Цифровые деньги

Обнаружение повторной оплаты

Цифровые деньги обладают тем свойством, что использующий их человек остается анонимным, если он тратит их не более одного раза, но если он оплачивает покупку теми же банкнотами, его личность разоблачается. Начнем с простых вещей.

Вот как это работает в общих чертах: Алиса открывает счет в банке обычным образом. Банк знает, что Алиса открыла счет, и Алиса и банк знают номер ее счета. Когда Алиса снимает деньги со счета, она связывается с банком, предоставляет доказательства того, кто она и какой у нее номер счета, и банк предоставляет ей некоторую суму в виде цифровых денег. Цифровые деньги представляют собой некую информацию, которая , возможно, хранится в файле на кредитной карточке или на жестком диске. Позже она тратит эти деньги, отправляя или вручая их Бобу (продавцу).

Боб может проверить, что деньги предоставлены банком. Он принимает деньги в любом случае, отдавая покупки Алисе. Потом он отсылает деньги в банк, занося их на свой счет.

Отметим, что, по существу, это можно было бы реализовать с помощью цифровой подписи на основе RSA. Банк мог бы послать Алисе сообщение, говорящее: “это 1$” и подписанное банком. Боб мог бы проверить, что сообщение было действительно подписано банком и, следовательно, понимая, что никто, кроме банка, не мог создать это сообщение. Боб принимает его и отсылает банку, который оплачивает его, так как узнает на нем свою подпись.

Проблема в том, что повторная оплата не может быть обнаружена или предупреждена, так как все банкноты одинаковые. Это можно исправить, если деньги будут иметь уникальный серийный номер. Теперь, когда Боб будет принимать деньги от Алисы, он может позвонить в банк и спросить, кто-нибудь уже положил на свой счет банкноту с номером 123456? Если нет, то он принимает деньги и кладет их на свой счет. Такая система платежей называется «онлайновыми электронными деньгами»; продавец должен созваниваться с банком при каждом платеже.

Эта улучшенная простая модель не может называться цифровыми деньгами, потому что не обладает отличительной особенностью цифровых денег: она не является анонимной. Когда банк видит банкноту с номером 123456, положенную на счет, он знает, что это та самая банкнота, которую Алиса сняла со своего счета. Поэтому банк может проследить, что Алиса потратила деньги у Боба. На основе этой информации может быть построено досье, раскрывающее всю конфиденциальную информацию об Алисе.

Чтобы понять, каким образом систему платежей можно сделать анонимной, займемся математикой. Мы хотим создать подпись на основе шифросистемы RSA, которую невозможно будет подделать, но такую, что банк не узнает, что она принадлежит Алисе.

Деньги в этой модели имеют следующий вид (x, f(x)^(1/3)) mod n, где n открытый ключ банка. f() (и, ниже, g()) необратимые функции такие, которые вычисляются легко, но обратные для них вычислить невозможно или сложно. Также сложно должны подбираться два различных y,z ,такие что f(y) = f(z).

Существуют два аргумента, на основании которых вышеприведенное выражение будет приниматься в качестве денег. Во-первых, только банк может вычислить что-то в степени (1/3) mod n. Больше никто не может вычислять

кубические корни. Если бы f() не была односторонней, то для случайного y, находя y^3, получаем пару (y^3, y)- мы подделали подпись. Теперь у нас есть число и его кубический корень. Однако нам не пришлось вычислять кубический корень, чтобы найти число. Поэтому вычисляется кубический корень из необратимой функции. Чтобя подделать подпись, не вычисляя кубические корни, нужно создать пару (finv(y^3), y), которая соответствует вышеуказанной модели, но вычислить обратную функцию f() нельзя. Это может служить формальной математической моделью неофициальных «денег»: банкнота, подписанная цифровой подписью, с серийным номером. Здесь x серийный номер, который специальным образом подписывается цифровой подписью.

