Jędrzej Wydra

Czy Twoje pikantne wiadomości są bezpieczne, czyli jak działa szyfrowanie?

Ostatnio pisząc do koleżanki z pytaniem “Co masz na sobie?” zacząłem się zastanawiać co sprawia, że nasza rozmowa jest prywatna i jak to jest, że nikt postronny nie może odczytać naszych wiadomości (całe szczęście). Przecież popularne komunikatory takie jak Whatsapp czy Messenger działają za pośrednictwem Internetu, czyli nasze wiadomości muszą przejść przez globalną sieć. Więc jak to jest, że tylko nadawca i odbiorca mają do nich dostęp? To oczywiście proste, bo prywatność zapewnia nam szyfrowanie. Tylko jak ono w zasadzie działa?

Pierwszym pomysłem jaki przychodzi mi na myśl, kiedy myślę o szyfrowaniu jest przestawianie liter. Ustalmy, że zamiast “A” będę pisał “B”, zamiast “B” będę pisał “C” i tak dalej, dodatkowo zamiast “Z” będę pisał “A”. Wówczas słowo “wydra” zaszyfrujemy jako “xzesb”. Wygląda dobrze, ale czy jest bezpieczne? Zauważmy, że gdybym chciał taką wiadomość wysłać do odbiory, to musiałbym też w jakiś sposób poinformować go jak wiadomość odczytać. Czyli w bezpieczny i prywatny sposób muszę przesłać odbiorcy klucz. Tylko, gdybym mógł bezpiecznie przesłać klucz, to po co byłoby to całe szyfrowanie? Nie lepiej bezpiecznie przesłać samą wiadomość? No więc potrzebny jest inny sposób.

Z pomocą przychodzi tzw. szyfrowanie asymetryczne, czy technika pozwalająca nadawcy i odbiorcy wspólnie ustalić klucz szyfrowania. Jednym z takich sposobów jest protokół Diffiego-Hellmana. Ten polega na wspólnym ustalaniu tzw. kluczy publicznych i kluczy prywatnych umożliwiających zaszyfrowanie i odszyfrowanie wiadomości. Przy czym bezpieczeństwo gwarantuje to, że żadna strona nie posiada wszystkich informacji.

Wyobraźmy sobie, że osoba A i osoba B wspólnie wybierają jakąś liczbę pierwszą, np. 23. Następnie, w tajemnicy wybierają po jednej liczbie, niech A wybierze 4, a B niech wybierze 5. Teraz A oblicza iloczyn 23 * 4 = 92, a B oblicza iloczyn 23 * 5 = 115. Te liczby będą stanowiły klucze publiczne, którymi osoby jawnie się wymieniają.

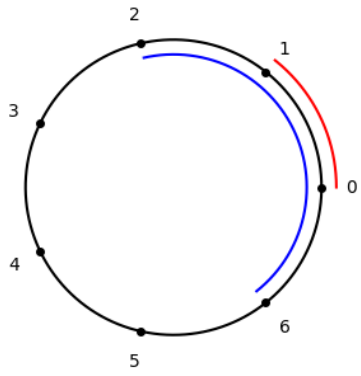
Wówczas osoba A dysponuje informacjami: 1) klucz publiczny osoby B to 115, 2) klucz prywatny osoby A to 4. Osoba B dysponuje informacjami 1) klucz publiczny osoby A to 92, 2) klucz prywatny osoby B to 5. Obie osoby trzymają swoje klucze prywatne w tajemnicy. Posiadając wspomniane informacje obie osoby mogą dojść do tego samego klucza szyfrującego, czyli 115 pomnożone przez 4 to 460, więc dokładnie tyle samo, ile 92 pomnożone przez 5. Teraz gdy obie osoby ustalą (mogą to zrobić publicznie), że litery w wiadomościach zmieniają na liczby biorąc ich numer porządkowy według alfabetu, czyli A to 1, B to 2, C to 3 itd., to mogą szyfrować wiadomości bez konieczności bezpiecznego połączenia. Odbywa się to następująco:

Słowo „wydra” zmienia się w ciąg liczb „23-25-4-18-1”. Osoba B szyfrując tę wiadomość doda do każdej liczby 460 otrzymując „483-485-464-478-461” i prześle ten ciąg do A. Natomiast osoba A otrzymawszy zaszyfrowaną wiadomość odejmie od każdej liczby 460 i otrzyma wyjściowy kod „23-25-4-18-1”, z którego już bardzo łatwo odczytać, że chodziło o „wydra”. Wydaje się, że pomysł jest dobry. Liczby 92 (klucz publiczny osoby A) i 115 (klucz publiczny osoby B) są powszechnie znane, ale liczby niezbędne do otrzymania klucza szyfrującego (którym jest 460), czyli klucze prywatne, są tajne i znają je tylko osoby zainteresowane.

