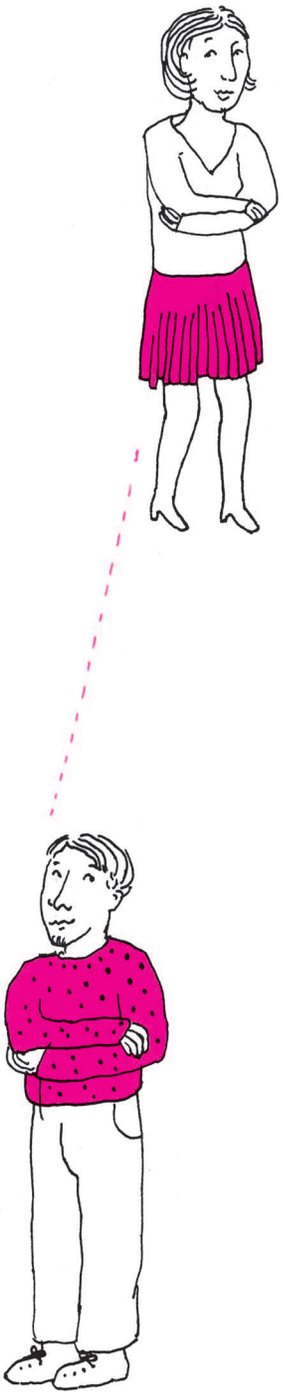
Kocha, lubi, szyfruje...

Tomasz KAZANA[*]



W fizyce szkolnej nieustannie przewijającym się motywem są dwa znane miasta: miasto A oraz miasto B. W kryptografii takimi gwiazdami są Alicja i Bob, którzy ciągle się komunikują, uwierzytelniają, a zwykle przeszkadza im w tym złowroga Ewa.

Tym razem jednak zadanie stojące przed Alicją i Bobem jest wyjątkowo trudne. Chcą sprawdzić, czy się nawzajem kochają, jednak bez ujawniania swoich uczuć. Co przez to rozumiemy? Otóż chcemy opracować następujący protokół komunikacyjny.

Alicja posiada bit a, który informuje o tym, czy kocha Boba, czy nie (to oznacza, że $a = 1$, jeśli Alicja jest zakochana w Bobie, jeśli zaś nie jest, to $a = 0$). Analogicznie, Bob posiada bit b opisujący jego uczucia względem Alicji. Oczywiście teraz ich wzajemna miłość może być opisana w tej notacji przez wyrażenie $m := a \wedge b$, które jest równe 1 tylko wtedy, gdy $a = b = 1$. Naszym celem jest opracowanie takiego protokołu, w wyniku którego Alicja na końcu będzie znała a oraz m (i nic więcej!), natomiast Bob b oraz m. Intuicja stojąca za takimi założeniami jest następująca: chcemy uniknąć krępującej sytuacji, gdy, na przykład, Alicja kocha Boba, ale Bob nie odwzajemnia tych uczuć, a dowiaduje się o uczuciu Alicji. Innymi słowy: jeśli Bob wie, że $b = 0$ oraz $m = 0$, to nie powinien umieć wyznaczyć a.

Powyższe założenia łatwo uzyskać, gdy dopuścimy trzecią zaufaną stronę, która, gdy pozna a i b, obliczy m oraz przekaże tę informację obu zainteresowanym osobom. Nasze zadanie jest jednak bardziej ambitne, ponieważ pożądany efekt chcemy uzyskać za pomocą protokołu, w którym jedynymi stronami są Alicja i Bob. Zadanie samo w sobie wydaje się trudne, a wręcz można by przypuszczać, że tak określony protokół nie jest możliwy do zrealizowania. Pokażemy jednak, że jest to możliwe, ale przy pewnych założeniach obliczeniowych. Mamy tu na myśli klasyczne założenia dotyczące szyfrowania algorytmem RSA: a więc, że strona posiadająca jakąś wiadomość zaszyfrowaną oraz klucz publiczny bez znajomości klucza prywatnego nie jest w stanie odtworzyć wiadomości jawnej. Należy tu poczynić uwagę, że bez założeń tego typu (tj. bez ograniczeń obliczeniowych, inaczej mówiąc: bezpiecznie teorio-informacyjnie) żądanego protokołu nie da się skonstruować.

