Marek R. OGIELA, Piotr SUŁKOWSKI
AGH – AKADEMIA GÓRNICZO-HUTNICZA W KRAKOWIE
Al. Mickiewicza 30, 30-059 Kraków

Wykrywanie wielokrotnych nieprawidłowych transakcji w protokołach wymiany elektronicznej gotówki

Prof. dr hab. Marek R. OGIELA

Profesor zwyczajny na Akademii Górniczo – Hutniczej w Krakowie. Prowadzi badania nad kognitywnymi systemami informacyjnymi nowej generacji, a także kryptografią i podziałem sekretów. Jest członkiem wielu renomowanych towarzystw naukowych, a także autorem ponad 270 publikacji o zasięgu międzynarodowym.

e-mail: mogiela@agh.edu.pl

Mgr inż. Piotr SUŁKOWSKI

Absolwent kierunku informatyka stosowana na AGH z roku 2013. Specjalista w zakresie kryptograficznych protokołów wymiany elektronicznej gotówki. Ponadto interesuje się bezpieczeństwem systemów oraz architekturą oprogramowania. Jest pracownikiem Asseco Poland S.A. na stanowisku programisty.

e-mail: piotr.sulkowski@o2.pl

Streszczenie

W pracy zostały opisane protokoły wymiany elektronicznej gotówki. Dla standardowego algorytmu wykorzystywanego w celu umożliwienia realizacji takich transakcji zostanie zaprezentowany możliwy atak, który po wielokrotnym wydaniu banknotu, umożliwia oszukiwanie zarówno ze strony sprzedawców jak i klienta. Głównym tematem pracy jest zaprezentowanie autorskiego protokołu, który pozwala zapobiec podobnym atakom i oszustwom, a tym samym pozwala na realizację nieodwracalnych płatności elektroniczną gotówką realizowanych w trybie off-line.

Słowa kluczowe: elektroniczna gotówka, anonimowe transakcje, protokoły wymiany elektronicznych pieniędzy.

Detection of repeated counterfeit transactions in protocols for exchanging electronic coins

Abstract

This paper presents an improvement to the well-known protocol by David Chaum for anonymous currency exchange. We show its vulnerability to serious frauds by both parties: the client and the sellers, after an electronic coin is spent at least twice. In this case, the sellers can cooperate and pretend that the client spent the same coin many times. The system cannot successfully determine how many times the client spent the coin and how many times the seller faked the transaction. Therefore, the bank is not able to charge the real abuser. As a consequence of this limitation the original system cannot guarantee that all valid checks will be cashed. This leads to the conclusion that it cannot be securely used for irreversible off-line transactions, although it was originally designed to that purpose. In this paper we show the algorithm which enables sellers to cheat on other parties of the protocol. We also propose an improvement to this system that allows overcoming this vulnerability by making all transactions signed by one-time secret client keys.

Keywords: electronic cash, anonymous transaction, protocols for exchanging electronic coins.

1. Wstęp

W dzisiejszych czasach pomimo istnienia bardzo rozwiniętej infrastruktury bankowości internetowej oraz kart płatniczych, nie ma możliwości całkowicie swobodnego przesyłania środków pieniężnych w sposób gwarantujący anonimowość. Oznacza to, że każda transakcja elektroniczna nie może być anonimowa i zawarta w całkowitej tajemnicy, tak jak często przebiega to z wykorzystaniem normalnej gotówki. Z wielu przyczyn wymiana normalnej gotówki nie jest jednak wygodnym sposobem, dlatego też bardzo przydatna byłaby możliwość anonimowego przekazu pieniędzy za pośrednictwem Internetu. Istnieją co najmniej dwie możliwości skonstruowania takiego systemu. Jedną z nich jest stworzenie wirtualnej waluty. Przykładem takiego systemu jest sieć Bitcoin [1]. Drugą możliwością jest utworzenie systemu elektronicznej gotówki. W rozwiązaniu tym w przekazie środków pieniężnych pośredniczy bank, a klienci wymieniają się jedynie anonimowymi czekami o określonej wartości. Utworzenie takiego protokołu nie jest trywialne między innymi z uwagi na łatwość skopiowania elektronicznie zapisanych danych.

Warto zaznaczyć, że pierwszym sposobem który umożliwił anonimową wymianę czeków jest protokół wymyślony przez Davida Chauma i opisany w pracy [2]. Pomimo iż został on opracowany ponad 20 lat temu oraz doczekał się wielu udoskonaleń i modyfikacji, to do dzisiaj jest on podstawowym rozwiązaniem, a znaczna część stworzonych później protokołów opiera się właśnie na nim [3, 4, 5, 6, 7, 8]. Głównym tematem niniejszej publikacji będzie przedstawienie ataku na protokół Davida Chauma, oraz propozycja jego modyfikacji, która pozwoli wyeliminować próby oszustwa w tym protokole. W dalszej części pracy zostaną scharakteryzowane własności systemów wymiany elektronicznej gotówki, zostanie także opisany mechanizm działania podstawowej wersji protokołu Davida Chauma, a następnie zostanie przedstawiona autorska propozycja rozszerzenia tego protokołu, która uniemożliwi dokonanie oszustwa przez klienta lub sprzedawców, podczas wielokrotnych transakcji przeprowadzanych w trybie off-line.

