



Networked System Security

## Introduction to Security

Module TA: Stylianos Gisdakis, [gisdakis@kth.se](mailto:gisdakis@kth.se)

**Panos Papadimitratos**  
Networked Systems Security Group  
<http://www.ee.kth.se/nss>

---

---

---

---

---

---

---



## Outline

- Introduction
  - Adversary and attacks
  - Security goals



- Cryptographic tools
  - Basic tools
    - Hash functions
    - Encryption
  - Using the tools
    - Message Integrity Codes
    - Digital Signature



2014-11-06 EP2500 Networked Systems Security

2/43

---

---

---

---

---

---

---



## Why security?



- Alice wants to communicate with Bob
- What could possibly go wrong?
- Threats
  - Passive attacks
  - Active attacks

2014-11-06 EP2500 Networked Systems Security

3/43

---

---

---

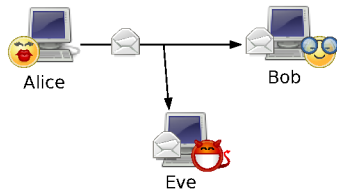
---

---

---

---

## Eavesdropping



- Wiretapping
- Wireless sniffing
- Interception

---

---

---

---

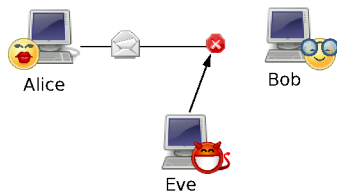
---

---

---

---

## Disruption



- Link Sabotage
- Jamming
- Interruption

---

---

---

---

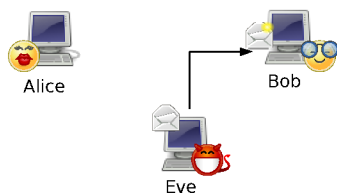
---

---

---

---

## Forging



- Impersonating
- Masquerading
- Spoofing
- Injecting

---

---

---

---

---

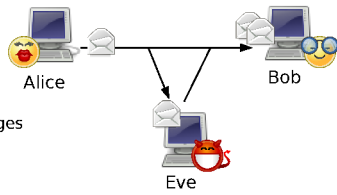
---

---

---



## Replaying



- Acquire legit messages
- Relay

2014-11-06

EP2500 Networked Systems Security

7/43

---

---

---

---

---

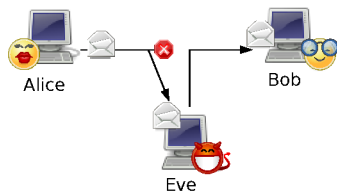
---

---

---



## Man-In-The-Middle



- Intercept messages
- Replay them
- Impersonate the source
- Make them believe they are talking directly to each other (hidden)

2014-11-06

EP2500 Networked Systems Security

8/43

---

---

---

---

---

---

---

---



## Security Goals

- **Confidentiality**
  - No one can **read** our data / communication *unless we want* them to
- **Integrity**
  - No one can **manipulate** our data / processing / communication *unless we want* them to
- **Availability**
  - We can **access** our data / conduct our processing / use our communication capabilities *when we want* to



src: berkeley

2014-11-06

EP2500 Networked Systems Security

9/43

---

---

---

---

---

---

---

---



## Security Goals (cont'd)

- Authentication
  - No one can **read** / **manipulate** our data
  - We can **access** it
  - **Demonstrate** who you are
- Authorization
  - “unless we want to”
  - Manage **rights** to perform a task
- Accountability (Auditing)
  - **Tracking** of the performed task
    - Billing
    - Analysis
    - Logging / Forensics



2014-11-06

EP2500 Networked Systems Security

10/43

---

---

---

---

---

---

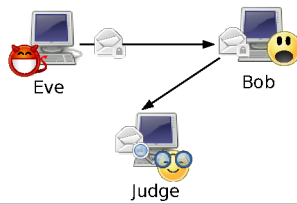
---

---



## Security Goals (cont'd)

- Non Repudiation
  - Nobody else can manipulate our data
  - We cannot **deny** the content



