ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ (ЭЦП)

электронная цифровая подпись=цифровая подпись (синонимы).

аутентификация информации, которая понимается как установление подлинности и неизменности сообщения, установления и доказуемости авторства сообщения, доказательства факта приема сообщения. Особенно важна эта услуга в сфере финансового и юридического электронного документооборота.

Целью аутентификации электронных документов является их защита от возможных видов злоумышленных действий, к которым относятся:

- ***активный перехват*** - нарушитель, подключившийся к сети, перехватывает документы (файлы) и изменяет их;
- ***маскарад*** - абонент С посылает документ абоненту В от имени абонента А;
- ***ренегатство*** - абонент А заявляет, что не посылал сообщения абоненту В, хотя на самом деле послал;
- ***подмена*** - абонент В изменяет или формирует новый документ и заявляет, что получил его от абонента А;
- ***повтор*** - абонент С повторяет ранее переданный документ, который абонент А посылал абоненту В.

Украинский стандарт электронной цифровой подписи: ДСТУ 4145-2002

§ 1. Определение ЭЦП. Ее свойства. Применение.

Электронная цифровая подпись – реквизит электронного документа, служащий для определения источника данных и защиты документа от подделки. Для получения ЭЦП выполняется криптографическое преобразования электронного документа, которое потом присоединяется к документу или логически объединяется с ним. Это дает возможность подтвердить его целостность и идентифицировать подписавшего.

Назначение ЭЦП:

1. ***Аутентификация лица***, подписавшего электронный документ.
2. ***Контроль целостности*** документа: при любом изменении документа подпись станет недействительной, ибо вычислена на основании исходного состояния документа и соответствует лишь ему.
3. ***Защита от изменений*** (подделки) документа.
4. ***Невозможность отказа от авторства*** (создать подпись может только владелец закрытого ключа, поэтому он потом не может отказаться от своей подписи под документом).
5. Доказательное ***подтверждение авторства*** документа.

Где используется ЭЦП: электронная торговля; таможенные декларации; регистрация сделок в банках; контроль исполнения

государственного бюджета, в системах обращения к органам власти; электронный документооборот.

История возникновения:

Понятие ЭЦП было введено в 1976 году Диффи и ХеллманОм, высказавшими предположение о ее существовании.

В 1977 году Ривест, Шамир и Адлеман разработали криптосистему RSA, которую можно было использовать для создания примитивных ЭЦП.

Вскоре после RSA разработаны алгоритмы цифровой подписи Рабина, Меркле.

В 1984 году Гольдвассер, Микали и Ривест сформулировали требования безопасности к алгоритмам ЭЦП, описали атаки на ЭЦП.

В 1994 году Главным управлением безопасности связи Федерального агентства правительственной связи и информации при Президенте России был разработан первый российский стандарт ЭЦП – ГОСТ Р 34.10-94. В 2002 году сменен на стандарт ГОСТ Р 34.10-2001, основанный на вычислениях в группе точек эллиптической кривой.

Украинский стандарт ЭЦП – ДСТУ 4145-2002.

Схема электронной подписи включает в себя:

- ***алгоритм генерации ключевой пары*** пользователя, который случайно выбирает закрытый ключ и вычисляет соответствующий ему открытый ключ;
- ***формирование подписи*** – для электронного документа с помощью закрытого ключа вычисляется подпись
- ***проверку (верификацию) подписи*** – для документа и подписи с помощью открытого ключа определяется действительность подписи.

В зависимости от алгоритма функция, формирующая подпись, может быть детерминированной или вероятностной. Детерминированные функции всегда находят одинаковую подпись для одного и того же документа. Вероятностные функции вносят в подпись элемент случайности, что усиливает криптостойкость алгоритмов ЭЦП. Но для вероятностных схем необходимы либо аппаратный генератор шума, либо криптографически надёжный генератор псевдослучайных битов, что усложняет их использование.

Различают одноразовые схемы ЭЦП, в которых после проверки подписи надо провести замену ключей, и многоразовые схемы, которые этого не требуют.

Различия между цифровой и собственноручной подписями

Собственноручная подпись	ЭЦП
Не зависит от текста, всегда одинакова	Зависит от текста, меняется
Неразрывно связана с подписывающим лицом, нельзя	Определяется секретным ключом, принадлежащим подписывающему.

