Электронная подпись «для чайников». Часть 2

http://habrahabr.ru/blogs/infosecurity/97283/

Продолжая раскрывать «тайное знание» о цифровой подписи простым языком, выясним, что же нам необходимо для удобной и эффективной работы с ней, а также разберемся в основном различии между S/MIME + X.509 и PGP. Перед тем, как рассматривать особенности этих двух технологий, стоит обсудить, какая информация нужна получателю для проверки подписи (а наш зашифрованный хэш уже вполне можно называть подписью), и в каком виде ее можно передать.

Пакеты для ключей.

Каждую из частей информации можно передать вместе с открытым ключом, или вместе с нашей подписью, а можно и с тем и с тем, для большего удобства, ведь данные можно и не разделять. Но в этом случае, каждый раз, отправляя подписанную информацию, мы отправляем одно и то же. Примерно как если бы к каждому отправляемому нами бумажному письму (даже короткому, в две строки), мы бы прикладывали дополнение вида «Здравствуйте! Это я, В. Пупкин, которого вы встречали на Красной площади Москвы, где мы и познакомились, потом пошли в ресторан…». Согласитесь, слегка неудобно.

Вернемся к данным, необходимым для проверки подписи.

Начнем с информации, которая позволит нам узнать, кто же сделал эту подпись. Как мы уже договорились, ассиметричное шифрование позволяет однозначно связать наш открытый ключ и полученную подпись. Беда в том, что сам по себе открытый ключ – набор байт. При этом он, конечно, связан с закрытым, которым мы (то есть отправитель) владеем, но связь эта для получателя неочевидна. У него есть набор байт от В. Пупкина, от И. Петрова, от С. Сидорова… И от десятка других людей. И как ему их идентифицировать? Держать отдельный реестр для того, кому какой набор байт принадлежит? Это что же, получается уже **второй**реестр (помимо того, где должно быть записано, с помощью какой хэш-функции какой хэш сделан)! И опять, неудобно!

Значит, надо связать каждый открытый ключ с информацией о том, кому этот ключ принадлежит, и присылать это все одним пакетом. Тогда проблема реестра решается сама

собой – пакет (а если более правильно, контейнер) с открытым ключом можно будет просто посмотреть и сразу понять его принадлежность.

Но эту информацию все так же надо связать с подписью, пришедшей получателю. Как это сделать? Надо соорудить еще один контейнер, на сей раз для передачи подписи, и в нем продублировать информацию о том, кто эту подпись создавал.

Продолжая нашу аналогию с красивым замком, мы пишем на ключе «Этот ключ открывает замок В. Пупкина». А на замке тоже пишем «Замок В. Пупкина». Имея такую информацию, получатель нашей коробочки не будет каждый из имеющихся у него ключей вставлять наугад в наш замок, а возьмет наш ключ и сразу его откроет.

Теперь, по переданной информации при проверке можно найти контейнер открытого ключа, взять оттуда ключ, расшифровать хэш и…

А собственно, что «и»? Мы ведь пока так и не решили проблему, как донести до получателя данные о том, какая хэш-функция применялась для хэша, а ведь для проверки подписи эта информация **необходима**! Решить ее можно достаточно просто: положить эту информацию в контейнер вместе с нашим открытым ключом. Ведь именно связка «хэширование – шифрование результата хеширования» считается процедурой создания цифровой подписи, а ее результат – подписью. А значит, достаточно логичным представляется объединение в связку алгоритма шифрования хэша и хэш-функции, с помощью которой он сформирован. И доставлять эту информацию тоже надо в связке.

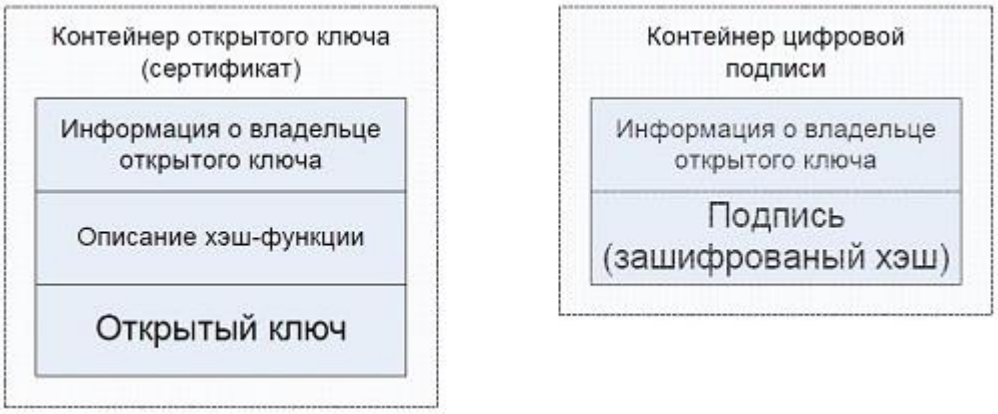

Рисунок 1. Контейнеры открытого ключа и цифровой подписи.