Jest tylko mały problem. Na początku osoby A i B publicznie ustaliły liczbę 23. Gdy klucze publiczne zostaną podzielone przez tę liczbę, to można odtworzyć informację o tajnych kluczach prywatnych. To oznacza, że szyfrowanie wciąż nie jest bezpieczne. No ale co, gdyby operacje na liczbach nie były takie proste? Okazuje się, że wcale nie muszą być proste.

Jednym z podstawowych pojęć algebry abstrakcyjnej jest grupa, czyli struktura składająca się z pewnego zbioru i określonego na nim działania, które jest łączne (w przypadku operacji na trzech elementach można przestawiać nawiasy), posiada element neutralny (element, którego działanie na inny element nie powoduje jego zmiany, w dodawaniu zero jest elementem neutralnym, bo gdy dodamy zero do dowolnej liczby, to otrzymamy wyjściową liczbę, w mnożeniu będzie to jedynka) oraz dla każdego elementu można znaleźć element odwrotny (gdy na dany element zadziała się jego elementem odwrotnym, otrzyma się element neutralny, w dodawaniu będzie to liczba przeciwna, bo choćby 2 + (-2) = 0; a w mnożeniu liczba odwrotna, bo 2 * ½ = 1). Liczy rzeczywiste z dodawaniem tworzą grupę; liczby rzeczywiste z mnożeniem też tworzą grupę. Tylko w tych grupach działania są proste.

Geometrycznie, liczby rzeczywiste tworzą nieskończoną prostą, okazuje się, że działania na strukturze okręgu są trudniejsze.



Weźmy okrąg i wybierzmy na nim pewną skończoną liczbę punktów, np. siedem punktów jak na rysunku powyżej. Niech teraz dodawanie oznacza przechodzenie od punktu do punktu przeciwnie do ruchu wskazówek zegara o określoną liczbę kroków. W

ten sposób 6 + 3 = 2, bo szóstkę i dwójkę dzielą trzy kroki idąc przeciwnie do ruchu wskazówek zegara. W takiej strukturze analogicznie można określić mnożenie, które będzie wielokrotnym dodawaniem, przykładowo 2 * 4 = 2 + 2 + 2 + 2 = 2 + 6 = 1. Jest to tak zwane mnożenie modulo, ponieważ jego wynik jest resztą z dzielenia zwykłego iloczynu przez liczbę punktów na okręgu. Przykładowo, zwykłym iloczynem jest 2 * 4 = 8, gdy teraz 8 podzielimy na 7 to otrzymamy 1 i resztę 1. Reszta jest wynikiem mnożenia na okręgu. Analogicznie 5 * 4 = 20, a 20 podzielone na 7 to 2 i reszta 6, więc na okręgu 5 * 4 = 6. Aby podkreślić, że chodzi o działania na okręgu w nawiasie podaje się określenie operacji modulo, czyli w tym przypadku powinno być 5 * 4 = 6 (mod 7). Ale dla uproszczenia zapisu, dalej będziemy to pomijać.

Tylko uwaga! Aby na okręgu utworzyć grupę z mnożeniem, należy pominąć zero.

Gdy teraz osoby A i B wykonają protokół Diffiego-Hellmanna na określonym okręgu, to otrzymają bezpieczniejsze szyfrowanie. Niech odbędzie się to na sześciopunktowym okręgu. Osoby A i B wspólnie wybierają pewną liczbę spośród 1, 2, 3, 4, 5, 6. Nie powinni robić tego całkiem dowolnie, bo powinni wybrać tzw. generator grupy, czyli liczbę, która podnoszona do kolejnych potęg da wszystkie elementy grupy, czyli:

- $1^1 = 1$, $1^2 = 1$, $1^3 = 1$, $1^4 = 1$, $1^5 = 1$, $1^6 = 1$, $1^7 = 1$;

- $2^1 = 2$, $2^2 = 4$, $2^3 = 1$, $2^4 = 2$, $2^5 = 4$, $2^6 = 1$, $2^7 = 2$;

- $3^1 = 3$, $3^2 = 2$, $3^3 = 6$, $3^4 = 4$, $3^5 = 5$, $3^6 = 1$, $3^7 = 3$;

- $4^1 = 4$, $4^2 = 2$, $4^3 = 1$, $4^4 = 4$, $4^5 = 2$, $4^6 = 1$, $4^7 = 4$;

- $5^1 = 5$, $5^2 = 4$, $5^3 = 6$, $5^4 = 2$, $5^5 = 3$, $6^6 = 1$, $5^7 = 5$;

- $6^1 = 6$, $6^2 = 1$, $6^3 = 6$, $6^4 = 1$, $6^5 = 6$, $6^6 = 1$, $6^7 = 6$.

