Электронная цифровая подпись (ЭЦП)

Угрозы в фокусе темы

Модель протокола формирования и проверки ЭЦП

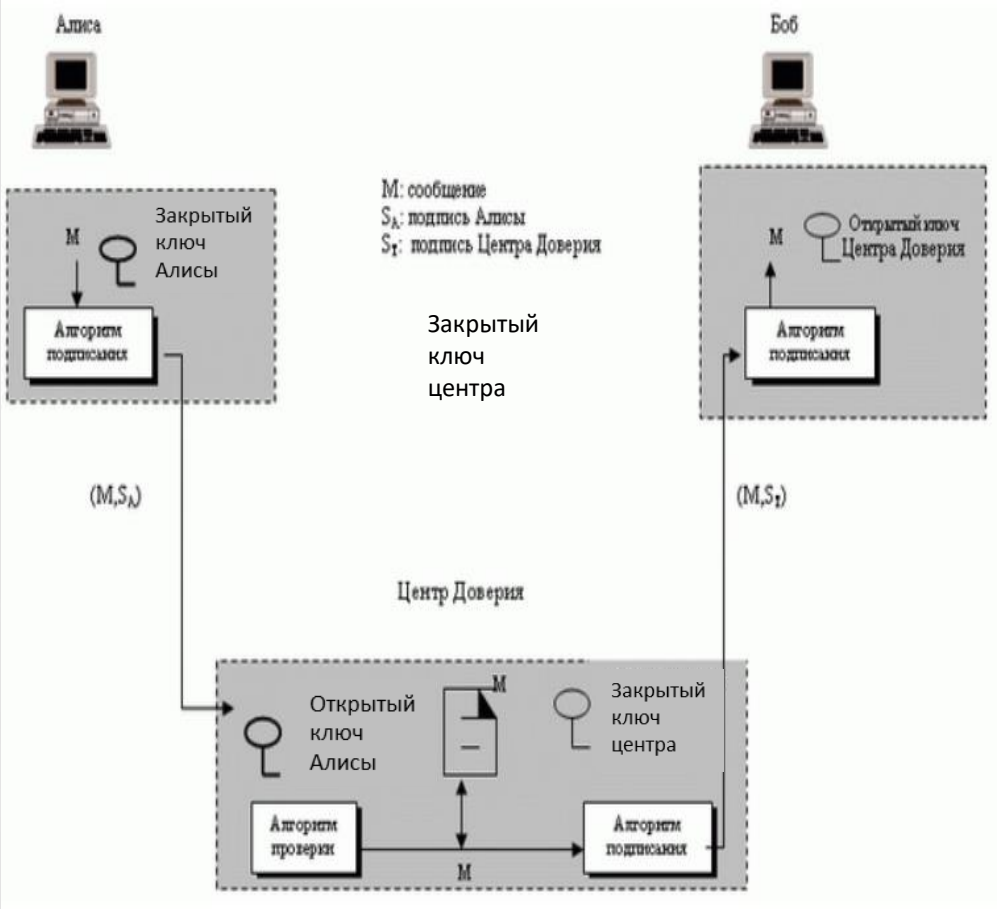
Сравнение рукописных и цифровых подписей

- Отношение к документу : В случае обычной подписи есть отношения "один ко многим" между подписью и документами. В случае цифровой подписи разным документам (сообщениям) соответствуют существенно различные значения подписи
- Метод проверки: В случае обычной подписи получатель сравнивает подпись на документе с эталоном. При цифровой подписи получателю предоставляется сообщение и алгоритм проверки, но копия эталона подписи нигде не хранится

Цифровая подпись против угроз

- Защита от модификации сообщения - Целостность сообщения контролируется, если криптопреобразованию подвергается все сообщение, поэтому нельзя получить ту же самую подпись, если сообщение изменено.
- Защита от имитации источника сообщения – установление подлинности цифровой подписи возможно, поскольку цифровая подпись создается с помощью персонального (закрытого ключа) отправителя

Цифровая подпись против угроз



- Исключение отказа от авторства сообщения:
  - Отравитель создает подпись из своего сообщения (S_A) и передает в Центр доверия, свой идентификатор, идентификатор получателя, а также подпись.
  - Центр проверяет по подписи и с помощью ключа отправителя, что источник сообщения правильный. Затем Центр сохраняет копию сообщения с подписью, идентификаторами отправителя и получателя, а также с меткой времени, в своем архиве.
  - Центр использует свой секретный ключ, чтобы создать из сообщения с подписью другую подпись (S_t). Затем Центр передает получателю сообщение, новую подпись, идентификатор отправителя и получателя.
  - Получатель проверяет сообщение, используя общедоступный ключ Центра, которому он доверяет.

Виды подделок цифровой подписи

Экзистенциальная
(existential forgery)

- Противник, НЕ владеющий закрытым ключом, создает пару (сообщение, подпись), которая будет принята алгоритмом проверки цифровой подписи
- Противник никак не контролирует выбор того сообщения, для которого в итоге будет подделана подпись – очень вероятно, это сообщение будет бессмысленным

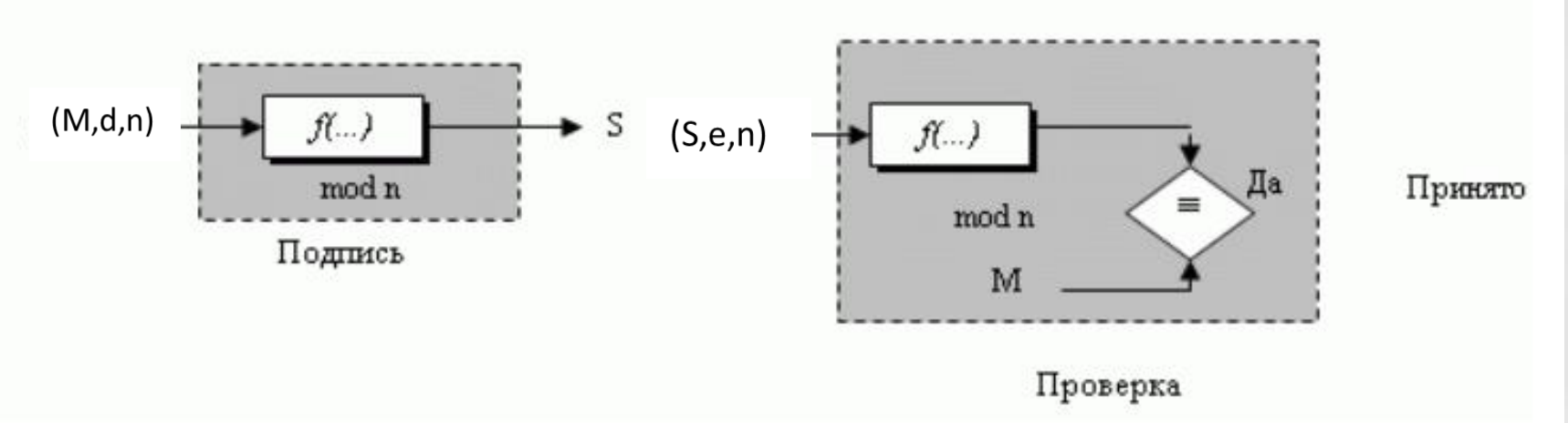
Селективная
(*selective forgery*)

- Противник, НЕ владеющий закрытым ключом, выбирает осмысленное сообщение (отсюда название угрозы)
- Далее, получив открытый ключ, пытается подделать цифровую подпись для этого выбранного сообщения.

Цифровая подпись RSA

Схема цифровой подписи RSA

- Схема цифровой подписи меняет роли закрытых и открытых ключей:
  - Используются открытый (*e*) и закрытый (*d*) ключи отправителя
  - Отправитель использует свой собственный закрытый ключ (d) для подписи документа.
  - Получатель использует открытый ключ отправителя (e), чтобы проверить подпись документа



RSA генерация ключей

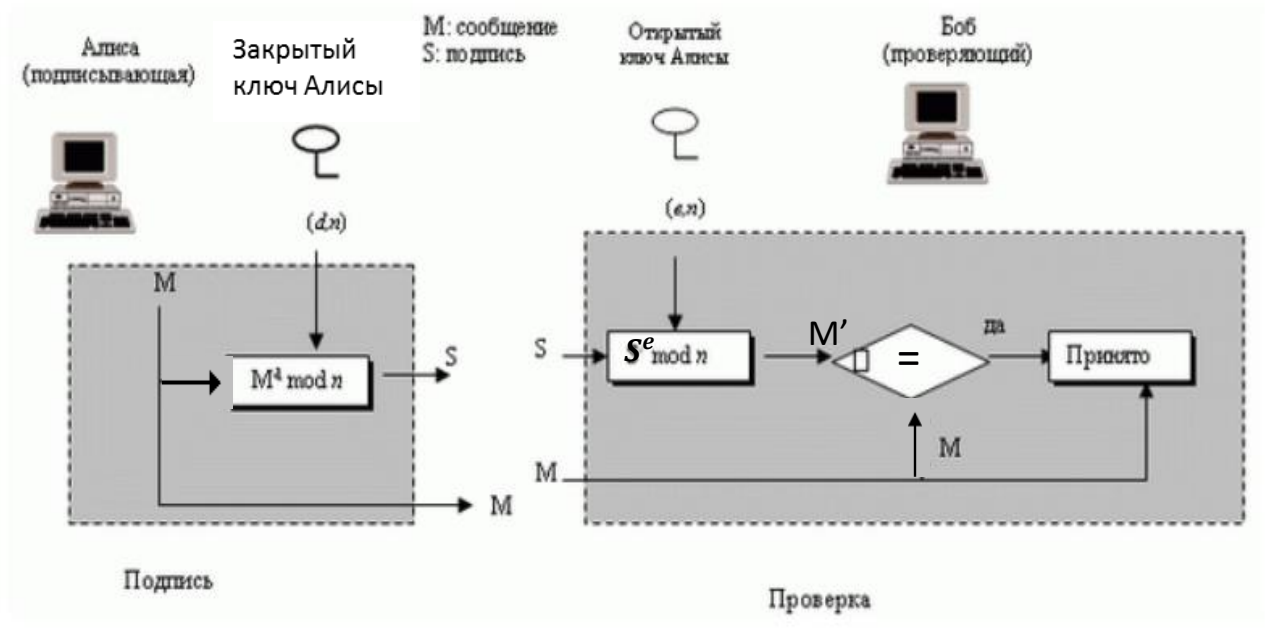
- Выбираются два больших простых числа *p* и *q*
- Вычисляется *n=p*q*
- Выбирается произвольное число *e (e<n)*, взаимно простое с *(p-1)*(q-1)*
- Вычисляется закрытый ключ (расширенный алгоритм Евклида) :

$$e \times \boldsymbol{d} \equiv 1\ mod\big((p-1)*(q-1)\big) \equiv 1\ mod\ (p-1)*(q-1)$$

