Technika ślepych podpisów

Ślepe podpisy są jedną z podstawowych technik współczesnej kryptografii. Dzięki nim osoba podpisująca dokument nie widzi jego treści. Jeżeli więc klient banku sam utworzył banknot z niepowtarzalnym numerem seryjnym, to bank otrzymuje ten banknot do podpisu zgodnie z protokołem ślepych podpisów. Nie widzi więc numeru seryjnego banknotu i nie jest później w stanie przyporządkować klientowi jego wydatków.

Do generowania ślepych podpisów w dużej mierze wykorzystuje się asymetryczny algorytm RSA, który służy do tworzenia zwykłych podpisów cyfrowych. Dzięki wykorzystaniu własności arytmetyki modularnej bank podpisuje ciąg, który dopiero przy znajomości pewnego klucza może zostać przetworzony na oryginalną, podpisywaną wiadomość. Algorytm został tak utworzony, aby przy podpisywaniu zamaskowanego ciągu podpis był jednocześnie prawdziwy dla oryginalnej wiadomości.

Banknoty cyfrowe tym różnią się od prawdziwych, że mogą opiewać praktycznie na dowolną sumę i nie są uzależnione od ściśle określonych nominałów. Może się więc zdarzyć, że klient podsunie bankowi do podpisu banknot opiewający na sumę 1000 zł, a powie, że przysyła banknot na 100 zł. Ponieważ bank zgodnie z protokołem ślepych podpisów nie widzi treści banknotu, nie może sprawdzić prawdziwości danych podanych przez klienta.

Problem ten ma dwa popularne rozwiązania. Firma realizująca system eCash wybrała rozwiązanie prostsze, polegające na wystawianiu jedynie monet elektronicznych, czyli banknotów o ściśle określonej sumie, np. 1 dolar. System ten jest mało wydajny, bo przy zakupach za 1000 dolarów trzeba zaopatrzyć się w 1000 monet. Zważywszy że szyfrowanie asymetryczne nie jest szybkie, a używane liczby są duże, zarówno przesyłanie danych, jak i wykonywanie obliczeń jest czasochłonne.

Bardziej wyrafinowany sposób rozwiązania tego problemu umożliwia generowanie banknotów o dowolnym nominale. W tym celu klient generuje aż 100 banknotów o tym samym nominale. Wszystkie wysyłane są do banku zakryte, jak w protokole ślepych podpisów. Bank wybiera jeden z nich i nie odkrywając go, żąda ujawnienia pozostałych. Jeżeli wszystkie odkryte banknoty faktycznie opiewają na podaną sumę, bank może z dużym prawdopodobieństwem założyć, że wybrany banknot ma również odpowiedni nominał. Może go więc podpisać, mając praktycznie pewność, że klient nie oszukuje.

Podstawową cechą ślepego podpisu jest więc rozdzielenie dwóch czynności, które w przypadku zwykłego podpisu cyfrowego występują razem. Podmiot podpisujący potwierdza autentyczność określonego obiektu kryptograficznego, ale nie poznaje informacji, którą ostatecznie swoim podpisem uwierzytelnia. Po zakończeniu procedury odbiorca dysponuje prawidłowym podpisem możliwym do zweryfikowania przy użyciu klucza publicznego podpisującego. Jednocześnie podpisujący nie powinien być w stanie powiązać później przedstawionego dokumentu z konkretną operacją jego podpisania. To właśnie ta cecha sprawia, że ślepe podpisy znalazły zastosowanie przede wszystkim w rozwiązaniach, w których trzeba pogodzić możliwość potwierdzenia autentyczności z ochroną anonimowości użytkownika.

W klasycznym rozwiązaniu opartym na RSA klient rozpoczyna od przygotowania wiadomości, która ma zostać podpisana. Może nią być na przykład ciąg danych reprezentujący monetę elektroniczną wraz z jej numerem seryjnym. Następnie wybiera losową wartość, określaną jako czynnik zaślepiający. Przy użyciu klucza publicznego banku oraz działań wykonywanych modulo liczba stanowiąca moduł RSA klient tworzy z wiadomości postać zamaskowaną. Z punktu widzenia banku otrzymany ciąg nie ujawnia treści właściwej wiadomości.

Bank wykonuje na zamaskowanym ciągu operację przy użyciu swojego klucza prywatnego. Rezultat przekazuje klientowi.