2. Schemat działania anonimowych transakcji

W systemie wymiany elektronicznej gotówki płatność przebiega w trzech etapach.

W pierwszym klient kontaktuje się z bankiem i pobiera elektroniczny czek. Następnie klient kontaktuje się ze sprzedawcą i wydaje u niego pobrany z banku czek. W trzecim etapie sprzedawca kontaktuje się z bankiem, gdzie przedstawia poświadczenie zawarcia transakcji z klientem, w zamian za co otrzymuje jego równowartość pieniężną.

Taki system wymiany elektronicznej gotówki zakłada, że biorące w nim udział trzy strony tj. bank, klient oraz sprzedawca są całkowicie od siebie niezależne. System musi zatem gwarantować każdej ze stron, że nie zostanie oszukana, nawet w przypadku złej woli oraz współpracy dwóch pozostałych stron. Ponadto, każdy z trzech etapów tj. tworzenia wydawania oraz realizacji banknotów może być wykonywany oddzielnie. Każdy z tych etapów wymaga zatem innego, specjalnego protokołu. Banknoty, które posiada klient po zakończeniu komunikacji z bankiem nazywane są elektroniczną gotówką ze względu na gwarantowaną anonimowość, tak jak w przypadku papierowej gotówki. Po wydaniu ich w drugim etapie sprzedawca jest w posiadaniu poświadczeń transakcji, które przedstawia do banku w dowolnym momencie po zakończeniu komunikacji z klientem.

Przy tak realizowanych protokołach transakcyjnych głównym problemem jest to, aby bank nie był w stanie skojarzyć przedstawionych mu przez sprzedawcę poświadczeń transakcji z czekiem, który wystawił klientowi.

Jednym z głównych wymagań stawianych wszystkim systemom związanym z pieniędzmi jest bezpieczeństwo ich funkcjonowania. Każda ze stron powinna mieć gwarancję, że nie zostanie oszukana. Przez bezpieczeństwo należy także rozumieć brak możliwości kradzieży środków innym użytkownikom systemu, jak również gwarancję zachowania anonimowości transakcji. Główne cechy systemów przekazywania elektronicznej gotówki są następujące:

- anonimowość - system musi gwarantować, że transakcje nie zostaną wyśledzone przez bank,
- proces realizacji musi nastąpić podczas wydawania banknotów (on-line) lub może mieć miejsce w późniejszym czasie (systemy off-line),
- możliwość odwoływania transakcji.

Wymagania dotyczące dostępu do danych. Istnieją systemy, które wymagają, aby użytkownicy posiadali banknoty zapisane wyłącznie na specjalnie zabezpieczonych kartach, przechowujących dane o banknotach, ale nie umożliwiających ich odczytywania przez nieuprawnione osoby.

3. Podstawowy protokół wymiany elektronicznej gotówki

Pierwszy i jednocześnie najbardziej znany sposób realizacji systemu elektronicznej gotówki, który spełnia wszystkie postawione powyżej założenia zaproponował David Chaum w pracy [2]. Schemat ten stał się referencyjnym przykładem realizacji wymiany elektronicznej gotówki przedstawianym np. w [7, 8, 9, 10]. W rozwiązaniu tym banknot składa się z podpisanego n elementowego ciągu par P_i ($i \in 1,2,\dots n$) o następującej strukturze:

$$P_i = (h(a_i, c_i), h(a_i \oplus u, d_i)),$$

gdzie:

u - unikalny identyfikator klienta, znany także bankowi,
a_i - liczba losowo wybrana przez klienta w celu ukrycia wartości u,
c_i, d_i - liczby wybrane losowo przez klienta w celu utworzenia funkcji haszującej z hasłem.

Liczba n jest pewną stałą w systemie i stanowi o jego bezpieczeństwie (kosztem ew. wydajności). Aby otrzymać taki banknot elektroniczny klient komunikuje się z bankiem według następującego protokołu:

1. klient losuje $2 * n$ par P_i oraz odpowiadających im współczynników zaciemniających r_i;
2. klient przesyła do banku wszystkie zaciemnione pary $r_i^e * h(P_i)$ (liczba e to część publiczna klucza banku);
3. bank wybiera n spośród nadesłanych par i prosi klienta o przesłanie odpowiadających im wartości r_i, a_i, c_i oraz d_i;
4. klient przesyła do banku żądane liczby;
5. bank sprawdza ich poprawność z przesłanymi wcześniej zaciemnionymi skrótami;
6. bank podpisuje wszystkie pozostałe pary, mnoży je przez siebie i odsyła klientowi;
7. pomniejsza także stan konta klienta o wartość banknotu;
8. klient odbiera, a następnie zdejmuje zaciemnienie z przesłanych par.

