Jest to zapis odczytu wygłoszonego na XLIII Szkole Matematyki Poglądowej, *Wbrew intuicji*, Grzegorzewice, sierpień 2009.

# O problemie milionerów, czyli bezpieczne obliczenia wielostronne

*Małgorzata MISZTAL, Warszawa*

### Bezpieczne obliczenia wielostronne

Bezpieczne obliczenia wielostronne w kryptografii po raz pierwszy pojawiły się w tzw. problemie milionerów, sformułowanym przez Andrew Yao w r. 1982. Dwóch milionerów chce ustalić, który z nich jest bogatszy, ale żaden z nich nie zamierza ujawniać wartości swojego majątku. Pozornie wydaje się to niewykonalne. Yao skonstruował protokół, który pozwoli obu stronom zaspokoić swoją ciekawość przy zachowaniu tego ograniczenia.

Załóżmy w ogólności, że mamy kilku nie ufających sobie graczy – uczestników protokołu. Chcą oni wspólnie obliczyć wartość pewnej funkcji, zależnej od wartości poufnych danych wejściowych należących do poszczególnych uczestników. Dane te mają pozostać nieujawnione innym uczestnikom w trakcie obliczeń.

Problem jest trywialny, jeżeli włączymy tzw. zaufaną trzecią stronę $T$. $T$ gromadzi po prostu dane od wszystkich stron, oblicza wartość funkcji $F$ i ogłasza wynik. W taki właśnie sposób przeprowadzane są wybory – uczestnikami są głosujący obywatele, a zaufaną trzecią stroną, obliczającą sumy głosów oddanych na poszczególnych kandydatów – Państwowa Komisja Wyborcza.

Protokoły bezpiecznych obliczeń wielostronnych konstruowane są po to, aby uzyskać taką samą poufność danych i poprawność wyniku bez wykorzystywania w obliczeniach zaufanej trzeciej strony. Przedstawimy podstawowe przykłady takich protokołów. Zakładamy w nich, że obie strony są „uczciwe, ale ciekawskie" (*honest but curious model*) – tzn. postępują zgodnie z krokami protokołu, podają prawdziwe wartości, a jednocześnie na podstawie posiadanych danych mogą próbować wyciągnąć dodatkowe informacje.

### RSA – algorytm szyfrowania z kluczem publicznym (asymetryczny)

W protokołach będziemy wykorzystywać algorytm RSA. Jest to algorytm szyfrowania z kluczem publicznym, czyli taki, w którym umiejętność zaszyfrowania wiadomości – za pomocą powszechnie znanego *klucza publicznego* – nie jest równoważna umiejętności jej odszyfrowania – do czego służy *klucz prywatny*. Klucza prywatnego nie da się w praktyce odtworzyć na podstawie znajomości klucza publicznego.

Działanie algorytmu opiera się na twierdzeniu Eulera: Jeżeli $a$ i $n$ są względnie pierwsze, wówczas:

$$a^{\phi(n)} \equiv 1 \bmod n$$

gdzie

$$\phi(n) \stackrel{\text{def}}{\equiv} |\{i \leqslant n : \mathrm{nwd}(i,n) = 1\}|$$

Żeby wygenerować klucz RSA, losujemy dwie duże liczby pierwsze $p$ i $q$ oraz liczbę $e$ względnie pierwszą z $\phi(n) = (p-1)(q-1)$. Następnie obliczamy $d = e^{-1} \bmod \phi(n)$. Służy do tego rozszerzony algorytm Euklidesa.

Obliczamy też moduł szyfrowania $n = p \cdot q$ . Kluczem publicznym będzie para $(e, n)$, kluczem prywatnym – para $(d, n)$. Klucz prywatny zachowuje dla siebie strona będąca adresatem zaszyfrowanych wiadomości.

