Inf3510 - Summary

Week 1

Information Security Basic Concepts

Security is about protecting assets from damage or harm and it focuses on all types of assets.

- Security → Sikkerhet
- Safety → Trygghet
- Certainty → Visshet

Information Security focuses on protecting *information assets* from damage or harm. Assets like data files, software, IT equipment and infrastructure.

Information Security defined: The preservation of confidentiality, integrity and availability of information; in addition, other properties such as authenticity, accountability, non-repudiation and reliability can also be involved. (ISO27001)

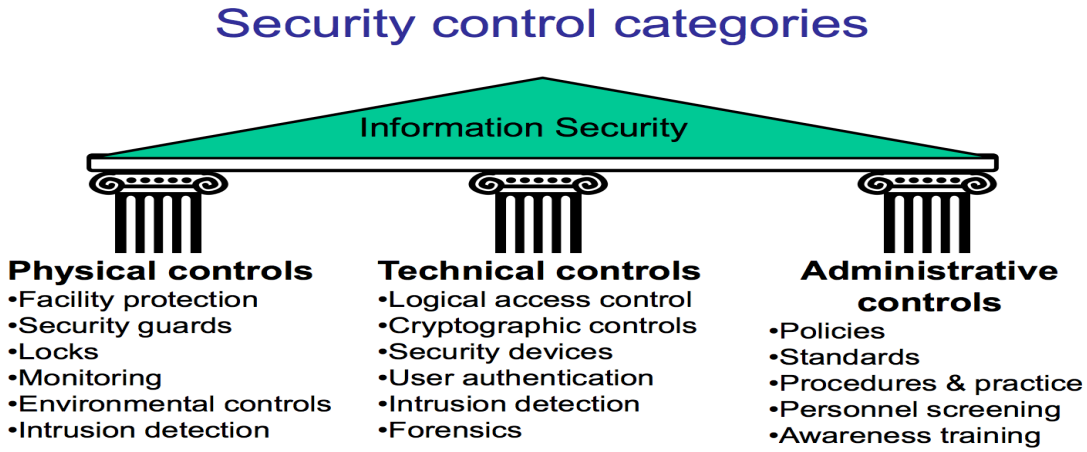
Information security management has as **goal** to avoid damage and to control risk of damage to information assets. IS management focuses on:

– Understanding threats and vulnerabilities
– Managing threats by reducing vulnerabilities or threat exposures
– Detection of attacks and recovery from attacks
– Investigate and collect evidence about incidents (forensics)

Reasons why we can't solve all security problems once and for all:

– Rapid innovation constantly generates new technology with new vulnerabilities
– More activities go online
– Crime follows the money
– Information security is a second thought when developing IT
– New and changing threats
– More effective and efficient attack technique and tools are being developed

Information security doesn't have a final goal, it's a continuing process.



Preventive controls: prevent attempts to exploit vulnerabilities

• Example: encryption of files.

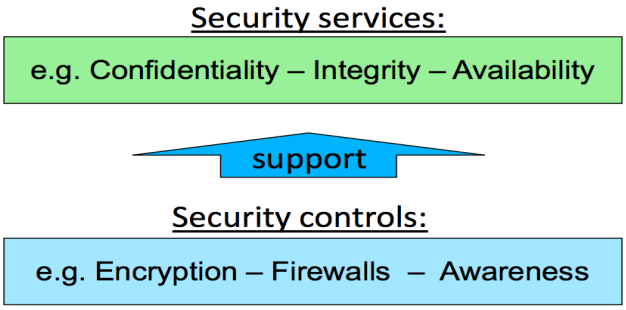
Detective controls: warn of attempts to exploit vulnerabilities

• Example: Intrusion detection systems (IDS).

Corrective controls: correct errors or irregularities that have been detected.

• Example: Restoring all applications from the last known good image to bring a corrupted system back online.

Confidentiality, integrity, availability are the **three main security properties** (CIA properties).



Security services (goals or properties) are supported by specific security controls (mechanisms).

Confidentiality: The property that information is not made available or disclosed to unauthorized individuals, entities, or processes. (ISO 27001)

• Can be divided into:

– **Secrecy**: Protecting business data

– **Privacy**: Protecting personal data

– **Anonymity**: Hide who is engaging in what actions

• *Main threat*: Information theft, unintentional disclosure

• *Controls*: Encryption, Access Control, Perimeter defence

Integrity: **Data Integrity**: The property that data has not been altered or destroyed in an unauthorized manner. (X.800) **System Integrity**: The property of safeguarding the accuracy and completeness of assets (ISO 27001)

• *Main threat*: Data and system corruption

• *Controls*:

– Cryptographic integrity check,

– Encryption,

– Access Control

– Perimeter defence

– Audit

– Verification of systems and applications

Availability: The property of being accessible and usable upon demand by an authorized entity. (ISO 27001)

• *Main threat*: Denial of Service (DoS)

– The prevention of authorized access to resources or the delaying of time critical operations

• *Controls*: Redundancy of resources, traffic filtering, incident recovery, international collaboration and policing

Authenticity is another type of security service that can be divded in various types:

- **User authentication**: The process of verifying a claimed identity of a (legal) user when accessing a system or an application.
    - *Identification*: who you claim to be, method: (user)name, biometrics
    - *User authentication*: prove that you are the one you claim to be
    - *Main threats*: Unauthorized accsess
    - *Controls*: passwords, personal cryptographic tokens(OTP generators, bankbrikke etc), biometrics(id cards), cryptographic security/authentication protocols.
- **Organisation authentication**: The process of verifying a claimed identity of a (legal) organisation in an online interaction/session.
- **System authentication** (peer entity authentication): The corroboration (verification) that a peer entity (system) in an association (connection, session) is the one claimed (X.800).
    - *Goal*: establish the correct identity of remote hosts.
    - *Main threat:* network intrusion, masquerading attacks, replay attacks, (D)DOS(distributed denial-of-service) attacks.
    - *Controls:* Cryptographic authentication protocols based on hashing and encryption algorithms, Examples: TLS, VPN, IPSEC.
- **Data origin authentication** (message authentication): The corroboration (verification) that the source of data received is as claimed (X.800).
    - *Goal*: Recipient of a message (i.e. data) can verify the correctness of claimed sender identity(But 3rd party may not be able to verify it).
    - *Main threat*: False transactions, False messages and data
    - *Controls*: Encryption with shared secret key, MAC (Message Authentication Code), Security protocols, Digital signature with private key, Electronic signature

Non-Repudiation(Security Service)

- *Goal*: Making sending and receiving messages undeniable through unforgible evidence.
    - Non-repudiation of origin: proof that data was sent.
    - Non-repudiation of delivery: proof that data was received.
    - **NB**: imprecise interpretation: Has a message been received and read just because it has been delivered to your mailbox?
- *Main threats*: Sender falsely denying having sent message, Recipient falsely denying having received message.
- *Control*: digital signature (Cryptographic evidence that can be confirmed by a third party).

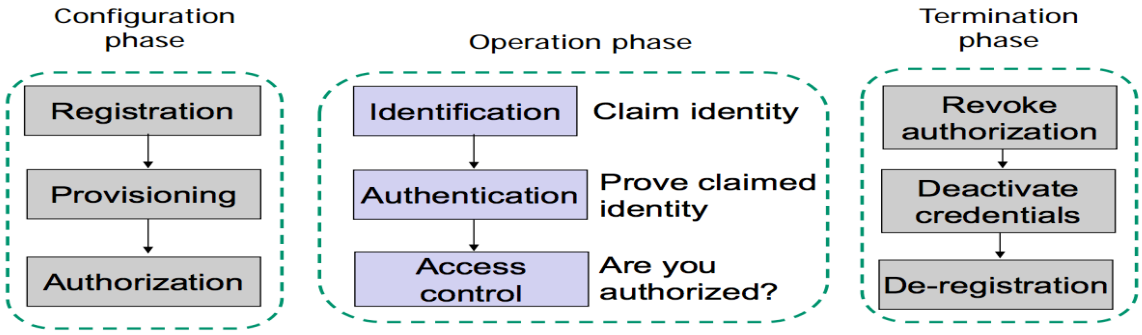
Data origin authentication and non-repudiation are similar. Data origin authentication only provides proof to recipient party. Non-repudiation also provides proof to third parties.

Accountability(security service)

- *Goal*: Trace action to a specific user and hold them responsible
  - ”Audit information must be selectively kept and protected so that actions affecting security can be traced to the responsible party”
- *Main threats*: Inability to identify source of incident, Inability to make attacker responsible
- *Controls*: Identify and authenticate users, Log all system events (audit), Electronic signature, Non-repudiation based on digital signature, Forensics

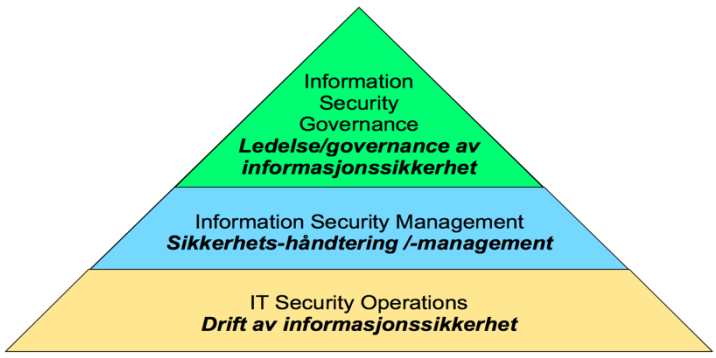
Authorization is to specify access and usage permissions for entities, roles or processes.
– Authorization policy normally defined by humans
– Issued by an authority within the domain/organisation
– Authority can be delegated, example: Management → Sys.Admin.
– Implemented in IT systems as configuration/policy
"A user may be authorized to access the files on the file server, but until she is properly identified and authenticated, those resources are out of reach."

Identity and Access Management (IAM) Phases:



Week 2

IT Security Management concepts



Defining Information Security Governance: IS governance provides strategic direction, ensures objectives are achieved, manages risk appropriately, uses organizational resources responsibly, and monitors the success or failure of the enterprise security programme.

COBIT = Control Objectives for Information and Related Technology
COBIT is a framework for IT management & governance. It is a set of controls and processes for bridging the gap between business risks and IT control requirements. COBIT defines key IT process activities together with their input and output, IT process objectives, performance measures and an elementary maturity model. COBIT also describes security management

processes. COBIT is published and maintained by **ISACA, the Information Systems Audit and Control Association**.
Goals of information security governance as defined in COBIT by ISACA:

1. Strategisk tilpasning av sikkerhetsprogrammet (Strategic alignment of security program)
– IS-aktiviteter skal støtte organisasjonens helhetlige strategi.

2. Risikohåndtering (Risk management)
– Gjøre nødvendige undersøkelser for å avdekke trusler, sårbarheter og risiko som organisasjonen står overfor, og bruke adekvate virkemidler for å redusere risiko til et akseptabelt nivå.

3. Verdiskapning (Value delivery)
– Søk optimal balanse mellom reduksjon av risiko og tap, og kostnader forbundet med sikkerhetsvirkemidler.

4. Ressursbruk (Resource management)
– Arbeidet med informasjonssikkerhet skal gjøres effektivt

5. Målbarhet (Performance measurement)
– Effekten av sikkerhetsarbeidet skal måles

6. Integrering av sikkerhetsområder (Assurance process integration)
– Separate områder relatert til sikkerhet (fysisk, finansiell, IT osv.) skal i størst mulig grad integreres

What is information security management?
Includes:
- Risk management,
- Security policies (creation and maintenance)
  - Documented goals, rules and practice for IS
- Plan and organisation for managing the security activities
  - Information Security Management System (ISMS)
- Information classification
- Definition of security procedures, standards & guidelines
- Deployment and maintenance of security controls
- Security education and training
- Disaster recovery and business continuity planning

Who is responsible for ISM?
– Management
- CEO, CSO, CIO
- Allocate resources, endorse and abide security policies

– IT Security staff
– General security staff, i.e. guards, janitors etc.
- Important for physical security

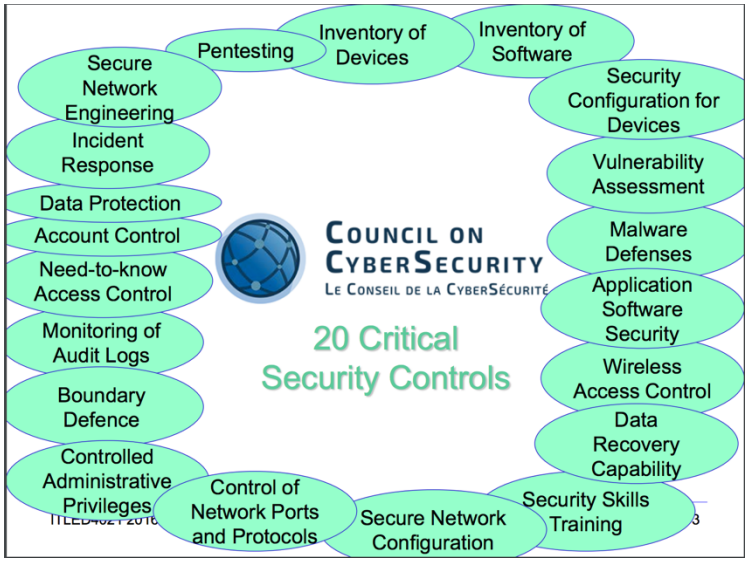
– IT staff
– Users
– Third parties
- Outsourced information security management
- Customers, suppliers, business partners

IS Management Standards

There are many different types of standards, but the most common or mostly used are: ISO/IEC 27K (**Must be bought)**, NIST (National Institute for Standards and Technology) **(Free->)**, COBIT, 20 CSC (Critical Security Controls)

ISO: International Standards Organization
IEC: International Electro-technical Committee
ISO/IEC is correct, but people mostly refer to the standards as ISO
ISO 27001: Information Security Management System (ISMS)
ISO 27002: Code of practice for information security management

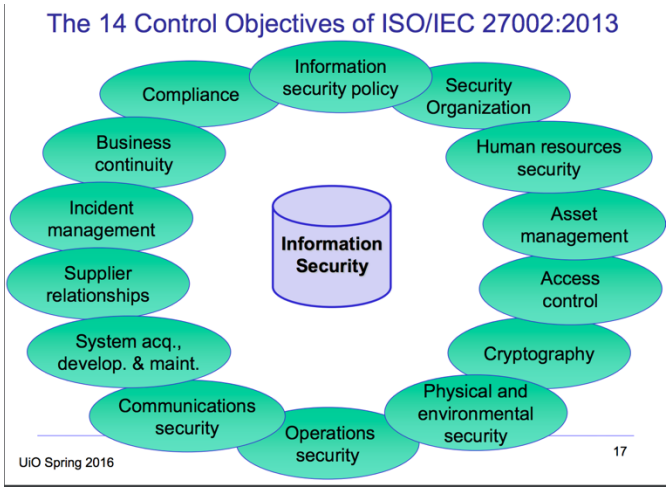
-100: Information Security Handbook: A Guide for Managers
-53: Recommended Security Controls for Federal Info Systems
-35: Guide to Information Technology Security Services
-39: Managing Information Security Risk
-30: Guide for Conducting Risk Assessment
-27: Engineering Principles for Information Technology Security
-18: Guide for Developing Security Plans for Federal Info Systems
-14: Generally Accepted Principles and Practices for Securing Information Technology Systems
-12: An Introduction to Computer Security: The NIST Handbook
-26: Security Self-Assessment Guide for Information Technology Systems



ISO 27002 provides a checklist of general security controls to be considered implemented/used in organizations

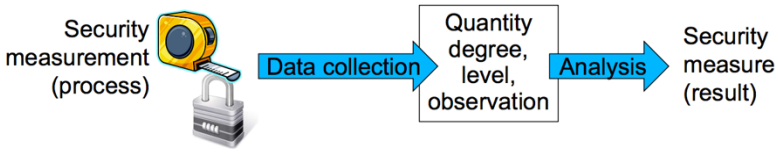
– Contains 14 categories (control objectives) of security controls
– Each category contains a set of security controls
– In total, the standard describes 113 generic security controls

Objective: "… gives guidelines for […] information security management practices including the selection, implementation and management of controls taking into consideration the organization's information security risk environment(s)."

