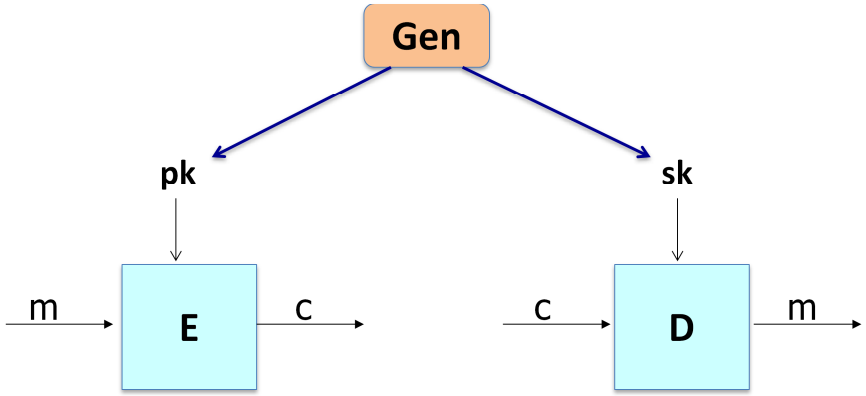
Kryptografia i bezpieczeństwo danych - Kryptografia klucza publicznego ElGamal

Sławomir Samolej
ssamolej.kia.prz.edu.pl
ssamolej@prz.edu.pl

Przypomnienie: kryptografia klucza publicznego (Gen, E, D)



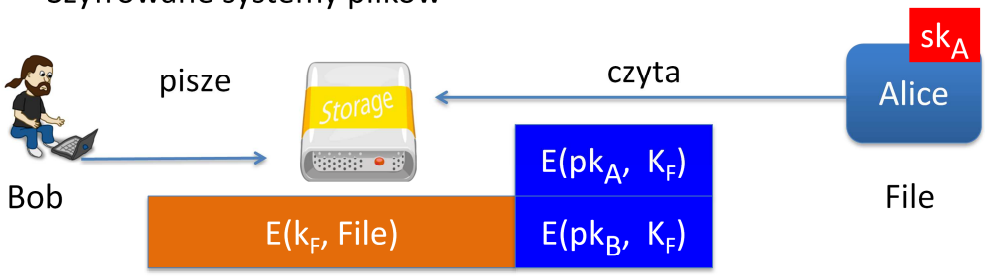
2

Zastosowania kryptografii z kluczem publicznym (1)

Wymiana kluczy (np. in HTTPS)

Szyfrowanie w nieinteraktywnym środowisku:

- Chroniony Email: Bob ma publiczny klucz Alice i wysyła do niej wiadomości
- Szyfrowane systemy plików



3

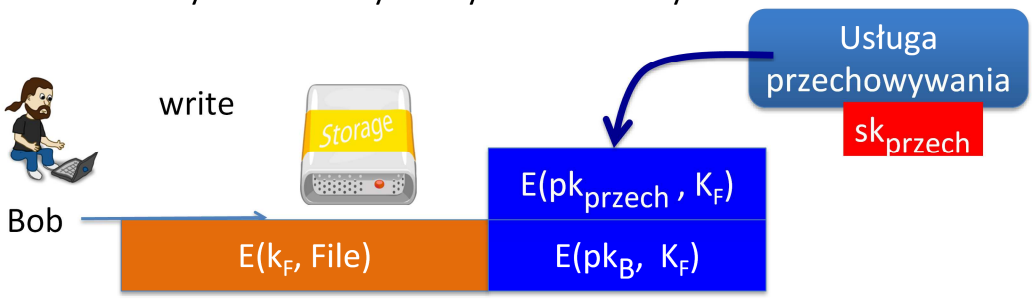
Przypomnijmy wspomniane wcześniej zastosowania kryptografii z kluczem publicznym. Po pierwsze może ona służyć do efektywnej chronionej wymiany kluczy symetrycznych w środowisku sieciowym (SSL) (serwer wysyła swój klucz publiczny do przeglądarki; przeglądarka wybiera klucz/sekret i szyfruje ten sekret kluczem publicznym serwera i odsyła do serwera; serwer odszyfrowuje wiadomość i teraz obie strony dysponują tylko sobie znanym sekretem.
Jeśli interakcja nie jest możliwa, to kryptografia z kluczem publicznym jest bezpośrednio stosowana do szyfrowania wiadomości. Przykładem jest tu szyfrowana poczta elektroniczna. Nadawca szyfruje kluczem publicznym odbiorcy dane, wiadomość przychodzi na skrzynkę pocztową i odbiorca po jej pobraniu może ją odszyfrować.
Okazuje się, że kryptografia z kluczem publicznym dobrze nadaje się do chronionego współdzielenia plików. Załóżmy, że Bob chce przechowywać zaszyfrowane pliki na dysku sieciowym. Generuje on klucz symetryczny kF i używa go do zaszyfrowania pliku. Potem szyfruje klucz kF z zastosowaniem swojego klucza publicznego. To daje Bobowi dostęp do klucza później, kiedy będzie go potrzebował. Po prostu z zastosowaniem swojego klucza prywatnego odszyfruje on klucz kF i będzie mógł odszyfrować plik. Jeśli bob będzie chciał udostępnić plik Alice, to może on go zaszyfrować kluczem publicznym Alice i wtedy ona będzie mogła go sobie odszyfrować swoim kluczem prywatnym.

