Introduction to Security

Avinanta Tarigan

Universitas Gunadarma

Layout

- Problems
- General Security
- Cryptography & Protocol reviewed

Problems

- Life was beautiful before computer, getting worse after Internet
- Distributed systems: each depends on others
- How can we assure system behaves correctly & securely ?
- Can we trust systems on the other side ?

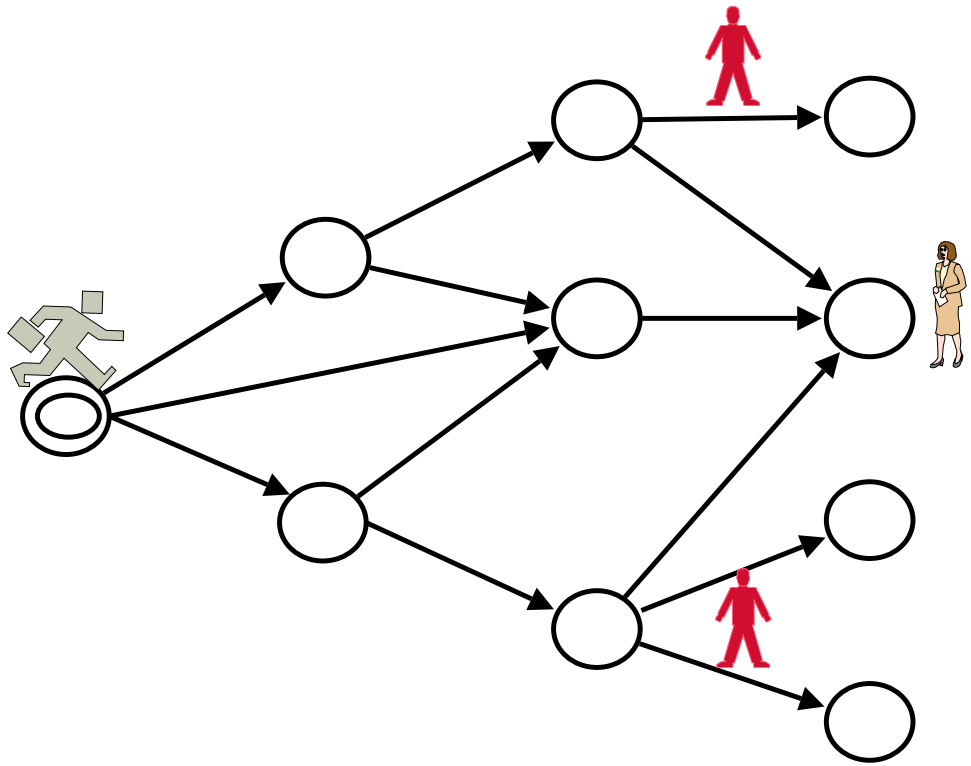
Problems .. (cont)

- How to assure security of the network
- How to quantify risk
- What are the boundaries of the system ?
- Relationship to political, social, enomical aspects is not well understood
- Uncertainty

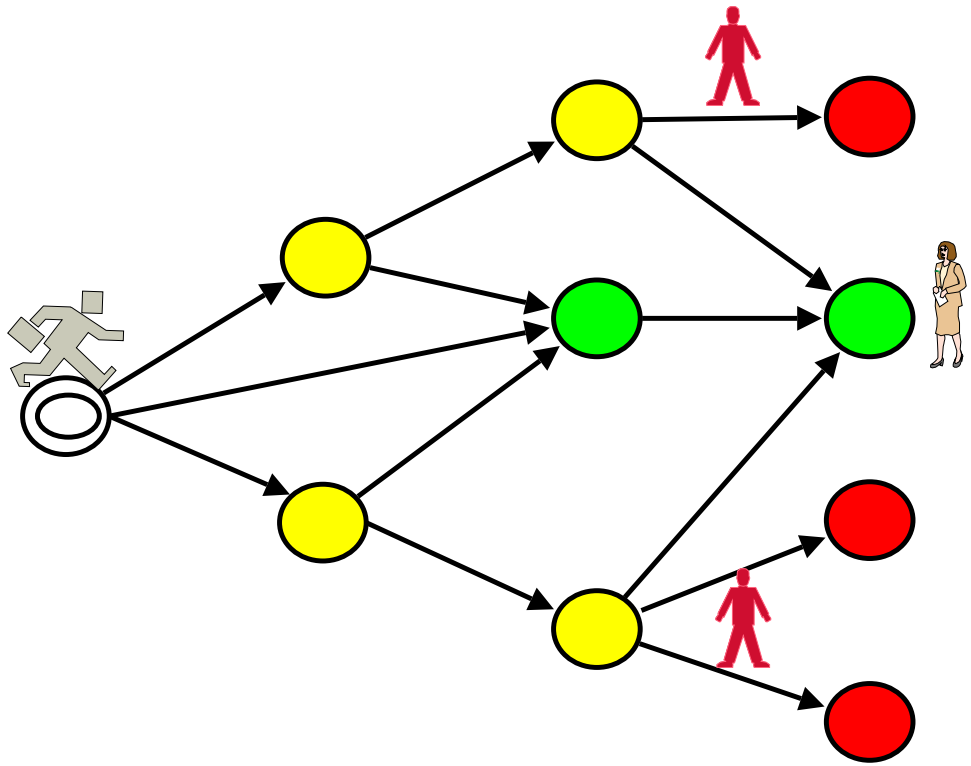
Definition

- In General :
  - Computer Security deals with the method against unauthorized actions in computer systems
- More General :
  - Dependability to other institution is also insecurity