Пара чисел *(e, n)* объявляются открытым ключом, *d* выбирается закрытым ключом

- *p* и *q* нужно уничтожить

RSA подписание и проверка



- Формирование подписи отправителем:
  - Ключ подписания (закрытый ключ) – пара чисел *(d, n)*
  - $S=(M^d) mod\ n$
- Проверка подписи получателем:
  - Ключ проверки (открытый ключ) – пара чисел *(e, n)*
  - $M' = (S^e) mod\ n$
  - Если $M' \equiv M\ mod\ n$ подпись верна

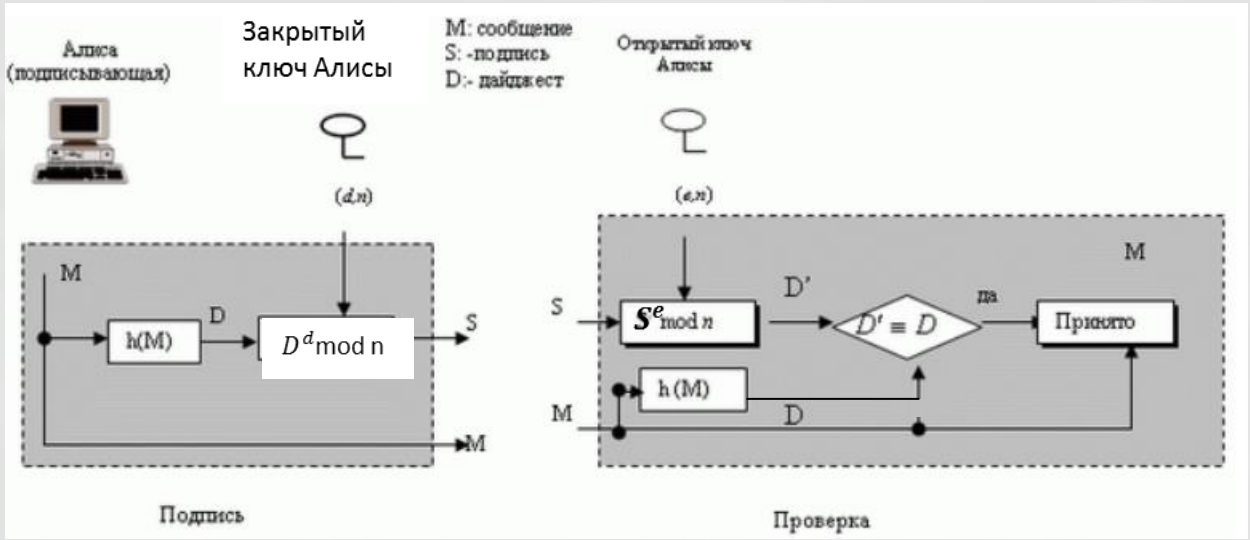
Примечание

- Подписи, созданные с применением алгоритма RSA, называются *детерминированными,* так как для одного и того же сообщения с использованием одного и того же закрытого ключа каждый раз будет создаваться одна и та же подпись

Подделка цифровой подписи RSA

- Экзистенциальная подделка. Перехватываются две пары (M_1, S_1), (M_2, S_2).Подписи созданы с помощью одного ключа d. Создается новое сообщение $M = M_1 \times M_2$ и соответствующая подпись $S = S_1 \times S_2 = M_1^d \times M_2^d = (M_1 \times M_2)^d = M^d$
- Селективная подделка. Целенаправленно создается $M = M_1 \times M_2$ и с помощью обмана отправителя противник получает подписи S_1 и S_2, что позволяет ему сформировать $S = S_1 \times S_2$ (если использовался один и тот же ключ).

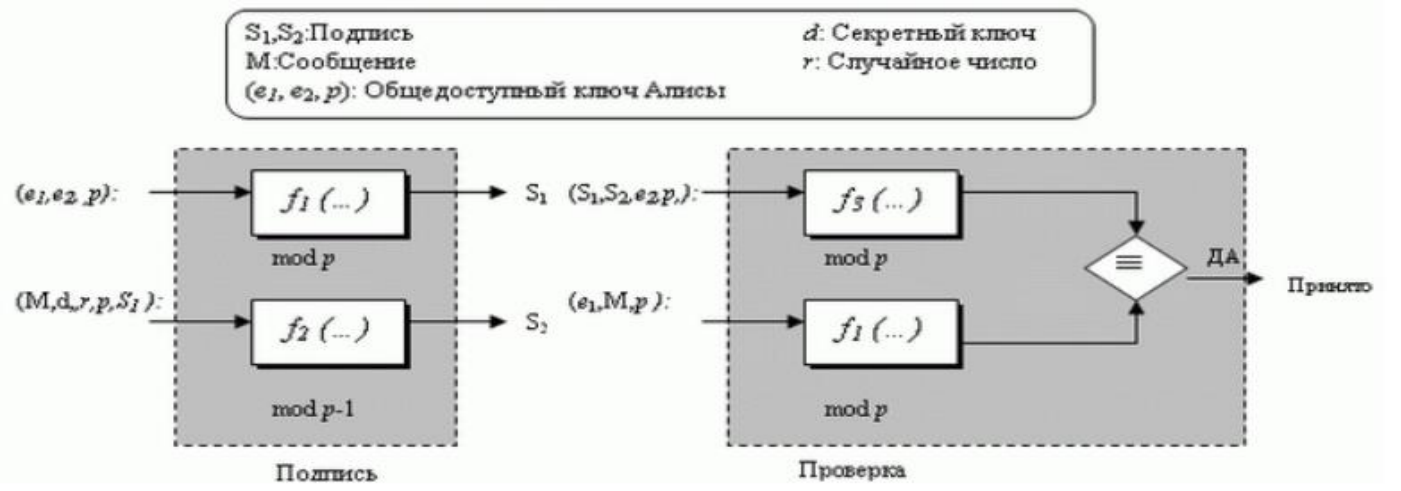
RSA подпись на дайджесте сообщения



- Экзистенциальная подделка. Перехватываются две пары $(M_1, S_1), (M_2, S_2)$. Подписи созданы с помощью одного ключа. Создается подпись $S = S_1 \times S_2$. Атака будет успешной, если найдется сообщение M, такое, что $h(M) = h(M_1) \times h(M_2)$ (зависит от устойчивости хэш-функции к прообразу)
- Селективная подделка. Целенаправленно создается $M = M_1 \times M_2$ и с помощью обмана отправителя противник получает подписи S_1 и S_2, что позволяет ему сформировать $S = S_1 \times S_2$ (если использовался один и тот же ключ). Атака будет успешной, если найдется осмысленное сообщение M' близкое по смыслу к М, такое, что $h(M') = S^e mod\ n$ (зависит от устойчивости хэш-функции к прообразу)

Цифровая подпись Эль-Гамаля

Схема цифровой подписи Elgamal

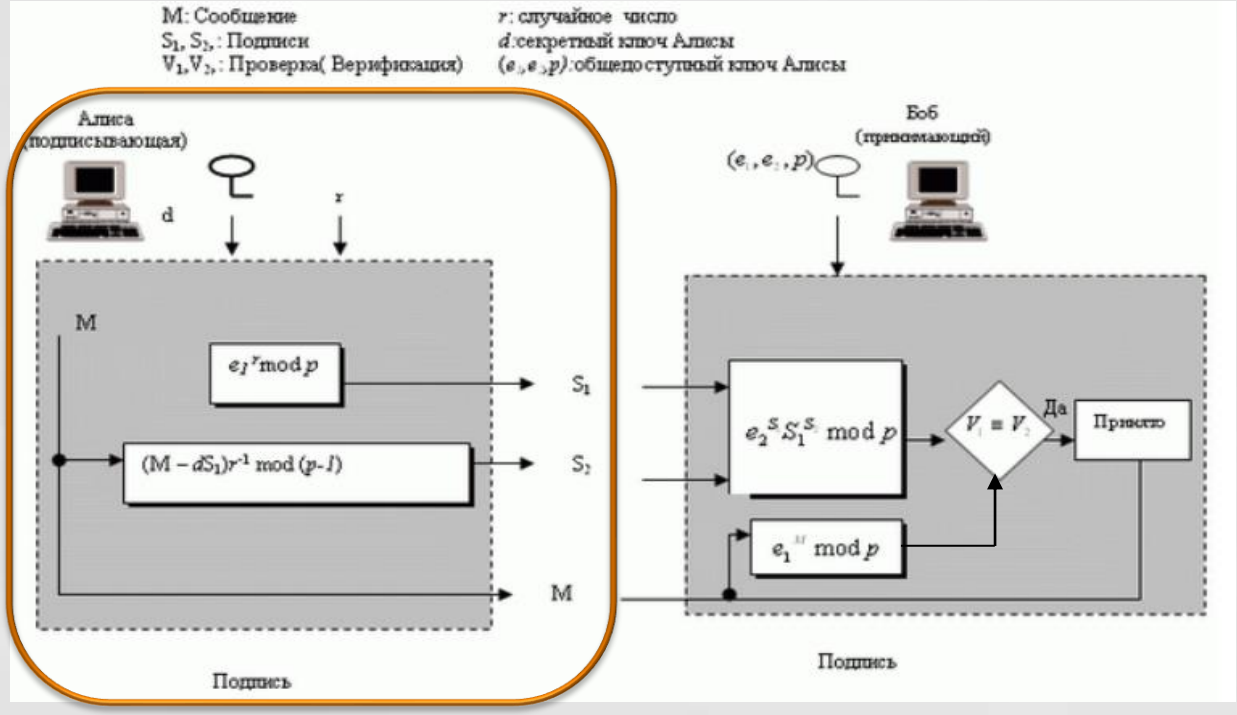


- В процессе подписания две функции f_1 и f_2 создают две части подписи
- В процессе проверки выходы двух функций f_1 и f_3 сравниваются между собой
- Сообщение присутствует на входе f_2 , при подписании (отправляемое сообщение), а также часть входа к функции f_1 при подтверждении (полученное сообщение)
- Вычисления в функциях f_1 и f_3 проводятся по модулю p, а функции f_2 - по модулю $p - 1$

