

Jak wzrost mocy obliczeniowej i postęp w badaniach matematycznych zagraża współczesnym rozwiązaniom bezpieczeństwa informatycznego

Piotr Kotlarz

*Piotr KOTLARZ, mgr, Uniwersytet Kazimierza Wielkiego, Instytutu Mechaniki Środowiska i Informatyki Stosowanej, ul. Chodkiewicza 30, 85-072 Bydgoszcz,
e-mail: piotrk@ukw.edu.pl*

Abstrakt

Współczesne rozwiązania z dziedziny bezpieczeństwa informatycznego bardzo często wykorzystują rozwiązania kryptografii asymetrycznej. Mo to oczywiście bardzo istotne podłoże praktyczne i funkcjonalne. Pamiętać należy jednak o tym, że na przykład szyfry asymetryczne, współcześnie bardzo popularny jest algorytm RSA [1][2]. Z samej swojej konstrukcji matematycznej są możliwe do złamania, oczywiście przy spełnionych pewnych warunkach. Bezpieczeństwo tak popularnych protokołów jak podpis elektroniczny czy elektroniczny znacznik czasu nie jest, więc bezpieczeństwem „danym raz na zawsze”. W pracy tej próbujemy pokazać, jakie zagrożenia dla obecnie stosowanych rozwiązań może nieść postęp w badaniach naukowych oraz wzrost mocy obliczeniowej komputerów. W większości przypadku mówiąc o bezpieczeństwie RSA, rozważa się problem faktoryzacji. Atakiem wywodzącym się niemal z definicji, jest atak polegający na próbie uzyskania na podstawie przechwyconego klucza publicznego, klucza prywatnego niezbędnego do odczytania zaszyfrowanej wiadomości. W pracy tej opisujemy możliwość przeprowadzenia ataku na algorytm RSA, z wykorzystaniem dwóch metod. Faktoryzacji oraz z wykorzystaniem funkcji Eulera. Przeprowadzona zostanie również dyskusja jak możliwości przeprowadzenia skutecznego ataku na RSA wpływają na bezpieczeństwo dokumentu elektronicznego.

1. Wstęp

Obecnie coraz powszechniej mówi o elektronicznym dokumencie. Kluczowym aspektem jest tu problem bezpieczeństwa. Współcześnie w celu zapewnienia tego bezpieczeństwa stosowane są trzy usługi kryptograficzne: szyfrowanie, podpis elektroniczny oraz elektroniczny znaczek czasu. Dwa najważniejsze warunki jakie powinien spełniać bezpieczny dokument elektroniczny to możliwość zweryfikowania autorstwa dokumentu i jego integralności oraz potwierdzenie czasu w jakim dokument istniał w danej postaci. Spełnienie wymogu tych dwóch postulatów obecnie zapewniają elektroniczny znaczek czasu oraz podpis elektroniczny. W tym artykule zamierzy przeprowadzić rozważania problemu bezpieczeństwa podpisu elektronicznego w wybranych aspektach. Obecnie rozwiązania komercyjne podpisu elektronicznego wykorzystują kryptografię asymetryczną. Biorąc pod uwagę że bezpieczeństwo kryptografii asymetrycznej (np. RSA [1][2]) jest w dużej mierze zależne od jakości zastosowanych kluczy oraz to, że z samej definicji szyfr asymetryczny jest możliwy do złamania. To znaczy bezpieczeństwo chronionej przez niego informacji, a dokładniej czas gwarantowanej ochrony, zależy od wielkości zastosowanych kluczy, choć nie jest to jedyny czynnik. Zasadnym wydaje się, postawienie pytanie o to na ile można zaufać współczesnym rozwiązaniom w zakresie podpisu elektronicznego opartego na kryptografii asymetrycznej. W pracy tej, oczywiście nie zamierzy polemizować z dogmatami współczesnej kryptografii, chcemy jedynie pokazać istotę problemu. Ponadto zależy nam na pokazaniu tego, co mogłyby się zdarzyć jeśli współczesna nauka rozwiązałaby pewne problemy. Istotnym elementem poza postępowaniem w badaniach naukowych z punktu widzenia bezpieczeństwa jest również wzrost mocy obliczeniowej współczesnych maszyn liczących. W kolejnych rozdziałach zamierzamy właśnie przeprowadzić rozważanie bezpieczeństwa podpisu elektronicznego pod kątem wyżej wspomnianych zagrożeń. Zanim jednak przejdziemy do rozważań, dokonamy niezbędnego wprowadzenia wybranych pojęć fundamentalnych.

2. Wprowadzenie – fenomen funkcji jednokierunkowej

Przez długi okres czasu, odkąd świat zaczął stosować kryptografię, a dokładniej rzecz biorąc szyfrowanie jako sposób na utajnienie informacji istniała największa nie niedogodność: problem dystrybucji klucza. Polegał on na konieczności ustalenia w bezpieczny sposób klucza przy użyciu którego szyfrowanie ma przebiegać. Problem był długo nierozwiązany, z pomocą jednak przyszła niezawodna matematyka. Whitfield Diffie pracownik amerykańskiej firmy Sun Microsystems oraz Martin Hellmann profesor Uniwersytetu Stanforda w Kalifornii. Wykorzystali funkcję jednokierunkową [3] do ustalenia, w bezpieczny sposób wartości klucza, w 1976 opublikowali na ten temat przełomową pracę [1]. Kryptograficzna funkcja jednokierunkowa leży u podstaw współczesnej kryptografii asymetrycznej [4].

