1. Confidentiality

Sikkerhedsmål

- Kort intro til confidentiality, authenticity og availability
- Lille eksempel: når man sender passwords til server

Kryptosystemer

- Tegn den flotte tegning

Unconditional vs. computational

- Unconditional: One time pad
- Praktiske formål

Secret key systemer

- Begge parter har en delt nøgle
- Stream ciphers
  - Kan kryptere alle længder beskeder
  - Producerer lang bitstreng udfra key og nonce
  - Svær at lave effektiv og sikre
  - Alle block ciphers kan bruges som stream ciphers
- Block ciphers
  - Bryder beskeder op i blokke af fixed length
  - Modes of operation (CBC, CTR og OFB)
    - Benytter nonce så ens beskeder ikke har samme ciphertext
    - CBC (tegn tegning og forklar)
    - Med CBC kan authenticity automatisk enforces med CBC-MAC
    - CTR kan beregnes parallelt, samme encrypt og decrypt funktion
  - Eksempler er AES og DES
- Et par ord om exhaustive search inden overgang

Public key systemer

- Egenskaber ved public key
  - Con: Langsommere end secret key
  - Pro: Der behøver ikke være en delt hemmelig nøgle på plads
- Tegn “Cryptosystems” diagram
  - Nøgler genereres på egen maskine og public key deles
- RSA
  - Key generation pk = e,n og sk = d,n
  - Factoring problem, derfor længere keys end secret key
- Key enveloping

2. Authentication

Sikkerhedsmål

Kort intro til confidentiality, authenticity og availability

Lille eksempel; pengeoverførsel - man-in-the-middle ændrer til hans konto (data integrity);

B udgiver sig for at være A og siger “overfør x til B’s konto” (identity spoofing)

Tegn tegning med generering af keys og de to algoritmer

Unconditional vs. computational

Unconditional: Tabel med beskeder og tilhørende MAC → tegn!

Praktiske formål

Secret key systemer

To algoritmer S (signer besked, laver MAC) og V (verificerer MAC)

CBC-MAC - den sidste block i CBC kryptering bruges som MAC

Public key systemer

Egenskaber ved public key

Con: Langsommere end secret key

Pro: Der behøver ikke være en delt hemmelig nøgle på plads

Alle public key kryptosystemer kan bruges til authenticity

RSA eksempel (omvendt af kryptering)

Hash funktioner

Egenskaber: Tag vilkårligt input, output fast længde, hurtig, svært at lave kollisioner

Signatur af hashet besked er det samme som signatur af besked

Dette fikser tidsproblem ved public key

Digital signatur **(validering)**

Public key systemer kan bruges pga. non repudiation

Rapporter nøgle stjålet → Ingen dokumenter er gyldige

Løsning: ekstern timestamp af dokumenter

Replay attacks

Modtager skal huske alle sekvensnumre

Benyt timestamps, kan være svært på upålidelig forbindelse

Interaction: Modtager sender nonce, afsender authenticater besked + nonce

3. Key management and Infrastructures

Standard 2-party secret key kommunikation

- Fælles key til kryptering af session keys
- Ved mange brugere kommer der for mange nøgler

Key Distribution Centers (KDC)

- Kender alle secret keys
- Kræver brugeres fulde tillid
- Single point of failure

Certification Authorities (CA)

- Hvad er et certifikat $Cert(ID_A, pk_A, sign_{sk\_CA}(ID_A, pk_A))$
- Brugere skal have tillid til at disse ikke udsteder falske certifikater
  - Brugere skal verificeres fysisk
- Brugere kan kommunikere efter at have udvekslet certifikater
- Certificate chaining (ikke stærkere end det svageste led)
- CA'ers certifikater kommer med browseren (kræver tillid til browserfabrikanten)

Access Control

Password sikkerhed

- How password is chosen (passphrases)
- How password is transmitted (eks. FTP og eavesdropping)
- How password is stored by the user (looking over the shoulder, social engineering)
- How password is stored by the verifier (eks. UNIX og dictionary angreb)
  - Mere detalje om dictionary attacks

Hardware sikkerhed

- Tamper evident og tamper resistant
- Sikrer at passwords kun er et sted

Biometrics

- Baseret på målinger der aldrig er ens
- Skal tillade variation af målingen men ikke for meget
- Kan ikke erstatte kryptografisk authentification, men forbedre access control til nøgler

4. Network Security

Authenticated key exchange

A skal være sikker på at tale til B og omvendt

De skal udveksle en key til kryptering

Needham-Schroeder

Tegn og forklar protokollens 3 interaktioner

$A \rightarrow E_{pkB}(ID_A, n_A)$

$B \rightarrow E_{pkA}(n_A, n_B)$

$A \rightarrow E_{pkB}(n_B)$

Hvorfor er begge parter sikre på den andens identitiet

Sikkerhedshul

SSL

Tegn og forklar protokollens 5 interaktioner

$C \rightarrow n_C$

$S \rightarrow n_S + Cert_S$

$C \rightarrow E_{pkS}(pms) + Cert_C + Sign(E_{pkS}(pms))$

$S \rightarrow MAC_{pms}$ på alt

$C \rightarrow MAC_{pms}$ på alt

Særligt *final authentication of views*

Hvorfor er begge parter sikre på den andens identitiet

Password-Authenticated Key Exchange

One-sided SSL (ofte brugt i praksis)

Derefter sendes password for at authentificere brugeren

IPSec

Pro: Applikationer behøver ikke bekymre sig om kryptering og authentification

Con: Data dekrypteres på transport layer så malware kan læse cleartext

Diffie-hellmann + final authentication of views

5. System Security and Models for Security Policies

Indefra og udefrakommende trusler

Firewalls

- Beskyttelse fra udefrakommende trusler
- Packet filtering
  - To lag (stadig for primitivt)
- Proxy firewalls
  - Pro: Ingen port-scanning, da subnettet er usynligt for outsiders
  - Con: Applikationer skal kunne benytte en proxy
- Stateful firewalls
  - Kan blokere pakker, som ikke tilhører en allerede oprettet forbindelse
  - Kan maskere subnettet, *masquerading firewall*

Intrusion detection

- Rule based og statistical
  - Eks. stateful firewalls
- Honeypot

Malicious software

- Trojan, virus og worms
- Virus scannere
- Social engineering

Security policy models

- Hvad er en security policy
- Bell-Lapadula for confidentiality og omvendt for authenticity
- Prevent-detect-recover

Access control

- Beskyttelse fra brugere af systemet
- Access control matrix
  - Access control list
  - User capabilities

6. Threats and Pitfalls

Fjendens mål: STRIDE

- **S**poofing identify
- **T**ampering
- **R**epudiation
- **I**nformation disclosure
- **D**enial of Service
- **E**levation of privileges

Hvordan: X.800

- **Passive attacks**:
  - Eavesdropping
  - Traffic analysis
- **Active attacks**:
  - Replay
  - Modification

Hvem og hvor: EINOO

- External attackers
- Insiders/internal attackers
- Network attacks
- Offline attacks
- Online attacks

Hvorfor: TPM

- Threat model
- Policy
- Mechanism

Eksempler

- IIS - STR**I**D**E**, active, TP**M**
- IBM - STR**I**D**E**, active, T**P**M
- CBC - STR**I**DE, passive + active, **T**PM
- Cross-site scripting - **S**TR**I**DE, passive + active, **T**PM
- Needham-Schroeder - **S**TR**I**D**E**, active + passive, TP**M**