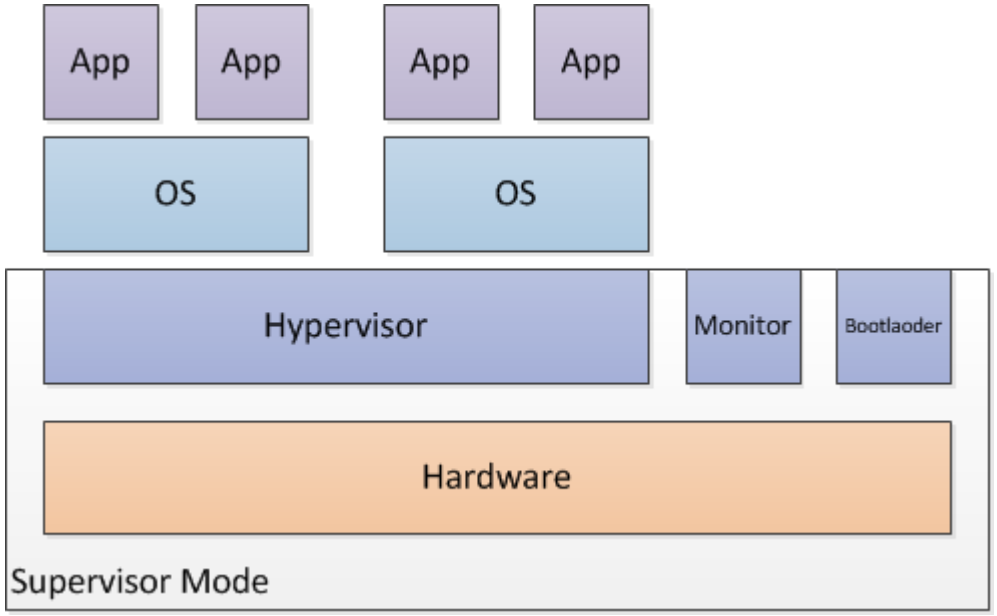


Computer System Issues

- Problem:
  - OS -> rich operating systems
  - Impossible/out of scope to «securitise»