Łatwo teraz zauważyć, że generatorami są 3 i 5. Ustalmy, że osoby A i B wybrały 3. Teraz obie wybierają swoje klucze prywatne. Niech A wybierze 2, B wybierze 3. Następnie tworzą klucze publiczne podnoszą ustalony generator do odpowiedniej potęgi. Czyli kluczem publicznym osoby A jest $3^2 = 2$, a kluczem publicznym osoby B jest $3^3 = 6$. Aby teraz poznać klucz szyfrujący, każda z osób w tajemnicy podnosi klucz publiczny tej drugiej do potęgi, której wykładnikiem jest klucz prywatny. Czyli osoba A oblicza $6^2 = 1$, a osoba B oblicza $2^3 = 1$. Okazuje się, że tak jak w grupie rzeczywistej z dodawaniem obie osoby otrzymały ten sam klucz szyfrujący, przy czym żadna nie posiada pełnej informacji.

Ale dlaczego jest to bezpieczniejsze niż w przypadku grupy rzeczywistej z dodawaniem? Publicznie znane są informacje, że osoby A i B działały w sześcioelementowej grupie z mnożeniem, czyli wykonywały operacje modulo 7; wiadomo też, że wspólnie ustaliły liczbę 3, klucz publiczny osoby A to 2, a klucz publiczny osoby B to 6.

Bezpieczeństwo grupy skończonej polega na tym, że w przeciwieństwie do grupy liczb rzeczywistych, nie ma w niej prostego sposobu na obliczenie klucza prywatnego. Tzn. nie ma efektywnego sposobu na rozwiązanie równania, w którym generator podniesiony do potęgi o wykładniku równym kluczowi prywatnemu pierwszej osoby jest równy kluczowi publicznemu drugiej osoby. W omawianym przykładzie przyjmuje postać:

$$3^a = 2$$

lub

$$3^b = 6$$

gdzie a i b to odpowiednio klucze prywatne osób A i B.

W liczbach rzeczywistych z mnożeniem wystarczy zlogarytmować obie strony. W grupie skończonej nie da się tego zrobić. Aby rozwiązać to równanie trzeba sprawdzić wszystkie możliwości. Gdy działamy w tak małej grupie jak w przykładzie, to zadanie jest proste. Ale gdybyśmy wybrali zamiast sześcioelementowej grupy z mnożeniem, np. grupę o liczbie elementów równej 10 006 (wówczas mówi się, że grupa ma rząd równy 10 006), to zadanie jest w zasadzie niewykonalne bez komputera. Z pomocą komputera rozwiązanie równania w grupie o rzędzie 100 000 000 000 061 mogłoby trwać nawet około 1000 lat. Przy czym liczba 100 000 000 000 061 jest liczbą 47-bitową (do jej zapisania w systemie binarnym potrzeba 47 bitów). W praktyce szyfrowania używa się nawet 2048-bitowych liczb. Wobec tego próby obliczenia klucza prywatnego na podstawie publicznych informacji są w zasadzie bezcelowe.

Na tym etapie widać też, dlaczego osoby A i B powinny były wybrać generator grupy, a nie po prostu dowolną. Zauważmy, że równanie pozwalające na wyliczenie klucza prywatnego będzie miało więcej niż jedno rozwiązanie, gdy wspólnie ustalona liczba nie będzie generatorem. Wówczas, będzie istniał dodatkowy klucz, który umożliwi odszyfrowanie wiadomości, co zmniejszy poziom bezpieczeństwa.

Czy takie szyfrowanie już jest całkiem bezpieczne? Nie do końca. Owszem, obliczenie klucza prywatnego jest praktycznie niewykonalne, ale nadal można przeanalizować otrzymany szyfrogram i wyciągnąć wnioski na podstawie częstości występowania poszczególnych liczb. Przykładowo, gdy wiem, że w języku polskim najczęściej pojawiającą się literą jest „a”, to mogę częściowo odszyfrować wiadomość, gdy jest ona odpowiednio długa. Wobec tego, gdy w przechwyconej wiadomości bardzo często będzie pojawiało się np. 3522, to jest wysoce prawdopodobne, że oznacza literę „a”. Wówczas mogę zamienić wszystkie wystąpienia tego ciągu literą „a” i próbować odgadnąć wiadomość. Dalej mogę skorzystać z kolejnego często pojawiającego się znaku i zamienić go na kolejną popularną w danym języku literę. Po kilku takich próbach uda się odtworzyć wiadomość bez znajomości klucza prywatnego. Jednak i na to jest rozwiązanie. Jest nim szyfrowanie probabilistyczne, które realizuje chociażby algorytm

ElGamala. Szyfrowanie probabilistyczne wprowadza element losowy, który sprawia, że ta sama wiadomość (w szczególności ten sam znak) za każdym razem otrzyma inny szyfrogram.

