Лабораторная работа № 7

Тема: Система защиты от угроз нарушения целостности информации с использованием электронной подписи.

Цель работы: Освоить построение электронной подписи, позволяющей осуществлять подтверждение подлинности и целостности цифровых документов

7.1. Краткие теоретические сведения

При обмене электронными документами по сети связи возникает проблема аутентификации автора документа и самого документа, т.е. установления подлинности автора и отсутствия изменений в полученном документе, поскольку могут иметь место следующие действия злоумышленников:

– третий абонент (нарушитель), подключившийся к сети, перехватывает сообщения (документы) между первым и вторым абонентами с целью их скрытой модификации (активный перехват);

– третий абонент посылает документ второму абоненту от имени первого абонента (маскировка – имитация);

– первый абонент заявляет, что не посылал документ второму абоненту, хотя на самом деле он все-таки посылал (отказ – ренегатство);

– второй абонент изменяет или формирует новый документ и заявляет, что получил его от первого абонента именно в таком виде (модификация или подделка);

– третий абонент повторяет ранее переданный документ, который первый абонент посылал второму абоненту (повтор).

В обычной (бумажной) информатике эти проблемы решаются за счет того, что информация в документе и рукописная подпись автора жестко связаны с физическим носителем (бумагой). В электронных документах на машинных носителях проблема решается использованием электронной подписи (ЭП), которая представляет собой относительно небольшое количество дополнительной цифровой информации, передаваемой вместе с подписываемым текстом. Функционально ЭП аналогична обычной рукописной подписи и обладает ее основными достоинствами:

1. ЭП позволяет доказать, что именно законный автор, и никто другой, сознательно подписал документ.

2. ЭП представляет собой неотъемлемую часть документа. Невозможно отделить подпись от документа и использовать ее для подписывания других документов.

3. ЭП обеспечивает невозможность изменения подписанного документа и отказ от обязательств, связанных с подписанным текстом (в том числе и для самого автора).

4. ЭП делает невозможным отказ от авторства подписанного документа. Факт подписывания документа становится юридически доказуемым.

Использование электронной подписи позволяет осуществить:

1. Контроль целостности передаваемого документа: при любом случайном или преднамеренном изменении документа подпись станет недействительной, потому что вычислена она на основании исходного состояния документа и соответствует лишь ему.

2. Защиту от изменений (подделки) документа: гарантия выявления подделки при контроле целостности делает подделывание нецелесообразным в большинстве случаев.

3. Невозможность отказа от авторства. Так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец не может отказаться от своей подписи под документом.

4. Доказательное подтверждение авторства документа: Так как создать корректную подпись можно, лишь зная закрытый ключ, а он должен быть известен только владельцу, то владелец пары ключей может доказать свое авторство подписи под документом. В зависимости от деталей определения документа могут быть подписаны такие поля, как «автор», «внесенные изменения», «метка времени» и т.д.

Система ЭП включает две процедуры:

1. Процедуру постановки подписи, когда документ зашифровывается секретным ключом подписывающего.

2. Процедуру проверки подписи, когда получатель, используя общедоступный открытый ключ подписывающего, расшифровывает подпись, сличает ее с оригиналом и убеждается, что подпись верна.

В качестве подписываемого документа может быть использован любой файл. Подписанный файл создается из неподписанного путем добавления в него одной или более электронных подписей.

Каждая подпись содержит следующую информацию:

– дату подписи;

– срок окончания действия ключа данной подписи;

– информацию о лице, подписавшем файл (Ф.И.0., должность, краткое наименование фирмы);

– идентификатор подписавшего (имя открытого ключа);

– собственно электронную подпись.

Принципиальным моментом в системе ЭП является невозможность подделки ЭП пользователя без знания его секретного ключа подписывания.