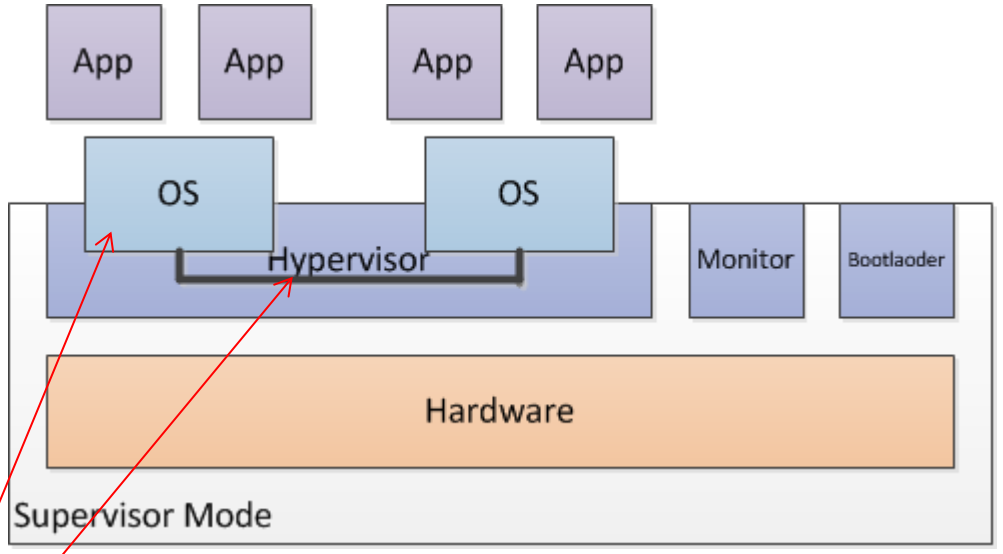


- Idea!
  - Use a hypervisor as a security kernel



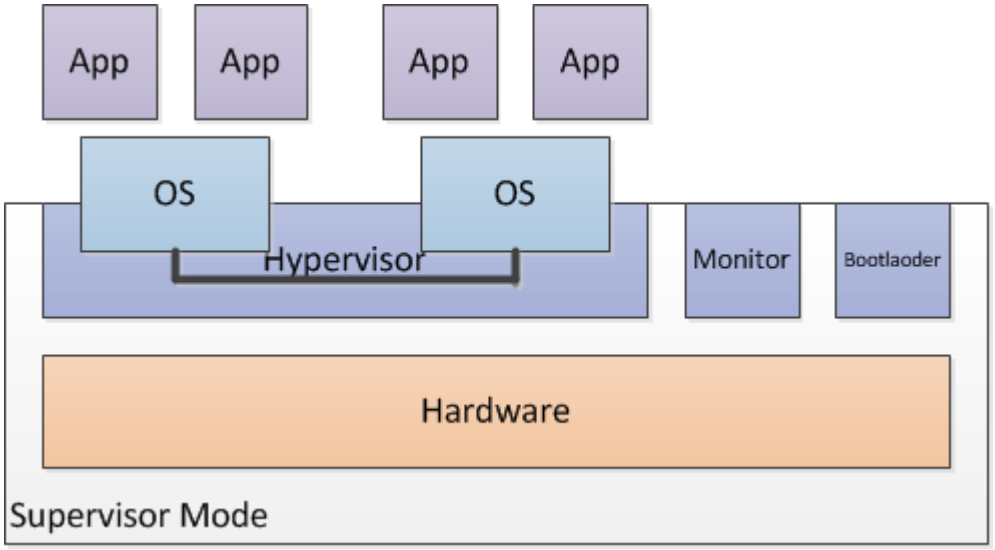
Embedded Hypervisor

- Hypervisor:
  - Known from large computer systems hosting multiple Virtual Machines (VM)
  - Compact design
  - Two types
    - Type 1: Runs exclusively in supervisor mode
    - Type 2: Hosted by OS as an application
  - Embedded hypervisors
    - Type 1
    - Microkernel architecture
    - Generally Implements separation kernel



Embedded Hypervisor

- Hypervisor:
  - Ignores security issues from debug and test infrastructure
- Microkernel
  - Kernel handles scheduling and inter-process communication
- Separation Kernel
  - Behaves as if the applications (OS) are on separate CPU's
  - End up with a concept like ->



