

# Bezpečnost v distribuovaných systémech (vybrané problémy)



Přednášky z Distribuovaných systémů  
Ing. Jiří Ledvina, CSc.

## Motivace napadání



- Internet používají důležité služby
  - Finanční transakce, medicína
- Bude využíván pro kritické služby
  - Energetické systémy, dopravní systémy, vojsko
- Internet je otevřená síť
  - Globální, bezdrátová



# Problémy síťové bezpečnosti

- Napadnutí hostitelského systému
  - Převzetí kontroly nad počítačem
- Zamezení služby
  - Bránění využívat legálním uživatelům síťové zdroje
- Útok většinou zahrnuje obě kategorie
  
- Nové útoky na směrovací protokoly (bezdrátové sítě)
  - Sinkholes (falešné nastavení minimální metriky)
  - Wormholes (vytvoření tunelu)



# Napadení hostitelského systému

- Internet worm (červ) – 1988 – BSD počítače
  - Dnes lze napadnout 10miliónů počítačů během 5 minut
- Cíle útoku
  - Čtení dat, mazání dat
  - Kompromitace jiného počítače
  - Příprava DoS útoku



## Definice napadajících programů

- **Worm** – sám sebe replikuje, používá Stack overflow
- **Virus** – program, který se sám připojí k jinému programu
- **Trojský kůň** – program, který dovolí hackerovi vytvořit zadní vrátka, spoléhá se na využití uživatele
- **Botnet** – soubor programů, pracujících autonomně a ovládaných na dálku, může být použito k rozšíření červů, příprava DDoS útoku



## Napadení systému

- Operační systémy vyžadují ochranu na třech úrovních:
  - ochrana zdrojů – ochrana proti neoprávněnému použití prostředků v OS
  - bezpečná komunikace – vlastní ochrana přenášené informace
  - ověřování uživatelů – zabezpečení, aby zprávy přicházely od ověřeného zdroje a bez modifikace
- Napadení systému:
  - pasivní
    - odposlech
    - analýza přenosu – odkud, kam, kolik, ...
  - aktivní
    - modifikace, zadržování nebo podstrkávání zpráv
    - modifikace toku dat – změna obsahu, opakování, změna pořadí, rušení, syntéza zpráv, změna adresy, změna dat, atd



# Ohodnocení bezpečnosti

- **Cíl zabezpečení:**
  - prevence pasivního útoku
  - detekce aktivního útoku.
- **Ohodnocení bezpečnosti**
  - Existuje více způsobů, uvedeme TCB (Trusted Computing Base)
  - skupina D – bez zajištění bezpečnosti, minimální ochrana (MS-DOS)
  - skupina C – volná ochrana (ponechána na uvážení)
  - skupina B – nařízená nebo vynucená ochrana
  - Skupina A – verifikovaná ochrana – vyžaduje úplný formální návrh systému, který je orientován na klasifikaci informace



# Ohodnocení bezpečnosti

- skupina C – volná ochrana (ponechána na uvážení)
  - třída C1 – volná ochrana – oddělení uživatele od dat, ochrana dat před neautorizovaným přístupem
  - třída C2 – řízená ochrana přístupu – přihlašování přes LOGIN+heslo



# Ohodnocení bezpečnosti

- skupina B – nařízená nebo vynucená ochrana
  - třída B1 – značená ochrana bezpečnosti – všechny objekty mají klasifikační značky, přístup subjektů (procesů) k objektům (zdrojům) řízen, přístup pouze k objektům s nižší klasif. značkou než uživatel
  - třída B2 – strukturovaná ochrana – systém od základů psán podle formálního modelu bezpečnosti, musí být identifikován každý kanál, který může ohrozit bezpečnost systému – rozumná úroveň zabezpečení
  - třída B3 – zavádí oblasti bezpečnosti – systém musí obsahovat monitor odkazů, jsou vytvořeny oblasti bezpečnosti, tj. seznamy uživatelů a skupin s jejich přístupovými právy k objektu a dále seznam uživatelů a skupin, pro které není zaručen žádný přístup



# Ohodnocení bezpečnosti

- skupina A – verifikovaná ochrana – vyžaduje úplný formální návrh systému, který je orientován na klasifikaci informace
  - třída A1 – systém s verifikovaným návrhem – obdoba B3 a navíc úplný formální návrh

# Ochrana zdrojů v distribuovaných systémech



- přístupovou maticí – obsahuje – model informačního toku, objekt, jeho typ a povolené operace, subjekty, které mají právo manipulovat, přístupová práva a oblasti jejich použití
- přístupovým seznamem
- seznamem schopností (capabilities)