Приятный факт, касающийся этой модели, состоит в том, что банк подписывает значение, не зная самого значения. Вот как это происходит. Алиса выбирает x, который представляет собой сумму денег. Вычисляет f(x), но вместо того, чтобы послать это в банк для подписи (возведение в степень (1/3)), она сначала выбирает случайное число r и отсылает f(x)*r^3 банку. Банк вычисляет (1/3) степень этого числа, получая r * f(x)^(1/3). Вспомним, что банк не знает по отдельности r и f(x), только их произведение. Каждый из них может быть каким угодно.

Банк высылает r * f(x)^(1/3) Алисе, она делит это число на r, которое ей известно. Это вычисление дает ей f(x)^(1/3), Алиса соединяет f(x)^(1/3) с x и получает цифровые деньги: (x, f(x)^(1/3)). То есть, у Алисы есть некоторая сумма денег, которая могла быть подписана только банком, но банк не узнает эти деньги, когда их положат на счет.

Рассмотрим другие вещи- не из области математики. Как было сказано выше, Алиса должна доказать банку, что это именно она. Примем для простоты, что все банкноты имеют одно и то же значение. В реальной системе различным значениям соответствуют различные показатели.

Когда Алиса пополняет счет Боба, тот должен связаться с банком, чтобы убедиться, что деньги тратятся впервые, это так называемая «онлайновая» система. Хотя банк не узнает x (он никогда его не видел), он запомнит все x-ы, которые были положены на счета банка и таким образом сможет предупредить Боба, если деньги были потрачены ранее. И Боб, и банк смогут проверить цифровую подпись на банкнотах и принять их.

А теперь рассмотрим схему, которая позволяет разоблачить тех, кто тратит одни и те же деньги дважды. Эта система называется «оффлайновыми» электронными деньгами, то есть Бобу больше не нужно будет связываться с банком при каждой оплате. Он принимает деньги от Алисы, зная, что если она обманывает, банк примет деньги Боба, а Алису заставит возместить убытки.

Начнем с представления самих денег. Деньги- это произведение k/2 чисел, где k- параметр безопасности, влияющий на возможность обманщика уйти от наказания. Каждое число имеет вид f(xi,yi)^(1/3), где f- необратимая функция двух аргументов, как и f выше. ("xi", "yi", "ai", etc. Отдельные значения для каждого i от 0 до k/2.) xi и yi таковы: xi = g(ai, ci), где ai и ci- случайные числа,а g- другая необратимая функция. yi выглядит от части замысловато. В основном, yi= g(ai xor <info>, di). di- еще одно случайное число, а <info>- информация о счете Алисы. Это ее номер счета, объединенный с серийным номером банкноты.

Зачем все это? Вот зачем. Если бы Вы могли узнать как ai , так и (ai xor <info>), для некоторого i, то Вы бы установили личность Алисы. (ai xor (ai xor

<info>))=<info>. Когда Алиса тратит одни и те же деньги дважды, как ai, так и ai xor <info> станут известны.

Вот, что происходит, когда Алиса тратит деньги. Для каждого i от 0 до k/2 Боб выбирает случайным образом 0 или 1. Если он выбирает 1, ему сообщается ai (и некоторая другая информация). Если он выбирает 0 ему становится известным ai xor <info> (и другая информация). Другие вещи, которые ему становятся известны, позволят ему подтвердить подлинность денег.

Теперь, когда Боб так сделал, он знает набор ai's и знает набор (ai xor <info>)'s, но они для различных i's. Он не знает ai и (ai xor <info>) для какого-то одного i. Так что он не может нарушить анонимность Алисы.

Когда Боб кладет эти деньги на счет, он передает банку информацию о ai's, полученную от Алисы.

Теперь предположим, что Алиса обманывает. Она снова тратит эти деньги где-то в другом месте, например, у Чарли. Чарли проделывает все то же самое, что и Боб, выбирая случайным образом 0 или 1 для каждого значения i. Вот где спятана уловка. Так как он выбирает случайным образом, мало вероятно, что он в точности выберет те же 0's и1's, что и Боб. (Именно здесь имеет значение размер k- чем больше k, тем менее вероятно, что Чарли и Боб выбирут одни и те же значения 0's и 1's. Хотя это делает более длинными вычисления.) Это означает, что для одного или более значений i, Чарли, наверное, выберет 0, где Боб выберет 1, и наоборот.