Elgamal генерация ключей

- Генерируется случайное простое число p
- Выбирается целое число e_1 такое, что $1 < e_1 < p$, и e_1-первообразный корень p
- Выбирается случайное целое число d такое, что $1 < d < p$
- Вычисляется $e_2 = e_1{}^d mod\ p$
- Открытым ключом объявляется тройка (e_1, e_2, p)
- Закрытым ключом назначается число d

Elgamal подписание

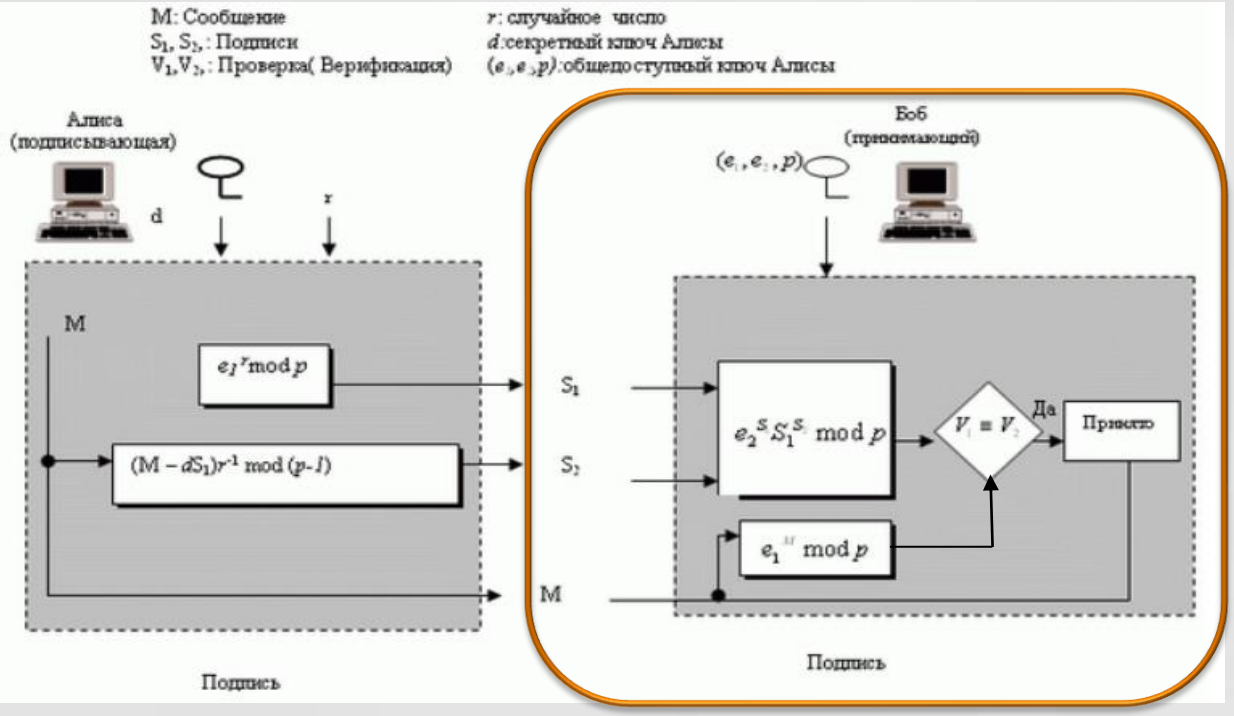


- Выбирается секретное случайное число r
- Вычисляется (f_1) первая часть подписи $S_1 = {e_1}^r mod\ p$
- Вычисляется (f_2) вторая часть подписи

$$S_2 = (M - d \times S_1) \times r^{-1} mod(p-1),$$

где r^{-1} - мультипликативная инверсия r по модулю $(p-1)$

Elgamal проверка



- Проверяем :
  - $0 < S_2 < p$
  - $0 < S_1 < p - 1$
- Вычисляем (f_1):
  - $V_1 = {e_1}^M mod\ p$
- Вычисляем (f_3):
  - $V_2 = {e_2}^{S_1} \times {S_1}^{S_2}\ mod\ p$
- Если $V_1 \equiv V_2 mod\ p$ подпись действительна

Обоснование критерия проверки

- Ранее принято:

$$e_2 = {e_1}^d mod\ p, S_1 = {e_1}^r mod\ p, V_1 = {e_1}^M mod\ p,\ V_2 = {e_2}^{S_1} \times {S_1}^{S_2}\ mod\ p$$

- Заменим критерий $V_1 \equiv V_2 mod\ p$ на эквивалентный (подстановками)
- $${e_1}^M \equiv {e_2}^{S_1} \times {S_1}^{S_2}\ mod\ p \equiv ({e_1}^d)^{S_1} \times ({e_1}^r)^{S_2} mod\ p \equiv {e_1}^{dS_1 + rS_2} mod\ p$$

- Поскольку e_1 - первообразный корень, то можно доказать, что полученное сравнение справедливо тогда и только тогда, когда

$$\mathrm{M} \equiv (dS_1 + rS_2) mod\ (p-1)\ ,\quad \text{поэтому}$$

$$S_2 \equiv \left((M - d \times S_1) \times r^{-1}\right) mod (p-1)$$

- Получен тот же результат, с которого начато подписание

Примечание

- Подписи, созданные с использованием алгоритма Elgamal называются *рандомизированными*, так как для одного и того же сообщения с использованием одного и того же закрытого ключа каждый раз будут создаваться разные части подписи (S_1, S_2), поскольку будет использоваться новое значение r

Подделка цифровой подписи Elgamal

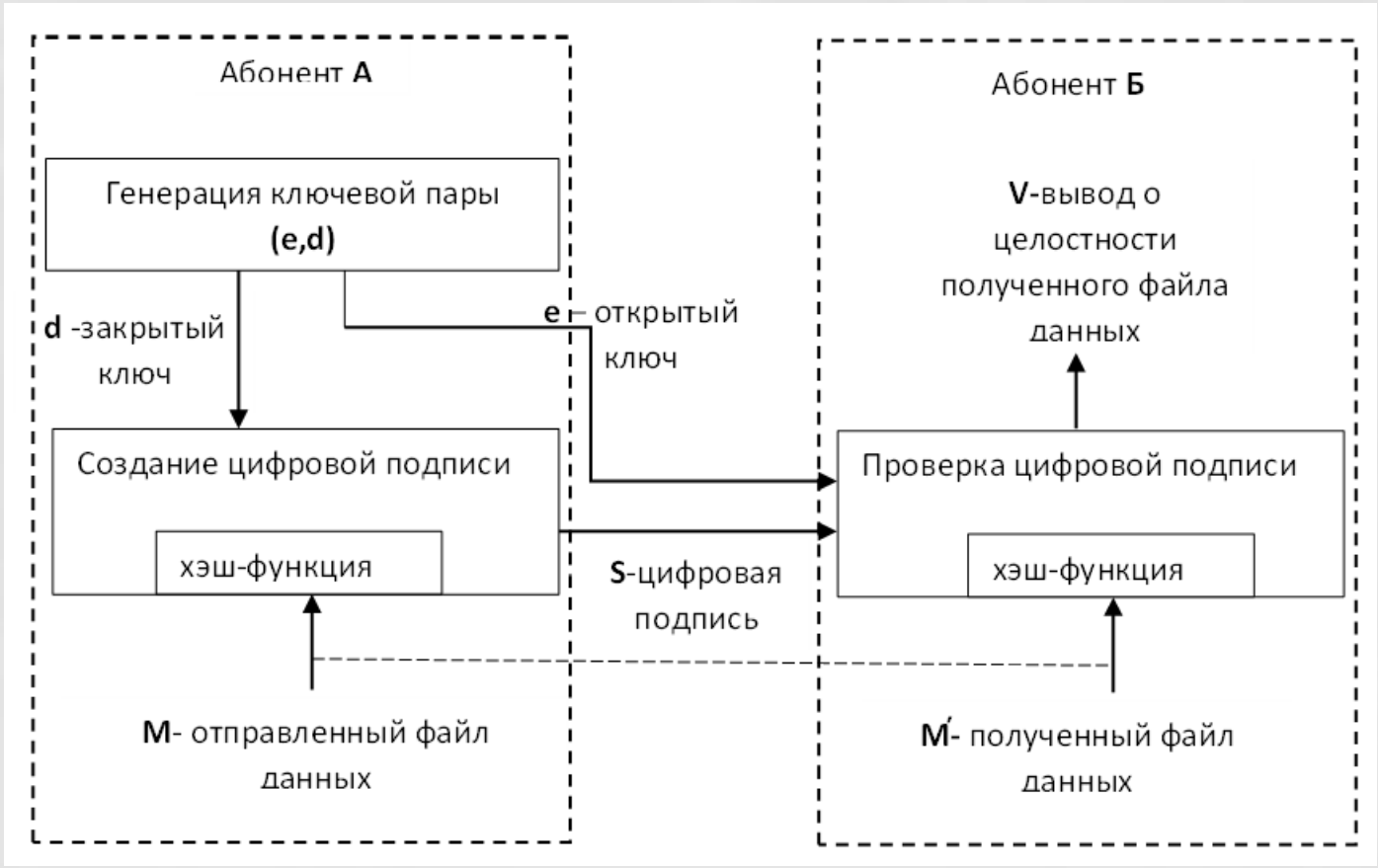
- Экзистенциальная подделка. Перехвачена цифровая подпись S_1 и S_2 и подбирается M', a, b удовлетворяющие сравнению

