**Лекция 11. Методы и средства обеспечения целостности и достоверности используемого программного кода. Методы защиты программ от несанкциорированных изменений. Схема подписи с верификацией по запросу**

**Методы защиты программ от несанкционированных изменений**

Решение проблемы обеспечения целостности и достоверности электронных данных включает в себя решение, по крайней мере, трех основных взаимосвязанных задач: подтверждения их авторства и подлинности, а также контроль целостности данных, Решение этих трех задач в случае защиты программного обеспечения вытекает из необходимости защищать программы от следующих злоумышленных действий:

* РПС могут быть внедрены в авторскую программу или эта программа может быть полностью заменена на программу-носитель РПС;
* могут быть изменены характеристики (атрибуты) программы;
* злоумышленник может выдать себя за настоящего владельца программы;
* законный владелец программы может отказаться от факта правообладания ею.

Наиболее эффективными методами защиты от подобных злоумышленных действий предоставляют криптографические методы защиты. Это обусловлено тем, что хорошо известные способы контроля целостности программ, основанные на контрольной сумме, продольном контроле и контроле на четность, как правило, представляют собой довольно простые способы защиты от внесения изменений в код программ. Так как область значений, например, контрольной суммы сильно ограничена, а значения функции контроля на четность вообще представляются одним-двумя битами, то для опытного нарушителя не составляег труда найти следующую коллизию: *f(k1)=f(k2),* где k1 - код программы без внесенной нарушителем закладки, а k2 с внесенной программным закладкой и *f* - функция контроля. В этом случае значения функции для разных аргументов совпадают при тестировании и, следовательно, РПС обнаружено не будет.

Для установления подлинности (неизменности) программ необходимо использовать более сложные методы, такие как аутентификация кода программ, с использованием криптографических способов, которые обнаруживают следы, остающиеся после внесения преднамеренных искажении,

В первом случае аутентифицируемой программе ставится в соответствие некоторый аутентификатор, который получен при помощи стойкой криптографической функции. Такой функцией может быть криптографически стойкая хэш-функция (например, функция ГОСТ Р 34,11-94) или функция электронной цифровой подписи (например, функция из ГОСТ Р 34.10-94). И в том, и в другом случае аргументами функции может быть не только код аутентифицируемой программы, но и время и дата аутентификации, идентификатор программиста и/или предприятия - разработчика ПО, какой-либо случаиттый параметр и т.п. Может использоваться также любой симметричный шифр (например, DES или ГОСТ 28147-89) в режиме генерации имитовставки. Однако это требует наличия секретного ключа при верификации программ на целостность, что бывает не всегда удобно и безопасно. В то время как при использовании метода цифровой подписи при верификации необходимо иметь только некоторую общедоступную информацию, в данном случае открытый ключ подписи. То есть контроль целостности ПО может осуществить любое заинтересованное лицо, имеющее доступ к открытым ключам используемой схемы цифровой подписи.

Можно еще более усложнить действия злоумышленника по нарушению целостности целевых программ, используя схемы подписи с верификацией по запросу. В этом случае тестирование программ по ассоциированным с ними аутентификаторам можно осуществить только в присутствии лица, сгенерировавшего эту подпись, то есть в присутствии разработчика программ или представителей предприятия-изготовителя программного обеспечения. В этом случае, если даже злоумышленник и получил для данной программы некий аутентификатор, то ее обладатель может убедиться в достоверности программы только в присутствии специалистов-разработчиков, которые немедленно обнаружат нарушения целостности кода программы и (или) его подлинности.

Комбинация схем подписи и интерактивных систем доказательств позволила создать схемы подписи с верификацией по запросу. Такие схемы используются для обеспечения целостности и достоверности программного обеспечения.

# **Схема подписи с верификацией по запросу**

В работах Д. Шаума впервые была предложена схема подписи с верификацией по запросу, в которой абонент V не может осуществить верификацию подписи без участия абонента S. Такие схемы могут эффективно использоваться в том случае, когда фирма  изготовитель поставляет потребителю некоторый информационный продукт (например, программное обеспечение) с проставленной на нем подписью указанного вида. Однако проверить эту подпись, которая гарантирует подлинность программы или отсутствие ее модификаций, можно только уплатив за нее. После факта оплаты фирма - изготовитель дает разрешение на верификацию корректности полученных программ.

