

Wykład 11. Elektroniczna gotówka

Wykładowca: Stefan Dziembowski

Skryba: Jakub Królikowski, Jacek Kołodziej

Streszczenie. Wykład omawia problem bezpieczeństwa transakcji zakupów elektroniczną gotówką przy użyciu kart kredytowych. Przedstawiono w nim pomysły i konkretne protokoły kryptograficzne zapewniające bezpieczeństwo pewnych aspektów takich transakcji.

1 Wstęp

Omawiać będziemy elektroniczną gotówkę. Dla ustalenia uwagi przyjmujemy, że mamy 3 uczestników: Bank B , Klienta K (który ma konto w Banku) i Sprzedawcę (który sprzedaje coś Klientowi i chce za to uzyskać pieniądze z Banku - dla uproszczenia przyjmijmy, że on też ma konto w banku B).

Obecnie podstawową metodą płacenia przez internet jest użycie kart kredytowych. Zasada jest prosta: K zgłasza się ze swoją kartą kredytową do S , podając mu numer karty oraz swoje dane personalne. S prosi B o przelanie określonej sumy pieniędzy na swoje konto, podając dane karty kredytowej K .

Ma to następujące wady w porównaniu z pieniędzmi papierowymi:

1. opiera się w silnym stopniu na zaufaniu (sprzedawca może użyć numeru karty klienta w nieuczciwych celach).
2. nie zapewnia anonimowości klienta (bank wie gdzie klient zrobił zakupy).
3. ze względu na niskie bezpieczeństwo koszt utrzymania karty jest dość wysoki, bo wymaga wprowadzenia wysokiego ubezpieczenia

Punkt 2 jest istotniejszy niż to się na pozór wydaje, gdyż klienci zagrożeni są *dyskryminacją cenową* (ang.: *price discrimination*) (patrz np. [2]). Niektórym firmom bardzo zależy na tym, żeby wiedzieć z jak zamożnym klientem mają do czynienia. Przy oferowanych usługach proponują wówczas odpowiednie ceny, tak żeby możliwie dobrze zarobić.

W wykładzie przedstawione zostaną metody płatności nie pozwalające bankowi na śledzenie płatności dokonywanych przez klienta. Inaczej mówiąc wprowadzimy

protokoły próbujące emulować papierową gotówkę - płacenie papierowymi pieniędzmi jest anonimowe i nie wymaga zaufania do sprzedawcy. Różnica, w której leży problem, polega na tym, że pieniądze elektroniczne nie są zmaterializowane i są przekazywane drogą elektroniczną, a nie fizycznie.

Należy zwrócić uwagę na to, że przy wszystkich swoich zaletach anonimowe protokoły gotówki elektronicznej mają pewne wady:

1. umożliwiają płacenie za towary niezgodne z prawem (np. pornografia dziecięca)
2. pozbawiają władze państwowe kontroli nad przepływem pieniędzy (zagrożenia: nielegalny transfer za granicę, pranie brudnych pieniędzy)
3. ułatwiają pracę szantażystom - to właściwie jest jeden z aspektów, konkretny przykład poprzedniego punktu; schwytanie szantażysty, bardzo często odbywa się w momencie przekazywania okupu, istnieje ryzyko, że przy anonimowej transakcji szantażysta mógłby bez przeszkód odebrać pieniądze.

Dlatego wprowadzono *uczciwe* (ang.: *fair*) protokoły elektronicznej gotówki, w których anonimowość może zostać zniesiona za (każdorazową) zgodą *zaufanej strony trzeciej*, ustalonego sędziego, którym może być np. urzędnik sądowy, albo wielu urzędników.