Formalnie funkcje jednokierunkowe zapisuje się jako :

$$f_n : \sum^N \rightarrow \sum^{l(n)} \text{ gdzie } f = \{f_n | n \in N\}, \text{ a } l(n) \text{ to wielomian zmiennej } n. \quad (1)$$

Rodzinę funkcji f nazywamy wyliczalną wielomianowo, jeśli określenie wartości :

$$f_n(x), x \in \sum^n \quad (2)$$

może być wykonane dla pewnego $t \in N$ w czasie \dots

Mniej formalnie funkcje jednokierunkowe to takie, dla których dla dowolnej wartości x łatwo można policzyć $f(x)$, ale wyliczenie $f^{-1}(y)$ dla dowolnego y jest trudne. Funkcje jednokierunkowe stosowane w kryptografii to funkcje cechujące się tym, że pozostają jednokierunkowe pod warunkiem zachowania tajności pewnej informacji (klucza prywatnego). Można, więc stworzyć pewną

publiczną funkcję szyfrującą, którą każdy dysponujący tzw. kluczem publicznym może wyliczyć, jednak wykonanie operacji odwrotnej (deszyfracji) bez znajomości pewnych dodatkowych danych (klucza prywatnego) jest nie możliwe w rozsądnie długim czasie. Wynika z tego, że każdy może dokonać szyfrowania własnej wiadomości po poznaniu ogólnie dostępnych informacji. Nie zachodzi przy tym konieczność spotkania się z odbiorcą w celu uzgodnienia tajnego klucza. Zastosowanie funkcji jednokierunkowych pozwoliło całkowicie wyeliminować potrzebę dystrybucji klucza w sposób tajny. Powstała kryptografia asymetryczna posługująca się dwoma kluczami, które służą do wzajemnie odwrotnych operacji na tekście jawnym lub szyfrogramie. Z matematycznego punktu widzenia nie istnieją funkcje dla których jednokierunkowość została udowodniona. Znane są jednak takie, które „zachowują się zgodnie z oczekiwaniami”. Wymienić tu można funkcję wykładniczą dla której funkcją odwrotną jest logarytm dyskretny czy iloczyn wielkich liczb pierwszych gdzie dla wyznaczenia odwrotności konieczna jest faktoryzacja. Bezpieczeństwo systemów asymetrycznych opiera się na trudności wykonania pewnych obliczeń. Sztandarowym przykładem tego jest problem faktoryzacji liczb stanowiących iloczyn dwóch liczb pierwszych. Wyliczenie iloczynu dwóch liczb jest zadaniem stosunkowo prostym:

$$p = 88\,801, q = 16\,339,$$

$$N = p * q = 88\,801 * 16\,339 = 1\,450\,919\,539$$

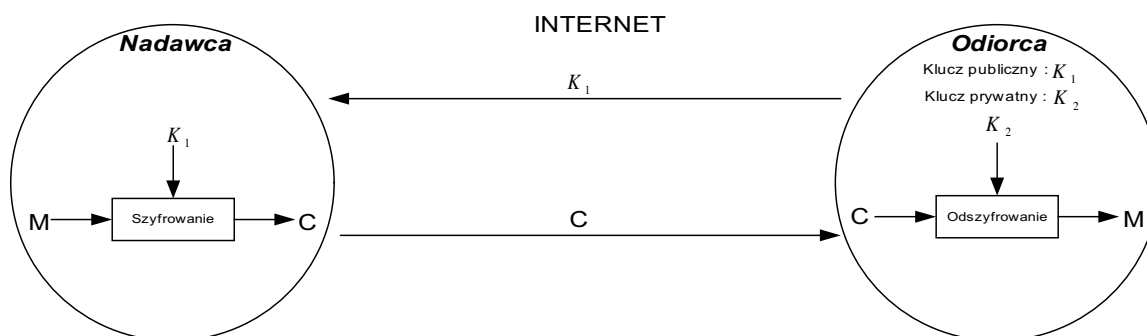
Natomiast znając tylko wartość N wyznaczenie p i q jest tym trudniejsze im N jest większe, ponieważ wyznaczenia p i q należało by wykonać poprzez faktoryzację. Natomiast znajomość choćby jednego z elementów p lub q powoduje zadanie wykonalnym.

$$q = 16\,339, N = 1\,450\,919\,539$$

$$p = N / q = 88\,801$$

3. Wprowadzenie – kryptografia asymetryczna

Wprowadzenie kryptografii asymetrycznej spowodowało pojawienie się nowych możliwości związanych z bezpiecznym obiegiem informacji elektronicznej. Można zaryzykować stwierdzenie, że gdyby nie kryptografia asymetryczna nie mówilibyśmy dzisiaj o podpisie elektronicznym, czy elektronicznym znaczniku czasu. Najpowszechniej wykorzystywanym szyfrującym algorytmem asymetrycznym jest RSA. W 1978 roku opublikowano pracę, która przedstawiała kryptosystem z kluczem publicznym o nazwie RSA [2]. Bezpieczeństwo tego algorytmu opiera się na trudności rozkładu dużych liczb na czynniki pierwsze. Kryptografia asymetryczna nie spowodowała wyparcia metod symetrycznych, obecnie uzupełniają się one w wielu protokołach kryptograficznych. Poniższy rysunek przedstawia ogólną koncepcję szyfrowania asymetrycznego [4].