Klient wykorzystuje następnie znany tylko sobie czynnik zaślepiający, a dokładniej jego odwrotność modularną, dzięki czemu usuwa maskowanie. Wynikiem jest podpis odnoszący się już do właściwej wiadomości, mimo że bank podczas operacji podpisywania nie poznał jej treści. Współczesne konstrukcje RSA blind signatures zachowują właśnie ten ogólny schemat: przygotowanie wiadomości, jej zaślepienie, wykonanie podpisu przez serwer oraz finalizację polegającą na odślepieniu i sprawdzeniu poprawności podpisu.

Najważniejszą własnością takiego protokołu jest ślepość, czyli niemożność ustalenia przez podpisującego, którą z później przedstawionych wiadomości podpisał podczas konkretnej interakcji z klientem. W elektronicznej gotówce pozwala to odtworzyć jedną z podstawowych cech tradycyjnej gotówki. Bank może potwierdzić, że moneta elektroniczna została przez niego prawidłowo wystawiona, ale nie powinien móc stwierdzić, który klient później się nią posłużył. Jeżeli użytkownik wypłaca elektroniczną monetę, a następnie płaci nią sprzedawcy, bank otrzymujący tę monetę do rozliczenia nie powinien być w stanie połączyć jej z wcześniejszą operacją wypłaty.

Właściwość ta różni elektroniczną gotówkę od wielu typowych systemów płatniczych. W przypadku zwykłej płatności elektronicznej operator systemu posiada zazwyczaj dane pozwalające powiązać płatnika, odbiorcę, kwotę i moment transakcji. Ślepy podpis umożliwia natomiast zaprojektowanie systemu, w którym bank potwierdza ważność środka płatniczego, nie otrzymując jednocześnie kompletnej informacji pozwalającej śledzić sposób jego późniejszego wykorzystania.

Anonimowość nie może jednak prowadzić do możliwości samodzielnego tworzenia pieniędzy przez klienta. Drugą podstawową własnością bezpiecznego systemu ślepych podpisów jest więc odporność na fałszowanie. Użytkownik, który uzyskał od banku podpis pod jedną określoną liczbą monet, nie powinien być w stanie wykorzystać przeprowadzonych operacji do utworzenia większej liczby prawidłowych podpisów. W

nowoczesnych analizach bezpieczeństwa ślepych podpisów zwraca się zatem uwagę nie tylko na samą ślepość, ale również na odporność na uzyskiwanie dodatkowych podpisów poza tymi, do których użytkownik był uprawniony.

Jednym z problemów wynikających bezpośrednio ze ślepoty jest właśnie kontrolowanie treści podpisywanej wiadomości. Bank nie może bezkrytycznie wykonywać operacji prywatnym kluczem na dowolnych danych przekazywanych przez klienta. W przeciwnym wypadku klient mógłby próbować wykorzystać bank jako tak zwaną wyrocznię podpisującą i uzyskiwać podpisy pod nieprawidłowo skonstruowanymi obiektami. Dlatego protokół musi zapewniać nie tylko ukrycie treści, ale również możliwość sprawdzenia, że podpisywany obiekt posiada właściwą strukturę.

Klasyczną odpowiedzią na ten problem jest metoda *cut-and-choose*. Jej logika jest stosunkowo prosta. Klient nie tworzy jednego kandydata do podpisania, lecz wiele równoważnych kandydatów. Wszystkie zostają zamaskowane i przesłane bankowi. Bank wybiera losowo jeden z nich, natomiast w odniesieniu do pozostałych żąda ujawnienia danych potrzebnych do sprawdzenia ich prawidłowości. Jeżeli sprawdzone egzemplarze zostały przygotowane zgodnie z regułami, zwiększa się prawdopodobieństwo, że również pozostawiony w ukryciu egzemplarz jest prawidłowy.

Jeżeli przykładowo klient przygotował sto obiektów, a bank losowo wybrał jeden i polecił otworzyć pozostałe dziewięćdziesiąt dziewięć, próba oszustwa polegająca na przygotowaniu tylko jednego nieprawidłowego egzemplarza ma niewielką szansę powodzenia. Oszust musi liczyć na to, że właśnie wadliwy egzemplarz będzie tym jedynym, którego bank nie każe ujawnić. Im większa liczba kandydatów podlegających losowemu sprawdzeniu, tym mniejsze prawdopodobieństwo skutecznego oszustwa. Ceną za zwiększenie bezpieczeństwa jest jednak większa liczba obliczeń i większa ilość przesyłanych danych.