# Zajištění bezpečné komunikace



- Zaměření na linku – zabezpečení na linkové úrovni
  - musí být transparentní pro uživatele
  - zabezpečení a šifrování kontinuálního toku dat
  - není vhodné pro otevřené systémy – není zaručena bezpečnost v koncových a mezilehlých uzlech (pouze dvoubodové sítě)
- Zaměření na koncové uzly – šifrování mezi koncovými uzly.
  - použitelné jak v dvoubodových, tak mnohabodových sítích.
- Zabezpečení na úrovni spojení – na relační nebo aplikační úrovni
  - volba úrovně zabezpečení programátorem



## Definice

- Počítačová bezpečnost – všeobecný název pro soubor prostředků, navržených k ochraně dat a maření úsilí hackerů
- Síťová bezpečnost – opatření k ochraně dat během přenosu
- Bezpečnost Internetu – opatření k ochraně dat během přenosu přes soubor propojených sítí
  - spočívá v opatření k odrazení, prevenci, detekci a korekci bezpečnostních hrozeb poškozujících přenos informace



## Bezpečnostní struktura OSI

- doporučení ITU-T X.800, v Internetu RFC 2828
- X.800 definuje 5 hlavních kategorií
  - **authentication** – ověření pravosti – ujištění, že entita je to, za co se vydává
  - **access control** – řízení přístupu – zamezení neautorizovaného využívání zdrojů
  - **data confidentiality** – důvěrnost dat – ochrana dat před neautorizovaným přístupem
  - **data integrity** – integrita dat – ujištění, že přijatá data byla odeslána ověřenou entitou
  - **non-repudiation** – nepopíratelnost – ochrana proti popření jednou z komunikujících entit



# Bezpečnostní mechanismy

- Šifrování
- Digitální podpisy
- Řízení přístupu
- Integrita dat
- Ověřování výměny dat
- Vyplňování přenosu
- Řízené směrování
- Ověřování třetí stranou



# Šifrování

- Symetrické (konvenční, tajný klíč, jeden klíč)
- Asymetrické (tajný a veřejný klíč)
- Požadavky
  - silný šifrovací mechanismus
  - šifrovací klíč zná pouze odesílatel a příjemce
  - známý šifrovací (a dešifrovací) algoritmus
  - bezpečný kanál pro distribuci klíče
- Šifrovací operace
  - Substituce (náhrada)
  - Transpozice (přehození)
- Šifra
  - bloková, proudová





# Algoritmy šifrování

- Symetrické
  - DES
  - 3DES
  - RC5
  - IDEA
  - CAST
  - BLOWFISH
  - IDEA
  - AES
- Asymetrické
  - RSA
  - Rabin system (1979)
- Výměna klíčů
  - Diffie-Hellman (Merkle)



# RSA (asymetrická šifra)

- Generování klíče
  - Nalezni dvě různá prvočísla  $p$  a  $q$
  - Necht'  $n=p*q$
  - Necht'  $\varphi = (p-1)*(q-1)$
  - Zvolme  $e$  takové, že je menší než  $n$  a relativní prvočíslo k  $\varphi$
  - Nalezněme  $d$  inverzní k  $e$ , tedy  $(e*d) = 1 \pmod{\varphi}$
- Šifrování
  - $c = m^e \pmod{n}$ ;  $m \in \{0, \dots, n-1\}$
- Dešifrování
  - $m = c^d \pmod{n}$

# Rabin system (asymetrická šifra)



- Generování klíče
  - Nalezni dvě různá prvočísla  $p$  a  $q$  taková, že  $p \equiv 3 \pmod{4}$  a  $q \equiv 3 \pmod{4}$
  - Nechť  $n = p \cdot q$  je veřejný klíč a  $p, q$  tajný klíč
- Šifrování
  - $C = m^2 \pmod{n}$ ;  $m \in \{0, \dots, n-1\}$
- Dešifrování
  - Nalezení  $a, b$  pro která platí  $a \cdot p + b \cdot q = 1 \pmod{n}$
  - $r = C^{(p+1)/4} \pmod{p}$ ,  $s = C^{(q+1)/4} \pmod{q}$
  - $x = (a \cdot p \cdot s + b \cdot q \cdot r) \pmod{n}$
  - $y = (a \cdot p \cdot s - b \cdot q \cdot r) \pmod{n}$
  - $m_1 = x$ ;  $m_2 = -x \pmod{n}$ ;  $m_3 = y$ ;  $m_4 = -y \pmod{n}$ ;

# Protokoly ověřování



Distribuované systémy – lekce 11

Ing. Jiří Ledvina, CSc.



