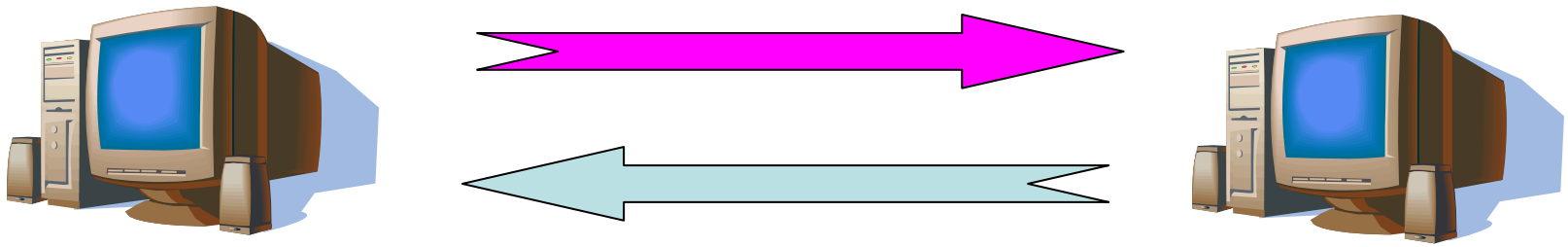
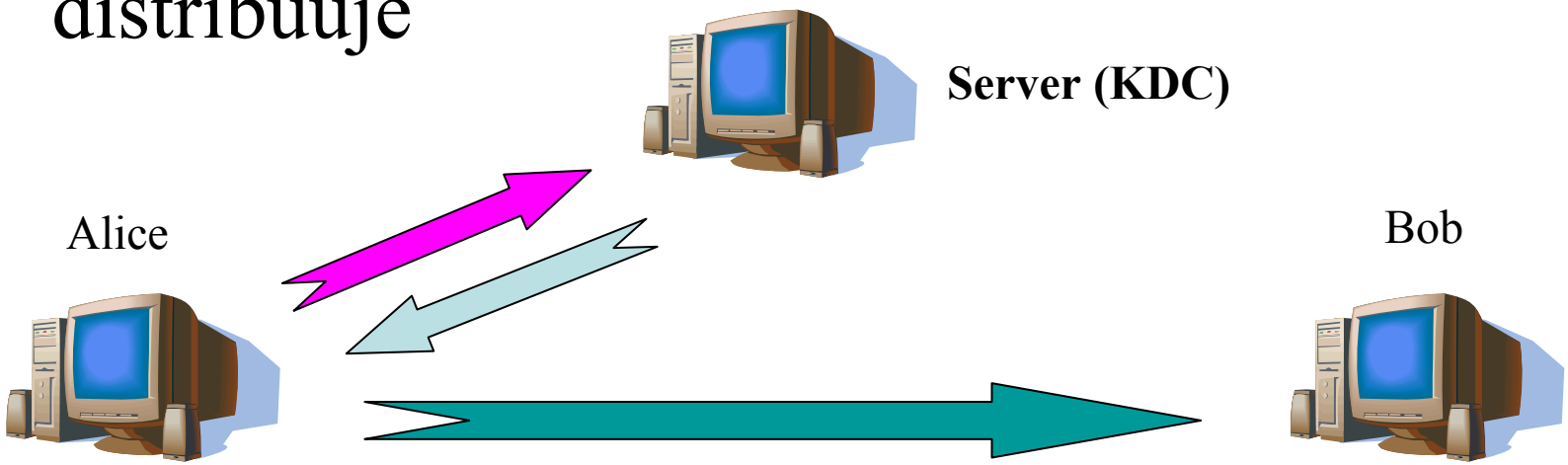
PV157 – Autentizace a řízení přístupu

Autentizační protokoly (pokračování)



Kerberos

- **KDC** (key distribution center) – server sdílí klíč s každým klientem; (klienti však mezi sebou klíče nesdílí); distribuuje klíče, které generuje.
- **KTC** (key translation center) – server negeneruje klíče sám; klíč dodá jedna ze stran; server klíč distribuuje

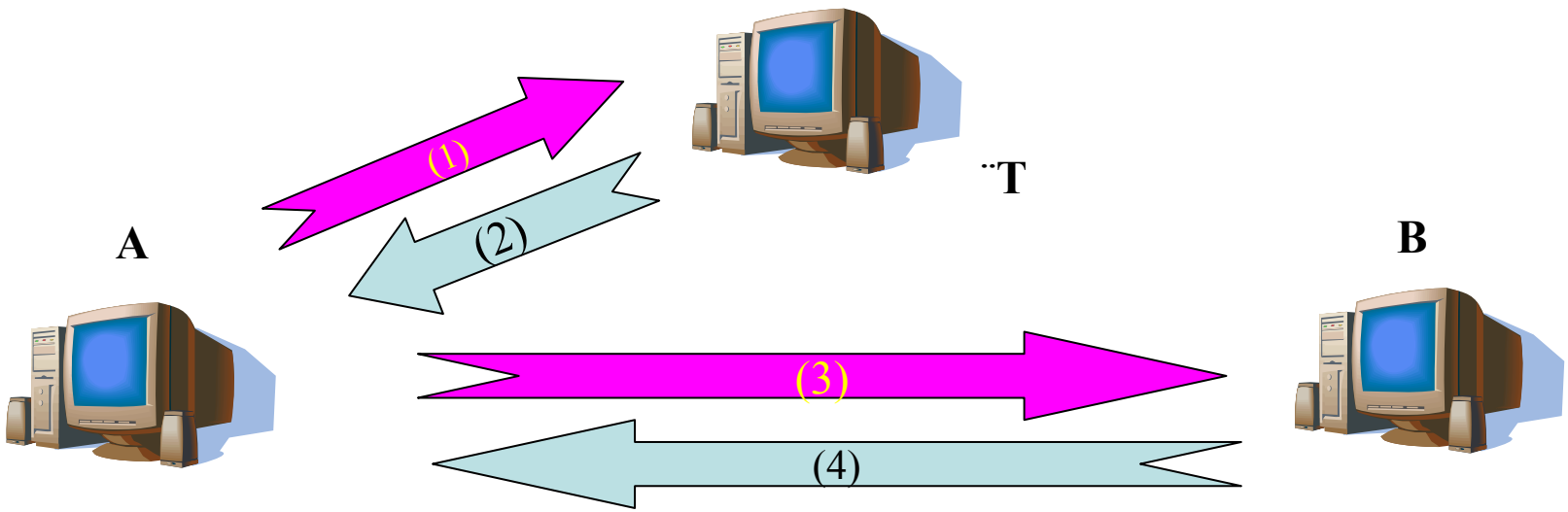


Kerberos

- Vznikl při projektu Athena na MIT
- Symetrická šifra E
- 2 strany (A, B) a důvěryhodný autentizační server (značíme T)
- Cíl:
  - autentizace subjektu A vůči B
  - ustavení klíče k (zvolí T)
  - případně distribuce tajemství sdíleného A a B
- Každá strana sdílí tajemství se serverem K_{AT}, K_{BT}