При формировании ЭП отправитель прежде всего вычисляет хэш-функцию *h*(*М*) подписываемого текста *М*. Вычисленное значение хэш-функции *h*(*М*) представляет собой один короткий блок информации *m*, характеризующий весь текст *М* в целом. Затем число *m* шифруется секретным ключом отправителя. Получаемая при этом пара чисел представляет собой ЭП для данного текста *М*.

При проверке ЭП получатель сообщения снова вычисляет хэш-функцию принятого по каналу текста *М* : *m* = *h*(*М*), после чего при помощи открытого ключа отправителя проверяет, соответствует ли полученная подпись вычисленному значению *m* хэш-функции.

Секретный ключ хранится отправителем в тайне и используется им для формирования ЭП. Открытый ключ известен всем другим пользователям и предназначен для проверки ЭП получателем подписанного электронного документа. Иначе говоря, открытый ключ является необходимым инструментом, позволяющим проверить подлинность электронного документа и автора подписи. Открытый ключ не позволяет вычислить секретный ключ.

Для генерации пары ключей (секретного и открытого) в алгоритмах ЭП, как и в асимметричных системах шифрования, используются разные математические схемы, основанные на применении однонаправленных функций. Эти схемы разделяются на две группы. В основе такого разделения лежат известные сложные вычислительные задачи:

– задача факторизации (разложения на множители) больших целых чисел;

– задача дискретного логарифмирования.

К первой группе относится алгоритм шифрования RSА, ко второй – алгоритм Эль Гамаля.

7.2. Алгоритмы формирования электронной подписи

7.2.1. Алгоритм RSA.

Система RSА, математическая схема которой была разработана в 1977 г. в Массачуссетском технологическом институте США, названная по фамилиям авторов Rivest, Shamir, Adleman, стала первой и наиболее известной во всем мире конкретной системой ЭП.

Сначала необходимо вычислить пару ключей – секретный ключ и открытый ключ.

Для этого отправитель (автор) электронных документов

1. Выбирает два очень больших простых числа *P* и *Q*.

2. Вычисляет произведения *N* = *P* **·** *Q* и *F = (P –* 1*)* **·** *(Q –* 1*)*.

3. Случайным образом выбирает число *E* ≤ *F* и взаимно простое с *F* (т.е. не имеющее с ним общих делителей кроме единицы).

4. Вычисляет значение *D*, удовлетворяющее условиям

*D* < *N* , (*D* **·** *E*) *mod* *F* = 1.

Числа *E* и *N* будут открытым ключом отправителя, а значение *D* – секретным ключом. Открытый ключ передается автором партнерам по переписке для проверки его цифровых подписей. Число *D* сохраняется автором как секретный ключ для подписывания. Числа *Р* и *Q* больше не нужны, однако их нельзя никому сообщать; лучше всего их вообще забыть.

На этом этап подготовки ключей закончен, и можно использовать основной протокол RSA для формирования электронной подписи.

Допустим, что отправитель хочет подписать сообщение *М* перед его отправкой. Сначала сообщение *М* (блок информации, файл, таблица) сжимают с помощью хэш-функции *h*() в целое число *m*:

*m* = *h*(*М*).

Затем вычисляют электронную подпись *S* под электронным документом *М*, используя хэш-значение *m* и секретный ключ *D* :

*S* = *mD* *mod* *N*.

Пара (*М*, *S*) передается партнеру-получателю как электронный документ *М*, подписанный электронной подписью *S*, причем подпись *S* сформирована обладателем секретного ключа *D.*

После приема пары (*М*, *S*) получатель вычисляет хэш-значение сообщения *М* двумя разными способами. Прежде всего он восстанавливает хэш-значение *m*', применяя криптографическое преобразование подписи *S* с использованием открытого ключа *E*:

*m*' = *SE mod* *N*.

Кроме того, он находит результат хэширования принятого сообщения *М* с помощью такой же хэш-функции *h*():

*m* = *h*(*М*).

Если соблюдается равенство вычисленных значений, т.е.