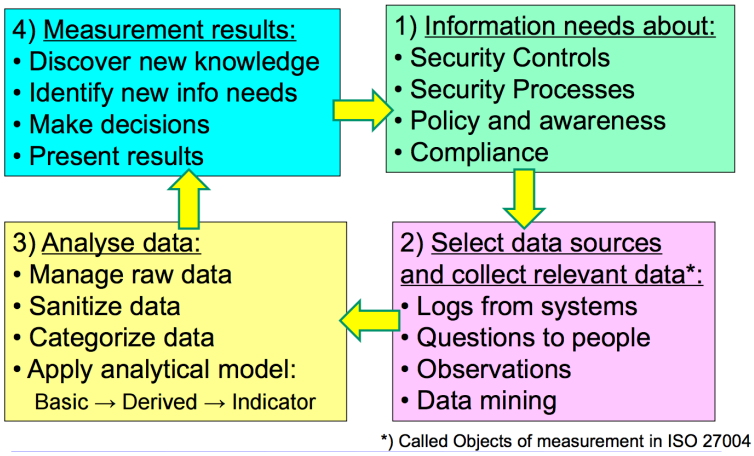


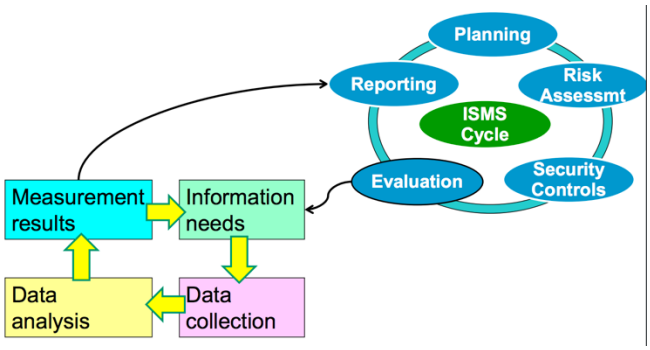
ISO 27001 specifies requirements for establishing, implementing, maintaining and continually improving an information security management system (ISMS) within the context of the organization. While the ISO 27002 (code of practice) defines a set of security goals and controls, ISO 27001 (ISMS) defines how to manage the implementation of security controls. ISO 27001 is to be used in conjunction with ISO 27002.

IS governance cycle as an interpretation of ISMS (ISO 27001). The steps in the cycle can be performed simultaneously. Good IS governance requires that all steps are implemented in the organization



IS Measurement Model (ISO 27004)





Process Capability Level

1. **Performed Ad Hoc**
   + Processes are ad-hoc and disorganized.
   + Risks are considered on an ad hoc basis, but no formal processes exist.
2. **Managed but intuitive**
   + Processes follow a regular pattern.
   + Emerging understanding of risk and the need for security
3. **Established process**
   + Processes are documented and communicated.
   + Company-wide risk management.'
   + Awareness of security and security policy
4. **Managed and Predictable**
   + Processes are monitored and measured.
   + Risks assessment standard procedures
   + Roles and responsibilities are assigned
   + Policies and standards are in place
5. **Optimized**
   + Security culture permeates organization
   + Organization-wide security processes are implemented, monitored and followed

Social Engineering Attacks

"The biggest threat to the security of a company is not a computer virus, an unpatched hole in a program, or a badly installed firewall. In fact the biggest threat could be you."

"What I found personally to be true was that it's easier to manipulate people rather than technology. Most of the time, organisations overlook that human element" - Kevin Mitnick

Social Engineering Tactics:

Develop Trust:

– People are naturally helpful and trusting
– Ask during seemingly innocent conversations
– Slowly ask for increasingly important information
– Learn company lingo, names of key personnel, names of servers and applications
– Cause a problem and subsequently offer your help to fix it (aka. reverse social engineering)
– Talk negatively about common enemy
– Talk positively about common hero

Induce strong affect:

– Heightened emotional state makes victim
  • Less alert
  • Less likely to analyse deceptive arguments
– Triggered by attacker by creating
  • Excitement ("you have won a price")
  • Fear ("you will lose your job")
  • Confusion (contradictory statements)

Information overload:

• Reduced the target's ability to scrutinize arguments proposed by the attacker
• Triggered by
  – Providing large amounts of information to produce sensory overload
  – Providing arguments from an unexpected angle, which forces the victim to analyse the situation from new perspective, which requires additional mental processing

Reciprocation:

• Exploits our tendency to return a favour
  – Even if the first favour was not requested
  – Even if the return favour is more valuable
• Double disagreement
  – If the attacker creates a double disagreement, and gives in on one, the victim will have a tendency to give in on the other
• Expectation
  – If the victim is requested to give the first favour, he will believe that the attacker becomes a future ally

Diffusion of responsibility and moral duty:

• Make the target feel the he or she will not be held responsible for actions
• Make the target feel that satisfying attacker's request is a moral duty

Authority

- People are conditioned to obey authority
  - Milgram and other experiments
  - Considered rude to even challenge the veracity of authority claim
- Triggered by
  - Faking credentials
  - Faking to be a director or superior
  - Skilful acting (con artist)

Commitment creep

- People have a tendency to follow commitments, even when recognising that it might be unwise.
- It's often a matter of showing personal consistency and integrity
- Triggered e.g. by creating a situation where one commitment naturally or logically follows another.
  - First request is harmless
  - Second request causes the damage

Multi-Level Defence against Social Engineering Attacks

Social Engineering Defence

Foundation

- The security policy must address SE attacks
  - Policy is always the foundation of information security
- Address e.g.: Shredding, Escorting, Authority obedience
- Ban practice that is similar to social attack patterns
  - Asking for passwords over phone is a typical SE attack method
  → Therefore never provide passwords over the phone
  - Calling a user and pretending to represent IT department is a typical SE attack
  → Therefore never call user, or make it possible/mandatory for user to authenticate the IT Department
  - Calling IT dep. and pretending to be user is a typical SE attack
  → Therefore make it possible/mandatory for IT department to authenticate the user

Awareness

- Security awareness training for all staff
  - Understanding SE tactics
  - Learn to recognise SE attacks
  - Know when to say "no"
  - Know what is sensitive

– Understand their responsibility
– Understand the danger of casual conversation
– Friends are not always friends
– Passwords are personal
– Uniforms are cheap

• Awareness of policy shall make personnel feel that the only choice is to resist SE attempts

Fortress

• Resistance training for key personnel
– Consider: Reception, Help desk, Sys.Admin., Customer service,
• Fortress training techniques
– Inoculation
• Expose to SE arguments, and learn counterarguments
– Forewarming
• of content and intent
– Reality check:
• Realising own vulnerability,

Persistence

• Ongoing reminders
– SE resistance will quickly diminish after a training session
– Repeated training
– Reminding staff of SE dangers
• Posters
• Messages
• Tests

Gotcha

• Social Engineering Detectors
– Filters and traps designed to expose SE attackers
• Consider:
– The justified Know-it-all
• Person who knows everybody
– Centralised log of suspicious events
• Can help discover SE patterns
– Call backs mandatory by policy
– Key questions, e.g. personal details
– “Please hold” mandatory by policy
• Time to think and log event
– Deception
• Bogus question
• Login + password of “alarm account” on yellow sticker

Offensive

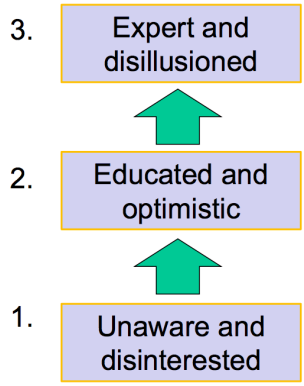
• Incident response
– Well defined process for reporting and reacting to
• Possible SE attack events,
• Cases of successful SE attacks
• Reaction should be vigilant and aggressive
– Go after SE attacker
– Proactively warn other potential victims

Stages of security learning Revealing a deeper problem

• This is far more complex than I first thought. I actually don't think this can ever be made secure.

• I understand it now, it's simple, and I know how to operate it

• I don't understand it, and I don't want to know about it. Why can't security simply be transparent?

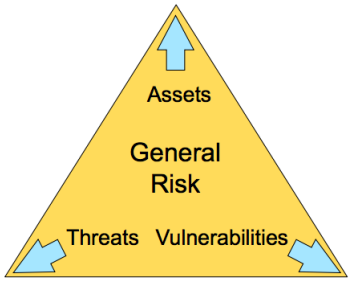


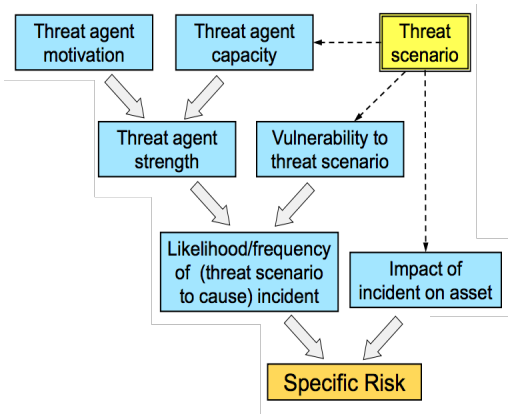
Week 3

Risk Management
Business Continuity Management

What is risk?
• ISO31000 Risk Management:
– **"Risk is the effect of uncertainty on objectives"**
– Also says: **"Risk is often expressed in terms of a combination of the consequences of an event (including changes in circumstances) and the associated likelihood of occurrence."**
• Harris, CISSP 6th ed.:
– **"Risk is the likelihood of a threat agent taking advantage of a vulnerability and the resulting business impact."**
• ISO 27005 (Information Security Risk)
– "Risk is the potential that a given threat will exploit vulnerabilities of assets and thereby cause harm to the organization."

Abstract Risk Model (NSM)
Models general risk in an abstract way
– The more assets you have, the more threats there are, and the more vulnerable you are, then the greater the risk.

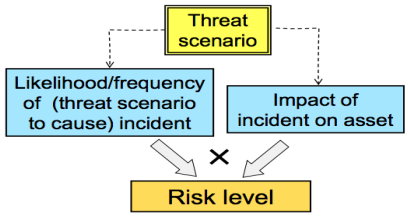




Specific risk model
• Each specific risk relates to a specific threat scenario that can affect specific assets.
• Motivation, capacity, vulnerability and impact determine the risk level for that specific risk

Many Risks
• Multiple different threats (threat scenarios) can be identified
• Each threat can potentially cause an incident
• Each potential incident has a risk level
• Multiple threats ⇒ Many risks

Practical risk model
• Practical risk analysis typically considers two factors to determine the level of each risk
1. Likelihood / frequency of each type of incident
2. Impact on assets (loss) resulting from each type of incident



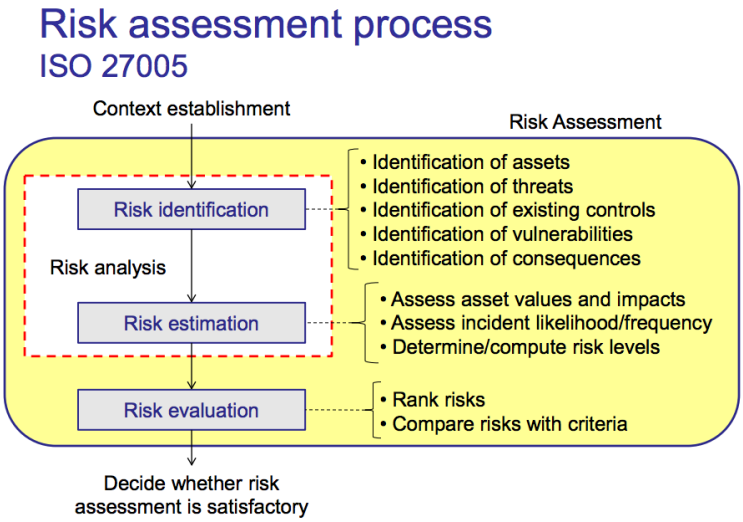
What is risk management?

• "IS risk management analyses what can happen and what the possible consequences can be, before deciding what should be done and when, to reduce risk to an acceptable level."

