Костина Виктория, 2004

Электронные платежные системы

Идеальная платежная система должна отвечать следующим требованиям:

1. Безопасность системы должна препятствовать воровству денег на всех этапах выполнения операции.
2. Себестоимость операции должна быть низкой для всех участников.
3. Система должна обеспечивать высокий уровень конфиденциальности для клиента.
4. Схема и реализация должны быть простыми (не нужно использовать сложных протоколов)
5. Система должна быть открытой (протоколы и тесты программ должны быть общедоступны).
6. Система должна уметь выполнять операции с любыми долями самой мелкой денежной единицы.
7. Должна предоставлять достаточное количество данных для целей аудита.
8. Система должна быть свободной, то есть не иметь ограничений владельца.

Ни одна из существующих платежных систем не отвечает всем этим требованиям в полной мере. Платежные схемы делятся на два основных вида:

- Электронные монеты
- Электронные чеки

Электронные монеты сами по себе имеют ценность, как золото, банкнота и т. д. При использовании электронных чеков ценность несут записи, которые хранятся у доверенной организации, а во время сделки участники обмениваются инструкциями о том, как менять эти записи.

Мы рассмотрим одну систему, использующую электронные чеки:

- SET – все пользователи должны присутствовать онлайн и доверять сертификатам друг друга

И две системы, использующие электронные монеты:

- Digicash – абсолютно анонимная система
- MicroMint – не анонимная система

Электронные чеки

Достоинство электронных чеков в том, что хранение записей о сделках – неотъемлемая часть системы. Это существенно упрощает разрешение конфликтных ситуаций.

Для примера рассмотрим SET.

Безопасный протокол электронных платежей SET

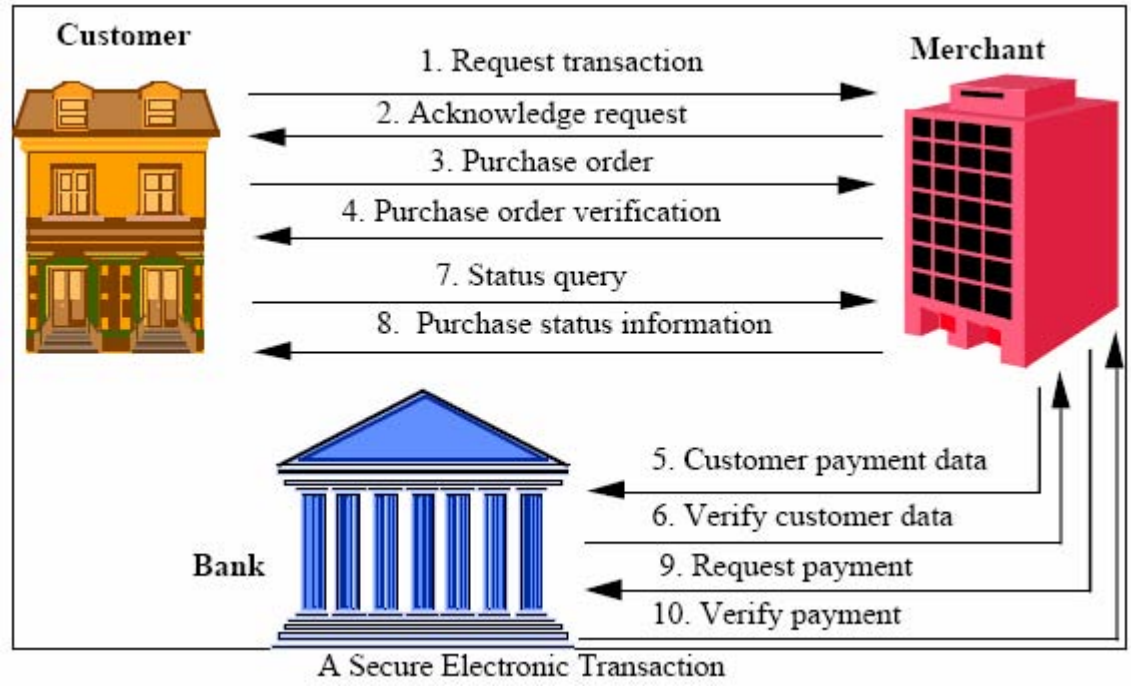
Для операций с кредитными карточками используется протокол SET (Secure Electronic transactions, 1996), разработанный совместно компаниями Visa, MasterCard, Netscape и Microsoft. Целью SET является обеспечение необходимого уровня безопасности для платежного механизма, в котором участвует три или более субъектов. При этом предполагается, что транзакция реализуется через Интернет.

