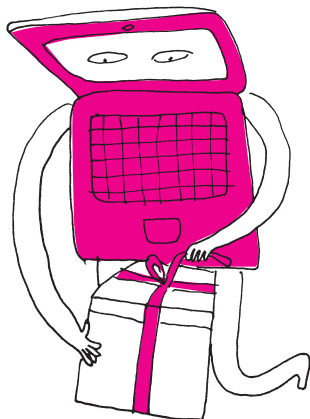


## BB84 zgłoś się

Łukasz RAJKOWSKI



W protokole Diffiego–Hellmana agent  $B$  i admirał  $M$  ustalają najpierw dużą liczbę pierwszą  $p$  oraz niezerową resztę  $g$  z dzielenia przez  $p$  (i nie boją się, że wartości te zostaną podsłuchane). Następnie każdy z nich wymyśla sobie liczbę naturalną (oznaczymy je odpowiednio przez  $x$  i  $y$ ), po czym  $B$  wysyła do  $M$  wartość  $b = (g^x \bmod p)$ , a  $M$  odsyła  $B$  wartość  $m = (g^y \bmod p)$ . Wspólnie ustalonym kluczem jest wówczas  $k \equiv b^y \equiv m^x \equiv g^{xy} \bmod p$ , co (jak na razie) jest trudne do obliczenia wyłącznie na podstawie  $p$ ,  $g$ ,  $b$  i  $m$ .

Czytelnikowi Obeznanemu z Algebrą Liniową proponujemy upewnić się, że przedstawione wyobrażenie jednokubitowego komputera kwantowego zgadza się z modelem obliczeń kwantowych, przedstawionym przez Tomasza Kazanę na stronie 2.

Jak można dowiedzieć się z rozlicznych filmów akcji, nieodłączną częścią życia każdego szanującego się tajnego agenta jest wymiana tajnych informacji, najlepiej takich z wielką, czerwoną pieczęcią „Top Secret”. Jeśli agent ma taką możliwość, najlepiej przekazać teczkę pełną tajemnic osobiście, jednak jest to luksus, na który może on pozwolić sobie w niewielu sytuacjach, gdyż nierzadko odbiorca tych tajemnic znajduje się na drugim końcu globu. W tej sytuacji konieczne staje się odpowiednie zaszyfrowanie naszych sekretów, aby nawet w przypadku przechwycenia ich przez oślizgłe macki szwarzcharakterów, pozostały one sekretami.

Najsukuteczniejszym (i najprostszym) sposobem szyfrowania jest *one time pad*, w którym tajny agent (nadamy mu kryptonim  $B$ ) ustala z odbiorcą swoich wiadomości (nazywanym dalej admirałem  $M$ ) klucz  $k$ , będący pewnym zerojedynkowym ciągiem. Kiedy  $B$  chce przesłać  $M$  uzyskane sekrety, najpierw konwertuje je w ustalony sposób na zerojedynkowy ciąg  $m$  (np. „a” = 00000, „b” = 00001, „c” = 00010 itd.), a następnie wysyła ciąg  $e$  (zwany *kryptogramem*), który na  $i$ -tej współrzędnej ma resztę z dzielenia przez 2 sumy  $i$ -tych współrzędnych ciągów  $m$  i  $k$  (operacja ta w informatycznej nowomowie nazywana jest *xorem* ciągów  $m$  i  $k$  i oznaczana przez  $m \oplus k$ ). W ten sposób, jeśli tajna informacja to  $m = 10110$ , a  $k = 10101$ , to kryptogramem jest  $e = 00011$ . Aby odcyfrować otrzymany kryptogram,  $M$  wykonuje tę samą operację, która posłużyła  $B$  do szyfrowania – oblicza  $d = e \oplus k$ . Nietrudno sprawdzić, że  $d = m$ , mamy bowiem  $d = e \oplus k = (m \oplus k) \oplus k = m \oplus (k \oplus k) = m$ . Ponadto, jeśli  $e$  zostanie przechwycone przez wrogie siły (reprezentowane przez nikczemnego doktora  $N$ ), to nie będą one w stanie wywnioskować stąd  $m$ , gdyż byłoby to równoważne z ustaleniem wartości klucza ( $k = e \oplus m$ ), o którym zakładamy przecież, że jest znany tylko  $B$  i  $M$ . Słabym punktem powyższej procedury jest konieczność spotkania się  $B$  i  $M$  w celu ustalenia klucza, nie mogą go bowiem przesłać na odległość, gdyż mógłby on zostać „podsłuchany” przez doktora  $N$ ... , a może jednak mogą?

