

# Ochrona i poufność danych

Maciej Nikodem

10.01.2011

- "claimant" nie ujawnia żadnej informacji na temat sekretu,
- wykonanie protokołu nie zwiększa wiedzy weryfikującego,
- weryfikujący nie potrafi powtórzyć protokołu wobec innych stron,
- jedyną informacją przekazywaną weryfikującemu jest "Tak, claimant zna sekret".

## Cel protokołu z wiedzą zerową

Udowadniający (ang. *prover*) musi przekonać weryfikującego (ang. *verifier*) o prawdziwości twierdzenia.

- dowody probabilistyczne,
- właściwość kompletności (ang. *completeness*)
  - uczciwy *prover* i *verifier*,
  - prawdopodobieństwo graniczące z pewnością poprawnej identyfikacji/autentykacji,
- właściwość trafności/poprawności (ang. *soundness*),
- właściwość wiedzy zerowej (ang. *zero-knowledge*),
  - uczciwy *prover* i nieuczciwy *verifier*,
  - wykonanie protokołu nie dostarcza wiedzy *verifier*'owi pozwalającej na podszycie się pod *prover*'a

- bezpieczeństwo mimo wielokrotnego stosowania,
- brak szyfrowania,
- większy narzut komunikacyjny i/lub obliczeniowy,
- podobne (te same) podstawy bezpieczeństwa,

Elementy dowodu (Alicja identyfikuje się wobec Boba):

- sekret i opcjonalna informacja publiczna,
- świadek (ang. *witness*)  $(A \rightarrow B)$ ,
- wyzwanie (ang. *challenge*)  $(A \leftarrow B)$ ,
- odpowiedź (ang. *response*)  $(A \rightarrow B)$ .

## Setup

- publiczna i wiarygodna informacja o  $n = pq$ ,
- sekret Alicji –  $s$ , informacja publiczna  $v = s^2 \bmod n$ ,

## Identyfikacja

- Alicja losuje  $1 \leq r \leq n - 1$ , i oblicza świadka  $x = r^2 \bmod n$ ,

$$A \rightarrow B : x$$

- Bob losuje  $e = \{0, 1\}$ ,

$$A \leftarrow B : e$$

- Alicja oblicza  $y = rs^e \bmod n$

$$A \rightarrow B : y$$

- Bob sprawdza czy  $y \neq 0$  i  $y^2 \bmod n = xv^e \bmod n$ .

# Nieuczciwy "prover"

Jeśli prover wylicza  $x = r^2/v \bmod n$  i  $y = r$ , to

$$y^2 \bmod n = \frac{r^2}{v} v^e \bmod n$$

i jeśli verifier wybrał  $e = 1$  to dowód zakończony pomyślnie.

## Nieuczciwy "prover"

Jeśli prover wylicza  $x = r^2/v \bmod n$  i  $y = r$ , to

$$y^2 \bmod n = \frac{r^2}{v} v^e \bmod n$$

i jeśli verifier wybrał  $e = 1$  to dowód zakończony pomyślnie.

Jeśli prover wylicza  $x = r^2 \bmod n$  i  $y = r$  to

$$y^2 \bmod n = r^2 v^e \bmod n$$

i jeśli verifier wybrał  $e = 0$  to dowód zakończony pomyślnie.

## Nieuczciwy "prover"

Jeśli prover wylicza  $x = r^2/v \bmod n$  i  $y = r$ , to

$$y^2 \bmod n = \frac{r^2}{v} v^e \bmod n$$

i jeśli verifier wybrał  $e = 1$  to dowód zakończony pomyślnie.

Jeśli prover wylicza  $x = r^2 \bmod n$  i  $y = r$  to

$$y^2 \bmod n = r^2 v^e \bmod n$$

i jeśli verifier wybrał  $e = 0$  to dowód zakończony pomyślnie.

Prover musi wybrać odpowiednie  $x$  zanim pozna  $e$  – musi zgadywać jakie będzie  $e$  a więc z prawdopodobieństwem  $1/2$  przeprowadzi protokół z sukcesem.

- jeśli  $e = 0$  to verifier nie dostaje żadnej informacji o  $s$ ,
- jeśli  $e = 1$  to verifier dostaje  $y = rs \bmod n$  ale nie zna  $r$  (zna tylko  $r^2 \bmod n$ ), więc nie potrafi odszukać  $s$ .

# Protokoły uzgadniania kluczy kryptograficznych

- zmniejszenie liczby szyfrogramów generowanych jednym kluczem,
- ograniczenia czasu stosowania i powtarzalności klucza,
- tworzenie kluczy tylko gdy są potrzebne,
- $\approx$  klucze jednorazowe.

