Электронная цифровая подпись «для чайников». Часть 3

http://habrahabr.ru/blogs/infosecurity/98323/

В этом материале мы сделаем небольшое отступление от собственно цифровых подписей в сторону того, без чего ЭЦП, да и вообще защиты информации, в привычном понимании, скорее всего бы и не было — шифрования. Ведь первое, что приходит на ум, когда идет речь о защите данных — это не дать нехорошему человеку эти данные прочитать. Поэтому, перед тем, как продолжить рассмотрение стандартов PGP и S/MIME, стоит закрасить некоторые остающиеся в знаниях белые пятна, и рассмотреть процесс шифрования немного поподробнее.

Шифры и коды существуют, наверное, с того момента, как человечество научилось записывать свои впечатления об окружающем мире на носителях. Если немного вдуматься, даже обыкновенный алфавит — уже шифр. Ведь когда мы читаем какой-либо текст, в нашей голове каждому нарисованному символу сопоставляется некий звук, сочетание звуков, или даже целое понятие, а в голове соседа, который читать не умеет, этого уже не происходит.

Не зная, какому символу и что сопоставлено, мы никогда не сможем понять, что же именно автор имел в виду. К примеру, попробуйте взять и прочитать что-то, написанное на иврите, или на китайском языке. Сами алфавиты этих языков будут являться для вас непреодолимым препятствием, даже если с помощью этих символов написаны понятия вашего родного языка.

Но, тем не менее, простое использование чужого алфавита все же недостаточная мера для защиты ваших данных. Ведь любой алфавит, так или иначе, создавался для удобства пользования им и является неразрывно связанным с языком, которому данный алфавит характерен. А значит, выучив этот язык и некоторый набор базовых понятий на нем (а то и просто воспользовавшись услугами человека, знающего данный язык), недоброжелатель может прочитать вашу информацию. Соответственно, чтобы этого не произошло, нам надо придумать алфавит, который знает только ограниченный круг лиц, и записать информацию с его помощью.

Алфавиты и ключи.

Наверняка все читали (или, по крайней мере, слышали) цикл историй про Шерлока Холмса. В этом цикле фигурировал алфавит, составленный из пляшущих человечков (а многие, я думаю, в детстве на его основе составляли свой). Однако, как показывает данная история, наблюдательный человек может разгадать, какой символ и к чему относится. А значит, наша информация опять попадет не в те руки.

Что же делать? Придумывать все более и более сложные алфавиты? Но чем более сложный и громоздкий алфавит, тем более неудобно с ним работать, хранить его в тайне. К тому же, насчет тайны есть замечательная поговорка: знают двое – знают все. Ведь самое слабое звено в любом шифре – это человек, который знает, как этот шифр расшифровать.

А почему бы не сделать так, чтобы способ шифрования был сразу известен всем, но расшифровать наши данные было бы нельзя без какого-то ключа? Ведь ключ (в отличие от всего алфавита) маленький, его достаточно легко сделать новый, и легко спрятать если что (опять же, в отличие от переработки всего алфавита). Наиболее наглядно плюсы «ключевых» систем показывает следующий пример: получателю надо прочитать присланное вами сообщение. Обычное, на бумаге. Допустим, вы используете секретный алфавит. Тогда, чтобы прочитать сообщение, получатель должен знать алфавит, иметь большой пыльный талмуд, в котором описаны способы расшифровки (ведь алфавит должен быть сложным, чтобы быть надежным) и понимать, как же с этим талмудом работать. С ключами же все проще: вы кладете сообщение в коробку с замком, а получателю достаточно просто вставить подходящий ключик, а знать, как устроен замок ему совершенно но нужно.

Итак, общеизвестные «алфавиты» и ключи — механизм, существенно более удобный, чем просто алфавиты. Но как же так зашифровать, чтобы все расшифровывалось простым ключом? И вот тут нам на помощь приходит математика, а конкретнее – математические функции, которые можно использовать для замены наших исходных символов на новые.

Виды шифрования.

На всякий случай давайте вспомним, что такое функция. Это некоторое соотношение, по которому из одного числа можно получить другое. Зная x и подставляя его в известное нам соотношение y=A*x, мы всегда получим значение y. В ряде случаев верно и обратное: зная y, мы можем легко получить x, но далеко не всегда. Для многих зависимостей получить y легко, а вот решить обратную задачу, определить x – уже очень трудно и его получение займет продолжительное время. Вот именно на таких зависимостях и базируется используемое сейчас шифрование.

Вернемся непосредственно к самому шифрованию. Шифрование подразделяют на симметричное, асимметричное и комбинированное. Рассмотрим подробнее , в чем суть каждого из них.

Симметричное шифрование, по большому счету, достаточно слабо отличается от старого доброго секретного алфавита. Собственно говоря, отличается оно как раз наличием ключа – некоторой сравнительно маленькой последовательности чисел, которая используется для шифрования и расшифровывания. При этом, каждая из обменивающихся информацией сторон должна этот ключ знать и хранить в секрете. Огромным плюсом такого подхода является скорость шифрования: ключ, по сути, является достаточно простой и короткой инструкцией, какой символ, когда, и на какой надо заменять. И работает данный ключ в обе стороны (то есть с его помощью можно как заменить все символы на новые, так и вернуть все как было), за что такой способ шифрования и получил название симметричного. Столь же огромным минусом является именно то, что обе стороны, между которыми информация пересылается, должны ключ знать. При этом, стоит нехорошему человеку заполучить ключ, как он тут же прочитает наши столь бережно защищаемые данные, а значит проблема передачи ключа принимающей стороне встает, как говориться, в полный рост.

Асимметричное шифрование реализовано несколько хитрее. Здесь и у нас, и у нашего получателя есть уже два ключа, которые называют открытый и закрытый. Закрытый ключ отправитель и получатель сохраняют у себя (заметьте, каждый хранит только свой ключ, а значит, мы уже выходим за пределы действия той самой поговорки про двоих знающих). Открытый мы можем спокойно передавать кому угодно – наш закрытый, секретный, по нему восстановить нельзя. Итого, мы используем открытый ключ получателя для шифрования, а получатель, в свою очередь, использует свой закрытый ключ для расшифровывания. Плюс данного подхода очевиден: мы легко можем начать обмениваться секретной информацией с разными получателями и при этом практически ничем не рискуем (принимая условие, что наш получатель свой закрытый ключ не потерял/отдал и т.п., то есть не передал его в руки нехорошего человека). Но, без огромного минуса не обойтись. И здесь он в следующем: шифрование и расшифровывание в данном случае идут очень, очень и очень медленно, на два-три порядка медленнее, чем аналогичные операции при симметричном шифровании. Соответственно ресурсов на это шифрование тратится также значительно больше. Да и сами ключи для данных операций существенно длиннее аналогичных для операций симметричного шифрования, так как требуется максимально обезопасить закрытый ключ

от подбора по открытому. А значит, большие объемы информации данным способом шифровать просто невыгодно.

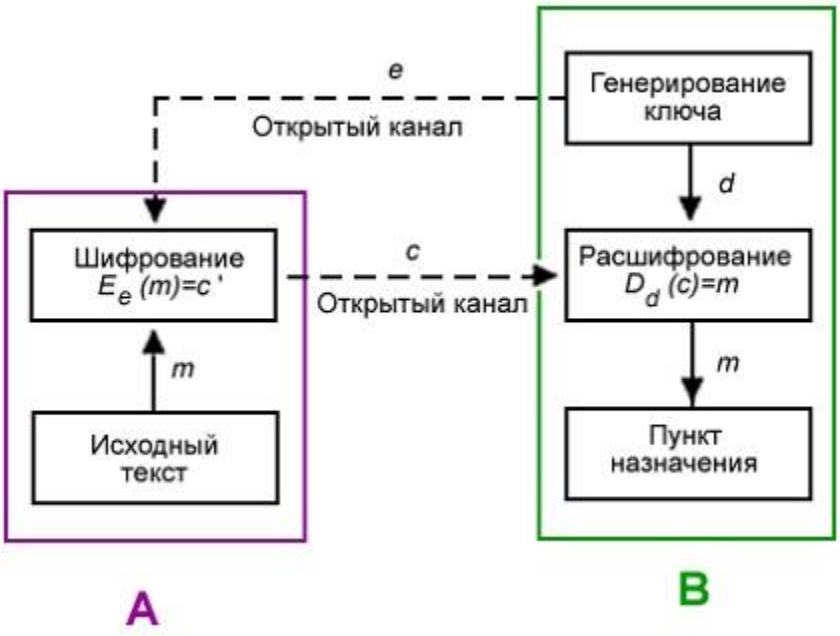

Рисунок 1. Пример использования асимметричного шифрования [использован материал из wikipedia.ru]

e — открытый ключ получателя B

d — закрытый ключ получателя B

m — исходная информация отправителя A

c — зашифрованная исходная информация

И снова возникает вопрос: что же делать? А делать нужно следующее: взять, и скомбинировать оба способа. Собственно, так мы и получаем комбинированное шифрование. Наш большой объем данных мы зашифруем по первому способу, а чтобы донести ключ, с помощью которого мы их зашифровали, до получателя, мы сам ключ зашифруем по второму способу. Тогда и получим, что хоть асимметричное шифрование и медленное, но объем зашифрованных данных (то есть ключа, на котором зашифрованы большие данные) будет маленьким, а значит расшифровывание пройдет достаточно быстро, и дальше уже в дело вступит более быстрое симметричное шифрование.

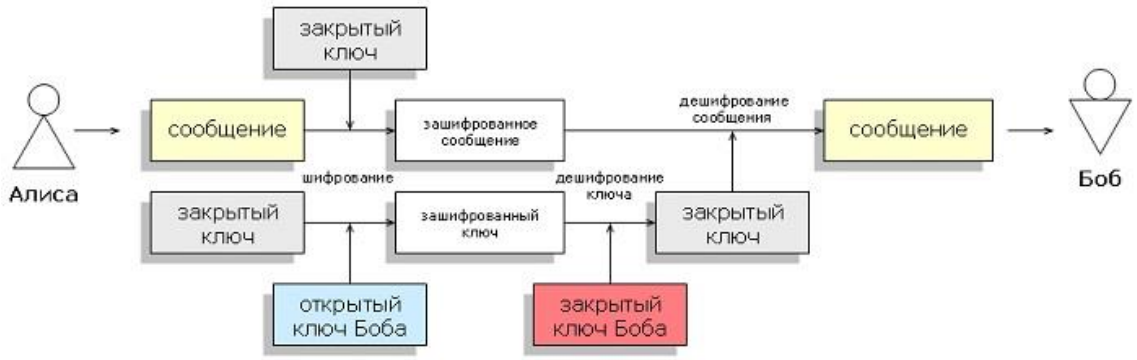

Рисунок 2. Пример применения комбинированной системы шифрования [источник:Wikipedia.ru]

Все перечисленные механизмы нашли свое применение на практике, и оба наших больших лагеря PGP и S/MIME их используют. Как говорилось в первой статье, асимметричное шифрование используется для цифровой подписи (а именно, для шифрования нашего хэша). Отличие данного применения от обычного асимметричного шифрования в том, что для шифрования используется наш закрытый ключ, а для расшифровывания достаточно наличие связанного с ним (то есть, тоже нашего) открытого ключа. Поскольку открытый ключ мы не прячем, наш хэш можем прочитать кто угодно, а не только отдельный получатель, что и требуется для цифровой подписи.

Комбинированное же шифрование применяется в обоих стандартах непосредственно для шифрования отправляемых данных.

Таким образом, начиная пользоваться цифровыми подписями для защиты наших данных от подмены, мы автоматически (для этих двух стандартов) получаем и замечательную возможность защитить наши данные еще и от прочтения, что, согласитесь, весьма удобно.

Теперь, познакомившись с общими технологиями, используемыми для защиты наших данных, мы можем наконец перейти к практике и выбору используемых механизмов. Но об этом уже в следующей статье.