Prosopon
Europejskie Studia Społeczno-Humanistyczne

Wydawca / Publisher:
Instytut Studiów
Międzynarodowych
i Edukacji HUMANUM
www.humanum.org.pl

11 (2) 2015
ISSN 1730–0266

Michał Stachura
Poland
E-mail: m.stachura@stawil.pl

Charakterystyka podstawowych statycznych metod szyfrowania informacji / *Presentation of basic static methods of information clearance*

Abstract
Information protection methods are to ensure the safe operation of data against unauthorized access. The magnitude that characterizes this safe operation of data is called the effectiveness of the applied algorithm. Efficiency is understood as the degree of information protection against infiltration. Unfortunately, all the previous studies and analyzes of individual methods do not take into account the decisive role of man, which has a great influence on the effectiveness of the applied algorithm.
The infiltration, or deliberate infiltration of unauthorized persons into the information collection, is divided into accidental and intentional, in turn, we divide into passive and active. For military computer systems one of the ways of infiltration can be described as computer espionage, which is particularly dangerous.

Key words: **infiltration, information.**

Metody ochrony informacji mają gwarantować bezpieczną eksploatację danych przed dostępem osób nieupoważnionych. Wielkością, która cechuje tą bezpieczną eksploatacje danych nazywamy skutecznością zastosowanego algorytmu. Skuteczność jest rozumiana jako stopień zabezpieczenia informacji przed infiltracją. Niestety we wszystkich dotychczasowych badaniach i analizach poszczególnych metod nie uwzględniono decydującej roli człowieka, która ma ogromny wpływ na skuteczność zastosowanego algorytmu.

Infiltracja, czyli celowe przenikanie nieupoważnionych osób do zbiorów informacji dzieli się na przypadkową i celową, te z kolei dzielimy na pasywną i aktywną.

Dla wojskowych systemów komputerowych jeden ze sposobów infiltracji możemy określić jako szpiegostwo komputerowe, które jest szczególnie niebezpieczne.

- Infiltracja pasywna jest bardzo często określana mianem infiltracji celowej. Osoba stosująca ten sposób infiltracji najczęściej podłącza się do przewodów transmisji danych i nasłuchuje na nich przebieg odpowiednich pakietów. Jeżeli uda jej się przechwycić coś interesującego, najczęściej potrafi z tych informacji skorzystać, ponieważ jest osobą wyspecjalizowana w tego typu działaniach. Formy infiltracji pasywnej to również bezpośrednie kradzieże z biur nośników informacji np. płyt CD, dyskietek itd.
- Infiltracja aktywna opiera się o takie metody jak np. nielegalne korzystanie z komputera w czasie prac konserwatorskich, uzyskiwanie dostępu do systemu przez osoby nieupoważnione, uzyskiwanie potwierdzenia tożsamości lub hasła prawidłowego użytkownika itd.

Innym rodzajem infiltracji aczkolwiek bardzo niebezpiecznym jest szpiegostwo i oszustwo, szczególnie szpiegostwo gospodarczych systemów komputerowych, które przynosi ogromne straty przedsiębiorstw a w systemach wojskowych może przynieść nieoszacowane szkody.

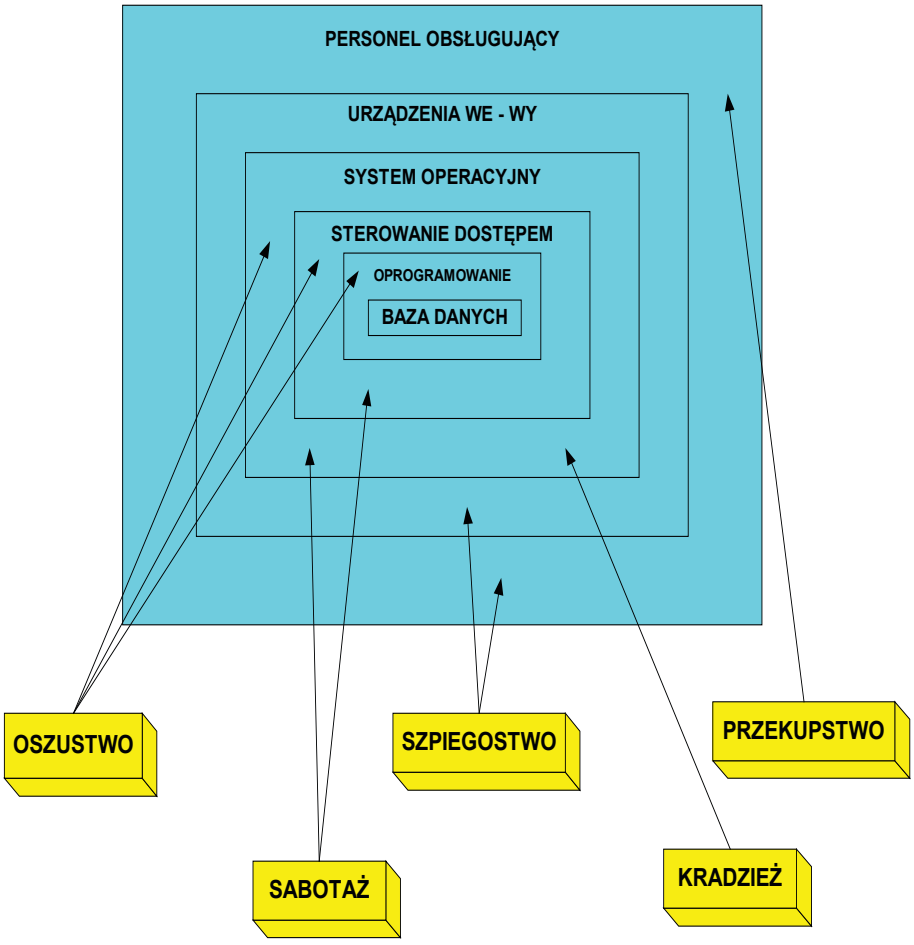
Można wyróżnić pięć podstawowych grup infiltracji, mianowicie:

1. przekupstwo
2. oszustwo
3. kradzież
4. szpiegostwo
5. sabotaż

Osoby, które chcą przechwycić informacje, albo uzyskać dostęp do baz danych mogą oddziaływać na następujące osoby:

- personel obsługujący
- system operacyjny
- oprogramowanie użytkowe
- urządzenia WE-WY
- pamięć zewnętrzna
- linie transmisji danych
- sterowanie dostępem do bazy danych

Na poniższym schemacie przedstawione zostały zależności pomiędzy grupami działań infiltracyjnych i elementami, które są związane z przetwarzaniem danych.



Ankieta przeprowadzona i opracowana przez Sokołowskiego na temat ochrony zbiorów informacji stwierdza, że oprócz metod organizacyjnych żadne inne metody nie są stosowane, nie biorąc pod uwagę zabezpieczeń zapewnionych przez system operacyjny, które w rzeczywistości stały się zwykła formalnością.

Skuteczność metod ochrony informacji najlepiej podzielić na:

1. skuteczność teoretyczną
2. skuteczność rzeczywistą inaczej pisząc praktyczną

W opisanych poniżej metodach szyfrowania informacji postaram się porównać obie skuteczności i przedstawić wiarygodne wyniki swojej pracy.

1.1 METODY SYMETRYCZNE

W tym podrozdziale przedstawię najbardziej znane symetryczne metody szyfrowania informacji, czyli metody wykorzystujące jeden stały klucz, który znają obie strony, nadawca meldunku jak i odbiorca. Metody te, mimo że już bardzo rzadko

wykorzystywane dzisiaj stanowią początek i filar ówczesnej kryptografii. Od nich wszystko się zaczęło i trwa po dzisiejszy dzień.

1.1.1 SZYFR CEZARA

XX wieków temu szyfr ten był jednym z nielicznych szyfrów, w którym entropia rosła proporcjonalnie do długości ciągu szyfrowego. Starożytni Rzymianie bardzo często stosowali tą metodę do zabezpieczania meldunków. Szyfr Cezara wykorzystywał najprostszą sposób na szyfrowanie, mianowicie każda litera w alfabecie była zamieniana na literę występującą trzy miejsca dalej. Alfabet był zapisywany na obwodzie koła, następnikiem Z była litera A.

Przebieg takiego szyfrowania wyglądał następująco:

B → E

C → F

…….

Y → B

Z → C

Tekst jawny nie zawierał spacji a duże i małe litery nie były rozróżniane.

Kluczem w tym szyfrowaniu informacji jest 3 (była to liczba kroków, o jaką należy przesunąć do przodu alfabet, by otrzymać szyfrogram)

Jak widać istnieje 25 możliwości uzyskania szyfrogramu. Jest to bardzo prosta metoda, jednak 2000 lat temu okazała się bardzo skuteczna.

W języku matematycznym metodę tą możemy zapisać jako:

C = P + S mod 26, jest to tzw. dodawanie reszt z dzielenia przez 26

P – litera tekstu jawnego

S – klucz (stała wartość)

C – otrzymana litera szyfru

Należy pamiętać o tym, że litery traktować powinniśmy jako liczby, zaczynając od litery A i przyporządkowując jej 0, poprzez B = 1 ….., aż do Z = 25. Jeżeli P + S jest większe od 26 to od sumy odejmujemy 26. Jeżeli ktoś chce złamać tą metodę to w ostateczności musi sprawdzić 25 kluczy, ponieważ 0 nie zmienia tekstu jawnego.

Do dzisiaj stosuje się w systemach Unix metodę opartą na szyfrze Cezara, zwaną ROT13, tutaj kluczem jest stała liczba = 13. Podwójne szyfrowanie tą metodą daje w rzeczywistości tekst jawny. W szyfrze Cezara należy zwrócić uwagę na to, że jednakowe znaki przyjmują taki sam szyfr.

