Politechnika Wrocławska

SPRAWOZDANIE Z ZAJĘĆ LABORATORYJNYCH
CYBERBEZPIECZEŃSTWO

Lab 05
Analia współczesnych algorytmów

Tomasz Mroczko, 266604

November 12, 2023

1 Zadania

Zadanie 1.1

Proszę ocenić czas potrzebny do odnalezienia pełnego klucza o długości 64, 128, 192, 256 bitów.

Do szyfrowania 64 bitów użyto algorytmu DES (CBC). Pozostałe długości wykorzystywały AES(CBC).

Długość klucz	Przewidywany czas
64 bit	$5 * 10^4$ lat
128 bit	$3.3 * 10^{25}$ lat
192 bit	$6.3 * 10^{44}$ lat
256 bit	$1.2 * 10^{64}$ lat

Table 1: Długość odnalezienia pełnego klucza dla różnych długości

Jak widać algorytmy symetryczne, nawet DES o długości klucza 64 bit, są bardzo odporne na ataki metodą brute force. Długość ataku brute force znacząco rośnie wraz z długościąużywanego klucza.

Zadanie 1.2

Proszę porównać czasy poszukiwania kluczy o tej samej długości (np. 128 bitów) dla różnych algorytmów. Przestestowano tylko długość klucza 128 bit, ponieważ jest najczęściej spotykana.

Algorytm	Przewidywany czas
IDEA	$1.9 * 10^{26}$lat
RC2	$1.8 * 10^{26}$ lat
3DES(CBC)	$5.2 * 10^{21}$ lat
AES(CBC)	$3.3 * 10^{25}$ lat

Table 2: Długość odnalezienia pełnego klucza dla różnych długości

Zadanie 1.3

Proszę określić czas poszukiwania klucza przy 4,8,12,16,20,24, ... nieznanych bitach. (Jedna gwiazdka w kluczu = 4 bity)

Przetestowano czas poszukiwania dla algorytmu AES(CBC) o długości klucza 128. Przy nieznanych wszystkich bitach, przewidywana długość ataku brute-force to około $3 * 10^{25}$ lat.

Liczba nieznanych bitów	Przewidywany czas
4	Natychmiast - tylko 16 opcji
8	Natcyhmiast - tylko 256 opcji
16	Naychmiast
20	3 sekundy
24	45 sekund
28	12 minut
32	3 godziny
36	2.3 dnia (około 55 godzin)
64	$1.6 * 10^6$ lat
96	$7.3 * 10^{15}$ lat
128(pełny klucz)	$3.3 * 10^{25}$ lat

Table 3: Długość odnalezienia pełnego klucza dla różnych ilości brakujących bitów

Długość odnalezienia klucza znacząco rosła, wraz z liczbą nieznanych bitów. Dla poszukiwanych 16 lub mniej bitów, było to dla percepcji człowieka natychmiastowe. Przy 20 bitach trwało około 3 sekund. Następnie, z każdą kolejną inkrementacją o 4, czas poszukiwania rósł około 16-krotnie.

Zadanie 1.4

Proszę sprawdzić czy pozycja nieznanych bitów w kluczy wpływa na czas poszukiwania klucza.

Niezależnie od pozycji usuniętych bitów, czas poszukiwania klucza pozostawał niezmienny.

Zadanie 1.5

Proszę ocenić jakość działania algorytmu łamiącego:
• Czy każdorazowo otrzymujemy poprawny klucz?
• Czy liczba szukanych bitów wpływa na jakość odtwarzanego klucza?
• Czy pozycja nieznanych bitów wpływa na jakość odtwarzanego klucza?

- W eksperymentach które przeprowadziłem, klucz był zawsze poprawny. Warto zaznaczyć że jeśli deszyfrowanie trwało więcej niż minutę, to nie czekałem na ukończenie ataku brute force. Skoro przeszukujemy całą przestrzeń kluczy (brute-force), to *musi* zostać znaleziony prawidłowy klucz.
- Liczba szukanych bitów nie powinna wpływać na jakość klucza. Przeszukiwana jest cała dostępna przestrzeń kluczy, niezależnie od liczby szukanych bitów. Problemem jest wykładniczo rosnący czas przeszukiwania.
- Pozycja nieznanych bitów, tak samo jak ich liczba, nie wpływa na jakość znalezionego klucza.

Zadanie 1.6

Czy współczesne algorytmy blokowe możemy uznać za bezpieczne (w świetle przeprowadzonych eksperymentów)?