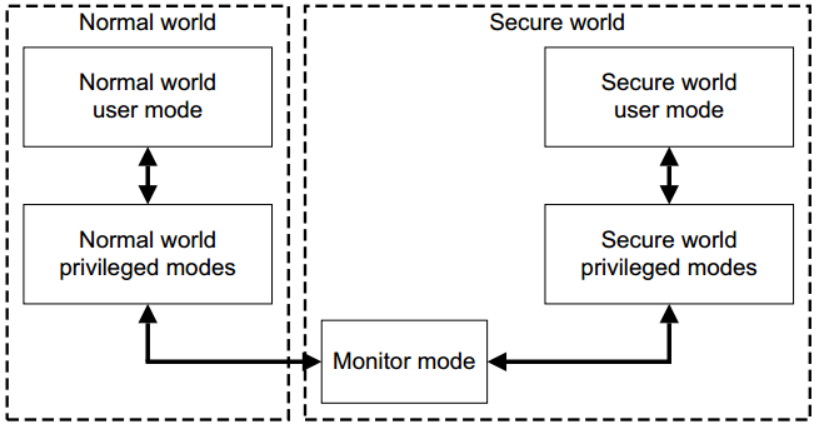


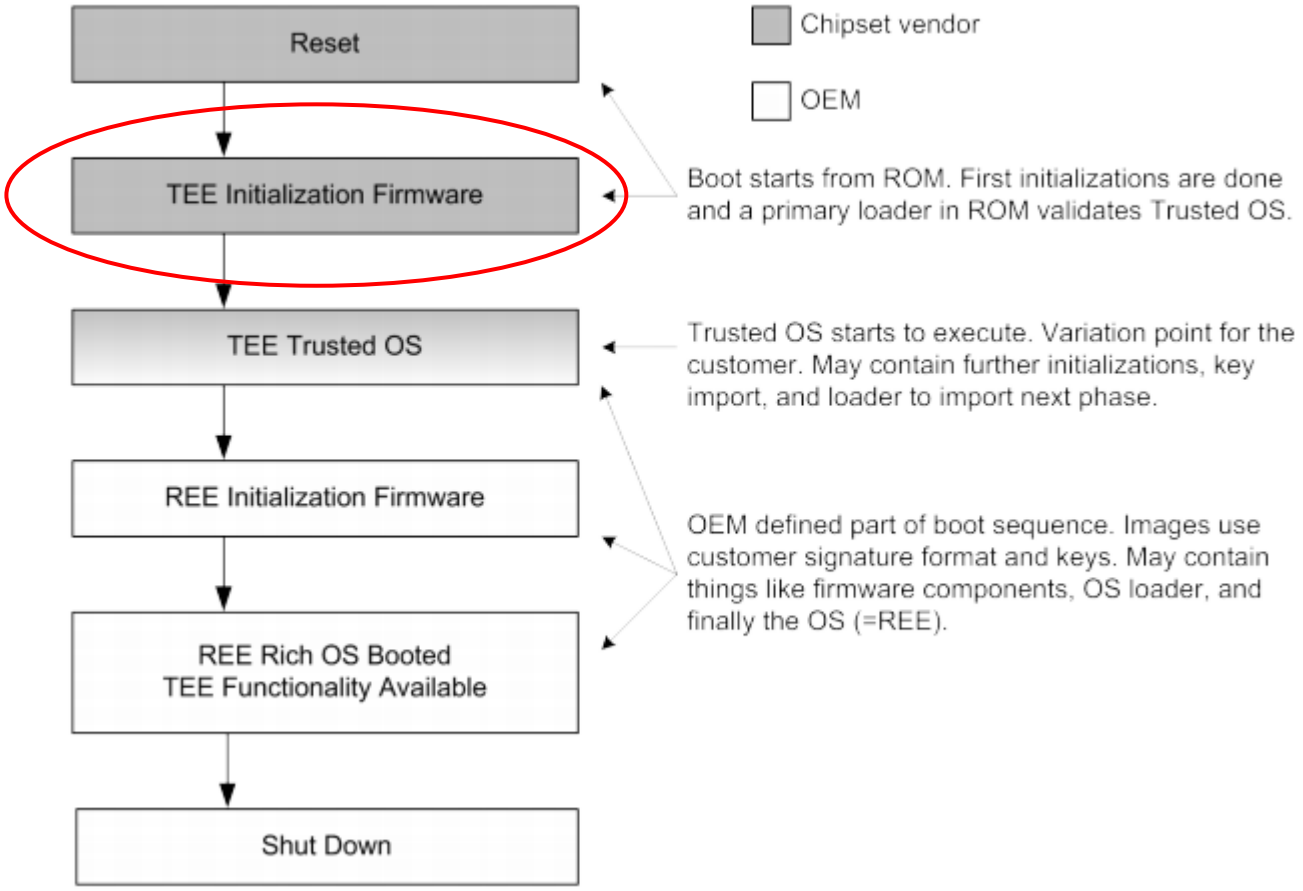
Figure 3-1 : Modes in an ARM core implementing the Security Extensions

http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf

Embedded Hypervisor

- Boot is critical
- Only with trusted boot can chain-of-trust be established

Figure 4-1: Simplified Platform Boot Sequence



Defined by GlobalPlatform – standards organisation of secure deveics GPD_TEE_SystemArch_v1.0.pdf

Secure Boot

- Boot sequence:
  - ROM bootloader ->
  - Transfers control to bootloader located in RAM
  - => need to make sure the right bootloader is being run

- Use for instance public-key signature scheme
  - Device contains public key (PuK)
  - Code includes private key
  - Code can be authenticated