утерять	Можно потерять
Неотделима от документа, отдельно подписывается каждый его экземпляр.	Легко отделима от документа. Верна для всех его копий.
Не требует дополнительных механизмов реализации	Требует дополнительных механизмов реализации алгоритмов подписи и проверки
Не требует создания поддерживающей инфраструктуры	Требует создания инфраструктуры сертификатов ключей

Алгоритмы ЭЦП делятся на два больших класса:

1) ***обычные цифровые подписи***, которые необходимо связывать с подписываемым сообщением.

2) ***цифровые подписи с восстановлением сообщения***, уже содержащие в себе подписываемый документ: в процессе проверки подписи автоматически вычисляется и тело документа.

§ 2. Две основные схемы построения ЭЦП

1. ***Симметричная схема ЭЦП*** на основе алгоритмов симметричного шифрования. Предусматривает наличие в системе третьего лица – ***арбитра***, пользующегося доверием обеих сторон. Аутентификацией документа является сам факт его шифрования секретным ключом и передача его арбитру. Используется редко (нет эффективных алгоритмов).

Преимущества:

- Стойкость симметричных схем вытекает из стойкости используемых блочных шифров, надежность которых также хорошо изучена.
- Если стойкость шифра недостаточна, его легко заменить другим.

Недостатки:

- Нужно подписывать отдельно каждый бит информации, что значительно увеличивает подпись (она может стать длиннее документа на порядок).
- Сгенерированные для подписи ключи можно использовать только один раз, так как после подписывания раскрывается половина секретного ключа.

2. ***Асимметричная схема ЭЦП*** на основе алгоритмов асимметричного шифрования. Они боле распространены и широко применяются в жизни. Если в асимметричных криптосистемах шифрование выполняют на открытом ключе, а дешифрование – на закрытом ключе получателя, то в схемах ЭЦП подпись производят с

помощью закрытого ключа, а проверку – с помощью открытого ключа пользователя, передающего сообщения (Схема ЭЦП меняет роли секретных и открытых ключей). Создать легитимную цифровую подпись без обладания закрытым ключом должно быть вычислительно сложно.

Недостатки:

- Асимметричные схемы ЭЦП базируются, как и асимметричное шифрование, на вычислительно сложных задачах (задаче дискретного логарифмирования или задаче факторизации), сложность которых строго математически не доказана. Поэтому прогресс в математике может вызвать из взлом. Вычисления могут производиться в группе точек эллиптических кривых (например, российская ЭЦП) и в полях Галуа (например, DSA).
- Для увеличения криптостойкости надо увеличивать длину ключей, что заставляет переписывать программы, реализующие схемы, и даже перепроектировать аппаратуру.

Кроме этого, существуют другие разновидности цифровых подписей (групповая подпись, неоспоримая подпись, доверенная подпись, слепая подпись).

§ 3. Использование хеш-функций для построения ЭЦП

Подписываемые документы имеют разную, часто большую длину, поэтому в схемах ЭЦП удобно использовать не сам документ, а его хэш. Хэша вычисляют с помощью криптографических хэш-функций, что гарантирует выявление изменений документа при проверке подписи.

Использование хеш-функций даёт преимущества:

- ***Выигрыш в вычислительной сложности***. Хэш документа имеет гораздо меньшую длину, чем сам документ, а алгоритмы вычисления хэша быстрее алгоритмов ЭЦП. Поэтому формировать хэш документа и подписывать его намного быстрее, чем подписывать сам документ.
- ***Совместимость***. Хеш-функцию можно использовать для преобразования произвольного входного текста в подходящий формат.
- ***Целостность***. Без использования хеш-функции большой документ в некоторых схемах надо делить на меньшие блоки, а потом подписывать. В этом случае при верификации невозможно определить, все ли блоки получены и в правильном ли они порядке. Хэширование снимает эти вопросы.

Замечание 1: хэширование не обязательно для цифровой подписи, а сама функция не является частью алгоритма ЭЦП, поэтому хэш-функция может использоваться любая или не использоваться вообще.

Замечание 2: все алгоритмы, подписывающие хэш документа, относятся к обычным ЭЦП.