W 1976 roku Whitfield Diffie i Martin Hellman zaproponowali protokół (znany teraz pod nazwiskami twórców), który pozwala na ustalenie klucza w taki sposób, że nawet jeśli komunikacja zostanie podsłuchana, to strona podsłuchująca nie będzie w stanie odcyfrować prawdziwej wartości klucza. Protokół ten wiąże się z trudnością wykonania tzw. *logarytmu dyskretnego*, czyli odpowiedzi na pytanie w rodzaju „do jakiej potęgi należy podnieść 123, aby otrzymać resztę 321 z dzielenia przez 983?”. Jak jednak można dowiedzieć się w tym numerze, w „postkwantowym” świecie z dostępem do wydajnych komputerów kwantowych logarytm dyskretny przestaje być zadaniem trudnym. Na szczęście (dla agenta  $B$ ) komputery kwantowe dostarczają również nowe narzędzia do kodowania wiadomości i to takie, których w „dowodliwy” sposób same nie mogą przełamać. W tym kontekście podstawową własnością komputerów kwantowych jest fakt, że sprawdzenie stanu ich pamięci często powoduje zmianę tego stanu, co pozwala na wykrycie próby podsłuchania informacji i zerwanie połączenia, a potem poszukiwanie bezpieczniejszego kanału komunikacji. Realizacja tej idei została opisana w 1984 roku przez Charlesa Bennetta i Gillesa Brassarda, a przedstawiony przez nich protokół to tytułowy BB84.

Aby wytłumaczyć działanie BB84, zanurzmy się w oparach absurdu i będziemy wyobrażać sobie najprostszy, jednokubitowy komputer kwantowy jako czarną skrzynkę, w której zamknięta jest tarcza zegara wraz ze wskazówką godzinową – godzinę przez nią wskazywaną nazwiemy stanem komputera. W jaki sposób można czegoś się o nim dowiedzieć? Można poprosić komputer o porównanie swojego stanu z konkretną „eLką” – pod tym pojęciem rozumiemy układ dwóch „prostopadłych” godzin, np. 1:30 i 4:30, a ogólnie godzinę  $h$  wraz z godziną  $h + 3$ . Jeśli poprosimy komputer znajdujący się w stanie  $x$  o porównanie z eLką  $(h_1, h_2)$ , to możemy otrzymać dwie odpowiedzi:  $h_1$  z prawdopodobieństwem  $\cos^2 \alpha_1$  oraz  $h_2$  z prawdopodobieństwem  $\cos^2 \alpha_2$ , gdzie  $\alpha_1, \alpha_2$  to kąty, jakie tworzy godzina  $x$  odpowiednio z godzinami  $h_1$  i  $h_2$  (prawdopodobieństwa te sumują się do 1, dlaczego?). Ponadto po przedstawieniu odpowiedzi komputer nagina do niej rzeczywistość, to znaczy zmienia swój stan na zgodny z odpowiedzią.

Dla przykładu, jeśli komputer był w stanie 4:30 i poprosiliśmy go o porównanie z eLką (2:30, 5:30), to z prawdopodobieństwem  $\cos^2 60^\circ = \frac{1}{4}$

$w_i$	$l_i$	$s_i$	$l'_i$	$s'_i$
1	1	↘	0	→
0	0	↑	0	↑
0	1	↗	1	↗
0	1	↗	0	→
1	0	→	1	↘
1	1	↘	1	↘
0	0	↑	0	↑
1	0	→	0	→
0	0	↑	1	↘
0	1	↗	1	↗

Przykładowy przebieg protokołu BB84. Szara czcionka występuje w wierszach, w których  $B$  i  $M$  użyli różnych eLek. Kolorem zaznaczone zostały wiersze zawierające wyrazy ciągu  $w$  upublicznione przez  $B$ . Jeśli nie pojawiły się żadne rozbieżności, wspólnym kluczem jest 001.