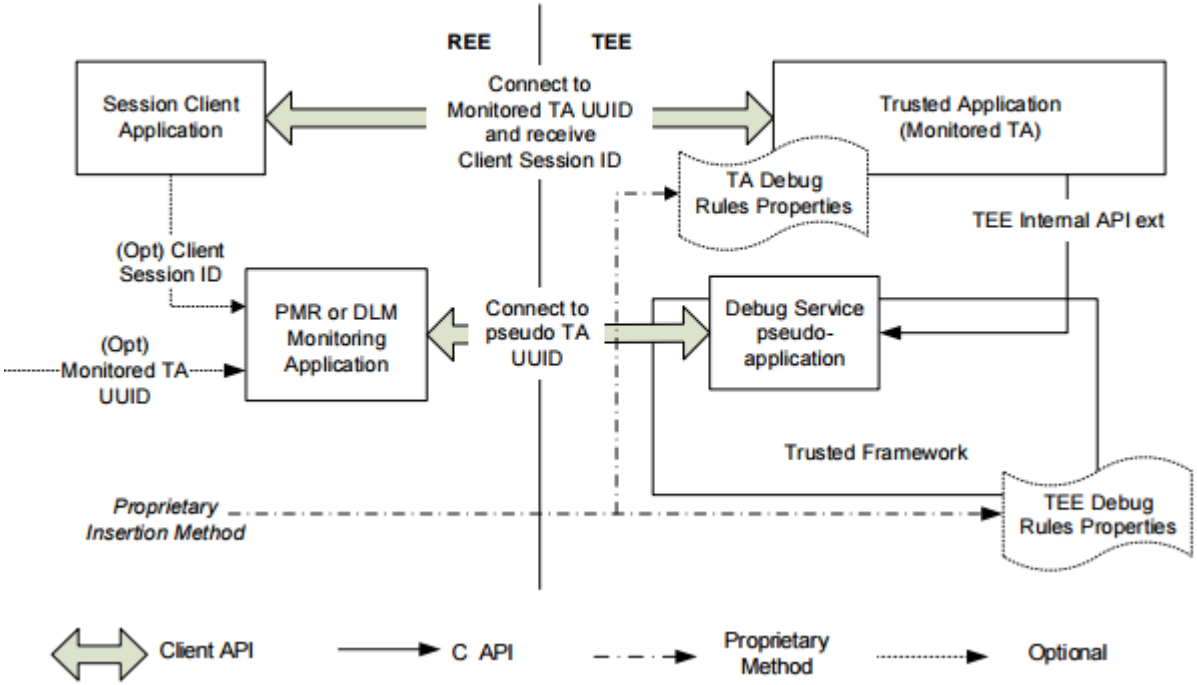
- PuK Storage
  - Secure boot implements chain-of-trust
    - One level authenticates the next layer
    - Makes sense to keep several different PuK's for each layer
  - PuK Must be secure so that attacker cannot swap
  - On-chip OTP-ROM generally smaller than 1024 bits
  - Store PuK off-chip but cryptographic hash on board (256 bits)
  - PuK needs to be stored in a safe location during (hash) verification

  - Chain of trust establishes the secure world
  - Secure world can monitor the normal world for signs of virus attack.

Debug -

- Secure-debuging also an issue
  - Global Platform offer an interface specification for debugging

Figure 2-1: General Case PMR and DLM Architecture Diagram



GPD_TEE_TA_Debug_Spec_v1.0.1_20160614.pdf

Application Development (1)

- With Secure and Normal World
  - Need to partition the application according to security concerns

- Example 1 Maxim DS5250
  - 25MHz. 8051 with encrypted RAM/ROM access
  - Self-destruct on tamper detection
  - For instance PIN key-pads