$$M' \equiv (\mathrm{a} \times S_1 + b \times S_2) mod\ (p-1)$$

- Селективная подделка. Имеется заданное сообщение M и требуется подобрать две части подписи S_1 и S_2 . Выбираем S_1 и пытаемся вычислить S_2 из ${e_2}^{S_1} \times {S_1}^{S_2} \equiv {e_1}^{M} mod\ p$. Это вычислительно трудная задача дискретного логарифмирования

$$S_2 \equiv log_{S_1}\ {e_2}^{-S_1} \times {e_1}^{M} mod\ p$$

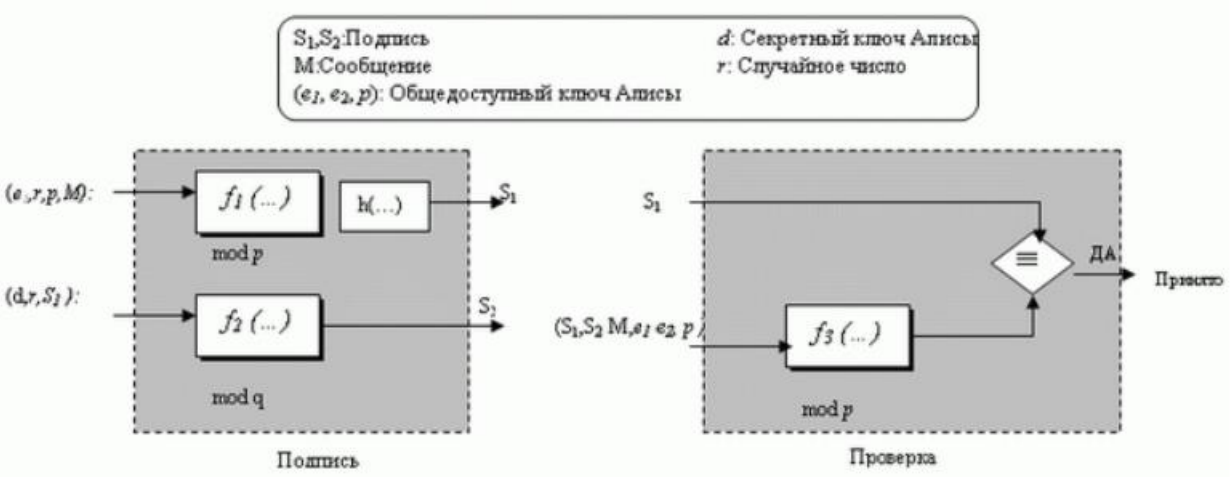
Детализация модели протокола ЭЦП



Цифровая подпись Шнорра

Claus-Peter Schnorr

Схема цифровой подписи Schnorr

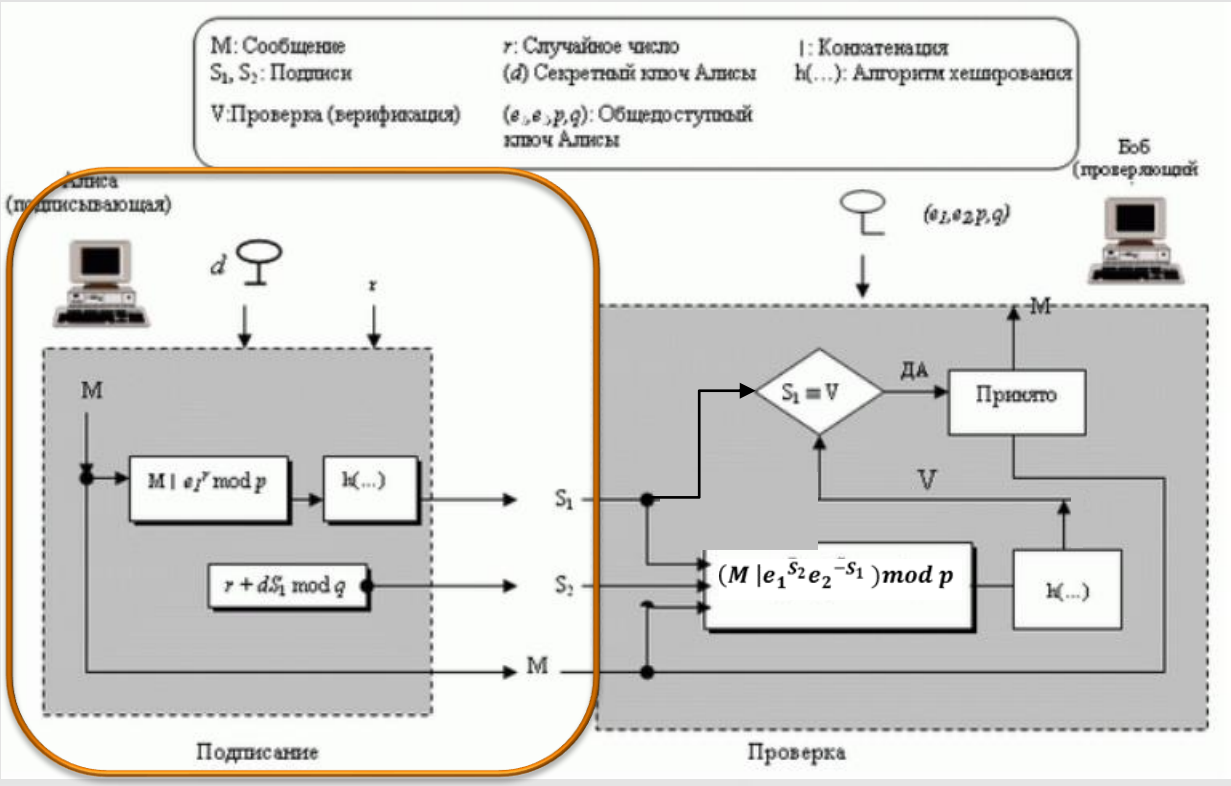


- Шнорр предложил новую схему, основанную на схеме Эль-Гамаля , но с уменьшенным размером подписи
- В процессе подписания две функции (f_1 и f_2) создают две части подписи. В процессе проверки выход функции f_3 сравнивается с первой частью подписи
- Схема использует два модуля: p и q. Функции f_1 и f_3 используют p, а функция f_2 использует q

Schnorr генерация ключей

- Выбирается простое число p, которое обычно равно по длине 1024 битам
- Выбирается другое простое число q, которое имеет тот же самый размер, что и дайджест (например, 160 битов),такое , что
$$(p-1) = 0 \, mod \, q$$
- Выбирается e_1, такое, что ${e_1}^q = 1 \, mod \, p$ путем вычисления $e_1 = {e_0}^{p-1/q} \, mod \, p,$ где e_0 первообразный корень p
- Выбирается целое $d < q$ и вычисляется $e_2 = {e_1}^d \, mod \, p$
- Объявляется открытый ключ (e_1, e_2, p, q)
- Назначается закрытый ключ d

Schnorr подписание



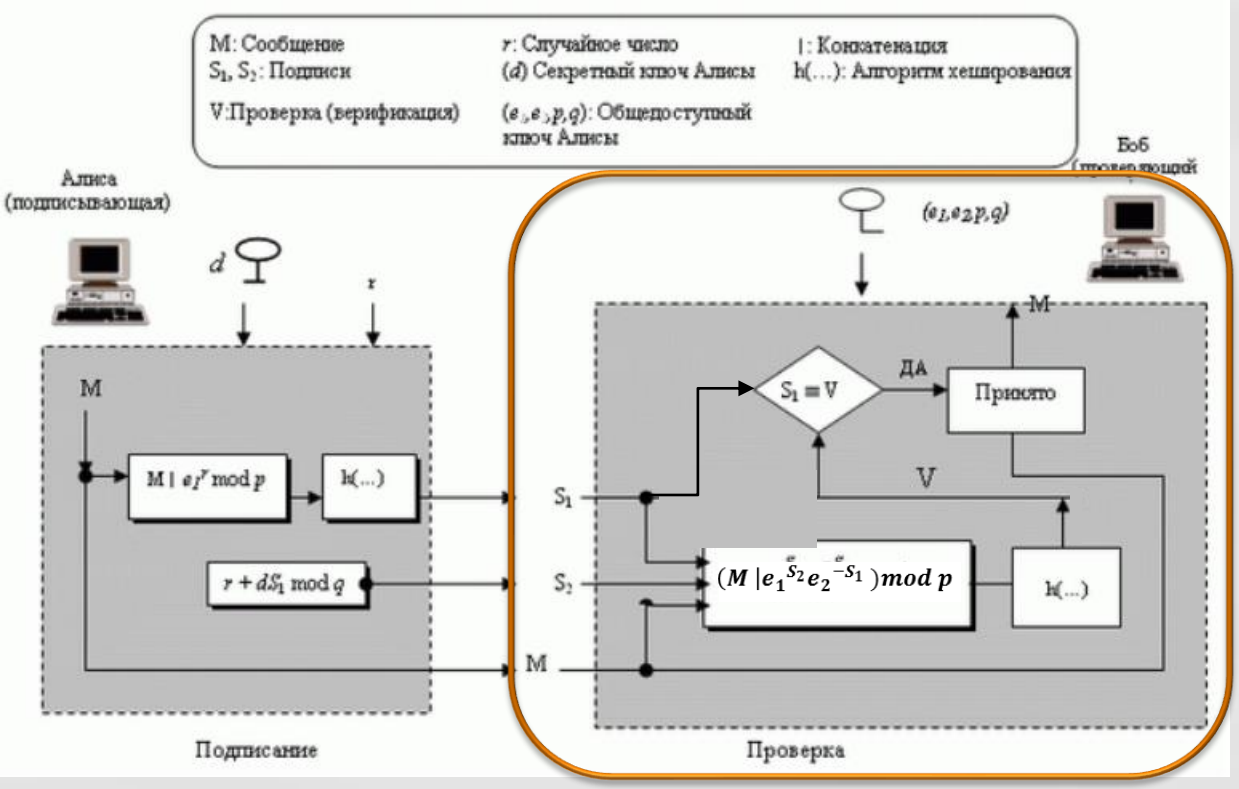
- Выбирается случайное число $r, 1 < r < q$ (r заново выбирается для каждого нового сообщения)
- Вычисляется (f_1) первая часть подписи

