będąca właścicielem opisywanej dalej w tym rozdziale innej metody kryptograficznej) ustanowiła nagrodę w wysokości 10 000 dolarów za sforsowanie algorytmu DES. Wysiłki powołanej do tego celu grupy i wolne moce kilkunastu tysięcy komputerów użytkowników Internetu zaowocowały złamaniem kodu w niecałe trzy miesiące. Dziesięć lat później Frontier Foundation polepszyła ten wynik, łamiąc zakodowany w DES tekst w dziewięć dni; wkrótce pojawiły się też oferty sprzętowych kodołamaczy (np. projekt COPACOBANA), czyli specjalizowanych komputerów zoptymalizowanych w celu łamania kodów za pomocą różnych metod (głównie rozmaitych metod siłowych).

Warto wiedzieć, że istnieją kodery sprzętowe (oparte na układach scalonych) DES, które są średnio 1000 razy szybsze od wersji software'owych.

Kodowanie asymetryczne

Kodowanie asymetryczne eliminuje wadę związaną z kłopotliwą logistyką i pilnowaniem kluczy napotkaną w algorytmach symetrycznych. Rozwiązuje efektywnie *problem transmisji klucza* w świecie, gdzie ważne jest, aby wiadomość dotarła do odbiorcy w ułamku sekundy, bez obarczania go dodatkową troską o wiarygodność otrzymanego klucza K.

Zakłada się, że odbiorca pragnący otrzymać wiadomość może wysłać nadawcy klucz szyfrujący w sposób jawny. Nadawca zaszyfrowuje nim swoją wiadomość i wysyła ją do odbiorcy przez publiczne łącza transmisyjne — podczas transmisji jest zatem zachowana poufność. Odbiorca odszyfrowuje ją *innym* kluczem, będącym wyłącznie w jego posiadaniu. Przechwycenie klucza szyfrującego, bez drugiego klucza — odszyfrowującego, nic nie daje!

Pierwszy klucz, szyfrujący, nazywa się *kluczem publicznym*, a drugi, odszyfrowujący — *kluczem prywatnym*. Klucza prywatnego należy strzec jak oka w głowie, klucz publiczny można swobodnie rozdawać tym, z którymi chcemy się komunikować. Każdy, kto otrzymał klucz publiczny, będzie mógł wysłać do nas zaszyfrowaną nim wiadomość, którą odczytamy tylko my, jako jedyni posiadacze klucza prywatnego.

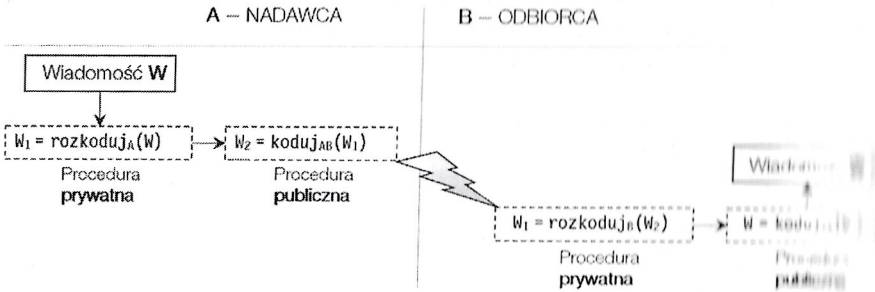
Metodę kodowania z kluczem publicznym, która całkowicie wyeliminowała dystrybucję klucza, wynaleziono w 1976 r. Jej wynalazcami byli Whitfield Diffie i Martin Hellman, jednak jej pierwsza praktyczna realizacja została opracowana przez Rona Rivesta, Adiego Shamira i Leonarda Adlemana; jest znana jako tzw. kryptosystem RSA. Metoda RSA gwarantuje bardzo duży stopień bezpieczeństwa przesyłanej informacji. Ponieważ została uznana przez matematyków za niemożliwą do złamania, momentalnie stała się obiektem zainteresowania komputerowych maniaków na całym świecie, którzy za punkt honoru przyjęli jej złamanie[4].

---

[4] Prawa do algorytmu RSA posiada *RSA SECURITY LLC* (obecnie część EMC Corporation), udzielająca płatnej licencji na jego używanie w programach innych producentów (skorzystano z niej w takich produktach jak Internet Explorer i Netscape Navigator). RSA

buję wyjaśnić ideę kryptografii z kluczem publicznym.

System kryptograficzny z kluczem publicznym został przedstawiony na rys. 18.2. Składa się z trzech procedur: dwóch *prywatnych* (rozkoduj$_A$ i rozkoduj$_B$) i jednej *publicznej* (koduj$_{AB}$).