- Note
  - IP security
  - Application Security

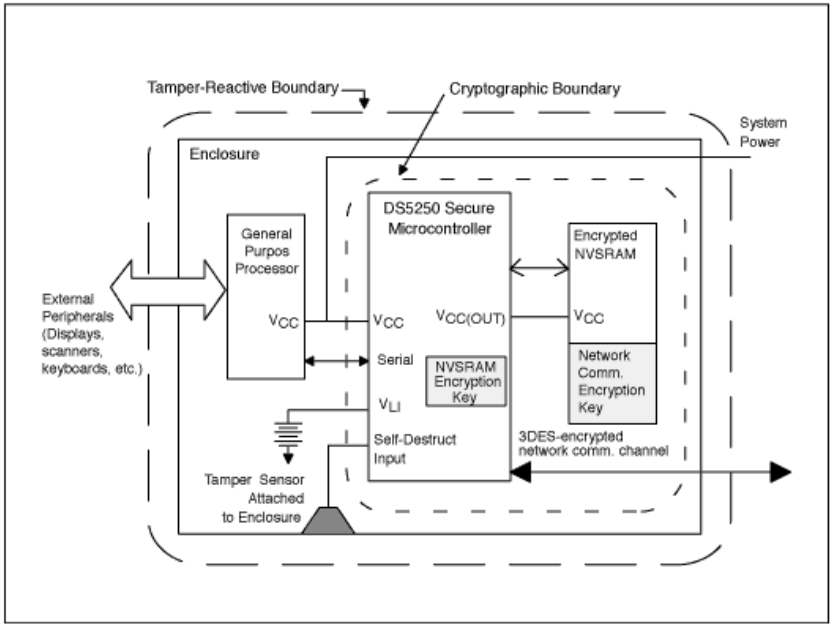


Figure 2. Secure embedded system using the DS5250.

Application Development (2)

- Example 2 ARM Gadget2008
  - The Gadget2008 product aims to provide a portable cellular handset, with a feature-rich operating system capable executing user downloaded applications.
  - To support the current trend for portable media, the handset will support audio and video playback. License deals with major content providers require the addition of a DRM content protection scheme in the software environment.
  - To provide additional revenues a companion content provision service, GadgetStore, will be developed. GadgetStore will allow the user to pay for downloaded content using standard banking facilities. The design should aim to reduce card transaction cost to the content provision service and minimise risk of the consumer being exposed to fraud.

Application Development (2)

