Authenticated Encryption

Inż. Kamil Zarychta
Opiekun: dr Ryszard Kossowski

Plan prezentacji

- Wprowadzenie
- Wymagania
- Opis wybranych algorytmów
- Porównanie mechanizmów
- Implementacja systemu
- Plany na przyszłość

Plan prezentacji

- Wprowadzenie
  - Podstawowe pojęcia
  - Idea Authenticated Encryption
- Wymagania
- Opis wybranych algorytmów
- Porównanie mechanizmów
- Implementacja systemu
- Plany na przyszłość

Podstawowe pojęcia

- *Poufność* – ochrona przed nieautoryzowanym rozpowszechnianiem danych
- *Uwierzytelnienie* – usługa umożliwiająca odbiorcy danych weryfikację tożsamości nadawcy
- *Integralność* – usługa umożliwiająca odbiorcy danych weryfikacje czy nie zostały one zmienione

Zabezpieczenie danych (1/2)

- W trakcie przesyłania przez sieć przed:
  - Podsłuchem
  - Nieautoryzowanym dostępem
- Składowanych w środowisku

Zabezpieczenie danych (2/2)

- Zapewnienie
  - Poufności – szyfrowanie danych
  - Integralności:
    - MAC
    - Podpis cyfrowy
  - W celu zapewnienia obydwu rozwiązaniem jest zastosowanie obu mechanizmów jednocześnie
  - Nie wszystkie kombinacje dają te same gwarancje bezpieczeństwa

Authenticated Encryption (1/2)

- Systemy szyfrujące jednocześnie chroniące wspomniane usługi
- Dostarcza zabezpieczenia przeciwko:
  - Atakom z wybranym tekstem jawnym (CPA, Chosen-planitext attack)
    - atakujący ma możliwość wybrania tekstu jawnego do zaszyfrowania i uzyskania szyfrogramu
    - umożliwia zdobycie informacji o zaszyfrowanej wiadomości oraz poznanie klucza szyfrującego
  - Atakom z wybranym szyfrogramem (CCA, chosen ciphertext attack)
    - atakujący ma możliwość wybrania kryptogramu i uzyskania odpowiadającego mu tekstu jawnego
    - umożliwia zdobycie danych i poznanie klucza

Authenticated Encryption (2/2)

- Dokument ISO/IEC CD 19772.2 – 2006-12-08
- Opis pięciu algorytmów zapewniających jednocześnie usługi poufności i uwierzytelnienia:
  - OCB 2.0 (Offset CodeBook ver. 2)
  - AES Key Wrap
  - CCM (Counter with CBC-MAC)
  - EAX
  - Encrypt-then-MAC
- Zdefiniowane metody mają na celu zapewnienie:
  - wspomnianych usług,
  - wydajności procesu przesyłania danych

Plan prezentacji

- Wprowadzenie
- Wymagania
- Opis wybranych algorytmów
- Porównanie mechanizmów
- Implementacja systemu
- Plany na przyszłość

Wymagania mechanizmów

- Odbiorca i nadawca informacji muszą:
  - Ustalić jeden z algorytmów opisanych w standardzie
  - Ustalić wybrany rodzaj szyfru symetrycznego
  - Współdzielić klucz do wybranego szyfru symetrycznego (Key Managment ISO/IEC11770)
  - Ustalić dodatkowe parametry dla konkretnych mechanizmów.

Plan prezentacji

- Wprowadzenie
- Wymagania
- Opis wybranych algorytmów:
  - OCB 2.0,
  - Encrypt-then-MAC,
    - Podstawowe symbole,
    - Specyficzne wymagania,
    - Procedury szyfrowania i deszyfrowania.
- Porównanie mechanizmów
- Implementacja systemu
- Plany na przyszłość

OCB 2.0 (1/5)

- Wykorzystane symbole:
  - $C_1, C_2, \ldots, C_m$ – sekwencja bloków bitów (każdy o długości n, z wyjątkiem C_m) uzyskanych jako wyjście procesu
  - $D_1, D_2, \ldots, D_m$ – sekwencja bloków bitów (każdy o długości n, z wyjątkiem D_m) uzyskanych w wyniku podziału danych wejściowych D
  - F, H – n-bitowe bloki wykorzystywane przy procesach szyfrowania i deszyfracji
  - m – liczba n-bitowych bloków w wiadomości do odszyfrowania, wiadomość zawiera (m-1)n+r bitów
  - M_2 – funkcja używana przy operacjach szyfrowania i operacji odwrotnej
  - P – n-bitowy blok wykorzystany w definicji funkcji M_2
  - r – liczba ($0<r\leq n$) bitów ostatniego bloku wiadomości do zaszyfrowania, uzyskana po podziale wiadomości na n-bitowe bloki