Rys. 1. Szyfrowanie asymetryczne

Jeśli wykorzystujemy algorytm asymetryczny w celu zapewnienia poufności przesyłanych lub przechowywanych danych do szyfrowania wykorzystywany jest klucz publiczny (K1). W celu odszyfrowania wiadomości zaszyfrowanej niezbędnym jest posiadanie klucza prywatnego (K2), stanowiącego parę z użytym w procesie szyfrowania kluczem prywatnym. Algorytm RSA wpisuje się w wyżej przedstawiony schemat.

Działanie algorytmu RSA :

Generowanie paru kluczy :

liczby pierwsze wybierane losowo : p, q oraz d spełniające warunek $NWD(d, (p-1)(q-1))=1$

$$N = p * q$$

$$ed \equiv 1 \pmod{(p-1)(q-1)} \tag{3}$$

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

klucz publiczny (K1): N, d

klucz prywatny (K2): e

Funkcja szyfrująca : $F_{k_1}(M) = C = M^d \pmod N$

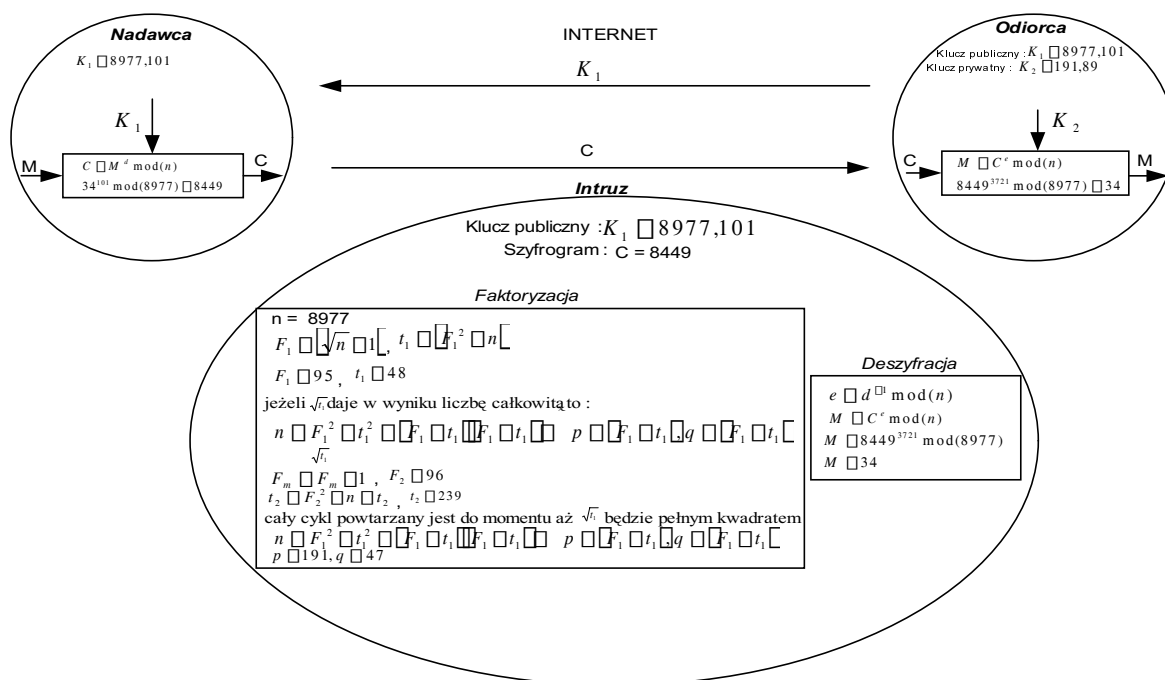
Funkcja deszyfrująca : $F_{k_2}(C) = M = C^e \pmod N$

Powyżej przedstawiona została zasada działania szyfru RSA. Wyznaczenia wartości e, stanowiącej klucz prywatny dokonuje się z wykorzystaniem rozszerzonego algorytmu Euklidesa [4]. Warto wspomnieć o pewnym mankamencie towarzyszącym stosowaniu kryptografii asymetrycznej. W przypadku opisanego tu algorytmu RSA, aby zapewnić wystarczający poziom bezpieczeństwa należy stosować odpowiednio duże liczby, co pociąga za sobą duży spadek wydajności szyfrowania w stosunku do metod symetrycznych.

Na tym zakończymy opis samego algorytmu RSA, ponieważ to, co zostało przytoczone powyżej wystarczy na potrzeby przeprowadzenia dalszych rozważań.