Поэтому, если Боб знает ai для i, Чарли знает ai xor <info>. Или если Боб знает ai xor <info>, Чарли знает ai. В любом случае, когда Чарли отсылает свои записи банку, банк сопоставит информацию Боба и Чарли и получит ai и ai xor <info> для одного i. Xor’ing их, банк получает <info>. Алиса поймана. Вот основная идея.

Другие числа ci's и di's служат для того, чтобы Боб мог подтвердить подлинность денег. Для каждого значения i Алиса должна предоставить Бобу достаточно информации для того, чтобы вычислить xi и yi. Если Боб выбирает 1, она сообщает ему ai, ci, и yi. Зная ai и ci, Боб может вычислить xi (=g(ai,ci)), а зная yi, он может вычислить f(xi,yi). Если Боб выбирает 0, она сообщает ему (ai xor <info>), а также di и xi. Зная (ai xor <info>) и di, Bob вычисляет yi (=g(ai xor <info>, di)), а зная xi он может вычислить f(xi,yi).

Так что, выбирает ли Боб 0 или 1, он получает достаточно информации для вычисления f(xi,yi). Перемножая все вместе, Боб убеждается, что количество денег равно исходной сумме денег, если ее возвести в степень (1/3). (вспомним, что деньги- это произведение f(xi,yi)^(1/3) для всех i). Только банк мог подписать эту необратимую функцию f.

Другая проблема состоит в том, что Алиса должна снять деньги со счета таким образом, чтобы банк не мог их узнать. Это означает, что она должна скрыть их значение. При этом банк хочет быть уверенным, что он подписывает «правильные» деньги. Он хочет быть уверенным, что информация об Алисе <info>, запрятанная глубоко под f's и g's, является честной. Но так как банк не видит, что он подписывает, то это сделать сложно.

Для этого можно использовать следующую схему . Алиса вычисляет f's и g's, как описано выше, аккуратно вкладывая информацию <info> о себе в каждый из них. Потом она умножает каждую f(xi,yi) на случайное число ri^3 и отсылает это банку на подпись.

Хитрость в том, что она отсылает в два раза больше значений, чем будет использовано. Она отсылает k значений, но только k/2 используются. Банк выбирает случайным образом k/2 (самое большое) из них, это и будут те банкноты, которые будут использоваться. Алисе придется отправить банку числа ri для тех значений, которые не были выбраны.

Идея состоит в том, что если Алиса попытается обмануть, вставляя «Бозо» вместо «Алиса» в поле <info>, она рискует. Во-первых, чтобы от этого была польза, она должна вставить “Бозо” во многие <info> поля для различных значений i. Когда Боб и Чарли сравнивают информацию после того, как Алиса дважды потратила деньги, каждое значение i, для которого они выбрали 0 или 1, которых будет примерно одинаковое число, обнаружат поле <info>. Если Алиса редко вставляла неправильную информацию , все равно есть шанс, что ее личность обнаружится.

Но если она фальсифицирует много значений, то, когда банк выберет половину, есть вероятность, что по крайней мере несколько окажутся в наборе, который банк не выберет. И когда Алиса отправит r's для этих значений проделка обнаружится. Банк раскроет неиспользованные f(xi,yi)'s и увидит подделанные <info> поля.

Недостатком этого метода является то, что Алисе нужно проделать вдвое больший объем работы, подготавливая деньги, половина из которых будет просто выброшена.

Итак, человек остается неизвестным до тех пор, пока он не обианывает. Все это немного сложно, но для этого и нужны компьютеры, ни Алиса, ни Боб не будут ничего делать вручную.

Стоит отметить, что “оффлайновые” электронные деньги удобны для небольших платежей, когда определить обманщика после обмана достаточно. В то время как онлайновые платежи необходимы для сделок, в которых предотвратить обман нужно перед оплатой.