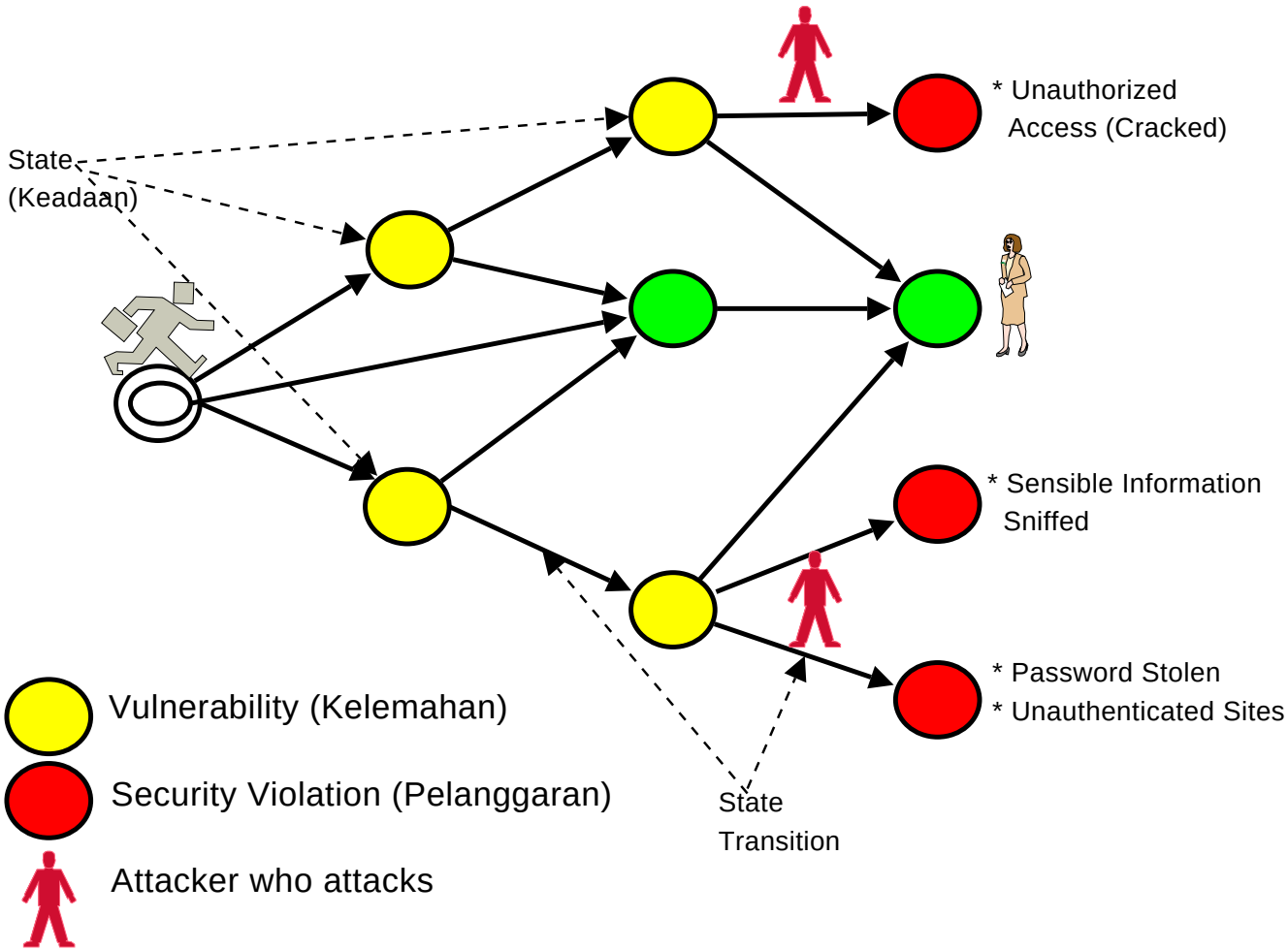
Basic



Basic



Basic


State
(Keadaan)
* Unauthorized
Access (Cracked)
* Sensible Information
Sniffed
* Password Stolen
* Unauthenticated Sites
State
Transition
Vulnerability (Kelemahan)
Security Violation (Pelanggaran)
Attacker who attacks

What is secure ?

