Kryptografia — systemy z kluczem publicznym

Mieliśmy więc . . .

. . . system kryptograficzny

$$\mathcal{P} \stackrel{f}{\Longrightarrow} \mathcal{C} \stackrel{f^{-1}}{\Longrightarrow} \mathcal{P},$$

gdzie funkcja f składała się z dwóch elementów:

- Algorytm (wzór) – np. $\mathcal{C} = f(\mathcal{P}) \equiv \mathcal{P} + b \bmod N$
- Parametry K_E (*enciphering key*) – tutaj: b oraz N.

W dotychczasowej praktyce algorytm szyfrowania był w zasadzie znany, tajne pozostawały wartości parametrów – a więc K_E i oczywiście K_D (*deciphering key*).

W systemach sprzed połowy lat 70-tych XX w. znając algorytm i K_E można było bez trudu (większego) skonstruować K_D – i odwrotnie. Np. w przypadku spisku Babingtona złamanie szyfru (odszyfrowanie tekstu) umożliwiło wysłanie „fałszywki”, która spowodowała ujawnienie przez Marię Stuart nazwisk spiskowców.

Zmiana paradygmatu

W roku 1976 Whitefield Diffie i Martin Hellman (Uniwersytet Stanforda, Palo Alto), a także doktorant M.H. – Ralph Merkle stwarzają ideę systemów kryptograficznych „ z kluczem publicznym”.

Podstawowe założenie: znajomość K_E *nie wystarcza* do odtworzenia K_D – mimo, że znana jest funkcja f to znalezienie f^{-1} jest piekielnie trudne (praktycznie niemożliwe).
Klucz $K_E = K_{E,u}$, ($u = user$) ogólnie dostępny (w książce telefonicznej).

Zmiana paradygmatu – dlaczego?

Bo kryptografia z obszaru nieco niszowego (dyplomacja, operacje wojskowe) wchodzi i to szerokim frontem do domeny publicznej (elektroniczna korespondencja, elektroniczna bankowość i w ogóle wymiana lawin informacji, często o charakterze poufnym, tajnym). Taka skala wymiany informacji wymaga nie tylko „dobrego zabezpieczenia” ale systemu niezbyt skomplikowanego w użyciu – nie do pomyślenia sa np. szyfry wymagające częstej zmiany klucza (i jego wymiany – P2P).

Potwierdzenie tożsamości

Dwoje użytkowników: **A**la i **B**olek. Niech:

- f_A oznacza przekształcenie szyfrujące, którego każdy użytkownik używa do zaszyfrowania wiadomości dla Ali.
- f_B oznacza przekształcenie szyfrujące, ... dla Bolka.
- SA oznacza podpis Ali

Każdy (! – nie tylko Ala) potrafi wysłać do Bolka $f_B(SA)$. Dlatego:

Ala: $\boxed{f_B\left({f_A}^{-1}(SA)\right)} \Rightarrow$ Bolek.

Bolek: $\boxed{{f_B}^{-1} f_B\left({f_A}^{-1}(SA)\right)} = {f_A}^{-1}(SA) = ??.$

Ponieważ to miała być Ala (i/albo wynika to z tekstu podpisu)
Bolek dokonuje prostej transformacji

$$f_A\left[{f_A}^{-1}(SA)\right] = SA$$

i odczytuje podpis. Może być pewny, że *to jest podpis Ali*,
bo tylko Ala dysponuje znajomością funkcji f_A^{-1}.

Wymiana klucza – Diffie, Hellman; prosty przykład

Niech przekształceniem kodującym będzie prosta translacja:
$\mathcal{C} = \mathcal{P} + b \pmod N$. Algorytm jest znany; znana jest też wartość N.
Jak przesłać (Ala $\Leftrightarrow$ Bolek) wartość b?

- A i B ustalają, że wartość klucza to $b \equiv Y^x \pmod M$. W ramach „upubliczniania" powszechnie znane są wartości $Y = 7$ i $M = 11$. Pozostaje wymienić z sobą wartość x.
- Ala wybiera liczbę A, np $A = 3$; oblicza:
  $7^A \pmod{11} = 7^3 \pmod{11} \equiv 2 = \alpha$. Wysyła α do Bolka.
- Podobnie Bolek wybiera liczbę B, np $B = 6$; oblicza:
  $7^B \pmod{11} = 7^6 \pmod{11} \equiv 4 = \beta$. Wysyła β do Ali.
- Ala oblicza:
  $\beta^A \pmod{11} = 4^3 \pmod{11} = 64 \pmod{11} \equiv 9$
- Bolek oblicza:
  $\alpha^B \pmod{11} = 2^6 \pmod{11} = 64 \pmod{11} \equiv 9$

Oczywiście oboje obliczają: b $\equiv 7^{A \cdot B} \pmod{11} = 7^{18} \pmod{11} \equiv 9$

System RSA — Ronald Rivest (MIT), Adi Shamir (Inst. Weizmanna), Leonard Adleman (Berkeley)

Konstrukcja K_E i K_D.