$$S_1 = h(M \mid e_1{}^r mod\ p)$$

- Вычисляется (f_2) вторая часть подписи

$$S_2 = (r + d \times S_1)\ mod\ \ q$$

Schnorr проверка



- Вычисляется (f_3):
  - $V = h(M \,|e_1{}^{S_2} \times e_2{}^{-S_1}\,) mod\ p$
- Если $V \equiv S_1\ mod\ p$, то подпись действительна

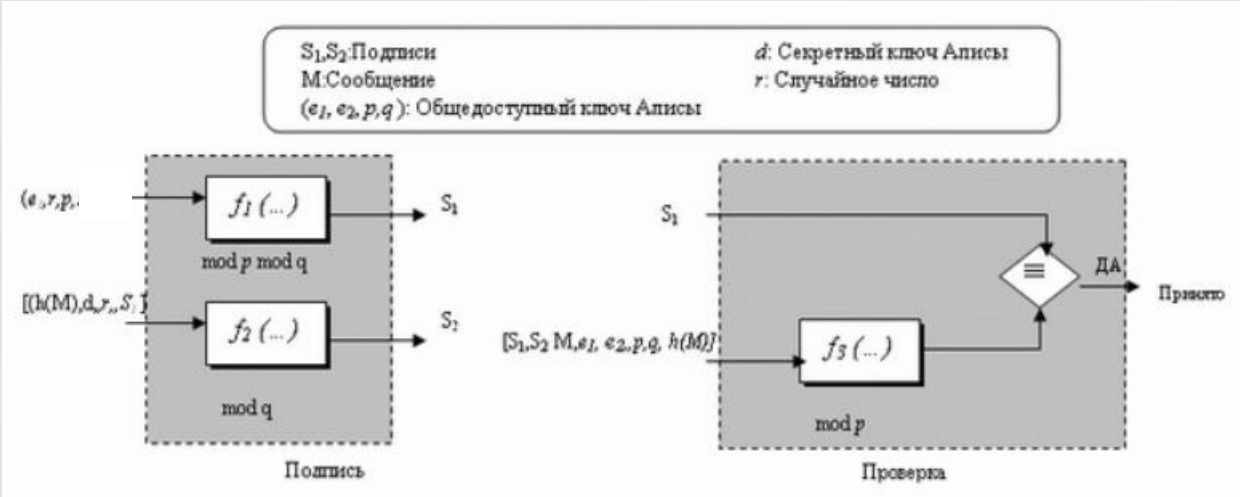
Подделка цифровой подписи Schnorr

- Все атаки на схему Эль-Гамаля могут быть применены к схеме Шнорра
- Однако схема Шнорра находится в лучшем положении, потому что $S_1 = h(M \mid {e_1}^r mod\ p)$ т.е. хэш-функция применяется к комбинации сообщение и ${e_1}^r$, в которой r является секретным.

Цифровая подпись DSA (Digital Signature Algorithm)

Стандарт цифровой подписи (DSS) принятый NIST в 1994 г.

Схема цифровой подписи DSA



- В процессе подписания две функции f_1 и f_2 создают две части подписи. В процессе проверки выход функции f_3 сравнивается с первой частью подписи.
- Эта схема использует дайджест сообщения (а не собственно сообщение), как часть входов к функциям f_1 и f_3
- Схема применяет два общедоступных модуля: p и q. Функции f_1 и f_3 используют оба модуля p и q, функция f_2 - только q

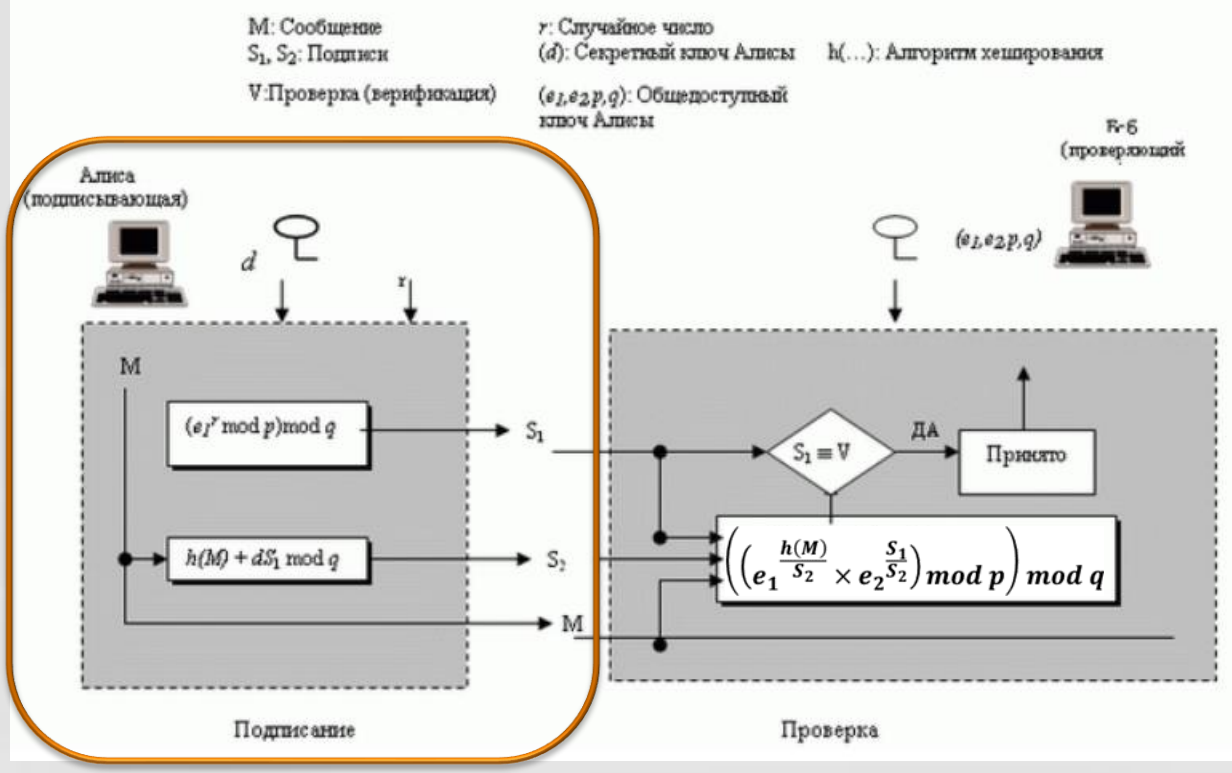
DSA генерация ключей

- Выбирается простое число p, длиной между 512 и 1024 битами. Число битов в p должно быть кратно 64
- Выбирается другое простое число q, которое имеет тот же самый размер, что и дайджест - 160 битов, такое , что

$$(p-1) = 0 \, mod \, q$$

- Выбирается e_1, такое, что ${e_1}^q = 1 \, mod \, p$ путем вычисления $e_1 = {e_0}^{p-1/q} \, mod \, p,$ где $e_0 \in Z_p$ (теорема Ферма)
- Выбирается целое $d < q$ и вычисляется $e_2 = {e_1}^d \, mod \, p$
- Объявляется открытый ключ (e_1, e_2, p, q)
- Назначается закрытый ключ d

