Politechnika Wrocławska, Informatyka Stosowana

Analiza współczesnych algorytmów

Cyberbezpieczeństwo, Laboratorium nr.5 - raport

Autor: Aleksander Stepaniuk
Nr. Indeksu: 272644

Zad 1. Własności algorytmów asymetrycznych

Tekst którego użyłem do analizy:

Tekst 1:

6489 pierwszych znaków pana Tadeusza:

(„litwo ojczyzno moja ty jestes jak zdrowie ile cie trzeba cenic ten tylko sie dowie kto cie stracil dzis pieknosc twa w calej ozdobie widze i opisuje bo tesknie po tobie panno swieta co jasnej bronisz czestochowy i w ostrej swiecisz bramie ty co grod zamkowy nowogrodzki ochraniasz z jego wiernym ludem jak mnie dziecko do zdrowia powrocilas cudem gdy od placzacej matki pod twoje opieke ofiarowany martwa podnioslem powieke i zaraz moglem pieszo do twych swiatyn progu isc za wrocone zycie podziekowac bogu tak nas powrocisz cudem na ojczyzny lono tymczasem przenos moje dusze uteskniona do tych pagorkow lesnych do tych lak zielonych szeroko nad blekitnym niemnem rozciagnionych do tych pol malowanych zbozem rozmaitem wyzlacanych pszenica posrebrzanych zytem gdzie bursztynowy swierzop gryka jak snieg biala gdzie panienskim rumiencem dziecielina pala a wszystko przepasane jakby wstega miedza zielona na niej z rzadka ciche grusze siedza srod takich pol przed laty nad brzegiem ruczaju na pagorku niewielkim we brzozowym gaju stal dwor szlachecki z drzewa lecz podmurowany swiecily sie z daleka pobielane sciany tem bielsze ze odbite od ciemnej zieleni topoli co go bronia od wiatrow jesieni dom mieszkalny niewielki lecz zewszad chedogi i stodole mial wielka i przy niej trzy stogi uzatku co pod strzecha zmiescic sie nie moze widac ze okolica obfita we zboze i widac z liczby kopic co wzdluz i wszerz smugow swieca gesto jak gwiazdy widac z liczby plugow orzacych wczesnie lany ogromne ugoru czarnoziemne zapewne nalezne do dworu uprawne dobrze na ksztalt ogrodowych grzadek ze w tym domu dostatek mieszka i porzadek brama na wciaz otwarta przechodniom oglasza ze goscinna i wszystkich w goscine zaprasza wlasnie dwokonna bryka wjechal mlody panek i obieglszy dziedziniec zawrocil przed ganek wysiadl z powozu konie porzucone same szczypiac trawe ciagnely powoli pod brame we dworze pusto bo drzwi od ganku zamknieto zaszczepkami i kolkiem zaszczepki przetknieto podrozny do folwarku nie biegl slug zapytac odemknal wbiegl do domu pragnal go powitac dawno domu nie widzial bo w dalekim miescie konczyl nauki konca doczekal nareszcie wbiega i okiem chciwie sciany starodawne oglada czule jako swe…”)

Zadanie 1.1;

Przy ocenie czasu potrzebnego na złamanie kluczy o różnych długościach dla algorytmów symetrycznych, zakładamy, że atakujący stosuje brute force, czyli sprawdza wszystkie możliwe kombinacje dla klucza. Zakładamy że atakujący może przetestować **10^12** kluczy na sekundę, przy 2^n możliwych kluczy dla klucza o długości n. Czas będzie więc wynosił 2^n podzielone przez liczbę sprawdzanych kluczy na sekundę (10^12)

Klucz 64-bitowy:

- Liczba możliwych kluczy: 2^64 ≈ 10^19
- Czas złamania: 10^19/10^12 ≈ 10^7 sekund ≈ **4 miesiące**

Klucz 128-bitowy:

- Liczba możliwych kluczy: 2^128 ≈ 10^38
- Czas złamania: 10^26 sekund ≈ **10^18 lat**

Klucz 192-bitowy:

- Liczba możliwych kluczy: 2^192 =~ 10^57
- Czas złamania: 10^45 sekund =~ **10^37 lat**

Klucz 256-bitowy:

- Liczba możliwych kluczy: 2^256 =~ 10^77
- Czas złamania: 10^65 sekund =~ **10^57 lat**

Zadanie 1.2;