RYSUNEK 18.2. System kodujący z kluczem publicznym

Nadawca A, kiedy chce wysłać do odbiorcy B wiadomość W, w pierwszym momencie robi rzecz dość dziwną: zamiast zwyczajnie zakodować ją i wysłać poprzez kanał transmisyjny do odbiorcy, dodatkowo używa funkcji rozkoduj$_A$ na niezaszyfrowanej wiadomości! Czynność ta, na pierwszy rzut oka dość absurdalna, ma swoje uzasadnienie praktyczne: na wiadomości W jest odciskany niepowtarzalny *podpis cyfrowy* nadawcy A, co w wielu systemach (np. bankowych) ma znaczenie wręcz strategiczne! Następnie podpisana wiadomość (W_1) jest szyfrowana przy powszechnie znaną procedurę szyfrującą koduj$_{AB}$ i dopiero po tym wysyłana do B.

Odbiorca B otrzymuje zakodowaną sekwencję kodową i używa swojej prywatnej funkcji rozkoduj$_B$, która jest tak skonstruowana, że na wyjściu odtworzy podpisaną wiadomość W_1. Podobnie specjalna musi być funkcja koduj$_{AB}$, która z cyfrowo podpisanej wiadomości W_1 powinna odtworzyć oryginalny komunikat W.

Wymogi bezpieczeństwa zakładają praktyczną *niemożność odtworzenia* tajnych procedur rozkodowujących na podstawie jawnych procedur kodujących.

Idea jest zatem urzekająca, wszakże pod warunkiem dysponowania trzema tajemniczymi procedurami, które na dodatek są powiązane ze sobą dość ostrymi wymaganiami! Dopiero rok od pojawienia się idei systemu z kluczem publicznym powstała pierwsza (i jak do tej pory najlepsza) realizacja praktyczna: system kryptograficzny o nazwie *RSA*. System ten zakłada, że odbiorca B wybiera

---

wchodzi w skład wielu standardów i protokołów sieciowych (np. S/MIME, SSL) oraz programów (PGP, warianty bezpiecznej poczty). Duże firmy często stosują mechanizm o nazwie RSA SecureID pozwalający na bezpieczny, zdalny dostęp do sieci firmowych, np. z laptopów osób pracujących poza biurem.

losowo trzy bardzo duże liczby pierwsze: *S*, *N1* i *N2* (zwykle stucyfrowe) i udostępnia publicznie tylko ich iloczyn[5] $N = N1 \cdot N2$ oraz pewną liczbę *P*, spełniającą warunek:

$$P \cdot S \bmod (N1-1) \cdot (N2-1) = 1.$$

Zostało udowodnione, że dla każdego ciągu kodowego *M* (tekst zostaje zamieniony na odpowiadający mu ciąg liczbowy o pewnej skończonej długości) spełniona jest równość: $MPS \bmod N = M$.

Kodowanie sprowadzi się zatem do obliczenia równości: {ciąg kodowy} = *koduj*(*M*) = $MP \bmod N$.

Dekodowanie jest równoważne obliczeniu: M = dekoduj({ciąg kodowy}) = {ciąg kodowy}$^S \bmod N$.

Pomimo pozornej trudności wykonania operacji na bardzo dużych liczbach okazuje się, że własności funkcji modulo powodują, iż zarówno ciąg kodowy, jak i jego zaszyfrowana postać należą do tego samego zakresu liczb. Złamanie systemu RSA byłoby możliwe, gdybyśmy umieli na podstawie znanych wartości *N* i *P* odtworzyć utajnione *S*, potrzebne do rozkodowania wiadomości! Nie znaleziono do tej pory algorytmu, który mógłby wykonać to zadanie w rozsądnym czasie.

Kodowanie Base64

Interesującym przykładem kodowania jest wbudowany w systemy operacyjne system kodowania o nazwie *base64*, w którym dane binarne są konwertowane na znaki alfabetu ASCII. Używa się do tego celu **jawnej** tabeli kodów pokazanej w tabelce 18.2.

Ponieważ tabela kodów jest znana, użycie base64 nie zapewnia poufności!

Warto wiedzieć, że base64 jest wbudowany w Linuksa i możesz przetestować następujące komendy w linii poleceń:

```
$ echo "ala ma kota" | base64
YWxhIGlhIGtvdGEK
$ echo "YWxhIGlhIGtvdGEK" | base64 --decode
ala ma kota
$ base64 plikwejsciowy.test > plikwyjsciowy.test
```

Trzecie wywołanie koduje podany w parametrze plik wejściowy na pewien plik wyjściowy. Wynik możesz zdekodować poleceniem `base64 -d plikwyjsciowy.test > plikodkodowany.test`.

Dlaczego tabela kodów jest oparta na podstawowym zestawie ASCII? Chodzi o ułatwienie bezstratnego przesyłania danych przez sieci komputerowe, np. Internet,

[5] Ponieważ [illegible] znalezienia metody rozkładu na czynniki pierwsze [illegible]