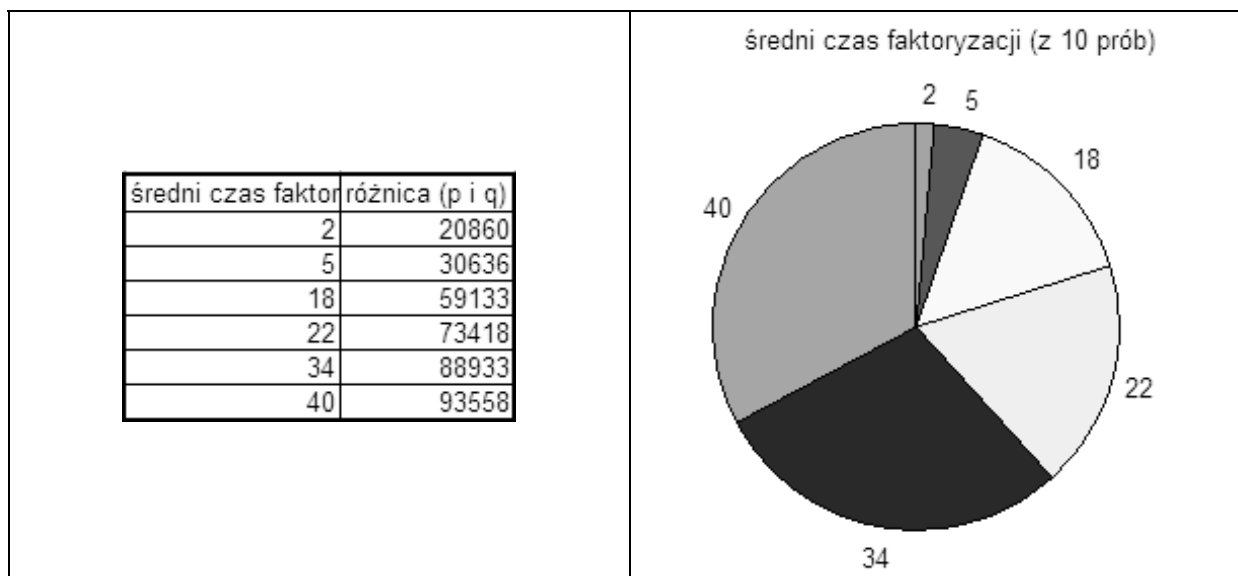
4. Problem bezpieczeństwa kryptosystemów asymetrycznych

Jak to zostało wspomniane wcześniej, o bezpieczeństwie rozwiązań kryptografii asymetrycznej [5][6][7][8] decyduje brak skutecznych metod matematycznych, umożliwiających szybkie odwracanie funkcji szyfrujących, jednokierunkowych z zapadką bez znajomości owej zapadki. W pracy tej na potrzeby prowadzonych rozważań wybrany został algorytm RSA. Jego bezpieczeństwo opiera się na problemie rozkładu dużych liczb na czynniki pierwsze. Współczesna matematyka nie zna wydajnych metod faktoryzacji. Obecnie znane są na tyle czasochłonne, że zastosowanie odpowiednio dużych liczb jako klucz publiczny zapewnia wymagany poziom bezpieczeństwa. Dyskusję ataku z wykorzystaniem faktoryzacji [8] przeprowadzimy z wykorzystaniem metody Fermata[7][4]. Poniżej prezentujemy przykład wykorzystania faktoryzacji dla potrzeb ustalenia wartości klucza prywatnego na podstawie znajomości klucza publicznego.



Rys.2. Faktoryzacja

Powyższy rysunek przedstawia sytuację, gdy podsłuchujący kanał transmisji intruz wejdzie w posiadanie szyfrogramu jak i klucza publicznego odbiorcy. Korzystając z faktoryzacji opartej na metodzie Fermeta uzyskuje klucz prywatny i odszyfrowuje przechwycony szyfrogram. Istotnym elementem z punktu widzenia powodzenia kryptoanalizy jest czas potrzebny na przeprowadzenia ataku. Jeśli zależy nam na uzyskaniu klucza prywatnego na podstawie klucza publicznego, to w przypadku RSA czas ten zależy jest w dużej mierze od wydajności zastosowanej metody faktoryzacji. Posługując się wyżej przedstawionym algorytmem czas ten nie tyle zależy od wielkości liczby (N), której czynników szukamy, co od odległości na osi liczbowej szukanych składowych iloczyn N=p*q. Poniżej przedstawiony jest wykres, który pokazuje jak zależy czas faktoryzacji od wielkości różnicy między wartościami p i q. Wartość różnicy na wykresie odwzorowana jest przez rozmiar wycinka koła wykresu.