# Ověřování

- Vytvoření a verifikace identity
- Lze realizovat
  - Něčím co máš (klíč, karta) – může být ukraden
  - Něčím co víš (hesla) – prozrazeno, sdíleno, ukradeno
  - Něčím čím jsi (biometrické údaje) – nákladné, kopírovatelné
- Vylepšení – kombinace
  - Karta, pin
- Hesla, vícenásobné použití – ukradení, odezření, odhalení (útok)
- Jednorázová hesla – nemohou být použita opakovaně
  - Systém Skey pro ověřování



# Ověřování

- Ověřovací funkce (ověření zprávy)
- Ověřovací protokol (ověření autorství zprávy)
- Ověřovací funkce
  - Šifrování zprávy
    - Ověřením je šifrovaný text
    - Příjemce těžko ověří, že zpráva nebyla modifikována
  - MAC – Message Authentication Code
    - Hodnota pevné délky jako produkt veřejné funkce, otevřeného textu a tajného klíče
  - Hashovací funkce
    - Veřejná funkce, otevřený text

# Ověřovací funkce šifrováním zprávy



- Vnitřní kontrola chyby
  - Vypočte se CRC, přidá se za zprávu a celé se to zašifruje
  - Zpráva se rozšifruje, kontrola – kontrola CRC
- Vnější kontrola chyby
  - Vypočte se šifrovaná zpráva
  - Ten, kdo s ní pracuje do zprávy nevidí
  - Vypočte se CRC a přidá se ke zprávě
  - Možnost napadení – změna zprávy i CRC
- Použití veřejného klíče
  - Utajení  $E_{K_{B+}}[P]$
  - Ověření plus podpis  $E_{K_{A-}}[P]$
  - Utajení, ověření, podpis  $E_{K_{A-}}[E_{K_{B+}}[P]]$

# Ověřovací funkce Message Authentication Code



- Kryptografický kontrolní součet
- Použití hashovací funkce
- Možnosti
  - Prefix  $MAC_K(x) = H(K \parallel x)$  – extension attack
  - Sufix  $MAC_K(x) = H(x \parallel K)$  - většinou O.K., problém pokud H není odolná proti kolizím
  - Zapouzdření  $MAC_K(x) = H(K_1 \parallel x \parallel K_2)$
- HMAC
  - $MAC_K(x) = H(K_1 \parallel H(x \parallel K_2))$
  - Přesněji viz dále
- UMAC – UMAC-30 – 10 x rychlejší než HMAC-MD5

# Ověřovací funkce

## Message Authentication Code



- Základní použití
  - $A \rightarrow B: [ M \parallel C_K(M) ]$
- Utajení a ověření vztažené k otevřenému textu
  - $A \rightarrow B: E_{K_2}[ M \parallel C_{K_1}(M) ]$
- Utajení a ověření vztažené k šifrovanému textu
  - $A \rightarrow B: E_{K_2}[ M ] \parallel C_{K_1}( E_{K_2}[ M ] )$

# Ověřovací funkce

## Výhody MAC oproti šifrování



- Efektivita – ověření může udělat pouze jeden ve skupině
- Selektivnost – ověření vybraných zpráv
- Ověřování kódu – počítačové programy mohou být ověřovány podle potřeby
- Zprávy bez utajení – např. SNMP
- Přizpůsobivost architektury
  - Ověření na aplikační úrovni
  - Šifrování na transportní úrovni
- Dlouhá perioda ochrany – MAC může být uloženo pro pozdější vyhodnocení

# Ověřovací funkce

## Data Authentication Algorithm



- DAA – Data Authentication Algorithm
  - FIPS (Federal Information Processing Standard)
  - FIPS PUB 113, ANSI X9.17
  - Vládní standard USA
  - Výstupem algoritmu je DAC – Data Authentication Code
- Používá DES v režimu CBC
  - IV nastaven na 0
  - Zpráva rozdělena na 64 bitové bloky
  - Ověření podle posledního bloku
- Dnes není považován za dostatečně bezpečný

# Hashovací funkce



- Vlastnosti
  - Nepoužívá klíč (pouze veřejně známý algoritmus)
  - Není reverzibilní (není známa inverzní funkce)
  - Malé změny ve zprávě způsobí náhodné změny výsledku
  - Aplikovatelná na blok libovolné (omezené) délky
  - Výstupem je blok pevné délky
  - Relativně jednoduchý výpočet
  - Netriviální (neproveditelný) zpětný výpočet
  - Pro dané  $x$  je obtížné najít takové  $y$ , aby  $H(x) = H(y)$
  - Obtížné najít pár  $(x,y)$  tak, aby  $H(x) = H(y)$



## Hashovací funkce - použití

- Utajení a ověření
  - $A \rightarrow B: E_K[ M \parallel H(M) ]$
- Ověření (obdoba MAC)
  - $A \rightarrow B: [ M \parallel E_K[ H(M) ] ]$
- Digitální podpis (ověření)
  - $A \rightarrow B: [ M \parallel E_{K_A^-}[ H(M) ] ]$
- Digitální podpis a utajení
  - $A \rightarrow B: E_K[ M \parallel E_{K_A^-}[ H(M) ] ]$
- Ověření
  - $A \rightarrow B: [ M \parallel H(M) \parallel S ]$
- Ověření a utajení
  - $A \rightarrow B: E_K[ M \parallel H(M) \parallel S ]$



