Politechnika Wrocławska

SPRAWOZDANIE Z ZAJĘĆ LABORATORYJNYCH
CYBERBEZPIECZEŃSTWO

Lab 06
Funkcje skrótu

Tomasz Mroczko, 266604

November 19, 2023

1 Zadania

Zadanie 1.2

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

Licza bitów / Funkcja skrótu	MD2	MD5	SHA	SHA-1
8	0.01s	0.00s	0.00s	0.00
16	0.04s	0.00s	0.00s	0.01
24	0.25s	0.00s	0.02	0.00
32	6.82s	0.32s	0.15s	0.14s
40	132s	1.32s	2.25s	4.34s
48	> 40min	50.21s	68.78s	15.43s
64	7.5dni	200min	274min	185min

Table 1: Czas przeprowadzenia ataku

Zadanie 1.3

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

Czas realizacji ataku rośnie wraz ze wzrostem długości wspólnego ciągu bitów. Jest to wzrost bardzo znaczący, być może wykładniczy. Przy wartościach poniżej 32 bitów ataki testowanych funkcji skrótu są bardzo szybkie. Dla wartości 64 bitów i więcej, czas staje się znaczny a następnie rośnie do ogromnych wartości.

Zadanie 1.4

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

Wybór funkcji skrótu ma wpływ na czas realizacji poszukiwania kolizji. Zdecydowanie najwolniejsza było dla funkcji skrótu MD2. Jeśli chcemy zneleźć dużą liczbę powtarzających się bitów, to przewidywany czas poszukiwania kolizji będzie bardzo duży dla wszystkich funkcji.

Zadanie 1.5

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

W przypadku modyfikowania tylko jednego dokumentu, wartość funkcji skrótu do której dążymy jest ustalona poprzez niemodyfikowany dokument. Jest to wartość stała, której osiągnięcię poprzez modyfikację tylko jednego dokumentu jest bardzo trudne (ponieważ mamy tylko jedną wartosc którą chcemy osiągnąć).

W przypadku modyfikowania obu dokumentów, atak ma więcej swobody. Modyfikacja obu dokumentów pozwala na łatwiejsze znalezienie wspólnej wartości. Wystarczy znaleźc jedną, dowolną wspólną wartość (W przeciwieństwie do narzuconej z góry wartości do której trzeba dążyć modyfikując jeden plik) Jednak dalej jest to bardzo wymagające zadanie.

Zadanie 1.6

Czy w świetle uzyskanych wyników możemy ufać funkcjom skrótu?

W świetle uzyskanych wyników możemy ufać funkcjom skrótu. Czas poszukiwania kolizji rośnie bardzo znacząco wraz z liczbą pokrywających się bitów. Obecnie zaleca się jednak unikanie funkcji skrótu takich jak MD2, MD5 oraz SHA-1, ponieważ wykazano ich słabe strony. Udokumentowano techniki dużo wydajniejsze niż brute-force, pozwalające na znalezienie kolizji.

Aktualnie jedną z najczęściej stosowanych funkcji skrótu jest SHA-256. Algorytmy z rodziny SHA-2 oraz SHA-3 są zalecane i zatwierdzone przez NIST.