Kerberos

- Zjednodušená verze protokolu
  - L – doba platnosti („lifetime“)
  - Def.: $ticket_B = E_{K_{BT}}(k, \text{“A”}, L)$, $auth = E_k(\text{“A”}, T_A)$
  - (1) $A \rightarrow T$: ”A”, ”B”, n_A
  - (2) $A \leftarrow T$: $ticket_B$, $E_{K_{AT}}(k, n_A, L, \text{“B“})$
  - (3) $A \rightarrow B$: $ticket_B$, $auth$
  - (4) $A \leftarrow B$: $E_k(T_A)$



Asymetrické techniky přenosu klíče

- Zašifrování podepsaných klíčů
  - $A \rightarrow B: P_B(S_A(\text{“B”}, k , t_A))$
  - (volitelné) časové razítko t_A zároveň autentizuje A vůči B
- Separátní šifrování a podpis
  - $A \rightarrow B: P_B(k, t_A), S_A(\text{“B”}, k, t_A)$
  - Pouze v případě, kdy z podpisu nelze získat podepsaná data
- Podepsání zašifrovaných klíčů
  - $A \rightarrow B: t_A, P_B(\text{“A”}, k), S_A(\text{“B”}, t_A, P_B(\text{“A”},k))$

Asymetrické techniky přenosu klíče

- X.509 obousměrná autentizace s přenosem klíče
- Def.: $D_A = (t_A, r_A, \text{“B”}, P_B(k_1))$
  $D_B = (t_B, r_B, \text{“A”}, P_A(k_2))$
- Protokol
  - $A \rightarrow B$: $cert_A, D_A, S_A(D_A)$
  - $A \leftarrow B$: $cert_B, D_B, S_B(D_B)$

Asymetrické techniky ustavení klíče

- Diffie-Hellman protokol pro ustavení sdíleného tajemství
  - Společné prvočíslo p, generátor α v Z_p
  - A volí tajné x, B volí tajné y
  - $A \rightarrow B: \alpha^x \bmod p$
  - $A \leftarrow B: \alpha^y \bmod p$
  - A a B sdílí $K=\alpha^{xy} \bmod p$

Zero-knowledge protokoly

- Český překlad: protokoly s nulovým rozšířením znalostí
- Jdou dále než protokoly sdělující hesla i protokoly typu výzva-odpověď
- Zero-knowledge – umožňují demonstrovat znalost nějakého tajemství bez odhalení jakékoliv informace použitelné pro získání tajemství
- Úplnost (completeness) – poctivé strany vždy dosáhnou úspěšného výsledku
- Korektnost (soundness) – pravděpodobnost, že nepoctivý útočník se může úspěšně vydávat za jinou stranu je mizivá

Zero-knowledge protokoly

- Identifikační protokol Feige-Fiat
- Důvěryhodná strana T volí modulus $n = p \cdot q$ (jako v RSA), n zveřejní, ale p a q uchová v tajnosti
- A volí tajné **s** (nesoudělné s n, $1 \leq s \leq n-1$), spočítá $v = s^2 \bmod n$. Veřejný klíč A je **v**.
- Subjekt A se autentizuje subjektu B:
  - $A \rightarrow B$: $x = r^2 \bmod n$
  - $A \leftarrow B$: e = 0 nebo 1
  - $A \rightarrow B$: $y = r \cdot s^e \bmod n$
- Opakujeme t-krát. Pravděpodobnost podvádění je 2^{-t}.

Protokoly vyšší úrovně – SSL/TLS

Protokol SSL/TLS poskytuje:

- Autentizaci stran – strany jsou autentizovány pomocí certifikátů a protokolu výzva-odpověď
- Integritu – autentizační kódy (message authentication code - MAC) zajišťují integritu a autenticitu dat
- Důvěrnost – po úvodní inicializaci („handshake“), je ustaven symetrický šifrovací klíč, kterým je šifrována všechna následující komunikace (včetně přenosu hesel apod.)

Principy SSL/TLS

- Pozice SSL/TLS
  - Mezi aplikační vrstvou a protokolem TCP
  - SSL/TLS nevidí do aplikačních dat
  - SSL/TLS neprovádí elektronické podepisování přenášených dat

Aplikační vrstva
SSL/TLS
TCP/UDP
IP
Linková vrstva
Fyzická vrstva

Komponenty SSL/TLS

- Složení protokolu SSL/TLS z komponent
  - Record Layer Protocol – zpracovává aplikační data
  - Handshake Protocol – úvodní domluva parametrů
  - Change Cipher Specification Protocol – použití nových parametrů šifrování
  - Alert protocol – informace o chybách a varováních

Klíče v SSL/TLS

- Použití klíčů
  - Klient generuje PreMasterSecret, šifruje veřejným klíčem serveru a posílá serveru
  - Obě strany vytvoří blok klíčů z PreMasterSecret (posílá se šifrovaně) a náhodných čísel ClientHello a ServerHello (posílají se nešifrovaně)
  - Blok klíčů tvoří klíče pro
    - MAC klient → server
    - MAC server → klient
    - šifrování klient → server
    - šifrování server → klient
    - inicializační vektory

Record Layer Protocol

- Základní vrstva protokolu
- Pracuje nad TCP/IP (nebo jiným transportním protokolem).
- Umožňuje kombinaci s různými protokoly vyšší úrovně (HTTP, FTP, telnet apod.), které běží beze změny
- Posloupnost kroků
  - rozdělení dat na bloky o max. velikosti 2^{14} bajtů
  - komprimace dat
  - výpočet MAC
  - doplnění na délku bloku šifrovacího algoritmu
  - šifrování

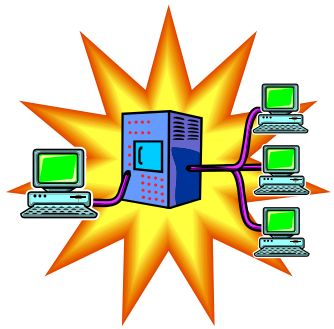
Inicializační fáze

- Handshake Protocol
  - Umožňuje vzájemnou autentizaci serveru a klienta
  - Implicitně je autentizace serveru povinná a autentizace klienta volitelná
  - Autentizace prezentací **digitálních certifikátů** a znalostí odpovídajících soukromých klíčů
  - Během inicializační fáze jsou vyměněna náhodná čísla a další data, nutná pro výpočet bloku klíčů

SSL/TLS

Client

Server



Client Hello

Server Hello, (Server Cert , Client Cert Request,…)

Client Key Exchange, Cipher Spec, (Client Cert , ...)