- Computer based system behaves according to
  1. algorithms (program/software)
  2. user direction (input)
- Given systems & environment:
  1. secure states (system maintains security properties)
  2. insecure states (violation of security policy)
  3. paths lead to insecure states (vulnerability)
- Correctness: maintain intended behavior according to correct specification while unintended behavior will not be reachable
- Security policy: definition of (1,2,3)
- Attack : **Persistence**, **Intentional**, Outsider vs Insider, Loss vs Gain

Threats ... (cont)

- Towards vulnerable system [Abadi] :
  - Interaction with uncertain physical, network, software environment.
  - Using public network, distributed administration, diverse operators
  - COTS, business demand is the priority, Monocultures

Threats ... (cont)

- Attack :
  - Physical attack
    ex. theft of harddisk/cdroms, bombing, etc.
  - Syntatic attack
    ex. buffer overflow, domain theft, SQL injection
  - Semantic attack
    ex. Social engineering, site phising

Security Policy

- We define secure states and insecure states
- Define paths which always bring system in secure states: what is permited
- Define also paths might lead to insecure states: what is not allowed
- Specified in formal language for clearness, unambiguity, consistency, and verificability
- Written in natural language for better understanding

Properties

- Confidentiality (Secrecy)
  Unauthorized disclosure of information is not reachable
  (Acces Control - Cryptography)
- Integrity
  Unauthorized modification of information is not reachable
  - Data integrity - Origin integrity / authentication
  - Prevention (access control) - Detection (hash function)

Properties ... (cont)

- Availability
  Prevention & detection of denial of service
- Accountability
  The availablity and completeness of the track of past system states
  Who - Whom - What - When - Where
  Implementation: should be forensic ready

Goals

- Prevention
  to fail the attack
- Detection
  to detect unprevented attack
- Recovery
  to stop the attack & repair attacked system

Security Mechanism

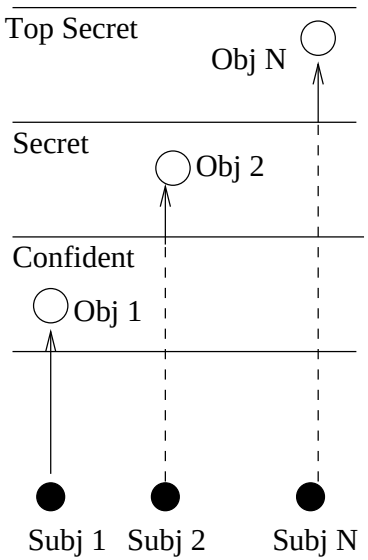
- Way to enforce security policies
- How to limit system behavior according to policies
- Specification - Design - Implementation - Operation & Maintenance - Audit (Forensic)
- Access Control & Cryptography

Access Control

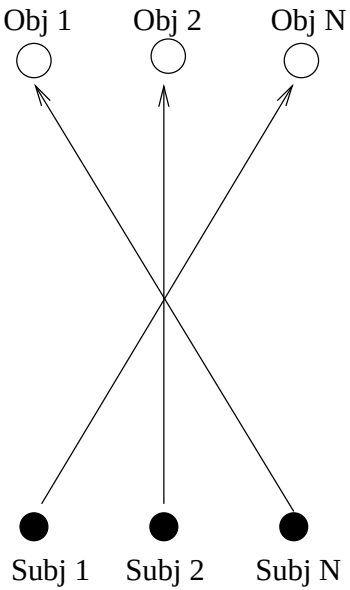
- Reference Monitor
- Set of precise rules according to security policy, applied as a filter to the transition states of the system, which prevents system in entering insecure state
- Authentication is mandatory
- **Subjects**, **Objects**, **Actions**, Time, Location, and other attributes

Model of Acces Control

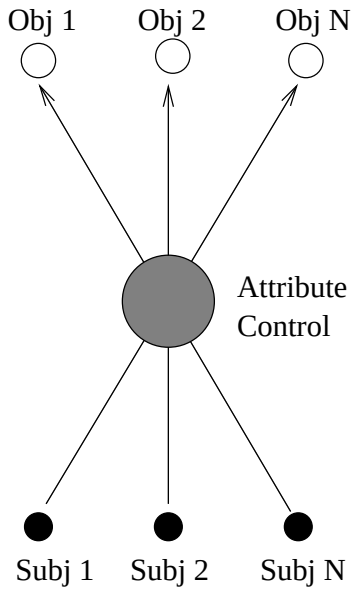
Mandatory Access Control (MAC)