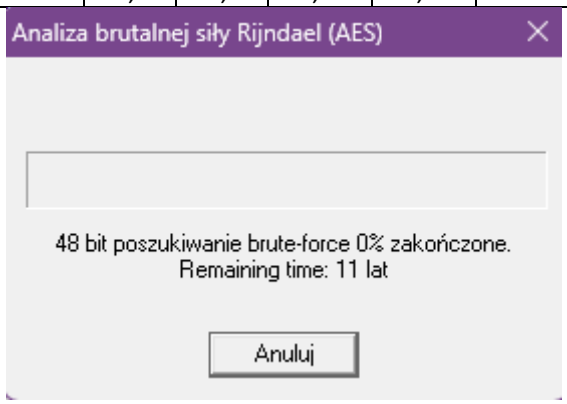
Czas złamania ataku brute-force dla każdego algorytmu 128-bitowego (AES, IDEA itd.) jest zbliżony, ponieważ przeszukiwanie kluczy zależy wyłącznie od długości klucza, a nie struktury algorytmu. Różnice są nieznaczne i pozostają na tym samym poziomie wielkości i czas poszukiwania będzie wynosić około **10^18 lat**. Wbudowane w cryptool narzędzie analizy ataku bruteforce pokazuje poniższe czasy, nieco różne od powyższych szacunków, ale również prawdopodobne i zbliżone na podobnej tej samej długości klucza:

Czas w zależności od długości klucza i algorytmu			
	IDEA	DES (CBC)	AES (CBC)
64 bity	X	2,6 * 10^4 lat	X
128 bitów	9,2 * 10^25 lat	X	1,3 * 10^25lat
192 bity	X	X	2,6 * 10^44 lat
256 bitów	X	X	5,1 * 10^63 lat

Zadanie 1.3-1.4;

Czasy poszukiwania klucza (AES, klucz 128 bitów):

	Ilość nieznanych bitów									
	4bity	8bity	12bity	16bity	20bity	24bity	28bity	32bity	40bity	48bity
Losowo	<0,2s	<0,2s	<0,2s	<0,4s	2s	20s	5min 8s	1h 23min	14,9dni	11lat
Początek	<0,2s	<0,2s	<0,2s	<0,4s	2s	20s	5min 17s	1h 23min	15,0dni	11lat
Koniec	<0,2s	<0,2s	<0,2s	<0,4s	2s	20s	5min 11s	1h 24min	15,0dni	11lat



Wniosek 1.4: Pozycja nieznanych bitów klucza **nie wpływa** na czas poszukiwania klucza

Zadanie 1.5;

Ocena jakości działania algorytmu łamiącego jest zdecydowanie pozytywna.

- Dla wszystkich testów otrzymano poprawny klucz (z wyjątkiem testów trwających dłużej niż kilka-kilkanaście minut)
- Liczba szukanych bitów nie wpływa na jakość odtwarzanego klucza, ponieważ i tak przeszukujemy wszystkie możliwe ustawienia klucza, więc nie pomijamy tego poprawnego. Liczba szukanych bitów wpływa jedynie proporcjonalnie na czas szukania klucza.
- Pozycja nieznanych bitów nie wpływa na jakość odtwarzanego klucza.

Pytanie 1.6;

Tak, współczesne algorytmy blokowe, takie jak AES z kluczami 128, 192 i 256-bitowymi, można uznać za bezpieczne. Dzięki długości klucza i zaawansowanej konstrukcji są odporne na ataki brute-force, które zajmowałyby miliardy lat przy aktualnych możliwościach obliczeniowych. Algorytmy te są również zaprojektowane z myślą o odporności na bardziej złożone ataki kryptoanalityczne, co sprawia, że są bezpiecznym wyborem w większości współczesnych zastosowań.

Pytanie 1.7;

Klucz o długości 128 bitów oferuje obecnie wystarczająco dobry poziom bezpieczeństwa, ponieważ atak brute-force wymagałby miliardów lat przy współczesnych możliwościach obliczeniowych komputerów, co sprawia, że taki atak staje się niepraktyczny. Przy długości 128-bitów klucz ma 2^128 możliwych kombinacji, co jest poza zasięgiem nawet najbardziej zaawansowanych komputerów. Dla zastosowań o wyjątkowo wysokich wymaganiach bezpieczeństwa, takich jak dane rządowe, rekomendowane są klucze 192 lub 256 bitowe, które zapewniają dodatkową odporność na przyszłe zagrożenia związane z rozwojem komputerów, w tym potencjalnie komputerów kwantowych.

Pytanie 1.8;

Tak, wielkość kryptogramu może mieć wpływ na możliwość jego złamania, ale nie w kontekście ataków brute-force na klucz. Dłuższy kryptogram daje więcej zaszyfrowanych danych, co może ułatwić zastosowanie zaawansowanych metod analitycznych, takich jak ataki statystyczne, różnicowe lub zależności liniowych, które opierają się na analizie wzorców i zależności. W skrócie: im więcej danych zaszyfrowanych jednym kluczem, tym większa szansa, że znajdą się wskazówki umożliwiające złamanie szyfru bez przeszukiwania wszystkich możliwych kluczy.