## Hashovací funkce - algoritmy

- MD5 (RFC 1321)
  - Rivest (RSA)
  - výstup 128 bitů, bloky 256 bitů
- SHA-1
  - NIST (National Institute of Standards and Technology)
  - FIPS PUB 180
  - Založeno na MD4 (předchůdce MD5)
  - Max délka zprávy  $2^{64}-1$  bitů
  - Výstup 160 bitů



# Hashovací funkce - algoritmy

- RIPEMD-160
  - RIPE project
  - Výstup 160 bitů
  - Max délka zprávy  $2^{64}-1$  bitů
- HMAC – RFC 2104
  - Používá MD5, SHA-1, ...
  - $HMAC_K = H[ (K \text{ xor opad} ) \parallel H[ (K \text{ xor ipad} ) \parallel M ] ]$
  - ipad, opad – konstanty
  - H je hashovací funkce



# Digitální podpis

- Chrání zprávy před napadením komunikujících entit
- Předpokládá nedůvěryhodné entity
- Odolný proti spiknutí
  - Třetí strana a podepisující
  - Třetí strana a příjemce
- Podobné vlastnosti jako podpis rukou
  - Podpis včetně datumu a času
  - Schopnost ověřit obsah zprávy v době podpisu
  - Podpis ověřitelný třetí důvěryhodnou stranou
  - Chrání před paděláním i popřením
  - Jednoduchá realizace i rozpoznání
  - Složitě napodobení (falšování)





## Kategorie digitálního podpisu

- Přímý – pouze mezi zúčastněnými stranami
- Arbitrovaný – každá zpráva podepsaná A prochází arbitrem T do B
- Možnosti přímého podpisu
  - $A \rightarrow B: E_{K_{B+}}[ M ]$  - pouze šifrovaná, nepodepsaná
  - $A \rightarrow B: [ M ] \parallel E_K[ H( M ) ]$  - pouze ověření, sdílené tajemství
  - $A \rightarrow B: E_{K_{B+}}[ E_{K_{A-}}[ M ] ]$  – šifrování (B+) a ověření (A-),
  - $A \rightarrow B: E_K[ M \parallel E_{K_{A-}}[ H(M) ] ]$  – symetrické šifrování, podpis
- Nevýhody
  - Odesílatel může tvrdit, že tajný klíč byl prozrazen
  - Použití starých zpráv, napadení času



## Digitální podpis - arbitrovaný

- Arbitr je důvěryhodný prostředník pro obě strany
  - $A \rightarrow T: [ M \parallel E_{K_{AT}}[ ID_A \parallel H(M) ] ]$
  - $T \rightarrow B: E_{K_{BT}}[ ID_A \parallel M \parallel E_{K_{AT}}[ ID_A \parallel H(M) ] \parallel TS ]$
- Prostředník nemusí rozpoznat obsah zprávy
  - $A \rightarrow T: [ E_{K_{AB}}[ M ] \parallel E_{K_{AT}}[ ID_A \parallel H( E_{K_{AB}}[ M ] ) ] ]$
  - $T \rightarrow B: E_{K_{BT}}[ ID_A \parallel E_{K_{AB}}[ M ] \parallel E_{K_{AT}}[ ID_A \parallel H( E_{K_{AB}}[ M ] ) ] \parallel TS ]$
- Nevýhoda
  - T se může spřáhnout s A nebo s B
  - A a T – může popřít podpis A
  - B a T – může vytvořit podpis A
- Řešení – použití veřejných klíčů

# Digitální podpis – veřejné klíče



- Použití veřejných klíčů
  - $A \rightarrow T: E_{K_{A-}}[ID_A \parallel E_{K_{B+}}[E_{K_{A-}}[M]]]$
  - $T \rightarrow B: E_{K_{T-}}[ID_A \parallel E_{K_{B+}}[E_{K_{A-}}[M]] \parallel TS]$
  - $E_{K_{A-}}$  - tajný klíč A (ověření A)
  - $E_{K_{B+}}$  - veřejný klíč B (šifrování pro B)
  - $E_{K_{T-}}$  - tajný klíč T (ověření T)

# Digital Signature Standard (DSS)



- FIPS PUB 186 of NIST
  - Federal Information Processing Standard
  - National Institute of Standards and technology
- Používá SHA-1
- Představuje novou techniku podpisu – Digital Signature Algorithm (DSA)
- Používá veřejné klíče
- Pouze podpis, nikoliv distribuce klíče nebo šifrování
- Používá algoritmus ElGamal
- Odlišný od běžně používaných technik



# Digital Signature Standard (DSS)

- $p$  – prvočíslo 512 až 1024 bitů
- $q$  – prvočíselný dělitel  $(p-1)$  160 bitů dlouhý
- $g$  – základ tak, že  $h^{((p-1)/q)} \bmod p > 1$ ;  $1 < h < (p-1)$
- $x$  - náhodné číslo  $0 < x < q$
- $y = g^x \bmod p$  – veřejný klíč
- $k$  – náhodné číslo  $0 < k < q$  / relační klíč (tajný)
  