§ 4. Угрозы цифровой подписи

Злоумышленник может:

- Подделать подпись для выбранного им документа.
- Подобрать другой документ к данной подписи.
- Подделать подпись для хоть какого-нибудь документа.
- Украв закрытый ключ, подписать любой документ от имени законного владельца ключа.
- Обманом заставить владельца подписать какой-либо документ, например, используя протокол слепой подписи.
- Подменить открытый ключ владельца на свой собственный, выдавая себя за него.

Создание поддельного документа сильно затрудняет хэширование подлинного документа перед подписыванием. Но угроза сохраняется, если хэш-функция слабая или неправильно использована.

§ 5. Схема цифровой подписи RSA

1 вариант – алгоритм подписи RSA с восстановлением сообщения.

Пусть пользователь **A** хочет подписать сообщение x и отправить его пользователю **B**.

Генерация ключей отправителя **A**: совпадает с генерацией ключей в алгоритме RSA. Выбираются два простых больших числа p и q, $n = pq$, выбирается нечетное число e, взаимно простое с числом $\varphi(n) = (p-1)(q-1)$; вычисляется $d \equiv e^{-1} (\mathrm{mod}\ \varphi(n))$. Открытый ключ – пара (n, e) объявляется публично, d – секретный ключ.

Формирование подписи отправителем **A**:

1. Вычислить подпись $S = x^d (\mathrm{mod}\, n)$.
2. Передать получателю **B** сообщение x и подпись S в виде пары (x, S).

Проверка подписи получателем **B**:

1. На открытом ключе отправителя **A** найти $x' = S^e (\mathrm{mod}\, n)$.
2. Если $x' = x$, то принять сообщение x как правильное.

Доказательство правильности процедуры:

$x' \equiv S^e (\mathrm{mod}\, n) \equiv (x^d)^e (\mathrm{mod}\, n) \equiv x (\mathrm{mod}\, n)$, так как $de \equiv 1 (\mathrm{mod}\ \varphi(n))$.

Пример. $n = 784319$ и $e = 313$ – открытые ключи пользователя **A**, $d = 160009$ – его закрытый ключ. Передать подписанное сообщение

$x = 19070$ пользователю **B**, который должен выполнить верификацию подписи.

Р е ш е н и е.

$S = (19070)^{160009} (\mathrm{mod}\, 784319) \equiv 210625$ Пару $x = 19070$ и $S = 210625$ передается пользователю **B**. Для проверки подписи пользователь **B** находит на открытом ключе пользователя **A**:

$$x' = (210625)^{313} (\mathrm{mod}\, 784319) \equiv 19070 \equiv x .$$

Подпись законна.

Для подписи длинного сообщения его придется разбить на блоки и подписывать каждый блок, что съедает много времени.

Атаки цифровой подписи RSA

1. ***Атака на основе известного открытого ключа***. Перехватив пару (x, S) и зная открытый ключ отправителя, криптоаналитик подбирает другое сообщение $x_1 = S^e (\mathrm{mod}\, n)$. По сложности решения это эквивалентно дискретному логарифмированию. Такое создание противником подписи какого-нибудь возможно бессмысленного сообщения называется ***экзистенциальной подделкой***. Обычно они бесполезны.
2. ***Атака на основе известных подписанных сообщений***. Пусть в руках криптоаналитика две пары подписи сообщений (x_1, S_1) и (x_2, S_2), полученные на одном закрытом ключе. Если $x = x_1 x_2$, то $S = S_1 S_2 (\mathrm{mod}\, n)$. Криптоаналитик посылает получателю новую пару (x, S), причем получатель может посчитать ее, посланной истинным владельцем ключей. Атака называется еще ***мультипликативной***. Сообщение x обычно не имеет смысла.
3. ***Атака по выбранным сообщениям*** тоже мультипликативная. Криптоаналитик подбирает два таких открытых текста x_1 и x_2, что $x = x_1 x_2$ будет нужным ему новым документом. Далее он может получить у законного отправителя подписи (x_1, S_1) и (x_2, S_2), а потом конструировать пару $(x_1 x_2, S_1 S_2)$, выдавая ее за подписанный документ отправителя. Такая атака называется еще ***селективной***, поскольку здесь создается подпись для заранее подобранного сообщения.

Один из способов противостоять атакам – подписывать хешированное сообщение и вносить зашумление.