- Example 2 ARM Gadget2008

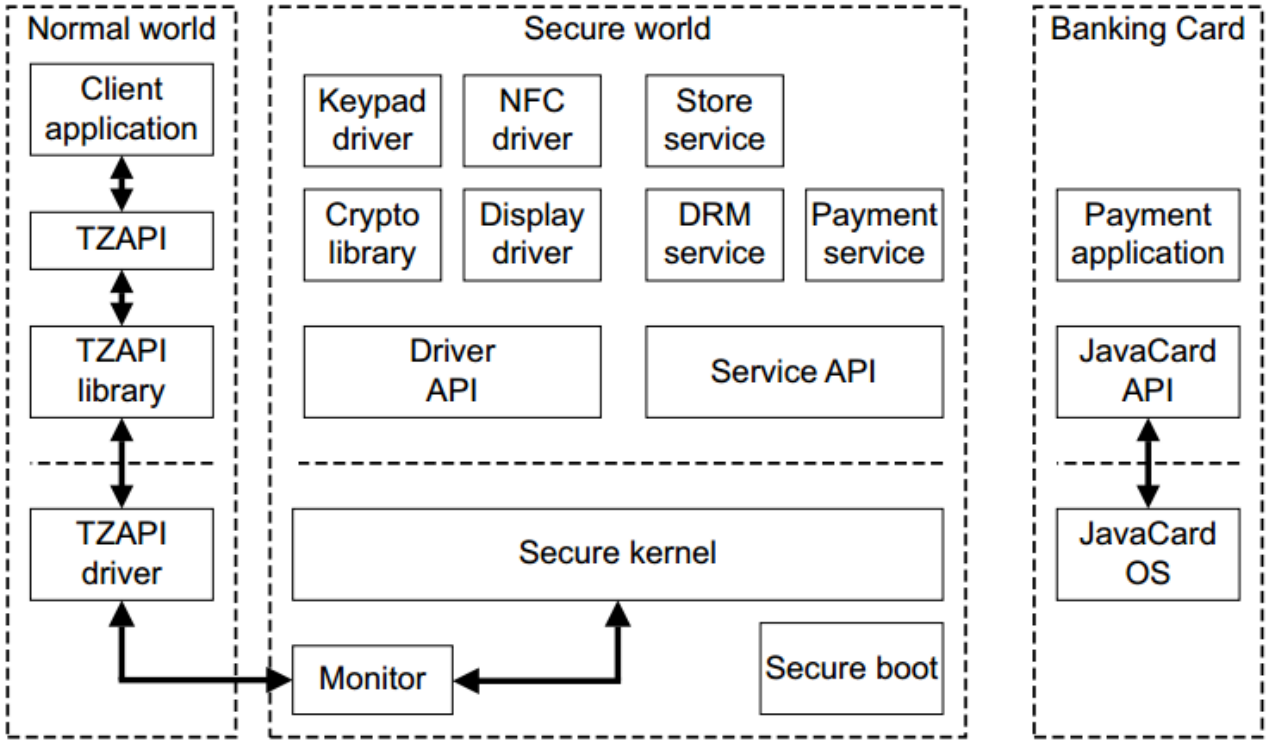


Figure 6-2 : The Gadget2008 software architecture

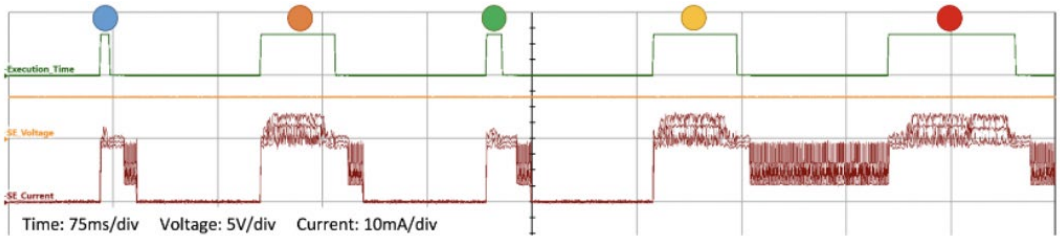
Secure Elements

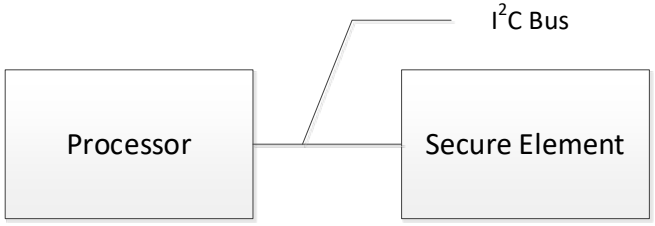
Zurich University of Applied Sciences
School of Engineering
InES Institute of Embedded Systems

Secure Elements are hardware components that perform security related tasks and store security related information

Basic operations supported (depending on device)

- Generate random pair
- Calculate an elliptic key pair
- Calculate a SHA256 hash
- Calculate a signature
- Verify the signature





	Operation	Ex. Time [ms]	Avg. Current [mA]	Energy [nWh]
●	Gen. Random Nr.	7.0	9.26	60
●	Gen. ECC Key Pair	53.8	11.17	552
●	SHA256	11.6	9.45	101
●	ECDSA Sign	60.6	11.27	627
●	ECDSA Verify	92.2	11.1	939

Zurich University
of Applied Sciences
School of
Engineering
InES Institute of
Embedded Systems

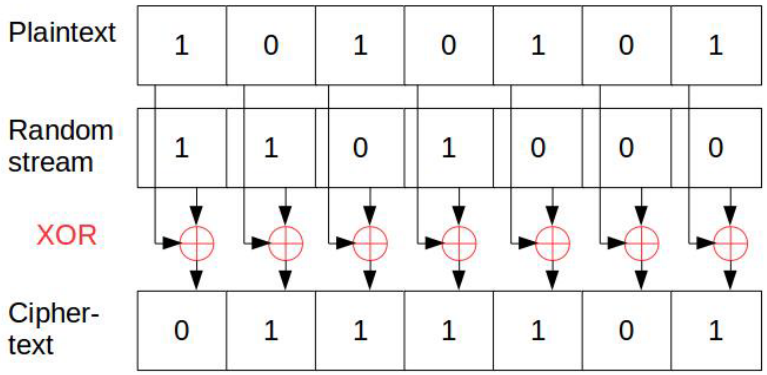
Some Algorithms

Symmetric Cryptography

«Algorithms that use the same secret key for encryption of plaintext and decryption of ciphertext»