Application Data

S E C U R E

IPsec

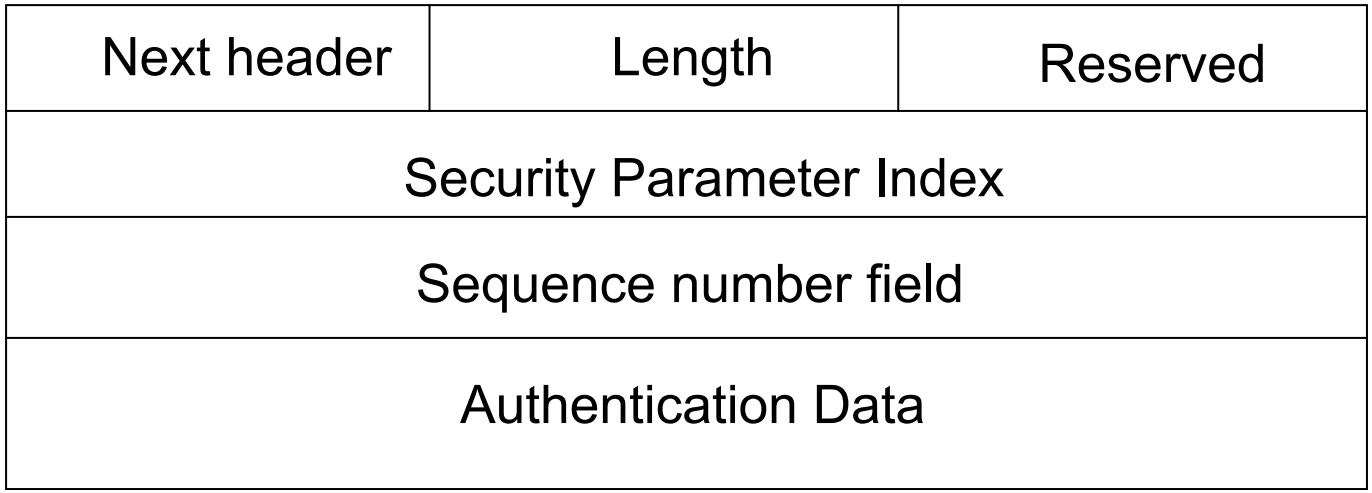
- Protokoly IPv4 – nedostatečná bezpečnost
- Historie
  - Myšlenka IPsec již v roce 1991
  - RFC v roce 1998
  - vývoj neustále pokračuje
  - IPsec pro IPv4 jen přechodné řešení, neboť IPv6 již řeší problémy bezpečnosti
- IPv6
  - Větší množství adres (adresy IPv4 nebudou již brzy stačit)
  - Bezpečnost (IPsec povinný)
  - Mobilita

IPsec

- IPsec zajišťuje
  - Autentizaci původu dat – každý datagram je ověřován, zda byl odeslán uvedeným odesilatelem
  - Integrita dat – ověřuje se, zda data nebyla při přenosu změněna
  - Důvěrnost dat – data jsou před přenosem šifrována
  - Ochrana před útokem přehráním – útočník nemůže zneužít odposlechnutou komunikaci k útoku přehráním
  - Automatický management klíčů