- współdzielenie/obliczenie/przesłanie kluczy szyfrujących,
- uzgodnienie klucza pomiędzy  $\geq 2$  stronami,
- wykorzystują:
  - szyfrowanie – poufność i autentykacja,
  - podpisy cyfrowe – autentykacja,
  - funkcje skrótu – wyznaczanie kluczy, integralność danych,
  - współdzielenie sekretu – wyznaczanie kluczy.

Strony biorące udział w protokole

- bez zaufanej trzeciej strony (ang. *Trusted Third Party*),
- z zaufaną trzecią stroną.

Strony biorące udział w protokole

- bez zaufanej trzeciej strony (ang. *Trusted Third Party*),
- z zaufaną trzecią stroną.

Uwierzytelnianie (ang. *authentication*)

- nie zapewniające autentykacji,
- zapewniające autentykację jednej z komunikujących się stron – unilateral,
- zapewniające obustronną autentykację – mutual.

## Współdzielony sekret

- bez współdzielonego sekretu – no-key protocols,
- z współdzielonym sekretem.

## Współdzielony sekret

- bez współdzielonego sekretu – no-key protocols,
- z współdzielonym sekretem.

## Tworzenie klucza

- z kluczem wyznaczanym przez jedną ze stron – key transport,
- pozwalające wyznaczyć klucz przez obie strony – key agreement,
- z kluczem predefiniowanym – key pre-distribution,
- z kluczem wyznaczanym dynamicznie – dynamic key.

- poufność,
- autentykacja użytkowników – zapewnienie, że druga strona jest tym za kogo się podaje,
- autentykacja klucza – zapewnienie, że druga strona zna właściwy klucz,
- "świeżość" i poprawność klucza,

## Przesyłanie klucza – key transport

# Współdzielony sekret – klucz symetryczny

$$A \leftarrow B : n_B$$

$$A \rightarrow B : E_K(r_A, n_A, n_B, B)$$

$$A \leftarrow B : E_K(r_B, n_B, n_A, A)$$

$$A \leftarrow B : n_B$$

$$A \rightarrow B : E_K(r_A, n_A, n_B, B)$$

$$A \leftarrow B : E_K(r_B, n_B, n_A, A)$$

Cechy:

- challenge–response,
- wzajemna autentykacja,
- klucz może być wyznaczony na podstawie  $r_A$  i  $r_B$ ,
- brak TTP,
- wymaga sprawdzania integralności przesyłanych danych,
- $\binom{n}{2}$  kluczy w sieci  $n$  użytkowników;  $n - 1$  kluczy u każdego użytkownika.

# Shamir no-key protocol

$$A \leftarrow B : k_1 = k^a \bmod p$$

$$A \rightarrow B : k_2 = k_1^b \bmod p = k^{ab} \bmod p$$

$$A \leftarrow B : k_3 = k_2^{a^{-1}} \bmod p = (k^{ab})^{a^{-1}} \bmod p = k^b \bmod p$$

$$B : k = k_3^{b^{-1}} \bmod p$$

# Shamir no-key protocol

$$A \leftarrow B : k_1 = k^a \bmod p$$

$$A \rightarrow B : k_2 = k_1^b \bmod p = k^{ab} \bmod p$$

$$A \leftarrow B : k_3 = k_2^{a^{-1}} \bmod p = (k^{ab})^{a^{-1}} \bmod p = k^b \bmod p$$

$$B : k = k_3^{b^{-1}} \bmod p$$

Cechy:

- publiczny parametr  $p$ ,
- $a, b$  takie, że  $\text{nwd}(a, p - 1) = \text{nwd}(b, p - 1) = 1$ ,
- duża złożoność obliczeniowa,
- można użyć **dowolnego**, przemiennej algorytmu szyfrowania, tj:

$$D_{K_1}(D_{K_2}(E_{K_1}(E_{K_2}(m)))) = m.$$

$A$             losowy klucz sesji  $k$

$A \leftarrow B$  :  $m_1 = k \oplus a$

$A \rightarrow B$  :  $m_2 = m_1 \oplus b = k \oplus a \oplus b$

$A \leftarrow B$  :  $m_3 = m_2 \oplus a = k \oplus b$

$B$             oblicza klucz sesji  $k = m_3 \oplus b$

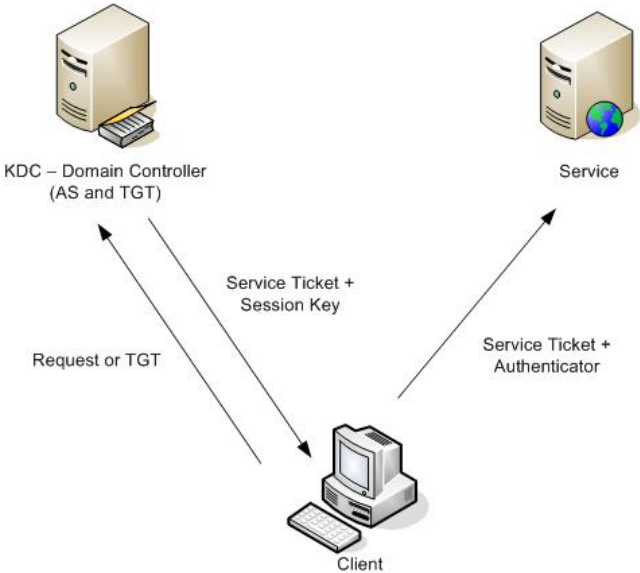
$A$             losowy klucz sesji  $k$   
 $A \leftarrow B$  :  $m_1 = k \oplus a$   
 $A \rightarrow B$  :  $m_2 = m_1 \oplus b = k \oplus a \oplus b$   
 $A \leftarrow B$  :  $m_3 = m_2 \oplus a = k \oplus b$   
 $B$             oblicza klucz sesji  $k = m_3 \oplus b$

Czy taki protokół jest bezpieczny?

# Pies z trzema głowami, czyli ...



# Kerberos



- 3 strony komunikacji – A, B i zaufana trzecia strona (TTP),
- TTP współdzieli sekret z każdą z pozostałych stron,
- A i B nie współdzielą żadnego sekretu,
- TTP umożliwia:
  - autentykację A wobec B i odwrotnie,
  - ustalenie klucza sesji.

- 3 strony komunikacji – A, B i zaufana trzecia strona (TTP),
- TTP współdzieli sekret z każdą z pozostałych stron,
- A i B nie współdzielą żadnego sekretu,
- TTP umożliwia:
  - autentykację A wobec B i odwrotnie,
  - ustalenie klucza sesji.

$A \rightarrow TTP : A, B, N_A$

$A \leftarrow TTP : E_{K_{BT}}(k, A, L), E_{K_{AT}}(k, N_A, L, B)$

$A \rightarrow B : E_{K_{BT}}(k, A, L), E_k(A, T_A)$

$A \leftarrow B : E_k(T_A)$

## Uzgadnianie klucza – key agreement

$$A \leftarrow B : n_B$$

$$A \rightarrow B : E_K(r_A, n_A, n_B, B)$$

$$A \leftarrow B : E_K(r_B, n_B, n_A, A)$$

$$A \leftarrow B : n_B$$

$$A \rightarrow B : E_K(r_A, n_A, n_B, B)$$

$$A \leftarrow B : E_K(r_B, n_B, n_A, A)$$

Wyznaczenie klucza sesji  $k$ :

①  $k = r_A | r_B$

②  $k = r_A \oplus r_B,$

③  $k = r_B = k' \oplus r_A,$

$$A \leftarrow B : n_B$$

$$A \rightarrow B : E_K(r_A, n_A, n_B, B)$$

$$A \leftarrow B : E_K(r_B, n_B, n_A, A)$$

Wyznaczenie klucza sesji  $k$ :

- 1  $k = r_A | r_B$
- 2  $k = r_A \oplus r_B$ ,
- 3  $k = r_B = k' \oplus r_A$ ,
- 4  $k = h(r_A, r_B)$ ,

$$\begin{aligned} A \rightarrow B & : P_B(k_1, A) \\ A \leftarrow B & : P_A(k_1, k_2) \\ A \rightarrow B & : P_B(k_2) \end{aligned}$$

$$A \rightarrow B : g^a \bmod p$$

$$A \leftarrow B : g^b \bmod p$$

$$A \rightarrow B : g^a \bmod p$$

$$A \leftarrow B : g^b \bmod p$$

- wspólny klucz –  $k = g^{ab} \bmod p$ , ale,
- żadna ze stron nie ma pewności, że druga wyliczyła ten sam klucz,
- protokół odporny na atak pasywny,
- barak autentykacji komunikujących się stron – protokół nie jest odporny na ataki aktywne.

$$A \rightarrow B : g^a \bmod p$$

$$A \leftarrow B : g^b \bmod p, E_k(\text{Sig}_B(g^a, g^b))$$

$$A \rightarrow B : E_k(\text{Sig}_A(g^a, g^b))$$

$$A \rightarrow B : g^a \bmod p$$

$$A \leftarrow B : g^b \bmod p, E_k(\text{Sig}_B(g^a, g^b))$$

$$A \rightarrow B : E_k(\text{Sig}_A(g^a, g^b))$$

- $\text{Sig}_A(x) = (H(x))^{d_a} \bmod n_a$ ,
- wspólny klucz  $k = g^{ab} \bmod p$ ,
- klucz potwierdzony poprzez zaszyfrowanie podpisów,
- atak aktywny nie jest możliwy, zakładając, że każda ze stron zna certyfikaty drugiej z nich.