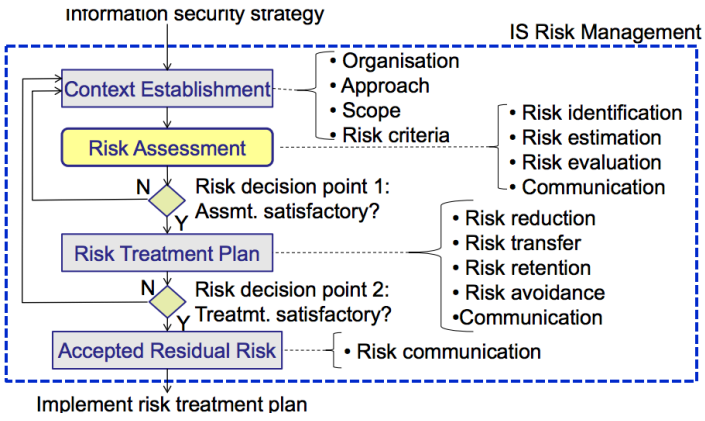
– ISO 27005

• "Risk management consists of coordinated activities to direct and control an organization with regard to risk."

– ISO31000 , ISO/IEC 27002



Risk management process



Basis for assessing risk

• Know the assets: identify, examine, and understand the information and systems currently in place

• Know the enemy: identify, examine, and understand threats facing the organization

• Know the losses your organisation can tolerate.

• Know responsibility of each stakeholders within an organization to manage risks that are encountered

Roles involved in risk management

• Management, users, and information technology must all work together

– Asset owners must participate in developing inventory lists

– Users and experts must assist in identifying threats and vulnerabilities, and in determining likelihoods

– Risk management experts must guide stakeholders through the risk assessment process

– Security experts must assist in selecting controls

– Management must review risk management process and approve controls

Problems of measuring risk

Businesses normally wish to measure risk in money, but almost impossible to do this

– Valuation of assets

• Value of data, hard to assess

• Value of goodwill and customer confidence, very vague

– Likelihood of threats

• Past events not always relevant for future probabilities

– The nature of future attacks is unpredictable

– The actions of future attackers are unpredictable

– Measurement of benefit from security control

• Problems with the difference of two approximate quantities

– Estimation of past and present risk

The Proportionality Principle:

– Apply a set of controls (physical, technical and administrative controls) that match the perceived risk to, and value of, an organisation's information assets.

Asset Valuation and Prioritization

• Questions help develop criteria for asset valuation
• Which information asset:
  – is most critical to organization's success?
  – generates the most revenue/profitability?
  – would be most expensive to replace or protect?
  – would be the embarrassing or cause liability if revealed?
• Prioritization
  – Create weighting for each category
  – Calculate relative importance of each asset
  – List the assets in order of importance using a weighted factor analysis worksheet

Threat scenario identification

• Realistic threat scenarios need to be described; unimportant threats can be ignored
• Threat assessment:
  – Which threats present danger to assets?
  – Which threats represent the most danger to information?
  – How much would it cost to recover from attack?
  – Which threat are most expensive to prevent?

Threat Scenario Modelling

• Attacker-centric
  – Starts from attackers, evaluates their goals, and how they might achieve them through attack tree. Usually starts from entry points or attacker action.
• System-centric (aka. SW-, design-, architecture-centric)
  – Starts from model of system, and attempts to follow model dynamics and logic, looking for types of attacks against each element of the model. This approach is e.g. used for threat modeling in Microsoft's Security Development Lifecycle.
• Asset-centric
  – Starts from assets entrusted to a system, such as a collection of sensitive personal information, and attempts to identify how security breaches of CIA properties can happen.

Vulnerability Identification

• Specific avenues threat agents can exploit to attack an information asset are called vulnerabilities
• Examine how each incident/threat could be perpetrated and list organization's assets and vulnerabilities
• Process works best when people with diverse backgrounds within organization work iteratively in a series of brainstorming sessions
• At end of risk identification process, list of assets and their vulnerabilities is achieved

Identifying specific risks

Threats / incidents	Vulnerabilities	Asset impacts
•Password compromise	•Weak passwords	•Deleted files
•SQL injection	•Poor awareness	•Damaged files
•Logical bomb in SW	•No input validation	•Damaged reputation
•Trojan infects clients	•Outdated antivirus	•Stolen files
•Cryptanalysis of cipher	•Weak ciphers	- sensitivity levels 1,2,3
•Brute force attack	•Short crypto keys	•Intercepted traffic
•Social engineering	•Poor usability	•False transaction
• …..	• …	• …

- A valid combinations of threat, vulnerability and asset impact represents a single specific risk
- All relevant specific risks should be identified

Estimating risk levels

Types of analysis

• *Qualitative*

– Uses descriptive scales. Example:

• Impact level: Minor, moderate, major, catastrophic

• Likelihood: Rare, unlikely, possible, likely, almost certain

Increasing Likelihood

Likelihood	Description
High	Is expected to occur in most conditions (1 or more times per year).
Medium	The event will probably happen in most conditions (every 2 years).
Low	The event should happen at some time (every 5 years).
Unlikely	The event could happen at some time (every 10 years).

Increasing Impact

Impact	Description
Major	**Major problems** would occur and threaten the provision of important processes **resulting in significant financial loss**.
Moderate	**Services would continue**, but would **need to be reviewed or changed.**
Minor	Effectiveness of services would be **threatened but dealt with**.
Insignificant	Dealt with as a part of **routine operations**.

• *Semi-quantitative*

– Qualitative scales assigned numerical values

– Can be used in formulae for prioritization (with caution)

Qualitative risk estimation - example

Qualitative risk levels: Add likelihood & impact level

Impact level

Likelihood \ Risk level	(0) Insignificant	(1) Minor	(2) Moderate	(3) Major
(3) High	(3) M	(4) H	(5) VH	(6) E
(2) Medium	(2) L	(3) M	(4) H	(5) VH
(1) Low	(1) VL	(2) L	(3) M	(4) H
(0) Unlikely	(0) N	(1) VL	(2) L	(3) M

Legend

E: extreme risk; immediate action required

(V)H: (very) high risk; senior management attention needed

M: moderate risk; management responsibility must be specified

(V)L: (very) low risk; manage by routine procedures

N: Negligible risk; To be ignored

Semi-quantitative risk estimation - example

Semi-quantitative risk levels: Multiply likelihood & impact level

Impact level

Risk Level / Likelihood	(0) Nil	(1) Insign.	(2) Minor	(3) Moderate	(4) Major
(4) High	(0) Nil	(4) M	(8) H	(12) VH	(16) E
(3) Medium	(0) Nil	(3) L	(6) M+	(9) H+	(12) VH
(2) Low	(0) Nil	(2) VL	(4) M	(6) M+	(8) H
(1) Unlikely	(0) Nil	(1) Neg	(2) VL	(3) L	(4) M
(0) Never	(0) Nil	(0) Nil	(0) Nil	(0) Nil	(0) Nil

M: moderate; Specify responsibililty
L: low; Manage by routine procedures
VL: very low; Manage by routine
Neg: Negligible; To be ignored
Nil: Nil; No risk exists

E: extreme; Immediate action required
VH: very high; Priority action action
H+: high +; Management attention
H: high; Management attention
M+: moderate +; Specifu responsib

Quantitative

– Use numerical values for both consequence (e.g. $$$) and likelihood (e.g. probability value)

Example quantitative risk analysis

• Risk description
  – Asset: Public image (and trust)
  – Threat: Defacing web site through intrusion
  – Impact: Loss of image
• Parameter estimates
  – AV(public image) = $1,000,000
  – EF(public image affected by defacing) = 0.05
  – SLE = AV × EF = $50,000
  – ARO(defacing) = 2
  – ALE = SLE × ARO = $100,000
• Justifies spending up to $100,000 p.a. on controls

Example quantitative risk analysis **method**

• Quantitative parameters
  – Asset Value (AV)
    • Estimated total value of asset
  – Exposure Factor (EF)
    • Percentage of asset loss caused by threat occurrence
  – Single Loss Expectancy (SLE)
    • SLE = AV × EF
  – Annualized Rate of Occurrence (ARO)
    • Estimated frequency a threat will occur within a year
  – Annualized Loss Expectancy (ALE)
    • ALE = SLE × ARO

Evaluate risks

- Compare
  - The level of risk found during risk analysis
  - The established risk criteria
  - NOTE: Consider analysis and criteria on same basis (qualitative/quantitative)

Risk listing and ranking

Threat scenario:	Existing controls & vulnerabilities:	Asset impact:	Impact level:	Likelihood description:	Likelihood:	Risk level:
Compromise of user password	No control or enforcement of password strength	Deleted files, breach of confidentiality and integrity	MODERATE	Will happen to 1 of 50 users every year	MEDIUM	HIGH
Virus infection on clients	Virus filter disabled on many clients	Compromise of clients	MODERATE	Will happen to 1 in 100 clients every year	HIGH	EXTREME
Web server hacking and defacing	IDS, firewall, daily patching, but zero day exploits exist	Reputation	MINOR	Could happen once every year	MEDIUM	MODERATE
Logical bomb planted by insider	No review of source code that goes into production.	Breach of integrity or loss of data	MAJOR	Could happen once every 10 years	UNLIKELY	MODERATE

Documenting the results of risk assessment

- Final summary comprised in ranked vulnerability risk worksheet
- Worksheet details asset, asset impact, vulnerability, vulnerability likelihood, and risk-rating factor
- Ranked vulnerability risk worksheet is initial working document for next step in risk management process: assessing and controlling risk

Risk Management Strategies
Once ranked vulnerability risk worksheet complete, must choose one of four strategies to control each risk:

- Reduce/mitigate risk (security and mitigation controls)
- Share/transfer risk (outsource activity that causes risk, or insure)
- Retain risk (understand tolerate potential consequences)
- Avoid risk (stop activity that causes risk)

Treating risk from the positive dimension

- Identify options for risk treatment by seeking opportunities that might increase positive outcomes without increasing the risk.
- Options include:
  - **Actively seek** an opportunity for creating value and profit
  - **Change the likelihood of opportunity** to enhance the likelihood of beneficial outcome
  - **Change the consequences** to increase the extent of the gains
  - **Sharing** the opportunity
  - **Retain** the residual opportunity

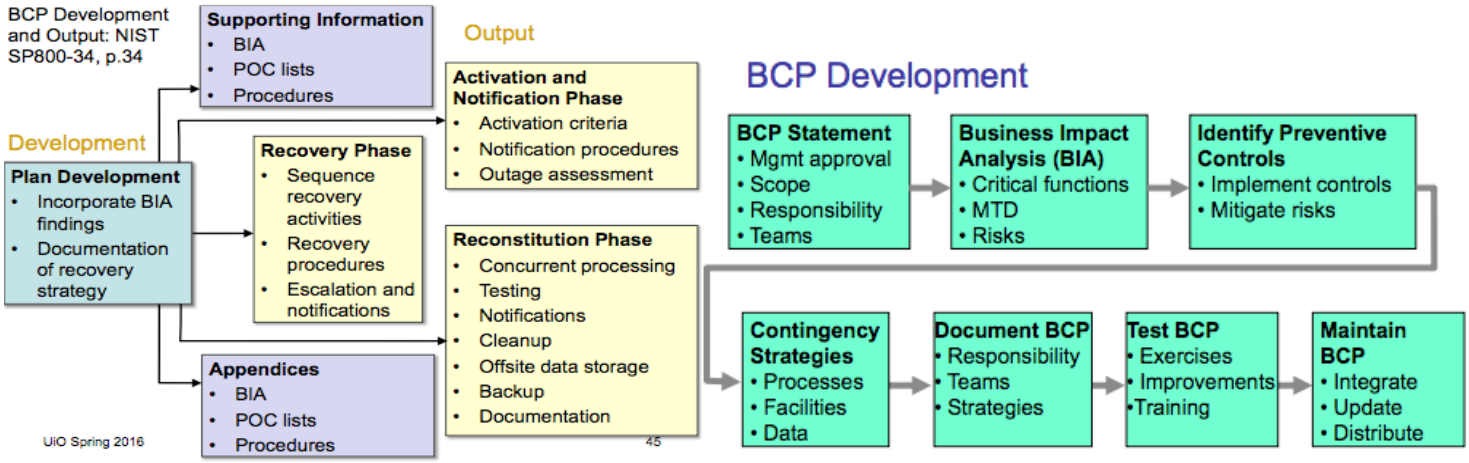
Business continuity management
The range of incidents and disasters to be considered include:

Acts of nature, like: Excessive weather conditions, Earthquake, Flood, Fire
Human acts, like: Hacker activity, Mistakes by operating staff, Theft, Fraud, Vandalism, Terrorism

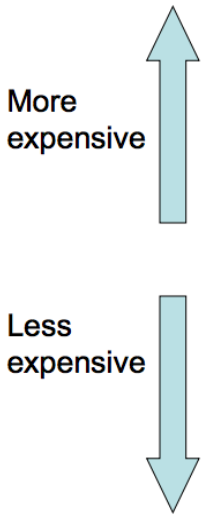
The business continuity plan describes:

– a sequence of actions – and the parties responsible for carrying them out
– in response to disasters
– in order to restore normal business operations as quickly as possible

• Business Continuity Plan
– Plan for restoring normal business functions after disruption
• Business Contingency Plan – Same as Business Continuity Plan
– Contingency means" something unpredictable that can happen"
• Disaster Recovery
– Reestablishment of business functions after a disaster, possibly in temporary facilities



A Business Impact Analysis (BIA) is performed as part of the BCP development to identify the functions that in the event of a disaster or disruption, would cause the greatest financial or operational loss.
Consider e.g.: IT network support, data processing, accounting, software development, payroll, customer support, order entry, production scheduling, purchasing, communications



Alternative sites
• Redundant site
– Mirror of the primary processing environment
– Operable within minutes
• Hot site
– Fully configured hardware and software, but no data
– Operable within hours
• Mobile site
• Warm site
– Partially configured with some equipment, but not the actual computers
– Operable within days
• Cold site
– Basic electricity and plumbing
– Operable within weeks

Strategy Selection
• Analyse alternative disaster recovery strategies
– Choosing data and software backup facility
– Choosing alternative site type and contract
– Human resources
– Insurance – Reciprocal and mutual aid agreements
– Multiple processing centres
– Data processing service bureaus
with respect to BIA, cost, restoration time and practicality