Pytanie 1.9;

Tak, format i wcześniejsze przetwarzanie dokumentu mogą wpłynąć na możliwość jego kryptoanalizy. Na przykład kompresja danych przed szyfrowaniem usuwa powtarzające się wzorce, co utrudnia zastosowanie ataków opartych na analizie statystycznej histogramu. Natomiast niektóre formaty dokumentów mogą zawierać znane nagłówki lub struktury, które po zaszyfrowaniu tworzą przewidywalne wzorce, co może ułatwić kryptoanalizę (atak znanego tekstu jawnego). Dlatego lepiej jest stosować kompresję i usuwać zbędne metadane przed szyfrowaniem, aby zwiększyć bezpieczeństwo danych.

Pytanie 1.10;

Obliczenia:

1 000 000 haseł * 60s * 60min * 24h * 365 dni = **3,15 * 10^13 haseł**

Jest to 31,5 biliona haseł - liczba ta jest zbyt mała, aby zagrozić współczesnym algorytmom symetrycznym z kluczami o długości 128 bitów lub większej, gdzie liczba możliwych kluczy wynosi 2^128, czyli około **10^38** kombinacji. Oznacza to, że bruteforce odpalony na jednym komputerze jest nieskuteczny przeciwko współczesnym algorytmom szyfrującym, co potwierdza ich odporność na tego typu ataki.

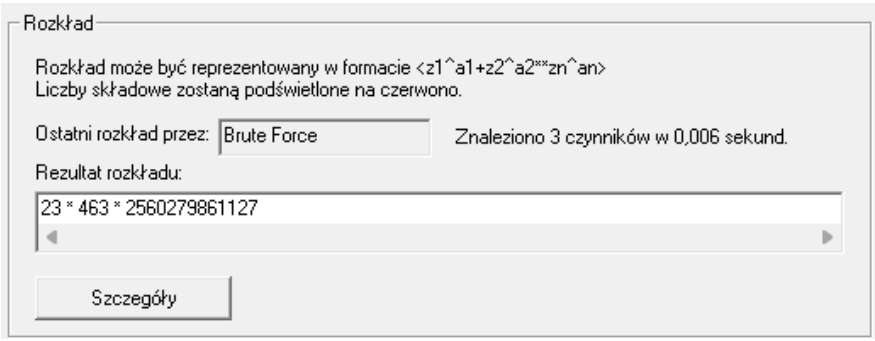
Zadanie 2.1;

Obliczenia:

- Numer indeksu: 272644
- Rok: 2024
- Miesiąc: 11
- Dzień: 4
- Godzina: 14
- Minuta: 23

Liczba: 27264420241141423

$$27264420241141423 = 23 \times 463 \times 2560279861127$$



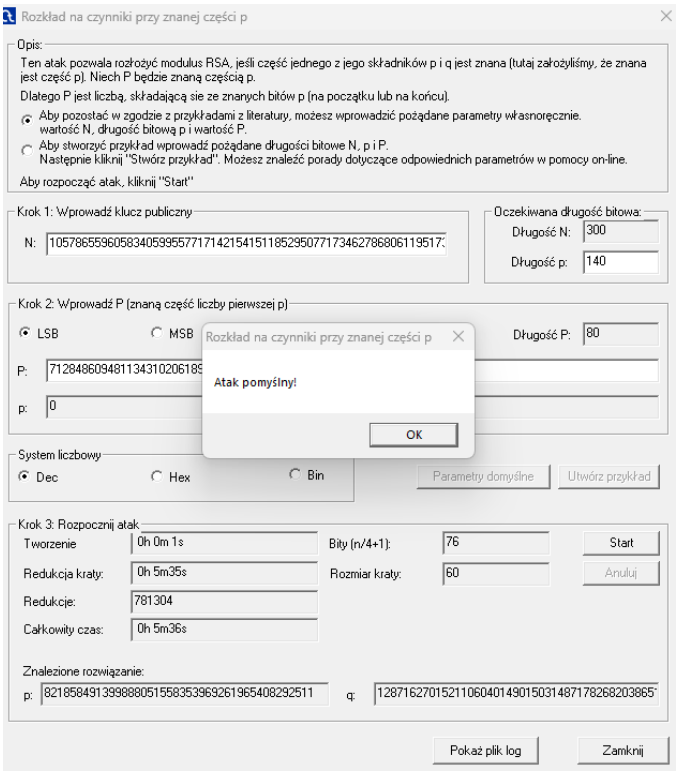
Zadanie 2.2;