- $r = (g^k \bmod p) \bmod q$
- $s = (k^{-1} (H(M) + xr)) \bmod q$
- Podpis:  $(r, s)$
  
- $w = (s^{-1} \bmod q)$
- $u_1 = (H(M)w) \bmod q$
- $u_2 = (r)w \bmod q$
- $(g^k \bmod p) \bmod q$
- $v = ((g^{u_1} y^{u_2}) \bmod p) \bmod q$
- Test:  $v = r$



# Protokoly vzájemného ověření

- Základní vlastnosti
  - Důvěryhodnost
    - Prevence proti maškarádě a kompromitaci relačního klíče
    - Vyžaduje existenci sdíleného klíče nebo veřejných klíčů
  - Časová souslednost
    - Prevence proti útoku opakováním (replay attack)
    - Jednoduché sekvenční číslování je nepraktické
    - Časové značky – vyžadují synchronizované hodiny
- Třídy protokolů
  - Přímé - vzájemné ověření dvou entit
  - S ověřovacím serverem
    - Dvouúrovňový hierarchický systém s KDC (Key Distribution Center)



# Protokoly vzájemného ověření

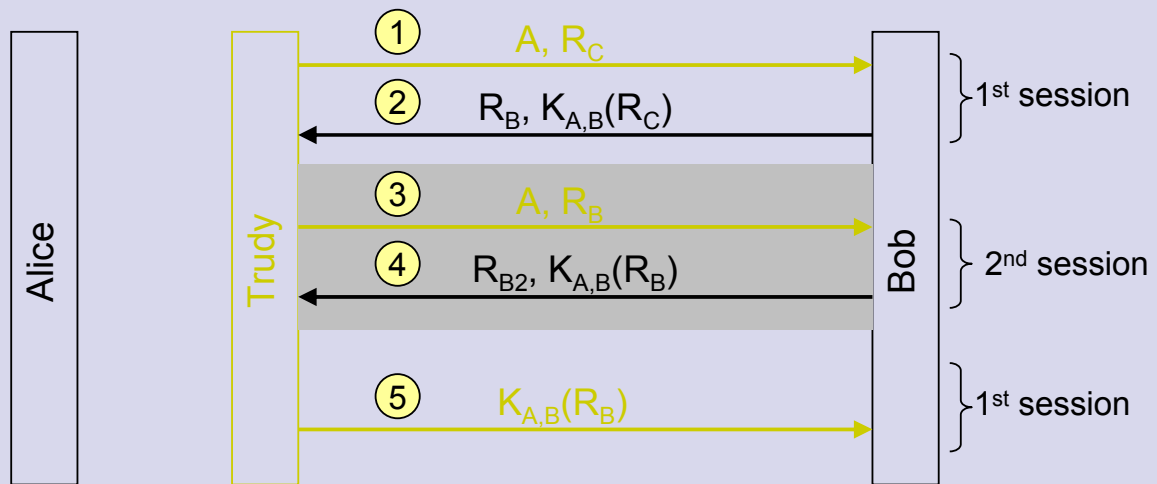
- Založené na sdíleném tajemství
- Založené na symetrickém šifrování
  - Sdílení tajného klíče s každou entitou
  - Generování relačního klíče
  - Relační klíče mají krátkou dobu života (eliminace prozrazení tajného klíče)
  - Použití ověření + šifrování přenosu
- Založené na asymetrickém šifrování
  - Distribuce veřejného klíče mezi entitami
  - Znalost veřejného klíče KDC
  - Veřejný klíč KDC slouží k ověření pravosti KDC



# Protokoly vzájemného ověřování Přímé ověřování – symetrické šifrování

- Základní varianta
  - $A \rightarrow B: [A]$
  - $B \rightarrow A: [N_B]$
  - $A \rightarrow B: [f(K \parallel N_B)]$
  - $A \rightarrow B: [N_A]$
  - $B \rightarrow A: [f(K \parallel N_A)]$
- Redukce počtu zpráv
  - $A \rightarrow B: [A \parallel N_A]$
  - $B \rightarrow A: [N_B \parallel f(K \parallel N_A)]$
  - $A \rightarrow B: [f(K \parallel N_B)]$
- Útok
  - $T \rightarrow B: [A \parallel N_A]$
  - $B \rightarrow T: [N_B \parallel f(K \parallel N_A)]$
  - $T \rightarrow B: [A \parallel N_B]$
  - $B \rightarrow T: [N'_B \parallel f(K \parallel N_B)]$
  - $T \rightarrow B: [f(K \parallel N_B)]$
- Modifikace
  - $A \rightarrow B: [A]$
  - $B \rightarrow A: [N_B]$
  - $A \rightarrow B: [N_A \parallel f(K \parallel N_B)]$
  - $B \rightarrow A: [f(K \parallel N_A)]$