Zastosowania kryptografii z kluczem publicznym (2)

Wymiana kluczy (np. in HTTPS)

Szyfrowanie w nieinteraktywnym środowisku:

- Chroniony Email: Bob ma publiczny klucz Alice i wysyła do niej wiadomości
- Szyfrowane systemy plików
- Przechowywanie kluczy: odzyskiwanie danych bez klucza Boba



4

Innym przykładem zastosowania kryptografii z kluczem publicznym w systemie nieinteraktywnym jest przechowywanie kluczy. Załóżmy, że Bob zamieścił dane na dysku i potem przestał być dostępny (może został zwolniony albo jest chory). Teraz przedsiębiorstwo chce uzyskać pliki Boba. Założenie, że Bob jest jedyną osobą mogącą odszyfrować swoje pliki jest niedopuszczalne z perspektywy firmy.
W zmodyfikowanym środowisku Bob, jak poprzednio szyfruje swoje pliki z zastosowaniem klucza symetrycznego, zabezpiecza ten klucz swoim kluczem publicznym, ale również zabezpiecza klucz kF innym kluczem, należącym do firmowej usługi przechowywania kluczy. Wtedy pod jego nieobecność istnieje możliwość uzyskania dostępu do zaszyfrowanych przez niego danych. Tutaj usługa przechowywania kluczy jest zupełnie offline. Ona tylko udostępnia swój klucz publiczny, a potem bez udziału Bob'a może odzyskać jego dane.

Konstrukcje

Dwie rodziny schematów z kluczem publicznym

- Wprowadzone: oparte na funkcjach zapadkowych (np. RSA)
  - Schematy: ISO standard, OAEP+, …
- Nowa część: oparte na protokole Diffie-Hellman'a
  - Schematy: szyfrowanie ElGamal i jego warianty (np. stosowane w GPG)

Poziom bezpieczeństwa: odporność na atak z wybranym szyfrogramem

5

Jak dotąd wprowadziliśmy również schemat szyfrowania z zastosowaniem kryptografii z kluczem publicznym oparty na funkcjach zapadkowych (RSA). Wprowadziliśmy ustandaryzowany schemat szyfrowania, oraz schematy stosowane w praktyce (PKCS). Teraz zostanie wprowadzony schemat szyfrowania z kluczem publicznym oparty o protokół Diffie-Hellman'a.

Przypomnienie protokołu Diffie-Hellman'a (1977)

Ustal skończoną grupę cykliczną G (np. $G = (Z_p)^*$) o rzędzie n

Ustal generator g in G (taki, że $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)