Szyfrowanie polega na podniesieniu liczby $m$ reprezentującej wiadomość do potęgi $e$ modulo $n$:

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

Żeby ją zdeszyfrować podnosimy zaszyfrowaną wiadomość do potęgi $d$. Zgodnie z twierdzeniem Eulera dostaniemy oryginalną wiadomość:

$$c^d = m^{ed} = m \bmod n$$

Nie znając klucza prywatnego $d$, nie potrafimy odczytać oryginalnej wiadomości z kryptogramu. Nie znając faktoryzacji $n$ na $p$ i $q$ nie znamy też prostej metody odtworzenia $d$ z $e$. Nie są znane żadne szybkie (działające w czasie wielomianowym względem liczby cyfr rozkładanej liczby) metody faktoryzacji. Na złożoności obliczeniowej faktoryzacji opiera się bezpieczeństwo RSA.

**Problem milionerów**

Dwoje milionerów, nazwijmy ich Apolonią i Bogusiem, pragnie ustalić, kto z nich jest bogatszy. Żadne z nich nie zamierza jednak zdradać drugiej stronie – ani nikomu innemu – wartości swojego majątku.

Apolonia ma $I$ milionów, Boguś ma $J$ milionów. Poufnymi danymi wejściowymi są więc w protokole $I$ i $J$, obliczaną funkcją – funkcja określająca, czy $I > J$. Przyjmujemy, że są to liczby z określonego przedziału: $I, J \in [1..M]$ (przyjmując odpowiednio duże $M$ nie zdradzimy dodatkowych informacji o $I$ i $J$)

Protokół rozwiązujący problem milionerów przebiega następująco:

| | Apolonia | | Boguś |
|---|---|---|---|
| | $I \in [1..M]$ | | $J \in [1..M]$ |
| 1) | $p, q, n = pq$<br>$n, e, d = e^{-1} \bmod \phi(n)$ – kl. prywatny | | $e$,$n$ – kl. publiczny |
| 2) | | | losowy $x$<br>$C = E_{e,n}(x) = x^e \bmod n$ |
| | | $\xleftarrow{m}$ | $m = C - J + 1$ |
| 3) | $Y_i = D_{d,n}(m + i - 1)$<br>$= (m + i - 1)^d \bmod n, i \in [1..M]$ | | |
| 4) | losowe $p \ll n$<br>$Z_i = Y_i \bmod p$, $i \in [1..M]$ | | |
| 5) | $W_j = Z_i + (i \geqslant I) \bmod p$, $i \in [1..M]$ | $\xrightarrow{p,W_1,..,W_M}$ | |
| 6) | | | wynik : $W_J \equiv x \bmod p$? |
| | | $\xleftarrow{wynik}$ | (nie $\Leftrightarrow I < J$ ) |

1. Apolonia generuje klucze RSA; klucz publiczny $(n, e)$ który udostępnia Bogusiowi i swój tajny klucz prywatny $d$.
2. Boguś wybiera losową wartość $x$. Szyfruje $x$ za pomocą klucza publicznego Apolonii, czyli oblicza $C = x^e \bmod n$, następnie wysyła do Apolonii wartość $m = C - J + 1$. Zauważmy, że dla Apolonii, która nie zna $x$, otrzymana liczba nie powie nic o tajnej wartości $J$ Bogusia.
3. Apolonia przeprowadza teraz – dla wszystkich możliwych wartości $J$ Bogusia – obliczenia odwrotne. Dla liczb $j \in [1..M]$ oblicza wartość $Y_j = (m + j - 1)^d \bmod n$. W przypadku gdy $j = J$ Apolonia otrzyma w wyniku $x$, we wszystkich pozostałych – przypadkowe wartości.
4. Apolonia oblicza $Z_j = Y_j \bmod p$ dla wszystkich $j \in [1..M]$. Należy $p$ dobrać w taki sposób, aby wszystkie zredukowane wartości różniły się co najmniej o 2. Redukcja, która jest operacją jednokierunkową, uniemożliwi później Bogusiowi przeprowadzenie obliczeń odwrotnych.
5. Apolonia dodaje 1 do wszystkich $Z_j$ o indeksie większym lub równym $I$. Następnie przekazuje Bogusiowi ciąg wartości $W_1, W_2, ...W_M$, czyli $Z_1$, $Z_2$, .. ,$Z_{I-1}$,$Z_I + 1$, $Z_{I+1} + 1$, .. $Z_M + 1$ i liczbę $p$. Zauważmy, że Boguś nie jest w stanie odtworzyć tajnego $I$ Apolonii, czyli miejsca w którym $W_j$ przestaje być równe $Z_j$. Aby to zrobić, Boguś musiałby odtworzyć obliczenia Apolonii, czyli najpierw złamać RSA.
6. Boguś sprawdza, czy $W_J \equiv x \bmod p$. Jeżeli nie, to znaczy że Apolonia zaczęła dodawać jedynkę do wartości $Z_j$ dla indeksów mniejszych niż $J$, czyli $I < J$. Tą wiadomością Boguś dzieli się z Apolonią.