Application:

- Block cipher : fixed block size as input, encrypted block of same length as output
- Stream cipher: plaintext digits combined with a pseudorandom cipher stream



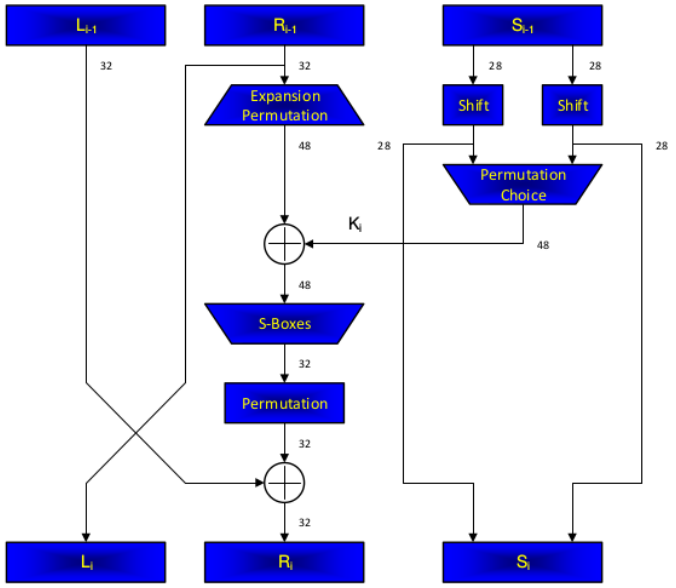
Problems:

- Shared key must be stored and kept secret
- Each pair of users needs another secret key

Basic Principle: Security through diffusion and confusion

Symmetric Cryptography - Examples

DES – Data Encryption Standard



Parameters:
- Block size: 64 bit
- Key Size: 56 bit (+ 8 parity bits)
- Number of rounds: 16

Algorithm:
- Initial permutation of 64 bit block
- Divide into 2 parts of 32 bit (**L**eft and **R**ight)
- Expansion permutation of right part (32bit → 48bit)
- Right part through S-Boxes (48bit → 32bit) (nonlinear substitution)
- Another permutation of right part
- Swap left and right part
- Start next round with expansion permutation

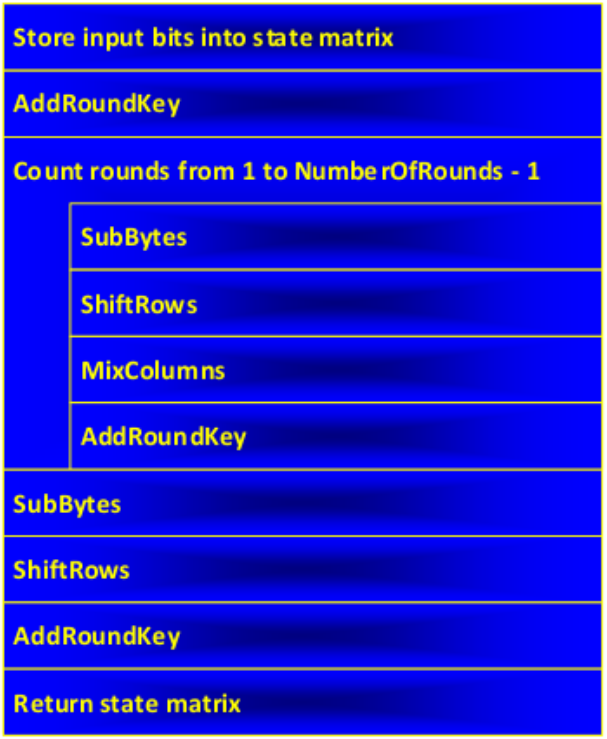
Security:
- Key length of 56 bits is too small!
- Solution: use Triple DES (3DES) → 3 times DES with 3 different keys

Symmetric Cryptography - Examples

AES – Advanced Encryption Standard

Parameters:
- Block size: 128 bit
- Key Size: 128/192/256 bit
- Number of rounds: 10/12/14 depending on key length

Algorithm:
- AddRoundKey: bitwise XOR between blocks and key
- SubBytes: Nonlinear byte substitution
- ShiftRows: shift rows a certain times left
- MixColumns: byte rows mixed between blocks



Security:
- Successor of DES – faster than 3DES with less key length
- All key length and associated number of rounds bring sufficient security
- For top secret information, 192 or 256 bit keys are recommended.