2014-11-06

EP2500 Networked Systems Security

11/43

---

---

---

---

---

---

---

---



## Authentication

- Link identity to actor
- Proof of identity by providing evidence of:
  - What I **know**
    - Secret password, challenge response
  - What I **have**
    - Smartcard, token
  - What I **am**
    - Fingerprint, retinal scan
  - **Where** I am
    - Geolocalization, which terminal



2014-11-06

EP2500 Networked Systems Security

12/43

---

---

---

---

---

---

---

---

## Authentication (cont'd)

- One system could be not enough
  - Weak passwords, stealing smartcard, fake fingerprints [1], etc.

**Strong authentication**

- Combining multiple methods
  - E.g. Password + phone
- Enhancing weak methods
  - E.g. **when** I am (weekdays) **plus where** (terminal at work)

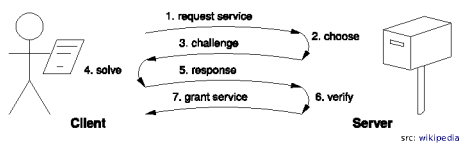


src: google

[1] Matsumoto, T. et al. "Impact of artificial gummy fingers on fingerprint systems". In proceedings of SPIE Vol. #4677, Optical Security and Counterfeit Deterrence Techniques IV, Jan. 2002.

## Authentication (cont'd)

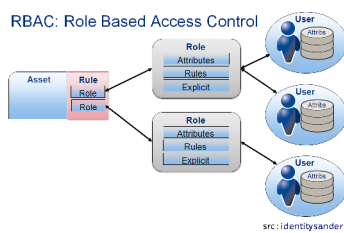
- Challenge Response**
  - Entity wants to access a service
  - Service provider presents a challenge
  - Entity must provide a valid answer



## Access Control



- Combines**
  - Authentication
  - Authorization
  - Monitor (audit)
  - Principle of least privilege

- Access Control List (ACL)**
  - List of permission
  - Attached to an object





## Cryptographic Tools

- Hash functions 
- Encryption 
- Information Theoretic Security
  - Security not depending on the hardness of computations
- Computational Security
  - Adversary has limited resources
  - Cryptography can be broken only after a very long time

2014-11-06

EP2500 Networked Systems Security

16/43

---

---

---

---

---

---

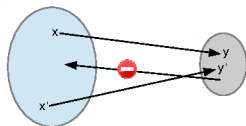
---

---



## Hash Function

- One way function  $\{0,1\}^* \rightarrow \{0,1\}^*$ 
  - Easy to compute
  - Hard to invert
  - E.g. Phone book



- Hash function  $\{0,1\}^* \rightarrow \{0,1\}^n$ 
  - One way function to a fixed length

- Pigeonhole principle (collision)
  - N items
  - M pigeonholes
  - $N > M$
  - At least one pigeonhole must contain more than one item



src: wikipedia

2014-11-06

EP2500 Networked Systems Security

17/43

---

---

---

---

---

---

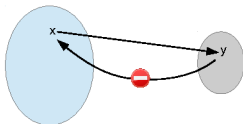
---

---



## Hash Functions: properties

- Preimage resistance
  - Given  $y$ , it is hard to find an  $x$ , such that  $h(x) = y$



2014-11-06

EP2500 Networked Systems Security

18/43

---

---

---

---

---

---

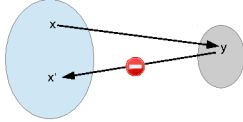
---

---

## Hash Functions: properties (cont'd)

### • Second-preimage resistance

Given  $x$  and  $y = h(x)$ , it is hard to find  $x' \neq x$ , such that  $h(x) = h(x')$




---

---

---

---

---

---

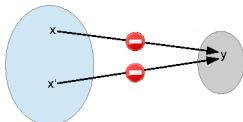
---

---

## Hash Functions: properties (cont'd)

### • Collision resistance:

It is hard to find any  $x, x'$  such that  $h(x) = h(x')$




---

---

---

---

---

---

---

---

## Hash Functions: properties (cont'd)

### • Avalanche effect

- Desirable property
- When an input changes **slightly**
- The output changes **significantly**