BC Activation Phase Plan
• Actions to take immediately after incident
– Procedures for contacting recovery teams
– Assessment of damage to primary site facilities
• Estimated outage time at primary site
• Compare with predefined MTD and activation criteria
– Notify BC management
– Management declares a disaster if criteria are met
– Start implementing BCP
• BCP activation responsibility
– Only one person
– CEO or other predefined role
– Succession of responsibility must be predefined

BC Recovery Phase Plan
• Evacuation and safety of personnel
– Always first priority
• Notifying alternative sites

• Securing home site
• Activation of recovery teams
• Relocation to alternative sites
• Resumption of critical business functions
• Reviewing how the organisation will interface with external parties (customers, partners) from alternative site

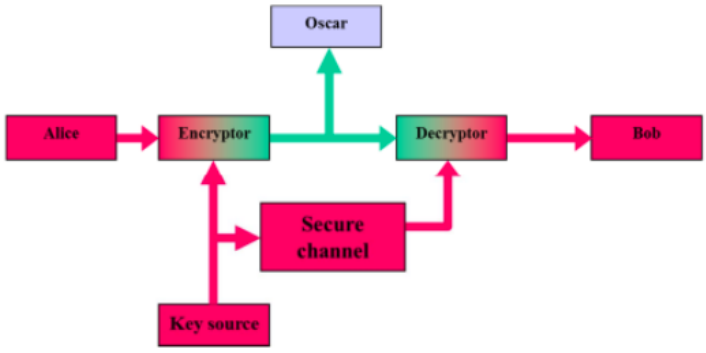
BC Reconstitution Phase Plan
• Plan for returning to normal operations at primary site
  – Repairing primary site, or prepare new site
  – Installing hardware and software
  – Testing business functions
  – Migrating business functions stepwise
• Least critical functions first
• Most critical functions last
  – Shutting down alternative site
  – Securing and removing sensitive data from alternative site

BCP Testing
• *Checklist test*
  – Copies of the BCP distributed to departments for review
• *Structured walk-through test*
  – Representatives from each department come together to go through the plan
• *Simulation test*
  – All staff in operational and support functions come together to practice executing the BCP
• *Parallel test*
  – Business functions tested at alternative site
• *Full interruption test*
  – Business functions at primary site halted, and migrated to alternative site in accordance with the BCP

Week 4

Cryptography

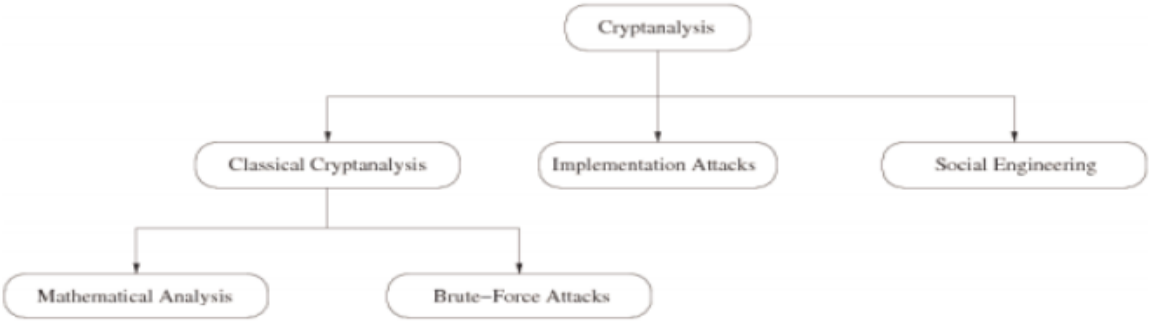
What is Cryptography?
Cryptography is the science of secret writing with the goal of hiding the meaning of a message. *Cryptanalysis* is the science and sometimes art of breaking cryptosystems.



Caesar cipher is a way of encrypting text.
Example: Caesar chipher
P = {abcdefghijklmnopqrstuvwxyz}
C = {DEFGHIJKLMNOPQRSTUVWXYZABC}
Plaintext: kryptologi er et spennende fag
Chiphertext: NUBSWRORJL HU HT VSHQQHQGH IDJ
Note: Caesar chipher in this form does not include a variable key, but is an instance of a "shift-cipher" using key K = 3.

• A cipher with a small keyspace can easily be attacked by **exhaustive search**
• A **large keyspace** is necessary for a secure cipher, but *it is by itself not sufficient*
• **Monoalphabetical substitution** ciphers can easily be broken

Cryptanalysis: Attacking Cryptosystems



Classical Attacks
- Mathematical Analysis
- Brute-Force Attack
  - Treats the cipher as a black box
  - Requires (at least) 1 plaintext-ciphertext pair (x_0, y_0)
  - Check all possible keys until condition is fulfilled: $d_K(y_0) = x_0$

Implementation Attack: Try to extract the key through reverse engineering or power measurement, e.g., for a banking smart card.
Social Engineering: E.g., trick a user into giving up her password

Kerckhoff's principles
• The system should be, if not theoretically unbreakable, unbreakable in practice.
• The design of a system should not require secrecy and compromise of the system should not inconvenience the correspondents (Kerckhoffs' principle).
• The key should be rememberable without notes and should be easily changeable
• The cryptograms should be transmittable by telegraph
• The apparatus or documents should be portable and operable by a single person
• The system should be easy, neither requiring knowledge of a long list of rules nor involving mental strain

LFSR – Linear feedback shift register
- Using n flip-flops we may generate a binary sequence of period 2 n -1
- Easy to implement in hardware
- Using "correct" feedback a register of length n may generate a sequence with period 2 n-1
- The sequence will provide good statistical properties
- Knowing 2 n consecutive bits of the key stream, will reveal the initial state and feedback
- The linearity means that a single LFSR is completely useless as a stream cipher, but LFSRs may be a useful building block for the design of a strong stream cipher

Block Ciphers: Modes of Operation
• Block ciphers can be used in different modes in order to provide different security services.
• Common modes include: **E**lectronic **C**ode **B**ook (ECB), **C**ipher **B**lock **C**haining (CBC), **O**utput **F**eed**b**ack (OFB), **C**ipher **F**eed**b**ack (CFB), **C**oun**t**e**r** Mode (CTR), **G**alois **C**ounter **M**ode (GCM) {Authenticated encryption}

Electronic Code Book

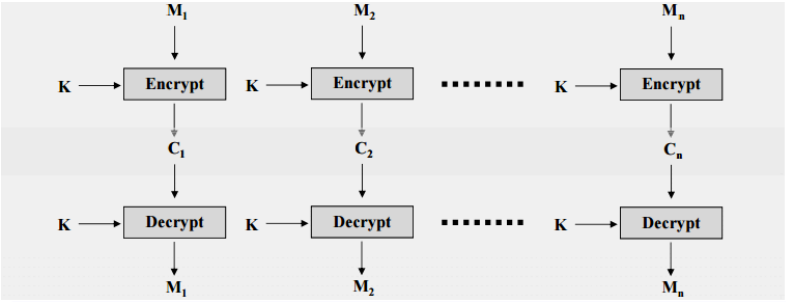
• *ECB Mode encryption*

– Simplest mode of operation

– Plaintext data is divided into blocks M1, M2, …, M_n

– Each block is then processed separately

• Plaintext block and key used as inputs to the encryption algorithm



• *ECB Mode Issues*

– Problem: For a given key, the same plaintext block always encrypts to the same ciphertext block.

• This may allow an attacker to construct a code book of known plaintext/ciphertext blocks.

• The attacker could use this codebook to insert, delete, reorder or replay data blocks within the data stream without detection

– Other modes of operation can prevent this, by not encrypting blocks independently

• For example, using the output of one block encryption as input to the next (chaining)

Integrity Check Functions

Applications of hash functions

• Protection of password

• Comparing files

• Authentication of SW distributions

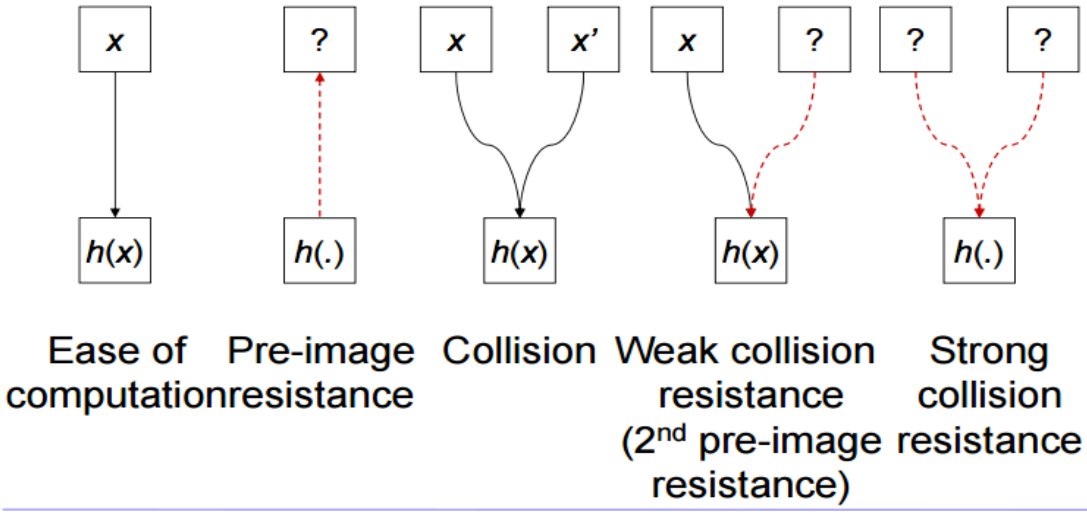
• Bitcoin • Generation of Message Authentication Codes (MAC)

• Digital signatures

• Pseudo number generation/Mask generation functions

• Key derivation

Properties of hash functions



Frequently used hash functions
• MD5: 128 bit digest. Broken. Often used in Internet protocols but no longer recommended.
• SHA-1 (Secure Hash Algorithm):160 bit digest. Potential attacks exist. Designed to operate with the US Digital Signature Standard (DSA);
• SHA-256, 384, 512 bit digest. Still secure. Replacement for SHA-1
• RIPEMD-160: 160 bit digest. Still secure. Hash function frequently used by European cryptographic service providers.
• NIST competition for new secure hash algorithm, announcement of winner in 2012.

MAC means two things:
1. The computed message authentication code h (M, k)
2. General name for algorithms used to compute a MAC

MAC algorithms, a.k.a. keyed hash functions, support data origin authentication services.

Diffie-Hellman key agreement (key exchange) (provides no authentication)
Applications:
- IPSec (IP Security)
  - IKE (Internet Key Exchange) is part of the IPSec protocol suite
  - IKE is based on Diffie-Hellman Key Agreement
- SSL/TLS
  - Several variations of SSL/TLS protocol including
    - Fixed Diffie-Hellman
    - Ephemeral Diffie-Hellman
    - Anonymous Diffie-Hellman

Examples of Cryptosystems
• RSA: best known asymmetric algorithm.
  – RSA = Rivest, Shamir, and Adleman (published 1977)
  – Historical Note: U.K. cryptographer Clifford Cocks invented the same algorithm in 1973, but didn't publish.
• ElGamal Cryptosystem
  – Based on the difficulty of solving the discrete log problem.
• Elliptic Curve Cryptography
  – Based on the difficulty of solving the EC discrete log problem.
  – Provides same level of security with smaller key sizes.

Hybrid Cryptosystems
• Symmetric ciphers are faster than asymmetric ciphers (because they are less computationally expensive), but ...
• Asymmetric ciphers simplify key distribution, therefore ...
• a combination of both symmetric and asymmetric ciphers can be used – a hybrid system:
  – The asymmetric cipher is used to distribute a randomly chosen symmetric key.
  – The symmetric cipher is used for encrypting bulk data.

Digital Signature Mechanisms
• A MAC cannot be used as evidence that should be verified by a third party.
• Digital signatures used for non-repudiation, data origin authentication and data integrity services, and in some authentication exchange mechanisms.
• Digital signature mechanisms have three components:
  – key generation
  – signing procedure (private)

– verification procedure (public)
• Algorithms
– RSA
– DSA and ECDSA

Digital Signatures
• To get an authentication service that links a document to A's name (identity) and not just a verification key, we require a procedure for B to get an authentic copy of A's public key.
• Only then do we have a service that proves the authenticity of documents 'signed by A'.
• This can be provided by a PKI (Public Key Infrastructure)
• Yet even such a service does not provide nonrepudiation at the level of persons.

Difference between MACs & Digital Signatures
• MACs and digital signatures are both authentication mechanisms.
• MAC: the verifier needs the secret that was used to compute the MAC; thus a MAC is unsuitable as evidence with a third party.
– The third party does not have the secret.
– The third party cannot distinguish between the parties knowing the secret.
• Digital signatures can be validated by third parties, and can in theory thereby support both non-repudiation and authentication.

Week 5 - Key Management and PKI

Key Management
• The strength of cryptographic security depends on:
1. The size of the keys
2. The robustness of cryptographic algorithms/protocols
3. The protection and management afforded to the keys
• Key management provides the foundation for the secure generation, storage, distribution, and destruction of keys.
• Key management is essential for cryptographic security.
• Poor key management may easily lead to compromise of systems where the security is based on cryptography

Key Usage
• A single key should be used for **only one** purpose, example: encryption, authentication, key wrapping, random number generation, or digital signature generation.
• Using the same key for two different purposes may weaken the security of one of both purposes
• Limiting the use of a key limits the damage that could be done if the key is compromised.
• Some uses of keys interfere with each other, example: an asymmetric key pair should only be used for either encryption or digital signatures, not both

Crypto Period
The crypto period is the time span during which a specific key is authorized for use, important because:
- Limits the amount of information, protected by a given key, that is available for cryptanalysis.
- Limits the amount of exposure and damage, should a single key be compromised.
- Limits the use of a particular algorithm to its estimated effective lifetime

Factors Affecting Crypto-Periods

• In general, as the sensitivity of the information or the criticality of the processes increases, the crypto-period should decrease in order to limit the damage resulting from compromise.
• Short crypto-periods may be counter-productive, particularly where denial of service is the paramount concern, and there is a significant overhead and potential for error in the re-keying, key update or key derivation process.
• The crypto-period is therefore a **trade-off**

Key Usage Periods

• A key can be used for protection and/or processing.
  – Protection: Key is e.g. used to encrypt or to generate DigSig
  – Processing: Key is e.g. used to decrypt or to validate DigSig
• The **crypto-period** lasts from the beginning of the protection period to the end of the processing period.
• A key **shall not** be used **outside** of its specified period.
• The processing period can continue after the protection period.