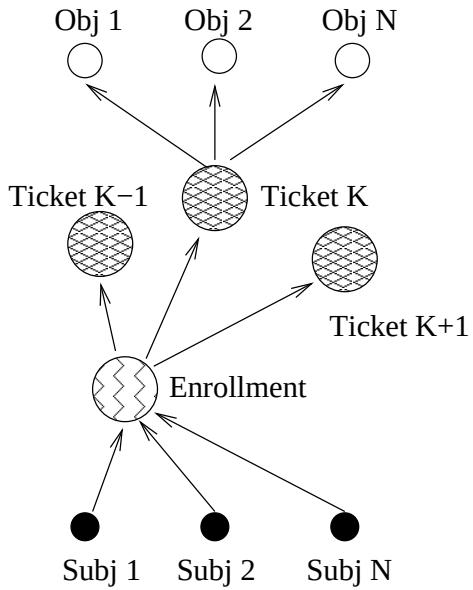
Discretionary Access Control (DAC)



Role Based Access Control (RBAC)



Safe Dealing (SD)



Implementation: Cryptography

- Algorithm to protect *secrecy* of data
- Also used to gain :
  - *authentication*
  - *integrity*
  - *non repudiation*
- Includes : *algorithm* and *key(s)*

Cryptography ... (cont)

- $Chipertext = Encrypt(Message, Key)$
- $Message = Decrypt(Chipertext, Key)$
- $Decrypt(Chipertext)$ hard without Key
- Research questions :
  Is there any algorithm which is hard to compute original message but easy to verify it
- In implementation requires a protocol (*Cryptographic Protocol*)

Symmetric Crypt.

$$\mathscr{A} \mapsto \mathscr{B} : \{\mathscr{M}\}_{\mathscr{K}_{ab}}$$

Principal $\mathscr{A}$ sends $\mathscr{B}$ message $\mathscr{M}$ *encrypted* with *shared-key* $\mathscr{K}_{ab}$

- Key is shared between 2 *principals*
- Needs N^2 keys for N principals
- Fast but *key management* is not easy
- Example of Chiper: DES, 3DES, Blowfish, AES

Asymmetric/Public Key Crypt.

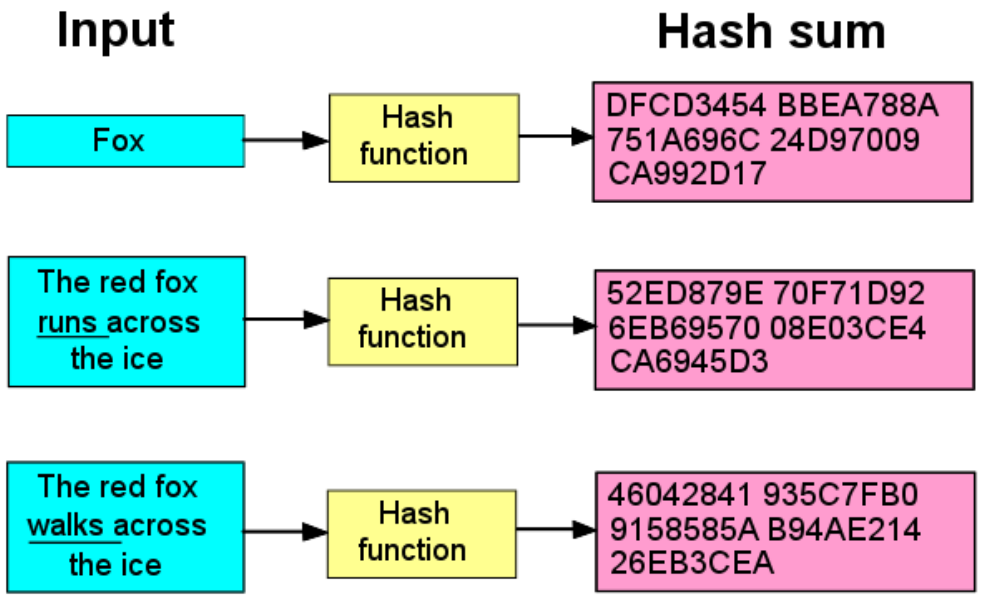
$$\mathscr{A} \mapsto \mathscr{B} : \{\mathscr{M}\}_{\mathscr{K}_b}$$

Principal $\mathscr{A}$ sends $\mathscr{B}$ message $\mathscr{M}$
***encrypted* with $\mathscr{B}$'s *public-key* $\mathscr{K}_b$**

- Only with *private-key* $\mathscr{K}_b^{-1}$, $\mathscr{B}$ can *decrypt* $\mathscr{M}$
- Principal has its own $\mathscr{K}$ which is *published* and $\mathscr{K}^{-1}$ which must be keeped *secret*
- Key management is less difficult, usualy managed by *Certification Authority*
- Example of Chiper: RSA (Rivest-Shamir-Addleman), Elliptic-Curve