$w_i$	$l_i$	$s_i$	$l_i^N$	$s_i^N$	$l'_i$	$s'_i$
1	1	↘	0	↑	0	↑
0	0	↑	1	↘	0	↑
0	1	↗	0	→	1	↗
0	1	↗	1	↗	0	→
1	0	→	0	→	1	↗
1	1	↘	1	↘	1	↘
0	0	↑	1	↘	0	→
1	0	→	0	→	0	→
0	0	↑	1	↗	1	↗
0	1	↗	0	↑	1	↘

Przykładowy przebieg podsłuchiwanego protokołu BB84. W czwartej i piątej kolumnie znajdują się eLki wykorzystywane przez podsłuchującego doktora  $N$  oraz stany, w jakich odpowiednio komputery kwantowe znalazły się po niepożądanym odczycie. W ostatnim wierszu wystąpiła rozbieżność między  $B$  i  $M$ , co dowodzi zaistnienia podsłuchu.

Warto zauważyć, że doktor  $N$  mógłby odczytać stany tylko kilku komputerów kwantowych w nadziei, że pozna pewne współrzędne  $w$ , a jego podsłuch nie zostanie wykryty. W obronie przed takim zagrożeniem poznane przez  $M$  i nieopublikowane przez  $B$  wartości  $w$  poddawane są ekstraktorem losowości – są to specjalne funkcje, dla których znajomość niewielkiej części argumentów nie niesie ze sobą żadnej istotnej informacji o wyniku. Dopiero tak uzyskana wartość jest wykorzystywana przez  $B$  i  $M$  jako klucz.

otrzymamy odpowiedź 2:30 (i wówczas stan komputera zmienia się na 2:30), a z prawdopodobieństwem  $\cos^2 30^\circ = \frac{3}{4}$  usłyszymy 5:30 i taką godzinę zacznie wskazywać wskazówka wewnątrz czarnej skrzynki. Zwróćmy ponadto uwagę, że jeśli stan komputera pokrywa się z jedną z godzin z wybranej przez nas eLki, to na pewno uzyskamy tę godzinę w odpowiedzi, a stan komputera nie ulegnie zmianie. Przejdźmy do opisu protokołu. Wyróżnijmy na początku dwa rodzaje eLek:  $L_0 = (0:00, 3:00)$  i  $L_1 = (1:30, 4:30)$ . Agent  $B$  zaopatruje się w  $n$  jednokubitowych komputerów kwantowych (gdzie  $n$  jest raczej duże) i wybiera losowo dwa ciągi zerojedynkowe:  $w = (w_1, \dots, w_n)$  oraz  $l = (l_1, \dots, l_n)$ , a następnie stan  $i$ -tego komputera ustawia na godzinę  $s_i$  będącą  $(w_i + 1)$ -szą współrzędną eLki  $L_{l_i}$ . W ten sposób, jeśli  $w_5 = 1, l_5 = 0$ , to stan piątego komputera zostanie ustawiony na drugą współrzędną  $L_0$ , czyli na 3:00. Następnie agent  $B$  przesyła pocztą wszystkie komputery do  $M$ . Ten ostatni również ustala zerojedynkowy ciąg  $l' = (l'_1, \dots, l'_n)$  i odczytuje stan  $i$ -tego komputera przy użyciu eLki  $L_{l'_i}$ . Później na swojej stronie internetowej (lub innym ogólnodostępnym medium, które tylko on może edytować) udostępnia użyty ciąg  $l'$ , w odpowiedzi na co agent  $B$  publikuje na swojej stronie ciąg  $l$ . Zauważmy, że wszędzie tam, gdzie  $l_i = l'_i$ , admirał  $M$  odczytał godzinę zakodowaną przez agenta  $B$ , a skoro wie również, jaka eLka posłużyła do jej zakodowania, pozna  $w_i$ . Zwróćmy ponadto uwagę, że szansa na to, by  $l_i = l'_i$  wynosi 50%, dlatego  $M$  powinien poznać około połowy wyrazów ciągu  $w$ . Na koniec  $B$  upublicznia połowę z tych wyrazów  $w$ , które powinien był poznać  $M$ . Dlaczego?