Liczby pierwsze:

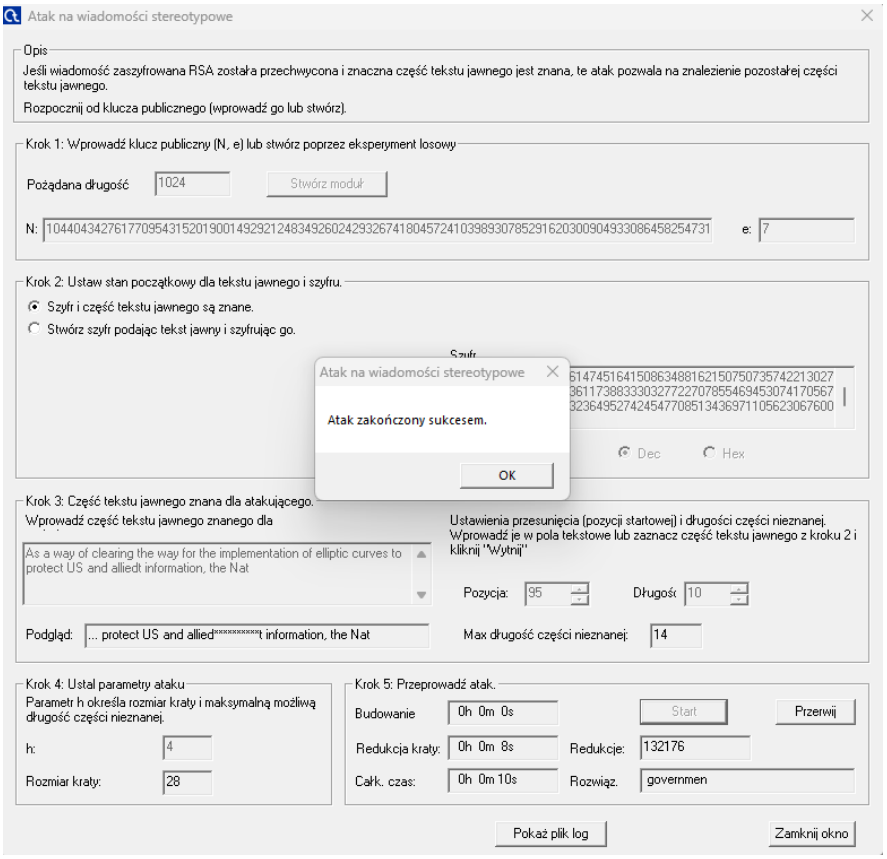
przedział	0-2^8	0-2^12	0-2^16	0-2^18	0-2^20	0-2^22
czas	<0,2s	<0,5s	4s	12s	40s	2min 42s

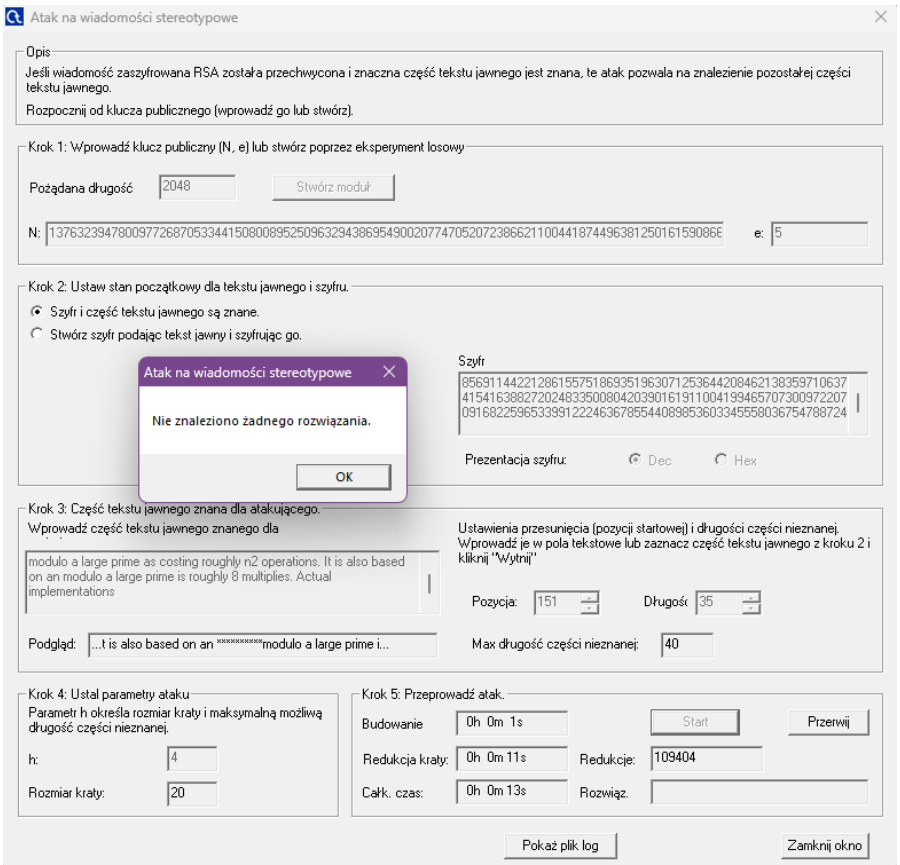
Zadanie 2.3;