# Útok odrazem



## Protokoly vzájemného ověřování Přímé ověřování – nesymetrické šifrování



- Základní varianta (asymetrické šifrování)
  - $A \rightarrow B: E_{K_B+} [ A \parallel N_A ]$
  - $B \rightarrow A: E_{K_A+} [ N_A \parallel N_B ]$
  - $A \rightarrow B: E_{K_B+} [ N_B ]$
- Útok (Gavin Lowe)
  - $A \rightarrow U(A): E_{K_U+} [ A \parallel N_A ]$
  - $U(A) \rightarrow B: E_{K_B+} [ A \parallel N_A ]$
  - $B \rightarrow U(A): E_{K_A+} [ N_A \parallel N_B ]$
  - $U(A) \rightarrow A: E_{K_A+} [ N_A \parallel N_B ]$
  - $A \rightarrow U(A): E_{K_U+} [ N_B ]$
  - $U(A) \rightarrow B: E_{K_B+} [ N_B ]$
- B je přesvědčen, že komunikuje s A
- $U(A)$  má k dispozici  $[ N_A \parallel N_B ]$ , které může např. použít ke konstrukci relačního klíče.

# Protokoly vzájemného ověření S ověřovacím serverem



- Needham-Shroeder (1978)
- Otway-Ress (1987)
- Andrew Secure RPC handshake (1987)
- Wide-mouthed-frog protocol (1989)
- Kerberos protocol (1987)
  
- Zdroj: [www.lsv.ens-cachan.fr-spore](http://www.lsv.ens-cachan.fr-spore)
- Cca 49 protokolů

# Protokoly vzájemného ověření S ověřovacím serverem



- Needham-Schroeder, symetrické šifrování
  - Používá náhodná čísla pro identifikaci zpráv
  - $ID_A$ ,  $ID_B$  jsou identifikátory entit
  - $K_{AB}$  je relační klíč pro komunikaci A a B
  - $K_A$ ,  $K_B$  jsou šifrovací klíče pro komunikaci KDC s A a KDC s B.
    - $A \rightarrow KDC: [ID_A \parallel ID_B \parallel N_1]$
    - $KDC \rightarrow A: E_{K_A}[K_{AB} \parallel ID_B \parallel N_1 \parallel E_{K_B}[K_{AB} \parallel ID_A]]$
    - $A \rightarrow B: E_{K_B}[K_{AB} \parallel ID_A]$ 
      - útok – zachycení, prolomení, podvrhnutí  $\rightarrow$  časové značky
    - $B \rightarrow A: E_{K_{AB}}[N_2]$
    - $A \rightarrow B: E_{K_{AB}}[N_2 - 1]$

# Protokoly vzájemného ověření S ověřovacím serverem



- Needham-Schroeder, nesymetrické šifrování

- Útok Gavin Lowe 1995

- $A \rightarrow KDC: [ID_A \parallel ID_B]$
- $KDC \rightarrow A: E_{K_S} [K_{B^+} \parallel ID_B]$
- $A \rightarrow B: E_{K_{B^+}} [N_A \parallel ID_A] \rightarrow A \rightarrow T: E_{K_{T^+}} [N_A \parallel ID_A]$
- $B \rightarrow KDC: [ID_A \parallel ID_B] \rightarrow T \rightarrow B: E_{K_{B^+}} [N_A \parallel ID_A]$
- $KDC \rightarrow B: E_{K_S} [K_{A^+} \parallel ID_A]$
- $B \rightarrow A: E_{K_{A^+}} [N_A \parallel N_B] \rightarrow B \rightarrow T: E_{K_{A^+}} [N_A \parallel N_B]$
- $T \rightarrow A: E_{K_{A^+}} [N_A \parallel N_B]$
- $A \rightarrow B: E_{K_{B^+}} [N_B] \rightarrow A \rightarrow T: E_{K_{T^+}} [N_B]$
- $T \rightarrow B: E_{K_{B^+}} [N_B]$

# Protokoly vzájemného ověření S ověřovacím serverem



- Otway Rees (1997)

- Symetrické šifrování

- $A \rightarrow B: [M \parallel ID_A \parallel ID_B \parallel E_{K_A} [N_A \parallel M \parallel ID_A \parallel ID_B]]$
- $B \rightarrow KDC: [M \parallel ID_A \parallel ID_B \parallel E_{K_A} [N_A \parallel M \parallel ID_A \parallel ID_B] \parallel E_{K_B} [N_B \parallel M \parallel ID_A \parallel ID_B]]$
- $KDC \rightarrow B: [M \parallel E_{K_A} [N_A \parallel K_{AB}] \parallel E_{K_B} [N_B \parallel K_{AB}]]$
- $B \rightarrow A: [M \parallel E_{K_A} [N_A \parallel K_{AB}]]$

