**Wrocław 01.10.2024**

**cyberbezpieczeństwo**

**Laboratorium nr 6**

**Jednokierunkowe funkcje skrótu  
Algorytmy wymiany klucza**

**Plan zajęć**

1. Podstawowe wiadomości na temat: funkcji skrótu, podpisu elektronicznego oraz wymiany kluczy kryptograficznych.
2. Zapoznanie się ze środowiskiem pracy - program CrypTool.
3. Jednokierunkowe funkcje skrótu.
4. Wymiana klucza kryptograficznego.
5. Raport z przeprowadzonych eksperymentów.

# Podstawowe wiadomości na temat: funkcji skrótu, podpisu elektronicznego oraz wymiany kluczy kryptograficznych.

Funkcja haszująca jest ważnym elementem techniki uwierzytelniania wiadomości. Otrzymuje ona dane wejściowe o różnej wielkości i wytwarza wartość skrótu o stałej wielkości. Funkcja haszująca używa funkcji kompresji w sposób powtarzalny, aby wygenerować n-bitową wartość wyjściową. W procedurze podpisu cyfrowego, wartość skrótu używa kluczy prywatnych i publicznych do przetwarzania. Uwierzytelnianie wiadomości zajmuje się ochroną wiadomości z zachowaniem integralności. Sprawdza tożsamość nadawcy komunikatu i niezaprzeczalność pochodzenia. Sprawdza, czy otrzymane wiadomości pochodzą od oryginalnego nadawcy. Zapewnia, że treść wiadomości nie jest modyfikowana lub zmieniana. Weryfikuje również kolejność i czas przesyłania wiadomości. Podpis cyfrowy jest techniką uwierzytelniania, który jest używany do sprawdzania strony nadawcy lub odbiorcy. Uwierzytelnianie podpisu cyfrowego odbywa się na dwóch poziomach. Nadawca wysyła podpisaną wiadomość do odbiorcy. Odbiorca porównuje obliczone kody hash z kodem hash, który otrzymał. Jeśli oba hashe się zgadzają, to wiadomość jest wiarygodna. Najpopularniejszą funkcją haszującą jest SHA (Secure Hash Algo-rithm). SHA są algorytmami kryptograficznymi, które zapewniają integralność danych i uwierzytelnianie. Aplikacje TLS, SSL, SSH i PGP używają SHA. W przeciwieństwie do szyfrów strumieniowych, które z krótkiego wyjścia tworzą długie wyjście, funkcje haszujące pobierają długie dane wejściowe i tworzą krótkie wyjście, zwane wartością haszującą lub skrótem.

A black and white text

Description automatically generated

Rysunek 1 Wejście i wyjście funkcji haszującej

Whitefield Diffie i Martin Hellman jako pierwsi opublikowali algorytm klucza publicznego, który rozwiązał problem uzgadniania lub wymiany kluczy. Algorytm ten jest ogólnie nazywany wymianą kluczy Diffie-Hellmana. Głównym celem algorytmu jest umożliwienie dwóm użytkownikom bezpiecznej wymiany klucza, który może być następnie użyty do szyfrowania wiadomości. Algorytm jest ograniczony tylko do wymiany kluczy i nie jest używany do szyfrowania lub deszyfrowania wiadomości.

Diffie-Hellman Key Exchange jest protokołem kryptograficznym wymieniającym klucze kryptograficzne poprzez sieć publiczną. Zapewnia on wysokie bezpieczeństwo wymiany kluczy pomiędzy podmiotami w niezabezpieczonym kanale. Wymienione klucze są później wykorzystywane do komunikacji symetrycznej przy użyciu popularnych szyfrów symetrycznych, takich jak AES, Blowfish, ChaCha20 itp.

DHKE jest również nazywany protokołem uzgadniania kluczy. Protokół wymiany kluczy Diffiego-Hellmana może być realizowany z wykorzystaniem logarytmów dyskretnych (klasyczny algorytm DHKE) lub kryptografii krzywej eliptycznej (algorytm ECDH). Poniżej przedstawiona jest wizualizacja wymiany kluczy pomiędzy użytkownikami.

A diagram of a public transport

Description automatically generated

Rysunek 2 Wymiana kluczy za pomocą DH

* Proszę zapoznać się z charakterystyką blokowych algorytmów szyfrowania. (Dokument w załączeniu oraz materiały do wykładów)
* Dodatkowe informacje mogą państwo znaleźć na stronach:
  + http://cacr.uwaterloo.ca/hac/about/chap9.pdf
  + http://cacr.uwaterloo.ca/hac/about/chap11.pdf
  + <http://pl.wikipedia.org/wiki/Funkcja_haszująca>
  + <http://pl.wikipedia.org/wiki/Atak_urodzinowy>
  + <http://pl.wikipedia.org/wiki/PKI>
  + <http://www.signet.pl/pomoc/pki.html>
  + <http://pl.wikipedia.org/wiki/Diffie-Hellman>