OCB 2.0 (2/5)

- Wykorzystane symbole:
  - S – zmienna wejściowa o długości n bitów
  - T – etykieta (t-bitów) przylegająca do zaszyfrowanej wiadomości w celu zapewnienia integralności danych
  - T` - przeliczona etykieta, generowana podczas procesu deszyfrowania
  - Z - n-bitowy blok używany podczas szyfrowania i deszyfrowania danych
  - # - funkcja przekształcająca liczbę a w blok o długości a-bitów.
    - jeżeli k jest liczbą naturalna ($0 \leq k \leq 2^a$) wtedy $\#_a(k)$ jest blokiem bitów o długości a, uznawanym za reprezentacje binarna liczby a z najbardziej znaczącymi bitami od lewej równymi k
  - $X|_S$ - lewo stronne obcięcie bloku bitów X
    - jeżeli długość bloku X jest większa lub równa S, wtedy $X|_S$ jest s-bitowym blokiem zawierającym s najbardziej znaczących bitów X (z lewej strony)

OCB 2.0 (3/4)

- Specyficzne wymagania:
  - Nadawca i odbiorca danych muszą uzgodnić:
    - długość ciągu bitów n (64 lub 128)
    - długość etykiety t w bitach, gdzie $0<t \leq n$
  - Definicja funkcji M_2:
    - Pobiera i zwraca n-bitowe bloki, gdzie
      - $n=64$, $P=0^{59}||11011$
      - lub $n=128$, $P=0^{120}||10000111$
    - Jeżeli X jest n-bitowym blokiem:
      - Jeżeli lewy-najbardziej znaczący bit X to zero, $M_2(X)=X\ll 1$
      - Jeżeli lewy-najbardziej znaczący bit X to zero, $M_2(X)=[X\ll 1]$ XOR P

OCB 2.0 (4/5)

- Procedura szyfrowania:
  - Nadawca, w celu ochrony ciągu danych D, powinien przeprowadzić następujące czynności:
    - Wybrać wartość początkową S odrębną dla każdej wiadomości i udostępnić ją odbiorcy
    - Podzielić D na bloki $D_1, D_2,\ldots, D_m$ wszystkie n-bitowe oprócz D_m (o długości r bitów). $D - (m-1)n + r$
    - $F = e_k(S)$ i $H=0^n$
    - Dla i=1, 2, …, m-1
      - $F = M_2(F)$
      - $H = H \text{ XOR } D_i$
      - $C_i = F \text{ XOR } e_k(D_i \text{ XOR } F)$
    - $F = M_2(F)$
    - $Z = e_k(\#_n(r) \text{ XOR } F)$
    - $C_m = D_m \text{ XOR } Z|_r$
    - $H = H \text{ XOR } [D_m || (Z|^{n-r})]$
    - $T = [e_K(H \text{ XOR } M_2(F) \text{ XOR } F)]|_t$
    - Wyjściem algorytmu jest ciąg $C= C_1 || C_2 || \ldots || C_m || T$, o długości $(m-1)n+r+t$ bitów
    - Przekazać odbiorcy ciąg C oraz wartość początkową S

OCB 2.0 (5/5)

- Procedura deszyfrowania
  - Odbiorca wiadomości w celu jej zweryfikowania powinien przeprowadzić następujące czynności:
    - Jeżeli długość ciągu C jest krótsza niż t zwróć na wyjście INVALID
    - Podzielić ciąg C na bloki $C_1 || C_2 || \ldots || C_m || T$ (zgodnie z tym, że C zawiera $(m-1)n+r+t$ bitów)
    - $F = e_k(S)$ i $H=0^n$
    - Dla i=1, 2, …, m-1
      - $F = M_2(F)$
      - $D_i = F \text{ XOR } d_k(C_i \text{ XOR } F)$
      - $H = H \text{ XOR } D_i$
    - $F = M_2(F)$
    - $Z = e_k(\#_n(r) \text{ XOR } F)$
    - $D_m = C_m \text{ XOR } Z|_r$
    - $H = H \text{ XOR } [D_m || (Z|^{n-r})]$
    - $T` = [e_K(H \text{ XOR } M_2(F) \text{ XOR } F)]|_t$
    - Jeżeli T=T` wtedy na wyjście podaj D, w przeciwnym wypadku INVALID