2 Wymagania

Przedstawmy więc wymagania wobec bezpiecznej elektronicznej gotówki:

1. trudne fałszerstwo - nie da się łatwo wygenerować elektronicznych pieniędzy
2. duplikacja jest niemożliwa lub może być wykryta - Klient nie może dwa razy wydać tych samych elektronicznych monet. Zupełne wykluczenie duplikacji jest dość trudnym problemem (jeśli w momencie płacenia nie ma łączności on-line z bankiem, nie da się sprawdzić czy identyczny „pieniądz” nie został przed chwilą wydany w innym sklepie), dlatego raczej podchodzi się do tego w taki sposób, że duplikację się dopuszcza (Klient ma możliwość dwukrotnego wydania tych samych pieniędzy), ale sytuacja taka jest natychmiast wykrywana i działanie klienta jest przerywane - nieuczciwa próba wydania pieniędzy kończy się np. aresztem.
3. zachowana jest anonimowość
4. jak najmniej operacji on-line - chcemy, żeby wszystkie strony transakcji komunikowały się ze sobą jak najmniej, np. żeby w momencie zakupu Klient ani Sprzedawca nie musieli się komunikować z bankiem

3 Piersza próba

Protokoły transakcji elektroniczną gotówką można podzielić na trzy części:

1. protokół pobrania pieniędzy przez Klienta K z Banku B
2. protokół zapłacenia elektroniczną gotówką przez K Sprzedawcy S
3. protokół ściągnięcia pieniędzy przez S z Banku B

W pierwszym pomysle przyjmujemy, że Bank używa klucza prywatnego SK_B do zaszyfrowania polecenia pobrania pieniędzy. Klient zakodowane polecenie sprawdza publicznym kluczem PK_B . Oznaczmy przez $\{M\}_{SK}$ polecenie M zaszyfrowane kluczem SK .

• Pobranie pieniędzy

1. K prosi B o banknot 100 zł.
2. B odpowiada wiadomością:

$$\{(To\ ja,\ banknot\ 100\ zł,\ \#4527)\}_{SK_B}$$

gdzie $\#4527$ to nr seryjny banknotu i zdejmuje 100 zł z konta K

3. K sprawdza szyfr i przyjmuje pieniądze.

• Zapłacenie pieniędzmi

1. K płaci S gotówką elektroniczną
2. S sprawdza szyfrowanie i przyjmuje pieniądze.

• Ściągnięcie pieniędzy

1. S przekazuje pieniądze do B
2. B sprawdza szyfrowanie i wypłaca pieniądze S

Ten prosty protokół nie spełnia jednak dwóch wymagań:

- występuje brak anonimowości - kiedy Sprzedawca ściąga z Banku pieniądze, Bank wie, kto nimi zapłacił.
- możliwa jest duplikacja

Dlatego wprowadzono:

4 Ślepe podpisy cyfrowe

Nieformalnie mówiąc *Ślepe podpisy cyfrowe* to takie w których podpisujący nie jest świadomy co podpisuje. Obrazowo rzecz ujmując, można sobie wyobrazić, że do Banku trafia od Klienta zalakowana koperta z czekiem, a Bank ją podpisuje, nie znając do końca jej zawartości. Wie tylko, że koperta jest od Klienta i ma obietnicę Klienta na jaką sumę opiewa czek, nie wie natomiast jak dokładnie wygląda czek, który Klient umieścił w kopercie i czy Klient nie kłamie. Klient mając podpisane przez Bank pieniądze, może nimi kupować, a Bank może sprawdzić podpis, kiedy zgłosi się do niego Sprzedawca. Nie jest jednak w stanie powiązać otrzymanych od Sprzedawcy pieniędzy z Klientem, bo nie widział ich wcześniej.

W pierwszej chwili może nie być jasne jaki jest pożytek z takich podpisów. Wyjaśnimy to niebawem. Na razie pokażemy implementację.

4.1 Ślepe podpisy RSA

Pomysł opiera się na tym, że Klient wraz z poleceniem pobrania gotówki wysyła losową liczbę, numer seryjny, ale tak, że Bank go nie poznaje. Do szyfrowania używamy RSA.