DSA подписание



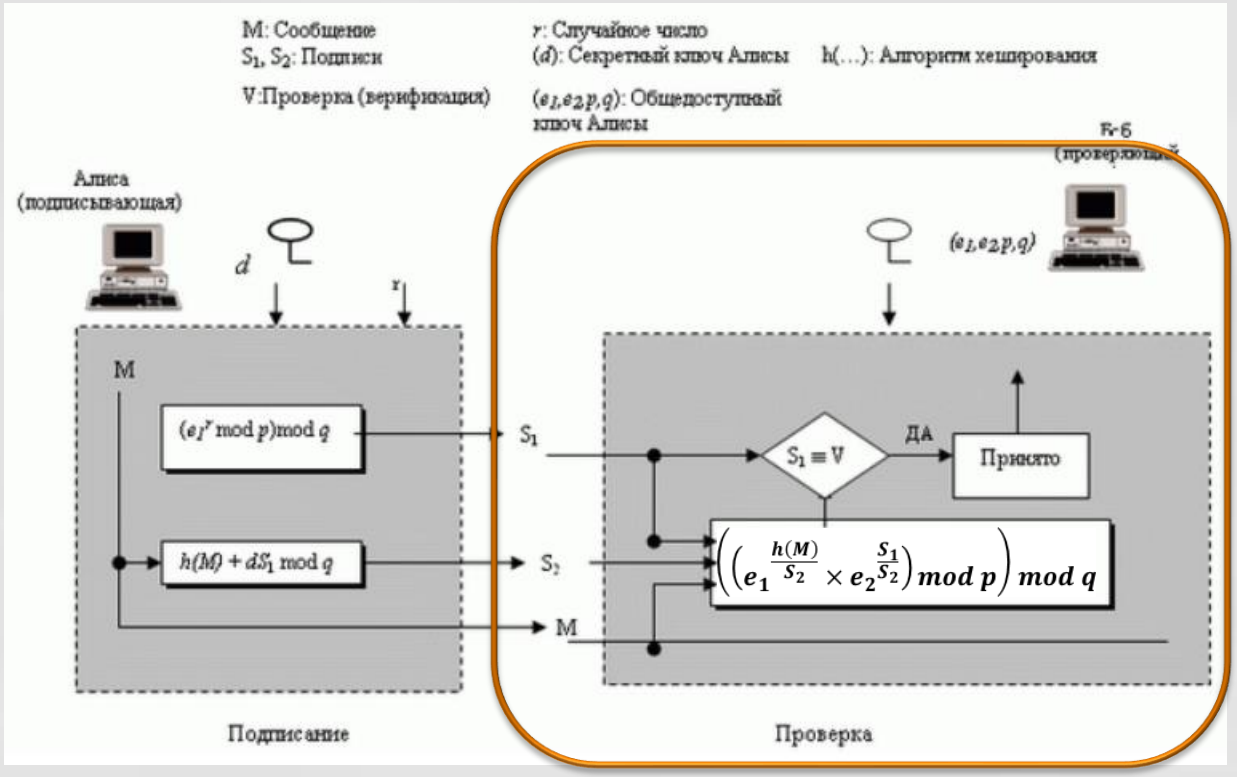
- Выбирается случайное число $r, 1 < r < q$ (r заново выбирается для каждого нового сообщения)
- Вычисляется первая часть подписи (f_1):

$$S_1 = (e_1^r mod\ p)\ mod\ q$$

- Вычисляется дайджест $h(M)$
- Вычисляется вторая часть подписи(f_2):

$$S_2 = r^{-1}(h(M)\ + d \times S_1\ )\ mod\ \ q$$

DSA проверка



- Проверяем :
  - $0 < S_2 < q$
  - $0 < S_1 < q$
- Вычисляем (f_3):

$$V = \left(\left(e_1^{\frac{h(M)}{S_2}} \times e_2^{\frac{S_1}{S_2}}\right) mod\ p\right) mod\ q$$

- Если $V \equiv S_1 mod\ q$ подпись действительна

Сравнительный анализ

- Вычисление DSA подписи быстрее, чем вычисление подписей RSA, при использовании того же самого p
- DSA подпись короче, чем подписи в схеме Эль-Гамаля и Шнорра, потому что q меньше, чем p
- Одним из главных аргументов против DSA является использование в данной схеме частного случая задачи вычисления дискретного логарифма. Этот вариант мало изучен и, возможно, имеет существенно меньшую сложность вскрытия

Цифровая подпись ГОСТ Р 34.10-94

Российский стандарт, введенный 01.01.1995 (утратил силу в 2001)

Общие сведения о стандарте

- Определяет алгоритм, который относится к семейству алгоритмов ElGamal и аналогичен алгоритму, реализованному в стандарте DSS
- Использует хэш-функцию стандарта ГОСТ Р 34.11-94, которая создает хэш-код длиной 256 бит. Это обуславливает требования к выбираемым параметрам (простым числам p и q)
- Определяет процедуру получения простых чисел p и q

ГОСТ генерация ключей

- Выбирается простое число p, $2^{509} < p < 2^{512}$ или $2^{1020} < p < 2^{1024}$ битами.
- Выбирается другое простое число q, $2^{254} < q < 2^{256}$, которое соответствует размеру дайджеста 256 битов, такое , что
$$(p-1) = 0 \, mod \, q$$
- Выбирается e_1, такое, что ${e_1}^q = 1 \, mod \, p$ путем вычисления $e_1 = {e_0}^{p-1/q} \, mod \, p,$ где $e_0 \in Z_p$ (теорема Ферма)
- Выбирается целое $d < q$ и вычисляется $e_2 = {e_1}^d \, mod \, p$
- Объявляется открытый ключ (e_1, e_2, p, q)
- Назначается закрытый ключ d

ГОСТ подписание

- Выбирается случайное число $r, 1 < r < q$ (r заново выбирается для каждого нового сообщения)
- Вычисляется первая часть подписи

$$S_1 = (e_1 mod\ p)\ mod\ q$$

- Вычисляется дайджест $h(M)$ по ГОСТ Р 34.11-94
- Вычисляется вторая часть подписи

$$S_2 = (r \times h(M)\ + d \times S_1\ )\ \ mod\ \ q$$

ГОСТ проверка

- Проверяем :
  - $0 < S_2 < q; 0 < S_1 < q$
- Вычисляем
  - $w = h(M)^{-1} mod\ q$
  - $u_1 = w \times S_2 mod\ q$
  - $u_2 = (q - S_1) \times w\ mod\ q$
  - $V = \left((e_1{}^{u_1} \times e_2{}^{u_2}) mod\ p\right) mod\ q$
- Если $V \equiv S_1 mod\ q$ подпись действительна

Примечание

- Алгоритм ГОСТ Р 34.10-94 имеет существенно больший запас стойкости по сравнению с DSA, поскольку параметр q имеет размерность 256 бит, а соответствующий параметр DSA ограничены длиной в 160 бит.
- Подписи, созданные с использованием алгоритмов ГОСТ Р 34.10-94 или DSS, называются *рандомизированными*, так как для одного и того же сообщения с использованием одного и того же закрытого ключа каждый раз будут создаваться разные подписи (S_1, S_2), поскольку каждый раз будет использоваться новое значение r

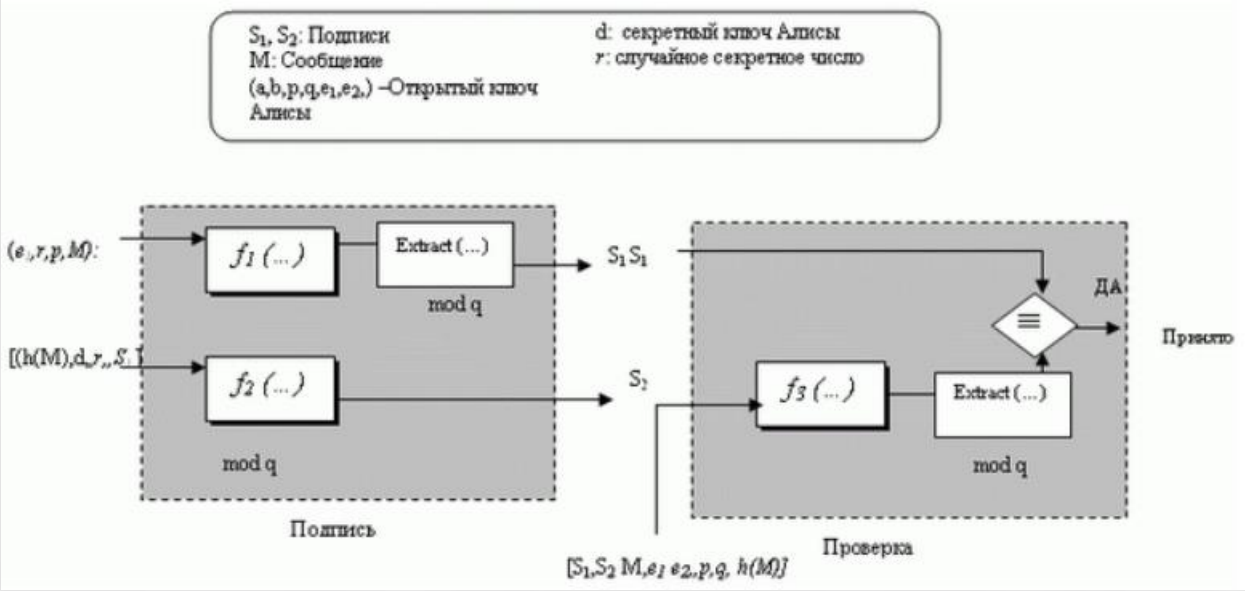
Цифровая подпись ECDSA
(Elliptic Curve Digital Signature Algorithm)

Стандарт цифровой подписи (ECDSS) принят ANSI в 1999 и NIST в 2000 г.