Recommended Crypto Periods

Key Type	Cryptoperiod	
	Originator-Usage Period OUP (Protection Period)	Recipient-Usage Period (Processing Period)
1. Private Signature Key	1-3 years	---
2. Public Signature Key	Several years (depends on key size)	
3. Symmetric Authentication Key	<= 2 years	<= OUP + 3 years
4. Private Authentication Key	1-2 years	
5. Public Authentication Key	1-2 years	
6. Symmetric Data Encryption Keys	<= 2 years	<= OUP + 3 years
7. Symmetric Key Wrapping Key	<= 2 years	<= OUP + 3 years
8. Symmetric RBG Key (Random Bit Generator)	(See SP800-90)	

Key Type	Cryptoperiod	
	Originator-Usage Period OUP (Protection Period)	Recipient-Usage Period (Processing Period)
9. Symmetric Master Key	About 1 year	
10. Private Key-Transport Key	<= 2 years	
11. Public Key-Transport Key	1-2 years	
12. Symmetric Key-Agreement Key	1-2 years	
13. Private Static Key-Agreement Key	1-2 years	
14. Public Static Key-Agreement Key	1-2 years	

Key Type	Cryptoperiod	
	Originator-Usage Period OUP (Protection Period)	Recipient-Usage Period (Processing Period)
15. Private Ephemeral Key Agreement Key	One key-agreement transaction	
16. Public Ephemeral Key Agreement Key	One key-agreement transaction	
17. Symmetric Authorization (Access Control) Key	<= 2 years	
18. Private Authorization (Access Control) Key	<= 2 years	
19. Public Authorization (Access Control) Key	<= 2 years	

Key Generation

• Most sensitive of all cryptographic functions.
• Need to prevent unauthorized disclosure, insertion, and deletion of keys.
• Automated devices that generate keys and initialization vectors (IVs) should be physically protected to prevent:
  – disclosure, modification, and replacement of keys,
  – modification or replacement of IVs.
• Keys should be randomly chosen from the full range of the key space
  – e.g. 128 bit keys give a key space of 2128 different keys

Random number generator seeds

- RNG keys are used to initialize the generation of random symmetric and asymmetric keys
- Knowing the seed may determine the key uniquely
- Requires confidentiality and integrity protection
    - Periods of protection for seeds, example:
        - Used once and destroyed
        - Used for multiple keys, destroyed after last key generation
        - Kept and destroyed at the end of the protection period

Compromise of keys and keying material

- Key compromise occurs when it is known or suspected that an unauthorized entity has obtained a secret/private key.
- When a key is compromised, immediately stop using the secret/public key for **protection**, and revoke the compromised key (pair).
- The continued use of a compromised key must be limited to processing of protected information.
    - In this case, the entity that uses the information must be made fully aware of the risks involved.
    - Continued key usage for processing depends on the risks, and on the organization's Key Management Policy.

Key Compromise Recovery Plan

A Compromise recovery plan **should** contain:

- The identification of the parties to notify.
- The identification of the personnel to perform the recovery actions.
- The re-key method.
- Any other recovery procedures, such as:
    - Physical inspection of equipment.
    - Identification of all information that may be compromised.
    - Identification of all signatures that may be invalid due to the compromise of a signing key.
    - Distribution of new keying material, if required.

Undetected Key Compromise

The worst form of key compromise is when a key is compromised without detection.

- Nevertheless, certain protective measures can be taken.

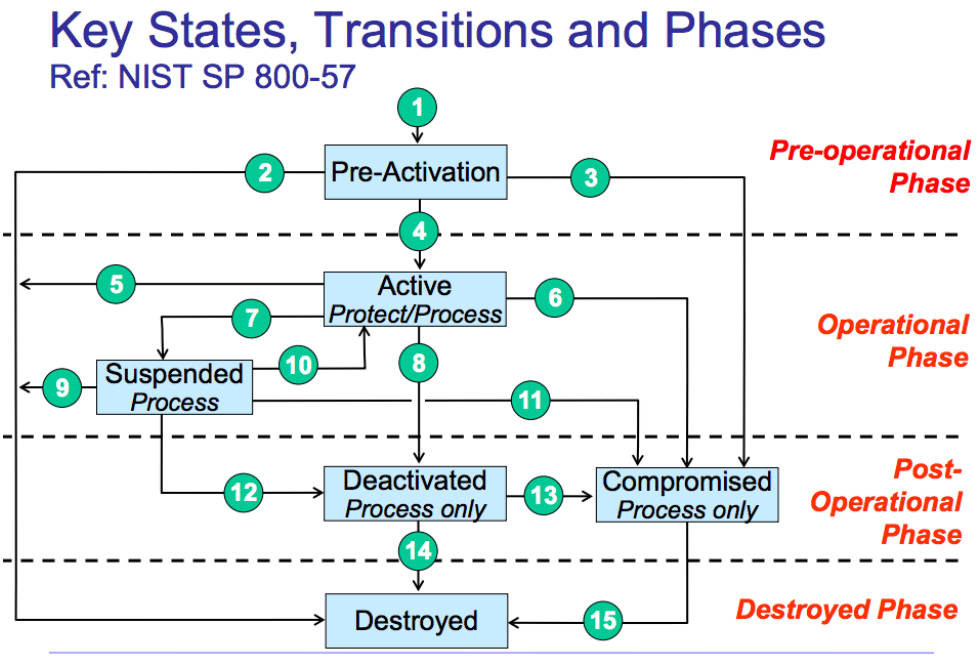
Key management systems (KMS) **should** be designed:

- To mitigate the negative effects of (unknown) key compromise.
- So that the compromise of a single key has limited consequences
- Example, a single key should be used to protect only a single user or a limited number of users, rather than a large number of users.

Often, systems have alternative methods for security

- Example, to authenticate systems and data through other means that only based on cryptographic keys.

Avoid building a system with catastrophic weaknesses.



Key protection

Active keys should be accessible for authorized users and protected from unauthorized users
Deactivated keys must be kept as long as there is data protected by keys. Policy must specify:

- Where keys shall be kept.
- How keys shall be kept securely.
- How to access keys when required.

Examples

- Symmetric ciphers
  - Never stored or transmitted 'in the clear'
  - May use hierarchy: session keys encrypted with master
  - Master key protection:
    - Locks and guards
    - Tamper proof devices
    - Password/passphrases
    - Biometrics
- Asymmetric ciphers
  - Private keys ned confidentiality protection
  - Public keys need integrity/authenticity protection

Key destruction

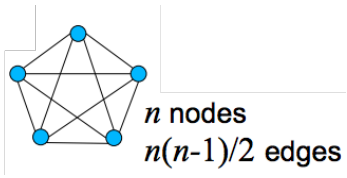
No key material should exist in in a volatile memory or on permanent storage media after destruction. Examples of key destruction methods:

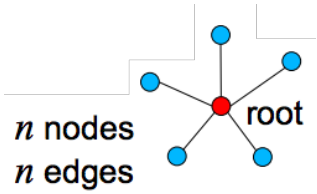
- *Simple* delete operation on computer
  - May leave undeleted key, example in recycle bin or on disk sectors
- *Special* delete operation on computer
  - That leaves no residual data, example by overwriting
- Magnetic media degaussing (deleting process)
- Destruction of physical device, example high temperature
- Master key destruction which logically destructs subordinate keys

Cryptography solves security problems in open networks, but creates *key management complexity*. **Public-key cryptography** simplifies the key management, but *creates trust management challenges*.

Key distribution: The Challenge

- Network with *n* nodes
- We want every pair of nodes to be able to communicate securely under cryptographic protection
- How many secure key **distributions** are needed?
    - Symmetric secret keys: confidentiality required?
        - n(n-1)/2 distributions, quadratic growth
        - Impractical in open networks
    - Asymmetric public keys: Authenticity required,
        - *n*(*n*-1)/2 distributions, quadratic growth
        - Impractical in open networks
    - Asymmetric public keys wit PKI: Authenticity required,
        - 1 root public key distributed to *n* parties
        - linear growth
        - … more difficult than you might think



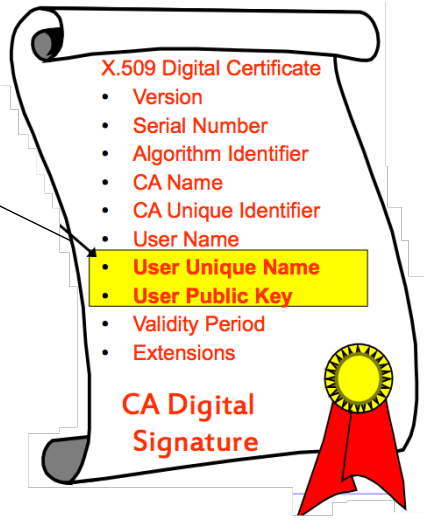


Public-key infrastructure

• Due to spoofing problem, public keys must be digitally signed before distribution.
• The main purpose of a PKI is to ensure authenticity of public keys.
• PKI consists of:
– **Policies** (to define the rules for managing certificates)
– **Technologies** (*t*o implement the policies and generate, store and manage certificates)
– **Procedures** (related to key management)
– **Structure of public key certificates** (public keys with digital signatures)

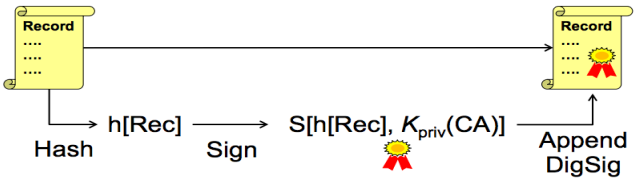
Public-Key Certificates

- A public-key certificate is simply a public key with a digital signature
- Binds name to public key
- Certification Authorities (CA) sign public keys.
- An authentic copy of CA's public key is needed in order to validate certificate
- **Relying party** validates the certificate (i.e. verifies that user public key is authentic)

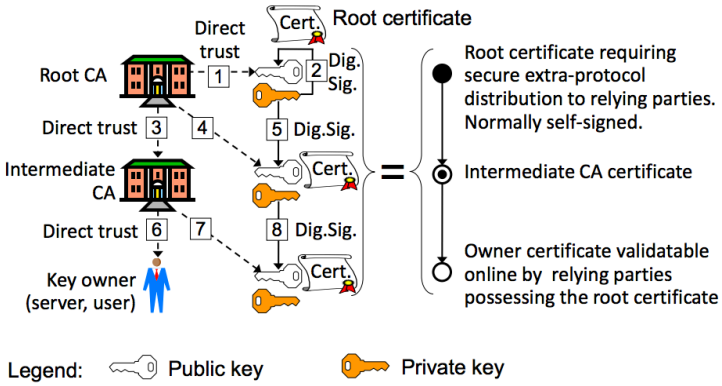


How to generate a digital certificate?

1. Assemble the information (name and public key) in single record Rec
2. Hash the record
3. Sign the hashed record
4. Append the digital signature to the record



PKI certificate generation



Self-signed root keys: Why?

• Many people think a root public key is authentic just because it is self-signed

• This is deceptive

– Gives impression of assurance

– Disguises insecure distribution of root key

– Gives false trust

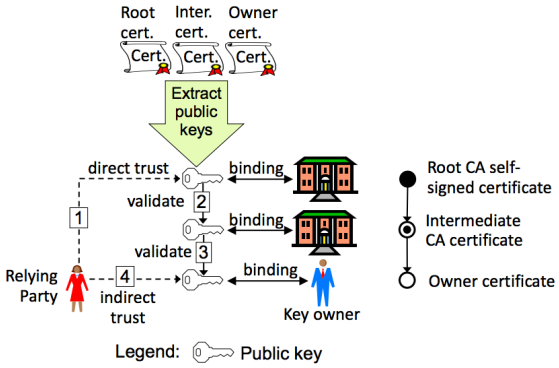
• Self-signing provides absolutely no security

• Only useful purposes of self-signing:

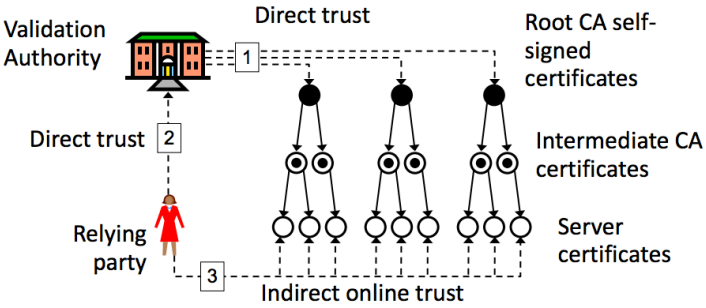
– X.509 certificates have a field for digital signature, so an empty field might cause applications to malfunction. A self-signature is a way to fill the empty field

– Self-signature can be used to specify a cert as a root

Certificate and public key validation

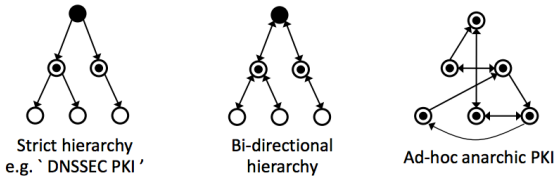


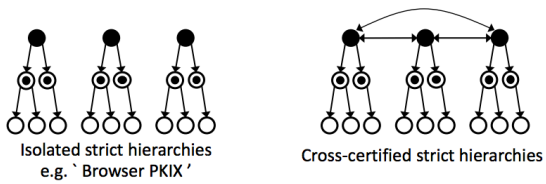
Validation Authorities



• A validation authority can assist relying parties to validate certificates

PKI Trust Models





PKI trust models

Strict hierarchical model

• Advantages:

– works well in highly-structured setting such as military and government

– unique certification path between two entities (so finding certification paths is trivial)

– scales well to larger systems

• Disadvantages:

– need a trusted third party (root CA)

– 'single point-of-failure' target

– If any node is compromised, trust impact on all entities stemming from that node

– Does not work well for global implementation (who is root TTP?)