Klient w punkcie 6. otrzymuje iloczyn podpisów wszystkich wybranych par:

$$I = \prod (r_i^e * h(P_i))^d \bmod n, i \in L,$$

gdzie:

e - publiczna eksponenta klucza banku,
d - prywatna eksponenta klucza banku,
n - moduł podpisu banku,
L - zbiór indeksów banknotów wybranych do podpisu przez bank.

Poprzez podzielenie tej wartości przez kolejne czynniki zaciemniające r_i klient otrzyma podpisany banknot:

$$Z = I * \prod r_i^{-1} \bmod n, i \in L,$$

a zatem:

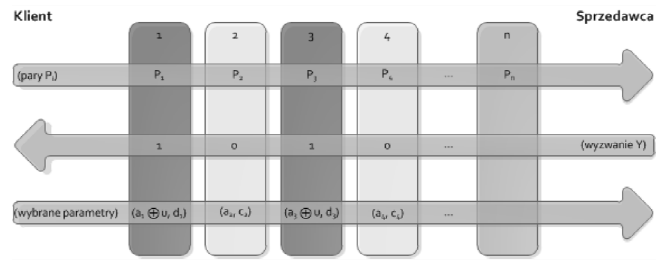
$$Z = \prod h(P_i)^d \bmod n, i \in L.$$

Na tym etapie klient posiada podpisany banknot, składający się z n par P_i oraz podpisu Z poświadczającego te pary. Należy także zapamiętać wszystkie wartości współczynników a_i, b_i, c_i oraz d_i.

W kolejnym etapie aby dokonać płatności klient prezentuje sprzedawcy wszystkie pary P_i oraz związany z nimi podpis banku Z. Sprzedawca po sprawdzeniu poprawności banknotu oraz jego zgodności z podpisem tworzy tzw. wyzwanie Y. Jest to ciąg zer i jedynek o długości n, z których część jest stała i przypisana do sprzedawcy, a część jest losowana. Następnie wysyła je klientowi. Odpowiednio dla każdego elementu tego ciągu klient odsyła sprzedawcy:

A. wartości a_i oraz c_i – w przypadku gdy i-ty element ciągu Y wynosi 0,
B. wartości $a_i \oplus u$ oraz d_i – w przypadku gdy i-ty element ciągu Y wynosi 1.

Odpowiedź klienta na wyzwanie ilustruje rys. 1. Sprzedawca na bieżąco sprawdza zgodność przesłanych wartości z ich skrótami przesłanymi wcześniej przez klienta w postaci elementów par P_i. Jeśli wszystko się zgadza, płatność zostaje przyjęta.



Rys. 1. Odpowiedź klienta na wyzwanie sprzedawcy
Fig. 1. Customer response to the challenge of the seller

Na tym etapie sprzedawca jest w posiadaniu ciągu par P_i, odpowiadającego im podpisu Z, a także dla każdej pary P_i wartości (a_i, c_i) lub ($a_i \oplus u$, d_i). Dane te nazywać będziemy poświadczeniem transakcji. Aby spieniężyć otrzymane poświadczenia sprzedawca kontaktuje się z bankiem i przedstawia mu wszystkie otrzymane od klienta dane oraz wygenerowane w trakcie sprzedaży wyzwanie Y. Bank sprawdza poprawność banknotu, podpisu cyfrowego, a także czy wartości (a_i, c_i) oraz ($a_i \oplus u$, d_i) odpowiadają swoim skrótom w P_i. Jeśli któraś z wartości jest niepoprawna, wina leży po stronie sprzedawcy, ponieważ miał on możliwość samodzielnie sprawdzić poprawność banknotu podczas dokonywania płatności przez klienta. Jeśli banknot jest poprawny, bank sprawdza w swojej bazie danych, czy taki sam banknot nie został już kiedykolwiek użyty. Jeśli nie, to wpłaca na konto sprzedawcy odpowiednią, ustaloną z góry kwotę. Jeżeli natomiast banknot taki znajduje się już w bazie danych, to bank próbuje ustalić kto jest winny tej sytuacji.

Sprzedawca ma możliwość skopiowania poświadczenia transakcji, ale wszystkie kopie będą poprawne tylko dla jednego wyzwania Y. Przy każdej transakcji jest ono losowane, a sprzedawca otrzymuje inne poświadczenie. Jeśli więc przedstawi dwa takie same, to z dużym prawdopodobieństwem można stwierdzić, że to on próbuje popełnić nadużycie. Jeśli poświadczenia są różne, oznacza to, że klient użył dwa razy tego samego banknotu w różnych transakcjach.