1. Użytkownik A wybiera dwie duże (!!) liczby pierwsze p_A i q_A.
   Liczby te zachowuje w tajemnicy.
2. Oblicza $N_A = p_A \times q_A$ oraz $\phi(N_A) = \phi(p) \cdot \phi(q) = (p-1)(q-1)$;
   wybiera losowo, z przedziału $[1, \phi(N_A)]$ liczbę e_A;
   $NWD(e_A, N_A) = 1$.
   Liczby N_A i e_A podaje jako swój klucz publiczny $- K_{E,A}$.
3. Dla konstrukcji K_D: A oblicza d_A jako rozwiązanie równania
   $\boxed{e_A \cdot d_A \equiv 1 \pmod{\phi(N_A)}}$.
   Liczby d_A i $\phi(N_A)$ zachowuje w tajemnicy.
4. $$\mathcal{C} \equiv \mathcal{P}^{e_A} \pmod{N_A}; \qquad \mathcal{P} \equiv \mathcal{C}^{d_A} \pmod{N_A}.$$

System RSA – uzasadnienie

$$\mathcal{C} \equiv \mathcal{P}^{e_A} \pmod{N_A}; \qquad \mathcal{P} \equiv \mathcal{C}^{d_A} \pmod{N_A}.$$

Z obliczeń modułowych:

$$\{\mathcal{P}^{e_A}\}^{d_A} \equiv \ldots \boxed{\begin{array}{c} e_A \cdot d_A \equiv 1 \pmod{\phi(N_A)} \\ e_A \cdot d_A = \phi(N_A) \cdot t + 1 \end{array}} \ldots$$

$$\equiv \left[\mathcal{P}^{\phi(N_A)}\right]^t \mathcal{P}^1 \pmod{N_A} \equiv \mathcal{P} \pmod{N_A}$$

uwagi praktyczne:

Przypuśćmy, że tekst „jawny” zapisujemy w postaci bloków k-literowych, a tekst zaszyfrowany – w postaci bloków l-literowych i używamy N-literowego alfabetu.

(Blok $k(l)$-literowy to liczba $k(l)$-cyfrowa w systemie pozycyjnym o podstawie k.)

Niech $l > k$.

W praktyce staramy się aby liczby N^l i N^k miały około dwustu cyfr dziesiętnych (uzasadnienie za chwilę). Wówczas powinno zachodzić $N^l > N_A = p_A q_A > N^k$.

System RSA – przykład dla wartości *zupełnie nierealistycznych!*

Operując zestawem 26 (0–25) znaków wartość liczbowa bloku **YES** to:
$\mathcal{P} = 24 \cdot 26^2 + 4 \cdot 26 + 18 = 16\,346.$
Klucz publiczny $(N_A, e_A) = (46\,927, 39\,423)$.
Obliczamy:

$$\mathcal{C} = f(\mathcal{P}) = 16\,346^{39\,423} \pmod{46\,927} \equiv 221\,166 = 1{\cdot}26^3 + 5{\cdot}26^2 + 8{\cdot}26 + 2.$$

Wartość „tekstowa” tej liczby to **BFIC** i to otrzymuje adresat A.
Klucz K_D: $N_A = p_A \cdot q_A = 281 \cdot 167$;
(zauważmy $NWD(e_A, 281 \cdot 167) = 1$.)
A (znając p_A i q_A) oblicza $d_A \cdot e_A \equiv 1 \pmod{280 \cdot 166} \rightarrow d_A = 26767$.
A odkodowuje tekst: $221\,166^{26767} \pmod{46\,927} = 16\,346 =$ **YES**.

System RSA – czy łatwo złamać?

Wkrótce po wprowadzeniu idei kryptografii z kluczem publicznym Martin Gardner w artykule w *Scientific American* ogłosił konkurs, polegający na rozszyfrowaniu tekstu, zakodowanego metodą RSA z liczbą

$N = 114\,381\,625\,757\,888\,867\,669\,235\,779\,976\,146\,612\,010\,218\,296$
$721\,242\,362\,562\,561\,842\,935\,706\,935\,245\,733\,897\,830\,597\,123\,563$
$958\,705\,058\,989\,075\,147\,599\,290\,026\,879\,543\,541; \qquad e = 9007.$

Twórcy RSA otrzymali wówczas ponad 3 tysiące listów z prośbami o instrukcje. Liczbę N rozłożył na czynniki zespół liczący ponad pół tysiąca amatorów. Zespół pracował 8 miesięcy – i 26 kwietnia 1994 ogłosił tryumfalnie $N = p \cdot q$, gdzie

$p = 3\,490\,529\,510\,847\,650\,949\,147\,849\,619\,903\,898\,133\,417\,764$
$638\,493\,387\,843\,990\,820\,577$ oraz

$q = 32\,769\,132\,993\,266\,709\,549\,961\,988\,190\,834\,461\,413\,177$
$642\,967\,992\,942\,539\,798\,288\,533.$

Rozszyfrowanie tzw. RSA-129 odebrali nagrodę – 100\$.
W lutym 1999 zespół informatyków rozłożył RSA-140 (dwa czynniki 70-cyfrowe), a 26/4 1999 inny zespół – liczbę RSA-155.