PKI trust models

User-centric model

- Each user is **completely responsible** for deciding which public keys to trust
- Example: Pretty Good Privacy (PGP)

PKI trust models

User-centric model

• Advantages:
  – Simple and free
  – Works well for a small number of users
  – Does not require expensive infrastructure to operate
  – User-driven grass-root operation
• Disadvantages:
  – More effort, and relies on human judgment
    • Works well with technology savvy users who are aware of the issues. Does not work well with the general public
  – Not appropriate for more sensitive and high risk areas such as finance and government

Browser PKI and malicious certificates

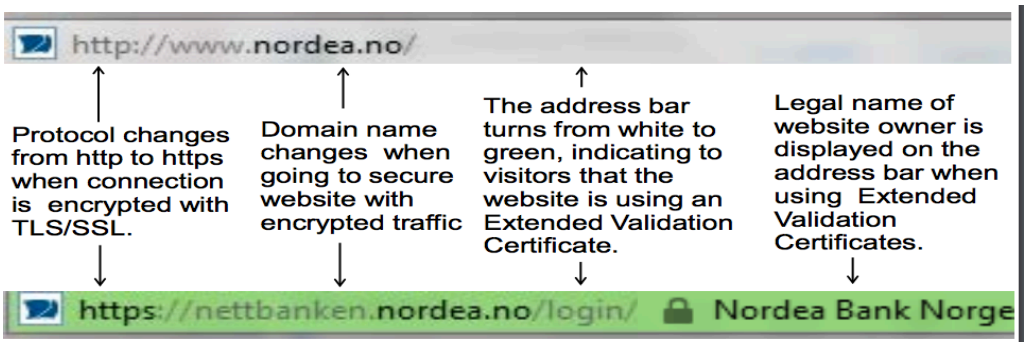
• The browser automatically validates certificates by checking: certificate name = domain name
• Criminals buy legitimate certificates which are automatically validated by browsers
  – Legitimate certificates can be used for malicious phishing attacks, e.g. to masquerade as a bank
  – **Malicious certificates are legitimate certificates !!!**
• Server certificate validation is not authentication
  – Users who don't know the server domain name cannot distinguish between right and wrong server certificates

Browser PKI root certificate installation

• Distribution of root certificates which should happen securely out-of-band, is often done through online downloading of browser SW
• Users are in fact trusting the browser vendor who supplied the installed certificates, rather than a root CA
• *Example*: used by Mozilla Firefox and Microsoft Internet Explorer
• Browser vendors decide which CA certs to distribute with browsers
  – This is an important political issue

Extended validation certificates

• Problem with simple certificates:
  – Can be bought by anonymous entities
• EV (Extended Validation) certificates require registration of legal name of certificate owner.
• Provides increased assurance in website identity.
• However, EV certificates are only about identity, not about honesty, reliability or anything normally associate with trust.
• Even the Mafia.com can buy EV certificates through legal businesses that they own.



Stuxnet with valid SW signature

- Stuxnet worm is described as the most advanced malware attack ever, because
  - It used multiple zero-day exploits
  - It targeted a specific industrial control system
  - It was signed under a valid software (SW) certificate
- Stuxnet worm could be automatically validated by every browser in the whole world
- Anybody can buy SW certificates and sign whatever they want, even the Mafia !!!
- SW certificates only give evidence about who signed the SW, not that the SW is trustworthy.

PKI services

- Several organizations operate PKI services
  - Private sector
  - Public sector
  - Military sector
- Mutual recognition and cross certification between PKIs is difficult
- Expensive to operate a robust PKI
- The Browser PKI is the most widely deployed PKI thanks to piggy-backing on browsers and the lax security requirements
- DNSSEC PKI might replace the browser PKI

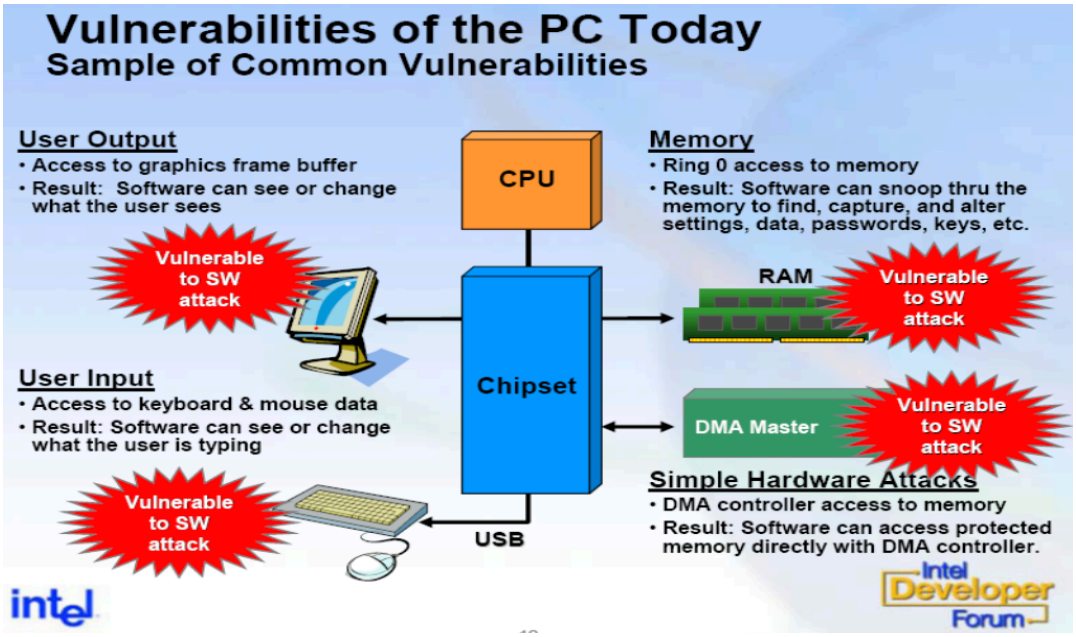
PKI Summary

- Public key cryptography needs a PKI to work
  - Reduces number of key distributions from quadratic to linear.
  - Digital certificates used to provide authenticity and integrity for public keys.
  - Acceptance of certificates requires trust.
  - Trust relationships between entities in a PKI can be modelled in different ways.
  - Establishing trust has a cost, e.g. because secure out-of-band channels are expensive.

Week 6 – Computer Security

System & Communication Security

"Using encryption on the Internet is the equivalent of arranging an armored car to deliver credit card information from someone living in cardboard box to someone living on a park bench" – Gene Spafford



Approaches to strengthening platform security

- Harden the operating system
  - SE (Security Enhanced) Linux, Trusted Solaris, Windows Vista/7/8
- Add security features to the CPU
  - Protection Layers, NoExecute, ASLR
- Virtualization technology
  - Separates processes by separating virtual systems.
- Trusted Computing
  - Add secure hardware to the commodity platform
  - Example: TMP (Trusted Platform Module)
- Rely on secure hardware external to commodity platform
  - Smart cards
  - Hardware tokens

TCB – Trusted Computing Base

• The trusted computing base (TCB) of a computer system is the set of all hardware, firmware, and/or software components that are critical to its security, in the sense that bugs or vulnerabilities occurring inside the TCB might jeopardize the security properties of the entire system.

• By contrast, parts of a computer system outside the TCB must not be able to breach the security policy and may not get any more privileges than are granted to them in accordance to the security policy

(TCSEC – Trusted Computer Evaluation Criteria, 1985)

Reference Monitor

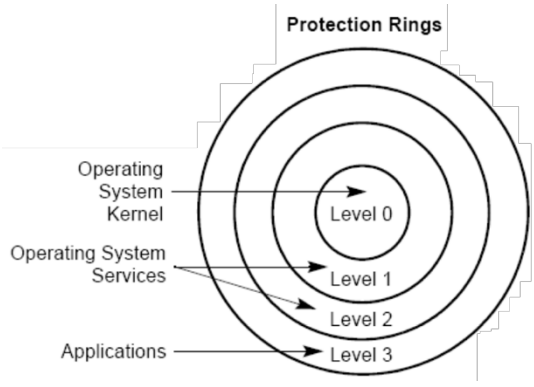
• Reference monitor is the security model for enforcing an access control policy over subjects' (e.g., processes and users) ability to perform operations (e.g., read and write) on objects (e.g., files and sockets) on a system.

– The reference monitor must always be invoked (complete mediation).
– The reference monitor must be tamperproof (tamperproof).
– The reference monitor must be small enough to be subject to analysis and tests, the completeness of which can be assured (verifiable).

• The security kernel of an OS is a low-level (close to the hardware) implementation of a reference monitor.

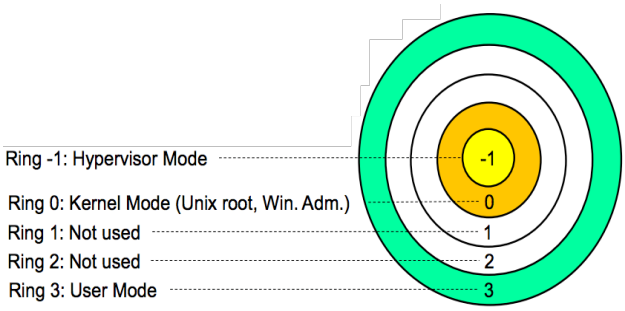
Hierarchic security levels were introduced in X86 CPU architecture in 1985 (Intel 80386)
4 ordered privilege levels

- Ring 0: highest
- Ring 3: lowest
- Intended usage → see diagram:



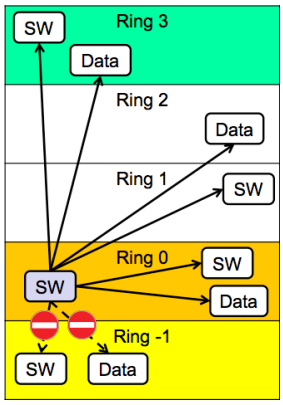
CPU Protection Ring structure from 2006

- New Ring -1 introduced to virtualization.
- Necessary for protecting hypervisor from VMs (Virtual Machines) running in Ring 0.
- Hypervisor controls VMs in Ring 0
- Ring 0 aka.: Supervisor Mode



Privileged Instructions

- Some of the system instructions (called "privileged instructions") are protected from use by application programs.
- The privileged instructions control system functions /such as the loading of system registers). They can be executed only when the Privilege Level is 0 or -1 (most privileged)
- If one of these instructions is attempted when the Privilege Level is not 0 or -1, then a general-protection exception (#GP) is generated, and the program crashes.



Principle of protection ring model

- A process can access and modify any data and software at the same of less privileged level as itself.
- A process that runs in kernel mode (Ring 0) can access data and SW in Rings 0, 1, 2 and 3, *but not in Ring -1*
- The goal of attackers is to get access to kernel or hypervisor mode.
  - through exploits
  - by tricking users to install software

User processes access to system resources

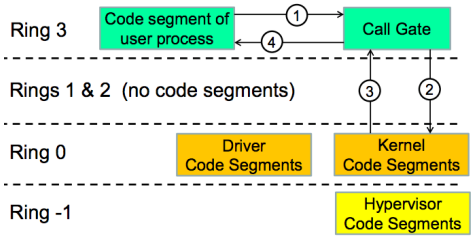
- User processes need to access system resources (memory and drivers)
- User application processes should not access system memory directly, because they could corrupt memory.
- The CPU must restrict direct access to memory segments and other resources depending on the privilege level.

• *Question 1*:
How can a user process execute instructions that require kernel mode, e.g. for writing to memory? – **Answer**: The CPU must switch between privilege levels

• *Question 2*:
How should privilege levels be switched?
– **Answer**: Through Controlled invocation of code segments

Controlled Invocation

- The user Process executes code in specific code segments.
- Each code segment has an associated mode which dictates the privilege level the code executes under.
- Simply setting the mode of user process code to Kernel would give kernel-privilege to user process without any control of what the process actually does. Bad idea!



- Instead, the CPU allows the user process to call kernel code segments that only execute a predefined set of instructions in kernel mode, and then returns control back to the user-process code segment in user mode.
- We refer to this mechanism as **controlled invocation**.

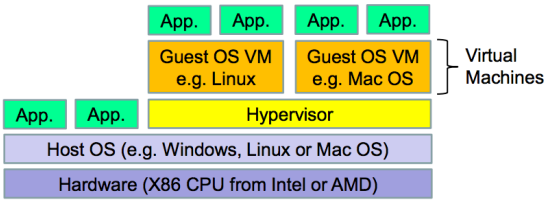
Virtual machines (VM)

- A software implementation of a machine (OS) that executes programs like a real machine (traditional OS) example:
    - Java Virtual Machine (JVM)
        - JVM accepts a form of computer intermediate language commonly preferred to as java bytecode.
            - "compile once, run anywhere"
        - The JVM translates the bytecode to executable code on the fly
    - Platform Virtualization
        - Simultaneous execution of multiple Oss on a single computer hardware, so each OS becomes a virtual computing platform

Platform Virtualization

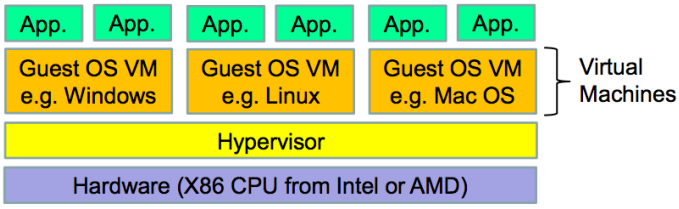
- Hypervisor (aka. VMM – Virtual Machine Monitor) is needed to manage multiple guest OSs (virtual machines) in the same hardware platform.
- Many types of hypervisors available
    - VMWare is most known Commercial product
        - Free version comes with a limitations
    - VirtualBox is a hypervisor for x86 virtual
        - It is freely availably under GPL
        - Runs on Windows, Linux, OS X and Solaris hosts
    - Hyper-V is Microsoft's hypervisor technology
        - Requires Windows Server

Type 2 VM Architecture (simple virtualization)



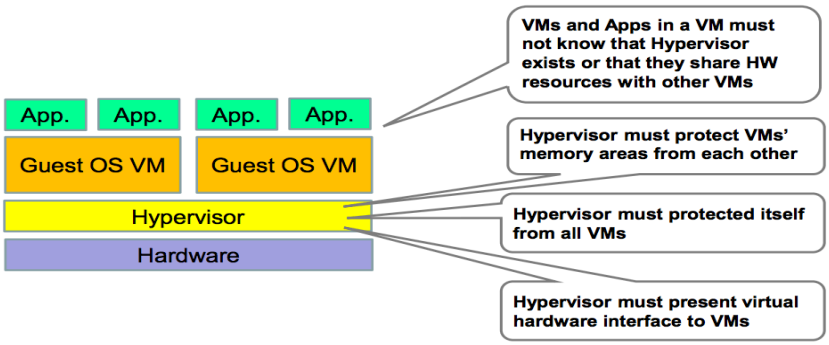
• Hypervisor runs on top of host OS
• Performance penalty, because hardware access goes through 2 Oss
• Traditionally good GUI
• Traditionally good HW support, because host OS drivers available

Type 1 VM Architecture (advanced virtualization)



• No host OS
• Hypervisor runs directly on hardware
• High performance
• Traditionally limited GUI, but is improved in modern versions
• HW support can be an issue

Challenges of Running VMs



Why use platform virtualization

- Efficient use of hardware and resources
  - Improved management and resource utilization
  - Saves Energy
- Improved security
  - Malware can only infect the VM
  - Safe testing and analysis of malware
  - Isolates VMs from each other
- Distributed applications bundled with OS
  - Allows optimal combination of OS and application
  - Ideal for cloud services
- Powerful debugging
  - Snapshot of current state of the OS
  - Step through program and OS execution
  - Reset system state

Hypervisor examples of use

- Cloud providers run large server parks
  - Each customer gets its own VM
  - Many customers share the same hardware
  - Migrated VMs between servers to increase/reduce capacity
- Testing and software analysis
  - Potentially damaging experiments can be executed in isolated environment
  - Take a snapshot of the current state of the OS
  - Use this later on to reset the system to that state
  - Malware Analysis

Basic idea of Trusted Computing

• Use specialised security hardware as part of TCB in a computer system
  – Can not be compromised by malware
  – Can verify the integrity of OS kernel
  – Can make physical tampering difficult
  – Can report status of system to remote parties
  – Can report identity of system to remote parties
• Gives increased level of trust that the system will perform as expected/specified

What is "trust" in the sense of TC?