Jeśli banknot został użyty dwukrotnie, istnieje duże prawdopodobieństwo, że dwa wylosowane w tych procesach wyzwania – odpowiednio Y_1 i Y_2 – różnią się co najmniej jednym elementem.

Jeśli tak, to istnieje co najmniej jedna taka para P_x, dla której znamy wartości zarówno a_x jak i $a_x \oplus u$. Bank jest zatem w stanie obliczyć wartość u, która jest unikalnym identyfikatorem klienta. Nie ma jednak możliwości poznania tego identyfikatora w przypadku, gdy klient zapłacił banknotem tylko raz. Bank ma wówczas do dyspozycji zawsze tylko wartość a_i albo $a_i \oplus u$. Wartość a_i nie niesie ze sobą żadnych wiadomości identyfikujących klienta. Sama wartość $a_i \oplus u$ także nie zawiera żadnej informacji, ponieważ przyjmując, że a_i jest całkowicie losowe, liczba ta także staje się całkowicie losowa.

David Chaum w pracy [2] podaje możliwość wprowadzenia dodatkowej modyfikacji protokołu, która umożliwi bankowi dowodzenie wielokrotnego użycia banknotu przez klienta. W przedstawionym powyżej schemacie bank ma możliwość wygenerowania dowolnego poświadczenia w imieniu klienta, bez jego wiedzy i zgody. Najprostszym sposobem, aby odebrać bankowi taką możliwość jest zmuszenie klienta do podpisywania się na otrzymanych banknotach. Bank nie będzie wtedy w stanie utworzyć banknotu bez udziału klienta, przez co po odkryciu tożsamości oszukującego klienta możliwe będzie udowodnienie mu tego oszustwa.

Aby umożliwić taką funkcjonalność w miejsce wartości unikalnego identyfikatora klienta u zostanie dodana dodatkowa, losowo wybrana przez klienta liczba z_i. Banknot składa się więc z n par P_i postaci:

$$P_i = (h(a_i, c_i), h(\, a_i \oplus (u \parallel z_i), d_i)),$$

gdzie operator $\parallel$ oznacza połączenie liczb w taki sposób, aby dało się odczytać każdą z nich oddzielnie (np. poprzez zapisanie obok siebie dwóch liczb o ustalonej długości bitów).

Proces wydawania banknotu dodatkowo zacznie się od przesłania przez klienta podpisanej listy skrótów wszystkich liczb z_i. Bank otrzyma więc liczbę:

$$Z = h(z_1), h(z_2), \ldots, h(z_n),$$

oraz podpis $C_K(Z)$. W późniejszym etapie klient przysyła właściwe zaciemnione banknoty. Bank wybiera połowę z nich i sprawdza, czy są poprawnie zbudowane. Otrzymuje wtedy od klienta także odpowiednie liczby z_i. Po wydaniu banknotu bank posiada połowę zobowiązań z_i oraz skróty wszystkich tych zobowiązań wraz z podpisami. Gdy klient użyje banknotu więcej niż raz, nie tylko ujawniona zostanie jego tożsamość, ale także co najmniej jeden nowy współczynnik z_i. Przedstawienie przez bank więcej niż połowy podpisanych zobowiązań z_i uznaje się za dowód wielokrotnego użycia banknotu przez klienta, ponieważ bank nie jest w stanie samodzielnie spreparować takich zobowiązań.

4. Analiza własności systemu Davida Chauma

Podstawowe własności przedstawionego systemu, to:

1. możliwość dokonywania transakcji bez potrzeby kontaktu z bankiem w chwili jej dokonywania,
2. brak konieczności użycia specjalnych urządzeń ani kart typu „tamper-proof",
3. zapewnienie anonimowości transakcji, w przypadku gdy klient postępuje uczciwie,
4. zagwarantowanie bezpieczeństwa każdej ze stron nawet w przypadku, gdy pozostałe dwie działają w zmowie przeciwko niej.

Oprócz wymienionych cech mogą wystąpić również pewne zagrożenia dla stron korzystających z tego systemu. Główne z nich to: wielokrotne wydawanie jednego banknotu, tworzenie fałszywych banknotów oraz utrata anonimowości klienta.

Rozważmy zatem przypadek, w którym klient będzie chciał wielokrotnie wydać banknot. W takim przypadku ryzyko banku oraz sprzedawcy w dużej mierze uzależnione jest od przyjętej strategii postepowania. Istnieją co najmniej dwie różne możliwości:

A. za każdy prawidłowy przedstawiony przez sprzedawcę banknot, bank wypłaca środki, nawet w przypadku, gdy klient wydał go wiele razy. To on zobowiązany będzie do zapłacenia za wszystkie wykonane transakcje;
B. w przypadku wykrycia, że banknot został użyty kilkakrotnie środki nie zostaną wpłacone na konto sprzedawcy. Otrzyma on jedynie dane osobowe nieuczciwego klienta. To sprzedawca zmuszony będzie wówczas do odwołania transakcji oraz ewentualnego dochodzenia odszkodowania.