Input	Hash function	Hash sum
000	Hash function	8AEF806C 426E07A0 A671A1E2 48884858 D094A750
001	Hash function	E193A01E CF8D30AD 0AFFEFD3 32CE934E 32FFCE72
010	Hash function	47AB9979 443FB7ED 1C193D06 773333BA 7876D94F

src: wikipedia

---

---

---

---

---

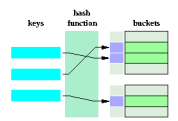
---

---

---

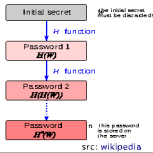
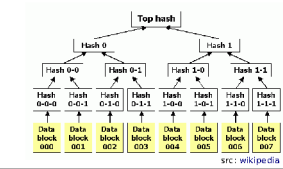
## Hash Functions: use cases

- Hash tables
- Hash chains
- Hash trees



src: [wikipedia](#)

### SkEY password generation



src: [wikiped](#)

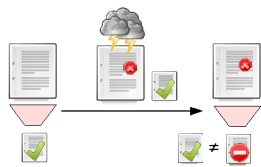
2014-11-06

EP2500 Networked Systems Security

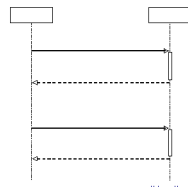
22/43

## Hash Functions: use cases (cont'd)

- Checksum



- Challenge Response Authentication Mechanism (CRAM)



src: [wikipedia](#)

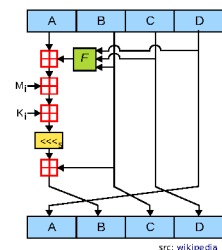
2014-11-06

EP2500 Networked Systems Security

23/43

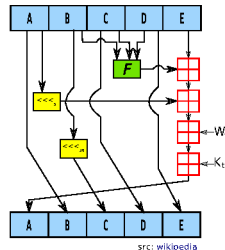
## Hash Functions: implementations

- Message-Digest v5 (MD5)



src: wikipedia

- Secure Hash Algorithm (SHA-1)



src: [wikiped](#)

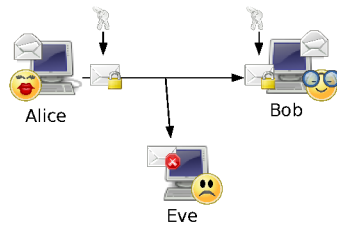
2014-11-06

EP2500 Networked Systems Security

24/43



## Encryption



- Kerckhoffs's principle  
The enemy **knows** the system
- The cryptographic secret or private keys must be kept secret

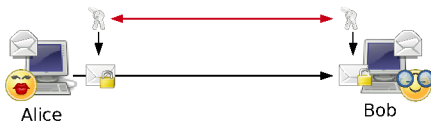
2014-11-06

EP2500 Networked Systems Security

25/43

## Symmetric Key

- Same key to encrypt and decrypt
- Computationally efficient



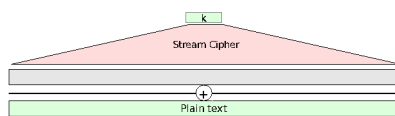
2014-11-06

EP2500 Networked Systems Security

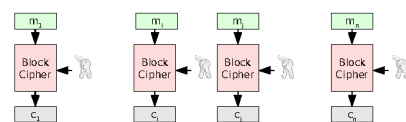
26/43

## Symmetric Key (cont'd)

- Stream ciphers
  - RC4
  - A5/1



- Block ciphers
  - DES
  - AES



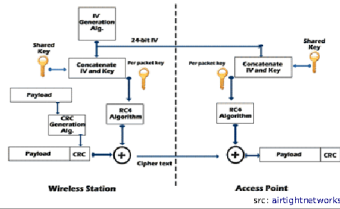
2014-11-06

EP2500 Networked Systems Security

27/43

## Examples: WiFi Security

- Wired Equivalent Privacy (WEP)
  - Provide confidentiality comparable to the wired connections
  - Has numerous flaws