2 вариант – обычная цифровая подпись по схеме RSA

Формирование подписи отправителем **A**:

1. С помощью криптографической функции хэширования вычислить хэш сообщения $H(x)$.
2. Вычислить подпись $S = H(x)^d \pmod n$.
3. Передать получателю **B** пару (x, S).

Проверка подписи получателем **B**:

1. На открытом ключе отправителя вычислить $H'(x) = S^e \pmod n$;
2. Вычислить для переданного сообщения x его хэш $H(x)$.
3. Проверка равенства $H(x) = H'(x) \pmod n$. Если оно верно, то подпись законна, в противном случае – незаконна.

Если криптоаналитик подготовит коллизию $H(x_1) = H(x_2)$, то он может дать на подпись сообщение x_2, а затем подменить им x_1, составив вместо пары (x_1, S) пару (x_2, S). $\Rightarrow$ подмена в схеме простой подписи RSA зависит от стойкости алгоритма хэширования к коллизиям.

Недостатки ЭЦП RSA.

1. При выборе ключей надо проверять много дополнительных условий, что сделать практически трудно (*Невыполнение любого из этих условий делает возможным фальсификацию цифровой подписи со стороны того, кто обнаружит такое невыполнение. При подписании важных документов нельзя допускать такую возможность даже теоретически*).
2. Чтобы обеспечить такую же криптостойкость ЭЦП, как и при шифровании DES (10^{18}), надо использовать числа порядка 2^{512} (или около 10^{154}) каждое, что требует больших вычислительных затрат.
3. ЭЦП RSA уязвима к мультипликативной атаке.

§ 6. Схема электронной подписи Эль-Гамаля (EGSA)

Название EGSA происходит от слов El Gamal_ Signature Algorithm (алгоритм цифровой подписи Эль-Гамаля).

Идея Эль-Гамаля – использование для ЭЦП задачи дискретного логарифмирования, вычислительно более сложной, чем факторизация чисел.

Пусть пользователь хочет подписать сообщение M и отправить его пользователю **B**.

Схема алгоритма

Генерация ключей

1. Необходимо выбрать большое простое целое число p (порядка ~10^{308} или ~2^{1024}) и какой-либо первообразный корень g по модулю p (размер g ~10^{154} или ~2^{512}).
2. Отправитель **A** выбирает случайное целое число x из интервала $(1; p-1)$ и вычисляет $y = g^x \pmod p$.

Числа p и g – ***параметрами домена*** (т.е. открытые параметры системы, используемые одновременно многими пользователями). ***Открытый ключ отправителя*** – тройка чисел (p, g, y), передается всем потенциальным получателям документов. ***Закрытый ключ отправителя*** – число x держится в секрете.

Формирование подписи отправителем **A**.

1. Вычисляется хэш сообщения $H = H(M)$. При этом должно выполняться условие $1 < H(M) < p-1$.

2. Выбирается случайное число r (это рандомизатор, держится в секрете) из интервала $(1; p-1)$, взаимно простое с $p-1$, и вычисляется

$$a = g^r \pmod p.$$

Находится число b из уравнения $H = ax + rb \pmod{p-1}$, т.е.

$$b = (H - ax) r^{-1} \pmod{p-1}.$$

3. Подписью сообщения M является пара (a, b). Получателю отправляется тройка $\langle M, a, b \rangle$.

Проверка подписи получателем **B**:

Зная открытый ключ (p, g, y), подпись (a, b) сообщения M проверяется следующим образом:

1. Проверяется выполнимость условий: $0 < a < p$ и $0 < b < p-1$. Если хотя бы одно из них не выполнено, то подпись – незаконна.
2. Вычисляется хэш сообщения $H = H(M)$.
3. Вычисляется $y^a a^b \pmod p$ и $g^H \pmod p$.
4. Подпись принимается только при условии, что

$$y^a a^b \equiv g^H \pmod p.$$

Математические обоснование законности проверки подписи:

$$y^a a^b \pmod p \underset{y \equiv g^x,\ a \equiv g^r}{\equiv} g^{xa} g^{rb} \pmod p \equiv g^{xa+rb} \pmod p \underset{b = (H-ax)r^{-1} + (p-1)n,\ n \in Z}{\equiv}$$

$$\equiv g^{H + (p-1)nr} \pmod p \equiv g^H \pmod p$$ (учитывая, что g – первообразный корень по модулю p, $g^{\varphi(p)} \equiv 1 \pmod m$, $\varphi(p) = p-1$). $\Rightarrow$ отправителем

сообщения M был обладатель именно данного секретного ключа x и отправитель подписал именно этот документ.