*SE mod* *N* = *h*(*М*),

то получатель признает пару (*М*, *S*) подлинной.

Только обладатель секретного ключа *D* может сформировать цифровую подпись *S* по документу *М*, а определить секретное число *D* по открытому числу *E* не легче, чем разложить модуль *N* на множители.

Кроме того, результат проверки цифровой подписи *S* будет положительным только в том случае, если при вычислении *S* был использован секретный ключ *D*, соответствующий открытому ключу *E*. Поэтому открытый ключ *E* иногда называют «идентификатором» подписавшего.

Пример 7.1. Пусть автор документа А хочет передать его получателю Б. В этом случае вначале автор А должен подготовить открытый и закрытый ключи. Пусть им выбраны, например, следующие параметры:

*Р* = 3, *Q* = 11, *N* = 3 **·** 11 = 33.

Тогда *F* = (*P* – 1) **·** (*Q* – 1) = (3–1) **·** (11–1) = 20.

Затем автор А выбирает любое число *E* ≤ *F*, не имеющее общих делителей с *F* (это необходимо для того, чтобы зашифрованное сообщение можно было потом однозначно восстановить). Пусть *E* = 17. Это число будет одним из компонентов открытого ключа.

Далее необходимо найти число *D*, которое можно будет использовать в качестве секретного ключа. Значение *D* должно удовлетворять соотношению

*D* < *N* , (*D* **·** *E*) *mod* *F* = 1.

Для малых значений *F* число *D* можно найти подбором. В нашем случае подходит *D* = 13. (Проверяем: 13**·**17 *mod* 20 = 221 *mod* 20 = 1.)

Теперь автор А должен запомнить свой секретный ключ 13, отправить открытый ключ (17, 33) получателю Б и уничтожить числа *Р* = 3 и *Q* = 11.

Пусть в результате хэширования документа *М* будет получено хэш-значение *m* = 8.

Затем А вычисляет электронную подпись *S* под электронным документом *М*, используя хэш-значение *m* и секретный ключ *D*:

*S* = *mD* *mod* *N* = 813 *mod* 33 = 17.

Пара (*М*, *S*) передается партнеру-получателю Б. Он восстанавливает хэш-значение *m*', применяя криптографическое преобразование подписи *S* с использованием открытого ключа *Е* :

*m*' = *SE mod* *N* = 1717 *mod* 33 = 8.

Т.к. оно будет совпадать с хэш-значением *m* = 8, то документ *М* и электронная подпись *S* будут признаны подлинными.

7.2.2. Алгоритм Эль Гамаля.

В 1984 г. американцем арабского происхождения Тахером Эль Гамалем был разработан алгоритм электронной подписи (EGSA). В 1991 г. НИСТ США обосновал перед комиссией Конгресса США выбор алгоритма электронной подписи Эль Гамаля в качестве основы для национального стандарта.

Идея ЕGSА основана на том, что для обоснования практической невозможности фальсификации цифровой подписи может быть использована более сложная вычислительная задача, чем разложение на множители большого целого числа, – задача дискретного логарифмирования. Кроме того, Эль Гамалю удалось избежать явной слабости алгоритма цифровой подписи RSА, связанной с возможностью подделки цифровой подписи под некоторыми сообщениями без определения секретного ключа.

Рассмотрим алгоритм цифровой подписи Эль Гамаля. Для того чтобы генерировать пару ключей (открытый ключ – секретный ключ):

1. Выбирают некоторое большое простое целое число *Р* и большое целое число *G*, причем *G* < *Р*. Отправитель и получатель подписанного документа используют при вычислениях одинаковые большие целые числа *Р* (~10308 или ~21024) и *G* (~10154 или ~2512), которые не являются секретными.

2. Отправитель выбирает случайное целое число *X*, 1 < *Х* < (*Р* – 1), и вычисляет

*Y* = *GX* *mod* *Р*.