**Problem miłosny**

Apolonia kocha Bogusia. Jeżeli Apolonia wyzna Bogusiowi miłość, a uczucie jest wzajemne, wtedy wszystko skończy się happy endem. Jeżeli natomiast Boguś nie kocha Apolonii, to Apolonia będzie niezadowolona, że niepotrzebnie zdobyła się na to wyznanie. Czy możemy pomóc Apolonii? Innymi słowy, czy możemy skonstruować protokół, dzięki któremu Apolonia sprawdzi, czy Boguś ją kocha, nie wyznając mu niepotrzebnie swoich uczuć? Chcemy też, żeby protokół był symetryczny, tzn. pozwolił na to samo Bogusiowi.

Równoważnym problemem jest bezpieczne obliczanie iloczynu logicznego. Każda ze stron $A$ i $B$ ma swój sekretny bit – $a$ i $b$. Jeżeli obliczymy w sposób bezpieczny wartość $a \wedge b$ i ujawnimy obu stronom wynik, to $A$ pozna $b$ tylko wtedy, gdy $a = 1$; analogicznie dla $B$. Przekładając to na język pierwszego problemu – niech 0 oznacza „nie kocha", a 1 – „kocha". Boguś poznając wartość $a \wedge b$ dowie się, że Apolonia go kocha tylko wówczas, gdy on sam odwzajemnia jej uczucie.

**Przekaz nierozróżnialny „1 z 2".** Skonstruujemy najpierw protokół, który działa następująco:

| $A$ | | $B$ |
|---|---|---|
| sekrety $s_0$, $s_1$ | | tajny bit $b \in \{0,1\}$ – (bit wyboru) |
| | $\xrightarrow{s_b}$ | |
| $b =?$ | | |

Pozwala on na przekazanie dokładnie jednego, wybranego przez $B$ sekretu strony $A$ stronie $B$, przy czym $A$ nie dowie się, który sekret został przekazany. Jest to tzw. przekaz nierozróżnialny 1 z 2 *(1 out of 2 oblivious transfer)*:

| | Apolonia | | Boguś |
|---|---|---|---|
| | sekrety $s_0$, $s_1$ | | bit $b \in \{0,1\}$ |
| 1) | Klucz RSA: $n, e, d$<br>losowe $x_0$, $x_1$ | | |
| | | $\xrightarrow{n,e,x_0,x_1}$ | |
| 2) | | | losowe $k$<br>$q = (k^e \bmod n) + x_b$ |
| | | $\xleftarrow{q}$ | |
| 3) | $t_0 = s_0 + ((q - x_0)^d \bmod n)$<br>$t_1 = s_1 + ((q - x_1)^d \bmod n)$ | | |
| 4) | | $\xrightarrow{t_0,t_1}$ | |
| | | | $t_b - k = s_b$ |