Схема состоит из трех этапов (протоколов), к которым относятся непосредственно этап генерации подписи, этап верификации подписи с обязательным участием подписывающего (протокол верификации) и этап оспаривания, если подпись или целостность подписанных сообщений подверглась сомнению (отвергающий протокол).

## **Схема ПВЗ**

Пусть каждый пользователь S имеет один открытый ключ Р и два секретных ключа S1 и S2. Ключ S1 всегда остается в секрете, - он необходим для генерации подписи. Ключ S2 может быть открыт для того, чтобы конвертировать схему подписи с верификацией по запросу в обычную схему электронной цифровой подписи.

Вместе с обозначениями секретного и открытого ключей

и (взятых из отечественного стандарта на электронную цифровую подпись) введем также обозначения S1=x и S2=u, и , а также открытый клоч P=(,y,w), где wu(*mod p*). Открытый ключ Р публикуется в открытом сертифицированном справочнике.

## **Протокол ГП**

Подпись кода m вычисляется следующим образом. Выбирается и вычисляется rk(*mod p*). Затем вычисляется s[xr+mku](*mod p*). Пара (r,s) является подписью для кода m. Подпись считается корректной тогда и только тогда, когда ru swy-rw(*mod р*) , где wm-1(*mod q*).

Проверка подписи (с участием подписывающего) осуществляется посредством следующего интерактивного протокола.

### **Протокол верификации ПВ**

Абонент V вычисляет rw y-rw(*mod р*) и просит абонента S доказать, что пара (r,s) есть его подпись под кодом m. Эта задача эквивалентна доказательству того, что дискретный логарифм по основанию r равен (по модулю р) дискретному логарифму w по основанию , то есть, что . Для этого:

1. Абонент V выбирает a,bRZq, вычисляет (*mod p*) и посылает абоненту S.
2. Абонент S выбирает tRZq, вычисляет ,  и посылает h1 и h2 абоненту V.

З. Абонент V высылает параметры а и b.

4. Если (*mod p*), то абонент S посылает V параметр t; в противном случае - останавливается.

5. Абонент V проверяет выполнение равенств и 

Если проверка завершена успешно, то подпись принимается как корректная.

### **Протокол ОП**

В отвергающем протоколе S доказывает, что . Следующие шаги выполняются в цикле *l* раз.

1. Абонент V выбирает , . Вычисляет , если =0 и , если =1. Посылает S значения а, b, d.
2. Абонент S проверяет соотношение . Если оно выполняется, то *a*=0, в противном случае *a*=1. Выбирает , вычисляет (*mod p*), и посылает V значение *с*.

З, Абонент V посылает абоненту S значение e.

1. Абонент S проверяет, что выполняются соотношения из следующих двух их пар: и . Если да, то посылает V значение R, иначе останавливается.
2. Абонент V проверяет, что (*mod p*).

Если во всех *l* циклах проверка в п.5 выполнена успешно, то абонент V принимает доказательства.

Теорема.1. Протокол верификации является интерактивным протоколом доказательств с абсолютно нулевым разглашением.

Доказательство: Требуется доказать, что вышеприведенный протокол удовлетворяет трем своиствам: полноты, корректности и нулевого разглашения.

Полнота означает, что если оба участника (V и S) следуют протоколу и (r,s) - корректная подпись для сообщения m, то V примет доказательство с вероятностью близкой к 1. Из описания протокола верификации очевидно, что абонент S всегда может надлежащим образом ответить на запросы абонента V, то есть доказательство будет принято с вероятностью 1.

Корректность означает, что если V действует согласно протоколу, то какие действия не предпринимал бы S, он может обмануть V лишь с пренебрежимо малой вероятностью. Здесь под обманом понимается попытка S доказать, что  когда на самом деле эти логарифмы не равны .

Предположим, что . Ясно, что для каждого *а* существует единственное значение b, то которое дает данный запрос . Поэтому не содержит в себе никакой информации об *а*. Если S смог бы найти h1, h2, tl и t2 такие, что

и

,

где a1 a2, то тогда выполнялось бы соотношение



Отсюда, очевидно, следует, что  В самом деле, пусть

. Тогда

(mod р) ,

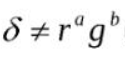
что противоречит предположению. Следовательно, какие бы hl, h2, t1, t2 не выбрал S, проверка, которую проводит V, может быть выполнена тол для одного значения *а*. Отсюда вероятность обмана не превосходит 1/q. Отметим, что протокол верификации является безусловно стойким для абонента V, то есть доказательство корректности не зависит ни от каких предположений о вычислительной мощности доказывающего (S).