Rys. 3. Czas faktoryzacji – zależność od jakości klucza

Przykład ten pokazuje jak ważnym elementem z punktu widzenia bezpieczeństwa jest właściwy dobór kluczy. Z całą pewnością jedynym parametrem nie może być wielkość klucza (liczby N). Znaczenie ma również różnica pomiędzy wartościami p i q . Stosując metodę faktoryzacji opartą na małym twierdzeniu Fermata, uzyskujemy tym lepszy wynik czasowy im ta różnica jest mniejsza. W większości prac traktujących o bezpieczeństwie RSA i algorytmach jemu podobnych metodą ataku, w których celem jest uzyskanie klucza prywatnego najczęściej jest opisywana metoda faktoryzacji. W następnym rozdziale prezentujemy podejście do problemu uzyskania klucza prywatnego na podstawie klucza publicznego. Proponujemy wykorzystanie do ataku funkcji Eulera, a w dalszej części zrównoleglenie procesu wyznaczania wartości tej funkcji. Jak pokażemy w dalszej części algorytm wyznaczania wartości funkcji Eulera jest bardzo wygodny do zrównoleglenia, co w znacznym stopniu poprawia wydajność czasową ataku.

5. Wyznaczenie odwrotności modulo z wykorzystaniem funkcji Eulera, a bezpieczeństwo RSA.

Problem bezpieczeństwa obecnie stosowanych algorytmów szyfrujących z rodziny szyfrów asymetrycznych opiera się na trudności rozkładu dużych liczb na czynniki pierwsze. O czym zostało wspomniane powyżej. Można go jednak również zdefiniować jako problem szukania odwrotności modulo n . Posługując się przykładem algorytmu RSA, można to przedstawić następująco. Intruz po przechwyceniu klucza publicznego (N , d) oraz szyfrogramy C , może dążyć do uzyskania klucza prywatnego (e) na podstawie klucza publicznego wyznaczając odwrotności :

$$e = d^{-1} \bmod(N) \quad (4)$$

Jednym ze sposobów wyznaczenia wartości e jest wykorzystanie funkcji Eulera [9][10]. Funkcja Eulera $\varphi(N)$ jest definiowana jako liczba elementów w zredukowanym zbiorze reszt. Zredukowany zbiór reszt to podzbiór zbioru pełnego reszt, którego elementy są względnie pierwsze z N . Dla przykładu jeśli jako N przyjmiemy 8 to pełen zbiór reszt modulo zawiera elementy $\{1,2,3,4,5,6,7\}$. Natomiast zredukowany zbiór reszt to $\{1,3,5,7\}$, ponieważ tylko te elementy nie posiadają wspólnego czynnika z N czyli 8. Wracając, więc do definicji funkcji Eulera jej wartość dla $N=8$ wynosi 4. Inaczej zdefiniować funkcję Eulera dla N można jako ilość wszystkich liczb, całkowitych, dodatnich względnie pierwszych z N z przedziału od 1 do $N-1$. Nadmienić należy jeszcze, że :

$$\text{jeśli } N \text{ jest liczbą pierwszą to } \varphi(N) = N-1, \quad (5)$$

$$\text{jeśli } N=p*q \text{ to } \varphi(N)=(p-1)(q-1). \quad (6)$$

Można postawić pytanie co daje nam to z punktu widzenia uzyskania na podstawie klucza publicznego wartości klucza prywatnego. Wspomnieliśmy wcześniej że:

$$e = d^{-1} \bmod(N) \quad (7)$$

Więc w celu wyznaczenia klucza prywatnego e oraz chcąc wykorzystać do tego funkcję Eulera, klucz prywatny otrzymamy wykonując obliczenie

$$e = d^{\varphi(N)-1} \bmod(\varphi(N)) \quad (8)$$

W powyższym równaniu znane są wartości: N , oraz d które stanowią składowe klucza publicznego. Wyznaczenia wymaga wartość $\varphi(N)$. W przypadku klucza publicznego RSA - N na pewno

nie będzie liczbą pierwszą ponieważ $N=p*q$, czyli nie możemy skorzystać z własności (5). Z całą pewnością intruz nie będzie dysponował również wartościami p i q , czyli nie wchodzi w grę skorzystanie z własności (6). Nie pozostaje więc nic innego jako wyznaczenie wartości $\varphi(N)$ korzystając z metody numerycznej. Poniżej przedstawiony jest przykład realizacji algorytmu uzyskania klucza prywatnego z wykorzystaniem wartości funkcji Eulera. Wyjaśnienia wymaga jeszcze fakt, że poniższy przykład jest pokazany jedynie dla celów poglądowych i wykorzystanie wielkości kluczy są zdecydowanie dużo mniejsze niż te stosowane w praktyce.

Klucz publiczny : $\varphi(N)=\varphi(4242331)= 4232592$,
 $N=4242331$ $\varphi(\varphi(N))= 995328$,
 $d= 6967$ $e = d^{\varphi(\varphi(N))-1} \bmod(\varphi(N)) = 6967^{995328-1} \bmod(4232592) = 3170647$,
 Szyfrogram: $M = C^e \bmod N = 3706005^{3170647} \bmod(4242331) = 342$
 $C=3706005$

W powyższym przykładzie liczona jest dwa razy wartość funkcji Eulera. Nie znając wartości p i q wyznaczenie wartości klucza prywatnego na podstawie znajomości publicznego w rezultacie sprowadzi się do ataku brutalnego. Dla sprawdzenia jaka jest liczba elementów w zredukowanym zbiorze reszt dla danego N wymaga się wykonania $N-1$ iteracji. Czyli złożoność tego procesu rośnie wraz ze wzrostem wartości N czyli wielkości stosowanych kluczy. Całą operację można trochę przyspieszyć skracając czas wyznaczenia $\varphi(\varphi(N))$ poprzez podział przez 2 lub przez 4. Czy liczymy najpierw $\varphi(\varphi(N)/4)$ a otrzymany wynik mnożymy przez 4. Otrzymana wartość będzie taka sama jak gdy byśmy liczyli $\varphi(\varphi(N))$.