Apolonia ma dwa sekrety (zakodowane do postaci liczb): $s_0$ i $s_1$. Boguś ma swój tajny bit $b$, odpowiadający numerowi sekretu Apolonii, który chce poznać. Protokół przebiega następująco:

1. Apolonia generuje klucz publiczny i prywatny RSA oraz dwie losowe liczby $x_0$ i $x_1$. Klucz publiczny, czyli parę $(n, e)$ oraz wartości $x_0$ i $x_1$ przekazuje Bogusiowi.
2. Boguś szyfruje losowo wybraną liczbę $k$ za pomocą klucza publicznego Apolonii i dodaje do wyniku tę z losowych liczb Apolonii, która odpowiada wartości bitu $b$ – np. oblicza $k^e x_0$, jeśli $b = 0$. Boguś przesyła tę wartość Apolonii. Odnotujmy, że nie da jej to możliwości poznania $b$, gdyż nie zna ona wartości $k$ wykorzystanej przez Bogusia w obliczeniach.
3. Apolonia wykonuje teraz obliczenia dla obu możliwych wartości $b$. Oblicza $t_i = s_i + ((q - x_i)^d \bmod N)$ dla $i = 0$ i dla $i = 1$. Jeżeli $i = b$, Apolonia w wyniku otrzyma $s_i + k$, jeżeli $i \neq b$ – przypadkową liczbę. Dla Apolonii obie liczby będą wyglądały równie losowo.
4. Apolonia przekazuje Bogusiowi $t_0$ i $t_1$. Boguś oblicza $t_b - k$, czyli od właściwego $t_i$ odejmuje swoją początkową wartość $k$. W wyniku otrzymuje

sekret $s_b$. Boguś nie pozna drugiego sekretu Apoloii – żeby to zrobić, musiałby złamać algorytm RSA.

**Bezpieczne obliczenie iloczynu logicznego.** Protokół ten możemy wykorzystać do rozwiązania problemu miłosnego, czyli do bezpiecznego obliczenia iloczynu logicznego bitów $a$ i $b$:

| Apolonia | | Boguś |
|---|---|---|
| $s_0 = 0, \quad s_1 = a$ | | $b \in \{0,1\}$ |
| | $\xrightarrow{s_b = a \wedge b}$ | |
| | | $s_b$ |
| | $\xleftarrow{s_b}$ | |
| $s_b$ | | |

Dwoma sekretami Apolonii będą 0 i jej tajny bit $a$ (równy jeden, jeżeli Apolonia kocha Bogusia i zero w przeciwnym przypadku). Bitem, za pomocą którego Boguś wybierze sobie sekret Apolonii będzie jego tajny bit $b$. Przekazany sekret będzie równy iloczynowi logicznemu $a \wedge b$:

- Jeżeli Apolonia nie kocha Bogusia i Boguś nie kocha Apolonii, to przekazany zostanie sekret $s_0$ równy zawsze 0.
- Jeżeli Apolonia nie kocha Bogusia, a Boguś kocha Apolonię, to przekazany zostanie sekret $s_1$, równy w tym przypadku 0. (Pamiętamy, że Apolonia nie wie, który sekret wybiera sobie Boguś.)
- Jeżeli Apolonia kocha Bogusia, a Boguś nie kocha Apolonii, wtedy Boguś wybierze sobie sekret $s_0 = 0$ i nie pozna drugiego z jej sekretów.
- Jeżeli oboje się kochają, wówczas Boguś poprosi o sekret $s_1$, który będzie teraz równy 1.

Uzyskanym wynikiem Boguś dzieli się następnie z Apolonią.