Эллиптическая криптография

- Безопасность RSA и Elgamal обеспечивается ценой использования больших ключей
- Требуется альтернативный метод, который дает тот же самый уровень безопасности, но с меньшими размерами ключей
- Одним из этих перспективных вариантов является криптосистема на основе метода эллиптических кривых (*Elliptic Curve Cryptosystem — ECC*)

Схема цифровой подписи ECDSA



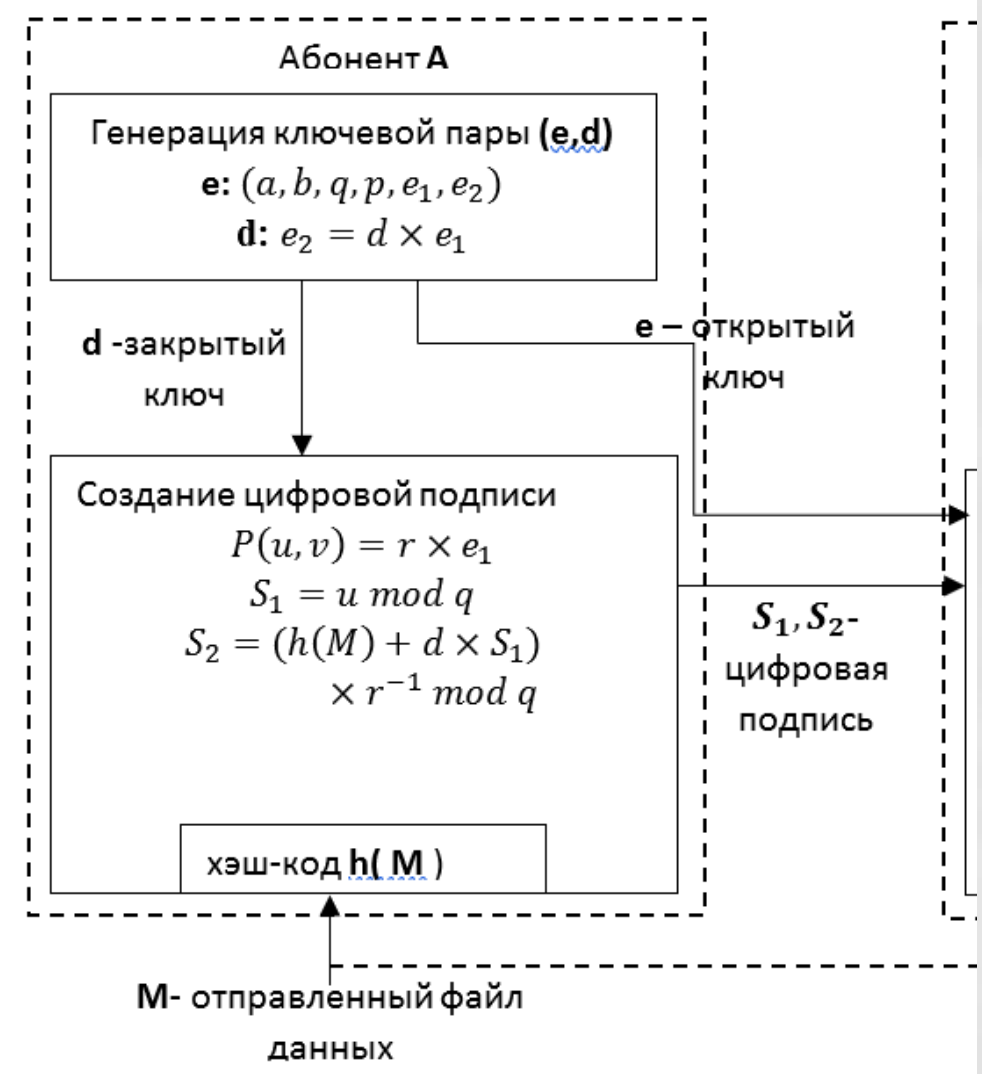
- Функция f_1 создает новую точку для секретного ключа подписывающего лица
- Функция f_2 создает новую точку из двух общедоступных ключей подписывающего лица
- Каждый экстрактор $Extract$ извлекает первые координаты соответствующей точки в модульной арифметике

- В процессе подписания две функции f_1 и f_2 и экстрактор (извлекающее устройство) создают две части подписи
- В процессе проверки (верификации) обрабатывают выход одной функции f_2 (после прохождения через экстрактор) и сравнивают ее с первой частью подписи

Генерация ключей ECDSA

- Выбирается эллиптическая кривая $E_p(a,b)$, p — простое
- Выбирается точка на кривой $\boldsymbol{e_1}$= (x_1,y_1)
- Для дальнейших вычислений выбирается другое простое число q - порядок циклической подгруппы группы точек эллиптической кривой : $q \times (x_1,y_1)$=O
- Выбирается целое число d , $1 < d < q-1$ и назначается закрытым ключом
- Вычисляется другая точку на кривой $\boldsymbol{e_2} = d \times \boldsymbol{e_1}$
- Объявляется открытый ключ $(a,b,p,q,\boldsymbol{e_1},\boldsymbol{e_2})$

ECDSA подписание



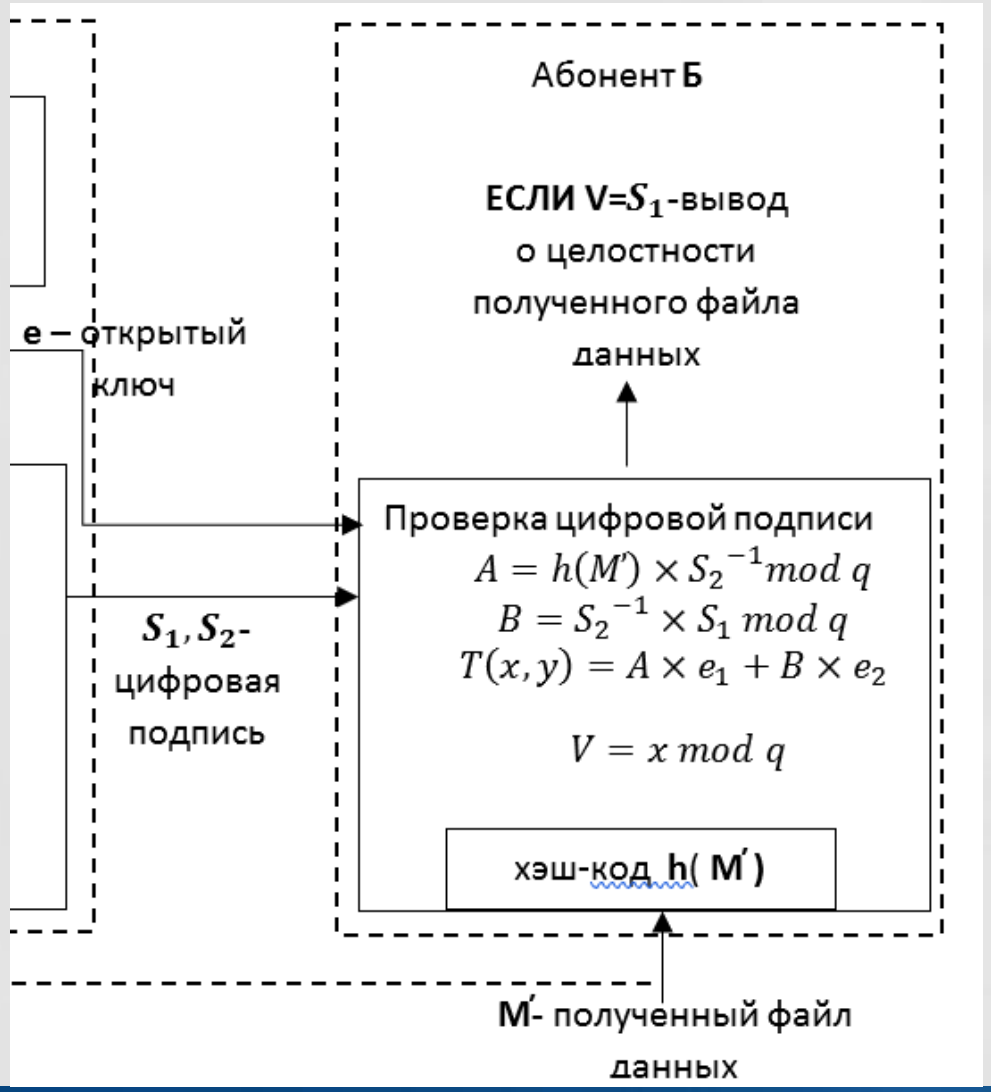
- Выбирается секретное случайное число, $r,\ 1 < r < q - 1$
- Выбирается третья точка на кривой, $\boldsymbol{P}(u, v) = r \times \boldsymbol{e_1}$
- Используем абсциссу u , чтобы вычислить первую часть подписи

$$S_1 = u \bmod q$$

- Используем дайджест сообщения $h(M)$, закрытый ключ d, секретное случайное число r и S_1, чтобы вычислить вторую часть подписи

$$S_2 = (h(M) + d \times S_1) \times r^{-1} \bmod q$$

ECDSA проверка



- Используем M, S_1, S_2 для получения промежуточных результатов A и B:

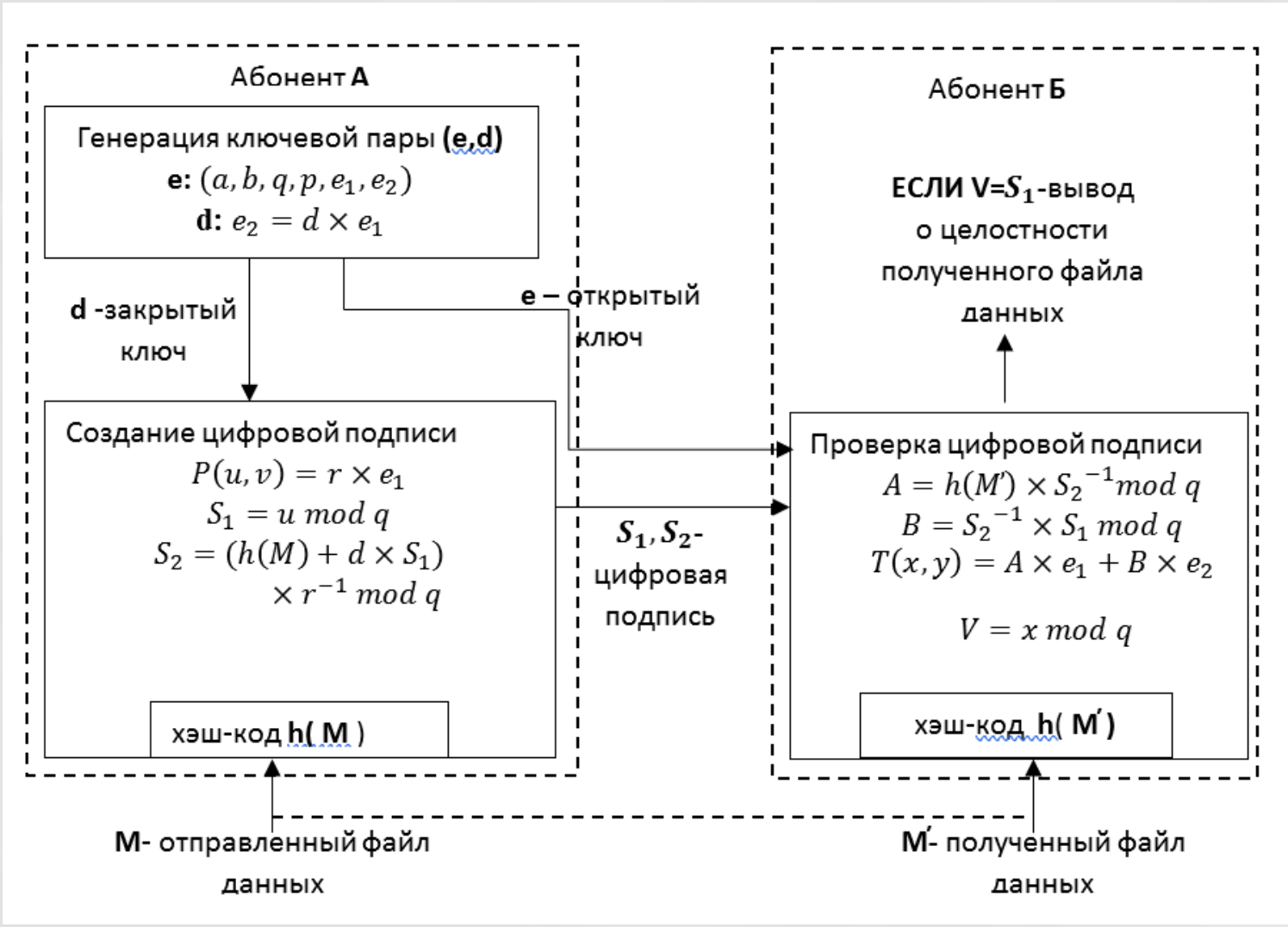
$$A = h(M) \times S_2{}^{-1} mod\ q$$
$$B = S_2{}^{-1} \times S_1 mod\ q$$