Протокол SET поддерживает все возможности, предоставляемые современными кредитными карточками:

1. Регистрацию держателя карточки
2. Регистрацию продавца
3. Запрос покупки
4. Авторизацию платежа
5. Перевод денег
6. Кредитные операции
7. Возврат денег
8. Отмену кредита
9. Дебитные операции

Этот протокол рассчитан на международную ничем не ограниченную систему платежей. Многие современные WEB-броузеры поддерживают протокол SET. Это позволяет осуществлять торговлю товарами и услугами с использованием WWW-технологии.

Транзакция SET



A Secure Electronic Transaction

1. *Покупатель* инициализирует покупку. При этом покупатель выбирает продавца, просматривает его WEB-сайт, принимает решение о покупке, заполняет бланк заказа. Все

это делается до вступления в дело протокола SET. SET начинает свою работу, когда покупатель нажимает клавишу оплаты.

2. *Продавец* посылает покупателю сообщение, которое запускает соответствующую программу. Процедура эта может быть реализована с помощью PHP- или CGI-скрипта, или JAVA-аплета.

3. Программа *покупателя* посылает заказ и информацию об оплате. Для этого формируется два сообщения, одно содержит данные о полной стоимости покупки и номере заказа, второе – номер кредитной карточки покупателя и банковскую информацию. Сообщение о заказе шифруется с использованием симметричного метода (например, DES) и вкладывается в цифровой конверт, где используется общедоступный ключ продавца. Сообщение об оплате шифруется с привлечением общедоступного ключа банка. Таким образом, продавец не получает доступа к номеру кредитной карточки покупателя. Программа генерирует хэш-дайджест обоих сообщений с использованием секретного ключа покупателя. Это позволяет продавцу и банкиру проконтролировать целостность сообщения.

4. *Продавец* подтверждает своё намерение довести сделку до конца.

5. *Продавец* выделяет часть, содержащую сообщение об оплате, и перенаправляет её банкиру. Программа SET WEB-сервера продавца генерирует запрос авторизации серверу банка, где находится счет продавца. При формировании запроса авторизации используется электронная подпись продавца, базирующаяся на его секретном ключе, что позволяет однозначно его идентифицировать. Этот запрос шифруется с помощью ключа сессии и вкладывается в цифровой конверт, где используется общедоступный ключ банка.

6. *Банк* проверяет действительность кредитной карточки, дешифрует запрос авторизации продавца и идентифицирует продавца. После этого осуществляется проверка авторизации покупателя. Банк продавца авторизует данную операцию, и посылает подтверждение, подписанное электронным образом, WEB-серверу продавца

7. *Покупатель* имеет право справиться у продавца о статусе начатой транзакции.

8. *Продавец* извещает покупателя о статусе его авторизации и снова подтверждает своё намерение завершить сделку.

9. Только после авторизации (шага 5) *продавец* запрашивает оплату. Это сообщение является подтверждением продавца завершить сделку. Это сообщение, также как и ответ на него в шаге 10, шифруется симметричным ключом.

10. Та сумма, которая была зарезервирована на кредитном счёте покупателя на шаге 6, теперь переводится на счёт продавца.

Безопасность

Итак, видно, что каждый шаг реализации протокола SET сопровождается аутентификацией. Это препятствует какому-то внешнему субъекту стать посредником и видоизменять сообщения. Для нормальной работы протокола SET все участники должны зарегистрироваться и снабдить партнеров своим общедоступным ключом.

«Переиграть» сделку невозможно, так как текущее время включено в каждое сообщение.

Электронные монеты

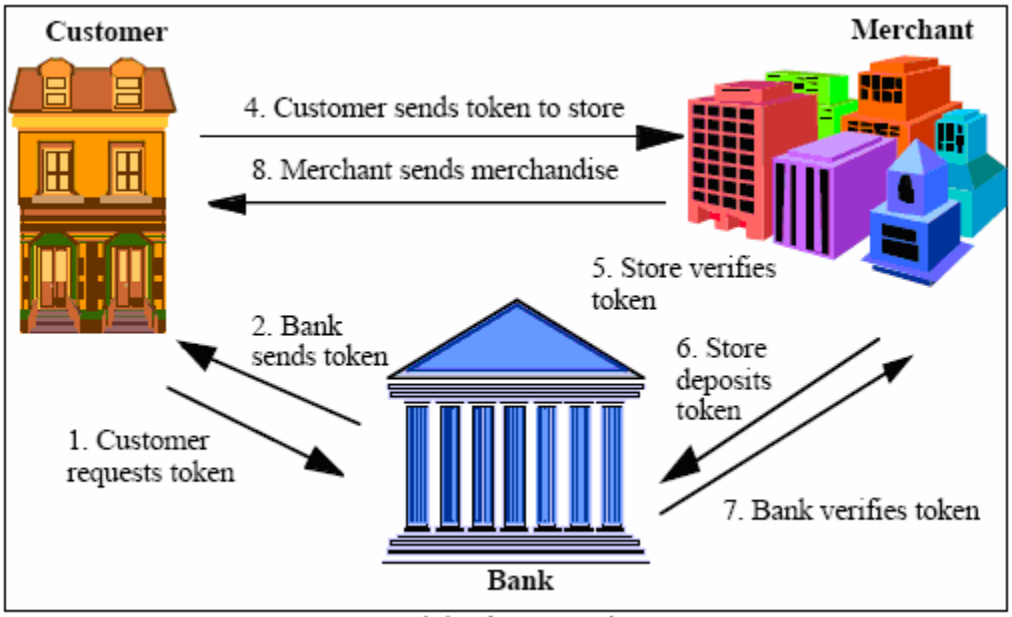
При использовании электронных монет последовательность бит, передаваемая в транзакции, сама по себе ценна, в отличие от электронных чеков, где последовательность бит в транзакции содержит инструкции о том, как менять счёт в бухгалтерской книге. Доллар сам по себе имеет ценность и не является долговым обязательством, как, например, чек. Из-за своей независимости, электронные монеты не привязаны к определённому лицу или сделке.

Digicash

Система Digicash (Chaum, 1985) была первой и остаётся канонической системой электронных монет в Интернет-торговле. В ней была представлена идея «слепых подписей», согласно которой, банк проверяет подлинность денежных знаков своих клиентов, но не может отследить путь денег от одного клиента к другому. Идея реализуется при помощи цифровых подписей RSA.

Компания Digicash объявила банкротство в 1998 г. Сейчас ключевые технологии находятся в распоряжении у фирмы eCash Technologies, Inc. В настоящий момент система продвигается несколькими банками Европы, Северной Америки и Австралии, включая германский Deutsche Bank 24.

Транзакция Digicash



A Digicash Transaction

1. *Покупатель*:
   - Выбирает случайное число k, называемое «слепящим множителем»
   - Создаёт жетон m
   - $r=(m\ k^e)$ mod n посылает в банк для утверждения, где e – открытый ключ банка

2. *Банк* подписывает жетон соответствующим секретным ключом d: $r^d = (m\ k^e)^d$ mod n и сохраняет полученное число r, которое содержит всю информацию, скрытую «слепящим множителем»

3. *Покупатель*, получив r^d, удаляет «слепящий множитель», вычисляя

$$t = r^d k^{-1} \bmod n = (m k^e)^d k^{-1} \bmod n = m^d \bmod n$$

Жетон подписан и считается утверждённым. Однако, банк никогда не видел его и не сможет сказать, какой жетон он подписал для своего клиента.

4. *Покупатель* посылает заказ на товар и жетон (в качестве оплаты) торговцу.

5. *Продавец* убеждается, что подпись банка правильная.

Предполагается, что жетон имеет особую форму, и проверка подписи банка с помощью его открытого ключа контролирует, что жетон был подписан банком.

6. *Продавец* кладёт жетон в банк.

7. *Банк* подтверждает взнос.

Тот факт, что жетон подписан, совсем не означает, что он и в самом деле имеет ценность, поскольку продублировать цифровой жетон - пустяковое дело. Если жетон уже был погашен, его копия, полученная продавцом, ничего не стоит.

8. *Продавец* доставляет заказанные и оплаченные товары.

Заметим, что по жетону, полученному банком на 5-м шаге, нельзя определить, что это именно тот жетон, который банк подписывал в шаге 2. Это важнейший момент, который и обуславливает анонимность Digicash.

Безопасность

Digicash не предоставляет информации, необходимой для разрешения конфликтов и предотвращения недоразумений.

В самом деле, если протокол прерывается между оплатой и доставкой товара, покупатель оказывается обманутым. Так как после доставки жетона продавцу невозможно доказать, кому же он изначально принадлежал, покупатель не может просто попросить продавца переслать товар. Если продавец честный, он перешлёт товар, например, по тому же IP-адресу. Однако, он также может утверждать, что не получал жетона, так как у банка нет никакого способа отследить путь подписанного им жетона после того, как он выдал его своему клиенту.

Покупатель сам может мошенничать, пытаясь обналичить жетон быстрее, чем это сделает получивший его продавец. Один из вариантов разрешения этого спора между покупателем и продавцом – предположить, что покупатель всегда прав. Для этого потребовалось бы хранить имён недовольных покупателей и продавцов, на которых они жалуются, в то же время сохраняя анонимность успешных сделок. Другой вариант, которому отдаётся предпочтение в позднее разработанной Чомом (Chaum) альтернативной системе – пожертвовать анонимностью жетонов, заключая в них идентификационную информацию.

Micromint

Micromint использует необратимость хеш-функции и «парадокс дня рождения» (Mosteller,1965) для создания электронных денег.

Протокол Micromint был предложен Райвестом и Шамиром в 1996 г. Однако, полной коммерческой версии протокола до сих пор не выпущено из-за некоторых проблем, связанных с его реализацией.

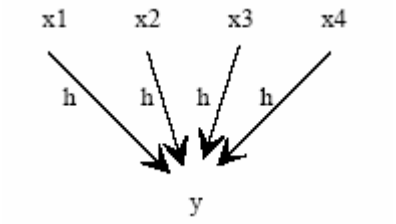
Парадокс «дня рождения» заключается в следующем: сколько людей должно быть в комнате, чтобы с 50%-ой вероятностью у двоих из них дни рожденья совпали? Только 23. А сколько людей должно быть в комнате, чтобы с 99,99%-ой вероятностью у двоих из них дни рожденья совпали? Ответ – 100, хотя на первый взгляд может показаться, 365. Причина в том, что вычисление соответствующих дней рождений – случай выборки с перестановкой. Формула вычисления вероятности, что из x человек двое имеют день рожденья в один, есть:

$$1 - \frac{365!}{(365\text{-}x)!\ 365^x}$$

Тот же самый принцип применяется для вычисления столкновений хеш-значений: из всех x-знаковых чисел, выберем два числа с одинаковым хеш-значением. Чтобы найти такие числа обычным методом проб и ошибок, атакующему потребовалось бы $O(2^{n/2})$ операций, где n – размерность хеш-значения. Парадокс «дня рождения» - частный случай k-направленной коллизии.

Оценим вычислительное время, требуемое для нахождения коллизии в хеш-значениях. Чтобы получить одну k-направленную коллизию, нужно вычислить $2^{n(k-1)/k}$ хеш-значений. Однако, проверив $c*2^{n(k-1)/k}$ значений, при условии, что $1<<c<<2^{n/k}$, получим c^k k-направленных коллизий. Следовательно, коль скоро первая коллизия найдена, производство новых коллизий эффективно ускоряется. Монета в Micromint – k-направленная хеш-коллизия, т.е. набор k чисел $h(x_1, x_2, x_3,. . . x_k)$, имеющих одинаковые хеш-значения:

$$h(x_1)=h(x_2)=h(x_3)=. . .=h(x_k)$$



Используя хеш-значения, которые всенародно известны после того, как монеты были выпущены, продавцы могут проверить подлинность монет в оффлайновом режиме, без необходимости созваниваться с продавцом. Чтобы предотвратить использования одной и той же монеты дважды, идентификационная информация клиента встроена в хэш-значения, проданные ему:

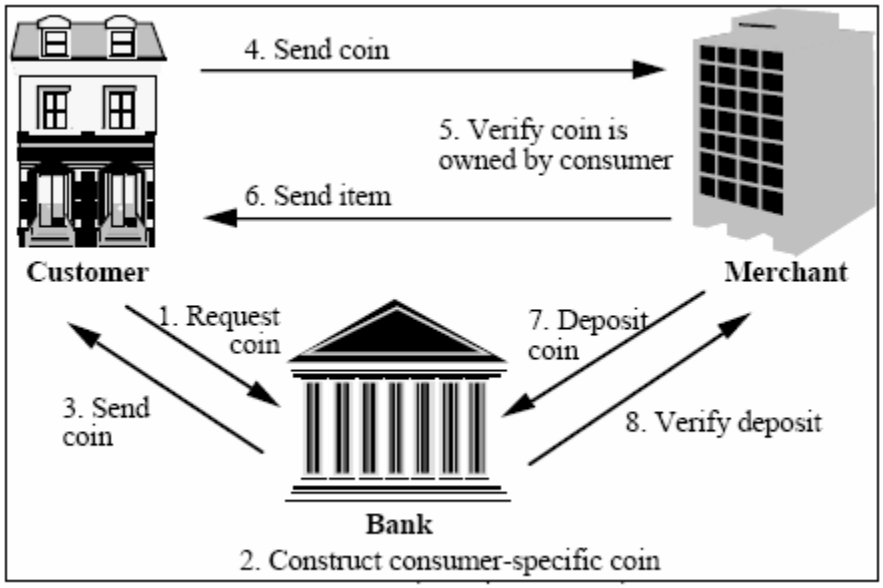
$$h(\text{монета}) = h(x_1\ x_2, x_3,. . . x_k) = h(\text{идентификационная информация}).$$

Таким образом, двойное использование денег будет разоблачено на стадии, когда монеты кладутся в банк. Рекомендуется использовать 16-битовые хеш-значения по состоянию вычислительной техники на конец XX века. Таблица, приведённая ниже, показывает стоимость выпуска монет. Таблица представляет случай 36-битовых хеш-значений, причем для создания монеты требуется 4 коллизии. Из таблицы видно, что для создания первой монеты требуется гигантские усилия, но по мере того, как количество сгенерированных монет растёт, падает их себестоимость.

Количество хэш-значений	Количество монет	Хэш-значений на монету
$2^0 \ldots 2^{26}$	0	•
2^{27}	1	2^{27}
2^{29}	2^8	2^{21}
2^{32}	2^{20}	2^{12}
2^{36}	2^{32}	2^3

С возможностью крупномасштабного мошенничества борются, изменяя хеш-значение ежемесячно. Более того, банк может заметить фальшивые монеты в любое время и объявить новую хэш-функцию или использовать скрытые предикаты для ежедневного обновления, т. е. конструировать монеты таким образом, что каждое x_i имеет определённую форму. (Скрытый предикат – особая характеристика числа, которая не очевидна непосвящённым. Для генерации числа со скрытым предикатом, некоторые знаки числа делаются функциями от других, случайных знаков.)

Транзакция Micromint.



A MicroMint Transaction

1. *Клиент* запрашивает монету. Он должен подтвердить свою личность, чтобы получить монету. Как именно он это делает, не специфицировано в протоколе Micromint.
2. *Банк*, получив идентификационную информацию в шаге 1, создаёт монету. Затем он создаёт монету, связанную с личностью клиента. Заметим, что банк хранит огромную базу данных известных хеш-значений, из которых можно сконструировать монету, а не начинает хеширование заново при каждом запросе клиента.
3. *Банк* доставляет запрошенные монеты клиенту. После этого шага клиент может доказать, что у него подлинные Micromint монеты, и может доказать своё владение этими монетами. Однако, чтобы доказать своё владение ими, ему придётся раскрыть свою личность.

4. *Клиент* посылает продавцу информацию, необходимую для сделки: название товара, цену, монеты и идентификатор личности. Последнее удостоверяет монетами, что монеты, которые предоставляет покупатель, являются его собственностью.
5. *Продавец* проверяет монеты и их принадлежность покупателю.
6. *Продавец* доставляет товар покупателю.
7. *Продавец* удостоверяется, что монеты не были ещё потрачены, кладя их в банк.

Так как Micromint сделки не анонимны, покупатель может быть идентифицирован, когда продавец кладёт монеты в банк, однако совершенно необязательно разрешать банку анонимно наводить справки в счёте продавца.

Безопасность

Параметры безопасности в Micromint включают:

- Размерность хеш-значения, используемого для создания монет
- Секретность хеш-значения перед тем, как оно выпускается

Если хеш-значение станет известно фальшивомонетчику заранее, он сможет генерировать монеты так же быстро, как и банк.

- Число коллизий для того, чтобы сконструировать монету.

Увеличение количества коллизий k, необходимых для создания монеты, увеличивает стоимость производства монет банком и стоимость проверки монет торговцем. Увеличение количества коллизий также улучшает защищенность монет. Разработчики предлагают k=4 как оптимальный вариант.

MicroMint рекомендует, чтобы банк и клиент имели общий ключ DES, чтобы обеспечить безопасность передачи монет между банком и клиентом. Этот же DES ключ может использоваться и для аутентификации.

Литература

1. Linda Jean Camp “PRIVACY & RELIABILITY IN INTERNET COMMERCE”, A Dissertation Submitted to the Graduate School in Carnegie Mellon University, 1996 http://reports-archive.adm.cs.cmu.edu/anon/1996/CMU-CS-96-198.ps

2. Don B. Johnson and Stephen M. Matyas “Asymmetric Encryption: Evolution and Enhancements”, The technical newsletter of RSA Laboratories, a division of RSA Data Security, Inc., 1996 http://security.ece.orst.edu/koc/ece575/rsalabs/crypto2n1.pdf

3. Семенов Ю.А. “Telecommunication technologies - телекоммуникационные технологии (v2.1)”, статья на основе материалов книг автора "Протоколы и ресурсы Интернет" (Радио и связь, М. 1996), "Сети Интернет. Архитектура и протоколы" (Сиринъ, М. 1998), "Протоколы Интернет. Энциклопедия" ("Горячая линия - Телеком", М. 2001). http://www.podgoretsky.com/ftp/Docs/Internet/Semenov/4/6/set_66.htm