Свойство нулевого разглашения означает, что в результат выполнения протокола абонент V не получает никакой полезной для себя информации (например, о секретных ключах, используемых S). Для доказательства нулевого разглашения необходимо для любого возможность проверяющего V\* построить моделирующую машину Мv\* которая является вероятностной машиной Тьюринга, работает за полиномиальное в среднем время и создает на выходе (без участия S) такое же распределение случайных величин, которое возникает у V\* в результате выполнет протокола. В нашем случае, случайные величины, которые «видит» V\* это hl, h2 и t. Необходимо доказать, что протокол верификации является доказательством с абсолютно нулевым разглашением, то е моделирующая машина создает распределение случайных величин (hl, h2, t) , которое в точности совпадает с их распределением, возникающим при выполнении протокола. Моделирующая машина Мv\* использует в своей работе V\* в качестве «черного ящика».

## **Моделирующая машина**

1. Запоминает состояние машины V\* , то есть содержимое всех ее лент, внутреннее состояние и позиции головок на лентах. Затем получает от V\* значение и после этого снова запоминает состояние машины V\* 
2. Выбирает и вычисляет и



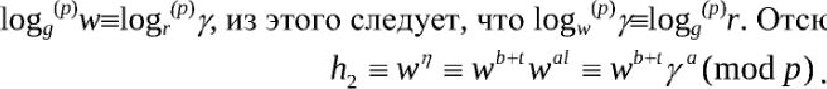
З. Получает от V\* значения а и b. Если (mod p) , то М я, останавливается.

1. Машина Мv\* «отматывает» V\* на состояние, которое было запомнено в конце шага 1. Выбирает и вычисляет .
2. Машина Мv\* передает V\* hl, h2 и получает ответ (a',b'). Возможны два варианта:
   1. а=а', b=b'. В этом случае моделирование закончено и Мv\* записывает на выходную ленту тройку (hl, h2, t) и останавливается.
   2. a или bb'. Отсюда следует, что . Предположим, что bb'.. Из этого следует, что a. Следовательно, Мv\* может вычислить

(mod р) . Отсюда  дискретный логарифм r по основанию .

1. Машина Мv\* «отматывает» V\* на состояние, которое было заполнено в начале шага 1. Получает от V\* значение .
2. Выбирает вычисляет  и  и передает их V\* .
3. Получает V\* от значения а и b. Если , то Мv\* останавливается. В противном случае вычисляет выдает на выходную ленту и останавливается.

К пп. 7 и 8 необходимо сделать следующее пояснение. Поскольку



Из описания моделирующеи машины Мv\* очевидно, что она работает за полиномиальное время. Величины (h1,h2,t), на шаге 5.1 выбираются в точности как в протоколе и поэтому имеют такое же распределение вероятностей. Кроме того, значения (h1,h2), выбираемые на шаге 7, имеют такое же распределение, как и в протоколе. Чтобы показать что и t имеет одинаковое распределение, достаточно заметить, что машина V\* не может по h1 и h2 определить, с кем она имеет дело - с S или Мv\* .

Поэтому, даже если бы V\* могла каким-либо «хитрым» образом строить а и b с учетом (hl,h2), распределение вероятностей величин а и b в обоих случаях одинаковы. Но значение t определяется однозначно четверкой величин а, b, h1, h2, при условии выполнения проверки на шаге 5 протокола.

**Теорема 2.** Отвергающий протокол является протоколом доказательства с абсолютно нулевым разглашением.

Доказательство. Полнота протокола очевидна. Если абоненты S и V следуют протоколу, тогда абонент V всегда примет доказательства абонента S,

Для доказательства корректности прежде всего заметим, что если то а и b, выбираемые абонентом V на шаге 1, не несут в себе никакой информации о значении . Поэтому, если S может ”открыть” *с*, сгенерированное им на шаге 2, лишь единственным образом (то есть выдать на шаге 4 единственное значение R, соответствующее данному *а*), то проверка на шаге 5 будет выполнена с вероятностью 1/2 в одном цикле и с вероятностью 1/2*l* во всех *l* циклах.