Замечание 1. Рандомизатор r должен сразу уничтожаться после вычисления подписи, так как, зная подпись (a,b) под известным m, из значения b находим секретный ключ:

$$b=(H-ax)r^{-1}(\mathrm{mod}\,p-1)\ \Rightarrow$$

$$x=(Hr^{-1}-b)a^{-1}r(\mathrm{mod}\,p-1)=(H-br)a^{-1}(\mathrm{mod}\,p-1).$$

Замечание 2. Следить, чтобы у числа $p-1$ был большой простой множитель, так как в противном случае удается провести логарифмирование.

Замечание 3. Условие $a<p$, проверяемое получателем, важно, так как в противном случае правильную подпись (a,b) для $H=H(M)$ можно переделать в подпись для другого значения $H'=H(M')$. Для этого криптоаналитик вычисляет элемент $\beta=H'H^{-1}(\mathrm{mod}\,p-1)$. Тогда $g^{H'}\equiv g^{\beta H}\equiv y^{\beta a}a^{\beta b}$. Полагая $b'=\beta b(\mathrm{mod}\,p-1)$, $a'=\beta a(\mathrm{mod}\,p-1)$, $a'=a(\mathrm{mod}\,p)$, можно по китайской теореме об остатках найти целое число $a'<p(p-1)$, при этом (a',b') будет подписью для $H'=H(M')$.

Пример. $p=11$, $g=2$ – параметры домена. Секретный ключ отправителя $x=8$. Вычислить третье число его открытого ключа и подписать сообщение M, хэш которого $H=5$. Провести верификацию подписи.

Р е ш е н и е. $y=g^x\bmod p=2^8(\mathrm{mod}11)=3$. Открытый ключ $(11,2,3)$. Хэш сообщения $H(M)=5$. Пусть рандомизатор $r=9$. Убеждаемся, что НОД$(k,\ p-1)=$ НОД$(9,10)=1$. Вычисляем

$$a=g^r(\mathrm{mod}\,p)=2^9(\mathrm{mod}11)=6$$

$$b=(H-ax)r^{-1}(\mathrm{mod}\,p-1)=(5-6\cdot 8)\cdot 9^{-1}(\mathrm{mod}10)\equiv 3(\mathrm{mod}10).$$

Подпись $(6,3)$. Подписанное сообщение $\langle M,6,3\rangle$ отсылаем получателю. Тот выполняет проверку подписи. Находит хэш $H=5$ и вычисляет два числа, зная все три части открытого ключа отправителя.

$$y^a a^b(\mathrm{mod}\,p)=3^6\cdot 6^3(\mathrm{mod}11)\equiv 10\ ;$$

$$g^H(\mathrm{mod}\,p)=2^5(\mathrm{mod}11)=10.$$

Эти числа равны, принятое сообщение признается подлинным.

Преимущества ЭЦП Эль-Гамаля:

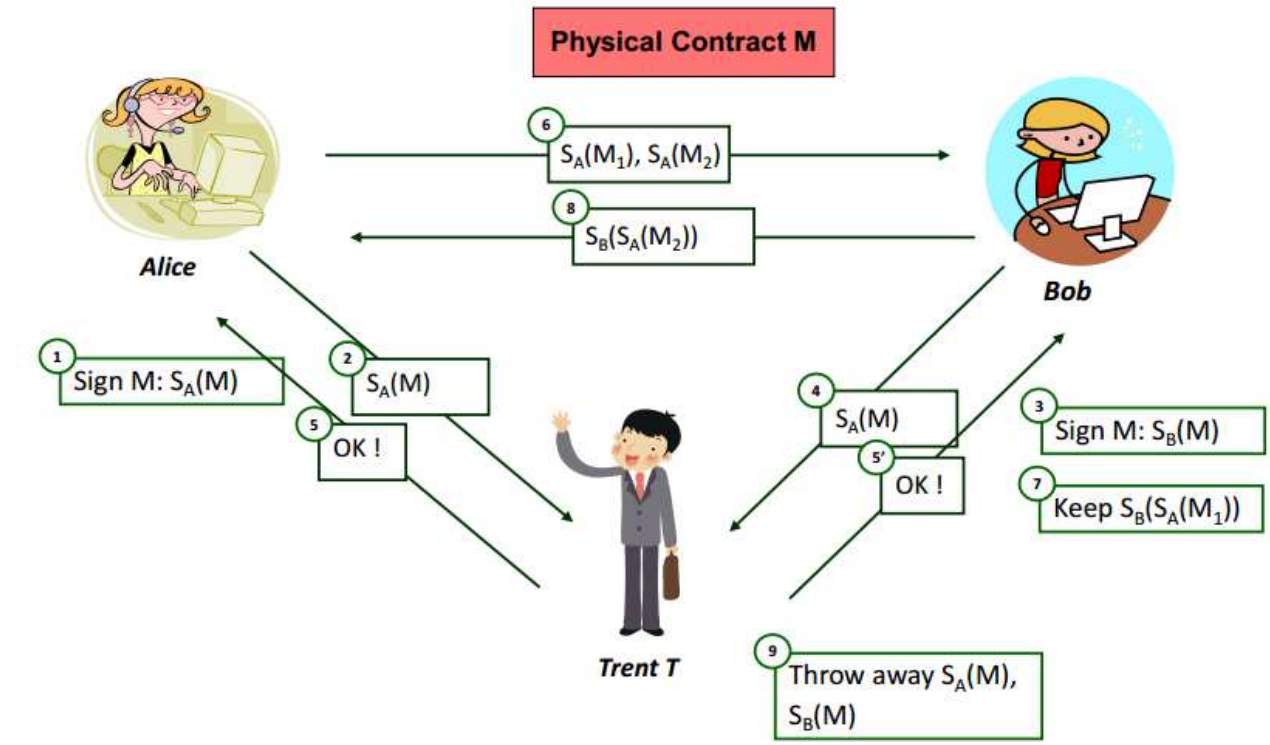
1. При одинаковом уровне стойкости используемые в вычислениях числа на 25% короче, что уменьшает сложность вычислений почти в 2 раза;
2. Нельзя вычислять цифровые подписи под новыми сообщениями без знания секретного ключа (как это было в ЭЦП RSA).
3. Требуется одно простое большое число p, а в алгоритме RSA – два.

Недостатки ЭЦП Эль-Гамаля – длина цифровой подписи в 1,5 раза больше, чем в RSA, а это увеличивает время вычисления.

Цифровая подпись Эль-Гамаля стала примером построения других подписей, схожих по своим свойствам. В их основе лежит выполнение сравнения : $y^A a^B = g^C \pmod p$,в котором тройка (A,B,C) принимает значения одной из перестановок $\pm a, \pm b, \pm H$ при каком-то выборе знаков. Например исходная схема Эль-Гамаля получается при $A = a$, $B = b, C = H$. На таком принципе построения подписи сделаны стандарты цифровой подписи США и России:

- в DSS (Digital Signature Standard), американском стандарте, используется значения $A = a$, $B = -b$;
- в Российском стандарте $A = -a, B = -H$, $C = b$. Сейчас в России используется новый (ГОСТ Р34.10-2001), принятый в 2001 году, на основе арифметики эллиптических кривых, определенных над простым полем Галуа.

Несмотря на то, что алгоритм Эль-Гамаля выглядит самым простым в этой цепи, он обеспечивает достаточную криптостойкость.



29 октября 2010 года, спустя 8 лет после принятия стандарта, **Украина, наконец, обрела первую версию работающей библиотеки для генерации ключей, генерации и проверки цифровой подписи по стандарту ДСТУ 4145-2002**! Ура

