|  |
| --- |
| Politechnika Wrocławska, Informatyka Stosowana |
| JEDNOKIERUNKOWE FUNKCJE SKRÓTU ALGORYTMY WYMIANY KLUCZA |
| Cyberbezpieczeństwo, Laboratorium nr.6 - raport |

|  |
| --- |
| Autor: Aleksander Stepaniuk  Nr. Indeksu: 272644 |

**Zad 1. Jednokierunkowe funkcje skrótu**

Teksty których użyłem do analizy:

**Tekst 1: (plik nieszkodliwy)**

””

[Ta wiadomosc zostala wyslana w formie jakiegos pliku]

wstep bedzie wiec taki sam dla tych plikow

Jeszcze gdy chodzilem do podstawowki, to byl tam taki Pawel, i ja jechalem na rowerze, i go spotkalem, i potem jeszcze pojechalem do biedronki na lody, i po drodze do domu wtedy jeszcze, juz do domu pojechalem

Z wyrazami szacunku,

Jan Kowalski

””

**Tekst 2: (plik niebezpieczny)**

””

[Ta wiadomosc zostala wyslana w formie jakiegos pliku]

wstep bedzie wiec taki sam dla tych plikow

Cesarz czesal wlosy cesarzowej cesarzowa czesala wlosy cesarza Dzdzystym rankiem gzegzolki i piegze zamiast wziac sie za dzdzownice nazarly sie na czczo miazszu rzezuchy i rzedem rzygaly do rozzarzonej brytfanny Idzie Sasza sucha szosa suszy sobie swoje szorty Gdzie jest kufel pyta brat Moze kufel w kufer wpadl Bracie zawsze ci tlumacze kufel wpadl do kufra raczej Wyjmij z kufra kufel bracie lepiej postaw go na blacie

Z wyrazami szacunku,

Jan Kowalski

””

**Zadanie 1.1-1.2;**

Przeprowadzono eksperymenty dla różnych wartości funkcji skrótu, liczby wspólnych bitów

**Pytanie 1.3;**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Wzrost czasu realizacji ataku wraz ze wzrostem wartości parametru opisującego długość ciągu bitów | | | | | | | |
|  | 8 bitów | 16 bitów | 32 bitów | 48 bitów | 64 bitów | 128 bitów | 160 bitów |
| MD2 | 0,1s | 0,1s | 16s | 1h 22min | 14,8 dni | 10^94 lat | X |
| MD4 | 0,1s | 0,1s | 2s | 3min 59s | 5h 10min | 10^94 lat | X |
| MD5 | 0,1s | 0,1s | 1s | 4min 33s | 5h 11min | 10^94 lat | X |
| SHA | 0,1s | 0,1s | 2s | 1min 50s | 8h 5min | 10^94 lat | 10^94 lat |
| SHA-1 | 0,1s | 0,1s | 2s | 1min 3s | 4h 40min | 10^94 lat | 10^94 lat |
| RIPEMD-160 | 0,1s | 0,1s | 3s | 3min 4s | 13h 46min | 10^94 lat | 10^94 lat |

W miarę wzrostu liczby wspólnych bitów w plikach, czas realizacji ataku na funkcje hash rósł. Dla funkcji o silniejszych właściwościach kolizyjnych(bardziej odpornych na ataki kolizyjne), takich jak RIPEMD-160 lub SHA-256, czas ataku może drastycznie wzrastać, podczas gdy dla MD5 czy SHA-1, czas ataku rośnie wolniej. Z nieznanego mi powodu czas dla MD2 rósł najszybciej.

Im dłuższy jest ustalony wspólny ciąg bitów, tym trudniej (i dłużej) jest przeprowadzić skuteczny atak, ponieważ liczba możliwych kolizji maleje. Czas ataku rośnie wykładniczo razem ze wzrostem długości parametru długości bitów.