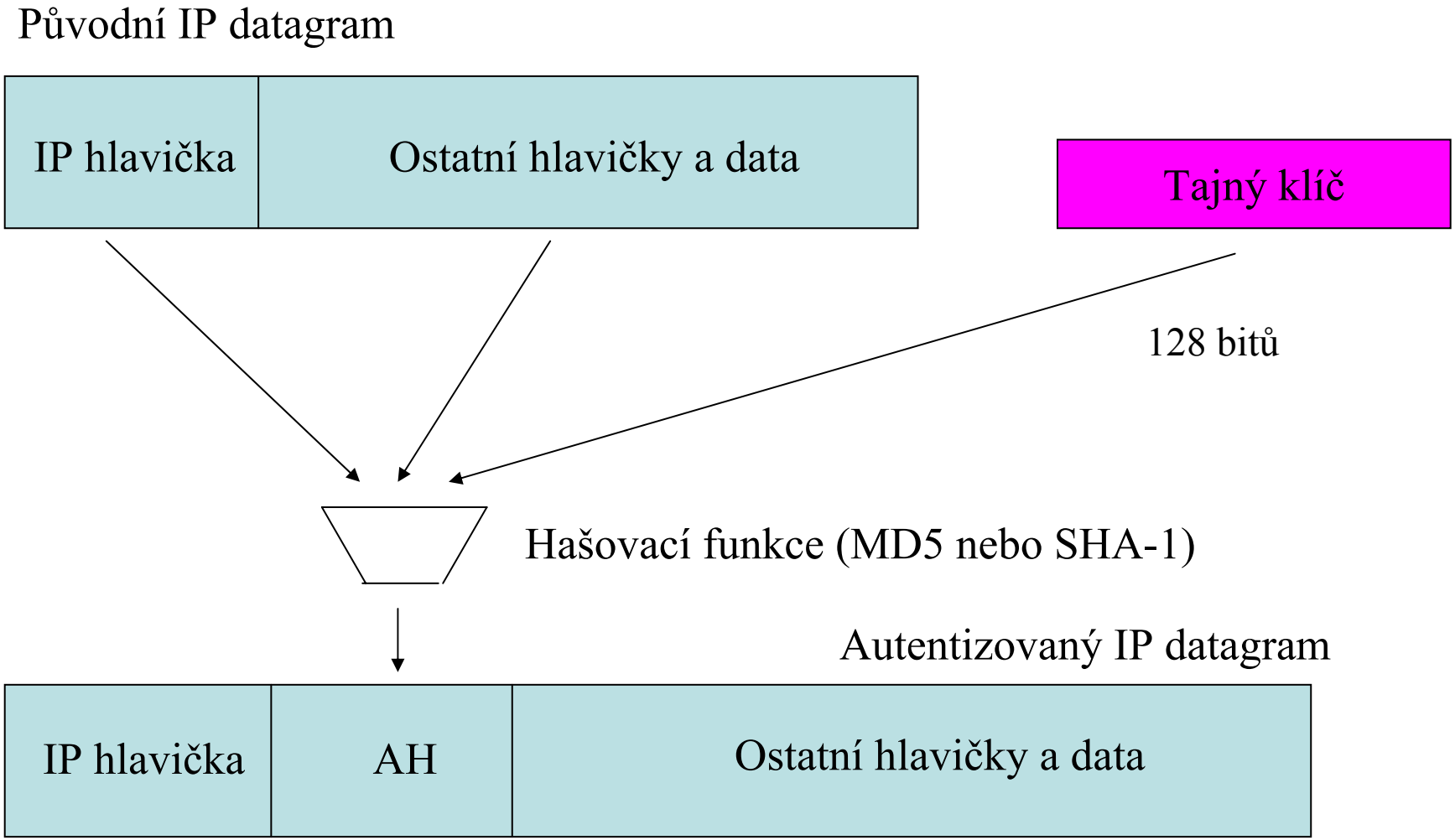
IPsec – AH

- Autentizační hlavička (AH)



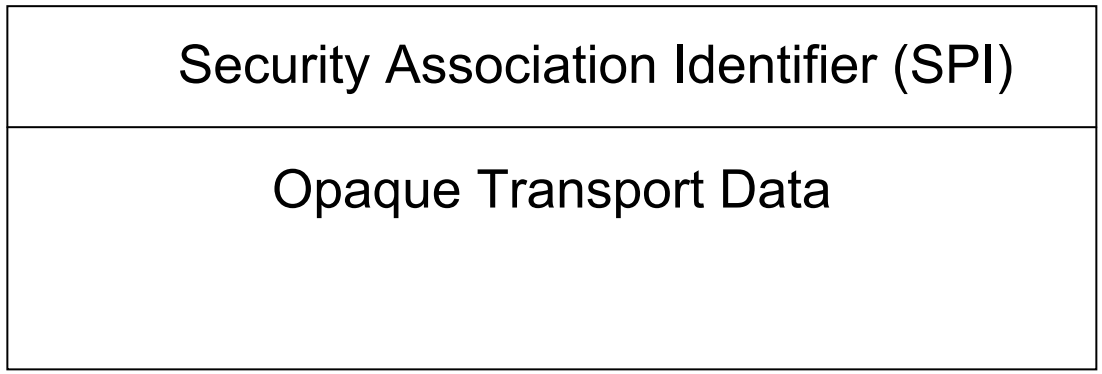
- Autentizační hlavička slouží k zajištění původu dat, integrity dat a chrání vůči útoku přehráním. Je použit MAC kombinovaný se sekvenčním číslem.

IPsec - AH


Původní IP datagram
IP hlavička
Ostatní hlavičky a data
Tajný klíč
128 bitů
Hašovací funkce (MD5 nebo SHA-1)
Autentizovaný IP datagram
IP hlavička
AH
Ostatní hlavičky a data

IPsec - ESP

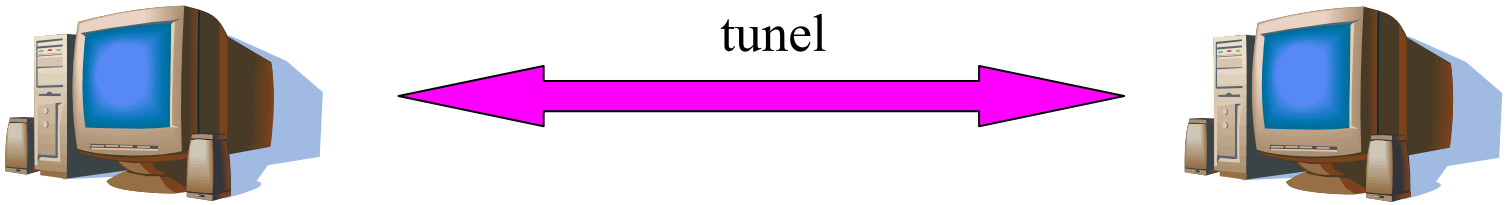
- Encapsulated Security Payload (ESP) header



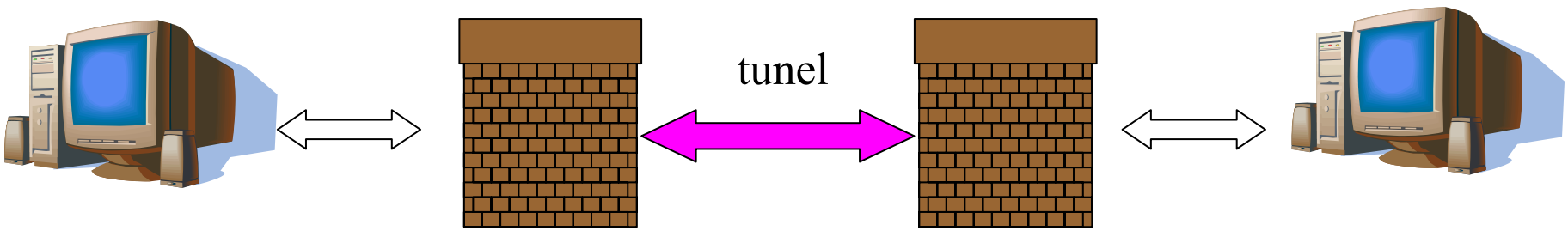
- EPS zajišťuje integritu a autenticitu dat, brání útokům přehráním a zajišťuje **důvěrnost dat**. Je použit symetrický šifrovací klíč sdílený oběma komunikujícími stranami.