Если же S может сгенерировать с таким образом, что с вероятностью, которая не является пренебрежимо малой, он может на шаге 4 ”открыть” оба значения *а*, то есть найти Rl и R2 такие, что и , то алгоритм, который использует S для этой цели, можно использовать для вычисления дискретных логарифмов: . Так как при случайном выборе значения d логарифм может быть вычислен с вероятностью, которая не является пренебрежимо малой, это противоречит гипотезе о трудности вычисления дискретных логарифмов.

Далее доказывается, что отвергающий протокол является доказательством с абсолютно нулевым разглашением. Для этого необходимо для всякого возможного проверяющего V\* построить моделирующую машину Мv\*, которая создает на выходе (без участия S) такое же распределение случайных величин (в данном случае, *с* и R), какое возникает у V\* в результате выполнения протокола.

## **Моделирующая машина**

Следующие шаги выполняются в цикле *l* раз.

1. Машина Мv\* запоминаег состояние машины V\* 

2.Получает от V\* значения а, b и d.

З.Выбирает  и вычисляет . Посылает V\* значение с.

4.Получает от V\* значение е.

5.Проверяет, было ли «угадано» на шаге 2 значение *а* (это значение было «угадано», если и *а*=0, либо и *а*=1). Если да, то записывает на входную ленту значение (c,R). В противном случае «отматывает» V\* на то состояние, которое было запомнено на шаге 1, и переходит на шаг 2.

Легко видеть, что распределения случайных величин (c,R), возникающее в процессе выполнения протокола и создаваемые моделирующей машиной Мv\*, одинаковы, поскольку R в обоих случаях чисто случайная величина, а величина *с* записывается на выходную ленту машины Мv\*, только тогда, когда *а* совпало с .

Поскольку значение *а* выбирается машиной Мv\* на шаге 3 случайным образом, а *с* не дает V\* никакой информации о значении *а*, на каждой итерации *а* будет угадано с вероятностью 1/2. Отсюда следует, что машина Мv\* работает за полиномиальное в среднем время.

В работе [Ка14] показано, как строить схемы конвертируемой и селективно конвертируемой подписи с верификацией по запросу на основе отечественного стандарта ГОСТ Р 34.10-94. В таких схемах открытие определенного секретного параметра некоторой схемы подписи с верификацией по запросу позволяет трансформировать последнюю в обычную схему цифровой подписи. При этом открытие секретного параметра в конвертируемой схеме подписи с верификацией по запросу дает возможность верифицировать все имеющиеся и сгенерированные в дальнейшем подписи, в то время как в селективно конвертируемых схемах подписи с верификацией по запросу можно верифицировать лишь какую-либо одну подпись.

### **Примеры применения схемы подписи верификацией по запросу**

Предположим фирма-изготовитель программного обеспечения распространяет свою продукцию посредством электронной почты, На каждом экземпляре ПО проставляется некоторая подпись (тип подписи пока не определен, но в лобом случае подпись представляет собои значение функции, аргументами которой обязательно являются подписываемые данные и секретный клоч подписывающего). Эта подпись гарантирует, что программы являются подлинными (то есть, разработаны фирмой-изготовителем) и не были модифицированы, Однако верифицировать эту подпись, можно только уплатив за программный продукт. В этом случае фирма - изготовитель во взаимодействии с потребителем устанавливают корректность подписи, а значит и подлинность проданных программ.

Селективно конвертируемая схема подписи с верификацией по запросу можно использовать следующим образом. В рассмотрнном ранее сценарии для схемы подписи с верификацией по запросу (ПВЗ) предположим, что фирме изготовителю программного обеспечения необходимо, чтобы некоторые из сотрудников этой организации могли бы верифицировать ПВЗ, однако это надо сделать так, чтобы в дальнейшем они не могли генерировать подпись с использованием полученной информации о некоторых секретных параметрах схемы ПВЗ. Возможна ситуация, когда фирма изготовитель по каким либо причинам прекращает свою деятельность. Тогда она открывает для пользователеи свою секретную информацию для верификации подписи с тем, чтобы программы фирмы можно еще было бы безопасно использовать. Предположим также следующий сценарий. Та же фирма изготовила новую версию программного обеспечения, в то время как старая версия морально устарела и ее нет смысла использовать в дальнейшем. Однако, фирма желает убедить потребителей в своей надежности и в том, что все это время они использовали достоверные программы. В этом случае фирма публикует свой некоторый секретный параметр, позволяющий пользователям убедиться в оригинальности программ.