Protokoly pro ustanovení kryptografického klíče

Shkodran Gerguri

2010

Obsah prezentace

- Základní pojmy
- Model útočníka
- Protokoly pro transport klíče
- Protokoly pro dohodu klíče
- Další protokoly – ve zkratce

Kryptografické protokoly

- Algoritmy používající kryptografická primitiva pro dosažení nějakého (kryptografického) cíle
- Autentizační protokoly
- Protokoly pro ustanovení klíče
- Obsahem prezentace – vybrané protokoly pro ustanovení klíče (především) pro dvě strany

Kryptografický protokol

- Komunikující strany
- Lokální výpočty na straně účastníků
- Posílané zprávy a odpovědi na ně
- Obvyklé značení komunikujících – (A)lice, (B)ob, zprávy značené šipkami od odesilatele k příjemci
- Přenosové médium?

Komunikační kanál

- Abstraktní médium, konkrétní podoba ani charakteristiky nejsou předepisovány
- Nezabezpečený, pokud se neřekne jinak
- (Potenciální) útočník má plnou kontrolu nad kanálem, zprávy může odposlouchávat, modifikovat, měnit jejich pořadí, zahazovat, opakovaně přehrávat,…
- Model poprvé popsaný v r. 1978 Rogerem Needhamem a Michaelem Schroederem, spolu s jedním z prvních protokolů pro výměnu klíče a digitální podpis

Protokoly pro ustanovení klíče - motivace

- Kryptografie veřejného klíče je pomalejší, často vyžaduje dlouhé klíče (problém s kvalitním zdrojem entropie)
- Symetrická kryptografie oproti tomu rychlá, algoritmy často snadno implementovatelné pomocí základních logických funkcí, klíče podstatně kratší
- Dvě strany si chtějí poslat šifrovaný text, k tomu ovšem potřebují sdílet tajný klíč
- Řešení - protokol

Typy protokolů pro ustanovení klíče

- Protokoly pro transport klíče – klíč volí jedna strana, ostatním se pošle
- Protokoly pro dohodu klíče – na klíči se podílí více stran, výsledný klíč spočítá na základě vstupů každá strana samostatně
- Využití symetrické i asymetrické kryptografie
- Trusted Third Party (TTP) – důvěryhodná třetí strana, často přítomna v protokolech používajících symetrickou kryptografii

Speciální protokoly

- Key pre-distribution – „předdistribuce klíče“
- Secret sharing schemes – „rozdělení klíče“ do částí
- Conference/Group/Multiparty key distribution/agreement – protokoly pro ustanovení klíče ve skupině (příklad na závěr)

Charakteristiky protokolů

- Využití důvěryhodné třetí strany
- Autentizace klíče
- Potvrzení klíče
- Explicitní autentizace klíče
- Druhotné charakteristiky – autentizace účastníka, čerstvost klíče, kontrola nad klíčem, obecné charakteristiky protokolu

Útoky na protokoly

- Aktivní útoky
- Pasivní útoky – pouze odposlech a následné výpočty nad získanými daty
- Speciální kategorie aktivních útoků – Man-in-the-middle, replay, relay, reflexion,…
- Kompromitování dlouhodobého klíče (perfect forward secrecy)
- Kompromitování dřívějšího session klíče (known-key attack)

Protokoly pro transport klíče

- Shamirův bezklíčový protokol
- Kerberos
- X.509 Strong Two-Way Authentication

Shamirův protokol bez klíče

- Strany nesdílí žádnou tajnou informaci, nemají veřejné klíče
- Veřejné parametry – prvočíslo *p*, určuje multiplikativní grupu pro modulární aritmetiku

Shamirův bezklíčový protokol

Protocol 1 Shamir's No-Key Protocol

Overview: Party A transfers to party B secret key K in three messages.

Initial setup:

- Choose a random prime p such that the discrete logarithm problem is computationally infeasible in $\mathbb{Z}_p^*$; make p public.
- A and B choose secret random numbers a and b, respectively, such that $1 \leq a, b \leq p-2$, and $\gcd(a, p-1) = \gcd(b, p-1) = 1$.
- A and B compute $a^{-1} \bmod (p-1), b^{-1} \bmod (p-1)$, respectively.