W świetle przeprowadzonych eksperymentów algorytmy blokowe można uznać za bezpieczne. Duża przestrzeń kluczy sprawia że ataki brute force są bardzo wymagające jeśli chodzi o czas i zasoby. Przeprowadzone eksperymenty nie uwzględniają ataków innego typu, jednak odporność na ataki brute force jest bardzo wysoka. Już dla klucza długości 64 bitów, przewidywany czas naiwnych ataków brute force był rzędu dziesiątek tysięcy lat, a wraz ze wzrostem jego długości - zwiększał się wykładniczno.

Zadanie 1.7

Jaka długość klucza oferuje nam wystarczający poziom bezpieczeństwa? (Dlaczego?)

Trudno ocenić jaki poziom jest 'wystarczający', ponieważ zależy to od wymaganego bezpieczeństwa oraz możliwości obliczeniowych. W przypadku mojego komputera oraz naiwnego podejścia brute force oferowanego przez narzędzie CrypTool, już klucz długości 64 bitów był zdecydowanie zbyt wymagający na rozwiązanie w sensownym czasie. Atak ten jednak nie wykorzystywał pełnej mocy procesora (wykorzystanie procesora podczas ataku wynosiło zaledwie 4%). Klucz 128 bitów wydaje się być wystarczajo bezpieczny niezależnie od mocy obliczeniowej - przewidywany czas ataku był rzędu 10^{25} lat.

Zadanie 1.8

Czy wielkość kryptogramu ma wpływ na możliwość jego złamania?

W kontekście przeprowadzonych eksperymentów, długość tekstu jawnego - a co za tym idzie kryptogramu, wykazała pewne zmiejszenie przewidywanego czasu ataku brute force. Zmniejszenie długości tekstu zróżnicowanego z 2000 do około 100 znaków, spowodowało zmniejszenie przewidywanego czasu o około 20%. Skrócenie ze 100 znaków do około 10, spowodowało zmniejszenie o kolejne 20-25%. Zwiększenie długości tekstu ponad 2000 znaków nie spowodowało zauważalnej zmiany w przewidywanym czasie.

Trudno ocenić skąd wynika ta zmiana - w każdym przypadku należy przetestować wszystkie klucze, niezależnie od długości tekstu/kryptogramu.

Jeśli chodzi o kryptoanalizę i inne niż brute force podejścia, długość kryptogramu na pewno ma jakieś znaczenie. Ataki na analizie statystycznej oraz częstotliwości mogą być trudniejsze lub łatwiejsze, w zależności od wzorócw przejawianych w tekście. Dłuższy tekst jawny bardziej przejawia wzorce danego języka takie jak najczęstsze litery, digramy oraz entropia tekstu.

Zadanie 1.9

Czy format i wcześniejsze przetwarzanie dokumentu (kompresja, zmiana formatu dokumentu,...) wpływa na możliwość jego kryptoanalizy?

Tak, formatowanie i przetwarzanie dokumentu przed zaszyfrowaniem może utrudnić kryptoanalizę. Ataki oparte na analizie statystycznej Kompresja zmniejsza rozmiar pliku oraz nadmiarowość danych, co utrudnia wyszukiwanie wzorców oraz analizę tekstu. Zmiana formatu pliku może wprowadzić pewne trudności oraz zwiększyć entropię. Przetwarzanie może w pewnym stopniu ukryć strukturę danych, co utrudnia zrozumienie informacji oraz prawidłowość klucza.

Zadanie 1.10

Ile możliwych haseł możemy sprawdzić przez rok nieustannej pracy na jednym komputerze, który sprawdza milion haseł w ciągu sekundy (2^{20})? Co ten wynik mówi o bezpieczeństwie współczesnych algorytmów?

W roku jest 31 556 926 sekund, liczba ta zbliżone jest do 2^{25}, czyli 33 554 432. W takim razie, jeśli w ciągu sekundy jesteśmy w stanie sprawdzić około 2^{20} haseł, to w ciągu roku wartość ta wynosi mniej więcej

$$2^{25} * 2^{20} = 2^{45}$$

Oznacza to że klucz długości 64 bitów wymagałby około 2^{19} lat (około pół miliona lat). Klucz o długości 128 bitów wymagałby przy tej mocy obliczeniowej 2^{83} lat. W świetle tych wyników, bezpieczeństwo tych algorytmów na ataki brute force jest bardzo duże. Trzeba jednak zaznaczyć że technologia rozwija się bardzo szybko a moc obliecznniowa rośnie. Jednak na obecną chwilę, algorytmy te są względnie bezpieczne.

Zadanie 2.2

Proszę sprawdzić, jak rośnie czas poszukiwania liczb pierwszych wraz ze wzrostem wartości przeglądanego przedziału.

