Московский Физико-Технический Институт
Факультет Радиотехники и Кибернетики

Цифровая подпись для электронной почты

Лушников П. А., студент 013 гр.

Введение

Сегодня самые консервативные из государств осознали, что без Интернета они будут отставать от своих конкурентов. Интернет все больше превращается в среду для бизнеса, а это требует надежных средств шифрования и цифровых подписей. В случае шифрования вы будете уверены, что никто, кроме вашего адресата, не сможет познакомиться с содержимым вашего письма, а цифровая подпись гарантирует, что если кто-то по пути открывал файл с вашим письмом и изменял содержимое, то это не останется незамеченным.

Представьте себе ситуацию: вам отправили по электронной почте документ с конфиденциальной информацией по финансированию на будущий год. Вам необходима абсолютная уверенность в том, что полученный файл не послан злоумышленником и совершенно идентичен оригиналу, а содержащиеся в нем цифры не были изменены «в пути». Пара скорректированных значений могут стоить вашей фирме круглой суммы. Подозрение, что документ "в пути" был подделан, появляется, если некоторые цифры не сходятся, а электронная передача велась через внешнюю почтовую систему. Как убедится в том, что полученный вами документ - абсолютная копия отправленного вам оригинала?

Рассмотренная ситуация не настолько искусственна, как может показаться с первого взгляда. В век, когда цифровая коммерция быстро становится реальностью, доверие пользователей к подобного рода системам целиком зависит от безопасности таких транзакций. Если отправить по электронной почте или передать на гибком диске файл с электронной таблицей, то каким образом получатель узнает о том, что никто, через кого эта информация прошла, не внес каких-либо изменений? Если переслать по сети Internet номер своей кредитной карточки, то как адресат убедится в том, что именно вы сделали этот заказ?

Создание цифровой подписи.

Решение вышеизложенных вопросов нужно искать в специальном разделе криптографии, связанном с подменой цифровых данных. В нем можно найти ответы на то, как проверить достоверность цифровых данных и как по аналогии с рукописной подписью на бумаге проставить визу на электронных документах, имея в распоряжении лишь последовательности нулей и единиц.

Контрольные суммы

Наиболее простой способ проверки целостности данных, передаваемых в цифровом представлении, - это метод *контрольных сумм*. Под контрольной суммой понимается некоторое значение, рассчитанное путем сложения всех чисел из входных данных. Если сумма всех чисел превышает максимально допустимое значение, заранее заданное для этой величины, то величина контрольной суммы равна коэффициенту полученной суммы чисел - то есть это остаток от деления итоговой суммы на максимально возможное значение контрольной суммы, увеличенное на единицу. Если сказанное записать в виде формулы, то для расчета контрольной суммы будет использоваться следующее выражение:

```
Checkssum = Total % (MaxVal + 1)
```

где Total - итоговая сумма, рассчитанная по входным данным, и MaxVal - максимально допустимое значение контрольной суммы, заданное заранее.

Недостаток метода контрольных сумм заключается в том, что хотя несовпадение значений этих сумм служит верным доказательством, что рассматриваемый документ подвергся изменению, равенство сравниваемых значений еще не дает гарантии, что информация осталась неизменной. Можно произвольным образом изменить порядок следования чисел в документе, а контрольная сумма при этом сохранит прежнее значение. И что еще хуже - можно изменить отдельные числа в документе и подогнать остальные таким образом, чтобы обеспечить прежнее значение контрольной суммы. При использовании для контрольных сумм 8-разрядной переменной вероятность того, что контрольные суммы двух совершенно случайно выбранных последовательностей данных будут одинаковы, равна 1/256. При увеличении длины переменной под контрольную сумму до 16 или 32 разрядов, вероятность совпадений уменьшается, однако этот механизм все равно слишком чувствителен к возможным ошибкам, чтобы обеспечить высокую степень доверия к представленным данным.

Контроль CRC

Более совершенный способ цифровой идентификации некоторой последовательности данных - это вычисление контрольного значения ее циклического избыточного кода (cyclic redundancy check - CRC).

Механизм CRC основан на полиномиальном распределении, где каждый разряд некоторой порции данных соответствует одному коэффициенту большого полиномиального выражения. Если некоторый «магический» полином (коэффициенты которого получены в соответствии с используемым алгоритмом CRC) разделить на полином, представляющий какую-то последовательность данных, то в результате получается полином-частное и полином-остаток. Второе из этих значений служит основой для создания контрольного параметра CRC.

Если в некотором огромном блоке данных лишь один разряд стал другим, то и контрольный параметр CRC со 100-процентной вероятностью также будет иметь другое значение. Если же изменятся два разряда, то вероятность обнаружения ошибки при длине параметра CRC в 16-разрядов, составляет более 99,99%. В отличие от контрольных сумм метод CRC сможет распознать всякие фокусы с перестановкой двух байт либо с добавлением 1 к одному из них и вычитанием 1 из другого.

Но даже контроль с помощью 32-разрядного значения CRC обладает определенными недостатками - он устойчиво обнаруживает случайные изменения во входной информации (например, возникающие в результате сбоев при передаче данных), однако недостаточно надежен в случае преднамеренных действий. Если для идентификации некоторого файла вы используете его 32-разрядный параметр CRC, то для кого-то не так уж сложно с помощью компьютера создать совершенно другой файл с тем же значением CRC.

Алгоритмы хэширования

Более высокой надежности, чем при контроле CRC, можно достичь при использовании односторонних алгоритмов хэширования; результатом их работы являются особые «хэшированные» значения. Под термином "односторонние" понимается следующее: имея на входе A, можно без особого труда получить на выходе B, но сделать обратное - то есть из B получить A - невозможно, или, во всяком случае, практически невозможно. Важная отличительная особенность любого хорошего алгоритма хэширования заключается в том, что генерируемые с его помощью значения настолько уникальны и трудноповторимы, что вряд ли кто-то даже с помощью серии суперкомпьютеров Cray и затратив колоссальное количество времени, сможет найти два набора входных данных, имеющих одинаковые значение хэширования. Как правило, эти параметры занимают не менее 128 разряды. Чем больше их длина, тем труднее

воспроизвести входной набор данных, то есть найти последовательность, обеспечивающую соответствующий результат.

Среди односторонних алгоритмов хэширования наибольшей известностью пользуются два из них: алгоритм MD5 (message digest), разработанный профессором Массачусетского технологического института Роном Ривестом (Ron Rivest) (один из авторов популярной криптосистемы с ключом общего пользования RSA), и алгоритм Secure Hash Algorithm (SHA), созданный совместными усилиями Национального института по стандартизации и технологическим разработкам (NIST) и Управления национальной безопасности США (NSA). Результат анализа последовательности входных данных с помощью алгоритма MD5 - 128-разрядный цифровой идентификатор, а при использовании алгоритма SHA - 160-разрядное значение. Учитывая, что пока никому не удалось подобрать ключ ни к одному из названных алгоритмов, можно считать, что восстановление исходных данных по некоторому хэшированному значению, являющемуся результатом работы алгоритма SHA либо по некоторому коэффициенту алгоритма MD5 нереально. Таким образом, если вам отправили какой-то файл и идентификатор, полученный в результате применения к нему алгоритма MD5 или SHA, и если вы выполнили с ним тот же алгоритм хэширования, и ваш результат совпал с исходным значением, определенно можно быть уверенным, что принятый вами файл не подвергся искажениям.

Система цифровой подписи

Как же осуществляется цифровая подпись письма? Рассмотрим еще один пример. Допустим, Сэм собирается отправить Тому контракт или номер своей кредитной карточки в цифровом виде. Для подтверждения подлинности этих документов Тому необходима цифровая подпись Сема. Сначала Сэм отправляет свой документ. Затем использует алгоритм хэширования для вычисления идентификатора этого документа, шифрует хэшированное значение с помощью своего личного ключа и отправляет его Тому. Это и есть цифровая подпись Сэма. Том с помощью того же алгоритма хэширования сначала вычисляет идентификатор принятого документа. Затем он расшифровывает значение, которое получил от Сэма, используя предоставленный Сэмом ключ общего пользования. Если два значения хэширования совпали, Том не только узнает, что этот документ подлинный, но и то, что подпись Сэма действительна. Понятно, что проведение коммерческих транзакций по такому сценарию значительно безопаснее, чем с использованием подписи от руки на бумаге, которую можно подделать. А если сведения, передаваемые Сэмом Тому, конфиденциальны (например, содержат номер кредитной карточки), то и их можно зашифровать так, чтобы прочитать их смог только Том.

Схема подписи отсылаемого документа

1. Сэм обрабатывает по специальному алгоритму документ, который собирается отправить Тому, в результате получает некоторый параметр, рассчитанный на основании содержимого документа. Обычно это занимает значительно меньше места, чем исходный документ - параметр 128 или 160 двоичных разрядов.
2. Затем Сэм с помощью своего личного ключа шифрует полученный параметр. Итоговое значение служит цифровой подписью Сэма.
3. Сэм отправляет Тому документ и свою цифровую подпись.
4. Том пропускает документ, полученный от Сема, через тот же алгоритм, которым пользовался Сем.
5. Затем Том дешифрует цифровую подпись, полученную от Сэма, пользуясь предоставленным Сэмом ключом общего пользования.
6. Том сравнивает значение параметра, полученного при выполнении операции 4, с расшифрованным значением цифровой подписи. Если эти значения совпадают,

значит, подпись подлинная и документ "в пути" не подвергся изменениям. В противном случае, либо документ искажен, либо подпись подделана, либо и то и другое.

Вероятнее всего именно по такой, или подобной схеме будут вестись дела через Internet или любую другую информационную службу. Этот алгоритм послужил основой проекта государственного стандарта США - Digital Signature Standard (DSS). В нем применяются: алгоритм Secure Hash Algorithm для расчета параметров хэширования и криптосистема с ключом общего пользования, известная под названием Digital Signature Algorithm (DSA) и предназначенная для получения цифровой подписи по параметрам хэширования. Ряд пунктов проекта DSS подверглись критике, однако многие из замечаний исходили от групп, финансово заинтересованных в отклонении данного проекта.

Стандарты цифровой подписи и центры сертификации

Использование зашифрованной и подписанной электронной почты может оказаться весьма желательным для самых разнообразных приложений. Сейчас используются два основных стандарта на защиту электронной почты — S/MIME и PGP. Оба они могут использоваться как для шифрования, так и для электронной подписи отправляемых сообщений. PGP тесно связан с одноименной программой (и ее клонами), созданной Филиппом Циммерманном. А S/MIME разработан группой компаний и поддерживается многими почтовыми клиентами, будь то продукты Microsoft, Netscape или других производителей.

Не вдаваясь глубоко в вопросы криптографии, можно отметить, что S/MIME использует комбинацию симметричных и асимметричных алгоритмов шифрования. При шифровании сообщения, как и в других гибридных схемах, основная часть письма преобразуется с помощью более быстрых симметричных алгоритмов. Для этой цели могут использоваться криптоалгоритмы RC2, DES или TripleDES с ключами длиной 40, 56 и 168 бит соответственно. Сам ключ, на котором шифруется письмо, защищается с помощью асимметричных преобразований (алгоритм RSA длиной ключа от 512 до 2048 бит). А для распределения отрытых ключей асимметричного шифрования используются сертификаты, соответствующие стандарту X.509. Дайджесты сообщений формируются с помощью алгоритмов SHA-1 или MD5.

Таким образом, для работы с защищенной почтой нужно сгенерировать пару криптографических ключей «открытый/секретный» и удостоверить открытый ключ с помощью цифрового сертификата. Выдается он центром сертификации (ЦС, английское обозначение CA - Certificate Authority). Сертификат содержит, кроме удостоверяемого открытого ключа, номер сертификата, информацию о пользователе, центре сертификации, даты выдачи и окончания срока действия, назначение удостоверяемого ключа (для защиты почты, для использования в SSL-соединениях, шифрования файлов и т. п.), а также некоторые дополнительные поля. Все это дополняется открытым ключом ЦС и подписывается им.

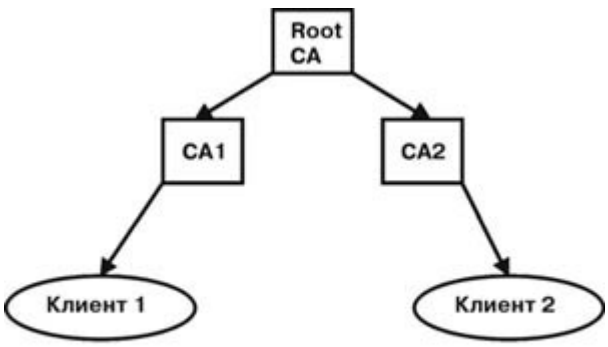

Схема 1

Центры сертификации обычно образуют древовидную структуру, пример которой представлен на схеме 1. Стрелками показано, кто чью подлинность удостоверяет. Особое место занимает корневой центр сертификации. Он удостоверяет сам себя. А его подлинность проверяется каким-либо особым способом. Например, сертификаты,

удостоверяющие ЦС-компании VeriSign, «зашиты» в продуктах Microsoft.

Если пользователь получает с письмом сертификат, выданный «незнакомым» ему ЦС (например, в соответствии со схемой 1, «Клиент 1» получил от «Клиента 2» сертификат, выданный CA2), то он имеет возможность отследить всю цепочку центров сертификации вплоть до корневого и, таким образом, проверить подлинность.

Кроме выдачи сертификатов ЦС обеспечивает возможность проверки их подлинности, а также периодически выпускает списки отозванных сертификатов (английской название CRL - Certificate Revocation List). В CRL заносятся сертификаты, аннулированные по каким-то причинам: например, выяснилось, что скомпрометирован секретный ключ, соответствующий удостоверяемому открытому ключу.

Теперь предположим, что две организации решили использовать шифрованную и подписанную электронной цифровой подписью почту для обмена какими-то важными сообщениями. Причем желательно сохранить не только конфиденциальность переписки, но и обеспечить защиту от разного рода мошенничества, связанного с отказом от факта передачи сообщений. Иными словами, стороны не доверяют друг другу.

Если был сделан выбор в пользу S/MIME, то нужно сгенерировать и удостоверить используемые открытые ключи. В принципе, можно поступить следующим образом. У специализированной организации, например, той же VeriSign, приобрести цифровые сертификаты для всех сотрудников, которые по статусу могут участвовать в подобной переписке. А можно поступить иначе - организовать в каждой организации свой корпоративный центр сертификации. Если раньше для этого требовалось специальное ПО, то сейчас это можно сделать с помощью службы Certificate Services из состава серверных ОС семейства Windows 2000.

Примеры использования цифровой подписи

В настоящее время цифровая подпись документов, пересылаемых по Internet с помощью электронной почты или другим образом, становится все более широко используемой. Например, для обеспечения целостности и достоверности сообщений, циркулирующих в системе Центрального банка Российской Федерации, приказом ЦБ РФ от 21 сентября 1993 г. N 02-159 был введен в действие стандарт Центрального банка Российской Федерации «Подпись цифровая электронная».

Другой пример - правительство Чувашии создало Удостоверяющий центр, который позволит органам исполнительной власти вести всю переписку по электронной почте с использованием электронной цифровой подписи (ЭЦП). Это - первый региональный удостоверяющий центр, создаваемый в рамках федеральной целевой программы «Электронная Россия». В Чувашии уже более полутора лет все финансовые документы в системе республиканского казначейства пересылались «под защитой» электронной подписи. Теперь масштабы использования ЭЦП многократно возрастают: ею будут оснащены все республиканские и муниципальные органы власти, бюджетные учреждения, органы ЗАГС и МВД, крупные предприятия.

Используемые источники:

1. http://fort.stup.ac.ru/wmaster/books/magazine/pcmag/9612/129611.htm
2. http://www.ci.ru/inform13_01/p17ms.htm
3. http://www.programs-gov.ru/cgi-bin/news.cgi?news=140&id_news=395