Przyjęcie strategii B spowoduje, że całe ryzyko będzie spoczywało na sprzedawcy i jeśli poniesie on jakiekolwiek koszty związane z nieuczciwą transakcją, to sam zobowiązany będzie do obciążenia nimi klienta. Oczywiście tego typu odpowiedzialność sprzedawcy skłaniają do przyjęcia w praktycznych rozwiązaniach strategii A i przeniesienie odpowiedzialności za nieuczciwych klientów na bank.

Niestety i w takim przypadku można sobie wyobrazić następujący scenariusz w którym klient może skutecznie próbować oszukiwać.

Klient z zamiarem dokonania oszustwa wydaje banknot dokładnie dwa razy. Poszkodowani sprzedawcy mogą teraz w tajemnicy wymienić się uzyskanymi informacjami. Zakładając, że ich wyzwania Y_1 i Y_2 różnią się na m bitach mogą oni wspólnie wygenerować aż 2^m różnych kombinacji poświadczeń transakcji. Wystarczy, że połączą część jednego poświadczenia z częścią drugiego. W ten sposób powstaje nowe poświadczenie transakcji. W tej sytuacji sprzedawcy posiadają poświadczenia transakcji, które nigdy się nie odbyły. Mogą oni następnie zgłaszać się do banku twierdząc, że zostali wielokrotnie oszukani, a bank nie jest w stanie stwierdzić ile razy faktycznie oszukał klient, a ile razy sprzedawcy. Bank nie może więc ustalić jaką kwotę naprawdę wydał klient, a co za tym idzie nie może domagać się od niego odszkodowania. Sytuacja ta wyklucza zatem możliwość przyjęcia przez bank strategii gwarantowanych wypłat w obecnym systemie. Bank zmuszony jest ograniczyć się jedynie do zapewnienia, że poda tożsamość nieuczciwych klientów. W przypadku większości nadużyć, prawdopodobnie będzie możliwe obciążenie nieuczciwego klienta podwójną kwotą transakcji, jednak należy podkreślić, że używając tylko tej metody nigdy nie ma takiej gwarancji. Sprzedawca nie może więc polegać na poprawności samego banknotu, dopóki nie zrealizuje go w banku.

Jak zatem wynika z przedstawionego przykładu system Chauma zdolny jest tylko do przeprowadzania w pełni odwracalnych transakcji, bez gwarancji uzyskania odszkodowania w przypadku oszustwa.

5. Proponowana modyfikacja umożliwiająca wykrywanie wielokrotnych transakcji

Aby dodać do systemu możliwość zawierania nieodwracalnych transakcji off-line konieczne jest wprowadzenie możliwości dowodzenia wszystkich zawartych transakcji. Umożliwi to dochodzenie odszkodowań w przypadku wielokrotnego wydawania banknotu.

W tym też celu konieczna jest zmiana poświadczeń wydawanych przez klientów w taki sposób, aby sprzedawcy nie byli w stanie wygenerować nowych na podstawie dowolnej ilości już posiadanych.

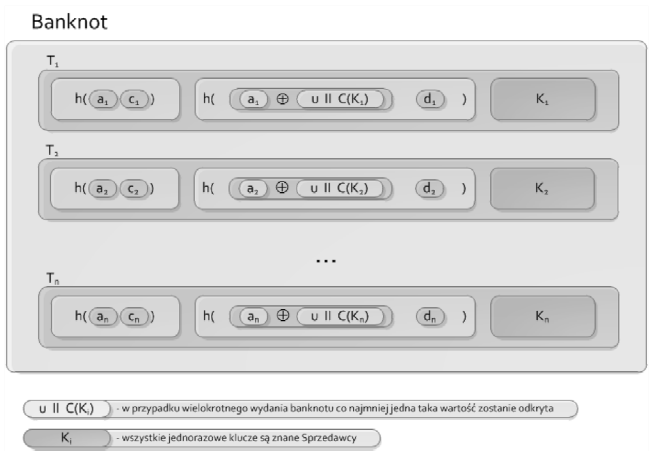
Proponujemy zatem taką modyfikację protokołu, w której klienci podpisują wszystkie otrzymane od sprzedawców wyzwania i przesyłają je razem z poświadczeniami. Jeśli jednak do podpisu użyją własnego klucza, to wówczas utracą swoją anonimowość. Konieczne jest zatem użycie jednorazowych kluczy, które należy powiązać w jakiś sposób z prawdziwym kluczem klienta. Jednym ze sposobów jest dołączenie go do banknotu, wraz z jego certyfikatem podpisanym przez klienta (budowa banknotu została przedstawiona na rys. 2.). Zamiast par P_i, klient będzie wówczas przesyłał trójki:

$$T_i = (h(a_i, c_i), h(a_i \oplus (u \parallel C(K_i)), d_i), K_i),$$

gdzie:

a_i, c_i, d_i - losowe liczby wybrane przez klienta
u - unikalny identyfikator klienta
K - jednorazowy klucz publiczny RSA utworzony przez klienta. Można go zapisać np. jako $(e \parallel n)$, gdzie e oznacza publiczną eksponentę tego klucza, natomiast n jego moduł
C - certyfikat klienta potwierdzający prawidłowość danego klucza. Może stanowić go np. liczba $h(K)^d \bmod m$, gdzie liczby (d, m) stanowią prywatną część klucza RSA opublikowanego przez klienta (tzw. klucza głównego)

Przed przystąpieniem do tworzenia banknotów wymagane jest, aby klienci zarejestrowali w banku swoje główne klucze publiczne [10, 11]. Najlepiej jeśli klienci korzystać będą z kluczy certyfikowanych przez pewien urząd certyfikacji. Rejestracja kluczy następuje tylko raz dla każdego klienta, podczas tworzenia konta.



Rys. 2. Zmodyfikowany banknot cyfrowy
Fig. 2. Modified digital note

Protokół tworzenia banknotu, w którym uczestniczy bank oraz klient wygląda podobnie jak w poprzedniej wersji. Różnicę stanowi sam banknot, na który składa się n trójek T_i, a nie jak w poprzednim przypadku par P_i. Klient wysyła do banku $2*n$ zaciemnionych banknotów, z czego bank wybiera połowę i prosi klienta o zdjęcie z nich zaciemnienia. Po otrzymaniu od klienta wszystkich potrzebnych współczynników bank, tak jak poprzednio, sprawdza poprawność banknotów. Dodatkowo musi on upewnić się czy wszystkie certyfikaty $C(K_i)$ zawarte w przesłanych banknotach są poprawnymi certyfikatami kluczy K_i, tzn. dotyczą klucza K_i zawartego w dalszej części banknotu oraz są podpisane głównym kluczem klienta (który bank ma w swojej bazie danych). Jeśli wszystko się zgadza, to wówczas bank odsyła klientowi podpisany banknot dokładnie w ten sam sposób, co w poprzedniej wersji protokołu. Klient po usunięciu zaciemnienia posiada więc liczbę:

$$Z = \prod h(T_i)^d \bmod n, i \in L,$$

gdzie:

d - prywatna eksponenta klucza banku
n - moduł podpisu banku
L - zbiór indeksów banknotów wybranych do podpisu przez bank

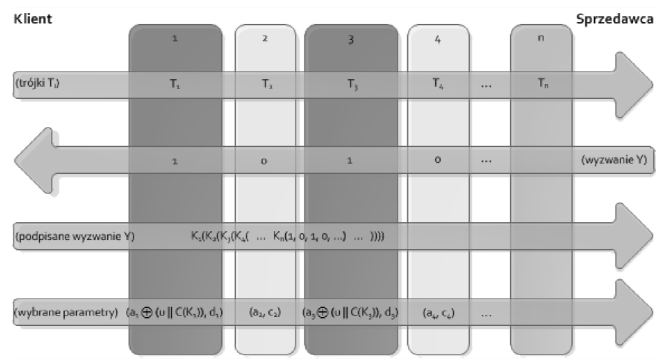
Istota całego usprawnienia polega na tym, aby każda transakcja przeprowadzona przez klienta pozostawiała unikalny i niemożliwy do podrobienia ślad. Aby osiągnąć taką funkcjonalność protokołu, klient podpisuje wyzwanie przesłane mu przez sprzedawcę używając kluczy K_i zawartych w banknotach. Protokół wymiany banknotu pomiędzy klientem a sprzedawcą (przedstawiony na rys. 3) wygląda zatem następująco:

1. klient przesyła podpisany przez bank banknot Z;
2. klient przesyła także trójki T_i (czyli wartości $h(a_i, c_i)$, $h(a_i \oplus (u \parallel C(K_i)), d_i)$ oraz K_i);
3. sprzedawca sprawdza, czy banknot Z jest poprawnym podpisem przesłanych trójek;
4. sprzedawca przesyła klientowi wyzwanie Y;
5. klient odsyła sprzedawcy wartość wyzwania Y podpisaną wszystkimi kluczami jednorazowymi K_i:

$$R = K_1(K_2(\ldots K_n(Y)\ldots)),$$

gdzie: $K_i(x)$ - podpis wartości x przy użyciu klucza K_i

6. sprzedawca weryfikuje poprawność podpisu R;
7. klient przesyła sprzedawcy odpowiednio wartości (a_i, c_i) lub $(a_i \oplus (u \parallel C(K_i)), d_i)$ w zależności od wartości wyzwania Y na i-tym bicie (analogicznie jak w poprzedniej wersji protokołu);
8. sprzedawca sprawdza czy przesłane dane odpowiadają skrótom znajdującym się w trójkach T_i. Jeśli wszystko się zgadza płatność zostaje przyjęta.