1.1.2 SZYFR VIGENERA

Szyfr Vigenere'a różni się od szyfru Cezara tylko zastosowanym kluczem i ilością kombinacji z tym związaną. Jednak i tu występuje numerowanie liter alfabetu, czyli przypisywanie im określonych wartości. Informacje szyfrujemy za pomocą

klucza, metodą podstawienia. Łatwo dojść do wniosku, że jest to metoda Cezara, zmodyfikowana występującym tu kluczem. Moc kryptograficzna szyfru, dzięki zastosowaniu innego klucza, jakim jest słowo wzrosła. Jednak i w tej metodzie nie uniknęliśmy powtarzalności klucza. Szyfr Vigenere'a można opisać podobnie jak metodę Cezara wzorem matematycznym, który w tym wypadku przybiera następującą postać:

C = P + S mod 26, jest to tzw. dodawanie reszt z dzielenia przez 26

P – litera tekstu jawnego

S – klucz = określone słowo kluczowe np. ABCD

C – otrzymana litera szyfru

Widać tutaj pewną zmianę w porównaniu do szyfru Cezara, ponieważ odpowiednim literą tekstu jawnego nie muszą być przyporządkowane te same litery w kryptogramie. Prowadzi to do zaburzenia schematu, ale nie jest to na tyle trudne by nie dało się złamać tak uzyskanego szyfrogramu.

Sposób łamania szyfru Vigenere'a omówię w stosownym do tego czasie.

Ponieważ dzisiejsze komputery operują na bitach i bajtach, dlatego wymyślono binarny szyfr Vigenere'a. Bit jest literą z dwuelementowego alfabetu, w którym znajdujemy tylko „0" i „1", literę tą zapisujemy jako L. Dodawanie modulo 2 określone na tym dwuelementowym alfabecie odpowiada binarnej operacji XOR, czyli tzw. różnicy symetrycznej.

Dla różnicy symetrycznej:

0 + 0 = 0

0 + L = 1

L + 0 = 1

L + L = 0

Kluczem w tym wypadku jest skończony ciąg, lecz w tym wypadku zamiast dodawać znaki, dodajemy bity. Deszyfrowanie następuje poprzez ponowne szyfrowanie z użyciem tego samego klucza. Do dnia dzisiejszego wykorzystuje się tą metodę szyfrowania informacji, jednak stosowanie binarnego szyfru Vigenere'a nie zmienia tej metody ani jej kryptoanalizy, która nie nastręcza zbyt wiele problemów.

Szyfr ten wymaga spełnienia podstawowego warunku:

N ≤ K, gdzie K – długość klucza a N – długość przesłanej wiadomości.

1.1.3 SZYFR VERNAMA

Szyfr ten różni się tym od szyfru Vigenere'a, że kluczem jest odpowiednio długie słowo.

Szyfr ten wymaga spełnienia podstawowego warunku:

N ≤ K

K – długość klucza

N – długość przesłanej wiadomości.

Szyfr Vernama, mimo że mógłby spełnić własności szyfru idealnego nie nadaje się do szyfrowania informacji, ponieważ na każdy bajt wiadomości musiałby przypadać bajtowy klucz, co przy dzisiejszych rozmiarach przesyłanych informacji, lub wielkościach baz danych nie jest możliwe do wprowadzenia na rynek. Na 25 Gb informacji musiałoby przypadać 25 Gb klucza. Do tego klucz jest wartością stałą, którą też należy odpowiednio zabezpieczać.

1.2 METODY ASYMETRYCZNE

W poprzednich rozdziałach zajmowałem się przedstawieniem podstawowych metod szyfrowania symetrycznego, w których klucz szyfrujący jest też używany do rozszyfrowywania wiadomości. Często w metodach symetrycznych bywało tak, że do odszyfrowywania wiadomości potrzebna była inna procedura niż do szyfrowania (Vigenere, Cezar itd.), ale klucz ciągle pozostawał ten sam. Symetria tego rodzaju tyczyła się klucza, nigdy nie metody.

W szyfrach asymetrycznych jest zupełnie inaczej. Metody te zwane także metodami klucza publicznego często posiadają dwie odrębne procedury szyfrowania i odszyfrowywania, ale posiadają też dwa odrębne klucze. Klucz prywatny (private key) jest ściśle związany z algorytmem deszyfrującym a klucz publiczny (public key) z algorytmem szyfrującym. Mimo, że widać tu pewną symetrie pomiędzy tymi kluczami są one zupełnie różne. Asymetria w tym wypadku polega na tym, że za pomocą klucza publicznego ciężko wyliczyć klucz prywatny (w sensie kryptologicznym), natomiast obliczenie klucza publicznego za pomocą tajnego staje się wykonalne.

Metody asymetryczne umożliwiają nam szyfrowanie informacji za pomocą naszego klucza publicznego przesłanego do osoby, która takową wiadomość chce nam przesłać.