Właśnie z tego powodu technika *cut-and-choose*, mimo swojej przejrzystości, nie jest rozwiązaniem idealnym dla dużych i wydajnych systemów. Generowanie wielu obiektów tylko po to, aby zdecydowana większość została następnie odrzucona, powoduje znaczny narzut obliczeniowy i komunikacyjny. W literaturze dotyczącej elektronicznej gotówki wielokrotnie wskazywano nieefektywność systemów opartych na rozbudowanych procedurach tego rodzaju. Późniejsze konstrukcje kryptograficzne poszukiwały więc sposobów zapewnienia podobnych gwarancji przy mniejszej liczbie operacji.

Jednym z takich rozwinięć są podpisy częściowo ślepe. W klasycznym ślepym podpisie podpisujący nie powinien znać treści podpisywanej wiadomości. W podpisie częściowo ślepym strony mogą natomiast uzgodnić, że pewna część informacji pozostanie jawna. Pozwala to na przykład bankowi znać nominał elektronicznej monety albo termin jej ważności, podczas gdy niepowtarzalny numer służący do identyfikacji samej monety pozostaje ukryty. Rozwiązanie takie może znacznie ułatwić projektowanie praktycznego systemu płatności.

W przypadku nominału możliwe są również inne rozwiązania. Bank może na przykład stosować różne klucze podpisujące dla różnych wartości monet. Jeden klucz służy wówczas do wystawiania monet o wartości jednego złotego, inny monet dziesięciozłotowych, a jeszcze inny monet o większej wartości. Sam fakt wykorzystania określonego klucza pozwala określić nominał, natomiast ukryta pozostaje pozostała treść monety. Takie podejście eliminuje możliwość przedstawienia bankowi monety o większej wartości jako monety o wartości mniejszej, choć równocześnie komplikuje zarządzanie większą liczbą kluczy. Literatura dotycząca elektronicznej gotówki wskazuje właśnie stosowanie różnych kluczy dla poszczególnych nominałów jako jedno z klasycznych rozwiązań tego problemu.

Odrębnym i bardzo ważnym problemem elektronicznej gotówki jest możliwość wielokrotnego wydania tej samej monety. Fizycznego banknotu po przekazaniu sprzedawcy nie posiada już kupujący.

Dane cyfrowe można natomiast kopiować praktycznie bez ograniczeń. Osoba posiadająca plik reprezentujący monetę elektroniczną może więc próbować przesłać identyczny plik kilku sprzedawcom.

Najprostszy sposób zapobiegania takiemu zjawisku polega na wykorzystaniu unikalnego numeru seryjnego każdej monety. Kiedy sprzedawca otrzymuje elektroniczną gotówkę, bank sprawdza, czy dany numer nie znajduje się już w bazie monet wykorzystanych. Jeżeli numer pojawia się po raz pierwszy, płatność może zostać zaakceptowana i numer trafia do bazy. Jeżeli ten sam numer został już wcześniej przedstawiony do rozliczenia, bank stwierdza próbę ponownego wydania tej samej monety.

Takie rozwiązanie wymaga jednak kontaktu z bankiem podczas każdej transakcji albo przynajmniej przed jej ostatecznym zaakceptowaniem. Jest to system typu *online*. Zapewnia stosunkowo prostą ochronę przed podwójnym wydawaniem, ale powoduje konieczność stałej dostępności infrastruktury bankowej. Rozwiązania typu *offline* próbowały pozwolić sprzedawcy przyjąć monetę bez natychmiastowego połączenia z bankiem. W takim przypadku nie zawsze da się zapobiec podwójnemu wydaniu w chwili transakcji, dlatego mechanizmy kryptograficzne projektowano tak, aby możliwe było późniejsze wykrycie oszustwa i w określonych konstrukcjach ustalenie tożsamości osoby, która wykorzystała tę samą monetę więcej niż raz. Klasyczne systemy e-cash wykorzystujące *cut-and-choose* stosowały tę technikę właśnie również w mechanizmach związanych z wykrywaniem podwójnego wydawania.

Istotą dobrze zaprojektowanego systemu jest pogodzenie dwóch z pozoru sprzecznych celów. Uczciwy użytkownik powinien zachować anonimowość, a bank nie powinien mieć możliwości odtworzenia historii jego zakupów. Jednocześnie system nie może zapewniać pełnej bezkarności osobie próbującej wielokrotnie wykorzystać tę samą monetę. W niektórych konstrukcjach informacje identyfikujące właściciela są więc kryptograficznie zakodowane w monecie w taki sposób, że pojedyncza prawidłowa transakcja

ich nie ujawnia, natomiast zestawienie danych pochodzących z dwóch różnych prób wydania tej samej monety pozwala wykryć oszusta.