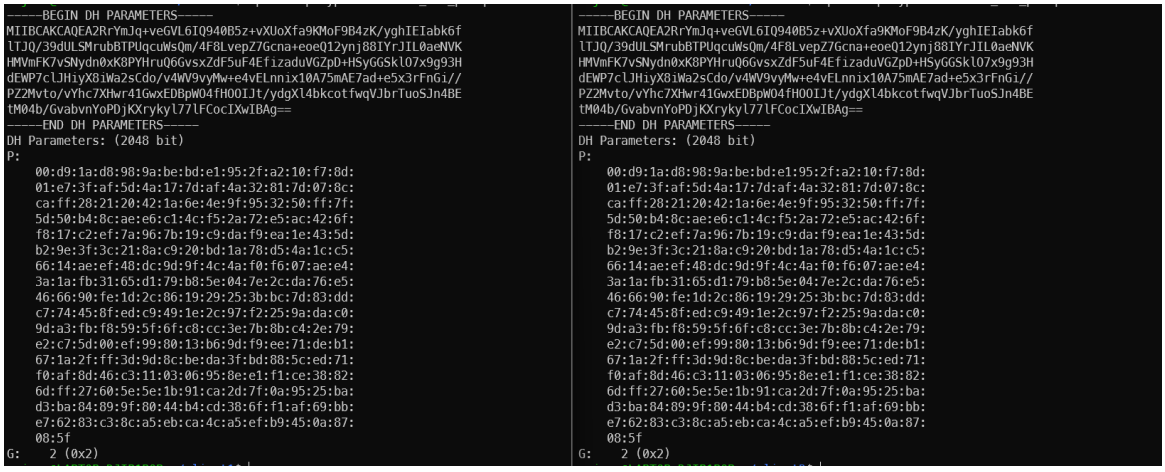
Zadanie 1.6

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

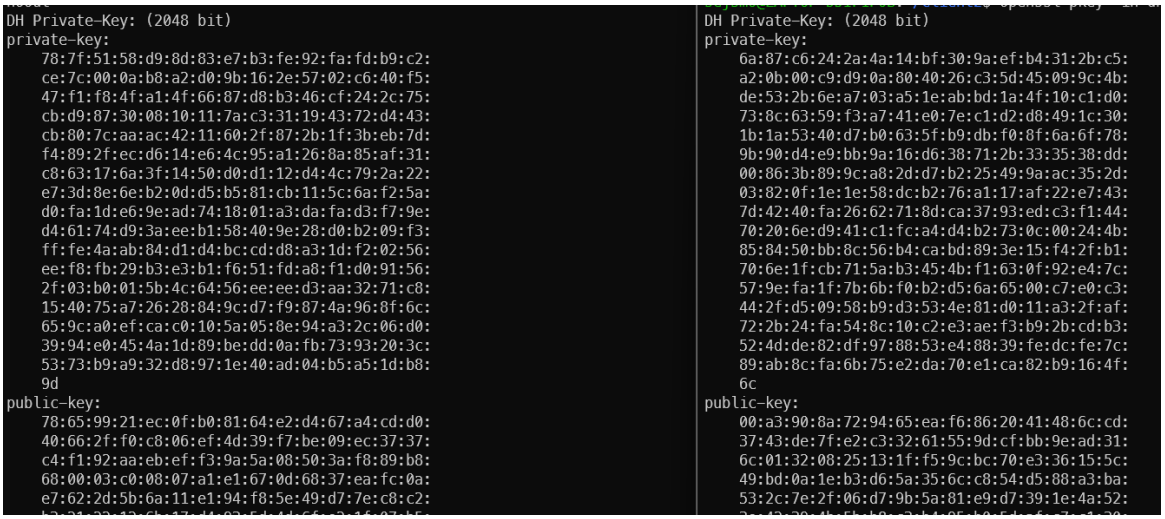
Problemem który został zidentyfikowany jest podatność na ataki kolizyjne. Stosunkowo niskie długości skrótów (128 dla MD5 oraz 160 dla SHA-1), sprawiają że są są one bardziej podatne na kolizje. Funkcje te są stosunkowo szybkie, co zwiększa ich podatność na ataki brute-force. Wykazano sposoby skuteczniejsze niż metoda brute-force pozwalające na łamanie obu funkcji. MD5 już w 2004 zostało podważone solidnym dowodem. W 2017 roku udowodniono że SHA nie jest już bezpieczny za pomocą ataku SHAttered.

Generowanie klucza metodą Diffiego-Hellmana

- Wspólny klucz obu klientów.