Oto pierwsza, naistotniejsza część protokołu:

- **Pobranie pieniędzy:** $((e, n), (d, n))$ - klucz publiczny i prywatny Banku)

1. K losuje liczbę $r \bmod n$.
2. K oblicza $M' = M * r^e \bmod n$
3. K przekazuje M' do B
4. B zwraca zaszyfrowaną M' , powiedzmy $s' = (M')^d \bmod n$. Zauważmy, że:

$$s' = (M')^d = M^d * (r^e)^d = M^d * r$$

5. B zdejmuje 100 zł z konta K
6. jako, że K zna r może podzielić s' przez r i dostanie:

$$s = s' * r^{-1} = M^d$$

Protokół daje anonimowość Klienta, ale pozwala mu teraz oszukiwać przy poleceniu pobrania gotówki - Bank nie wie czy w kopercie jest rzeczywiście 100 zł. Poza tym protokół ten nie rozwiązuje problemu duplikacji.

4.2 Strategia „dziel i wybieraj”

W przedstawionym do tej pory protokole wzbogaconym o ślepe podpisy, możliwy jest scenariusz w którym klient K przychodzi do Banku, mówi że chce wypłacić 100 zł, po czym podaje Bankowi do podpisania banknot o nominale 10000 zł. Bank do tej pory nie był w stanie wykryć takiego oszustwa, co wynika bezpośrednio ze „ślepości” ślepych podpisów RSA. Problem ten można rozwiązać na dwa sposoby.

Pierwszym sposobem jest używanie osobnej pary kluczy do podpisywania każdego z nominałów. Wówczas Sprzedawca musiałby posiadać tyle kluczy publicznych Banku ile jest nominałów, i przy przyjmowaniu gotówki od klienta, S sprawdzałby czy banknot jest podpisany kluczem przynależnym danemu nominałowi. Podobnie robiłby Bank przyjmując gotówkę od Sprzedawcy.

W powyższym rozwiązaniu obowiązek sprawdzania że Klient nie oszukał przy pobieraniu pieniędzy z Banku spada na Sprzedawcę. Znacznie wygodniejsza byłaby metoda, w której Bank już przy wykonywaniu ślepego podpisu na banknocie upewniał by się że Klient zachował się uczciwie. Aby uzyskać taką funkcjonalność, zmienimy protokół pobierania gotówki z Banku w sposób następujący (strategia „dziel i wybieraj”):

1. Klient tworzy 100 banknotów o nominale 20 zł.
2. Klient wkłada każdy banknot do osobnej koperty (ślepe podpisy RSA).
3. Klient przesyła 100 kopert do Banku.
4. Bank otwiera losowe 99 kopert (żąda od Klienta ujawnienia losowości r użytej w protokole ślepego podpisu) i sprawdza czy rzeczywiście były w nich banknoty o nominale 20 zł.
5. Jeśli nie wykryte zostało oszustwo, Bank podpisuje jenną kopertę której nie otworzył w poprzednim kroku i wysyła ją do Kliena.

Przy takiej metodzie weryfikacji prawdopodobieństwo, że Klientowi uda się oszukać Bank wynosi $\frac{1}{100}$. Należy zatem zapewnić że każda wykryta próba oszustwa Banku będzie się wiązała z dużymi sankcjami (np. z karą pozbawienia wolności). Za oszustwo uznaje się sytuację w której Klient odmawia otworzenia koperty (wysłania liczby r użytej do jej zamknięcia), bądź otwarta koperta zawiera banknot o nominale innym niż deklarowany przez Klienta.

5 Zapobieganie kopiowaniu „banknotów”

Istotnym problemem elektronicznej gotówki jest łatwość kopiowania jej. Kluczowym bowiem zabezpieczeniem papierowych banknotów jest trudność ich powielania. W przypadku gotówki elektronicznej, kopiowanie sprowadza się do kopiowania ciągów bitów. Dlatego też należy wprowadzić zabezpieczenie przed dwukrotnym (wielokrotnym) użyciem tego samego „banknotu”.