Czas potrzebny na atak faktoryzacji modułu N algorytmu RSA			
dł. N	dł. p	dł. P	czas
200	80	80	1s
200	120	80	1s
200	80	60	1s
300	120	120	1s
300	150	100	1s
300	150	90	1s
300	150	80	6min 20s
300	140	80	5min 35s
500	200	141	18s (niepowodzenie)
500	250	160	1s

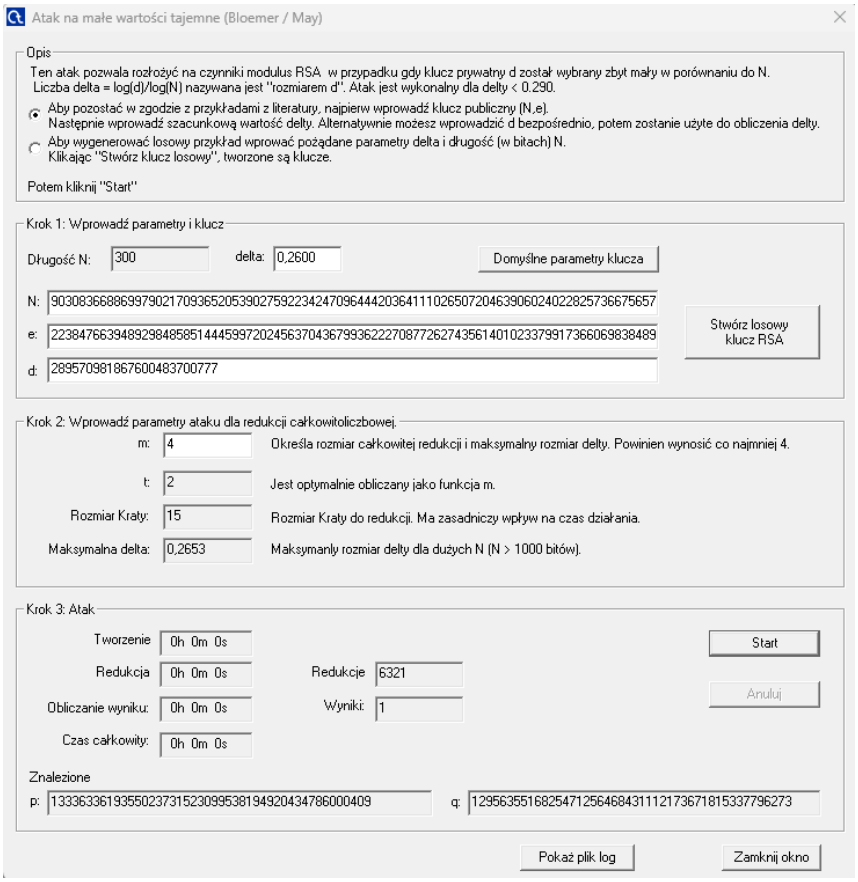


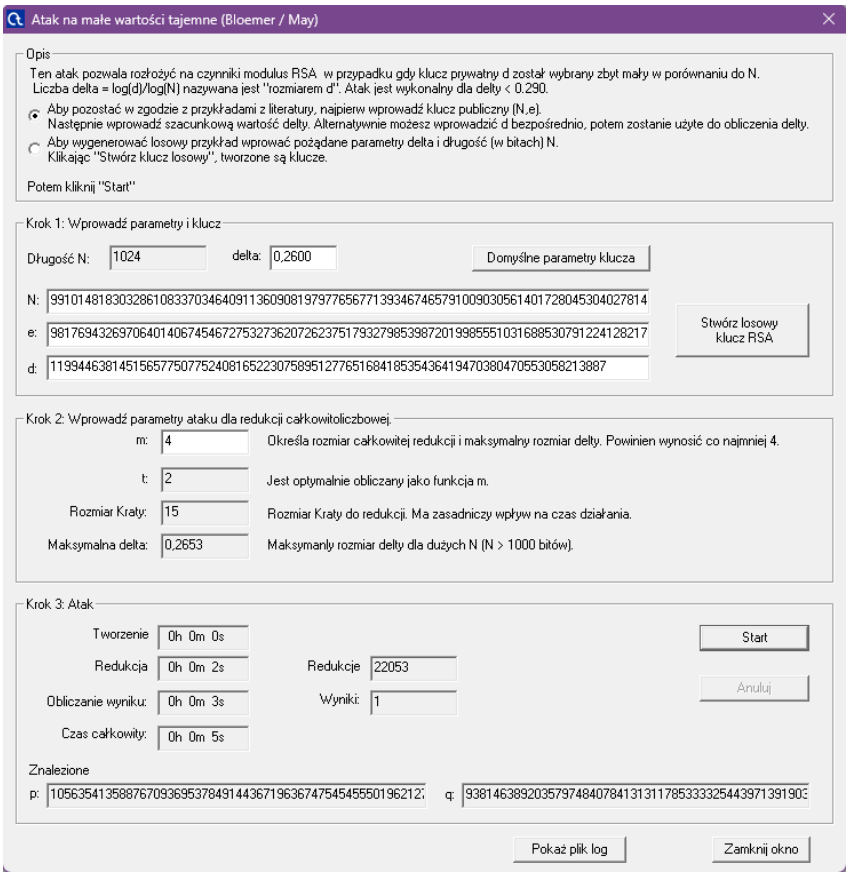
Zadanie 2.4;