Klucz publiczny możemy dać każdemu – bez obawy o bezpieczeństwo klucza prywatnego[1]. Klucz prywatny jest tylko naszą własnością, którą nie ujawniamy nikomu. Osoba chcąca przesłać nam wiadomość w formie zaszyfrowanej, wykorzystuje do tego nasz klucz publiczny, którym szyfruje wiadomość przesyłaną do nas. Po otrzymaniu szyfrogramu, osoba mająca klucz prywatny w bardzo prosty sposób deszyfruje wiadomość. Klucz publiczny rozpowszechniamy tylko po to by móc szyfrować wiadomość a nigdy po to, żeby móc ją odszyfrowywać. Dlaczego tak dostępne stało się stosowanie metod asymetrycznych?, ponieważ z zaszyfrowanego kluczem publicznym ciągu nie można odczytać jawnej treści[2] nikt nie może wyliczyć klucza publicznego za pomocą klucza prywatnego[3]. Algorytmy asymetryczne stały się tak praktyczne, że w bardzo krótkim czasie wyparły wysłużone już algorytmy symetryczne. Niestety metody te są zbyt wolne, by szyfrować nimi

1 Reinhard Wobst, Kryptologia – Budowa i łamanie zabezpieczeń, Warszawa 2002, RM, s. 138
2 j.w.
3 j.w.

dłuższe wiadomości. Nadają się one jedynie do szyfrowania klucza sesji, dlatego wykorzystuje się je np. w podpisach cyfrowych.

1.2.1 METODA RSA

Metoda RSA, zwana metodą klucza publicznego jest najbardziej znaną metodą opartą o algorytm asymetryczny. Nazwa metody, tak jak to najczęściej bywa w kryptologii jest sumą trzech pierwszych liter nazwisk jej twórców: Rona Rivesta, Adi Shamira, Leonarda Adlemana. Metoda ta została opublikowana w roku 1978. Algorytm opiera się o faktoryzacje bardzo dużych liczb pierwszych. Algorytm ten utracił prawa patentowe 20 września 2000 roku. Do dzisiaj jest wykorzystywany w przesyłaniu kluczy sesji i stosowany między innymi w podpisie cyfrowym. W algorytmie wykorzystano tez małe twierdzenie Fermata (przystawanie modulo n). Dział matematyki zajmujący się kongruencją(przystawaniem) nazywany jest działem arytmetyki modularnej. W arytmetyce tej operuje się tylko na liczbach całkowitych. Od roku 1978 systematycznie zwiększa się długość klucza, który stanowi o bezpieczeństwie tej metody. Obecnie za bezpieczny uchodzi klucz 1024 bitowy. Algorytm RSA cechuje prostota w budowie i implementacji, wytrzymał lata bardzo intensywnej kryptoanalizy, ale tylko dzięki ciągłemu zwiększaniu długości klucza co automatycznie wiązało się ze znajdowaniem coraz to większych liczb pierwszych, stosowanych w tym algorytmie. Tak jak pisałem wcześniej, zabezpieczenie metody RSA polega na trudności faktoryzacji dużych liczb pierwszych. Znając tylko wartość iloczynu n = p * q dwóch liczb pierwszych, niezwykle trudno jest znaleźć oba jego czynniki (tj. p i q)[4].

Klucze prywatne i publiczne mają niekiedy po 100 lub więcej cyfr i są funkcjami pary dużych liczb. Algorytm ten jest bardzo wolnym algorytmem i mimo, że w latach osiemdziesiątych twórcy tej metody twierdzili, że nadaje się do szyfrowania informacji, tak naprawdę wraz ze wzrostem wymagań i zwiększaniem licz pierwszych używanych do szyfrowania, zmniejszała się szybkość algorytmu. Obecnie algorytm ten nie nadaje się do szyfrowania informacji. Jest natomiast wykorzystywany do przesyłania kluczy sesji a dalsza część przesyłania informacji jest szyfrowana algorytmem symetrycznym.

Niestety do utajniania informacji w komputerach osobistych, lub baz danych, ten bardzo sprawdzony algorytm nie nadaje się. Kryptoanalitycy tak naprawdę nigdy nie udowodnili ani też nie zdyskwalifikowali bezpieczeństwa tego algorytmu. Kryptoanaliza sugeruje poziom zaufania w teoretycznych ocenach tego algorytmu[5].