Alice **Bob**

Wybierz losowy **a** in {1,…,n} Wybierz losowy **b** in {1,…,n}

$\mathbf{B^a} = (g^b)^a = \mathbf{k_{AB} = g^{ab}} = (g^a)^b = \mathbf{A^b}$

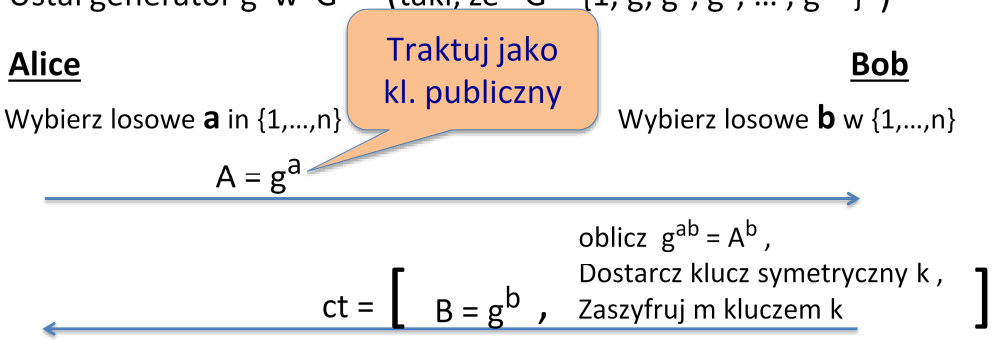
6

Protokół działa w następujący sposób. Alicja wybiera losową liczbę całkowitą z zakresu 1 do n-1. Wtedy oblicza g^a. Przypisuje otrzymany wynik do zmiennej A. Wartość A jest wysyłana bo Bob'a. Bob robi to samo. Wybiera liczbę losową z przedziału 1 do n i oblicza g^b. Przypisuje otrzymany wynik do zmiennej B. Wartość B jest wysyłana do Alice. Tajny klucz wynosi g^{ab}.
Atakujący widzi g^a i g^b i g i ma wyliczyć g^{ab}, co okazuje się złożonym obliczeniowo problemem.

ElGamal: zamiana na system z kluczem publicznym (1984)

Ustal skończoną grupę cykliczną G (np. $G = (Z_p)^*$) o rzędzie n

Ustal generator g w G (taki, że $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)



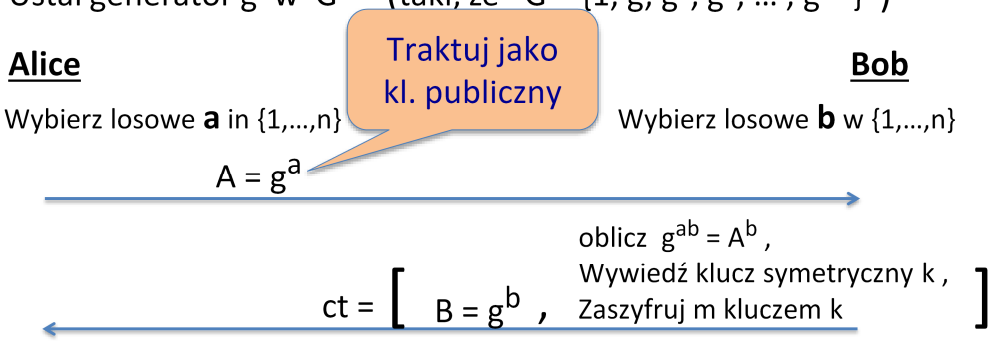
7

Protokół Diffiego-Hellmann'a można przekształcić w system szyfrowania z kluczem publicznym. Podobnie, jak w przypadku protokołu Diffie-Helmann'a dysponujemy skończoną grupą cykliczną i znalezioną w niej wartością generatora. Koncepcję wymiany klucza zastępujemy akcjami oddzielonymi w czasie. Akcja wykonana przez Boba nie następuje natychmiast w odpowiedzi na klucz wysłany przez Alice. Pierwszy etap protokołu jest generowaniem klucza A. Duże **A** będzie rozumiane jako klucz publiczny a małe **a** będzie kluczem prywatnym. Zauważmy, że odzyskanie małego a z dużego A, to problem wyznaczenia dyskretnego logarytmu, który jest uważany za trudny. Jeśli Bob chce później zaszyfrować wiadomość do Alice, to oblicza wartość „swojej połówki" protokołu Diffiego-Hellmann'a, czyli $B=g^b$. Oblicza wartość g^{ab}, z tej wartości wyprowadza klucz szyfrowania symetrycznego k, a następnie szyfruje wiadomość m z zastosowaniem tego klucza. Ostatecznie „odsyła" do Alice swój wkład w protokół Diffiego-Helmann'a (g^b) oraz zaszyfrowaną wiadomość m kluczem k wywiedzionym z wartości g^{ab}.