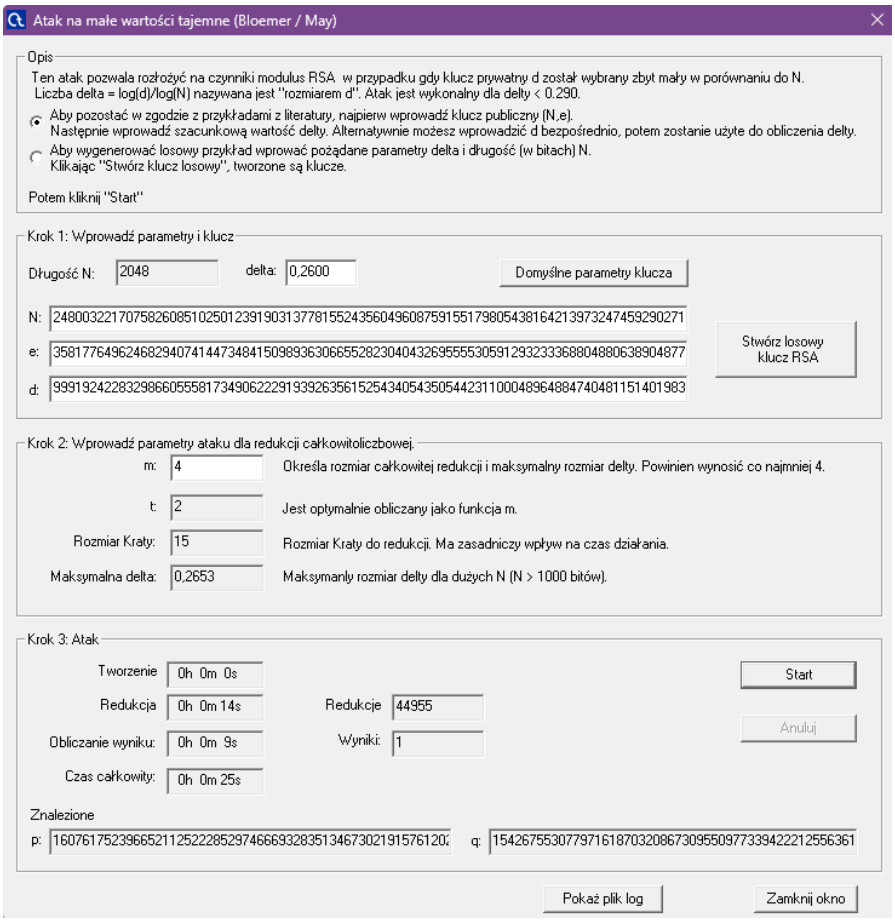

Atak na wiadomości stereotypowe
Opis
Jeśli wiadomość zaszyfrowana RSA została przechwycona i znaczna część tekstu jawnego jest znana, te atak pozwala na znalezienie pozostałej części tekstu jawnego.
Rozpocznij od klucza publicznego (wprowadź go lub stwórz).
Krok 1: Wprowadź klucz publiczny (N, e) lub stwórz poprzez eksperyment losowy
Pożądana długość
1024
Stwórz moduł
N:
e:
7
Krok 2: Ustaw stan początkowy dla tekstu jawnego i szyfru.
Szyfr i część tekstu jawnego są znane.
Stwórz szyfr podając tekst jawny i szyfrując go.
Szyfr
Atak na wiadomości stereotypowe
Atak zakończony sukcesem.
OK
Dec
Hex
Krok 3: Część tekstu jawnego znana dla atakującego.
Wprowadź część tekstu jawnego znanego dla
As a way of clearing the way for the implementation of elliptic curves to protect US and alliedt information, the Nat
Ustawienia przesunięcia (pozycji startowej) i długości części nieznanej.
Wprowadź je w pola tekstowe lub zaznacz część tekstu jawnego z kroku 2 i kliknij "Wytnij"
Pozycja:
95
Długość
10
Podgląd:
... protect US and allied**********t information, the Nat
Max długość części nieznanej:
14
Krok 4: Ustal parametry ataku
Parametr h określa rozmiar kraty i maksymalną możliwą długość części nieznanej.
h:
4
Rozmiar kraty:
28
Krok 5: Przeprowadź atak.
Budowanie
0h 0m 0s
Start
Przerwij
Redukcja kraty:
0h 0m 8s
Redukcje:
132176
Całk. czas:
0h 0m 10s
Rozwiąz.
governmen
Pokaż plik log
Zamknij okno


Atak na wiadomości stereotypowe
Opis
Jeśli wiadomość zaszyfrowana RSA została przechwycona i znaczna część tekstu jawnego jest znana, te atak pozwala na znalezienie pozostałej części tekstu jawnego.
Rozpocznij od klucza publicznego (wprowadź go lub stwórz).
Krok 1: Wprowadź klucz publiczny (N, e) lub stwórz poprzez eksperyment losowy
Pożądana długość
2048
Stwórz moduł
N:
e:
5
Krok 2: Ustaw stan początkowy dla tekstu jawnego i szyfru.
Szyfr i część tekstu jawnego są znane.
Stwórz szyfr podając tekst jawny i szyfrując go.
Szyfr
Atak na wiadomości stereotypowe
Nie znaleziono żadnego rozwiązania.
OK
Prezentacja szyfru:
Dec
Hex
Krok 3: Część tekstu jawnego znana dla atakującego.