Wróćmy teraz do naszej Alicji i naszego Boba, którzy już za chwilę zaczną ze sobą rozmawiać.

Szukany protokół opiera się na idei tak zwanego *transferu utajnionego* (ang. *oblivious transfer*). Jest to bardzo ciekawy i użyteczny protokół, niejednokrotnie wykorzystywany jako cegiełka do budowy innych protokołów. Założenia są następujące: Alicja posiada parę liczb (x_0, x_1), a Bob bit s. Po zakończeniu protokołu Alicja nie dowie się niczego nowego, natomiast Bob pozna liczbę x_s. Taki protokół istnieje, pokażemy go później. Natomiast korzystając z transferu utajnionego, możemy już łatwo wykonać nasz docelowy protokół.

Niech mianowicie Alicja przyjmie:

$$x_0 = 0, \qquad x_1 = a,$$

a Bob:

$$s = b.$$

W pierwszej fazie testu na miłość wykonujemy klasyczny transfer utajniony dla podanych parametrów. W jego wyniku Bob otrzymuje wartość $y = x_s$. Wówczas wysyła ją Alicji i jest to wynik naszego protokołu.

Łatwo sprawdzamy poprawność: jeśli $b = 0$, to wynikiem protokołu jest $x_0 = 0$, co jest zgodne z oczekiwaniami, gdyż

$$m = (b \wedge a) = (0 \wedge a) = 0,$$

niezależnie od wartości a. W drugim przypadku ($b = 1$) wynikiem jest $x_1 = a$, co również jest poprawną odpowiedzią, gdyż:

$$m = (b \wedge a) = (1 \wedge a) = a.$$

[*]Instytut Informatyki,
Uniwersytet Warszawski

Transfer utajniony

W tej części pokażemy, jak zrealizować transfer utajniony. Jako narzędzia potrzebujemy szyfrowania z kluczem publicznym i prywatnym – dobrym przykładem jest tu wspomniany już algorytm RSA.

Przypomnijmy tutaj potrzebne informacje na temat algorytmu RSA. Alicja wybiera dwie duże liczby pierwsze p, q i oblicza $n = pq$. Następnie wyznacza liczbę e względnie pierwszą z $\phi(n)$ (czyli liczbą liczb względnie pierwszych z n i mniejszych od n) oraz jej odwrotność d modulo $\phi(n)$. Jej kluczem publicznym jest para (n, e), a kluczem prywatnym para (n, d). Szyfrowanie wiadomości m odbywa się według wzoru $c = m^e \bmod n$, a odszyfrowywanie według wzoru $m = c^d \bmod n$.

Przypomnijmy założenia: Alicja posiada parę liczb x_0 oraz x_1, natomiast Bob ustala bit s. Teraz:

1. Alicja inicjalizuje algorytm RSA, tzn. wybiera liczbę n oraz klucze d i e. Zakładamy, że $0 \leqslant x_0, x_1 < n$. Swój klucz publiczny, czyli parę (n, e), wysyła do Boba.
2. Alicja losuje niezależnie dwie różne liczby $0 \leqslant y_0, y_1 < n$ oraz wysyła je Bobowi.
3. Bob losuje liczbę $0 \leqslant k < n$ i oblicza $v = (y_s + k^e) \bmod n$. Bob wysyła wartość v do Alicji.
4. Alicja oblicza kolejno: $k_0 = (v - y_0)^d \bmod n$ oraz $k_1 = (v - y_1)^d \bmod n$, $x'_0 = (x_0 + k_0) \bmod n$, $x'_1 = (x_1 + k_1) \bmod n$ oraz wysyła Bobowi x'_0 i x'_1.
5. Bob potrafi obliczyć $(x'_s - k) \bmod n$.