Теперь, ненадолго вернемся к информации о подписывающем. Какого рода эта информация должна быть? ФИО? Нет, В. Пупкиных много. ФИО + год рождения? Так и родившихся в один день В. Пупкиных тоже предостаточно! Более того, это может быть Василий, Виктор, или даже Василиса или Виктория Пупкины. Значит, информации должно быть больше. Ее должно быть столько, чтобы совпадение всех параметров, по которым мы идентифицируем человека, было максимально невероятным.

Безусловно, такой пакет информации создать возможно. Вот только, работать с ним уже трудновато. Ведь надо наши контейнеры открытых ключей надо сортировать, хранить, использовать, в конце концов. А если для каждого использования придется указывать по полсотни параметров, то уже на втором контейнере станет понятно, что что-то надо менять. Решение этой проблемы, конечно же, было найдено.

Контейнеры и идентификаторы.

Чтобы понять, в чем же оно заключалось, обратимся к бумажному документу, который есть у всех нас: к паспорту. В нем можно найти и ФИО, и дату рождения, и пол, и много другой информации. Но, главное, в нем можно найти серию и номер. И именно серия и номер являются той уникальной информацией, которую удобно учитывать и сортировать. Кроме того, они существенно короче всей оставшейся информации вместе взятой, и при этом все так же позволяют опознать человека.

Применяя этот же подход к контейнерам открытых ключей, мы получим, что у каждого контейнера должен быть некий номер, последовательность символов, уникальная для него. Эту последовательность символов принято называть **идентификатором**, а сами контейнеры – **сертификатами**, либо просто ключами.

Вот здесь и начинаются принципиальные различия в идеологиях OpenPGP и S/MIME + X.509. Для краткого понимания их, вернемся к нашей аналогии с паспортом.

Паспорт вы можете использовать при покупках билетов, при оформлении документов, для выдачи пропуска на какую-либо территорию и даже на территории других стран! То есть, вы используете его для идентификации вашей личности в самых различных, зачастую абсолютно не связанных друг с другом, местах, с самыми различными людьми. И везде ваш паспорт принимают. Гарантом того, что вы – это вы выступает третья сторона в ваших взаимоотношениях с другими: государство. Именно оно выдало вам ваш паспорт, специально оформленный, подписанный и заверенный, и именно поэтому ваш паспорт является таким универсальным документом.

С другой стороны, в кругу друзей, или внутри компании вам достаточно представиться так: «В. Пупкин из твоей группы в институте» или же «В. Пупкин из отдела продаж». И людям, с которыми вы контактируете в этом кругу уже не нужна третья сторона, они и так помнят Пупкина из группы с которым проучились пять лет, или Пупкина из отдела продаж, с которым недавно ходили обедать, и предоставленной вами информации им вполне достаточно.

Так же различаются и технологии OpenPGP и S/MIME + X.509.

Стандарты и несовместимость.

Сертификат X.509 – это аналог нашего паспорта. Здесь сертификаты вам выдаются суровой третьей стороной, гарантом вашей личности: Удостоверяющим Центром (УЦ). Получающий ваши подписи человек всегда может обратиться в УЦ и спросить интересующую его информацию по вот этому конкретному сертификату.

В свою очередь PGP (и стандарт OpenPGP, появившийся в дальнейшем) создавался на основе так называемых сетей доверия. Такая идея подразумевает, что обмениваются подписями люди, которым третья сторона для их взаимоотношений не нужна, а нужна только защита от нехороших лиц.

Конечно, с течением времени такое разделение стало уже достаточно условным, так как на данный момент и в S/MIME+X.509 и в PGP можно использовать методы лагеря соперников. Но все же, стандарты достаточно продолжительное время развивались параллельно и развились до той степени, что взаимная совместимость между ними стала невозможной.

Более популярным стандартом, в силу своей ориентированности на участие более компетентной третьей стороны, стал стандарт S/MIME + X.509, однако и у PGP есть некоторое количество козырей за пазухой, с помощью которых он не только не погибает, но и продолжает успешно развиваться.

Более подробное рассмотрение каждого из форматов, а также рекомендации, когда, где и какой из них использовать мы расскажем в следующих статьях.