System RSA – czy łatwo złamać?

Więcej o RSA, o faktoryzacji dużych liczb, można znaleźć wykorzystując linki

w Wikipedii

a także na stronie „laboratoria RSA”

System RSA – wariacje

Zamiast funkcji Eulera można użyć funkcji Carmichaela, $\lambda(N)$.
Funkcja szyfrująca – przy znanym module $N = pq$ i losowo wybranym k, przy czym $NWD(k, \lambda(N)) = 1$ ma postać analogiczną:

$$\mathcal{C} = f(\mathcal{P}) \equiv \mathcal{P}^k \pmod{N}.$$

Jak pamiętamy $\lambda(N) = NWW(p-1, q-1) = \dfrac{(p-1)(q-1)}{NWD[(p-1)(q-1)]}$;
jest ona trzymana w tajemnicy (jak i p i q). Odkodowanie:

$$\mathcal{P} = f^{-1}(\mathcal{C}) \equiv \mathcal{C}^{k'} \pmod{N}; \qquad kk' \equiv 1 \pmod{\lambda(N)}.$$

Uzasadnienie – analogiczne do poprzedniego:

$$\mathcal{P} = (\mathcal{C})^{k'} = \left(\mathcal{P}^k\right)^{k'} = \left(\mathcal{P}^k\right)^{m\lambda(N)+1} = \left(\mathcal{P}^k \lambda(N)\right)^m \cdot \mathcal{P} = \mathcal{P}$$

Idee Diffie'go-Hellman – wariacje

1. Zespół użytkowników uzgadnia liczbę pierwszą q. Każdy użytkownik wybiera (w tajemnicy przed innymi) liczbę e – $0 < e < q-1$ i dodatkowo $e \perp q-1$. Następnie oblicza d – liczbę odwrotną do e modulo $q-1$: $de \equiv 1 \pmod{q}$.
2. Ala wysyła do Bolka wiadomość $\mathcal{P}$. Wysyła $\mathcal{P}^{e_A}$.
3. Bolek nic nie rozumie, ale odsyła Ali $(\mathcal{P}^{e_A})^{e_B}$.
4. Ala podnosi to do potęgi d_A i odsyła do Bolka.
5. Ponieważ $e_A \cdot d_A \equiv 1$ Bolek otrzymuje $\mathcal{P}^{e_B}$ i bez kłopotu odczytuje to, podnosząc do potęgi d_B.

Wady:

1. system wymaga dodatkowo *dobrej* metody potwierdzania tożsamości – każdy może „podszyć" się pod Bolka i wysłać do Ali $(\mathcal{P}^{e_A})^{e_C}$ i dostanie od Ali tekst do odczytania (zna d_C).
2. nieuczciwy Bolek, który odczytał już wiele wiadomości i zna wiele par $(\mathcal{P}, \mathcal{P}^{e_A})$ może je wykorzystać do znalezienia e_A (poprzez obliczenie logarytmu dyskretnego e_A; a następnie $d_A = {e_A}^{-1} \pmod{q-1}$) – wówczas będzie mógł czytać wszystkie informacje rozsyłane przez Alę.

Podpis cyfrowy – schemat El Gamala

Spośród wielu schematów rozpatrzmy: podpis $S = (a, b)$, którym Ala podpisuje pewną wiadomość M.

1. Generacja klucza podpisu Ali:
   - Ala wybiera (losowo) p oraz g i x (obie $< p$).
   - Oblicza $y \equiv g^x \pmod{p}$.
   - Ala upublicznia (y, g, p) – ale x pozostaje tajne.
2. Generacja podpisu
   - Ala wybiera (losowo) $e \perp (p-1)$. Zachowuje tę liczbę jako tajną.
   - Ala oblicza $a \equiv g^e \pmod{p}$ oraz $b \equiv e^{-1}(M - xa) \pmod{p-1}$.
   - Ala podpisuje wiadomość M przy pomocy $S = (a, b)$.
3. Bob musi sprawdzić czy $y^a a^b \equiv g^M \pmod{p}$.
4. Weryfikacja:

$$y^a a^b = (g^x)^a \ (g^e)^{e^{-1}(M-xa)+k\cdot(p-1)} \equiv g^{xa}\, g^{[e\cdot e^{-1}](M-xa)} \left[g^{(p-1)}\right]^k \equiv g^M \pmod{p}.$$

5. Odmiana tego schematu została wykorzystana do stworzenia (NIST, 1991) Standardu podpisu cyfrowego – DSS (*Digital Signature Standard*).