# Zapoznanie się ze **środowiskiem pracy - program CrypTool**.

* Na tych zajęciach laboratoryjnych będą wykorzystywane głównie narzędzia dostępne z zakładki z programu CrypTool:
  + - *Algorytmy/Hasz*
    - *Kryptoanaliza/Hasz*
    - *Algorytmy/Demosntracja protokołu Diffie-Hellman*

# Środowisko.

W laboratorium należy wykorzystać maszyny wirtualne:

* **Ubuntu Server,**
* **Kali,**

# Zadnia

## Jednokierunkowe funkcje skrótu.

### Proszę spróbować dokonać ataku na funkcję skrótu dla utworzonych przez siebie plików. **Zakładka:** *Kryptoanaliza/Hasz/*

### Proszę sprawdzić różnice w realizacji tego typu ataku dla różnej konfiguracji parametrów (inna funkcja skrótu, inna liczba wspólnego ciągu bitów, ...)

**Pytania/Wnioski:**

### Jak rośnie czas realizacji ataku wraz ze wzrostem wartości parametru opisującego długość ustalonego ciągu bitów?

### Czy wybór funkcji skrótu ma wpływ na czas realizacji zadania poszukiwania kolizji?

### Na czym polega przewaga modyfikowania dwóch dokumentów nad poszukiwaniem kolizji przy modyfikacji tylko jednego dokumentu?

### Czy w świetle uzyskanych wyników możemy ufać funkcjom skrótu?

### Jakiego typu problemy zostały zidentyfikowane dla popularnych funkcji skrótu (MD5,SHA)?

## Wymiana klucza kryptograficznego.

Zapoznaj się z demonstracją algorytmu Diffiego-Hellmana dostępną w Cryptool. W celach demonstracyjnych użyj wartości wygenerowanych w Cryptool. Następnie do praktycznego zastosowania protokołu DH posłużymy się biblioteką OpenSSL. Użyj dwóch maszyn wirtualnych jako klientów (Kali , Ubuntu Server) do wymiany kluczy.

Krótki opis kolejnych kroków:

* Wygeneruj wspólny klucz publiczny DH dla obu klientów,
* Na podstawie wspólnego klucza publicznego DH klienci generują klucze publiczne i prywatne,
* Klienci muszą wymienić swoje klucze publiczne,
* Na podstawie klucza publicznego i klucza prywatnego tworzony jest wspólny tajny klucz.

1. Wygeneruj wspólny klucz Diffiego-Hellmana dla obu klientów:

openssl genpkey -genparam -algorithm DH -out dh\_com\_pub.pem

1. wyświetl wspólny klucz
   * w postaci kodowanej

cat dh\_com\_pub.pem

* w postaci tekstowej:

openssl pkeyparam -in dh\_com\_pub.pem -text

1. Korzystając z uzgodnionego wspólnego klucza DH, każdy z klientów powinien uzgodnić własny klucz tajny:

openssl genpkey -paramfile dh\_com\_pub.pem -out dh\_client1\_key.pem

* wypisz klucze:

openssl pkey -in dh\_client1\_key.pem -text -noout

1. Wygeneruj klucz publiczny:

**openssl pkey -in** dh\_client1\_key.pem **-pubout -out** dh\_client1\_pub\_key.pem

1. Wymień klucze publiczne pomiędzy klientami, a następnie uruchom poniższą komendę w celu w ustalenia wspólnego klucza tajnego:

openssl pkeyutl -derive -inkey dh\_client1\_key.pem -peerkey dh\_client2\_pub\_key.pem -out secret\_key1.bin

1. Porównaj współdzielone klucza przy pomocy komendy cmp.
2. Wygeneruj zrzut klucza korzystając z komendy xxd.

**Pytania/Wnioski:**

### Który sposób wymiany kluczy gwarantuje większe bezpieczeństwo i dlaczego (RSA vs DH) ?

### Jakie jest ryzyko dla podmiotów korzystających z protokołu wymiany kluczy DH?

### Jakie są inne metody określenia wspólnego klucza kryptograficznego (oprócz DH, RSA)?

### Który element protokołu DH nie jest przesyłany między klientami? Jak to wpływa na kwestie bezpieczeństwa?

### Sprawdź, czy w ostatnim punkcie udało się ustalić ten sam klucz dla obu klientów. Uwzględnij w raporcie treść uzgodnionego tajnego klucza

# Raport z przeprowadzonych eksperymentów.

Raport powinien mieć postać dokumentu elektronicznego, w którym zostanie zapisany przebieg realizowanych w trakcie zajęć eksperymentów (np. fragment treści tekstu jawnego, parametry algorytmu szyfrowania, fragment kryptogramu) oraz wnioski, spostrzeżenia, odpowiedzi na pytania umieszczone w instrukcji laboratoryjnej, itp.