Wprowadź część tekstu jawnego znanego dla
modulo a large prime as costing roughly n2 operations. It is also based on an modulo a large prime is roughly 8 multiplies. Actual implementations
Ustawienia przesunięcia (pozycji startowej) i długości części nieznanej.
Wprowadź je w pola tekstowe lub zaznacz część tekstu jawnego z kroku 2 i kliknij "Wytnij"
Pozycja:
151
Długość
35
Podgląd:
...t is also based on an **********modulo a large prime i...
Max długość części nieznanej:
40
Krok 4: Ustal parametry ataku
Parametr h określa rozmiar kraty i maksymalną możliwą długość części nieznanej.
h:
4
Rozmiar kraty:
20
Krok 5: Przeprowadź atak.
Budowanie
0h 0m 1s
Start
Przerwij
Redukcja kraty:
0h 0m 11s
Redukcje:
109404
Całk. czas:
0h 0m 13s
Rozwiąz.
Pokaż plik log
Zamknij okno

Zadanie 2.5;


Atak na małe wartości tajemne (Bloemer / May)
Opis
Ten atak pozwala rozłożyć na czynniki modulus RSA w przypadku gdy klucz prywatny d został wybrany zbyt mały w porównaniu do N.
Liczba delta = log(d)/log(N) nazywana jest "rozmiarem d". Atak jest wykonalny dla delty < 0.290.
Aby pozostać w zgodzie z przykładami z literatury, najpierw wprowadź klucz publiczny (N,e).
Następnie wprowadź szacunkową wartość delty. Alternatywnie możesz wprowadzić d bezpośrednio, potem zostanie użyte do obliczenia delty.
Aby wygenerować losowy przykład wprowadź pożądane parametry delta i długość (w bitach) N.
Klikając "Stwórz klucz losowy", tworzone są klucze.
Potem kliknij "Start"
Krok 1: Wprowadź parametry i klucz
Długość N:
300
delta:
0,2600
Domyślne parametry klucza
N:
e:
d:
Stwórz losowy klucz RSA
Krok 2: Wprowadź parametry ataku dla redukcji całkowitoliczbowej.
m:
4
Określa rozmiar całkowitej redukcji i maksymalny rozmiar delty. Powinien wynosić co najmniej 4.
t:
2
Jest optymalnie obliczany jako funkcja m.
Rozmiar Kraty:
15
Rozmiar Kraty do redukcji. Ma zasadniczy wpływ na czas działania.
Maksymalna delta:
0,2653
Maksymalny rozmiar delty dla dużych N (N > 1000 bitów).
Krok 3: Atak
Tworzenie
0h 0m 0s
Redukcja
0h 0m 0s
Redukcje
6321
Obliczanie wyniku:
0h 0m 0s
Wyniki:
1
Czas całkowity:
0h 0m 0s
Start
Anuluj
Znalezione
p:
q:
Pokaż plik log
Zamknij okno