Przypomnijmy, że cały ten ambaras miał na celu popsucie szyków nikczemnemu doktorowi  $N$ , który przechwycił paczkę z komputerami i postanowił odczytać ich stany. Poprzez odczyt stanu  $i$ -tego komputera stwarza on szansę na zmianę tego stanu i tylko w tej sytuacji możliwe jest, aby  $M$ , pomimo równości  $l_i = l'_i$ , odczytał złą wartość  $w_i$ . Rozpatrzmy sytuację, w której doktor  $N$  użył  $L_0$  do odczytania stanu  $i$ -tego komputera, przy czym  $B$  i  $M$  użyli na tej współrzędnej identyczne eLki (a zatem, gdyby nie podsłuch,  $M$  na pewno poznałby wartość  $w_i$ ). Jeśli  $B$  również zakodował wiadomość przy użyciu eLki  $L_0$ , to  $N$  odczytał pierwotny stan  $i$ -tego komputera (w związku z czym również  $w_i$ ) i go nie zmienił, w związku z czym szpiegostwo pozostanie niewykryte. Jeśli jednak  $B$  użył  $L_1$ , to  $N$  zmienił stan komputera na którąś z godzin 0:00, 3:00 i w każdym z tych przypadków  $M$  będzie miał szansę 50% na błędny odczyt  $w_i$ . Podsumowując, jeśli doktor  $N$  wybrał eLkę  $L_0$  do odczytu stanu  $i$ -tego komputera, to z prawdopodobieństwem 25% admirał  $M$  odczyta złą wartość  $w_i$ .

Okazuje się, że jest tak niezależnie od eLki wybranej przez doktora  $N$  (uzasadnienie jest wdzięcznym ćwiczeniem z trygonometrii). Fakt ten tłumaczy ostatnią, „kontrolną” fazę naszego protokołu: jeśli po ujawnieniu przez agenta  $B$  połowy wyrazów ciągu  $w$ , dla których  $l_i = l'_i$ , admirał  $M$  stwierdzi jakakolwiek niezgodność ze swoimi odczytami, oznaczać to będzie, że komunikacja została podsłuchana i w związku z tym należy ją powtórzyć, najlepiej przy użyciu bardziej wiarygodnej poczty. Jeśli natomiast wszystkie wartości ujawnione przez  $B$  zgadzały się z odczytami  $M$ , to prawdopodobieństwo takiego zdarzenia przy założeniu o podsłuchiwanie kanału wyniosłoby  $75\%^{k/2}$  (gdzie  $k \approx n/2$  to liczba indeksów  $i$ , dla których  $l_i = l'_i$ ), co dla odpowiednio dużych wartości  $n$  jest na tyle małe, że można z czystym sumieniem odrzucić hipotezę o podsłuchu i wykorzystać nieujawnione przez agenta  $B$  i poprawnie obliczone przez  $M$  wartości  $w_i$  jako wspólny klucz.

Najwyższy czas opuścić opary absurdu i stawić czoła brutalnej, szpiegowskiej rzeczywistości – przecież żaden tajny agent nie będzie wysyłał pocztą tysiąca jednokubitowych komputerów kwantowych. Poczta pewnie nie, ale już światłowodem bez problemu! Okazuje się bowiem (za czym stoi fizyczna magia, o której trochę piszemy w tym numerze), że jednokubitowy komputer kwantowy, ta czarna skrzynka z zamkniętym w środku zegarem, to (w rozsądnym uproszczeniu) po prostu foton, a tych przecież nie brakuje i przesyłać przy użyciu światłowodu też je można. Nie jest to jednak tanie – agenci chcący zaopatrzyć się w parę odbiorników oraz odpowiedni światłowód muszą liczyć się z wydatkiem rzędu 100 tysięcy dolarów; czego jednak się nie robi w tajnej służbie Jej Królewskiej Mości. . .