ElGamal: zamiana na system z kluczem publicznym (1984)

Ustal skończoną grupę cykliczną G (np. $G = (Z_p)^*$) o rzędzie n

Ustal generator g w G (taki, że $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)



Aby odszyfrować:
oblicz $g^{ab} = B^a$,
wywiedź k i odszyfruj

8

Alice, żeby odszyfrować wiadomość również oblicza g^{ab}, stosuje te wartość do obliczenia klucza k, który to z kolei może posłużyć do odszyfrowania wiadomości.
Proszę zwrócić uwagę na interesujące właściwości tak opracowanego protokołu. Po pierwsze pokazany schemat jest schematem zrandomizowanym. Za każdym razem wartość b jest losowana na nowo.

System ElGamal (współczesne podejście) (1)

- G: skończona grupa cykliczna rzędu n
- (E_s, D_s) : symetryczny system kryptograficzny z uwierzytelnieniem (K,M,C)
- H: $G^2 \longrightarrow K$ funkcja hash (mieszająca)

Konstruujemy system szyfrowania z kluczem publicznym (Gen, E, D):

- Generator kluczy Gen:
  - Wybierz losowy generator **g** z G i losową **a** z Z_n
  - zwróć sk = a , pk = $(g, h=g^a)$

9

Bardziej formalnie. System szyfrowania z kluczem publicznym oparty na protokole wymiany kluczy Diffiego-Helmanna można opisać w następujący sposób. G jest skończoną grupą cykliczną rzędu n. (E_s, D_s) jest symetrycznym systemem kryptograficznym z uwierzytelnianiem, H jest funkcją mieszającą (hash) mapująca zbiór G^2 na zbiór kluczy K. Tworzony system jest złożony z 3 elementów (Gen, E, D). Gen jest generatorem kluczy. W ramach tego algorytmu jest wybierany losowy generator grupy g oraz losowa liczba a należąca do Zn. Kluczem sekretnym systemu jest wybrana liczba a, natomiast kluczem publicznym para wartości: $(g, h=g^a)$.

System ElGamal (współczesne podejście) (2)

- G: skończona grupa cykliczna rzędu n
- (E_s, D_s) : symetryczny system kryptograficzny z uwierzytelnieniem (K,M,C)
- H: $G^2 \longrightarrow K$: funkcja hash (mieszająca)

E(pk=(g,h), m) :

$b \xleftarrow{R} Z_n$, $u \longleftarrow g^b$, $v \longleftarrow h^b$

$k \longleftarrow H(u,v)$, $c \longleftarrow E_s(k, m)$

wyjście (u, c)

D(sk=a, (u,c)) :

$v \longleftarrow u^a$

$k \longleftarrow H(u,v)$, $m \longleftarrow D_s(k, c)$

wyjście m

10

Jeśli Bob chce zaszyfrować wiadomość, to bierze klucz publiczny złożony z wartości g i h. Losuje własną liczbę b. Następnie oblicza wyrażenie g^b, a potem oblicza sekret DH jako h^b. Kolejny krokiem jest obliczenie funkcji mieszającej (Hash) z wejściami u i v. Jej rezultatem jest symetryczny klucz k. Wiadomość jest z kolei szyfrowana z zastosowaniem klucza k. Ostatecznie Bob wysyła do Alice parę (u i c).
Alce odszyfrowuje wiadomość w następujący sposób: Podnosi wartość u do potęgi a i otrzymuje v. Oblicza wartość klucza k poprzez wyliczenie wyjścia funkcji mieszającej (hash) z parametrami u i v. Odszyfrowuje z zastosowaniem klucza k szyfrogram c i otrzymuje wiadomość m.

Wydajność ElGamal

E(pk=(g,h), m) :
$b \leftarrow Z_n$, $u \leftarrow g^b$, $v \leftarrow h^b$

D(sk=a, (u,c)) :
$v \leftarrow u^a$

Szyfrowanie: 2 potęgowania (o stałej podstawie)
- Można przygotować wcześniej
  [$g^{(2\hat{}i)}$, $h^{(2\hat{}i)}$ for $i=1,\ldots,\log_2 n$]
- Potrójne przyspieszenie (albo lepiej)

Odszyfrowywanie: 1 potęgowanie (różne podstawy)

11

Najbardziej czasochłonne operacje podczas szyfrowania to wyliczenie dwóch wartości potęg (kilka milisekund na współczesnych komputerach). Odszyfrowywanie to obliczenie jednej potęgi. Wyliczenie wartości potęgowania przy szyfrowaniu można przyspieszyć zauważając, że będziemy podnosić do potęgi zawsze tę samą podstawę. Można więc przygotować w tablicy kwadratowe potęgi g i h. W rezultacie szyfrowanie może odbywać się nawet szybciej niż odszyfrowywanie.