Encrypt-then-MAC (1/4)

- Wykorzystane symbole:
  - C` - ciąg bitów uzyskany w wyniku zaszyfrowania ciągu D
  - δ – funkcja deszyfrująca – $\delta_k(C)$; jako argumenty pobiera klucz blokowy K i zaszyfrowany ciąg bitów C
  - ε – funkcja szyfrująca – $\varepsilon_k(D)$;
  - f – funkcja MAC - $f_k(X)$; K klucz MAC, X – wejściowy ciąg
  - K_1 – sekretny klucz do szyfru blokowego
  - K_2 – sekretny klucz do funkcji MAC
  - S – zmienna startowa o długośći n-bitów
  - T – etykieta (n-bitów), dołączona do zaszyfrowanej wiadomości aby zapewnić integralność danych
  - T` - przeliczona wartość etykiety, wygenerowana w trakcie procesu deszyfrowania

Encrypt-then-MAC (2/4)

- Specyficzne wymagania:
  - Nadawca i odbiorca danych muszą uzgodnić:
    - Tryb szyfru blokowego (nie powinno się używać trybu ECB) – ISO/IEC 10116
    - Metodę wyliczania funkcji MAC - ISO/IEC 9797
    - Metodę opracowywania pary sekretnych kluczy (K_1, K_2) z klucza K
      - Długość klucza K musi być co najmniej tak długa jak łączna długość kluczy zarówno szyfru blokowego i funkcji MAC
      - Możliwe metody opracowania pary kluczy (K_1, K_2) z klucza K:
        - $K = K_1 \,||\, K_2$
        - Poprzez podzielenie ciągu bitów z funkcji skrótu h(K) o odpowiedniej długości

Encrypt-then-MAC (3/4)

- Procedura szyfrowania:
  - Nadawca, w celu ochrony ciągu danych D, powinien przeprowadzić następujące czynności:
    - wybrać zmienną początkową S odrębną dla każdej wiadomości
    - $C` = \varepsilon_{K1}(D)$
    - $T = f_{K2}(C`)$
    - Wyjściem algorytmu jest ciąg $C= C` || T$

Encrypt-then-MAC (4/4)

- Procedura deszyfrowania
  - Odbiorca wiadomości w celu jej zweryfikowania powinien przeprowadzić następujące czynności:
    - Jeżeli długość ciągu C jest mniejsza niż t, na wyjście INVALID
    - Podzielić ciąg C na dwie części C i T(końcowe t-bitów)
    - $T` = f_{K_2}(C`)$
    - Jeżeli T jest różne od T`, przerwij i wystaw na wyjście INVALID
    - $D = \delta_{K_1}(C`)$
    - Na wyjście podaj D.

Plan prezentacji

- Wprowadzenie
- Wymagania
- Opis wybranych algorytmów
- Porównanie mechanizmów
- Implementacja systemu
- Plany na przyszłość

Porównanie mechanizmów (1/3)

- Wszystkie mechanizmy wymagają:
  - wyboru szyfru blokowego (ISO/IEC 18033-3). Długość bloku n musi być równa co najmniej 64, sugerowaną wartością jest 128 bitów (wymagana w algorytmach 3 i 4).
  - od nadawcy i odbiorcy danych uzgodnienia klucza, znanego tylko im (ewentualnie trzeciej zaufanej stronie)

Porównanie mechanizmów (2/3)

- Mechanizm 1 (OCB 2.0):
  - dobór długości etykiety t (t≤n). Sugerowaną wartością jest t≥64
- Mechanizm 2 (AES Key Wrap):
  - Wymagana długość n bloku szyfru to 128 bitów.
- Mechanizm 3 (CCM):
  - Wymagana długość n bloku szyfru to 128 bitów. Sugerowaną wartością długości etykiety t jest t≥8 (w zapisie ósemkowym)
- Mechanizm 4 (EAX):
  - dobór długości etykiety t (t≤n). Sugerowaną wartością jest t≥64
- Mechanizm 5 (Encrypt-then-MAC):
  - Wymaga wyboru trybu operacji szyfrowania i funkcji MAC. Bezpieczeństwo mechanizmu zależy od bezpieczeństwa składowych.