Число *Y* является открытым ключом, используемым для проверки подписи отправителя. Число *Y* открыто передается всем потенциальным получателям документов.

Число *Х* является секретным ключом отправителя для подписывания документов и должно храниться в секрете.

Для того чтобы подписать сообщение *М*, сначала отправитель хэширует его с помощью хэш-функции *h*() в целое число *m*:

*m* = *h*(*М*), 1<*m*<( *Р* – 1),

и генерирует случайное целое число *К*, 1 < *К* < (*Р* – 1), такое, что *К* и (*Р* – 1) являются взаимно простыми. Затем отправитель вычисляет целое число *а*:

*а* = *GK* *mod* *Р*

и, применяя расширенный алгоритм Евклида, вычисляет с помощью секретного ключа *Х* целое число *b* из уравнения

*m* = (*Х* **·** *а* + *К* **·** *b*) *mod* (*Р* – 1).

Пара чисел (*а*, *b*) образует цифровую подпись *S*:

*S* = (*а*, *b*),

которая проставляется под документом *М*.

Тройка чисел (*М*, *а*, *b*) передается получателю, в то время как пара чисел (*Х*, *К*) держится в секрете.

После приема подписанного сообщения (*М*, *а*, *b*) получатель должен проверить, соответствует ли подпись

S = (а, b)

сообщению *М*. Для этого получатель сначала вычисляет по принятому сообщению *М* число

*m* = *h*(*М*),

т.е. хэширует принятое сообщение *М*.

Затем получатель вычисляет значение

*А* = (*Ya***·***ab* ) *mod* *Р*

и признает сообщение *М* подлинным, только в том случае, если

*А* = (*Gm*) *mod* *Р*.

Т.е. получатель проверяет справедливость соотношения

(*Y*a**·**ab) *mod Р* = (*Gm*) *mod* *Р*.

Последнее равенство будет выполняться тогда, и только тогда, когда подпись *S* = (*а*, *b*) под документом *М* получена с помощью именно того секретного ключа *X*, из которого был получен открытый ключ *Y*. Таким образом, можно надежно удостовериться, что отправителем сообщения *М* был обладатель именно данного секретного ключа *X*, не раскрывая при этом сам ключ, и что отправитель подписал именно этот конкретный документ *М*.

Следует отметить, что выполнение каждой подписи по методу Эль Гамаля требует нового значения *К*, причем это значение должно выбираться случайным образом. Если нарушитель раскроет когда-либо значение *К*, повторно используемое отправителем, то он сможет раскрыть секретный ключ *Х* отправителя.

Пример 7.2. Выберем: числа *Р* = 11, *G* = 2 и секретный ключ *Х* = 8. Вычисляем значение открытого ключа:

*Y* = *GX* *mod* *Р* = 28 *mod* 11 = 3.

Предположим, что исходное сообщение *М* характеризуется хэш-значением *m* = 5.

Для того чтобы вычислить цифровую подпись для сообщения *М*, имеющего хэш-значение *m* = 5, сначала выберем случайное целое число *К* = 9. Убедимся, что числа *К* и (*Р* – 1) являются взаимно простыми. Действительно, числа 9 и 10 не имеют общих делителей кроме единицы. Далее вычисляем элементы *а* и *b* электронной подписи:

*а* = *GK* *mod* *Р* = 29 *mod* 11 = 6,

элемент *b* определяем, решая уравнение:

*m* = (*Х* **·** *а* + *К* **·** *b*) *mod* (Р – 1).

При *m* = 5, *а* = 6, *Х* = 8, *К* = 9, *Р* = 11 получаем

5 = (6**·**8+9**·**b) *mod* 10

или

9**·***b* = (–43) *mod* 10, 9**·***b* = 7 *mod* 10

Отсюда *b* = 3.

Следовательно, электронная подпись представляет собой пару: *а* = 6, *b* = 3. Далее отправитель передает подписанное сообщение (*M*, *a*, *b*).