# Protokoly vzájemného ověření S ověřovacím serverem



- Andrew Secure RPC (1987)
  - Symetrické šifrování
  - Modifikace BAN (Burrows, Abadi, Needham 1989)
  - Modifikace Garwin Lowe 1996
  
- $A \rightarrow B: [ ID_A \parallel N_A ]$
- $B \rightarrow A: E_{K_{AB}} [ N_A \parallel K'_{AB} \parallel ID_B ]$
- $A \rightarrow B: E_{K'_{AB}} [ N_A ]$
- $B \rightarrow A: [ N_B ]$
  
- $K'_{AB}$  – nový relační klíč
- $N_B$  – nové náhodné číslo

# Protokoly vzájemného ověření S ověřovacím serverem



- Wide-mouthed frog protocol (1989)
  - Modifikace Gavin Lowe
  - Symetrické šifrování
  
- $A \rightarrow KDC: [ A \parallel E_{K_A} [ TS_A \parallel ID_B \parallel K_{AB} ] ]$
- $KDC \rightarrow B: E_{K_B} [ TS_{KDC} \parallel ID_A \parallel K_{AB} ]$
- $B \rightarrow A: E_{K_{AB}} [ N_B ]$
- $A \rightarrow B: E_{K_{AB}} [ N_B + 1 ]$



# Protokoly vzájemného ověření S ověřovacím serverem



- Denningovo distribuční schéma
  - Používá časové značky TS
  - Používá symetrické šifrování
  - Časová značka musí být  $|C - TS| < \Delta t_1 + \Delta t_2$
  - Znovupoužití relačního klíče může být detekováno B
  
- $A \rightarrow KDC: [ ID_A \parallel ID_B ]$
- $KDC \rightarrow B: E_{K_A}[ K_{AB} \parallel ID_B \parallel TS \parallel \underline{E_{K_B}[ K_{AB} \parallel ID_A \parallel TS ]}]$
- $A \rightarrow B: \underline{E_{K_B}[ K_{AB} \parallel ID_A \parallel TS ]}$
- $B \rightarrow A: E_{K_{AB}}[ N_1 ]$
- $A \rightarrow B: E_{K_{AB}}[ f(N_1) ]$

# Zabezpečení replikovaných serverů



- Klient zadá požadavek skupině replikovaných serverů
- Servery mohou být s Byzantinskými chybami
- Servery získají odpovědi od všech serverů a hledají majoritu hlasováním
- Problém: klient potřebuje ověřit každý server (porušení transparentnosti replik)
- Řešení: sdílené tajemství
  - Žádný z uživatelů nezná celé tajemství

# Zabezpečení replikovaných serverů



- Cíl
  - Předpokládejme, že můžeme tolerovat  $C$  chyb
  - Potřebujeme kombinovat odpovědi tak, že  $C+1$  korektních serverů je schopné dát korektní odpověď
- $(k,n)$  schéma podpisu
  - Máme jeden veřejný klíč  $K^+$
  - $N$  sdílených odpovídajících privátních klíčů  $K^-$
  - Můžeme sdílet hodnoty v šifrované sdíleným privátním klíčem
  - Klient může dešifrovat  $v$  s použitím  $K^+$  pouze tehdy, jestliže zná alespoň  $k$  hodnot zašifrovaného  $V$

## Internet Key Exchange



Distribuované systémy – lekce 11  
Ing. Jiří ledvina, CSc

# Úvod



- Standardní protokol pro vytváření a údržbu Security Association
  - Spojení ISAKMP/Oakley
  - ISAKMP - Internet Security Association and Key Management Protocol
    - Definuje procedury formáty paketů k vytváření, potvrzování, modifikaci a rušení Security Association
  - Oakley – výměna zpráv
- IKEv2 – RFC4306
- Použití
  - Mezi hosty, mezi bránami, mezi hostem a branou
- Ověřování pomocí Diffie-Hellman

# IKE fáze



- Fáze I: vytvoření bezpečného komunikačního kanálu
  - Obsahuje kryptografické algoritmy, metody ověřování, klíče
  - Vzájemné ověřování
- Fáze II: použití tohoto kanálu
  - Účastníci vytváří IPsec Security Association
  - Jedna fáze I může ochránit více výměn fáze II
  - Důvodem je náročná fáze I, méně náročná fáze II



## IKE fáze I

- Zahrnuje dva stavy (výměna zpráv Request – Response)
  - INIT
    - Dohadování kryptografických algoritmů
    - Výměna náhodných čísel
    - Provedení Diffie-Hellman výměny
    - Probíhá bez zabezpečení
  - AUTH
    - Zprávy jsou chráněny klíči odvozenými v předchozí fázi
    - Ověřuje předchozí zprávy
    - Vytváří první CHILD-SA