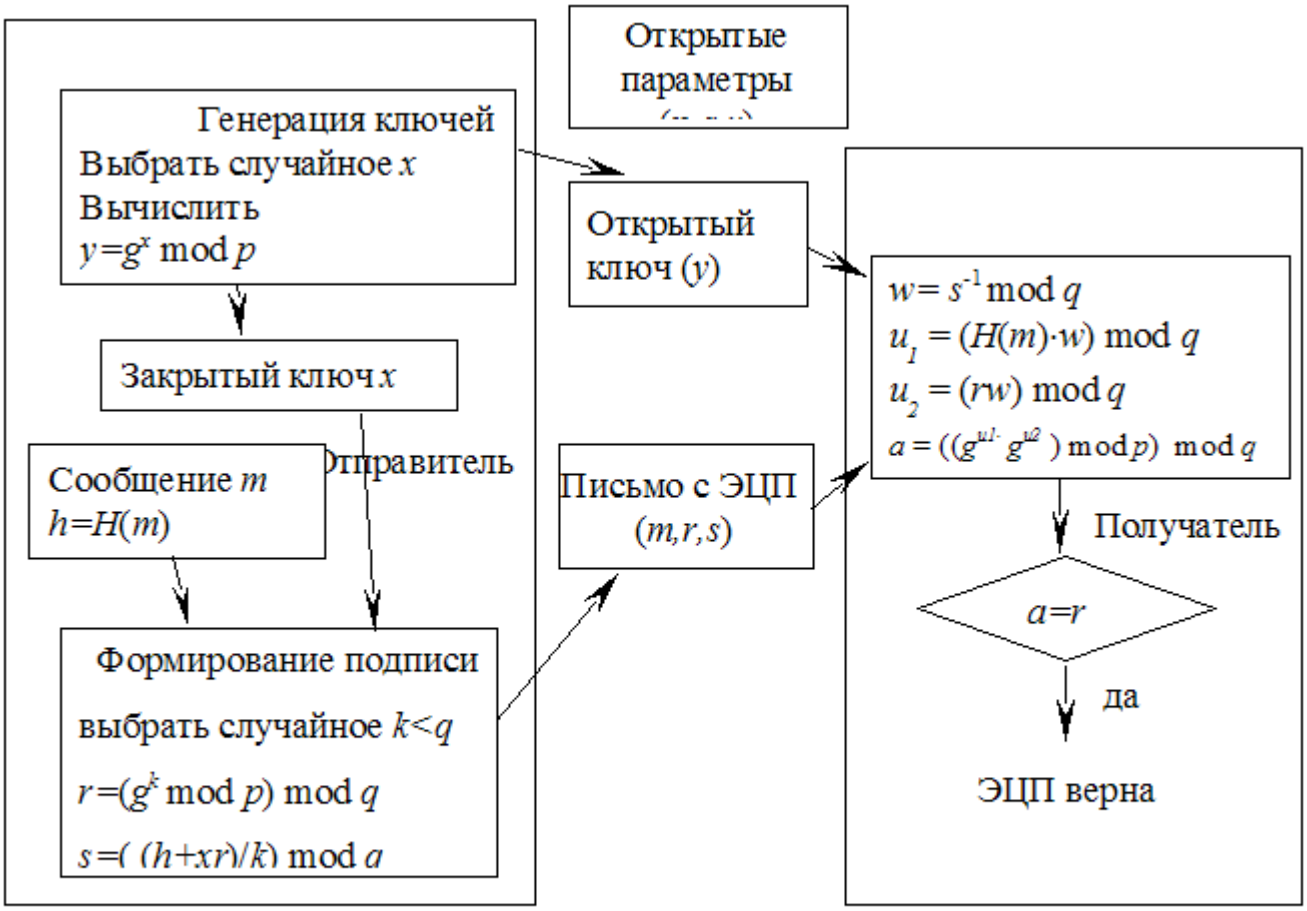


Рис.4.2. Схема алгоритма ЭЦП DSA

товарищи!

12. Discuss the advantages and disadvantages of RSA and El Gamal for their use for encryption and digital signatures.

ElGamal encryption and signatures require twice the bandwidth overhead of RSA as a mask needs to be sent in addition to the cipher text / signature

EL Gamal encryption and decryption have the same computational complexity. Similarly ElGamal signing an verification have the same complexity. For RSA we can choose small public exponent e (if e is 3 bits only three square and multiply loops are required to compute the exponent). However d will still be of the same order of the modulus n (for example, 2000 bits). Thus, in RSA

1) encryption can be fast but decryption slow (comparable to ElGamal encryption and decryption)
2) signing is slow (comparable to ElGamal signing and verification) but verification can be fast.