Protocol run:

$$
\begin{aligned}
& A : K, 1 \leq K \leq p-1; K^a \bmod p \\
\mathbf{A} \rightarrow \mathbf{B} & : K^a \bmod p && (1) \\
& B : (K^a)^b \bmod p \\
\mathbf{A} \leftarrow \mathbf{B} & : (K^a)^b \bmod p && (2) \\
& A : (K^{ab})^{a^{-1}} \bmod p \\
\mathbf{A} \rightarrow \mathbf{B} & : (K^{ab})^{a^{-1}} \bmod p && (3) \\
& B : (K^{aba^{-1}})^{b^{-1}} \bmod p
\end{aligned}
$$

Shamir - slabiny

- Parametry *a, b* de-facto dlouhodobé klíče, negarantuje perfect forward secrecy
- Chybí kontrola klíče – vnucení starého klíče
- Man-in-the-Middle útok – simultánní zahájení běhu protokolu s A i B
- Odolnost proti pasivním útokům založena na problému diskrétního logaritmu

Kerberos

- Protokol pro autentizaci uživatelů a ustanovení session keys
- Systém realizující protokol
- Protokol využívající TTP pro snížení celkového počtu klíčů (úplný graf -> hvězda)
- TTP předpřipraví data pro komunikující dvojici
- Předpřipravená data – ticket – časově omezena

Kerberos

Protocol 2 Kerberos

Overview: Party A (mutually) authenticates to party B with the help of TTP T and establishes a secret session key

Symbols – optional fields and messages denoted by (+):

E_k symmetric encryption algorithm with secret key k

n_A nonce chosen by A

t_A timestamp created by A using its local clock

k session key chosen by T for use by A and B

L ticket lifetime

$\text{ticket}_B \stackrel{def}{=} E_{K_{BT}}(k, A, L)$

$\text{auth} \stackrel{def}{=} E_k(A, t_A, A^+_{subkey})$

Initial setup:

- T establishes long-term secret symmetric keys with all parties.

Protocol run:

$$
\begin{aligned}
& A : N_A \\
\mathbf{A} \rightarrow \mathbf{T} &: A, B, n_A && (1)\\
& T : k, L = (begin, end);\ E_{K_{AT}}(k, N_A, L, B); \\
& \quad : \text{ticket}_B = E_{K_{BT}}(k, A, L) \\
\mathbf{A} \leftarrow \mathbf{T} &: \text{ticket}_B, E_{K_{AT}}(k, n_A, L, B) && (2)\\
& A : \text{decrypt message, verify data, save L};\ A, T_A, A^+_{subkey}; \\
& \quad : \text{auth} = E_k(A, T_A, A^+_{subkey}) \\
\mathbf{A} \rightarrow \mathbf{B} &: \text{ticket}_B, \text{auth} && (3)\\
& B : \text{decrypt ticket, check data, save subkey} \\
& (+)B : B_{subkey^+}; E_k(T_A, B^+_{subkey}) \\
(+)A \leftarrow B &: E_K(t_A, B^+_{subkey}) && (4)\\
& (+)A : \text{decrypt message, check data, save subkey}
\end{aligned}
$$

Kerberos - charakteristiky

- Oboustranná autentizace + potvrzení klíče
- Tickets mají nastavenu dobu platnosti -> stačí zprávy č. 3 a 4
- Není specifikováno, jak se odvodí klíč (záleží na aplikaci)
- Malé okno, ve kterém lze provést known-key attack (drift mezi hodinami dvou stran)
- Negarantuje perfect forward secrecy
- Podobné protokoly – Needham-Schroeder (bez doby platnosti ticketu), Bellare-Rogaway (MAC, TTP provede finální distribuci)

X.509 Strong Two-Way Authentication

- Protokol pro vzájemnou autentizaci s výměnou klíče
- Používá asymetrickou kryptografii
- Netradičně – podpis zašifrované zprávy (signature stripping), obrana zanořením důležitých informací do kryptotextu
- Protokol je plně symetrický – obsah zpráv je (strukturně) shodný

X.509 Two-Way Authentication

Protocol 3 X.509 strong two-way authentication

Overview: Party A mutually authenticates to party B with (mutual) key establishment and key authentication

Symbols – optional fields in messages are denoted by $(+)$:

$P_X(y)$ encryption of y using public key of party X

$S_X(y)$ signature on y using private key of party X

r_A, r_B unique, non-repeatable numbers

$cert_X$ public-key certificate of party X

$D_A = (t_A, r_A, B, data_1^+, P_B(k_1)^+)$

$D_B = (t_B, r_B, A, data_2^+, P_A(k_2)^+)$

Initial setup:

- Each party has a public/private keypair for encryption and signature generation
- A obtained and authenticated B's public key prior to protocol run

Protocol run:

$$A : t_A, r_A, k_1^+, data_1^+$$

$$\mathbf{A} \rightarrow \mathbf{B} : cert_A, D_A, S_A \quad (1)$$

$$B : \text{verify } cert_A, \text{check signature, check data} \stackrel{?}{\Rightarrow} t_B, r_B, k_2^+, data_2^+$$

$$\mathbf{A} \leftarrow \mathbf{B} : cert_B, D_B, S_B \quad (2)$$

$$A : \text{verify } cert_B, \text{check signature, check data}$$

X.509 Two-Way Authentication

- Použití časových značek a neopakovatelných čísel pro ochranu proti přehrání
- Neposkytuje perfect forward secrecy (dlouhodobé klíče svázané s certifikátem)
- Podobný protokol – Needham-Schroeder, verze s veřejným klíčem