Приняв подписанное сообщение и открытый ключ *Y* = 3, получатель вычисляет хэш-значение для сообщения *М* : *m* = 5,

а затем вычисляет два числа:

(*Ya***·***ab)* *mod* *Р* = (36 **·** 63) *mod* 11 =157464 *mod* 11 = 10;

*Gm* *mod* *Р* = 25 *mod* 11 =32 *mod* 11 = 10.

Так как эти два целых числа равны, принятое получателем сообщение признается подлинным.

7.3. Задание на лабораторную работу.

Разработать программу формирования и проверки электронной подписи, пользовательский интерфейс которой приведен на рис.7.1.

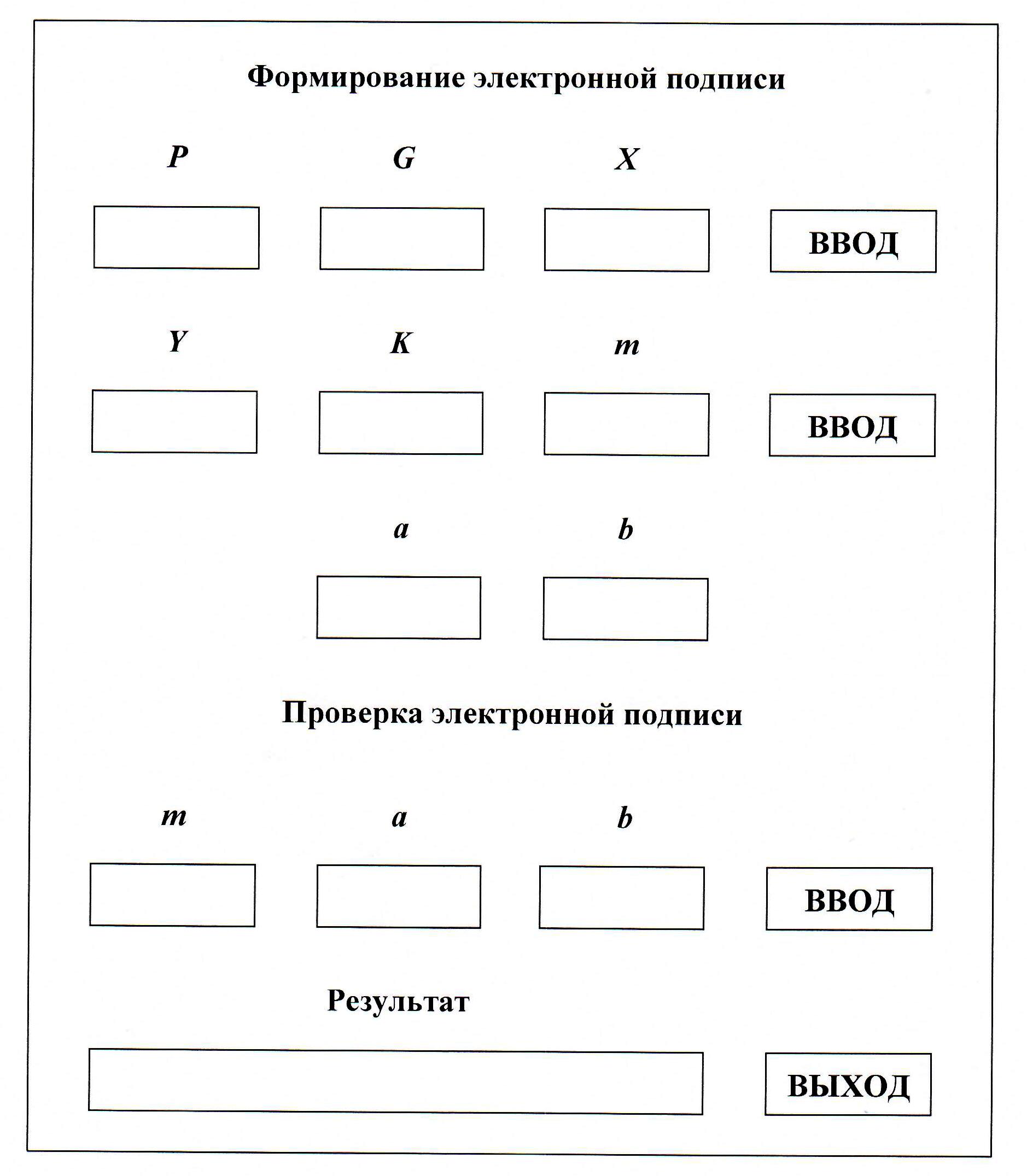


Рис. 7.1.

7.4. Краткое описание работы программы:

Формирование электронной подписи

1. В соответствующих окнах вводятся с клавиатуры:

– большое простое целое число *Р*;

– большое целое число *G*, удовлетворяющее условию *G* < *P*;

– целое число *X* – секретный ключ, удовлетворяющее условию

1 < *X* ≤ (*P* – 1).

2. Нажимается клавиша «Ввод».

3. Если какое-либо число (*P*, *G* или *X*) не удовлетворяет указанным условиям, то фон в соответствующем окне становится красного цвета, и ввод неправильного значения – повторяется.

Если введенные значения *P*, *G*, *X* удовлетворяют указанным условиям, то выполнение программы продолжается.

4. Вычисляется *Y* = *G* *X**mod* *P* – открытый ключ.

5. В соответствующих окнах вводятся с клавиатуры:

– заданное преподавателем целое число *m*, удовлетворяющее условию

1 < *m* < (*P* – 1);

– целое число *K*, удовлетворяющее условиям 1 < *K* < (*P* – 1), *K* и (*P* – 1) –

взаимно простые.

6. Нажимается клавиша «Ввод».

7. Если какое-либо число (*m* или *K*) не удовлетворяет указанным условиям, то фон в соответствующем окне становится красного цвета, и ввод неправильного значения – повторяется.