### Przekaz nierozróżnialny

Pierwotną wersję przekazu nierozróżnialnego zaproponował w r. 1981 Michael Rabin. Protokół ten pozwala na przekazanie tajnej wiadomości od $A$ do $B$ z prawdpopodobieństwem 0,5. Innymi słowy – Apolonia po zakończeniu protokołu nie wie, czy zdradziła Bogusiowi swój sekret, czy nie.

| Apolonia | | Boguś |
|---|---|---|
| sekret s | $\xrightarrow{p=0,5}$ | |
| | | *sekret s* |
| ? | | |

Przebieg protokołu:

| Apolonia | | Boguś |
|---|---|---|
| sekret $s$ | | |
| $p, q \in \mathbb{P}, n = pq$ | | |
| klucz publ. RSA $(e, n)$ | | |
| $c = E_{e,n}(s) = s^e \bmod n$ | | |
| | | losowe $x \in \mathbb{Z}_n^*$ |
| | | $a = x^2 \bmod n$ |
| | $\xleftarrow{a}$ | |
| Wybierz $y$ spośród | | |
| czterech $\sqrt{a} \bmod n$ | $\xrightarrow{y}$ | |
| | | Jeżeli $y \not\equiv \pm x \bmod n$ |
| | | użyj $y$ do faktoryzacji $n$ |
| | | i odszyfruj $c$ |

Wykorzystujemy tu znowu kryptosystem RSA oraz własności pierwiastków kwadratowych $\bmod n$, gdzie $n = pq$, $p, q \in \mathbb{P}$. Jeżeli $a$ jest resztą kwadratową

mod $n$, a $n$ jest iloczynem dwóch liczb pierwszych, to $a$ ma cztery pierwiastki kwadratowe mod $n$: $x, -x, y$ i $-y \bmod n$ . Mamy na przykład cztery różne pierwiastki kwadratowe z 1 mod 35: 1, 6, 29 i 34. Jeżeli $n = pq$, $p, q \in \mathbb{P}$, $x^2 \equiv y^2 \bmod n$ i jednocześnie $x \not\equiv \pm y \bmod n$, wówczas mamy:

$$x^2 \equiv y^2 \bmod n \Leftrightarrow n|y^2 - x^2 \Leftrightarrow pq|(y-x)(y+x)$$

czyli dzielnikami $n$ muszą być liczby $\text{nwd}(n, y-x)$, $\text{nwd}(n, y+x)$. Możemy je obliczyć za pomocą algorytmu Euklidesa. Znajomość dwóch odpowiednich pierwiastków kwadratowych mod $n$ pozwala więc na łatwą faktoryzację $n$. Jeżeli znamy tylko $x$ i $-x$, otrzymamy trywialną i bezużyteczną równość $n|n \cdot 0$.

W protokole Apolonia wysyła do Bogusia wiadomość zaszyfrowaną kluczem publicznym RSA. Boguś nie zna klucza prywatnego – bez niego nie odczyta wiadomości. Następnie Boguś wybiera sobie dowolne $x \in \mathbb{Z}_n$ i przekazuje Apolonii kwadrat tej liczby $a = x^2 \bmod n$. Apolonia, znając faktoryzację $n$, potrafi obliczać pierwiastki kwadratowe modulo $n$. Wybiera $y$ – jeden z czterech pierwiastków kwadratowych z $a$, odsyła go Bogusiowi. Z prawdopodobieństwem 0,5 Apolonia wybrała pierwiastek „pożyteczny" dla Bogusia, czyli takie $y$, że $y \not\equiv \pm x \bmod n$. Boguś użyje go, aby rozłożyć $n$ na czynniki, obliczyć klucz prywatny $d$ i odczytać wiadomość.

**Literatura**


[1] Michael O. Rabin, *How to exchange secrets by oblivious transfer* (Technical Report TR-81), Aiken Computation Laboratory: Harvard University, 1981.

[2] A. C. Yao, *Protocols for Secure Computations*, Proceedings of the 21st Annual IEEE Symposium on the Foundations of Computer Science, 1982.

[3] S. Even, O. Goldreich, and A. Lempel, *A Randomized Protocol for Signing Contracts*, Communications of the ACM, Volume 28,1985.