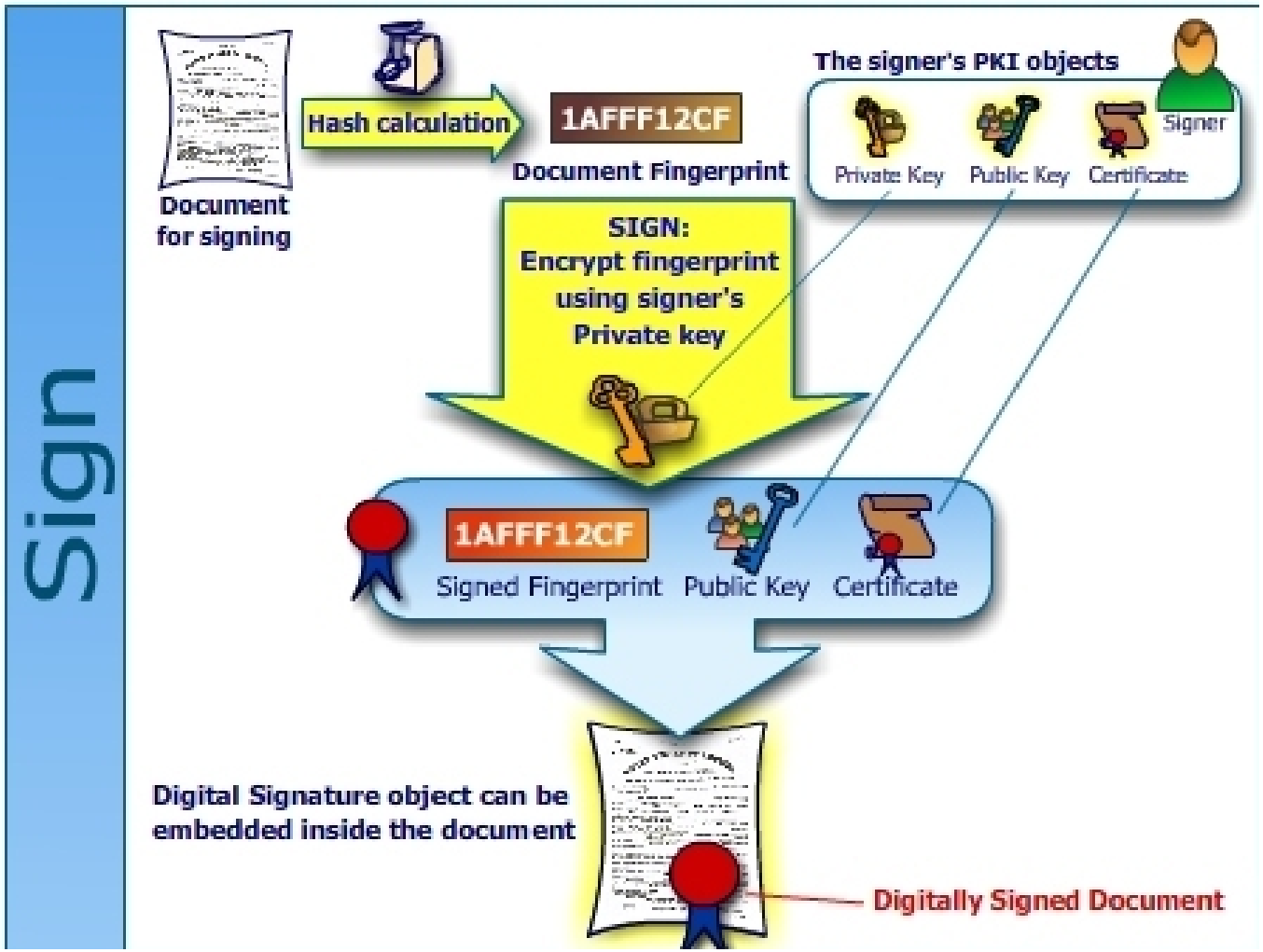
One-Way-Hash

- Algorithm to compute large data into small integer, producing *fingerprint of the message*