Protokoly pro dohodu klíče

- Diffie-Hellman Key Agreement
- MTI/A0 Two-Pass Key Agreement
- Station-to-Station Key Agreement

Diffie-Hellman

- Založeno na praktické nerealizovatelnosti výpočtu diskrétního logaritmu ve velkých grupách
- Veřejné parametry – náhodné prvočíslo *p*, generátor α grupy Z_p^*

Diffie-Hellman

Protocol 5 Diffie-Hellman key agreement

Overview: A and B exchange secret inputs and compute shared secret key K

Initial setup:

- Choose random prime p and generator α of $\mathbb{Z}_p^*$.

Protocol run:

$$
\begin{aligned}
& A : x,\ 1 \le x \le p-2 \\
\mathbf{A} \rightarrow \mathbf{B} &: \alpha^x \bmod p \qquad (1) \\
& B : y,\ 1 \le y \le p-2 \\
\mathbf{A} \leftarrow \mathbf{B} &: \alpha^y \bmod p \qquad (2) \\
& B : K = (\alpha^x)^y \\
& A : K = (\alpha^y)^x
\end{aligned}
$$

Diffie-Hellman

- Absence klíčů a tajných informací
- Minimální počet zpráv pro dohodu klíče
- Zajišťuje perfect forward secrecy
- Odolný vůči known-key attack
- Man-in-the-Middle útok
- Odvozené protokoly - ElGamal key agreement, Diffie-Hellman key pre-distribution

MTI/A0

- Podobné Diffie-Hellmanovi, ale používá speciální „dlouhodobé klíče“
- Zajištuje perfect forward secrecy, odolné vůči MITM
- Known-key útoky – paralelní běhy, Burmesterova triangulace

MTI/A0

Protocol 6 MTI key agreement protocol (A0 variant)

Overview: A and B exchange secret inputs and compute shared secret key K

Initial setup:

- Choose random prime p and generator α of $\mathbb{Z}_p^*$; $2 \leq \alpha \leq p-2$.
- A selects a, $1 \leq a \leq p-2$, and computes $z_A = \alpha^a \bmod p$. z_A is A's long-term public key.
- B selects b, $1 \leq b \leq p-2$, and computes $z_B = \alpha^b \bmod p$. z_B is B's long-term public key.

Protocol run:

$$A : x, 1 \leq x \leq p-2$$

$$\mathbf{A} \rightarrow \mathbf{B} : \alpha^x \bmod p \quad (1)$$

$$B : y, 1 \leq y \leq p-2$$

$$\mathbf{A} \leftarrow \mathbf{B} : \alpha^y \bmod p \quad (2)$$

$$A : k = (\alpha^y)^a z_B^x \bmod p = \alpha^{bx+ay} \bmod p$$

$$B : k = (\alpha^x)^b z_A^y \bmod p = \alpha^{bx+ay} \bmod p$$

Station-to-Station

Protocol 7 Station-to-Station key agreement protocol

Overview: A and B exchange secret inputs and compute shared secret key K, achieving mutual entity and explicit key authentication

Symbols:

$E_k(y)$ encryption of y using key k with algorithm E

$S_X(y)$ X's signature on data y

Initial setup:

- Choose random prime p and generator α of $\mathbb{Z}_p^*$; $2 \leq \alpha \leq p-2$.
- A creates its public/private keypair.
- B creates its public/private keypair.

Protocol run:

$$
\begin{aligned}
& A : x,\ 1 \leq x \leq p-2 \\
\mathbf{A} \rightarrow \mathbf{B} & : \alpha^x \bmod p && (1) \\
& B : y,\ 1 \leq y \leq p-2;\ \alpha^y \bmod p; \\
& \quad : k = (\alpha^x)^y \bmod p;\ E_k(S_B(\alpha^y, \alpha^x)) \\
\mathbf{A} \leftarrow \mathbf{B} & : \alpha^y \bmod p,\ E_k(S_B(\alpha^y, \alpha^x)) && (2) \\
& A : k = (\alpha^y)^x \bmod p;\ \text{decrypt \& verify signature}; \\
& \quad : \text{signature correct} \stackrel{?}{\Rightarrow} E_k(S_A(\alpha^x, \alpha^y)) \\
\mathbf{A} \rightarrow \mathbf{B} & : k = E_k(S_A(\alpha^x, \alpha^y)) && (3) \\
& B : \text{decrypt \& verify signature}
\end{aligned}
$$