Żeby uzyskać klucz publiczny i prywatny, należy wybrać dwie duże liczby pierwsze p i q o jednakowej długości, wtedy zabezpieczenie będzie maksymalne. Należy obliczyć iloczyn tych dwóch ogromnych liczb pierwszych:

n = p * q

4 Reinhard Wobst, Kryptologia – Budowa i łamanie zabezpieczeń, Warszawa 2002, RM, s. 146
5 Bruce Schneier, Kryptografia dla praktyków, Warszawa 2002, NT, s. 572

Następnym krokiem jest wybranie losowego klucza szyfrującego **e**, który jest liczbą względnie pierwszą z (p -1) * (q – 1). Najczęściej wybieraną liczbą **e,** która ma zagwarantować szybkość metody jest 3, 17 lub 65 537, które jest równe $2^{16} + 1$. Dla tych to wartości, można bardzo szybko obliczyć $\mathbf{m^e}$**.**

Teraz znając n, e, p i q obliczamy za pomocą algorytmu Euklidesa klucz prywatny **d**, który opisany jest wzorem:

$$d = e^{-1} \bmod (p - 1)\ (q - 1).$$

Otrzymane liczby d i n, są liczbami względnie pierwszymi.

Liczba **d** jest kluczem prywatnym – ściśle tajnym, tylko dla wiadomości właściciela.

Liczby **e** i **n** są kluczem publicznym, są to wartości znane – jawne.

Następnie twórcy metody RSA, każą dwie liczby pierwsze (p i q), z których po paru przekształceniach otrzymaliśmy klucz publiczny i prywatny wymazać z pamięci i nigdy ich nie ujawniać. Wymazać z pamięci oznacza wykasowanie pamięci zewnętrznej (mózg człowieka) oraz pamięci wewnętrznej komputera, dzięki któremu obliczyliśmy te wartości.

W celu zaszyfrowania wiadomości **m**, najpierw dzielimy ją na bloki mniejsze niż **n** (dla danych w postaci binarnej wybieramy największą potęgę 2 mniejszą niż **n**).[6]

Oznacza to, że jeśli zarówno p, jak i q są liczbami pierwszymi o 100 cyfrach, to liczba n będzie miała mniej niż 200 cyfr.[7]

Szyfrowanie odbywa się za pomocą wzoru:

$$c = m^e \pmod n$$

Odszyfrowywanie natomiast za pomocą wzoru:

$$m = c^d \pmod n$$

Należy pamiętać o tym, że szyfrujemy blokami. Bloki według twórców metody możemy uzupełniać zerami z lewej strony, tak żeby były one zawsze mniejsze od **n**.

Aby odszyfrować wiadomość, bierze się każdy z zaszyfrowanych bloków i dokonujemy obliczeń zgodnie z modelem deszyfracji otrzymując jawną postać wiadomości.

Skrócony opis algorytmu RSA[8]:

Klucz publiczny:

n iloczyn dwóch liczb pierwszych p i q (p i q muszą zostać utajnione)

6 Bruce Schneier, Kryptografia dla praktyków, Warszawa 2002, NT, s. 572
7 j.w.
8 Bruce Schneier, Kryptografia dla praktyków, Warszawa 2002, NT, s. 573

e liczba względnie pierwsza z (p – 1) (q – 1)

Klucz prywatny:

d = e^{-1} (mod (p -1) (q – 1))

Szyfrowanie:

c = m^e (mod n)

Odszyfrowywanie:

m = c^d (mod n)

W metodzie RSA wykorzystuje się tabele przyporządkowania, która zamienia litery na liczby według następującego schematu:

A = 01 B = 03 C = 07 D = 04 E = 05 F = 10 G = 09 H = 08 I = 02 itd.

Szyfrowania dokonujemy na blokach podzielonej wcześniej wiadomości. Jeżeli chcemy zaszyfrować ciąg znaków EG, wcześniej odczytujemy z tabeli przyporządkowanie poszczególnych liter do liczb, w tym wypadku jest to E = 05 i G =09, co zapisujemy jako **m = 509**.

Następnie po wybraniu ściśle tajnych licz p i q otrzymuj wartość n, niech to będzie dla przykładu wartość **n = 52961**.

Teraz wybieramy liczbę względnie pierwszą z (p – 1) (q – 1), niech w naszym przykładzie będzie to liczba **e = 131.**

Wielkości te podstawiamy do wzoru przeznaczonego do szyfrowania informacji, czyli:

$c = m^e$ (mod n), więc otrzymujemy

$c = 509^{131}$ (mod 52961)

kolejne potęgi wyrażenia otrzymujemy według następujących zasad:

509^1 mod 52961 = 509
509^2 mod 52961 = 47237
509^4 mod 52961 = 34278
509^8 mod 52961 = 41499
509^{16} mod 52961 = 34164
509^{32} mod 52961 = 24378
509^{64} mod 52961 = 11503
509^{128} mod 52961 = 22431

Potęgę liczby 131 otrzymujemy z sumy liczb: 128 + 2 + 1:

(128 + 2); 22431 * 47237 (mod 52961) = 35381

(… + 1); 35381 * 509 (mod 52961) = 2189

Tą wielkość przesyłamy do odbiorcy jako szyfr c, są to dwie litery E i G zaszyfrowane za pomocą algorytmu RSA.