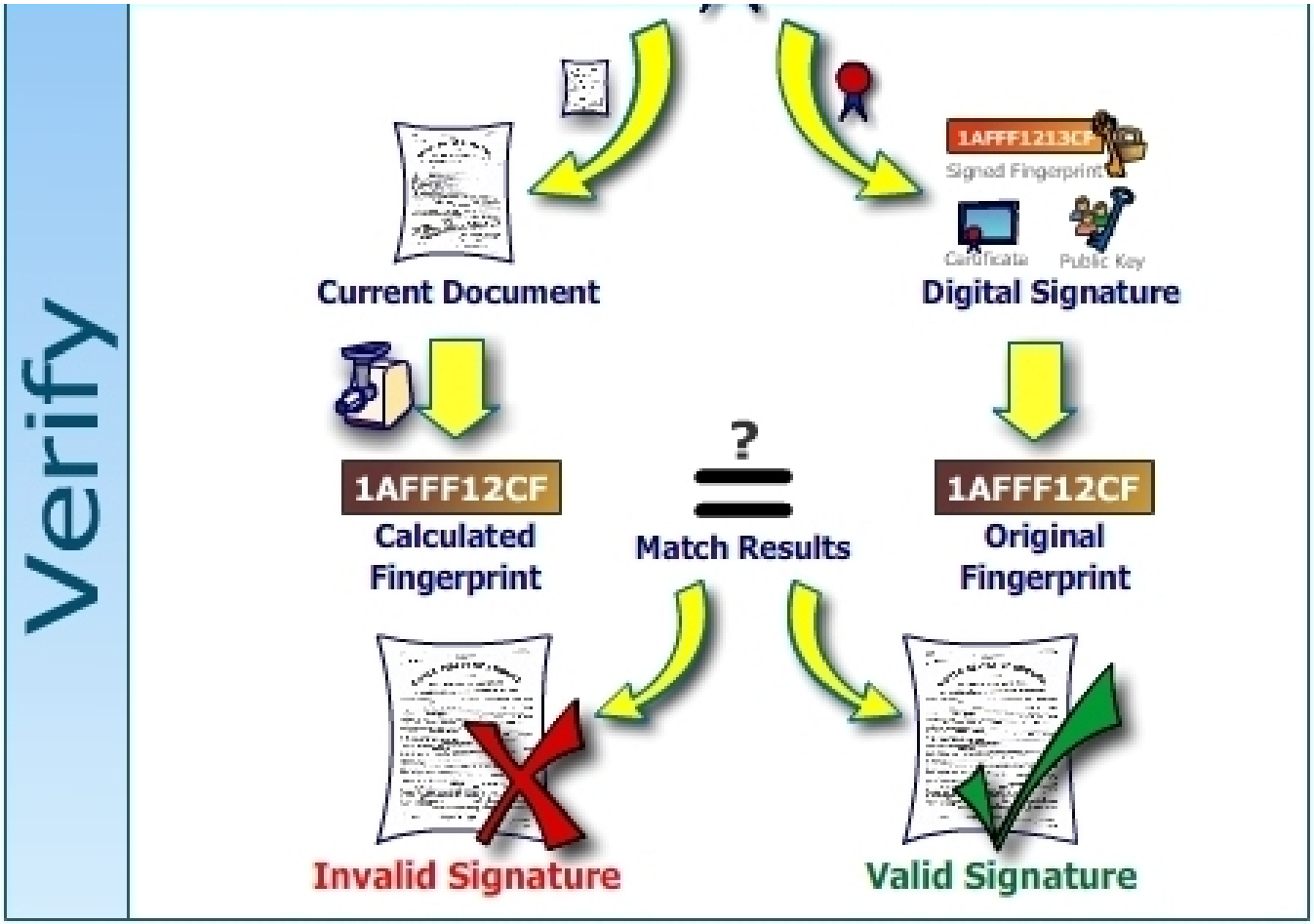


- Used for maintaining **integrity** of message being transferred
- Example: MD5, SHA1, SHA-256, Ripemd, Haval

Digital Signature (Sign)



Digital Signature (Verify)



Digital Signature

$$\mathscr{A} \mapsto \mathscr{B} : \{\mathscr{M}, \{Hash(\mathscr{M})\}_{\mathscr{K}_a}\}_{\mathscr{K}_b}$$

- *A*'s digital signature on a message is the hash of message encrypted with *A*'s private-key
- Authentication: only with *A*'s public-key, the hash can be decrypted
- Integrity: Hash function
- Confidentiality: message can be decrypted only with *B*'s private-key
- Non-Repudiation: explain for your self

Pictures of Cryptographer

Cryptographic Protocol

- Implementation of Cryptography Algorithm
- Achieving *security properties* (authentication, secrecy, etc.)
- Example :
  - **Needham-Schroede**r (*authentication*)
  - **Kerberos** (*authentication*)
  - **SSL/TLS** (*auth - secrecy*)

Cryptographic Protocol

- Example : **Needham-Schroeder Protocol**

M1 $\mathscr{A} \mapsto \mathscr{S} : A, B, N_a$

M2 $\mathscr{S} \mapsto \mathscr{A} : \{N_a, B, K_{ab}, \{K_{ab}, A\}K_{bs}\}K_{as}$

M3 $\mathscr{A} \mapsto \mathscr{B} : \{K_{ab}, A\}K_{bs}$

M4 $\mathscr{B} \mapsto \mathscr{A} : \{N_b\}K_{ab}$

M5 $\mathscr{A} \mapsto \mathscr{B} : \{N_b - 1\}K_{ab}$

- Intoducing Nonce (N)

Cryptographic Protocol

- More example : **Kerberos Protocol**

M1 $\mathscr{A} \mapsto \mathscr{S} : A, B$

M2 $\mathscr{S} \mapsto \mathscr{A} : \{T_s, L, B, K_{ab}, \{T_s, L, K_{ab}, A\}K_{bs}\}K_{as}$

M3 $\mathscr{A} \mapsto \mathscr{B} : \{T_s, L, K_{ab}, A\}K_{bs}, \{A, T_a\}K_{ab}$

M4 $\mathscr{B} \mapsto \mathscr{A} : \{T_a + 1\}K_{ab}$

- Introducing TimeStamp (T) and Lifetime (L)
- Used in many system, including Windows

Cryptographic Protocol

Problem :