5.1 Transakcje on-line

Najprostszym rozwiązaniem problemu duplikowania pieniędzy jest wprowadzenie konieczności łączenia się z Bankiem przy każdej wymianie pieniędzy. Wówczas podczas płacenia Sprzedawca wysyła użyty banknot do Banku. Bank sprawdza czy dany banknot nie był użyty wcześniej. Jeśli wszystko jest w porządku, Bank dodaje odpowiednią kwotę do stanu rachunku Sprzedawcy i zapamiętuje banknot. Jeśli Klient zapłaci dwa razy tym samym banknotem, to przy drugiej próbie płatności transakcja nie powiedzie się.

Metoda ta ma jedną podstawową wadę - wymaga stałej łączności z Bankiem. Nie jest to zatem rozwiązanie mogące symulować „fizyczną” gotówkę.

5.2 Transakcje off-line

Przy założeniu braku łączności z Bankiem w momencie płacenia pieniędzmi, nie jesteśmy w stanie *zapobiegać* podwójnemu wydawaniu tych samych banknotów, gdyż nie dysponujemy dostateczną ilością informacji. Możemy natomiast *wykrywać* oszustwa i ich sprawców.

Rozważmy naiwny protokół off-line w którym Klient podczas płacenia przesyła Sprzedawcy elektroniczną gotówkę, a ten ją prosto zapamiętuje. Sprzedawca co jakiś czas (np. raz dziennie) łączy się z Bankiem i przesyła mu zebrane od klientów banknoty w celu zwiększenia stanu własnego konta. Problem który teraz powstaje to odróżnienie oszustwa Klienta od oszustwa Sprzedawcy. Jeśli bowiem przesłany przez Sprzedawcę banknot pojawił się już wcześniej, to nie wiadomo czy Klient dwa razy zapłacił tym samym banknotem, czy może Sprzedawca dwa razy podsunął go Bankowi. Ponadto nawet gdybyśmy wiedzieli że to Klient oszukał, nie znamy jego tożsamości - sami bowiem wprowadzając ślepe podpisy zabiegaliśmy o uzyskanie anonimowości.

Zajmiemy się teraz rozróżnianiem który z dwóch scenariuszy oszustwa miał miejsce i ewentualną identyfikacją nieuczciwego Klienta.

5.2.1 Random Identity String (RIS)

RIS jest napisem generowanym podczas operacji płacenia danym banknotem. Ma on następujące własności:

- jest generowany przez Klienta w momencie płacenia.
- tylko Klient może wygenerować RIS dla własnego banknotu.
- jest różny przy każdej płatności tym samym banknotem.
- posiadanie dwóch RIS'ów dla jednego banknotu pozwala na uzyskanie identyfikatora Klienta (znika anonimowość Klienta).

Dysponując narzędziem o takich właściwościach, jesteśmy w stanie skutecznie wykrywać oszustwa (oraz poznawać tożsamość oszusta).

Zakładamy że przy każdej płatności jest generowany RIS. Każdy Sprzedawca wraz z banknotem który otrzymał od Klienta, przechowuje odpowiadający mu RIS. Gdy Sprzedawca zgłasza się do Banku, przesyła banknoty z ich RIS'ami. Jeśli Bank po raz drugi dostanie ten sam banknot z identycznymi RIS'ami to oznacza to że Sprzedawca skopiował banknot. Jeśli zaś do banknotów dołączone są inne RIS'y, oznacza to że Klient posłużył się tym samym banknotem dwa razy. Wówczas jednak na podstawie dwóch różnych RIS'ow jesteśmy w stanie poznać tożsamość oszusta (wynika to z zamieszczonej wyżej specyfikacji).