Low verification complexity is an especially desirable feature as high verification complexity can kead to easy denial of service attacks (an attacker can send random bits as signature- that the signature is random will be recognized only after the expensive verification process)

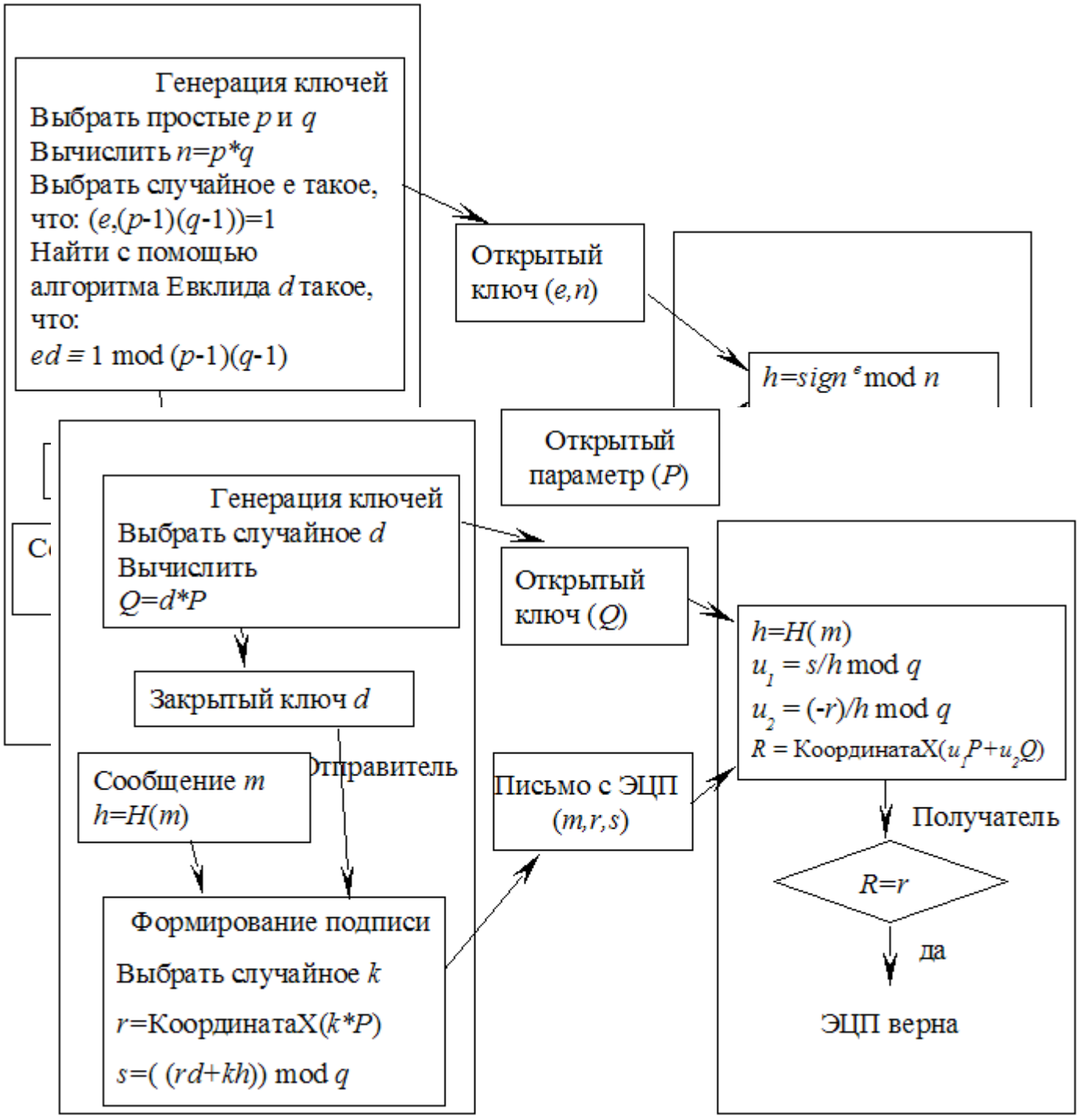


Рис.4.2. Схема алгоритма ЭЦП ГОСТ Р 34.10-2001