Przeszukiwany zakres	Czas generowania
Poniżej 2^{12}	Natychmiast
$0 - 2^{12}$	Natychmiast
$0 - 2^{13}$	Około 1s
$0 - 2^{14}$	Około 2s
$0 - 2^{15}$	Około 4s
$0 - 2^{16}$	Około 7s
$0 - 2^{17}$	Około 13s
$0 - 2^{18}$	Około 25s
$0 - 2^{19}$	Około 56s
$0 - 2^{20}$	Około 90s
$0 - 2^{21}$	Około 163s

Table 4: Czas szukania liczb pierwszych dla danego zakresu

Czas poszukiwania liczb pierwszych rośnie znacząco wraz ze wzrostem przedziału.

Zadanie 2.3

Proszę sprawdzić, jak zależy skuteczność oraz czas potrzeby na realizację ataku faktoryzacji modułu N algorytmu RSA dla różnych wartości parametrów: Długość N, Długość p, Długość znanego ciągu bitów P (lab5_numbers.txt)

Znaczenie parametrów
Aby wygenerować klucz RSA należy wybrać dwie liczby pierwsze, które oznaczane są jako p i q. Liczby powinny zostać wybrane losowo, a ich zapis binarny powinien mieć podobną długość. Przykładowo, dla popularnie stosowanego klucza 2048, zapis binarny p oraz q powinien mieć około 1024 bitów. Iloczyn tych dwóch liczb to liczba N, moduł dla kluczy.

Jeśli chodzi o poszczególne parametry faktoryzacji to:

- N: jest to moduł kluczy, iloczyn liczb pierwszych p oraz q
- P: to znana częśc jednej z liczb pierwszych p

- p: jest parametrem oznaczającym długość zapisu binarnego liczby b
- LSB/MSB: Sposób reprezentacji liczb binarnych. LSB to najmniej znaczący bit a MSB najbardziej.

Poszczególne przykłady z pliu

Parametr	Długość bitowa
N	200
P	80
p	80
Metoda	LSB
Czas trwania	Natychmiast

Parametr	Długość bitowa
N	300
P	120
p	120
Metoda	MSB
Czas trwania	1s

Parametr	Długość bitowa
N	300
P	141
p	200
Metoda	LSB
Czas trwania	NIEPOWODZENIE

Zadanie 2.4

Proszę zapoznać się z metodą ataku dostępną w zakładce: Kryptoanaliza/Algorytmy asymetryczne/ Ataki oparte na kracie /Ataki na wiadomości stereotypowe i spróbować odnaleźć brakujący fragment tekstu z załączonego pliku lab_5_number.txt

- W pierwszym przykładzie, dla długości klucza 1024 bitów, udało się po 35 sekundach uzyskać brakujący tekstu. Całość tekstu brzmi

  > As a way of clearing the way for the implementation of elliptic curves to protect US and allied goverment information, the Nat

- W drugim przykładzie atak zakończył się niepowodzeniem.

Zadanie 2.6

.Jaka jest minimalna długość modułu (liczba N) algorytmu RSA, która gwarantuje, że jej rozkład (znalezienie jej czynników pierwszych) będzie dostatecznie trudne

Przez długi czas za wystarczającą uznawana była długość 1024 bitów. Jednak wraz z postępem mocy obliczeniowej i technologii, zaczęto łamać klucze o długości coraz bardziej zbliżającej się do tej wartości. Aktualnie jako minimum rekomendowana jest długość modułu 2048 bitów.

Zadanie 2.7

Czy dla przyjętej we wcześniejszym punkcie bezpiecznej długości modułu, można przeprowadzić skuteczny atak faktoryzacji w oparciu o częściową znajomość wartości jednego parametru? (zadanie 3)

Jeśli znamy wystarczająco dużo bitów jednego z parametrów, atak jest możliwy a nawet stosunkowo szybki. Wymaga to jednak znajomości znacznej części (w testach potrzebowałem około 650 bitów, przy długości N 2048) jednego ze składników.

Zadanie 2.8

W jakich przypadkach szyfrowanie algorytmem RSA może być zagrożone przez atak realizowany w punkcie 4?

Atak oparty o wiadomości sterotypowe jest największym zagrożeniem, kiedy tekst jawny jest typowym tekstem przejawiającym wiele wzorców. Atak może ułatwić znajomość długości wiadomości oraz części tekstu jawnego. Jeśli ten sam klucz jest użyty wielokrotnie, to atakujący może starać się szukać wzorców i zdobywać coraz więcej informacji na podstawie każdej przechwyconej wiadomości.