Онлайн платежи

Рассмотрим три схемы онлайн платежей. Они используют одну схему для представления денег и их «девальвации» до определенного выбранного значения. Они отличаются тем, как пользователю возвращается сдача. В первой схеме вся сдача собирается пользователем в одной «банке печенья», которая может быть помещена в банк при следующей операции. Во второй и третьей схеме сдача представляет собой отдельные монеты, которые потом могут быть потрачены. В первой и второй схеме продавцу и банку известна максимальная стоимость банкноты, в третьей схеме эта информация скрыта.

Определение достоинства банкноты

Для простоты предположим, что 1 центу соответствует показатель 3 системы RSA, 2 центам- показатель 5, 4 центам- показатель 7 и так далее каждой степени двойки соответствует показатель (простое число) по модулю n. Кубический корень необратимой функции f и x соответствуют 1 центу, корень 7-ой степени- 4 центам, корень 21-ой степени- 5 центам. Конкретная сумма платежа состоит из произведения простых степеней.

Подпись на f «девальвируется» возведением в степень, соответствующую нужному значению монеты. Например, банкнота, имеющая

корень 21-ой степени может быть девальвирована из 5 центов в 1 цент возведением в 7-ую степень.

«банка печенья»

В этой схеме пользователь снимает деньги со счета, причем деньги имеют максимальное достоинство. Рассмотрим пример на Рис.1, в котором со счета снимаются две банкноты. n и ri случайные числа. Подпись банка соответствуют взятию h-ой степени , где h = 3*5*7*11.

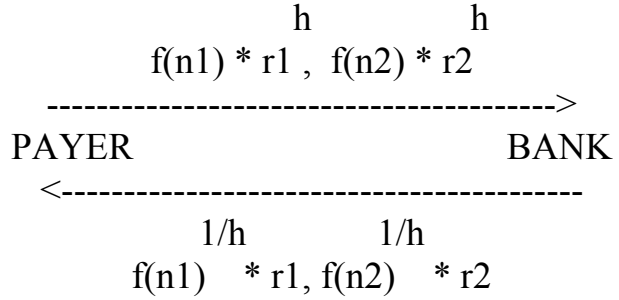

Рис. 1.1.снятие денег со счета

Пользователь делит полученные от банка значения на r1 и r2. Затем возводя их в степень 55, девальвирует значения с 15 до 5 центов. На Рис.1.2 показан первый платеж. Между пользователем и банком находится магазин, но он явно здесь не присутствует. Также не показаны сообщения, подтверждающие оплату.

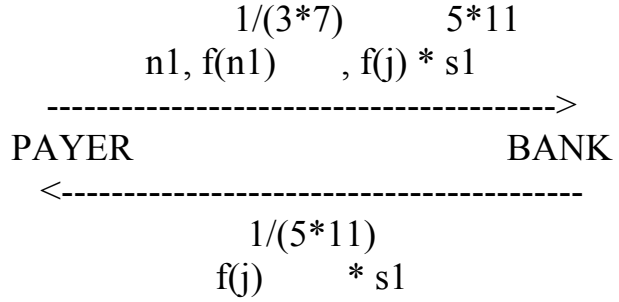

Рис.1.2

Банк легко проверяет, что первые два числа, посланные банку, n1 и f, стоят пять центов. Третье число- скрытая "банка печенья", представляющая собой скрытую функцию f от случайного числа j. Банк проверяет полученные деньги и если n1 ранее не использовалось, то он подписывает "банку печенья"с помощью степени, соответствующей сумме сдачи, и возвращает его.

Вторая оплата, показанная на рис.1.3 по существу такая же, как и первая, за исключением того, что 3 цента и "банка печенья" уже возведены в определенную степень.

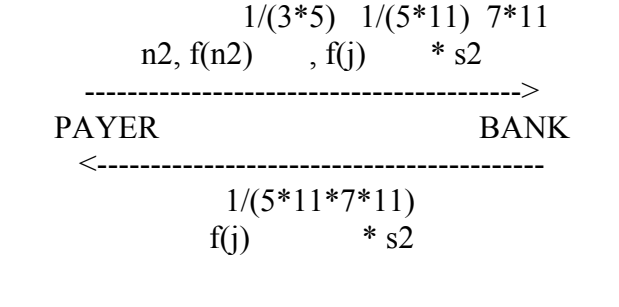

Рис. 1.3. Вторая оплата

"банка печенья" может быть помещена на счет, как показано на Рис.1.4, во время следующей операции снятия денег со счета. Она проверяется банком во многом также, как и банкнота: прообраз f не должен был ранее помещаться на счет.