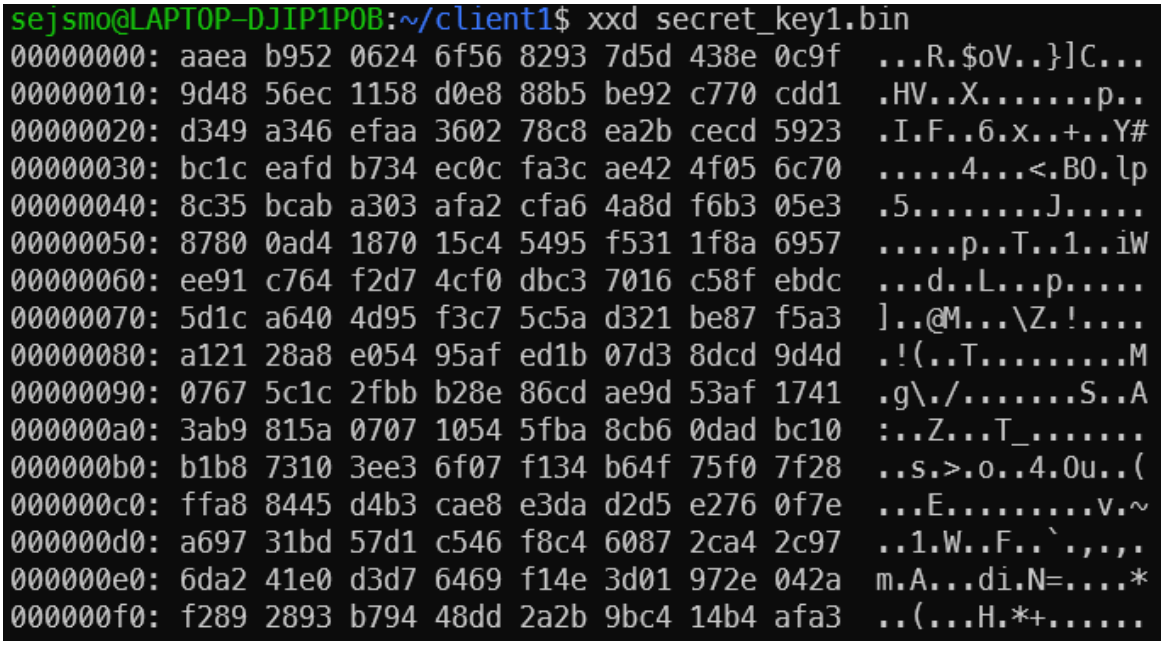
- Klucze prywatne klientów



- Po wymianie kluczy publicznych

- Porównanie klucza wspólnego

- Zrzut klucza komendą xxd



Zadanie 2.1

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

Trudno stwierdzić który sposób wymiany gwarantuje większe bezpieczeństwo. Protokół RSA opiera siłę na faktoryzacji liczb pierwszych, a Diffie-Hellman na obliczeniach logarytmów i arytmetyce modularnej. RSA uważa się generalnie za bezpieczniejszy niż DH ale jest to spowodowane głównie użyciem większych długości klucza. Przy tej samej długości klucza (zazwyczaj 1024bit), DH uznawany jest za nieco bezpieczniejszy. Protokół DH jest podatny na ataki Man in the Middle. W praktyce oba te algorytmy są szeroko stosowane i uznawane za stosunkowo bezpieczne o ile używane są zgodnie z zaleceniami. (https://www.geeksforgeeks.org/difference-between-diffie-hellman-and-rsa/)

Zadanie 2.2

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

Jednym z głównych niebezpieczeństw podczas korzystanie z tego protokołu jest możliwość ataku man in the middle. Jeśli atakujący jest w stanie przechwycić i manipulować informacjami przesyłanymi kanałem potencjalnie niebezpiecznym, to może ustalić własne klucze. Następnie dzięki temu ma możliwość odszyfrowania komunikacji. Aby temu zapobiec, często używa się dodatkowych mechanizmów, takich jak podpisy cyfrowe.

Zadanie 2.3

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

Poza tymi algorytmami istnieją jeszcze warianty DH oparte na (kryptografi krzywych eliptycznych). Jednym z wariantów jest ECDH (Elliptic Curve Diffie Hellman). Jest on bardziej wydajny (operacje na krzywych eliptycznych są zazwyczaj szybsze i wymagają mniej mocy obliczeniowe), szybszy oraz pozwala na użycie krótszych kluczy. Istnieje także wariant ECDHE(Elliptic Curve Diffie Hellman Ephermal). Protokół MQV (Menezes–Qu–Vanstone) to kolejny bazowany na DH sposów określenia klucza. (https://www.certicom.com/content/certicom/en/code-and-cipher/mqv.html)

Zadanie 2.4

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

Elementem który nie jest przesyłany jest klucz prywatny każdej ze stron. Klucz ten, w połączeniu z kluczem publicznym drugiej ze stron pozwala na ustalenie wspólnego klucza. Dzięki wykorzystaniu dużych liczb pierwszych i potęgowania w procesie tworzenia kluczy publicznych, nawet jeśli atakujący przechwyci klucze publiczne, to będzie miał problem z odwróceniem operacji w celu odtworzenia klucza prywatnego.

Zadanie 2.5

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

Tak, udało się ustalić ten sam klucz.

- Porównanie klucza wspólnego

```
:~/client2$ cmp secret_key1.bin ../client1/secret_key1.bin
:~/client2$
```

- Zrzut klucza komendą xxd

```
sejsmo@LAPTOP-DJIP1P0B:~/client1$ xxd secret_key1.bin
00000000: aaea b952 0624 6f56 8293 7d5d 438e 0c9f  ...R.$oV..}]C...
00000010: 9d48 56ec 1158 d0e8 88b5 be92 c770 cdd1  .HV..X.......p..
00000020: d349 a346 efaa 3602 78c8 ea2b cecd 5923  .I.F..6.x..+..Y#
00000030: bc1c eafd b734 ec0c fa3c ae42 4f05 6c70  .....4...<.BO.lp
00000040: 8c35 bcab a303 afa2 cfa6 4a8d f6b3 05e3  .5........J.....
00000050: 8780 0ad4 1870 15c4 5495 f531 1f8a 6957  .....p..T..1..iW
00000060: ee91 c764 f2d7 4cf0 dbc3 7016 c58f ebdc  ...d..L...p.....
00000070: 5d1c a640 4d95 f3c7 5c5a d321 be87 f5a3  ]..@M...\Z.!....
00000080: a121 28a8 e054 95af ed1b 07d3 8dcd 9d4d  .!(..T.........M
00000090: 0767 5c1c 2fbb b28e 86cd ae9d 53af 1741  .g\./.......S..A
000000a0: 3ab9 815a 0707 1054 5fba 8cb6 0dad bc10  :..Z...T_.......
000000b0: b1b8 7310 3ee3 6f07 f134 b64f 75f0 7f28  ..s.>.o..4.Ou..(
000000c0: ffa8 8445 d4b3 cae8 e3da d2d5 e276 0f7e  ...E.........v.~
000000d0: a697 31bd 57d1 c546 f8c4 6087 2ca4 2c97  ..1.W..F..`.,.,.
000000e0: 6da2 41e0 d3d7 6469 f14e 3d01 972e 042a  m.A...di.N=....*
000000f0: f289 2893 b794 48dd 2a2b 9bc4 14b4 afa3  ..(...H.*+......
```