Asymmetric (Public Key) Cryptography

«Systems that uses a key pair consisting of a public key – known to everyone – and a private key for each user kept as secret and never needed to transmit»

Application:

- Public Key Encryption: Message encrypted with public key can only be decrypted with the matching private key → Confidentiality
- Digital Signatures: Message signed by sender with private key, receiver can verify that it is from a trustworthy person and that message has not been changed during transmission → Authentication / Non–Repudiation / Integrity

Problems:

- Public key cryptographic infrastructures typically needs much computational effort

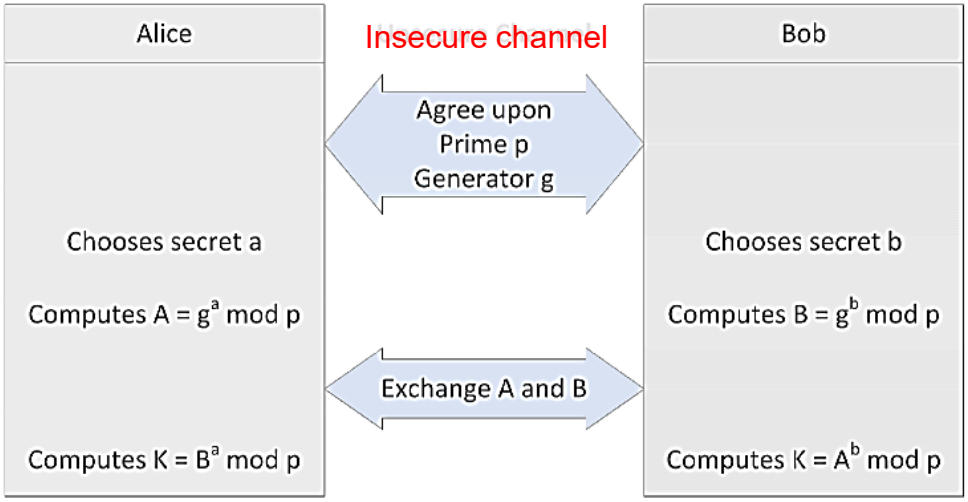
Basic Principle: One-way function that can easily be inverted when secret known but without knowledge no efficient algorithm to generate plaintext from ciphertext

Asymmetric (Public Key) Cryptography - Examples

Zurich University of Applied Sciences
School of Engineering
InES Institute of Embedded Systems

Diffie – Helman Key Exchange (DHKE)

Goal:
Two parties Alice and Bob want to exchange a key (e.g. for later usage in symmetric crypto scheme) over an insecure channel



$K = B^a \bmod p = (g^b)^a \bmod p = A^b \bmod p$

Security:

- no algorithm known that can solve the discrete logarithm
- Man-In-The-Middle attacks possible: Person could intercept insecure channel and perform an exchange with both Alice and Bob without being detected.

Asymmetric (Public Key) Cryptography - Examples

RSA (Rivest, Shamir, Adleman)

Goal:
Bob encrypts a message and sends it to Alice, only Alice is able to decrypt the message

Algorithm:
Key Generation:

- Alice chooses randomly to (large) prime numbers **p** and **q**
- Alice computes **n=p•q**
- Alice chooses integer **e** such that:
  - **1 < e < (q-1)•(p-1)**
  - **gcd(e, (q-1) •(p-1)) = 1**
- Alice calculates **d**, the modular inverse of **e**:
  - **e•d ≡ 1 mod (q-1)•(p-1)**
- Alice sends public components **(e, n)** to Bob and keeps **d** secret

Encryption / Decryption:

- Bob can compute cipher text out of message m: **$c = m^e \bmod n$**
- Alice can decrypt message: **$m' = c^d \bmod n$** where **$m' = m$**

Security:

- Security of RSA based on the fact that it is very hard to find the prime factors of a very large number consisting of large primes