Ponownie zaszyfrujmy słowo „wydra”. Najpierw należy wygenerować klucze publiczne i prywatne, a następnie przekształcić słowo na ciąg liczb. Wykorzystamy tę samą metodę co dotychczas.

Niech rzędem grupy skończonej będzie 10 006, a ustaloną liczbą będącą generatorem będzie 536. Osoba A jako klucz prywatny wybiera 345, a osoba B wybiera 567. Wówczas kluczem publicznym osoby A, który zawiera wszystkie publiczne informacje będzie:

10 007-536-8 336.

Pierwsza liczba określa operacje modulo, druga jest generatorem grupy, trzecia to właściwy klucz publiczny.

Analogicznie dla osoby B klucz publiczny zawierający wszystkie informacje to:

10 007-536-4 746.

Wiadomość „wydra” zostaje przekształcona do ciągu 23-25-4-18-1.

Teraz gdy osoba B chce zaszyfrować wiadomość dla osoby A, to przed szyfrowaniem każdego znaku wybiera losową liczbę k, którą nazwiemy kluczem tymczasowym. Postępowanie jest następujące:

1. Niech losowo wybraną liczbą będzie k = 3, wówczas generator jest podnoszony do potęgi z wykładnikiem klucza tymczasowego i klucza prywatnego osoby B: $(536^3)^{567}$ = 4 337; klucz publiczny osoby A jest podnoszony do potęgi klucza tymczasowego i klucza prywatnego osoby B: $(8\ 336^3)^{567}$ = 182. Otrzymany klucz szyfrujący jest mnożony przez znak do zaszyfrowania: 23 * 182 = 4 186. Jako szyfrogram pierwszego znaku podaje się wtedy parę: (4 337, 4 186);
2. Analogicznie postępuje się z kolejnymi znakami. Załóżmy, że wylosowano następujące klucze tymczasowe: 3, 5, 2, 1, zatem otrzymano zaszyfrowaną wiadomość postaci:

(4 337, 4 186)-(4 337, 4 550)-(1 549, 3 010)-(8 766, 8 157)-(4 746, 7 398).

Gdy teraz osoba A zechce odszyfrować tę wiadomość powinna podnieść pierwszy element każdej pary do potęgi swojego klucza prywatnego, a następnie znaleźć element odwrotny do takiej liczby. Można to zrobić chociażby korzystając z małego twierdzenia Fermata, czyli podnosząc liczbę do potęgi p – 2 (gdzie p jest liczbą pierwszą określającą operację modulo), w tym przypadku będzie to 10 005. Następnie, mnożąc drugi element pary przez tak otrzymaną liczbę, osoba A otrzyma wyjściowy kod. W naszym przypadku, drugi element z pary należy przemnożyć przez pierwszy element podniesiony do potęgi

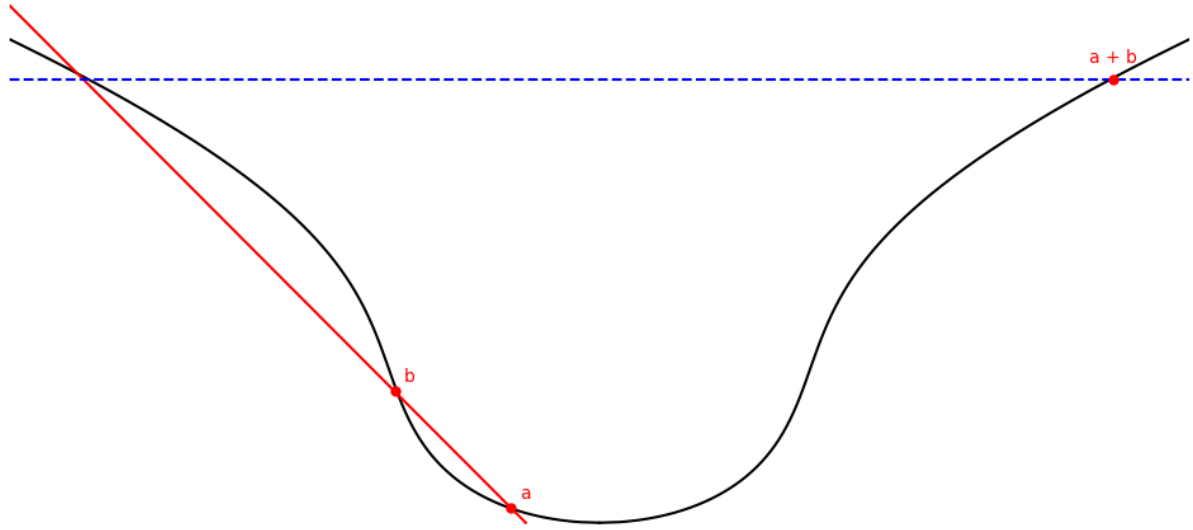
345 * 10 005 (jako liczony rzeczywisty). Ostatecznie otrzymamy wyjściowy kod 23-25-4-18-1, czyli łatwo odtworzyć tajną wiadomość „wydra”.