Ślepe podpisy mają zastosowania znacznie szersze niż sama elektroniczna gotówka. Podobny problem pojawia się na przykład w elektronicznym głosowaniu. System powinien potwierdzić, że osoba oddająca głos jest uprawniona do udziału w wyborach i nie głosuje wielokrotnie, ale jednocześnie nie powinien umożliwiać powiązania konkretnego głosu z konkretnym wyborcą. W takim modelu odpowiednia instytucja może potwierdzić uprawnienie do głosowania bez poznawania treści później wykorzystanej karty wyborczej.

Technika ta może być używana również w systemach anonimowych poświadczeń, tokenów dostępowych i innych mechanizmach ochrony prywatności. Współczesne specyfikacje wskazują zastosowania ślepych podpisów nie tylko w płatnościach, ale także w mechanizmach uwierzytelniania i rozwiązaniach pozwalających oddzielić proces przyznania określonego uprawnienia od jego późniejszego wykorzystania.

Warto jednocześnie zauważyć, że współczesne implementacje nie powinny wykorzystywać wprost najprostszego „podręcznikowego” RSA. Historyczny opis ślepego podpisu jest bardzo przydatny do zrozumienia mechanizmu algebraicznego, ale praktyczne protokoły wymagają odpowiedniego kodowania wiadomości, bezpiecznego generowania liczb losowych oraz ochrony przed atakami wynikającymi z błędnej implementacji. W aktualnej specyfikacji RSA Blind Signatures wykorzystywany jest między innymi mechanizm RSA-PSS, a końcowy podpis może zostać zweryfikowany przy użyciu standardowej procedury weryfikacji RSA-PSS.

Bardzo ważna jest jakość losowego czynnika zaślepiającego. Jeżeli wartości losowe są przewidywalne, wielokrotnie używane albo generowane w niewłaściwy sposób, anonimowość użytkownika może zostać osłabiona. Bezpieczeństwo ślepego podpisu zależy

więc nie tylko od matematycznych własności RSA, lecz również od prawidłowej implementacji całego protokołu. Podobnie jak w innych rozwiązaniach kryptograficznych, silny algorytm może okazać się nieskuteczny, jeżeli zostanie niewłaściwie zastosowany.

Trzeba również pamiętać, że anonimowość zapewniana przez sam podpis nie rozwiązuje wszystkich problemów prywatności. Nawet jeżeli bank nie potrafi matematycznie powiązać podpisanej monety z operacją jej wypłaty, użytkownik może zostać rozpoznany na podstawie innych informacji, na przykład adresu sieciowego, czasu wykonania transakcji, sposobu komunikowania się z systemem czy danych przekazanych sprzedawcy. Pełny system anonimowych płatności musi więc chronić nie tylko treść kryptograficzną transakcji, ale również informacje dodatkowe powstające podczas komunikacji.

Technika ślepych podpisów jest zatem interesującym przykładem rozwiązania kryptograficznego, które umożliwia jednoczesne potwierdzenie autentyczności i ograniczenie wiedzy podmiotu dokonującego podpisu. Bank może zagwarantować, że określony cyfrowy obiekt jest przez niego autoryzowany, nie uzyskując przy tym informacji pozwalających śledzić jego późniejsze wykorzystanie. W tradycyjnym podpisie cyfrowym podpisujący zna dokument, który zatwierdza. W podpisie ślepym relacja ta zostaje celowo przerwana.

Największą trudnością nie jest więc samo wykonanie podpisu, lecz stworzenie całego protokołu gwarantującego, że anonimowość nie będzie wykorzystywana do oszustwa. Konieczne jest kontrolowanie poprawności podpisywanych obiektów, uniemożliwienie tworzenia nieautoryzowanych monet oraz rozwiązanie problemu podwójnego wydawania. Techniki *cut-and-choose*, różne klucze dla różnych nominałów, podpisy częściowo ślepe oraz bardziej zaawansowane protokoły kryptograficzne są różnymi sposobami rozwiązania tych problemów.

Ślepy podpis pokazuje tym samym jedną z najważniejszych

właściwości współczesnej kryptografii: możliwość udowodnienia określonego faktu bez ujawniania wszystkich informacji związanych z tym faktem. Użytkownik może otrzymać potwierdzenie banku, nie pokazując bankowi danych, które pozwoliłyby później śledzić jego transakcję. Z tego względu koncepcja zaproponowana pierwotnie dla anonimowej elektronicznej gotówki stała się podstawą znacznie szerszej grupy rozwiązań służących ochronie prywatności w systemach informatycznych.

Jeśli szukają Państwo pomocy w napisaniu własnej pracy - potrzebują Państwo fachowych konsultacji to polecamy stronę pisanie prac - profesjonalna pomoc w pisaniu prac w granicach prawa.