Mimo to, czas uzyskania wartości e i tak w znacznym stopniu determinuje czas potrzebny na wyznaczenie $\varphi(N)$. Można jednak zastanowić się jakie rezultaty można by uzyskać w sytuacji jeśli zastosuje się operację zrównoleglenia procesu wyznaczania wartości funkcji Eulera.

5.1 Zrównoleglenie procesu wyznaczania wartości funkcji Eulera.

Sposób w jaki wyznaczana jest wartość funkcji Eulera umożliwia w prosty sposób zrównoleglenie procesu wyznaczania jej wartości. Jeśli założymy że dysponujemy dwoma komputerami PC (K_1, K_2) mogą one pracować równolegle nad wyznaczeniem $\varphi(N)$. Pierwszy będzie sprawdzał ilość wszystkich całkowitych, dodatnich liczb względnie pierwszych z N z przedziału od 1 do $\text{int}(N/2)$. Komputer K_2 natomiast przeszuka zakres od $\text{int}(N/2)+1$ do $N-1$. W rezultacie po zsumowaniu wyników otrzymanych przez K_1 i K_2 otrzymamy wartość $\varphi(N)$, w czasie w przybliżeniu o połowę krótszym niż gdyby to samo robić na pojedynczym komputerze.

Pokazuje to poniższy przykład:

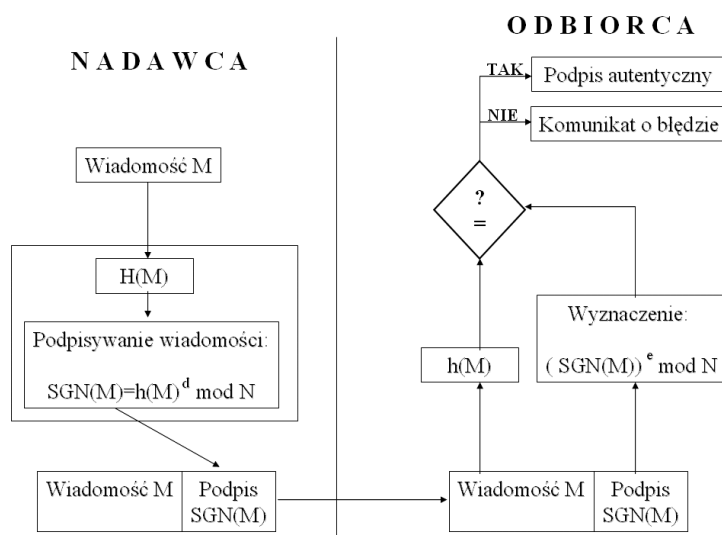
$N= 107399$
 $\text{int}(N/2) = 53699$
 $\text{int}(N/2)+1 = 53700$
 $K_1 \rightarrow \varphi_1(N) = 53340$
 $K_2 \rightarrow \varphi_2(N) = 53340$
 $\varphi_1(N)+ \varphi_2(N) = \varphi(N) = 106680$

Oczywiście nic nie stało by na przeszkodzie aby cały proces podzielić na większą liczbę równoległych pracujących komputerów. Należało by również zastanowić się na ile fakt przy podziale przez dwa wartości N (na : $\text{int}(N/2)$ i $\text{int}(N/2)+1$) uzyskiwać będziemy wynik cząstkowe takie

same dla różnych przypadków N , tak ja ma to miejsce w powyższym przykładzie. Jest to problem, który wymaga w ewentualnie dalszych badań nie tyle eksperymentalnych co próby uogólnienia problemu.

6. Protokół podpisu elektronicznego – aspekt bezpieczeństwa

Odwołujemy się do wersji podpisu elektronicznego realizowanego przez tak zwany protokół samowymuszający [11][12]. Strony ustalają klucze publiczne i prywatne, a następnie ogłaszają swoje klucze publiczne. W rozwiązaniach podpisów cyfrowych stosuje się jednokierunkowe funkcje skrótu. Dzięki czemu nie podpisuje się całego dokumentu lecz jego skrót, który ma charakter jednokierunkowy. Oszczędza się czas, ponieważ szyfrowany jest skrót dokumentu, a nie cały często liczony w megabajtach dokument. Oczywiście jest, że strony realizujące protokół wymienić muszą: informacje na temat stosowanej funkcji skrótu, klucze publiczne oraz ustalić algorytm asymetryczny. Poniższy schemat pokazuje przebieg takiego protokołu podpisu elektronicznego wykorzystujący funkcję skrótu.