Režimy IPsec

- Transportní režim (end-to-end)



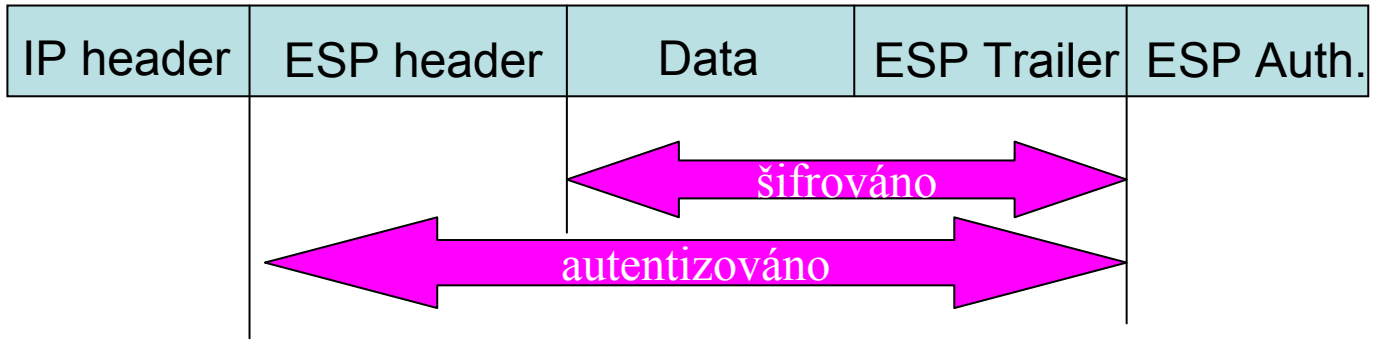
- Tunelovací režim (firewall-to-firewall)



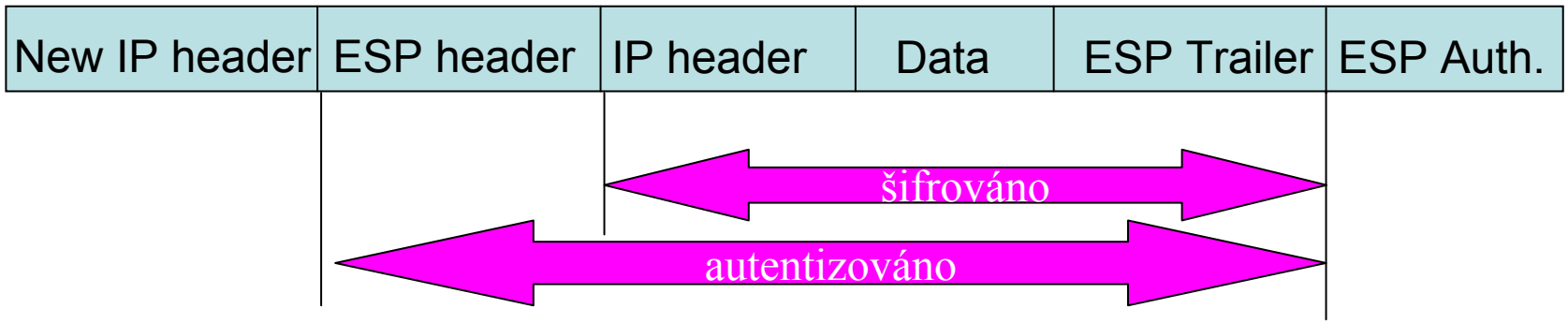
IPsec

- Standardní IP:

- Režimy provozu IPsec
  - Transportní režim (point-to-point)