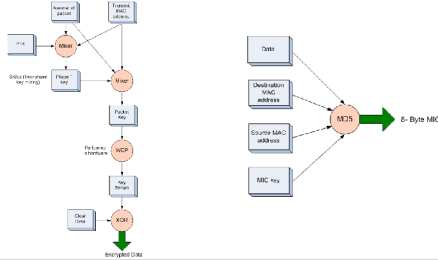
2014-11-06

EP2500 Networked Systems Security

28/43

## Examples: WiFi Security (cont'd)

- Wi-Fi Protected Access (WPA)
- Message Integrity Code



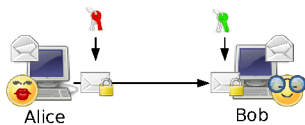
2014-11-06

EP2500 Networked Systems Security

29/43

## Asymmetric Key

- Different keys:
  - One for encryption (Public)
  - One for decryption (Private)
- Alice **encrypts** the message with Bob's **public** key
- Bob **decrypts** it with his **private** key
- Infeasible to figure out the keys
  - From the message
  - From the other key
- Computationally **less** efficient



2014-11-06

EP2500 Networked Systems Security

30/43

## Asymmetric Key (cont'd)

- Based on computationally (NP) hard problems
- Integer factorization
  - Given  $n$  as product of  $p$  and  $q$  primes, it is hard to find  $p$  and  $q$ .
  - RSA
- Discrete logarithm
  - Given  $g$  and  $y = g^x$  is hard to find  $x$  in modulo  $p$  prime
  - ElGamal

---

---

---

---

---

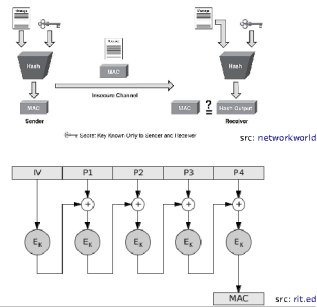
---

---

---

## Message Authentication

- HMAC
  - Using hash functions
- CBC-MAC
  - Using symmetric encryption




---

---

---

---

---

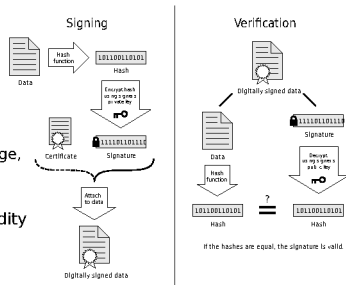
---

---

---

## Message Authentication (cont'd)

- Digital Signature
  - Use the **private** key to **sign** the message
  - Use the **public** key to **verify** the message
  - Instead of the whole message, sign only the **hash**
- How can you tell the validity of the signature?
  - Use **certificates**




---

---

---

---

---

---

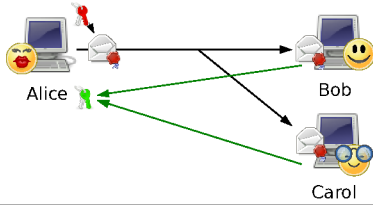
---

---

## Digital Signature: properties 1/3

### Publicly verifiable

- You cannot verify a MAC without know the shared key
- Anybody with the Public Key can verify the signature




---

---

---

---

---

---

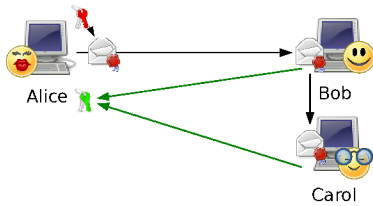
---

---

## Digital Signature: properties 2/3

### Transferable

- The signature travels with the message
- A third entity can always authenticate the original message