Rys. 4. Podpis elektroniczny

Tak więc na protokół składają się z następujące kroki:

Nadawca oblicza wartość SGN (powstaje w wyniku zaszyfrowania skrótu wiadomości z wykorzystaniem klucza prywatnego nadawcy).

Nadawca przesyła do odbiorcy dokument M i podpis SGN .

Odbiorca oblicza skrót dokumentu M za pomocą funkcji skrótu $h=H(M)$, deszyfruje podpis SGN przy użyciu klucza publicznego nadawcy, porównuje otrzymane wartości. Jeżeli obie otrzymane wartości są jednakowe stwierdza, że podpis jest autentyczny. Stosowanie funkcji skrótu skraca czas potrzeby do realizacji podpisu cyfrowego. Zmniejsza się również wielkość zaangażowanych zasobów komputera.

7. Integralność przesyłanych danych

Integralność zawartości wiadomości pozwala na zweryfikowanie czy przesłana wiadomość nie uległa zmianie podczas transmisji. W przypadku podpisu elektronicznego zapewnienie atrybutu integralności realizowane jest z wykorzystaniem funkcji skrótu. Miejsce funkcji skrótu w protokole

pokazane jest na rys.2. O jednokierunkowości funkcji pisaliśmy już w rozdziale 2. Teraz sprecyzujemy wybrane właściwości funkcji skrótu.

Funkcja skrótu [13][4] generuje ciąg o stałej długości (np. 128 bitów) dla wiadomości o dowolnej długości, czyli:

$$h: \Sigma^* \rightarrow \Sigma^n, \quad (9)$$

funkcja skrótu (h) koliduje jeżeli

$$h(m_1) = h(m_2) \quad (10)$$

dla dwóch różnych wiadomości.

Funkcja skrótu jest jednokierunkowa jeżeli znalezienie przeciwobrazu (m) na podstawie skrótu h(m) jest obliczeniowo trudne. Funkcja skrótu jest bezkolizyjna jeżeli dla danej wiadomości m znalezienie innej m' spełniającej warunek :

$$h(m) = h(m') \quad (11)$$

jest trudne obliczeniowo.

Funkcja skrótu jest bezkolizyjna jeżeli znalezienie jakiegokolwiek pary wiadomości m_1, m_2 której skrót koliduje, jest obliczeniowo trudne. Biorąc pod uwagę to co zostało opisane powyżej, można podać intuicyjną definicję funkcji skrótu. Można ją określić jako funkcję, która na podstawie wiadomości m generuje wartości jednoznacznie charakteryzującą m. Konstrukcja matematyczna funkcji skrótu powinna gwarantować, to że dla dwóch różnych wiadomości funkcja nie może wyznaczyć takiej samej wartości. Ponadto funkcja skrótu powinna generować wartość tej samej długości dla dowolnej długości wiadomości m. Przechodząc do rozważań na temat tego, jaki wpływ na bezpieczeństwo podpisu elektronicznego mogłaby mieć możliwość sfałszowania funkcji skrótu. Biorąc pod uwagę schemat protokołu podpisu elektronicznego, można dostrzec, że weryfikacja podpisu opiera się między innymi na stwierdzeniu, że nienaruszona została integralność wiadomości. Strona weryfikująca uzyskuje to poprzez obliczenie wartości funkcji skrótu (tej samej, jaka użyta została przy składaniu podpisu) i porównanie jej z odszyfrowaną przy pomocy klucza publicznego osoby podpisującej. Pokazaliśmy to na poniższym przykładzie, decydując się na użycie nierzeczywistych długości kluczy RSA oraz funkcji skrótu z powodu chęci zapewnienia pogłębienia poniższego przykładu.

m- podpisywana wiadomość, h(m)-skrót wiadomości, e-klucz prywatny podpisującego, d,N-klucz publiczny podpisującego, SGN = C- zaszyfrowany skrót wiadomości, podpis.

Generowanie podpisu

$m = 4264345435367547$

$h(m) = 34$

$e = 1229, d=12361169, N=18425597$

$SGN = C = h(m)^e \bmod N = 10096866$

Weryfikacja podpisu

$d=12361169, N=18425597$

$m = 4264345435367547, SGN= 10096866$

$h(M)34$

$h = C^d \bmod N$

$h = 10096866^{12361169} \bmod 18425597 = 34$

$h = h(m) \rightarrow$ pomyślna weryfikacja

Intruz w momencie przechwycenia: klucza publicznego (d, N), wiadomości (m), podpisu (SGN), mógłby próbować uzyskać klucz prywatny (e) wykorzystując jedną z metod, którą opisaliśmy w już powyżej. Zakładając, że udało mu się uzyskać klucz prywatny, można się zastanowić, co on w rzeczywistości uzyska? Zysk jest znaczący, ponieważ dysponując kluczem prywatnym intruz będzie w stanie wygenerować weryfikowalny podpis dla przechwyconej wiadomości, po dokonaniu zmian w treści dokumentu. Sytuacja taka jest możliwa, ponieważ po zmianie treści wiadomości m , intruz obliczy nową wartość funkcji skrótu dla nowej wiadomości (m') a dysponując kluczem prywatnym będzie w stanie wygenerować nowy podpis. Niczego nieświadoma osoba weryfikująca podpis, nie będzie w stanie stwierdzić, że podpis został sfałszowany. Pokazuje to poniższy przykład:

Falszowanie podpisu

$$d=12361169, N=18425597$$

$$m = 4264345435367547, SGN= 10096866$$

w wyniku ataku intruz uzyskuje :

$$e = 1229$$

falszowanie

$$m' = 4264345435363244$$

$$h(m')=567$$

$$SGN = C = h(m')^e \bmod N$$

$$SGN = C = 567^{1229} \bmod 18425597 = 8256758$$

Weryfikacja podpisu

$$d=12361169, N=18425597$$

$$m' = 4264345435363244, SGN= 8256758$$

$$h(m')=567$$

$$h = C^d \bmod N$$

$$h = 8256758^{12361169} \bmod 18425597 = 567$$

$h = h(m) \rightarrow$ pomyślna weryfikacja

Jak zostało to pokazane na przykładzie strona weryfikująca podpis nie będzie w stanie stwierdzić zaistniałego faktu fałszerstwa bez porównania swojej wersji wiadomości, którą otrzymała z dokumentem oryginalnym pozostającym w posiadaniu prawowitego nadawcy. W tym celu oczywiście niezbędnym było by nawiązanie kontaktu z autorem wiadomości. Co nie zawsze jest wygodne i pożądane w praktyce. Powyższy przykład pokazuje jaki wpływ na bezpieczeństwo podpisu elektronicznego z punktu widzenia, wymogu integralności może mieć zastosowanie bezpiecznego algorytmu szyfrującego ale nie zastosowanie odpowiednich kluczy. Współczesna kryptografia wymaga stosowania algorytmów szyfrujących, które muszą być jawne (opublikowane), natomiast bezpieczeństwo ma gwarantować odpowiedni dobór stosowanych kluczy. Ma to pewien praktyczny aspekt, ponieważ ujawnienie zasady działania nowego algorytmu szyfrującego pozwala poddać nowy szyfr szerokiej dyskusji.

8. Podsumowanie

Praca ta to pewne rozważania na temat „potencjalnych granic” bezpieczeństwa podpisu elektronicznego oraz kryptografii asymetrycznej. Po lekturze tego artykułu można by dojść do wniosku, że wszystko sprowadza się do kwestii zastosowania odpowiednio dużych kluczy. Przy takim podejściu zarówno metody faktoryzacji, jak i liczenie odwrotności modulo nie będzie realizowalne w rozsądnie długim czasie. Co może doprowadzić do sytuacji, że gdy uda się złamać szyfrogram to chroniona informacja straci jakąkolwiek wartość. Jednak problem bezpieczeństwa nie jest tu tak prosty. Jedną z wad kryptografii asymetrycznej jest problem dużej złożoności obliczeniowej. Może się więc okazać, że algorytmy które działają zadawalająco wydajnie dla kluczy o długościach np. 1024 bity przy nieco większych zaczynają powodować istotne opóźnienia. Więc

może się okazać, że zwiększanie długości klucza w nieskończoność nie jest receptą na bezwarunkowe bezpieczeństwo.

Bibliography

- [1] W. Diffe and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT-22:644-654, November 1976.
- [2] Ronald Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21:120-126, 1978.
- [3] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proc. 22nd ACM Symposium on Theory of Computing*, pages 387{394, Baltimore, Maryland, 1990. "Association for Computing Machinery (ACM)".
- [4] Josef Pieprzyk, Thomas Hardjono, Jennifer Seberry "Fundamentals of Computer Security", Springer – Verlag Berlin 2003
- [5] Gustavus J. Simmons and Michael J. Norris. Preliminary comments on the MIT public-key cryptosystem. *Cryptologia*, 1(4):406-414, October 1977.
- [6] D. Boneh. "Twenty years of attacks on the RSA cryptosystems", 1997
- [7] Daniel M. Gordon, Center for Communications Research, A survey of fast exponentiation methods, December 30, 1997
- [8] Stefania Cavallar, Walter Lioen, Herman te Riele, Factorizing of a 512 bit RSA Modules. MAS-R0007 February 29, 2000
- [9] Dario Catalano , Rosario Gennaro and Shai Halevi, IBM T.J.Watson Research Center, Computing inverses over a shared secret modulus.
- [10] Yair Frankel, Yvo G. Desmedt, Parallel reliable threshold multisignature, Tech. Report: Department of E.E. and C.S. University of Wisconsin-Milwaukee, WI 53201, TR-92-04-02
- [11] B. Pfitzmann. *Digital Signature Schemes*. LNCS, 1100, Springer, New York, 1996.
- [12] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, 1996.
- [13] D.R. Stinson. *Cryptography: Theory and Practice*. CRC Press, 1995.
- [14] Gary L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Science*, 13(3):300-317, 1976.
- [15] Paul C. Kocher, Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. Cryptography Research, Inc.
- [16] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997.