6 Ostateczna postać protokołu

Przedstawimy teraz ostateczną postać protokołu wymiany elektronicznej gotówki.

• Pobranie pieniędzy

1. Klient zgłasza Bankowi chęć otrzymania banknotu 20 zł
2. Klient przygotowuje banknoty M_i dla $i = 1, 2, \dots, 100$ postaci:

$$M_i = (\text{„To ja, banknot 20 zł”}, \text{losowość}(i), y_{i,1}, y'_{i,1}, \dots, y_{i,Q}, y'_{i,Q})$$

gdzie $\text{losowość}(i)$ jest losowym identyfikatorem banknotu różnym dla poszczególnych M_i , $y_{i,t} = H(x_{i,t})$, $y'_{i,t} = H(x'_{i,t})$ gdzie $x_{i,t}$ i $x'_{i,t}$ są losowo wybranymi liczbami takimi że:

$$x_{i,t} \oplus x'_{i,t} = \text{Id}_{\text{Klient}}$$

H jest „jednokierunkową” funkcją haszującą (czyli taką, że trudno dla niej znaleźć takie dwa elementy p i q że $H(p) = H(q)$).

3. Klient wysyła do Banku M'_i dla $i = 1, 2, \dots, 100$ utworzone z M_i zgodnie z protokołem ślepych podpisów RSA (punkt 4.1). Dla każdego banknotu jest używana inna losowość r_i służąca oślepianiu.
4. Bank wybiera 99 ze 100 banknotów i pyta Klienta o odpowiadające im losowości r_k oraz wartości $x_{k,1}, x'_{k,1}, \dots, x_{k,q}, x'_{k,q}$.
5. Bank sprawdza czy klient przysłał poprawne wartości r_i , czy zakodowane przy ich pomocy banknoty mają nominominały 20 zł, czy $H(x_{i,j}) = y_{i,j}$ oraz $H(x'_{i,j}) = y'_{i,j}$, a także czy $x_{i,j} \oplus x'_{i,j} = \text{Id}_{\text{Klient}}$
6. Bank bierze pozostały banknot (niech to będzie M'_{34}) który nie został otwarty i podpisuje go i wysyła do Klienta.
7. Bank zmniejsza stan konta Klienta o 20 zł.
8. Klient na podstawie podpisanego M'_{34} odtwarza podpisany M_{34} (patrz protokół ślepych podpisów).
9. Klient sprawdza czy uzyskany banknot jest poprawnie podpisany przez Bank.

Dla uproszczenia oznaczeń porzucimy teraz indeks podpisanego przez Bank klucza i składowe klucza przyjmą następujące oznaczenia:

$$M = (\text{„To ja, banknot 20 zł”}, \text{losowość}, y_1, y'_1, \dots, y_Q, y'_Q)$$

• Płacenie pieniędzmi

1. Klient płaci Sprzedawcy gotówką elektroniczną.
2. Sprzedawca odkodowuje banknot kluczem publicznym Banku.
3. Dla każdej pary y_j, y'_j (gdzie $j = 1, 2, \dots, Q$) Sprzedawca każe ujawnić Klientowi jedną z liczb: x_j lub x'_j , przy czym to Sprzedawca ustala (losowo) którą z nich. W sumie Sprzedawca dostaje Q liczb.
4. Sprzedawca sprawdza czy $H(x_j) = y_j$ (lub odpowiednio $H(x'_j) = y'_j$ jeśli poprosił o ujawnienie x'_j).
5. Sprzedawca oprócz banknotu zapamiętuje także RIS = (z_1, z_2, \dots, z_Q) gdzie z_j jest otrzymaną od Klienta liczbą x_j lub x'_j .