Atak na małe wartości tajemne (Bloemer / May)
Opis
Ten atak pozwala rozłożyć na czynniki modulus RSA w przypadku gdy klucz prywatny d został wybrany zbyt mały w porównaniu do N.
Liczba delta = log(d)/log(N) nazywana jest "rozmiarem d". Atak jest wykonalny dla delty < 0.290.
Aby pozostać w zgodzie z przykładami z literatury, najpierw wprowadź klucz publiczny (N,e).
Następnie wprowadź szacunkową wartość delty. Alternatywnie możesz wprowadzić d bezpośrednio, potem zostanie użyte do obliczenia delty.
Aby wygenerować losowy przykład wprowadź pożądane parametry delta i długość (w bitach) N.
Klikając "Stwórz klucz losowy", tworzone są klucze.
Potem kliknij "Start"
Krok 1: Wprowadź parametry i klucz
Długość N:
1024
delta:
0,2600
Domyślne parametry klucza
N:
e:
d:
Stwórz losowy klucz RSA
Krok 2: Wprowadź parametry ataku dla redukcji całkowitoliczbowej.
m:
4
Określa rozmiar całkowitej redukcji i maksymalny rozmiar delty. Powinien wynosić co najmniej 4.
t:
2
Jest optymalnie obliczany jako funkcja m.
Rozmiar Kraty:
15
Rozmiar Kraty do redukcji. Ma zasadniczy wpływ na czas działania.
Maksymalna delta:
0,2653
Maksymalny rozmiar delty dla dużych N (N > 1000 bitów).
Krok 3: Atak
Tworzenie
0h 0m 0s
Redukcja
0h 0m 2s
Redukcje
22053
Obliczanie wyniku:
0h 0m 3s
Wyniki:
1
Czas całkowity:
0h 0m 5s
Start
Anuluj
Znalezione
p:
q:
Pokaż plik log
Zamknij okno



Atak na mały klucz tajny był skuteczny dla wszystkich trzech przypadków. Najdłuższy czas (25s) dla najdłużej długości N, a najkrótszy dla najmniejszego N. We wszystkich przypadkach czas ten był relatywnie krótkim.

Pytanie 2.6;

Aby rozkład liczby N w algorytmie RSA był dostatecznie trudny, minimalna rekomendowana długość modułu wynosi obecnie 2048 bitów. Taka długość zapewnia odporność na współczesne ataki faktoryzacyjne, wymagające dużych zasobów obliczeniowych. Moduły RSA krótsze niż 2048 bitów (np. 1024-bitowe) uznaje się za podatne na złamanie przez zaawansowane techniki rozkładu na czynniki pierwsze. Przy kluczach 2048-bitowych RSA jest uważane za bezpieczne w perspektywie najbliższych lat, choć w przyszłości, w związku z rozwojem komputerów kwantowych, standardem może stać się 3072 lub nawet 4096 bitów.

https://en.wikipedia.org/wiki/Key_size

„Since 2015, NIST recommends a minimum of 2048-bit keys for RSA, an update to the widely accepted recommendation of a 1024-bit minimum since at least 2002."

Pytanie 2.7;

Kiedy znamy wystarczająco sporo bitów jednej z liczb (ponad 50%) to jesteśmy w stanie przeprowadzić skuteczny atak, w relatywnie rozsądnym czasie nawet dla dłuższych i uważanych za bezpiecznych długości modułu (liczba N). Jeśli atakujący zna jeden z tych czynników, może łatwo obliczyć drugi, a tym samym odzyskać klucz prywatny, co pozwoli na odszyfrowanie wiadomości.

Pytanie 2.8;

Szyfrowanie algorytmem RSA może być zagrożone przez atak na wiadomości stereotypowe w przypadku, gdy wiadomości są krótsze niż długość modułu N, gdy zaszyfrowane wiadomości nie różnią się zbytnio od siebie oraz gdy znamy znaczną część tekstu jawnego lub gdy do szyfrowania użyto klucza o małej długości

Pytanie 2.9;

Kiedy klucz publiczny użyty do szyfrowania jest znany oraz klucz prywatny jest zbyt małej długości (mniejszej niż 512 bitów). Przestrzeń możliwych kluczy zostaje zmniejszona drastycznie i pozwala to skutecznie złamać szyfrowanie.