Odbiorca po odebraniu zaszyfrowanej wiadomości rozpoczyna odszyfrowywanie jej za pomocą wzoru $\mathbf{m = c^d \ (mod\ n)}$ i otrzymuje:

$$\mathbf{m = 2189^{37271} \ (mod\ 52961)}$$

Adresat otrzymuje w wyniku końcowym wielkość m = 509, znając zasadę szyfrowania przyporządkowuje jednemu znakowi dwie cyfry, zaczynając od strony lewej i otrzymuje wynik 05 i 09. Zgodnie z tabelą przyporządkowania zamienia te wielkości na przypisane im litery i otrzymuje E i G, czyli postać jawną zaszyfrowanej wiadomości.

Oto krótki przykład według Bruce Schneier'a.[9]

Wybieramy p = 47 i q = 71, otrzymujemy zgodnie ze wzorami;

n = pq = 3337

obliczmy:

(p – 1) (q – 1) = 46 * 70 = 3220

Wybieramy e (losowo) o wartości 79, otrzymujemy:

$\mathbf{d} = 79^{-1} \ (mod\ 3220) = 1019$

Publikujemy **e** i **n**, a **d** utajniamy. Wymazujemy tez p i q.

Żeby zaszyfrować wiadomość m = 6882326879666683, dzielimy ją na trzycyfrowe bloki i otrzymujemy sześć bloków:

$m_1 = 688$
$m_2 = 232$
$m_3 = 687$
$m_4 = 966$
$m_5 = 668$
$m_6 = 003$

zaszyfrowany pierwszy blok oznaczmy jako c_1, więc mamy:

$\mathbf{c_1} = 688^{79} \ (mod\ 3337)$ i otrzymujemy wartość 1570. Dokonujemy dalszych obliczeń dla kolejnych bloków wiadomości. Bo dokonaniu wszystkich obliczeń zaszyfrowany ciąg cyfr zapisujemy jako:

$$\mathbf{c = 1570\ 2856\ 2091\ 2276\ 2423\ 158}$$

9 Bruce Schneier, Kryptografia dla praktyków, Warszawa 2002, NT, s. 573

Deszyfracja polega na wykonaniu tych samych obliczeń, tylko przy użyciu klucza deszyfrującego deszyfrującego wartości 1019, więc mamy dla c_1:

$1570^{1019} \pmod{3337} = 688 = \mathbf{m}_1$ i tak postępujemy dla kolejnych wartości c.

Zachodzi jednak jedno podstawowe pytanie, czy zamiana cyfr na inne cyfry może być nazwana szyfrowaniem?. Na to pytanie postaram się odpowiedzieć w stosownym do tego czasie.

Oczywiście istnieją też sprzętowe układy realizujące algorytm RSA, jednak najszybszy z nich układ pod nazwą VLSI realizuje algorytm RSA o długości 512 bitów.

RSA Security poinformował o złamaniu po 3 miesiącach 576-bitowego klucza szyfrującego. Dokonał tego ośmioosobowy zespół z Europy i Stanów Zjednoczonych przy użyciu około 100 komputerów. Ośmioosobowy zespół składa się z naukowców narodowości Niemieckiej pracujących w Scientific Computing Institute oraz Pure Mathematics Institute, Holendrów z National Research Institute for Mathematics and Computer Science i Kanadyjczyków, Amerykanów a także Brytyjczyków[10].

Natomiast Shamir i Tromer opublikowali pracę Factoring Large Numbers with the TWIRL Device, którą można uważać za kontynuację i rozwinięcie zeszłorocznego artykułu Bernsteina. Obie prace dotyczą układów scalonych dedykowanych do faktoryzacji (łamania) współcześnie używanych kluczy RSA. Według autorów urządzenie zdolne złamać klucz RSA o długości 512 bitów w ciągu 10 minut może kosztować ok. 10 tys. USD, a zdolne do złamania klucza 1024-bitowego w ciągu roku około 10 mln USD. Jest to nadal stosunkowo słaby rezultat za stosunkowo dużą cenę, ale niewątpliwie pokazuje, że łamanie uznawanych dziś za bezpieczne kluczy 1024-bitowych nie pozostaje poza zasięgiem ludzkich możliwości[11].

Dostępne układy realizujące RSA:

Szybkość działania algorytmu RSA jest uzależniona od długości modułu.

Dla 8-bitowego klucza publicznego publicznego komputera SPARC II wygląda następująco:

	512 bitów	**768 bitów**	**1024 bitów**
Szyfrowanie	0,03 s	0,05 s	0,08 s
Odszyfrowywanie	0,16 s	0,48 s	0,93 s
Podpisywanie	0,16 s	0,52 s	0,97 s
Sprawdzanie podpisu	0,02 s	0,07 s	0,08 s

Istnieje bardzo wiele sposobów ataku na algorytm RSA, przedstawię najważniejsze z nich:

10 www.hacking.pl, 2004-04-28
11 A. Shamir, E. Tromer, *Factoring Large Numbers with the TWIRL Device,* http://arch.ipsec.pl/twirl.pdf