- Wrong design could lead to flaw
  - Needham-Schroeder Protocol
  - SSLv1.0
- Wrong implementation could lead to vulnerability
  - Padding problem in SSL, SSH, and WTLS
  - User Interface design in Browser
- Vulnerability arise between two protection technologies (Anderson, Ross)

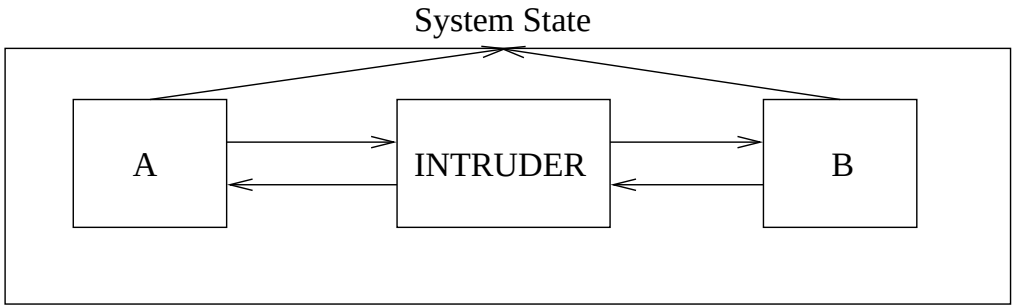
Assurance : Formal Method

- To prove correctness in achieving security properties which protocol carry out
- There are two development approach :
  - Extention from method used in communication
  - Newly developed method
- Four classifications :

1. General purpose tools
2. Logic based
3. Expert System
4. Algebraic approach

Formal Method
Using General Purpose Tools

- Treated as ordinary comm. protocol
- Adversary is explicit, capable in read, intercept, and modify messages
- Method : FSM, CSP, FDR, Petri Nets
  Example : Lotos, Ina Jo, Murphy



Formal Method
Using Expert System

- Investigate every possible scenario of ***Attack - Flaw - Defence***
- Needs to define **insecure states** and *search* paths to them
- More successful than General Purpose Tools
- Example : Interrogator by Millen, NRL Protocol Analyzer by Meadows, Longley and Rigby

Formal Method
Algebraic Approach

- Capabilties in **modeling knowledge** which represents component in cryptographic operation (*Nonce, Key(s), and old messages*)
- Example :
  - Dolev - Yao (term re-writing systems)
  - Sphi - Calculus by Abadi and Gordon (to prove secrecy)