Założenie obliczeniowe systemu Diffie-Helmann'a

G: skończona grupa cykliczna rzędu n

Założenia obliczeniowe DH (CDH) zachodzą w G,

jeśli: $g,\ g^a,\ g^b \not\Rightarrow g^{ab}$

dla wszystkich efektywnych algorytmów A:

$$Pr[\ A(g, g^a, g^b) = g^{ab}\ ] < \text{pomijalne}$$

gdzie $g \longleftarrow \{\text{generatory } G\}$, $a, b \longleftarrow Z_n$

12

Bezpieczeństwo systemu DH opiera się na założeniu, że znając g, g^a i g^b trudno jest wyznaczyć g^{ab}.
Można udowodnić **semantyczne bezpieczeństwo** systemu szyfrowania z kluczem publicznym opartego na protokole DH.

Udowodnienie bezpieczeństwa tego systemu a atak z wybranym szyfrogramem wprost z założenia obliczeniowego DH jest niemożliwe.

Założenia matematyczne tej konstrukcji trzeba przenieść w inną dziedzinę matematycznąi tam jest to do udowodnienia.

Warianty: bliźniaczy ElGamal [CKS'08]

KeyGen: $g \longleftarrow \{\text{generatory } G\}$, $a1, a2 \longleftarrow Z_n$

wyjście $pk = (g, h_1=g^{a1}, h_2=g^{a2})$, $sk = (a1, a2)$

E(pk=(g,h_1,h_2), m) : $b \longleftarrow Z_n$

$k \longleftarrow H(g^b, h_1^b, h_2^b)$

$c \longleftarrow E_s(k, m)$

wyjście (g^b, c)

D(sk=(a1,a2), (u,c)) :

$k \longleftarrow H(u, u^{a1}, u^{a2})$

$m \longleftarrow D_s(k, c)$

wyjście m

13

Ostatnie lata badań nad zagadnieniami kryptograficznymi dotyczyły między innymi poszukiwań nad algorytmami ElGamal o zwiększonym bezpieczeństwie. Jednym z nich był tzw. bliźniaczy ELGamal. W tym algorytmie, zamiast pojedynczej wartości a, losuje się dwie różne wartości a1 i a2. Klucz publiczny składa się wtedy z trzech pól: g, h1 i h2 obliczonych jak na slajdzie. Klucz sekretny także składa się z pary wartości a1 i a2. Wygenerowanie klucza polega na uruchomieniu funkcji mieszającej (hash) z trzema argumentami: g, h1 i h2. Samo szyfrowanie symetryczne wygląda tak samo, szyfrogram składa się z g^b i zaszyfrowanej kluczem symetrycznym k wiadomości m.
Odszyfrowywanie wiadomości wymaga wyliczenia ponownie hash z odpowiednich 3 wartości oraz odszyfrowania szyfrogramu c z zastosowaniem klucza symetrycznego k.
Dla takiej konstrukcji można udowodnić bezpieczeństwo na atak z wybranym szyfrogramem wprost z założeń obliczeniowych DH, jeśli funkcja hash jest nieodróżnialna od ciągu losowego o jednorodnym rozkładzie.
Istnieją prace naukowe pokazujące, jak udowodnić bezpieczeństwo na atak z wybranym szyfrogramem na omawiany system, bez wymagania od funkcji Hash tak dobrych parametrów losowych.

Literatur dodatkowa

- The Decision Diffie-Hellman problem. D. Boneh, ANTS 3, 1998
- Universal hash proofs and a paradigm for chosen ciphertext secure public key encryption. R. Cramer and V. Shoup, Eurocrypt 2002
- Chosen-ciphertext security from Identity-Based Encryption. D. Boneh, R. Canetti, S. Halevi, and J. Katz, SICOMP 2007
- The Twin Diffie-Hellman problem and applications. D. Cash, E. Kiltz, V. Shoup, Eurocrypt 2008
- Efficient chosen-ciphertext security via extractable hash proofs. H. Wee, Crypto 2010