Atak na algorytm RSA za pomocą wybranych szyfrogramów. Jest to dość skuteczny atak, mianowicie zakładamy, że Tomek przesyła zaszyfrowaną wiadomość c, którą przechwytuje Michał. Michał chce obliczyć $m = c^d$. Pobiera klucz publiczny Tomka e i oblicza $y = xc^e \bmod n$, gdzie $x = r^e \bmod n$ a r jest mniejsze niż n i jest przypadkową wielkością. Teraz Michał daje do podpisania Tomkowi y za pomocą jego klucza prywatnego, deszyfrując przy tym y. Tomek wysyła Michałowi $u = y^d \bmod n$, teraz Michał oblicza: tu $\bmod n = r^{-1} y^d \bmod n = r^{-1} x^d c^d \bmod n = c^d \bmod n = m$ i otrzymuje m. Jak widać na tym przykładzie, nigdy nie powinno się używać algorytmu RSA do podpisywania nieznanych dokumentów. Wprowadzenie standardu ISO 9796 udaremnia tego typu atak.

Atak na algorytm RSA przy wspólnym module. Może zdarzyć się tak, że każdy z użytkowników metody RSA otrzyma ten sam moduł n, ale za to inne wartości e i d. Jeżeli ta sama wiadomość zostanie zaszyfrowana dwoma innymi kluczami ale o tych samych modułach i te dwa klucze będą względnie pierwsze, to do otrzymania tekstu jawnego mogą nie być potrzebne żadne klucze deszyfrujące. Zakładając ze m jest tekstem jawnym a dwoma kluczami szyfrującymi są e_1 i e_2 oraz że n jest ich wspólnym modułem, to otrzymane szyfrogramy wyglądają następująco: $c_1 = m^{e1} \bmod n$ i $c_2 = m^{e2} \bmod n$. Znając n, e_1, e_2, c_1, c_2 można uzyskać odszyfrowaną wiadomość. Ponieważ e_1 i e_2 są wielkościami względnie pierwszymi, można użyć algorytmu Euklidesa do znalezienia takich liczb r i s, że $re_1 + se_2 = 1$. Zakładając, że e albo s są wartościami ujemnymi, można kolejny raz użyć algorytmu Euklidesa do obliczenia wartości: $(c_1^{-1})^{-r} * c_2^{s} = m \bmod n$. Metoda RSA mówi o tym, że nie należy stosować wspólnego modułu n, ale trzeba pamiętać o tym, że nie zawsze mamy NATO wpływ. Użytkowników metody RSA na całym świecie są miliony a liczb n możliwych do stosowania nie jest aż tak dużo.

Atak na algorytm RSA z małym modułem opiera się na wierze w to, że osoba wybierająca dwie liczby pierwsze z powodu na szybkość działania wybierze dość małe wartości tych liczb. Sposób łamania klucza tajnego przy wybraniu małych wartości p i q przedstawię podczas porównywania metod statycznych z dynamiczną metoda prof. Topolewskiego w przewidzianym do tego rozdziale. Łamanie klucza będzie odbywało się za pomocą programu napisanego przeze mnie.

Atak na algorytm RSA z małym wykładnikiem deszyfrującym został opracowany przez M. Wienera, który odzyskuje d. Możliwość odzyskania d zachodzi tylko wtedy, kiedy wartość n jest większa od d o 4 razy a e jest mniejsze niż n. Taki układ wielkości liczb zdarza się bardzo rzadko, ale nie można tego wykluczyć. Metoda RSA nakazuje wybieranie dużego d.

Atak przy użyciu kradzieży klucza publicznego. Założeniem metody RSA jest jak najlepsza ochrona klucza prywatnego. Niestety wszystko to jest kwestią implementacji. Klucze prywatne są często kodowane i przechowywane w miejscach niedostępnych dla nieupoważnionych osób. Kiedyś jednak przychodzi moment, w którym należy tych kluczy użyć, wtedy to bity poszczególnych kluczy są ładowane do pamięci operacyjnej komputera. W serwerach WWW hasła zabezpieczone przez protokół SSL znajdują się w pamięci przez cały czas. Wystarczy włamać się

na takowy serwer i dokonać zrzutu pamięci a potem odczytać informacje w niej przechowywaną. Dla dzisiejszych hackerów nie stanowi większego problemu włamanie się na serwer WWW, skoro potrafią włamywać się na serwery rządowe czy serwery NASA. Umieszczenie kluczy w wielu gigabajtach danych, tez nie daje rezultatu, ponieważ klucze mają swoją specyficzną strukturę ciągów losowych a kody programów oraz dane wykazują wyraźną strukturę bitową. Adi Shamir na zjeździe w Dreźnie w roku 1999 pokazał metodę poszukiwania klucza RSA na dysku twardym, która odbywała się w tym samym tempie, co odczyt danych z dysku. Na takie ataki najbardziej narażone są ogromnie długie klucze 1024 bitowe lub 2048 bitowe, ponieważ tyle bitów zupełnego chaosu potrafi zwrócić na siebie uwagę.