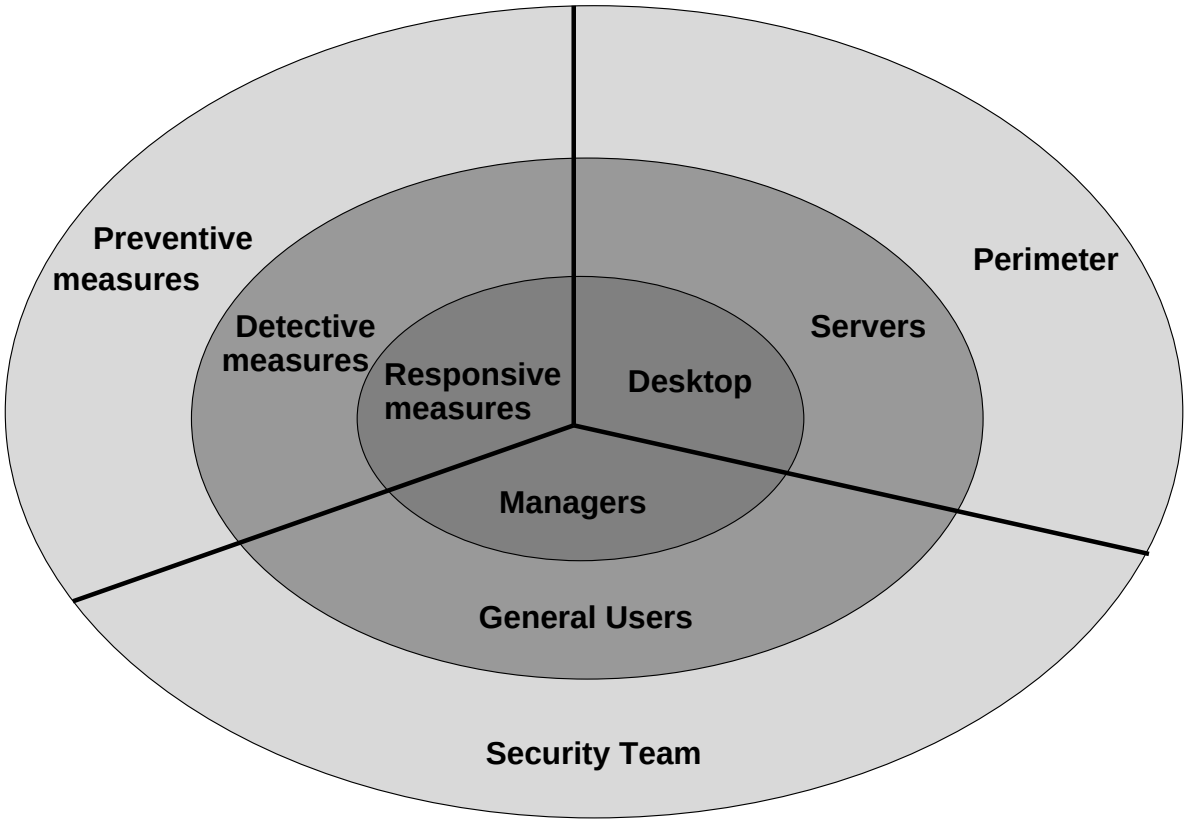
Formal Method
Logic Based

- One sees crypt. protocol as **distributed algorithm**
- Develop logics from **modal logic**
- There are **inference rules**
- Goal is to derived statements which **represents correct condition**
- Example : *BAN Logic and GNY Logic*

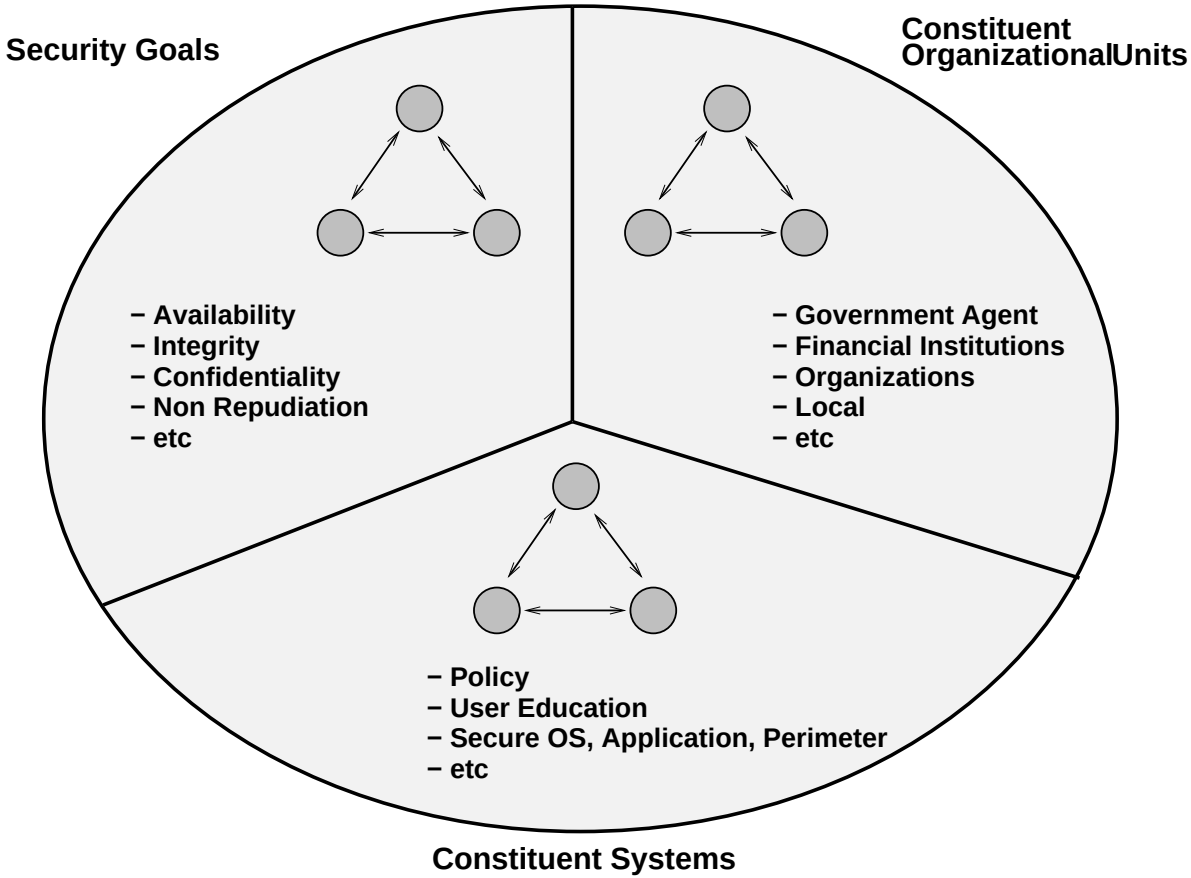
Towards Secure System

- Specification : Security Policy
- Implementation : Security Mechanism
- Correctness : Assurance
- Man - Machine - Management

Towards ... (cont)



Towards .. (cont)



Books, Papers, and Links

- Ross Anderson, “Security Engineering”
- Matt Bishop, “Computer Security”
- Schneider et. al. “Modelling and Analysis of Security Protocols”
- Martin Abadi’s homepage at *http://www.cse.ucsc.edu/~abadi*

The End

End of this presentation