Należy podkreślić, że w praktyce przyjmuje się znacznie większe liczby niż 10 007, a kodowanie liter na liczby odbywa się za pomocą standardowych metod jak np. kodowanie ASCII.

Algorytm ElGamala wyjaśnia też, dlaczego administrator serwisu, do którego logowanie wymaga podania ustalonego przez użytkownika hasła nie jest w stanie tego hasła poznać. Hasło użytkownika jest po prostu jego kluczem prywatnym.

Przedstawiony algorytm zapewnia wysokie bezpieczeństwo, ale mimo wszystko nowoczesne metody szyfrowania wykorzystują jeszcze bezpieczniejsze rozwiązania. Są one osiągalne np. poprzez skomplikowanie działania grupowego. Jak pokazano wyżej, działania na okręgu są trudniejsze niż działania na prostej i to zapewnia bezpieczeństwo. Algebra jednak zna jeszcze trudniejsze struktury. Działania na tzw. krzywych eliptycznych są bardziej skomplikowane i zapewniają jeszcze większe bezpieczeństwo.

Wynikiem działania a + b na krzywej eliptycznej będzie punkt symetryczny do trzeciego punktu przecięcia prostej która przecina krzywą w punktach a i b, czyli tak jak pokazano na rysunku poniżej.



W tym miejscu warto skomentować problem tzw. hipotezy Riemanna, która zakłada, że liczby pierwsze są ułożone w logicznej kolejności, więc łatwo odnaleźć. Istnieje pewna plotka, że udowodnienie hipotezy Riemanna sprawi, że świat jaki znamy się skończy, bo wówczas łatwe odnajdywanie liczb pierwszych sprawi, że wszystkie współczesne systemy kryptograficzne będzie można łatwo obejść. Chciałbym zwrócić uwagę na dwa problemy tej obawy.

Po pierwsze łatwo odnajdywanie liczb pierwszych nie ma żadnego znaczenia dla algorytmu ElGamala, bo w nim i tak jawnie podaje się używaną liczbę pierwszą. To nie ta liczba stanowi o bezpieczeństwie algorytmu. Więc chociażby w tym sensie, chociażby systemy oparte na pomyśle ElGamala nie są zagrożone przez hipotezę Riemanna.

Po drugie, hipoteza Riemanna jest jednak tylko hipotezą, czyli nie istnieje ani dowód na jej prawdziwość, ani na jej fałszywość. Oznacza to, że jak dotąd wszystkie próby jej zastosowania dostarczały przykładów przemawiających za jej prawdziwością. Gdyby tak nie było, to otrzymalibyśmy tzw. kontrprzykład, który byłby dowodem na fałszywość hipotezy. Stąd, gdyby jakiś hacker chciał wykorzystać hipotezę Riemanna do obejścia systemu kryptograficznego, to mógłbym to po prostu zrobić zakładając, że jest prawdziwa. Jak widać świat, nie mieliśmy w ostatnim czasie żadnej rewolucji, więc to oznacza, że nasze współczesne systemy są w miarę przygotowane na udowodnienie hipotezy Riemanna. Jej dowód byłby na pewno niesamowitym wydarzeniem naukowym, bardzo rewolucyjnym. Jednak nie spodziewałbym się, że zakończy świat jaki znamy.

Czy możemy zatem być spokojni wysyłając pikantne wiadomości? Ależ oczywiście, ale pod warunkiem, że dobrze pilnujemy swojego klucza prywatnego, czyli w większości przypadków, naszego hasła.