Rys. 3. Odpowiedź klienta na wyzwanie sprzedawcy w protokole zaproponowanym przez autorów
Fig. 3. Customer response to the challenge of the seller in the protocol proposed by the authors

W celu realizacji banknotu, sprzedawca w dowolnym momencie przedstawia bankowi podpisany banknot Z, ciąg trójek T_i, wygenerowane wyzwanie Y wraz z podpisem R oraz wszystkimi przesłanymi przez klienta wartościami zależnymi od tego wyzwania. bank jest w stanie sprawdzić poprawność wszystkich danych w ten sam sposób, co sprzedawca podczas wymiany banknotu z klientem.

Tak samo, jak w przypadku standardowej wersji protokołu, po jednokrotnym wydaniu banknotu klient ujawnia tylko połówkę z każdego zobowiązania. Jednocześnie w przypadku wydania co najmniej dwa razy tego samego banknotu z dużym prawdopodobieństwem bank wejdzie w posiadanie dwóch komplementarnych połówek, jednak dzięki modyfikacji pozna on nie tylko tożsamość klienta, ale także co najmniej jeden z certyfikatów $C(K_i)$.

W ten sposób bank zyskuje możliwość udowodnienia klientom wielokrotnego wydawania banknotów. Każda transakcja zostaje bowiem podpisana przez klienta wszystkimi kluczami K_i. W momencie oszustwa bank oprócz tożsamości klienta ma do dyspozycji co najmniej jeden zestaw zawierający zobowiązanie podpisane pewnym jednorazowym kluczem oraz certyfikat klienta poświadczający ten klucz. Jeżeli bank jest w stanie przedstawić klientowi podpis R danego wyzwania Y oraz wykazać mu, że podpis należy do niego, transakcję taką można uznać za udowodnioną. Nikt oprócz klienta nie potrafi bowiem stworzyć certyfikatu dla klucza K_i, którym podpisano wyzwanie. Nikt także nie jest w stanie podrobić tego podpisu, jako że banknot zawiera jedynie publiczną jego część.

Jeśli klient wydaje banknoty tylko raz, wówczas nie ma możliwości poznania ani jego identyfikatora u, ani żadnego z certyfikatów $C(K_i)$. Certyfikat taki w połączeniu z kluczem K_i mógłby być

także wykorzystany do znalezienia tożsamości klienta. Wystarczy, aby bank spróbował zweryfikować taki certyfikat używając wszystkich głównych kluczy klientów jakie posiada w swojej bazie. Jeden z nich byłby zapewne poprawny. Na tej podstawie tożsamość klienta mogłaby zostać odkryta, dlatego certyfikaty muszą także pozostać tajne, aż do momentu wystąpienia oszustwa. Jawny jest natomiast sam klucz K_i. Jest on tworzony przez klienta w sposób losowy i nie niesie ze sobą żadnej informacji, która mogłaby go zidentyfikować.

Po wprowadzeniu takiego rozwiązania bank jest w stanie wykazać ile dokładnie wydał klient. Co za tym idzie (zakładając że jest w stanie uzyskać tę należność od klienta przez skuteczną windykację), może on przyjąć strategię wypłacania środków wszystkim sprzedawcom, którzy przedstawią poprawne poświadczenia transakcji. Dzięki temu system umożliwia zawieranie transakcji off-line.

Wciąż pozostaje jednak problem kradzieży banknotów. Jeśli banknot znajdzie się w posiadaniu niepowołanej osoby, może on zostać wykorzystany do stworzenia nieograniczonego debetu na koncie właściciela. Aby temu przeciwdziałać klient po stwierdzeniu kradzieży może zgłosić ten fakt do banku, aby ten opublikował listę nieważnych banknotów. Ponadto sam bank po wykryciu dwukrotnej płatności może opublikować taki banknot jako kradziony. Jeśli jednak założymy, że transakcje są przeprowadzane bez kontaktu z bankiem, nigdy nie jesteśmy w stanie całkowicie wyeliminować tego problemu. Taką samą trudność można napotkać także w samym schemacie podpisu cyfrowego. Jeśli za każdym razem przed przystąpieniem do przyjęcia podpisu cyfrowego sprawdzamy w publicznej bazie danych czy nie został ukradziony, wówczas wyeliminujemy ten problem. Jeśli jednak postanowimy skonstruować system przyjmujący podpisy „off-line" – bez kontaktu z publiczną bazą danych skradzionych podpisów – nigdy nie możemy mieć pewności, że podpis nie został skradziony.