Sprawdźmy, że obliczona przez Boba wartość to rzeczywiście x_s:

$$x'_s - k \equiv x_s + k_s - k \equiv x_s + (v - y_s)^d - k \equiv x_s + (y_s + k^e - y_s)^d - k \equiv$$
$$\equiv x_s + k^{ed} - k \equiv x_s \pmod{n}.$$

Aby opisany protokół był użyteczny, potrzebne są dwie własności:

(∗) Alicja nie poznaje bitu s. Ten fakt jest prosty – wynika z symetrii. To znaczy: z punktu widzenia Alicji nie ma w protokole żadnego rozróżnienia między $s = 0$ a $s = 1$, gdyż jedyny komunikat otrzymany od Boba (wartość v) jest – z punktu widzenia Alicji – losowy i niezależny od wartości s.

(∗∗) Bob nie poznaje liczby x_{1-s}.

Pełny dowód (∗∗) pominiemy i pozostawimy Czytelnikowi: wskazówka jest taka, że należy pokazać (metodą nie wprost), że gdyby można było złamać nasz protokół (a więc Bob *dowiedziałby się czegoś* na temat x_{1-s}), to złamać potrafilibyśmy również RSA. Rozumowania tego typu są bardzo częste w dowodach w kryptografii, czasem nazywamy je *symulacją* jednego protokołu przez drugi.

Na sam koniec warto dodać, że opisany tutaj problem jest tylko drobnym przykładem na to, co potrafimy robić. Otóż okazuje się, że funkcja $a \wedge b$ została wybrana zupełnie arbitralnie: tak naprawdę można podać protokół obliczający wartość dowolnej funkcji opisanej za pomocą obwodu logicznego. Wówczas ilość potrzebnej informacji, którą wymienią między sobą Alicja i Bob, jest proporcjonalna do liczby węzłów wspomnianego obwodu. Zainteresowanego Czytelnika zachęcamy do dalszych poszukiwań.

Rozwiązanie zadania M 1350.
Podzbiory A zbioru $\{1, \ldots, n\}$ łączymy w nieuporządkowane pary postaci $\{A, \{1, \ldots, n\} \setminus A\}$. Takich par jest 2^{n-1}. Wobec założenia z treści zadania każdego studenta możemy utożsamiać jednoznacznie ze zbiorem pytań, na które zna odpowiedź. Gdyby studentów było więcej niż 2^{n-1}, to znalazłoby się dwóch, którym odpowiadałyby zbiory A i $\{1, \ldots, n\} \setminus A$. To jednak przeczyłoby założeniu, że na egzaminie było pytanie, na które obaj znali odpowiedź.

Rozwiązanie zadania M 1349.
Załóżmy, że taka funkcja f istnieje. Zauważmy, że

$$f(x)^2 \geqslant f(x+y)(f(x)+y) > f(x+y)f(x),$$

więc $f(x) > f(x+y)$, dla dowolnych $x, y > 0$, co oznacza, że f jest malejąca. Warunek z treści zadania można zapisać tak:

$$f(x) - f(x+y) \geqslant f(x) - \frac{f(x)^2}{f(x)+y} = \frac{1}{\frac{1}{f(x)} + \frac{1}{y}}.$$

Ustalmy $x > 0$ i weźmy taką liczbę całkowitą n, aby $nf(x+1) \geqslant 1$. Wtedy dla $k = 0, 1, \ldots, n-1$ mamy

$$f\left(x + \frac{k}{n}\right) - f\left(x + \frac{k+1}{n}\right) \geqslant \frac{1}{\frac{1}{f\left(x + \frac{k}{n}\right)} + n} \geqslant \frac{1}{2n},$$

skąd, sumując stronami n nierówności, dostajemy

$$f(x) - f(x+1) > \frac{1}{2}.$$

Zatem dla dowolnej liczby całkowitej dodatniej m zachodzi nierówność

$$f(x) - f(x+m) > \frac{m}{2}.$$

Biorąc m tak duże, aby było $m/2 > f(x)$, otrzymujemy $f(x+m) < 0$, co przeczy temu, że f przyjmuje wyłącznie wartości dodatnie.