  - Tunelovací režim



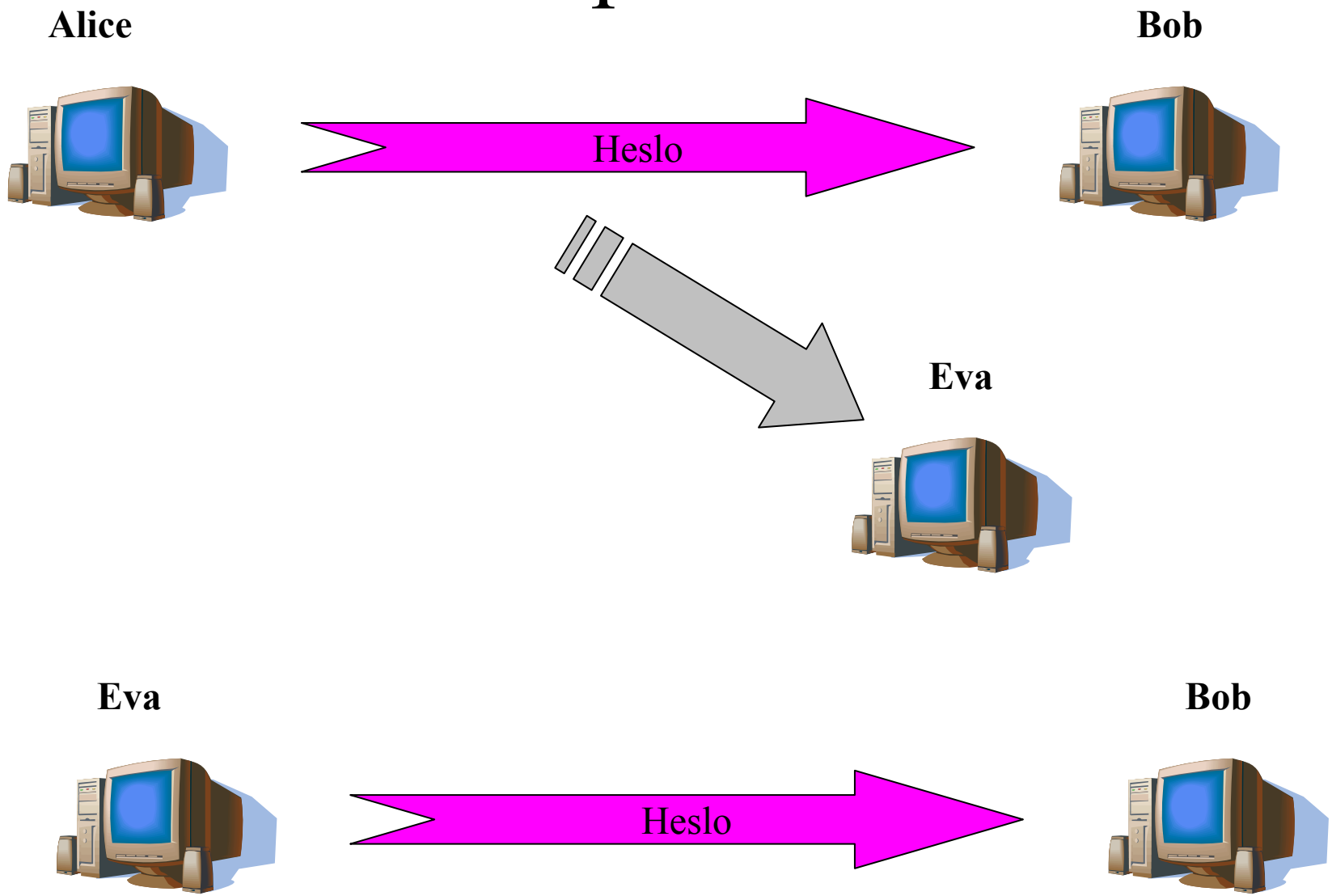
IPsec – správa klíčů

- Oakley
  - protokol pro ustavení společného klíče
  - založen na protokolu Diffie-Hellman, ale:
    - strany jsou autentizovány (brání man-in-the-middle útoku)
      - sdílené klíče, dohodnuté předem
      - Veřejné klíče DNS (viz DNSSEC)
      - RSA klíče podle PGP
      - RSA klíče včetně certifikátu podle X.509
      - DSS klíče včetně certifikátu podle X.509
    - pomocí časově proměnných parametrů se brání útokům přehráním
    - pomocí tzv. cookies se brání útokům typu „DoS“ (prováděné výpočty jsou totiž časově náročné)
    - umožňuje dohodu na použité grupě
- ISAKMP
  - framework (nezávislý na konkrétních šifrovacích algoritmech) pro správu klíčů a bezpečnostních atributů

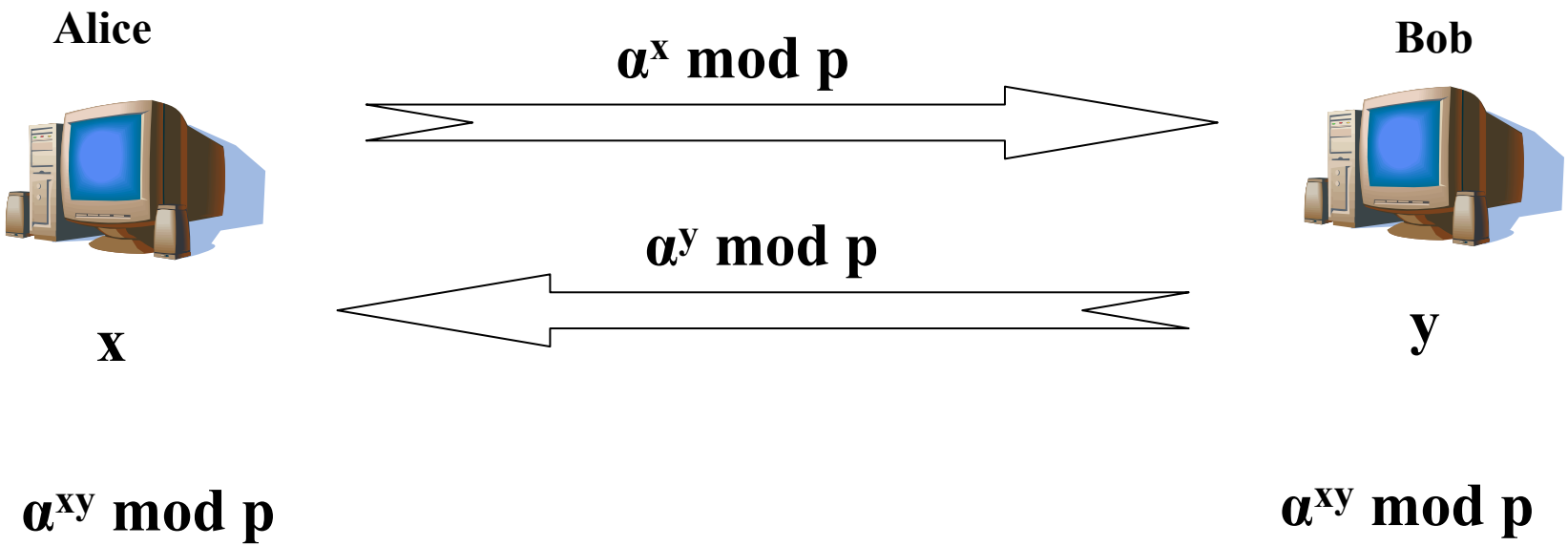
Útoky

- **Pasivní útočník** – analyzuje odchycená šifrovaná data
- **Aktivní útočník** – modifikuje data a/nebo vytváří nové zprávy
- **Zosobnění** (impersonation) – jedna strana se vydává za stranu jinou
- **Přehrání** (replay attack) – využití dříve poslané informace
- **Odraz** (reflection attack) – využití odeslané zprávy k okamžitému poslání odesilateli
- **Volený text** (chosen-text attack) – vhodné volení výzev (v protokolech výzva-odpověď) pro získání dlouhodobého klíče