Atak przy użyciu komputera kwantowego i faktoryzacji dużych liczb pierwszych. Jeden z twórców metody RSA opublikował w roku 1977zagadkę, której rozwiązanie miało zająć 40 kwadrylionów lat, jak łatwo się domyśleć twórca tej zagadki pył pewien, że przed jego śmiercią nie dowie się od nikogo, co ona zawierała. Jednak już po 17 latach ku swojemu zdziwieniu zobaczył wydruk z rozszyfrowanym tekstem. W roku 2004 udało się przybliżyć teoretyczną wizję komputera kwantowego. Naukowca udało się zebrać bardzo dużą rozproszoną moc obliczeniową dzięki komputerom pracującym równolegle w sieci. Udało się złamać klucz 568 bitowy, jak wiemy dzisiaj ten rekord jest już nieaktualny. Przypuszczenia naukowe dowodzą, że skonstruowanie komputera kwantowego zmienią całkowicie pogląd na metody szyfrowania, ponieważ faktoryzacja bardzo dużych liczb pierwszych stosowanych w metodzie RSA, będzie trwała dosłownie minuty. Na dzień dzisiejszy są to tylko przypuszczenia, ale chyba nikt o zdrowych zmysłach nie wykluczy potrzeby zmian w dotychczasowych metodach szyfrowania.

1.2.2 ONE-TIME-PAD

Jak dotychczas tylko klucz jednorazowy o długości równej lub większej od szyfrowanej informacji został uznany za jedyną stu procentowo bezpieczną metodę szyfrowania. Metoda z kluczem jednorazowym opiera się na znanych nam metodach polialfabetycznych podstawień. W przeciwieństwie do poznanych już metod polialfabetyczny podstawień ta charakteryzuje się nieskończonym okresem. Metoda ta do złudzenia przypomina szyfr Cezara. Wykorzystuje tablice przyporządkowania, w której litera A odpowiada 0, B to jest 1 a C = 2 i tak dalej aż do Z = 25. Leżące w tej samej kolumnie znaki są dodawane jak w szyfrze Cezara. Jeśli Cuma przekroczy 26, to odejmujemy od niej 26, tak by można było otrzymać liczbę, którą tłumaczymy na literę itd.

Przykład[12]:

NIEZMIERNIEDLUGIIZUPELNIEP

+ TEKSTJAWNYJESTNIECOKROTSZY

= GMORFRENAGNHDNTQMBIZVZGADN

12 Reinhard Wobst, Kryptologia – Budowa i łamanie zabezpieczeń, Warszawa 2002, RM, s. 46

Odbiorca tak zaszyfrowanej wiadomości znający właściwy klucz musi odjąć go od szyfrogramu i w ten sposób otrzyma tekst jawny

GMORFRENAGNHDNTQMBIZVZGADN

- NIEZMIERNIEDLUGIIZUPELNIEP

= TEKSTJAWNYJESTNIECOKROTSZY

Wadą tej metody jest to, że klucz można użyć tylko jeden raz. W przypadku tego klucza nie istnieją reguły pozwalające na kryptoanalizę. Wadą jest też pojemność otrzymanego szyfrogramu. Gdybyśmy chcieli zaszyfrować kluczem jednorazowym np. nasz dysk twardy to potrzebowalibyśmy drugie tak samo dużego dysku na przechowanie klucza, który musi być w jakiś sposób przechowywany, co wiąże się z wielkim niebezpieczeństwem przechowywanej informacji.

W metodzie tej należy szczególnie pamiętać o tym, że klucze nie mogą się powtórzyć, ponieważ przy użyciu dość prostych testów statystycznych istnieje możliwość złamania tej metody.

Jeżeli P_1 – tekst jawny oraz P_2 – tekst jawny, to szyfrowanie musi podlegać następującemu wzorowi:

$C_1 = P_1 + S_1$

$C_2 = P_2 + S_2$

Gdzie S_1 i S_2 są dwoma różnymi kluczami całkowicie niezależnymi od siebie.

Gdyby S_1 i S_2 były takie same, a przeciwnik przechwyciłby przesyłane szyfrogramy, to wtedy korzystając z prostej zależności:

$C_2 - C_1 = P_2 - P_1$

- otrzymuje wyraźne cechy statystyczne, które pozwalają wykorzystać metodę zygzakową, do odczytania wszystkich wiadomości zaszyfrowanych tym kluczem.

Jak łatwo zauważyć klucz ten nie nadaje się do publicznego szyfrowania informacji. Jego bezpieczeństwo zachowane jest tylko wtedy, gdy zarządzanie kluczami jest rygorystycznie przestrzegane. Występuje tu również problem wielkości otrzymanych szyfrogramów. Nie można ta metodą szyfrować plików poddanych wcześniejszej kompresji. Metoda teoretycznie i praktycznie jest metodą nie do złamania przy zachowaniu się do instrukcji jej używania.