• Ściągnięcie pieniędzy

1. Sprzedawca przekazuje Bankowi banknot wraz z odpowiadającym mu RIS'em.

2. Bank sprawdza czy banknot jest podpisany jego kluczem prywatnym oraz czy identyczny banknot nie był już wcześniej użyty.
3. Jeśli Bank stwierdzi że banknot był użyty był już wcześniej, to jeśli odpowiadające tym transakcjom RIS'y się różnią to istnieje takie k że w jednym RIS'ie znajduje się x_k a w drugim x'_k . Licząc $x_k \oplus x'_k$ dostajemy identyfikator nieuczciwego klienta. Jeśli RIS'y są identyczne, wnioskujemy że oszukującym jest Sprzedawca.

Jeśli Klient chciałby oszukać i nie zostać zidentyfikowanym, musiałby w obydwu transakcjach być pytany przez Sprzedawcę o dokładnie te same x_k lub x'_k dla każdego $k \in \{1, 2, \dots, Q\}$. Zatem prawdopodobieństwo udanego oszustwa wynosi 2^{-Q} .¹

7 Materialne banknoty wspomagane kryptografią

Ciekawym pomysłem z pogranicza tematu elektronicznej gotówki, jest koncepcja zabezpieczania papierowych banknotów przy użyciu kryptografii. Rozważane zabezpieczenie korzystałoby z włosków wmieszanych w masę, z której produkuje się papier. Technologia dodawania różnokolorowych włosków do papieru jest stosowana w zabezpieczeniach od lat. Używa się ją nie tylko w banknotach. Przykładowo bilety komunikacji miejskiej w Warszawie - zanim wprowadzono bilety z paskiem magnetycznym - również miały domieszkę włosków w papierze.

Kluczowym spostrzeżeniem jest to, iż łatwo wyprodukować papier z losowym rozkładem włosków, jednak praktycznie niemożliwe jest wykonanie papieru ze z góry zadany ich rozmieszczeniem. Zatem gdyby włoski wmieszane w papier świeciły w określonym rodzaju światła (np. w ultrafiolecie), można by w procesie produkcyjnym skanować banknot, kodować cyfrowo rozmieszczenie nitek, szyfrować kluczem prywatnym, a następnie nadrukowywać na banknocie kod paskowy zawierający zaszyfrowane dane.

W celu sprawdzania autentyczności banknotów, wystarczyłoby zeskanować banknot w ultrafiolecie, odczytać kod paskowy, odszyfrować go kluczem publicznym i porównać, czy rozmieszczenie włosków na banknocie odpowiada temu, co jest zapisane w kodzie paskowym. W ten sposób weryfikacja autentyczności banknotu mogłaby być dokonywana bez ingerencji człowieka².

Aby uniknąć zagrożenia wycieku klucza prywatnego, można by zastosować kryptografię progową - tak by nikt nie posiadał całego klucza prywatnego, ale odpowied-

¹Gdyby jednak zdarzyło się że przy dwóch transakcjach, sprzedawcy będą pytać o dokładnie te same x_k / x'_k wówczas RIS'y będą identyczne i oskarżenie o oszustwo spadnie niesłusznie na Sprzedawcę.

²Pod warunkiem że dało by się automatycznie sprawdzić że to co świeci w ultrafiolecie to są prawdziwe włoski, a nie nadruk wykonany odpowiednią farbą.

nia ilość współpracujących podmiotów była w stanie zaszyfrować dowolną wiadomość (banknot).

Literatura

- [1] A. M. Odlyzko. Privacy, economics, and price discrimination on the internet. In *ICEC2003: Fifth International Conference on Electronic Commerce*. ACM. <http://www.dtc.umn.edu/odlyzko/>.

Literatura

- [1] S. Glodwasser and M. Bellare. Lecture notes in cryptography. <http://www.cs.ucsd.edu/users/mihir/papers/gb.html>.
- [2] A. M. Odlyzko. Privacy, economics, and price discrimination on the internet. In *ICEC2003. Fifth International Conference on Electronic Commerce*. <http://www.dtc.umn.edu/odlyzko/>.