Porównanie mechanizmów (3/3)

Mechanizm	1	2	3	4	5
Szacowana liczba operacji szyfrowania potrzebna do zaszyfrowania L-bitowej wiadomości	L/n	12L/n	2L/n	2L/n	2L/n
Czy wymagana licencja ?	Tak	Nie	Nie	Nie	Zależy od użytych metod szyfrowania i MAC
Zaprojektowana specjalnie do użycia z krótkimi wiadomościami ?	Nie	Tak	Nie	Nie	Nie
Długość wiadomości musi być znana przed szyfrowaniem ?	Nie	Nie	Tak	Nie	Nie
Wymagana zmienna początkowa ?	Tak	Nie	Tak	Tak	Nie
Standaryzowany wcześniej?	Nie	Tak	Tak	Nie	Nie

Plan prezentacji

- Wprowadzenie
- Wymagania
- Opis wybranych algorytmów
- Porównanie mechanizmów
- Implementacja systemu
  - Serwer
  - Klient
- Plany na przyszłość

Implementacja systemu (1/3)

- Aplikacja pracująca w modelu klient-serwer, która umożliwia bezpieczne przesyłanie danych pomiędzy użytkownikami. W tym celu aplikacja wykorzystuje mechanizmy kryptograficzne, takie jak:
  - Poufność.
  - Infrastruktura klucza publicznego PKI.
  - Cyfrowy podpis.
  - Uwierzytelnianie.

Implementacja systemu (2/3)

- Aplikacja serwera pełni następujące funkcje:
  - Jest jednocześnie ośrodkiem rejestracji RA i certyfikacji CA w jednopoziomowej infrastrukturze PKI stworzonej na potrzeby aplikacji.
  - Umożliwia wystawienie certyfikatu ośrodkowi certyfikacyjnemu CA, tak zwany *self-signed certificate*.
  - Umożliwia zakładanie kont użytkownikom, którzy będą korzystali z aplikacji klienta.
  - Umożliwia wystawianie cyfrowych certyfikatów użytkownikom, a w razie potrzeby wygenerowanie dla użytkownika kluczy publicznego i prywatnego służących do weryfikacji i podpisu.
  - Umożliwia zarządzanie stworzonymi kontami, takie jak usuwanie konta, odwoływanie certyfikatu użytkownika, wystawienie nowego certyfikatu użytkownikowi.
  - Prowadzi rejestr wystawionych certyfikatów.
  - Prowadzi listę certyfikatów odwołanych CRL.
  - Umożliwia zarządzanie rejestrem wystawionych certyfikatów, takie jak usuwanie certyfikatów, odwoływanie certyfikatów, import, eksport, czyszczenie rejestru.
  - Umożliwia zarządzanie listą certyfikatów odwołanych CRL, takie jak import, eksport, czyszczenie listy.

Implementacja systemu (3/3)

- Aplikacja klienta umożliwia:
  - Przesyłanie zaszyfrowanych danych pomiędzy użytkownikami.
  - Przetrzymywanie listy znajomych użytkowników.
  - Zarządzanie listą znajomych użytkowników, takie jak dodawanie użytkowników do listy, usuwanie użytkowników z listy, importowanie i eksportowanie listy.
  - Wyszukiwanie użytkowników którzy mają stworzone konto na serwerze.
  - Usunięcie konta z serwera.
  - Odwołanie certyfikatu użytkownika.
  - Generowanie par kluczy, publicznego i prywatnego, do szyfrowania lub podpisywania.
  - Zarządzanie wygenerowanymi kluczami, takie jak usuwanie, importowanie, eksportowanie.

Plan prezentacji

- Wprowadzenie
- Wymagania
- Opis wybranych algorytmów
- Porównanie mechanizmów Implementacja systemu
- Plany na przyszłość

Plany na przyszłość

- Dokończenie implementacji aplikacji,
- Przeprowadzenie testów, umożliwiających porównanie wydajności wszystkich mechanizmów Authenticated Encryption,
- Napisanie pracy magisterskiej.

- Dziękuję za uwagę
- Pytania ?