Obraz zawierający tekst, elektronika, zrzut ekranu, wyświetlacz

Opis wygenerowany automatycznie

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Pytanie 1.4;**

Wybór funkcji skrótu ma wpływ na czas realizacji poszukiwania kolizji. Najdłuższy czas osiągnęły MD2 oraz RIPEMD-160. SHA osiągnęło krótszy czas, a MD4, MD5 oraz SHA-1 osiągnęły najkrótszy czas.

**Pytanie 1.5;**

Przewaga modyfikowania dwóch dokumentów jest możliwa dzięki tzw. „birthday paradox” i opiera się na tym, że potrzebujemy jakiejkolwiek pary równych haszy ze zbioru, a nie jakiegoś konkretnego:

* W przypadku ataku z modyfikacją jednego dokumentu, aby znaleźć kolizję musimy znaleźć konkretnie dopasowany hasz wygenerowany przez funkcję skrótu. Prawdopodobieństwo znalezienia takiej kolizji to .
* W przypadku ataku z modyfikacją dwóch dokumentów wystarczy znaleźć dwa identyczne hasze, niezależnie jakie. Ważne jest jedynie to, aby dla obu dokumentów powstał ten sam skrót. Zwiększa to prawdopodobieństwo kolizji do .

Prawdopodobieństwo rośnie więc znacząco z każdym kolejnym dokumentem.

**Pytanie 1.6;**

Starsze funkcje skrótu takie jak MD5 czy SHA-1, są podatne na ataki kolizyjne, przez co nie zapewniają już wystarczającego bezpieczeństwa. Oznacza to, że nie można im w pełni ufać tym funkcjom dla zastosowań krytycznych, takie jak podpisy cyfrowe czy certyfikaty. Nowsze funkcje skrótu, takie jak SHA-256 czy RIPEMD-160 są bardziej odporne na kolizje, więc to one są zalecane do stosowania w miejscach wymagających wysokiego poziomu bezpieczeństwa.

**Pytanie 1.7;**

MD5 i SHA są podatne na ataki kolizyjne, z uwagi na małą długość generowanego skrótu (128/160bitów), co umożliwia znalezienie dwóch różnych fragmentów danych o tym samym hashu, co zagraża bezpieczeństwu podpisów cyfrowych. Ich złamanie stało się łatwiejsze przy obecnych mocach obliczeniowych. Dodatkowo odkryto przypadki praktycznego wykorzystania tych podatności, takie jak np. tworzenie fałszywych certyfikatów SSL z identycznymi hashami poprzez lukę w MD5, co poważnie zagrażało bezpieczeństwu w sieci. Ataki, umożliwiły także przechwytywanie i modyfikację danych wrażliwych w transakcjach finansowych oraz systemach autoryzacyjnych. Ze względu na te zagrożenia, wiele instytucji i standardów międzynarodowych zaleca stosowanie bardziej zaawansowanych funkcji haszujących, takich jak SHA-3 lub SHA-256.

**Zad 2. Wymiana klucza kryptograficznego**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznieObraz zawierający tekst, zrzut ekranu, diagram, oprogramowanie

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, diagram, oprogramowanie

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, diagram, Oprogramowanie graficzne

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, wyświetlacz, Czcionka

Opis wygenerowany automatycznie**a)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznieb)

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatyczniec1)

**Obraz zawierający tekst, zrzut ekranu, Czcionka

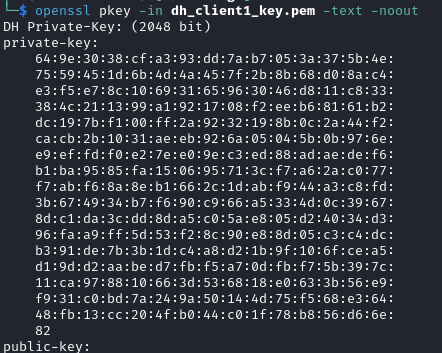
Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie**c2)

**Obraz zawierający tekst, zrzut ekranu, Czcionka, design

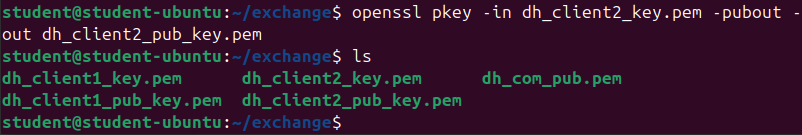
Opis wygenerowany automatycznie**

****

d)

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

****

e)

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

f)

cmp nie zwróciło żadnej informacji o różnicach, więc oba sekrety są dokładnie takie same.

**Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie**

g)

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**

**Pytanie 2.1;**

Wymiana kluczy za pomocą algorytmu Diffiego-Hellmana jest bezpieczniejsza w porównaniu do RSA, ponieważ pozwala na ustanowienie tajnego klucza bez jego bezpośredniego przesyłania, co chroni przed przechwyceniem go przez osoby trzecie. DH umożliwia też PFS (perfect forward secrecy), co oznacza, że nawet jeśli klucz prywatny jednej ze stron zostanie złamany, wcześniejsze sesje pozostaną bezpieczne. Jednak algorytm Diffiego-Hellmana nie pozwala uwierzytelniać autorów wiadomości, więc nie nadaje się on do podpisów elektronicznych – w tych zastosowaniach stosuje się RSA, które obsługuje zarówno szyfrowanie, jak i uwierzytelnianie za pomocą klucza prywatnego.

**Pytanie 2.2;**

Główne ryzyko, które wiąże się z wymianą klucza Diffiego-Hellmana polega na możliwości ataku pośredniego typu „*man in the middle*” (MitM), podczas którego atakujący przejmuje kontrolę nad komunikacją i wprowadza w błąd strony, które biorą w niej udział, tworząc oddzielne sesje szyfrowane z każdą ze stron, które pozostają tego nieświadome. DH nie posiada uwierzytelniania, więc bez dodatkowych mechanizmów uwierzytelniających strony nie mogą być pewne, czy faktycznie komunikują się akurat ze sobą. Oczywiście, jeśli stosowane są niewystarczająco silne parametry (np. małe liczby pierwsze) klucz wymieniony przez DH może być podatny na złamanie.

**Pytanie 2.3;**

Wśród innych metod ustalania wspólnego klucza kryptograficznego można nadmienić: „*Elliptic Curve Diffie-Hellman”* (ECDH), który jest wariantem Diffie-Hellmana opartym na krzywych eliptycznych, zapewniającym większe bezpieczeństwo przy krótszych kluczach oraz „*Kluczowanie symetryczne z centralnym serwerem”* (np. Kerberos), gdzie centralny serwer uwierzytelniający generuje i rozsyła klucze sesji do obu stron komunikacji. Kolejną metodą jest „*Password-Authenticated Key Agreement”* (PAKE), która pozwala stronom uzgodnić wspólny klucz kryptograficzny przy użyciu częściowej znajomości wspólnego hasła, zapewniając bezpieczeństwo nawet w przypadku podsłuchu i bez potrzeby przesyłania hasła w formie jawnej. Wszystkie te metody oferują różne poziomy bezpieczeństwa, zależnie od wymaganego zastosowania.

**Pytanie 2.4;**

Element protokołu DH nie przesyłany pomiędzy klientami to klucze tajne każdego uczestnika komunikacji. Używane są one jedynie do operacji generowania klucza publicznego. Wspólny klucz wynikowy jest efektem połączenia klucza prywatnego jednej strony z kluczem publicznym strony przeciwnej i będzie dokładnie taki sam dla obu stron. Ten fakt sprawia, że klucz tajny pozostanie tajny. Dodatkowo do generowania kluczy wykorzystuje się bardzo duże liczby pierwsze, przez co proces faktoryzacji tych liczb (a więc odwrócenia klucza) jest obecnie bardzo czasochłonny i nie do wykonania w sensownym czasie. W związku z tym nie jest możliwe podsłuchanie komunikacji i odtworzenie na tej podstawie klucza, ponieważ nie posiadamy dość informacji.

**Pytanie 2.5;**

**Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie**Tak udało się ustalić ten sam klucz dla obu klientów:

**Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie**