

## 42. Шифрование данных. Электронная подпись

Автор: Александр  
26.08.2014 15:49

---

Продолжим тему защиты информации. Ранее уже было рассмотрено шифрование как один из способов защиты данных, передающихся через Интернет. Теперь поговорим об электронной цифровой подписи (ЭЦП).

Шифрование берет начало в глубокой древности. Так, еще Цезарь создал классический шифр, который носит его имя. А вот об истории электронной подписи сказать особо нечего, что, впрочем, и неудивительно, ведь людям издавна хватало печати и росчерка пером, так что на ум приходит лишь папирус с сургучной печатью, и потому ограничимся современностью.

В алгоритмах электронной подписи и асимметричного шифрования используются секретный и открытый ключи. Причем секретный должен браться абсолютно случайно, например с датчика случайных чисел, а открытый - вычисляться из секретного таким образом, чтобы получить второй из первого было невозможно. Итак, предположим, вы с другом решили ставить электронную подпись под всеми своими сообщениями. Теоретически нужно проделать следующее.

1. Сначала создайте ключи электронной подписи. Как и в случае шифрования, они обычно хранятся в файлах, в частности на дискетах. Каждый из вас должен иметь свои секретный и открытый ключи.
2. Секретные ключи оставьте у себя, а открытыми обменяйтесь.
3. Секретным ключом подпишите письмо другу и отправьте свое послание вместе с подписью. Электронная подпись представляет собой последовательность нескольких цифр. На первый взгляд она выглядит хаотично, на самом же деле вычисляется по следующей упрощенной формуле:

$f(M, k_s)$ ,

## 42. Шифрование данных. Электронная подпись

Автор: Александр  
26.08.2014 15:49

---

где  $M$  - текст письма; а  $k_s$  - секретный ключ.

4. Получив письмо, снабженное электронной подписью, адресат с помощью вашего открытого ключа проверяет ее подлинность. Результат проверки - один из ответов: "верна - неверна". Электронная подпись подтверждает достоверность сообщения. Если в него в процессе пересылки были внесены какие-либо изменения, пусть даже совсем незначительные, то подмена обнаружится.

Секретный ключ вы должны тщательно хранить в тайне, ведь любой, кто узнает его, сумеет подделать вашу подпись. Если вы все же потеряете свой ключ, то обязательно предпримите определенные меры и, главное, сообщите всем своим потенциальным адресатам о том, что вашу подпись, которую они считали верной, отныне следует считать неверной. А до тех пор, пока вы этого не сделаете, считайте, будто только что подписали пачку пустых листов бумаги.

Еще одно, не менее важное назначение электронной подписи - подтверждение авторства сообщения. Обычно в файлы ключей ЭЦП помимо собственно ключа записываются разные дополнительные сведения вроде ФИО и места работы его владельца, срока действия подписи и т. п. А в подпись, стоящую под сообщением или документом, копируются данные из секретного ключа, и прежде всего сведения о его хозяине, что позволяет установить авторство. Значит, не потребуются запоминать, кто именно прислал открытый ключ, при проверке показавший, что ЭЦП верна, и это очень важно, ведь реально может быть не одна сотня открытых ключей. Кстати, «правильные» программы при расчете собственно электронной подписи сообщения включают и информацию об авторе, чтобы никому не пришлось в голову изменить ее. Результат проверки ЭЦП обычно выводится на экран в таком, например, виде:

*Подпись файла compromat.bmp верна (Автор: Иванов Василий Семенович).*

Как и любые криптографические алгоритмы с открытым ключом, ЭЦП удобны для распределения ключей «на лету», что особенно хорошо в Интернете - вы можете послать свой открытый ключ любому адресату непосредственно перед отправкой ему подписанного вами сообщения или, что еще проще, разместить его на каком-либо ресурсе в Интернете. Однако позволю себе процитировать классиков защиты информации: «Принципы доступности, удобства, быстродействия и функциональности вычислительной системы антагонистичны принципам ее безопасности» (И. Д. Медведовский, П. В. Семьянов, Д. Г. Леонов. «Атака на Internet»). В общем, за удобство придется заплатить существенным ослаблением безопасности. Здесь, как и при асимметричном шифровании, возможна подмена открытых ключей, правда приводящая к иным последствиям.

## 42. Шифрование данных. Электронная подпись

Автор: Александр  
26.08.2014 15:49

---

Вот как это бывает: вы с другом создали по паре ключей и обменялись открытыми. Все было бы хорошо, но тут вмешался злобный хакер. Он перехватил отправленный правильный открытый ключ, причем таким образом, что до вашего друга ключ так и не дошел, прочитал ваши ФИО, а затем создал новую пару ключей (секретный плюс открытый), записав туда сведения о вас. Секретный ключ злоумышленник оставил у себя, а открытый отправил другу от вашего имени. Теперь хакер сможет посылать ему любые письма, а вашу подпись под его ложными сообщениями друг будет считать верной до тех пор, пока обман не выплывет наружу, но у вас с вашим товарищем могут возникнуть серьезные проблемы.

К счастью, есть способ борьбы с подменой открытых ключей - это их сертификация.

Сейчас существует множество алгоритмов ЭЦП, в том числе:

- отечественный стандарт электронной подписи ГОСТ Р34.10-94, который, как и стандарт симметричного шифрования ГОСТ 28147-89, обязателен для применения в государственных организациях России и обменивающихся с ними конфиденциальной информацией коммерческих организациях;
- новый отечественный стандарт ГОСТ Р34.10-2001, который должен заменить предыдущий с 1 июля 2002 г.
- различные общеизвестные алгоритмы ЭЦП, например RSA (Rivest - Shamir - Adleman), Эль-Гамала, DSA (Digital Signature Algorithm).

Приведенная выше формула для электронной подписи дана несколько упрощенно; в более полном виде она выглядит так:

$$S = f(h(M), k_s),$$

где  $h(M)$  - хэш-функция.

## 42. Шифрование данных. Электронная подпись

Автор: Александр  
26.08.2014 15:49

---

Дело в том, что текстовое письмо может иметь самый разный размер - от пустого сообщения (непонятно, правда, зачем нужно его подписывать) до объемного файла, к тому же включающего графику, а алгоритмы ЭЦП предназначены для подписи сообщений определенной длины, в частности, ГОСТ Р34.10-94 для 32 байт. Поэтому задача хэш-функции заключается в том, чтобы из письма произвольного объема вычислить цифровую последовательность стандартного размера, скажем, те же 32 байта, равных 256 бит. Хэш-функция обладает или, по крайней мере, должна обладать следующими свойствами.

1. сообщения (хэш - результат работы хэш-функции) должен однозначно соответствовать ему и изменяться при его модификации.

2. Хэш-функция должна быть однонаправленной. Тогда, во-первых, даже зная хэш  $h(M)$ , невозможно вычислить само сообщение  $M$  и, во-вторых, для каждого сообщения  $M$  нельзя подобрать такое сообщение  $M'$ , для которого выполнялось бы условие:

$$h(M) = h(M').$$

Невыполнение второго условия позволило бы злоумышленнику подменять письма, оставляя подпись в них верной. Кроме того, у многих сообщений хэш одинаковый, поскольку, как говорят математики, множество допустимых писем (их количество практически безгранично) существенно больше множества хэш-значений, максимально возможное число которых всего-навсего  $2^{256}$ . А теперь, выражаясь языком криптографии, иначе сформулируем приведенные выше условия: «Трудоёмкость успешного вычисления сообщения  $M'$  по уже известному хэшу  $h(M)$ , удовлетворяющему условию  $h(M') = h(M)$ , не должна быть меньше трудоёмкости прямого перебора сообщений».

Заметим, что хэш-функции также широко используются для аутентификации пользователей и появилась масса криптографических протоколов, основанных на их применении.

Отечественный стандарт для хэш-функций - ГОСТ Р34.11-94; он используется совместно со стандартами ГОСТ Р34.10 - 94/2001 для ЭЦП. Из западных алгоритмов для хэш-функций стоит упомянуть, например, ряд MD (Message Digest).

## 42. Шифрование данных. Электронная подпись

Автор: Александр  
26.08.2014 15:49

---

Поскольку шифрование защищает сообщения от ознакомления, а ЭЦП - от подмены (это две основные угрозы информации в Интернете), то было бы логично для обеспечения более полной безопасности совместно применять ЭЦП и комбинированное шифрование. Для этого нужно выполнить следующее.

1. На подготовительном этапе двое друзей, например, создают две пары ключей: секретный и открытый для асимметричного шифрования, а также секретный и открытый ключи ЭЦП. Открытыми ключами они обмениваются, а затем один посылает другому сообщение, подписанное своим секретным ключом.

2. Затем первый друг генерирует случайный ключ симметричного шифрования  $K$ , которым шифрует отправляемое письмо, причем только это.

3. Далее, чтобы можно было сообщение расшифровать, он зашифровывает ключ  $K$  (а в открытом виде посылать ключ симметричного шифрования ни в коем случае недопустимо) на открытом ключе асимметричного шифрования своего друга и добавляет его к зашифрованному письму.

4. Второй друг, получив зашифрованное сообщение, расшифровывает своим секретным ключом асимметричного шифрования ключ  $K$ , которым затем расшифровывает и само письмо.

5. И наконец, он проверяет с помощью открытого ключа друга его ЭЦП в данном письме и убеждается, что оно пришло именно от его друга и в неизменном виде.

Здесь придаться практически не к чему - при грамотном использовании подобная система не оставит хакеру никаких шансов на успех. Правда, может показаться неудобным то, что приходится делать слишком много ключей. Для решения этой задачи предусмотрен алгоритм Диффи-Хеллмана (названный так от имен его авторов Diffie и Hellman), позволяющий, в частности, применять одну и ту же пару ключей ЭЦП как для собственно ЭЦП, так и для симметричного шифрования. Смысл данного алгоритма заключается в следующем. В стандарте ГОСТ Р34.10-94 для ЭЦП открытый ключ

## 42. Шифрование данных. Электронная подпись

Автор: Александр  
26.08.2014 15:49

---

вычисляется из секретного:

$$K_p = a^{K_s} \text{ mod } p,$$

где  $a$  и  $p$  - некоторые общеизвестные большие числа (могут принимать значения до  $2^{1024}$ , и это ужасающе большое значение). Предположим, что есть пользователи 1 и 2, сгенерировавшие свои секретные ключи и вычислившие из них открытые:

$$K_{p1} = a^{K_{s1}} \text{ mod } p; \quad K_{p2} = a^{K_{s2}} \text{ mod } p.$$

После обмена открытыми ключами у каждого из них появилась пара ключей: свой секретный и чужой открытый, т. е. абонент 1 имеет ключи  $K_{s1}$  и  $K_{p2}$ , а абонент 2 -  $K_{s2}$  и  $K_{p1}$ .

. Теперь вспомним математику (радуйтесь школьники и студенты, знающие математику наизусть!) и представим, что будет, если абоненту 2 вдруг вздумается возвести в степень своего секретного ключа открытый ключ абонента 1:

$$(K_{p1})^{K_{s2}} = (a^{K_{s1}})^{K_{s2}} \text{ mod } p = (a^{K_{s2}})^{K_{s1}} \text{ mod } p = (K_{p2})^{K_{s1}} = K_c.$$

Впечатляет ли вас подобный результат? Ведь получилось то же самое, что вышло бы, если бы абонент 1 захотел проделать аналогичное с имеющимися у него открытым ключом абонента 2 и своим секретным ключом! Значит, существует ключ  $K_c$ , обычно называемый «ключом парной связи», который могут вычислить только абоненты 1 и 2 с использованием имеющихся у них ключей, поскольку у злоумышленников нет  $K_{s1}$

или  $K_{s2}$

, и поэтому они не сумеют определить  $K_c$ .

. Затем с помощью общего ключа  $K_c$

можно быстро симметрично зашифровать сообщения по стандарту ГОСТ 28147-89.