• To have confidence in assumptions about security
• Trust is to believe that security assertions will hold
"A trusted component, operation, or process is one whose behaviour is assumed to be correct under any operating condition, and which is assumed to resist subversion by malicious software, viruses, and manipulations"
• A trusted component enforces the security policy as long as these assumptions hold
• A trusted component violates the security policy if it breaks
• Q1: How do you know that a component is 'trustworthy', i.e. that it will not break?
A1: Through 'assurance'
• Q2: Trusted by whom to do what?
  – Trusted by user, by vendor, or by 3rd party (NSA)
  – What if they have conflicting interests?

Characteristics of Trusted Hardware

• Physically secure hardware component
  – Assumed not to break because it's hardware
• Environmental monitoring (temperature, power supply, structural integrity)
• Tamper responsive
• Implementations
  – CPU
  – ROM for OS and application code
  – Specialized hardware for cryptography and for storing secrets

Boot protection

• BIOS /UEFI
  – First code run by a PC when powered on
  – BIOS/UEFI initializes and identifies system devices such as display and keyboard
  – BIOS/UEFI then loads software (OS) held on a peripheral device such as a hard disk
  – BIOS/UEFI firmware is stored in ROM
• Boot protection
  – Persistent OS infection is the goal of APT
  – Boot protection focuses on verifying the integrity of the OS during boot.
  – Does not protect against infection during runtime

Two Modes of boot Protection

• Secure boot with UEFI (not with TPM, see UEFI later)
  – The platform owner can define expected (trusted) measurements (hash values) of OS software modules.
  – Hash values stored in memory signed by private PK (Platform Key).
  – Public PK stored in secure firmware on platform
  – Measured has values can be compared with stored values.
  – Matching measurement values guarantee the integrity of the corresponding software modules.
  – Boot process terminates if a measurement does not match the stored value for that stage of the boot process.
• Authenticated/Measured boot with TPM
  – Records measured values in PCRs and reports to remote party
  – Does not terminate boot if measured values are wrong

Sealed Storage/Encryption

- Encrypts data so it can be decrypted
  - by a certain machine in given configuration
- Depends on
  - Storage Root Key (SRK) unique to machine
  - Decryption only possible on unique machine
- Can also extend this scheme upward
- Create application key for desired application version running on desired system version.
- Supports disk encryption

Remote Attestation (declaration)

- TPM can certify configuration to others
  - with a digital signature in configuration info
  - giving another user confidence in it
  - Based on Attestation Key (AK)

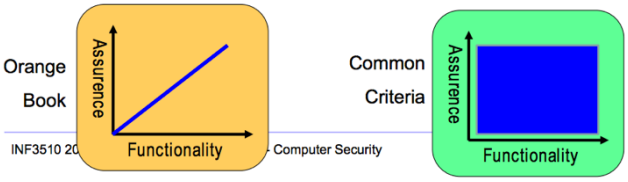
- Remote parties can validate signature based on a PKI
- Provides hierarchical certification approach
    - trust TPM, then trust the OS, then trust applications.

UEFI (Unified Extensible Firmware Interface)
- Replaces traditional BIOS (Basic Input-Output System)
- Like BIOS it hands control of the pre-boot environment to an OS
- Key Security **Benefits**: Secure Boot:
    - Prevents loading unsigned drivers or OS loaders
    - When secure boot is enabled, it is initially placed in "setup" mode, which writes public key known as the "Platform key" (PK) to firmware.
    - Once the key is written, secure boot enters "User" mode, where only drivers and loaders signed with the platform key can be loaded by the firmware.
    - "Key Exchange Keys" (KEK), signed by private PK, can be added to a database stored in memory to allow other signatures by other than PK.
    - Secure boot supported by Win 8, Win Server 2012, Fedora, OpenSuse, and Ubuntu
    - Does not require TPM

Security Evaluation
• How do you get assurance that your computer systems are adequately secure?
• You could trust your software providers.
• You could check the software yourself, but you would have to be a real expert, and it would take long.
• You could rely on an impartial security evaluation by an independent body.
• Security evaluation schemes have evolved since the 1980s; currently the **Common Criteria** are used internationally.

Target & Purpose
• Target of evaluation
– Product: “off-the-shelf” software component to be used in a variety of applications; has to meet generic security requirements
– System: collection of products assembled to meet the specific requirements of a given application
• Purpose of evaluation
– Evaluation: assesses whether a product has the security properties claimed for it
– Certification: assesses suitability of a product (system) for a given application
– Accreditation: decide to use a certain system

Functionality/Assurance Structure
• Two dimentions of evaluation
1. Functionality: the security features
2. Assurance: the robustness of the security features
• Orange Book: assurance levels for a given set of typical DoD requirements, considers both aspects simultaneously.
• CC: flexible evaluation framework that can deal with arbitrary feature sets at any assurance level; the two aspects are addressed independently.



Common Criteria

- Criteria for the security evaluation of products or systems, called the Target of Evaluation (TOE).
- Protection Profile (PP): a (re-usable) set of security requirements, including an EAL; should be developed by user communities to capture typical protection requirements.
- Security Target (ST): expresses security requirements for a specific TOE, e.g. by reference to a PP; basis for any evaluation.
- Evaluation Assurance Level (EAL): define the specific evaluation requirements that must be satisfied in an evaluation; there are seven hierarchically ordered EALs.

CC Standard

- Part 1 – Overview
- Part 2 – SFRs Security Functional Requirements
    - Security Functional Requirements (SFRs) are "what does the product does." Taken together, the SFRs a product claims describe the product's capabilities. A product's security features, for example, might be how it identifies and authenticates users.
- Part 3 – SARs: Security Assurance Requirements
    - Security Assurance Requirements (SARs) define the development environment in all its phases: specification, development tools and practices, for example, the use of automated tools to prevent unauthorized modifications to the product, the completeness of test coverage.

CC Assurance Levels

• EAL1 - functionally tested
• EAL2 - structurally tested
• EAL3 - methodically tested and checked
• EAL4 - methodically designed, tested, and reviewed
• EAL5 - semiformally designed and tested
• EAL6 - semiformally verified design and tested
• EAL7 - formally verified design and tested

Using the Common Criteria

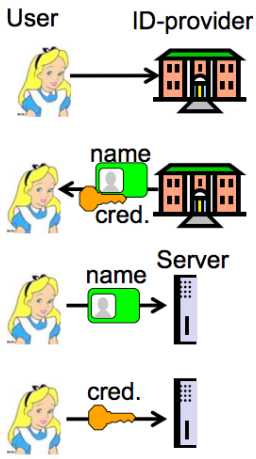
• CC is useful for:
– Specifying security features in product or system
– Assisting in the building of security features into products or systems
– Evaluating the security features of products or systems
– Supporting the procurement of products or systems with security features
– Supporting marketing of evaluated products
• But
– Evaluation is expensive and slow
– New versions of a product must be re-evaluated, but can be done more quickly than the original evaluation.

Week 8 – User Authentication

Outline

- Concepts related to authentication
    - Identify and authentication steps
- User Authentication
    - Knowledge-Based Authentication (**Password**)
    - Ownership-Based Authentication(**Tokens**)
    - Inherence-Based Authentication(**Biometrics**)
- Authentication frameworks for e-Government

Steps of User Authentication



Operation phase

1. ***Registration***
   - User visits ID-provider, with pre-authentication credents.
2. ***Provisioning***
   - ID-provider registers unique name and issues credential

Configuration phase

3. ***Self-identification***
   - User presents the unique name to claim identity
4. ***Verification of identity***
   - Proof of identity based on credentials

User authentication credentials

A credential is the 'things' used for authentication.

- May also be referred to as a "token" or "authenticator"
- *Example:* reusable passwords, PIN, biometrics, smart cards, certificates, cryptographic keys OTP (One-Time-Passwords) hardware tokens.

Credential categories:

1. Knowledge-Based Authentication (Something you know): **Password**
2. Ownership-Based Authentication (Something you have): **Tokens**
3. Inherence-Based Authentication (Something you are/do): **Biometrics**
   - physiological biometric characteristics
   - behavioral biometric characteristics

Combinations, called multi-factor authentication

Authentication: Static passwords

- Passwords are a simple and most-often-used authenticator.
  - Something the user knows
- Problems:
  - Easy to share (intentionally or not)
  - Easy to forget
  - Often easy to guess (weak passwords)
  - Can be written down (both good and bad)
    - If written down, then "what you know" is "where to find it"
  - Often remains in memory and cache

Secure password strategies

Passwords length ≥ 13 characters

Use ≥ categories of characters

- L-case, U-case, numbers, special characters

Do not use ordinary words (names, dictionary Wireless distribution system)

Change typically every 3 -13 months

Reuse only between low-sensitivity accounts

Store passwords securely

- On paper
- In clear text on offline digital device
- Encrypted on online digital device

Strategies for strong passwords

User education and policies

- Not necessarily with strict enforcement

Proactive password checking

- User selects a potential password which is tested
- Weak passwords are not accepted

Reactive password checking

- SysAdmin periodically runs password cracking tool (also used by attackers) to detect weak passwords that must be replaced.

Computer-generated passwords

- Random passwords are strong but difficult to remember
- FIPS PUB 181 http://www.itl.nist.gov/fipspubs/fip181.htm - specifies automated pronounceable password generator

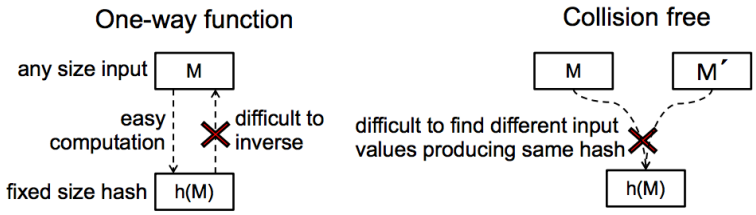
Password storage in OS

• /etc/shadow is the file where modern Linux/Unix stores it passwords
  – Earlier version stored it in /etc/passwd
  – Need root access to modify it
• \windows\system32\config\sam it the file Windows systems normally stores it passwords
  – Undocumented binary format

Prevent exposure of password file

- The computer verifies user passwords against stored values in the password file
- Password file must be available to OS
  - This file need protection from users and applications
  - Avoid offline dictionary attacks
- Protection measures
  - Access control (only accessible by OS kernel)
  - Hashing or Encryption
- In case a password file gets stolen, then hashing/encryption can provide protection.

Hash functions



A hash function is easy to compute but hard to invert.
Passwords can be stored as hash values.
Authentication function first computes hash of received password, then compares against stored hash value.

Cracking passwords

Brute force

- Trying all possible combinations

Intelligent search

- User name, Name of friends/relative, Phone number, Birth dates, Dictionary attack (Try all words from a dictionary, Precomputed hashes: Rainbow tables

Hash table and rainbow table attacks

- Attackers can compute and store hash values for all possible passwords up to a certain length
- A list of password hashes is a **hash table**
- A compressed hash table is a **rainbow table**
- Comparing and finding matches between hashed passwords and hash/rainbow table is used to determine cleartext passwords.

Password salting: Defense against password cracking

- Prepend or append random data (salt) to a user's password before hashing
    - In Unix: a randomly chosen integer from 0 to 4095-
    - Different salt for each user
    - Produces different hashes for equal passwords
    - Prevent that users with identical passwords get the same password hash value
    - Increases the amount of work required for hash table attacks and rainbow table attacks.