---

---

---

---

---

---

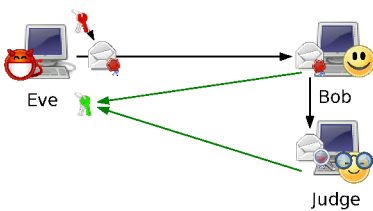
---

---

## Digital Signature: properties 3/3

### Non-repudiation

- Nobody else can manipulate the message and produce a valid signature without knowing the Private Key




---

---

---

---

---

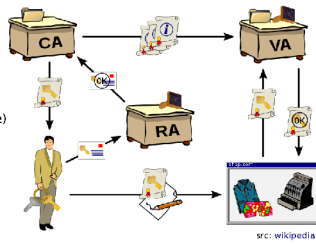
---

---

---

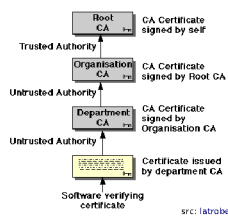
## Public Key Infrastructure (PKI)

- **Certificate Authority (CA)**
  - Trusted Third Party
  - Public Key is known
  - CertCA is a **signature** on:
    - The identity
    - Its public Key
    - Other information (E.g. lifetime)
- **Registration Authority (RA):**
  - Verifies the identity
- **Validation Authority (VA):**
  - Validates the public key

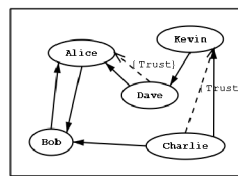


## Public Key Infrastructure (PKI) (cont'd)

### • Certificate Chain



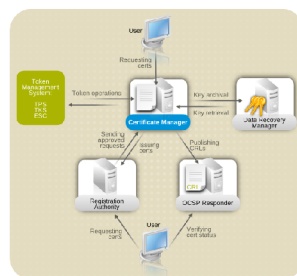
### • Web of Trust (PGP)



An example of the web of trust model

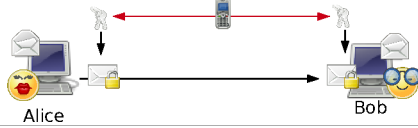
## Public Key Infrastructure (PKI) (cont'd)

- **Certificate Revocation**
  - Certificates can cease to be valid
    - Device or user is evicted
    - Cryptographic key is compromised
- **Certificate Revocation**
  - List of revoked certificates (CRL)
    - Δ-CRL (Incremental CRL)
  - Online Certificate Status Protocol



## Session Key

- Shared key
  - Already agreed on
  - Communicated over a secure channel
  - Usually symmetric
    - less computational overhead
  - Used only for the session
    - reduce cryptanalytic attack risks




---

---

---

---

---

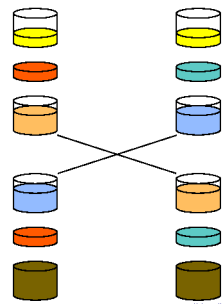
---

---

---

## Key Agreement

- Diffie-Hellman key exchange
  - Based on the discrete logarithm problem
  - Alice sends  $g^x$  and receives  $g^y$  from Bob
  - Both can evaluate  $g^{xy}$
  - Vulnerable to Man-in-the-Middle




---

---

---

---

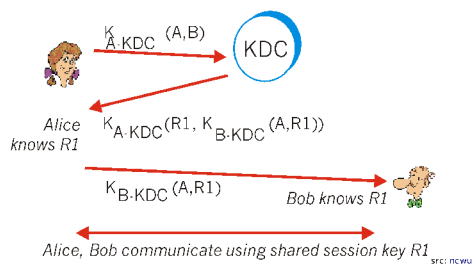
---

---

---

---

## Key Distribution Center




---

---

---

---

---

---

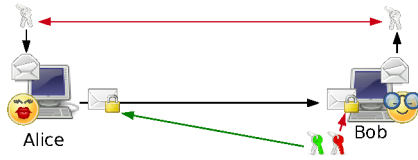
---

---



## Transport Keys

- Use the asymmetric key schema for exchanging the session key
  - Public key scheme's high computational cost used only for agreement
  - Authentication of both ends could be provided by certificates



2014-11-06

EP2500 Networked Systems Security

43/43