Jeśli jednak ryzyko kradzieży uznane zostanie za zbyt duże, zawsze istnieje możliwość powrócenia do strategii zawierania wyłącznie odwracalnych transakcji. Proponowana modyfikacja, poza kwestiami wydajnościowymi, w żadnym stopniu nie osłabia pierwotnego systemu.

6. Podsumowanie

System przedstawiony przez Davida Chauma w [2] jest pierwszym systemem, który był zdolny do realizacji anonimowych transakcji w trybie off-line i stanowił przełomowe odkrycie w dziedzinie elektronicznych i anonimowych płatności. Po co najmniej dwukrotnym wydaniu tego samego banknotu przez klienta, system ten staje się jednak podatny na atak zarówno ze strony oszukanych sprzedawców, jak i samego klienta. Słabość ta powoduje, że nie może on służyć do zawierania transakcji nieodwracalnych. Ten sam problem dotyczy także innych systemów opartych na podobnym protokole. Wprowadzenie zaproponowanej w niniejszej publikacji modyfikacji pozwala zapobiec podobnym atakom i oszustwom.

W obecnej chwili nie istnieje szeroko rozpowszechniony system elektronicznej gotówki oparty na schemacie elektronicznych czeków, jednak nic nie stoi na przeszkodzie, aby zbudować taki system. Konieczny jest dodatkowy narzut obliczeń przy każdej transakcji, a także odpowiednio większe zasoby pamięci, które umożliwią przez dłuższy czas przechowywanie danych dot. zużytych banknotów. Rozwiązania opisane w pracy [6] umożliwiają znaczną redukcję ilości koniecznych zasobów.

Wydaje się, że prędzej czy później technologia elektronicznej gotówki stanie się bardziej popularna i zacznie wypierać tradycyjne karty kredytowe oraz przelewy bankowe. Jest to bardzo prawdopodobne, ponieważ w przeciwnym wypadku banki wejdą w posiadanie ogromnej ilości poufnych informacji o swoich klientach. Zachowanie tajemnicy jest natomiast kluczowe dla bezpieczeństwa i rozwoju wielu firm, dlatego chętnie skorzystają one z tego nowego rozwiązania, aby zniwelować ryzyko utraty danych.

7. Literatura

[1] Strona fundacji Bitcoin zajmującej się rozwojem wirtualnej waluty o tej samej nazwie: http://bitcoin.org
[2] Chaum D., Fiat A., Naor M.: Untraceable Electronic Cash. CRYPTO '88 Proceedings on Advances in cryptology, pp. 319-327, Springer-Verlag, 1990.
[3] Brands S.: Untraceable Off-line Cash in Wallets with Observers. CRYPTO '93 Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology, pp. 302-318, Springer-Verlag, 1994.
[4] Brands S.: Off-Line Electronic Cash Based on Secret-Key Certificates. Lecture Notes in Computer Science, Vol. 911, pp. 131-166, Springer-Verlag, 1995.
[5] Deng R. H., Han Y., Jeng A.B., Ngair T.: A new on-line cash check scheme. Proceedings of the 4th ACM conference on Computer and communications security, pp. 111-116, ACM, 1997.
[6] Ferguson N.: Single term off-line coins. EUROCRYPT '93 Workshop on the theory and application of cryptographic techniques on Advances in cryptology, pp. 318-328, Springer-Verlag, 1994.
[7] Kim S., Oh H.: A new electronic check system with reusable refunds. International Journal of Information Security, nr 1/3, pp. 175-188, Springer-Verlag, 2002.
[8] Menezes A. J., van Oorschot P. C., Vanstone S. A.: Kryptografia stosowana. WNT, 2005.
[9] Goldwasser S., Bellare M.: Lecture Notes on Cryptography, Cambridge, 2008.
[10] Mao W.: Blind Certification of Public Keys and Off-line Electronic Cash. Hawlett-Packard Laboratories, 1996.
[11] Schneier B.: Kryptografia dla praktyków. Protokoły, algorytmy i programy źródłowe w języku C. WNT, 1995.

otrzymano / received: 30.11.2013
przyjęto do druku / accepted: 03.03.2014 *artykuł recenzowany / revised paper*

INFORMACJE

Bezpłatny dostęp do artykułów opublikowanych w PAK

Realizując idee Open Access przez miesięcznik PAK informujemy, że artykuły opublikowane w PAK są dostępne w wersji elektronicznej. Artykuły w łatwy sposób można znaleźć korzystając z wyszukiwarki artykułów. Bazę artykułów można przeszukać po nazwisku autora, tytule artykułu lub po słowach kluczowych.

Tadeusz SKUBIS
Redaktor naczelny Wydawnictwa PAK