Methods for storing passwords on server

Password example: 123456

Cleartext passwords (low security) - Stored on server: 123456

Hashed password (moderately security)

- Stored on server: example SHA1-hash of password: 7c4a8d09ca3762af61e59520943dc26494f8941b

Salted password (good security)

- Stored on server: salt + salted hash, example: "salt": f8b97abc30b72e54, SHA1-hash of password + salt: 1736f11fae29189749a8a54f45e25fb693c3959d

Problems with using passwords in the clear

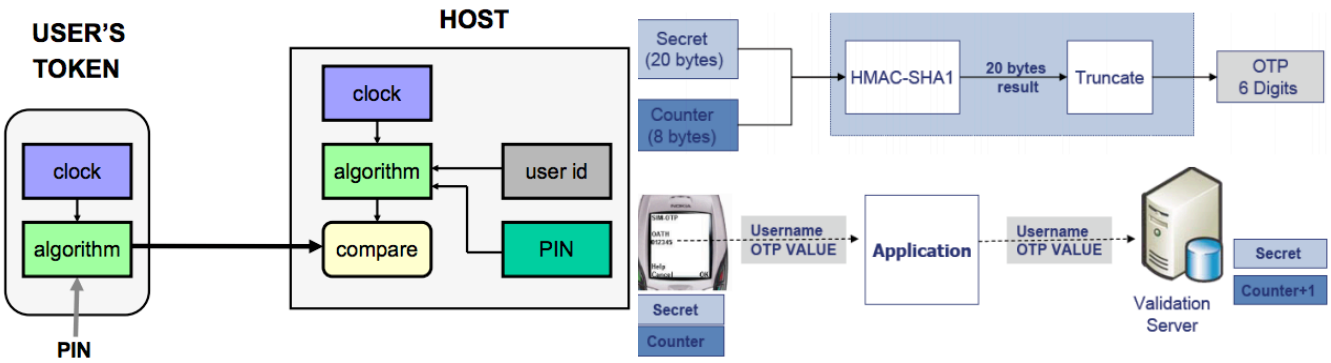
- A password sent "in clear" can be captured during transmission, so an attacker may reuse it.
- An attacker setting up a fake server can get the password from the user, **example**: phishing attack
- Solutions to these problems include: Encrypted communication channel, One-time passwords (token-based authentication), Challenge-response protocols

Synchronized OTP (One-Time-Password) Generator

- Using a password only once significantly strengthens the strength of user authentication.
- Synchronized password generators produce the same sequence of random passwords both in the token and at the host system.
    - OTP is 'something you have' because generated by token
- There are two general methods: Clock-based tokens **and** Counter-based tokens

Clock-based OTP tokens: Operation

- Tokens displays time-dependent code on display
    - User copies code from token to terminal to log in
- Possession of the token is necessary to know the correct value for the current time
- Each code computed for specific time window
- Codes from adjacent time windows are accepted
- Clocks must be synchronized
- Examples: BankID and SecurID



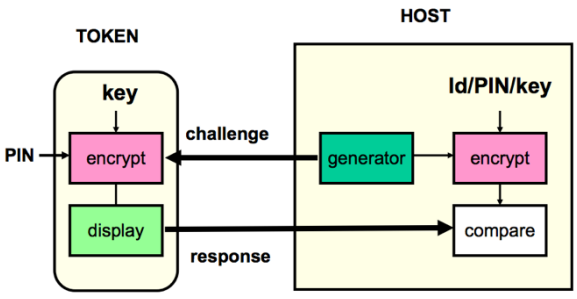
Clock-Based **Counter-based**

Counter-based OTP tokens: Overview

- Counter-based tokens generate a 'password' result value as a function of an internal counter and other internal data, without external inputs.
- HOTP is a HMAC-Based One-Time Password Algorithm described in RFC 4226 http://www.rfc-archive.org/getrfc.php?rfc=4226
    - Tokens that do not support any numeric input
    - The value displayed on the token is designed to be easily read and entered by the user.

Token-based User Authentication: Challenge Response Systems

- A challenge is sent in response to access request
    - A legitimate user can respond to the challenge by performing a task which requires use of information only available to the user (and possibly the host)
- User sends the response to the host
    - Access is approved if response is as expected by host.
- Advantage: since the challenge will be different each time, the response will be too – the dialogue can not be captured and used at a later time.
- Could use symmetric and asymmetric crypto.



Biometrics: Overview

What is it?

- Automated methods of verifying or recognizing a person based upon a physiological characteristic.
- Biometric modalities, examples: fingerprint, facial recognition, eye retina/iris scanning, hand geometry, written signature, voice print, keystroke dynamics.

Biometric: Requirements

- ***Universality***: Each person should have the characteristic;
- ***Distinctiveness:***
    - Any two persons should be sufficiently different in terms of the characteristic;
- ***Permanence:***
    - The characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time;
- ***Collectability:*** The characteristic should be measureable quantitatively.

Biometrics: Practical considerations

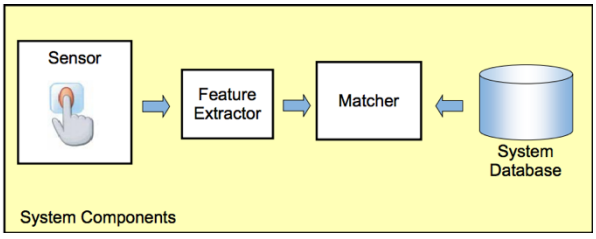
- ***Accuracy***:
  - The correctness of a biometric system, expressed as ERR (Equal Error Rate), where a low ERR is desirable.
- ***Performance:***
  - The achievable speed of analysis,
  - The resources required to achieve the desired speed.
- ***Acceptability:***
  - The extent to which people are willing to accept the use of a particular biometric identifier (characteristic)
- ***Circumvention resistance:*** The difficulty of fooling the biometric system
- ***Safety:*** whether the biometric system is safe to use

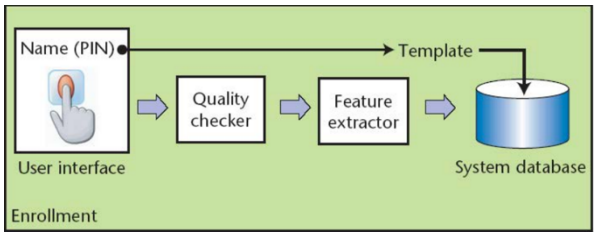
Biometrics Safety

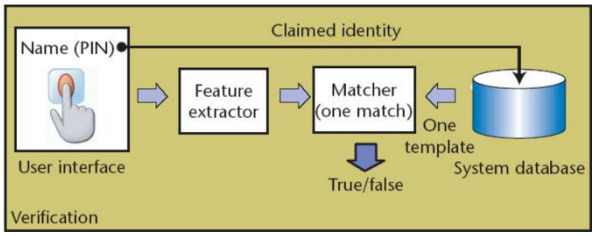
- Biometric authentication can be safety risk
  - Attackers might to "steal" body parts
  - Subjects can be put under duress to produce biometric authenticator
- Necessary to consider the physical environment where biometric authentication takes place.

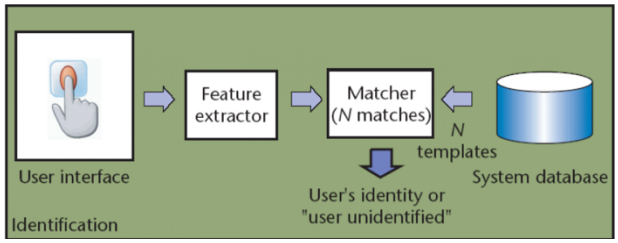
Biometrics Modes of operation

- ***Enrolment:***
  - Analog capture of the user's biometric attribute.
  - Processing of this captured data to develop a template of the user's attribute which is stored for later use.
- ***Identification*** (1: N, one-to-many)
  - capture of a new biometric sample.
  - search the database of stored templates for a match based solely on the biometric.
- ***Verification*** of claimed identity (1: 1, one-to-one):
  - capture of a new biometric sample.
  - comparison of the new sample with that of the user's stored template.









Evaluating Biometrics:

- Features from captured sample are compared against those of the stored template sample
- Score s is derived from comparison (better match leads to higher score)
- The system decision is tuned by threshold T:
    - System gives a **match** (same person) when the sample comparison generates a score s where $s \geq T$
    - System gives **non-match** (different person) when the sample comparison generates a score s where $s < T$
- **System Errors:**
    - Comparing biometric samples produces score s
    - Acceptance threshold T determines FMR and FNMR
        - If T is set low to make the system more tolerant to input variations and noise, then FMR increases.
        - On the other hand, if T is set high to make the system more secure, then FNMR increases accordingly
    - ERR (Equal Error Rate) is the rate when FMR = FNMR (low EER is good)

Matching algorithm characteristics

- True positive (User's sample matches → User is accepted)
- True negative (Attacker's sample does not match → Attacker is rejected
- False positives (Attacker's sample matches → Attacker is accepted
- False negatives (User's sample does not match → User is rejected
- False Match Rate and False Non-Match Rate
    - FMR = (# matching attacker samples) / (total # attacker samples)
    - FNMR = (# non-matching user samples) / (total # user samples)
- T determines tradeoff between FMR and FNMR

Authentication: Multi-factor

- Multi-factor authentication aims to combine two or more authentication techniques in order to provide stronger authentication assurance.
- Two-factor authentication is typically based on something a user knows (factor one) plus something the user has (factor two).
    - Usually this involves combining the use of a password and a token
    - Example: BankID OTP token with PIN + static password

Authentication Assurance

- Authentication assurance = robustness of authentication
- Resources have different sensitivity levels
    - High sensitivity gives high risk in case of authentication failure
- Authentication has a cost (Unnecessary authentication assurance is a waste of money)
- Authentication assurance should balance resource sensitivity

e-Authentication Frameworks for e-Government

- Trust in identity is a requirement for e-Government
- Authentication assurance produces identity trust.
- Authentication depends on technology, policy, standards, practice, awareness and regulation.
- Consistent frameworks allow cross-national and cross-organizational schemes that enable convenience, efficiency and cost savings.

Authentication Framework	*User Authentication Assurance Levels*				
OMB / NIST USA 2004 / 2011	Little or no assurance (1)		Some (2)	High (3)	Very High (4)
RAU / FAD Norway 2008	Little or no assurance (1)		Low (2)	Moderate (3)	High (4)
STORK QAA EU 2009	No or minimal (1)		Low (2)	Substantial (3)	High (4)
NeAF Australia 2009	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
e-Pramaan India 2012	None (0)	Minimal (1)	Minor (2)	Significant (3)	Substantial (4)
ISO 29115 ISO/IEC 2013	Low (Little or no) (1)		Medium (2)	High (3)	Very High (4)

UAAL: User Authentication Assurance Level

- UAAL is determined by the weakest of three links:

- Requirements for correct registration:
  - Pre-authentication credentials, example
    - Birth certificate
    - Biometrics
- Requirements for secure handling of credentials:
  - Creation
  - Distribution
  - Storage
- Requirements for mechanism strength:
  - Password length and quality
  - Cryptographic algorithm strength
  - Tamper resistance of token
  - Multiple-factor methods

User Identity Registration Assurance (UIRA) requirements

User Credential Management Assurance (UCMA) requirements

User Authentication Method Strength (UAMS) requirements

UAAL: User Authentication Assurance Levels

No Assurance	Minimal Assurance	Low Assurance	Moderate Assurance	High Assurance
Level 0	Level 1	Level 2	Level 3	Level 4
No registration of identity required	Minimal confidence in the identity assertion	Low confidence in the identity assertion	Moderate confidence in the identity assertion	High confidence in the identity assertion

Risk Analysis for Authentication

Determining the appropriate UAAL for an application

		Impact of e-Authentication failure				
		Insignificant	**Minor**	**Moderate**	**Major**	**Severe**
Likelihood	**Almost Certain**	None (0)	Low (2)	Moderate (3)	High (4)	High (4)
	Likely	None (0)	Low (2)	Moderate (3)	High (4)	High (4)
	Possible	None (0)	Minimal (1)	Low (2)	Moderate (3)	High (4)
	Unlikely	None (0)	Minimal (1)	Low (2)	Moderate (3)	Moderate (3)
	Rare	None (0)	Minimal (1)	Low (2)	Moderate (3)	Moderate (3)

RAU Norway (Rammeverk for Autentisering og Uavviselighet)
Framework for Authentication and Non-Repudiation

RAU Level 1: Little or no Authentication assurance
Alternative requirements:

- Online self-registration and self-chosen password
- Pre-authentication by providing person number

RAU Level 2: Low authentication assurance
Alternative requirements:

- Fixed password provisioned in person or by mail to user's address in national person register
- OPT calculator without PIN, provisioned in person or by mail to address in national person register.
- Lis of OTP (one-time passwords) provisioned in person or by mail to address in national person register.

RAU Level 3:
Alternative requirements:

- OTP calculator with PIN provisioned separately in person or by mail to address in national person register.
- SMS-based authentication, where enrolment of mobile phone is based on code provisioned in person or by mail to address in national person register.
- Personal public-key certificate with government, PKI
- List of OTP (one-time passwords) combined with static password and username provisioned in person or by mail to address in national person register.

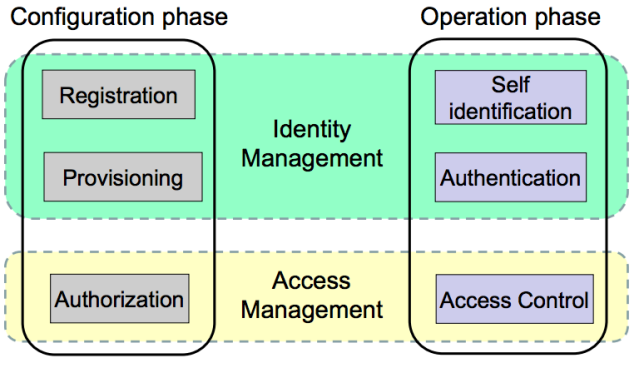
Example: MinID

RAU Level 4: High authentication assurance
Alternative requirements:

- Two-factor, where at least one must be dynamic, and at least one is provisioned in person. The other by mail to address in national person register. Also requires logging and auditing by third party. **Examples:** Buypass, Confides, BankID

Week 9 – Identity Management and Access Control

Concepts related to identity



- Entity
  - A person, organization, agent, system, etc.
- Identity ("Same one as last time")
  - A set of names/attributes of entity in a specific domain
  - An entity may have identities in multiple domains
  - An entity may have multiple identities in one domain
- Digital identity
  - Digital representation of names/attributes in a way that is suitable for processing by computers
- Names and attributes of entity
  - Can be unique or ambiguous within a domain

    - Transient or permanent, self defined or by authority, interpretation by humans and/or computer, etc.

Identity

- "First-time" authentication is not meaningful
    - because there is no "previous time"
- Authentication requires a first time registration of identity in the form of a name within a domain
- Registration can be take two forms:
    - Pre-authentication, from previous identity, e.g. passport
    - Creation of new identity, e.g. New born baby