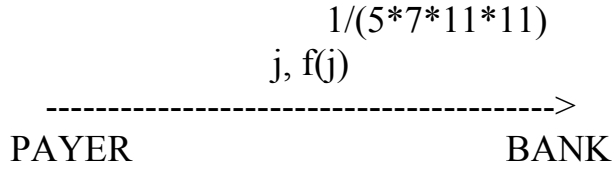

Рис.1.4. помещение на счет "банки печенья"

Банкноты определенного достоинства

Эта схема позволяет использовать сдачу, избегая промежуточной операции снятия денег. Деньги снимаются также, как и в схеме "банка печенья". Используются следующие обозначения: d – показатель, соответствующий сумме потраченных денег, g – достоинство банкноты, сдача c – это g/d, a h – максимальное достоинство банкноты. Оплата состоит из четырех чисел, первые два аналогичны числам в предыдущей схеме. Третье- это сдача, которую нужно вернуть пользователю. Четвертое- скрытое число m, которое может быть аргументом f в следующей оплате, также как n в этой.

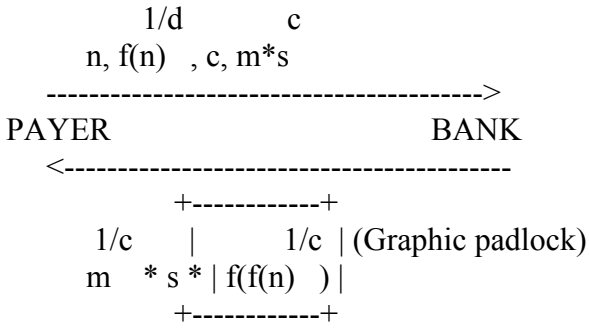

Рис.2

Возвращаемая подпись содержит множитель защиты (внутри прямоугольника). Без этого множителя пользователь мог бы попросить какое угодно количество сдачи.

Сдача, возвращаемая в виде отдельных монет

Сдача может быть разделена на части, которые составляют стоимость неиспользованных денег. Предположим, например, что при последней оплате d = 5*11, c = 3*7, a m равно выражению на рисунке.

$$m === f(n1)^{3} * f(n2)^{7}$$

$$a === m^{1/21}$$

Рис. 3.1. форма возвращаемой сдачи

$$u = 3^{-1} \bmod 7$$

$$v = 3u \operatorname{div} 7$$

$$f(n1) === (a^{1/7} * f(n2)^{3})^{-1u} * f(n1)^{-v}$$

$$f(n2) === a^{1/3} * f(n1)^{-1/7}$$

Рис. 3.2. Разделение сдачи на отдельные монеты

Такая схема возврата сдачи позволяет реже снимать деньги со счета.

Банкнота неизвестного достоинства

Плательщик может захотеть, чтобы банк или магазин не знал сумму сдачи c. В этой схеме сообщение об оплате выглядит также как и во второй схеме, за исключением того, что значение c не отсылается. В множителе защиты z-случайное число, выбираемое банком.

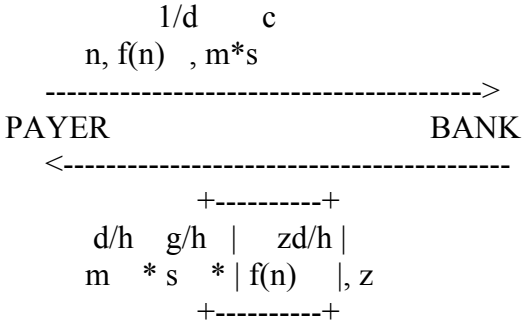

Рис. 4.

Если бы z было известно плательщику заранее, тогда он бы включил f(n) в третью компоненту m*s^c. Это дало бы ему возможность получить любую сдачу.