- Затем восстанавливаем третью точку

$$\boldsymbol{T(x,y) = A \times e_1 + B \times e_2}$$

- Верификатор $V = x\ mod\ q$ сравниваем с S_1

Схема протокола ECDSA



Цифровая подпись ГОСТ Р 34.10–2012

Российский стандарт, введен в действие 01.01.2013

Общие сведения о стандарте

- Определяет алгоритм, аналогичный алгоритму ECDSA
- Разработан Центром защиты информации и специальной связи ФСБ России с участием Открытого акционерного общества «Информационные технологии и коммуникационные системы» (ОАО «ИнфоТеКС»)
- Использует хэш-функцию стандарта ГОСТ Р 34.11–2012, которая создает хэш-код длиной 256 и 512 бит
- Процесс генерации ключей (для подписи и проверки подписи) не рассмотрен. Характеристики и способы реализации данного процесса определяются вовлеченными в него субъектами, которые устанавливают соответствующие параметры по взаимному согласованию
- Не определяет процесс генерации параметров схемы цифровой подписи. Конкретный алгоритм (способ) реализации данного процесса определяется субъектами схемы цифровой подписи исходя из требований к аппаратно-программным средствам, реализующим электронный документооборот

Генерация ключей ГОСТ

- Выбирается эллиптическая кривая $E_p(a,b)$: $y^2 \equiv x^3 + ax + b \; mod \; p, p > 3 -$ простое
- Выбирается простое число q (порядок циклической подгруппы точек):
  - $2^{254} < q < 2^{256}$, если длина хэш-кода 256
  - $2^{508} < q < 2^{512}$, если длина хэш-кода 512
- Выбирается базовая точка на кривой $\boldsymbol{e_1} = (x_1, y_1)$, q$\times \boldsymbol{e_1}$=0
- Выбирается целое число $d, (0 < d < q)$ и назначается закрытым ключом
- Вычисляется другая точку на кривой $\boldsymbol{e_2} = d \times \boldsymbol{e_1}$
- Объявляется открытый ключ $(a, b, p, q, \boldsymbol{e_1}, \boldsymbol{e_2})$

ГОСТ подписание

- Выбирается секретное случайное число, $r,\ 1 < r < q$
- Выбирается третья точка на кривой, $\boldsymbol{P}(u, v) = r \times \boldsymbol{e_1}$
- Используем абсциссу u , чтобы вычислить первую часть подписи

  $$S_1 = u \ mod \ q$$

- Используем дайджест сообщения $h(M),$ закрытый ключ $d,$ секретное случайное число r и S_1, чтобы вычислить вторую часть подписи

  $$S_2 = (r \times h(M) + d \times S_1) \ mod \ q$$

- S_1 и S_2 - две составляющие цифровой подписи

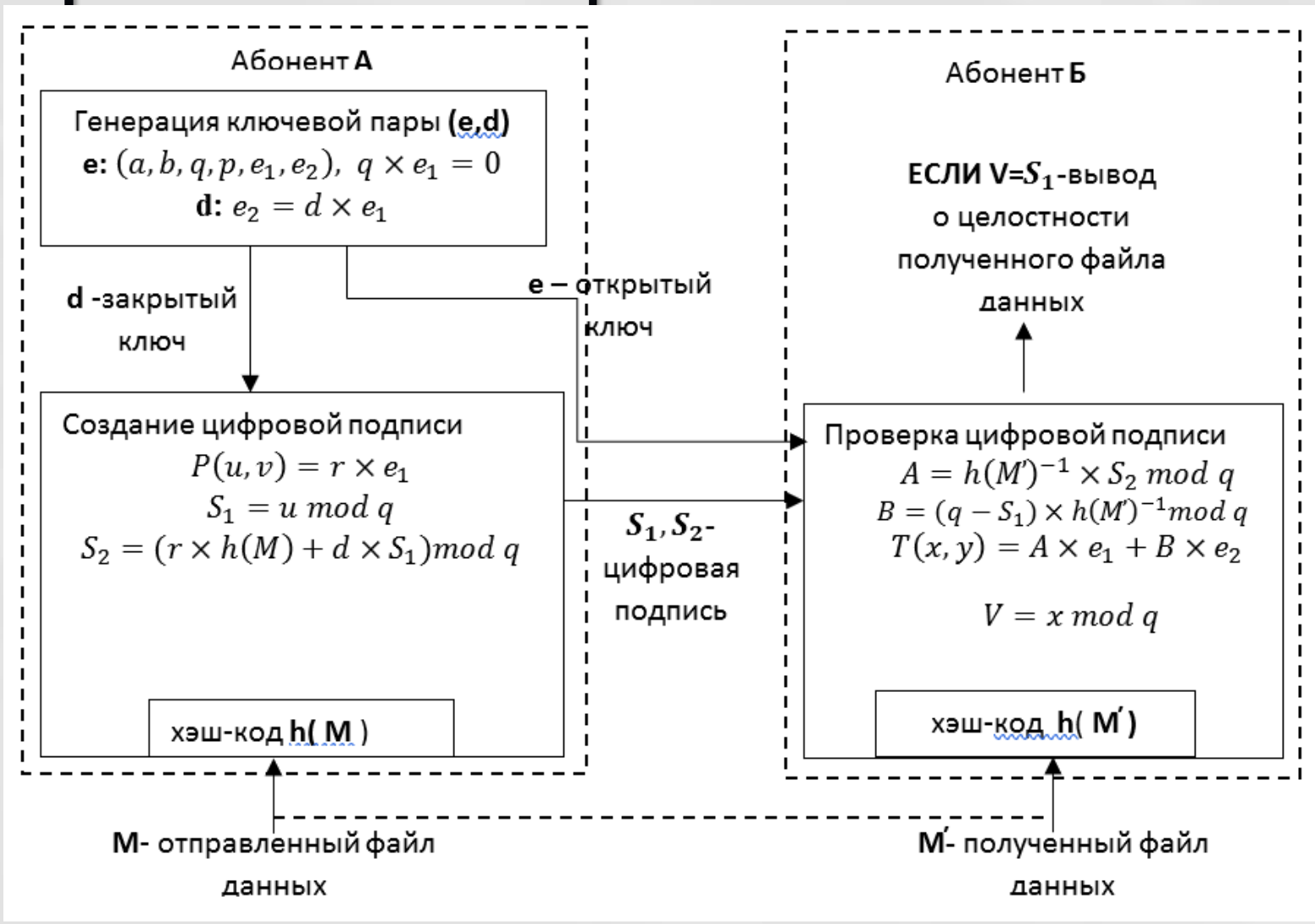
ГОСТ проверка

- Используем M, S_1, S_2 для получения промежуточных результатов A и B:
  - $A = h(M)^{-1} \times S_2 \, mod \, q$
  - $B = (q - S_1) \times h(M)^{-1} mod \, q$
- Затем восстанавливаем третью точку

$$\boldsymbol{T}(x, y) = A \times \boldsymbol{e_1} + B \times \boldsymbol{e_2}$$

- Верификатор $V = x \, mod \, q$ сравниваем с S_1

Схема протокола ЭЦП ГОСТ



ГОСТ примечание

- Криптостойкость цифровой подписи опирается на две компоненты — на стойкость хэш-функции и на стойкость самого алгоритма шифрования
- Вероятность взлома хэш-функции составляет 1.73×10^{-77} при подборе коллизии на фиксированное сообщение и 2.94×10^{-39} при подборе любой коллизии.
- Стойкость алгоритма шифрования основывается на проблеме дискретного логарифмирования в группе точек эллиптической кривой. На данный момент нет метода решения данной проблемы лучше, чем $O\left(\sqrt{q}\right)$ битовых операций. Таким образом при использонии 256-разрядное q, обеспечивается криптостойкость 10^{38} операций

Защита гибридной криптосистемы от атаки на секретный ключ

- Создать ЭЦП клиента на шифровке ключа или цифровом конверте
- Передать открытый ключ клиента серверу
- Проверить ЭЦП клиента на стороне сервера

Защищенный гибридный (RSA) протокол шифрования