## IKE fáze II

- Vytváří nové CHILD-SA
  - Relativně jednoduchá operace
  - Může být vyvolána jakoukoliv stranou
  - Zajišťuje „perfect forward secrecy“
    - I když útočník zaznamená všechna data poslaná přes bezpečné spojení
    - Nedokáže rekonstruovat klíče pro výměnu mezi CHILD-SA

# Public Key Infrastructure



Distribuované systémy – lekce 11  
Ing. Jiří ledvina, CSc

## Úvod



- Infrastruktura pro podporu použití šifrování veřejným klíčem
- PKI zahrnuje
  - Certifikační autority
  - Certifikáty
  - Úložiště pro obnovu certifikátů
  - Metody pro zneplatnění certifikátů
  - Metody pro vyhodnocení řetězu certifikátů od známého veřejného klíče po cílové jméno



## Certifikační autorita

- Důvěryhodná entita, která udržuje seznam veřejných klíčů pro ostatní entity
- CA generuje certifikáty
- CA dovede potvrdit pravost veřejného klíče
- Není třeba komunikovat s vlastníkem veřejného klíče



## Certifikát

- Certifikát je podepsaná zpráva zaručující, že dané jméno je spojeno s veřejným klíčem
  - Běžně omezeno časově
- Ověření certifikátu se děje na základě znalosti veřejného klíče certifikační autority
- Certifikát může také obsahovat informaci o službách, které držitel zaručuje



# Modely důvěrnosti

- Monopoly model
- Monopoly plus RA
- Delegation CA
- Oligarchy model
- Anarchy model



# Monopoly model

- Jedna CA je důvěryhodná pro všechny
- Všichni musí mít certifikát od této CA
- Veřejný klíč CA je základem důvěry a musí být zahrnut do všeho software i hardware používajícího PKI
- Nevýhoda
  - Neexistuje taková důvěryhodná organizace
  - Pokud takovou vybereme, těžko zvolíme jinou
  - Celý svět by ji využíval (výkonnost, bezpečnost)
  - Neexistovala by soutěž (cena)



## Monopoly plus RA

- RA – registrační autorita
- Registrační autority jsou přidružené k jedné CA a je jim důvěřováno
- RA kontrolují identitu uživatelů a poskytují CA odpovídající informaci (identitu a veřejný klíč) při vydávání certifikátů
- výhoda
  - Více míst pro vydávání certifikátů
  - Monopol pro registraci



## Delegované CA

- Kořenová CA vydává certifikáty ostatním CA (delegovaným CA), zaručujíc jejich důvěryhodnost jako CA
- Uživatelé mohou obdržet certifikáty od delegovaných CA jako by to bylo od kořenové CA
- Výhoda – více CA, menší nebezpečí úzkého místa
- Vydávání závisí na jednom CA
- Obtížnější ověřování certifikátu
  - Uživatel musí prohledávat řetěz certifikátů
  - Uspořádání do hierarchie prohledávání ulehčuje





## Oligarchy model

- Existuje několik důvěryhodných CA
- Je akceptován certifikát vydaný kteroukoliv z nich
- Obecně jsou takto konfigurovány web prohlížeče
- Výhoda – konkurence CA
- Snížení bezpečnosti
  - Možnost vložit do seznamu nedůvěryhodnou CA
  - Potřeba chránit více CA před kompromitováním



## Anarchy model

- Kdokoliv může podepsat certifikát pro kohokoliv
  - Neexistuje CA nebo seznam CA určený uživatelům
  - Uživatelé si sami určují kdo je důvěryhodný
  - Uživatelé si musí sami nalézt řetězec od důvěryhodné CA k cíli
- Certifikáty mohou být od
  - Zdrojů
  - Subjektů
  - Veřejných úložišť – web serverů
- Výhody
  - Mnoho potenciálních důvěrných kořenů
  - Není třeba zavádět drahou infrastrukturu

# Anarchy model



- Bezpečnost
  - Je důvěra tranzitivní
  - Co je to dost dobrá důvěra

# Rušení certifikátů



- CA může zrušit certifikát s veřejným klíčem pokud jej uživatel nechce dále používat
- Kompromitování certifikátu
- Náhrada klíče (nový klíč)
- Ukončení členství ve skupině (organizaci)
- Způsoby informování ostatních
  - Broadcast
  - Uložení na veřejném místě
  - Všichni se ptají CA



## Revocation list

- CA může periodicky vysílat CRL
  - Podepsaný CA
  - Před použitím certifikátu musí být prohlédnut CRL
  - Nevýhoda – četnost vysílání CRL
  - Nevýhoda počet certifikátů
- Delta CRL
  - Vysílají se pouze změny
  - Podstatně kratší
  - Celý seznam se vysílá méně často
  - Potřeba prohlédnout zěny i celý CRL



## On-line Revocation Servers

- ORLS je systém serverů, který může být dotazován na stav jednotlivých certifikátů
- Musí obsahovat celý CRL
- Místo toho by bylo možné udržovat seznam nezrušených certifikátů
- Nevýhoda je v délce seznamu