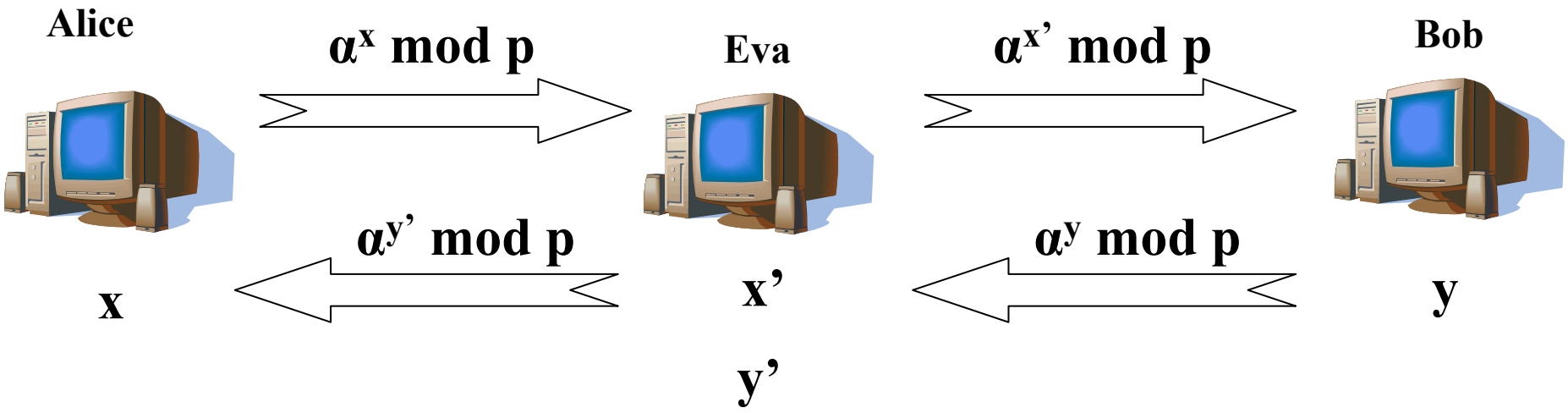
Útok přehráním



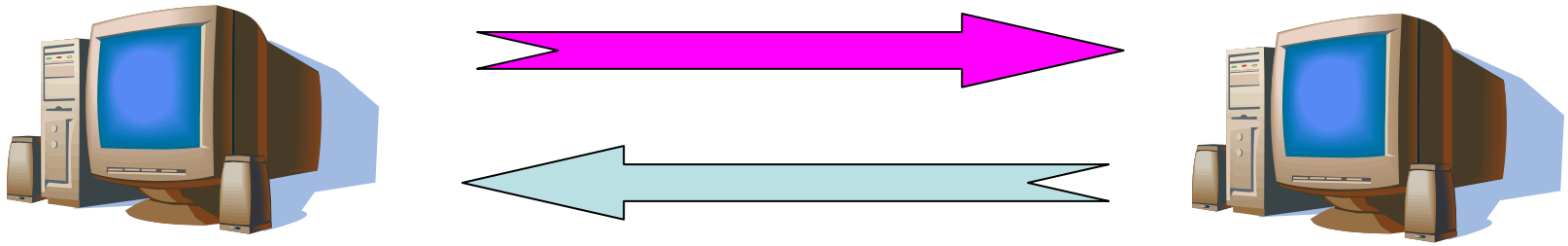
Protokol Diffie-Hellman (opak.)



Útok „Man in the middle“



Autentizace počítačů



Autentizace počítačů

- Na základě adresy počítače
  - MAC
  - IP
- Na základě tajné informace
  - Symetrická kryptografie
  - Asymetrická kryptografie

Autentizace podle adresy

- Autentizace na základě (síťové) adresy
  - MAC adresa ethernetové síťové karty
    - Přepínače (switch)
    - Svázání portu přepínače pouze s určitou MAC adresou
    - Svázání IP adresy pouze s určitou MAC adresou
  - IP adresa počítače
    - Řízení přístupu k síťovým službám (přístup k webovým stránkám na základě IP adresy)
    - Paketové filtry (součást firewallů) pracují na základě IP adresy a čísla portu odesilatele a příjemce (zdroj a cíl)

Autentizace podle adresy

- Úroveň bezpečnosti autentizace podle adresy
  - MAC adresy nejsou tajné (viz např. protokol/příkaz ARP)
  - MAC adresu ethernetové karty lze jednoduše změnit
  - IP adresu lze změnit
  - Je možné nesprávně uvést zdrojovou adresu (odesilatele) – IP spoofing
  - ! ! ! Automatické reakce na útoky (datagramy) s nesprávnou zdrojovou adresou (např. firewall odřeže přístup z určité domény)

Soubor .rhosts

- Soubor .rhosts
  - Nastavuje unixový uživatel se svém domovském adresáři (např. /home/zriha/.rhosts)
  - Globální důvěra: soubor /etc/hosts.equiv
  - Povoluje kdo může jeho účet používat (protokoly rlogin, rsh, rexec, …)
  - Nahrazuje autentizaci heslem (např. protokolem telnet)
  - Formát řádků: stroj [login]
    např. queen.math.muni.cz
    aisa.fi.muni.cz
    krusty.math.muni.cz riha
  - Uvedeným strojům důvěřujeme (že správně uvedou uživatelské jméno)
  - Možné útoky: počítač neuvede správně login uživatele, DNS, routing nebo IP spoofing