Если введенные значения *m*, *K* удовлетворяют указанным условиям, то выполнение программы продолжается.

8. Вычисляется и отображается в соответствующих окнах пара чисел (*a*, *b*), образующая электронную подпись S:

– *а* = *GK* *mod* *Р*;

– *b* как решение уравнения *m* = (*Х* **·** *а* + *К* **·** *b*) *mod* (Р – 1).

9. Считается, что *P*, *G*, *Y* не изменяются, и они известны при проверке электронной подписи, *m*, *a*, *b* могут изменять свои значения.

Проверка электронной подписи

1. В соответствующих окнах вводятся с клавиатуры значения *m*, *a*, *b* (сначала – с этапа «Формирование электронной подписи», а затем – отличные от этих значений).

2. Нажимается клавиша «Ввод».

3. Вычисляются

*A*1 = (*Y*a**·**ab) *mod Р* и *A*2 = (*Gm*) *mod* *Р*.

4.Если *A*1 = *A*2 , то в окне «Результат» выводится сообщение «Сообщение подлинное», если *A*1 ≠ *A*2  – то выводится сообщение «Сообщение недостоверное».

7.5. Контрольные вопросы

1. Какой ключ необходимо передать получателю сообщений для того, чтобы наладить обмен электронными сообщениями, имеющими электронную подпись?

2. Какова последовательность действий при подписании сообщений с помощью электронной подписи?

3. Какой ключ использует адресат для проверки подлинности электронной подписи отправителя?

4. Как называется действие, когда абонент С посылает документ абоненту В от имени абонента А?

5. Как называется действие, когда абонент А заявляет, что не посылал документ абоненту В, хотя на самом деле послал?

6. Что позволяет осуществить электронная подпись?

7. Какие сложные вычислительные задачи лежат в основе алгоритмов формирования электронной подписи RSА и Эль Гамаля?