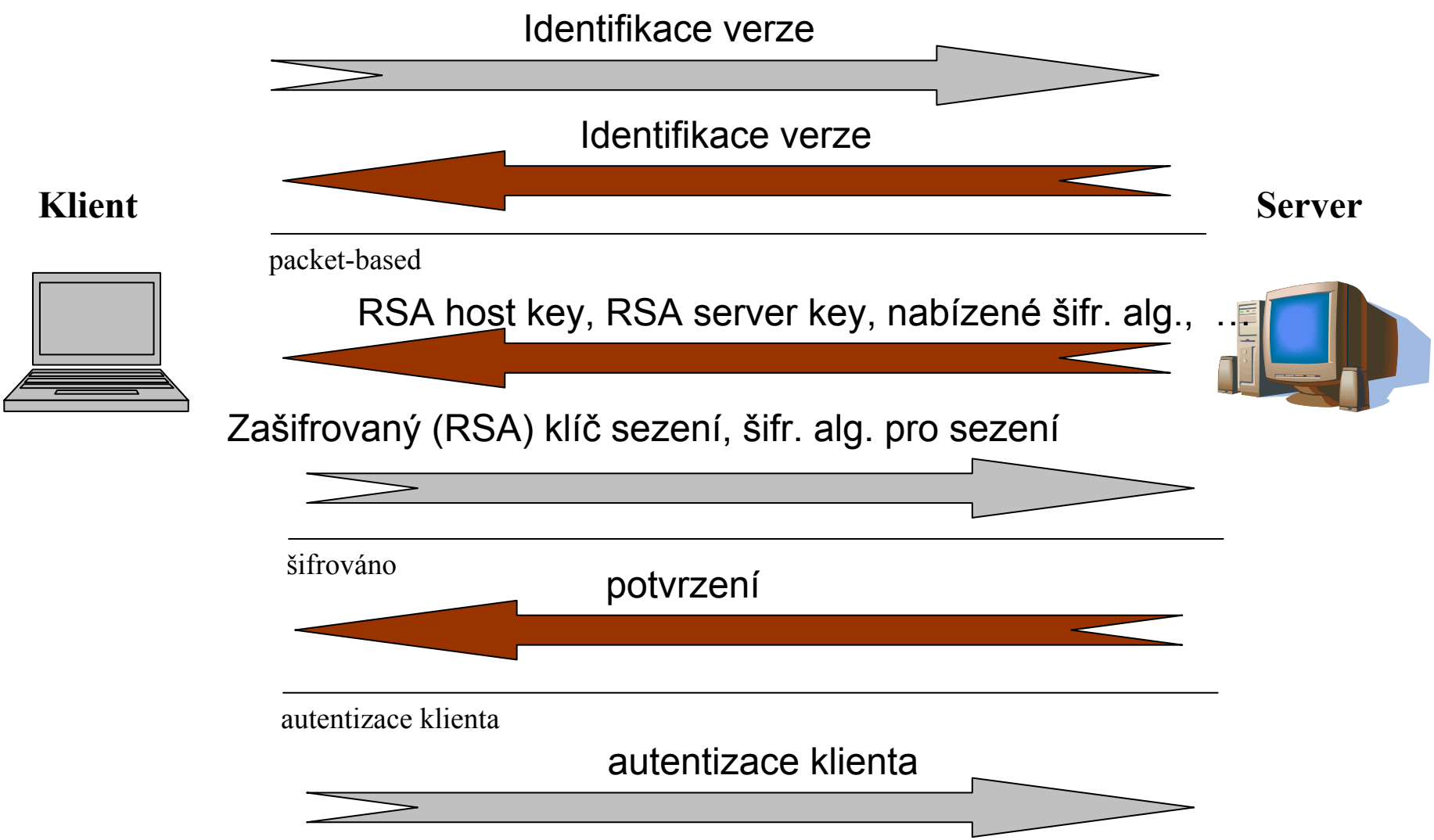
Autentizace na základě tajné informace

- Tajné informace
  - Hesla
  - Tajné symetrické klíče
  - Soukromé asymetrické klíče
- Jak tyto tajné informace ukládat?
  - Nešifrovaně v čisté podobě – počítač k nim má jednoduchý přístup, ale jsou přístupné i všem uživatelům s dostatečnými právy (hackeři)
  - Šifrovaně/chráněné heslem – při startu počítače (programu) je nutné manuálně zadat heslo/šifrovací klíč, slabé heslo znamená slabou ochranu, po celou dobu použití jsou tajné informace v paměti

Protokol ssh

- Protokol „secure shell“ (ssh)
- Slouží k přihlášení klienta (uživatele) k serveru
- Autentizace serveru i klienta
- Server
  - RSA host key (dlouhodobý)
  - RSA server key (generovaný každou hodinu)
- Metody autentizace klienta
  - .rhosts nebo /etc/hosts.equiv
  - .rhosts nebo /etc/hosts.equiv s RSA autentizací klienta (počítače)
  - RSA autentizace klienta (uživatele)
  - Heslo uživatele

Protokol ssh



Protokol ssh

- Šifrovací algoritmy pro šifrování sezení (klient vybírá z možností nabízených serverem)
  - 3DES (povinná podpora), ve verzi 1 i DES
  - AES - doporučené
  - Twofish - doporučené
  - Blowfish - doporučené
  - IDEA
  - Serpent
  - Arcfour
  - CAST128
- Šifrovací/podepisovací algoritmy pro autentizaci klienta/serveru
  - Od verze 2 je kromě RSA podporován i algoritmus DSA
- Obrana vůči útokům
  - Odposlech hesla a pozdější přehrání
  - DNS spoofing
  - IP spoofing
  - Routing spoofing

Protokol ssh: debug režim (ssh -v)

debug1: Connecting to aisa.fi.muni.cz [147.251.48.1] port 22.
debug1: Connection established.
debug1: identity file /home3/zriha/.ssh/identity type -1
debug1: Remote protocol version 1.99, remote software version OpenSSH_3.4p1
debug1: Local version string SSH-1.5-OpenSSH_3.1p1
debug1: Waiting for server public key.
debug1: Received server public key (768 bits) and host key (1024 bits).
debug1: Host 'aisa.fi.muni.cz' is known and matches the RSA1 host key.
debug1: Found key in /home3/zriha/.ssh/known_hosts:5
debug1: Encryption type: 3des
debug1: Sent encrypted session key.
debug1: Received encrypted confirmation.
debug1: Doing password authentication.
zriha@aisa.fi.muni.cz's password:
debug1: Requesting pty.
debug1: fd 3 setting TCP_NODELAY
debug1: Requesting shell.
debug1: Entering interactive session.

Asymetrické klíče pro autentizaci uživatele

- Soukromé klíče uživatele
  - ~/.ssh/identity
  - ~/.ssh/id_dsa
- Veřejné klíče uživatele
  - ~/.ssh/identity.pub
  - ~/.ssh/id_dsa.pub
- Vytvoření klíče: příkaz ssh-keygen

```
bash-2.05$ ssh-keygen -f /tmp/test -t rsa
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /tmp/test.
Your public key has been saved in /tmp/test.pub.
The key fingerprint is:
82:dd:71:7a:c4:ac:1c:de:b0:d3:d6:5b:63:7d:7c:76 zriha@queen.math.muni.cz
```

Protokol ssh

- Ověření integrity veřejného klíče serveru
- Soubory
  - /etc/ssh/known_hosts
  - /etc/ssh/known_hosts2
  - ~/.ssh/known_hosts
  - ~/.ssh/known_hosts2
  - Formát: počítač délka_klíče klíč
  - Např.: aisa 1024 37
    92648095391895266660461031814637345286469741285
    19463898291113200170437591638902829526627999663
    57470373079794594589737234564882145189758891946
    37391967788396230335631144998324780320375923657
    36181174418615708849459044374454744143100510826
    95360610857954348154578413482365924024485042273
    51129807154870221237653119

Protokol ssh

- Autentizace pomocí RSA/DSA klíče
  - Soubor
    - ~/.ssh/authorized_keys
    - ~/.ssh/authorized_keys2
  - Soubor obsahuje veřejné klíče uživatele(ů)
  - Obdoba .rhosts, ale pro silnou autentizaci
- Autentizační agent
  - ssh-agent
  - Zadám passphrase jen jednou
  - Agent uloží klíč do paměti
  - Následné autentizační požadavky řeší agent

Autentizace …

- Autentizace zpráv, protokoly
  - Postaveny na kryptografii
  - Redukce problémů na ochranu kryptografických klíčů
- Autentizace uživatelů
  - Hesla a PINy jsou obdobné (velmi slabé „klíče“)
  - Další možnosti (tokeny, biometriky)
  - viz následující blok přednášek

Otázky?

Příští přednáška: 13. 3. 2006 14:00

matyas@fi.muni.cz & zriha@fi.muni.cz