МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность Информационные системы и технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №10 НА ТЕМУ:

Исследование алгоритмов генерации и верификации электронной цифровой подписи

Выполнила студентка 3 курса 1 группы

Пригодич Вера Валерьевна

Минск 2023

**Цель:** изучение алгоритмов генерации и верификации электронной цифровой подписи и приобретение практических навыков их реализации.

**Теоретические сведения**

Электронная цифровая подпись (ЭЦП) является важным элементом современных информационных систем, использующих методы и технологии криптографического преобразования информации.

Электронная цифровая подпись – контрольная характеристика сообщения, которая вырабатывается с использованием личного ключа, проверяется с использованием открытого ключа, служит для контроля целостности и подлинности сообщения и обеспечивает невозможность отказа от авторства.

Таким образом, ЭЦП выполняет те же функции, что и собственноручная (поставленная «от руки») подпись:

• аутентифицирование лица, подписавшего сообщение;

• контроль целостности подписанного сообщения;

• защита сообщения от подделок; • доказательство авторства лица, подписавшего сообщение, если это лицо отрицает свое авторство.

Важнейшие отличительные особенности ЭЦП:

• ЭЦП представляет собой бинарную последовательность (в отличие от графического образа, каковым является подпись от руки);

• указанная бинарная последовательность зависит от содержания подписываемого сообщения.

ЭЦП могут быть реализованы на основе классических методов зашифрования/расшифрования:

• на основе симметричных систем (с тайным ключом);

• на основе симметричных систем и посредника;

• на основе асимметричных систем (с открытым ключом).

Классическая технология использования ЭЦП предусматривает подписание не самого сообщения (обозначим его здесь *М*о), а его хеша, *H*(*Mо*). Это сокращает время генерации/верификации подписи и снижает вероятность появления случайных ошибок в итоговом документе.

Основу рассматриваемых протоколов составляют методы асимметричной криптографии и эллиптических кривых.

Общая структура подписанного электронного документа – *Мо* – *М*' – представляет собой, как правило, конкатенацию этого документа и ЭЦП *S*.

Важное свойство цифровой подписи заключается в том, что ее может проверить (верифицировать) каждый, кто имеет доступ к открытому ключу ее автора.

При генерации ЭЦП (по классической схеме) для сообщения М отправитель последовательно выполняет следующие действия:

• вычисляет хеш (хеш-образ) сообщения *М*: *Н*(*М*);

• вычисляет содержание ЭЦП (собственно ЭЦП *S*) по хешу *Н*(*М*) с использованием своего закрытого ключа *d*: *S* = *Сd*(*Н*(*М*));

• присоединяет (конкатенирует) ЭЦП к сообщению *М* и некоторой служебной информации, создавая таким образом итоговое сообщение *М*';

• посылает сообщение *М*' получателю; – получив сообщение *М*', другая сторона последовательно выполняет следующие действия:

• отделяет цифровую подпись *S* от сообщения *М* (для общего случая применим одинаковые символьные обозначения);

• применяет к сообщению *М* операцию хеширования, используя ту же функцию, что и отправитель, и получает хеш-образ полученного сообщения; • используя открытый ключ отправителя, расшифровывает *S*, т. е. извлекает из ЭЦП хеш-образ отправленного сообщения;

• проверяет соответствие (равенство) обоих хеш-образов, и если они совпадают, то отправитель действительно является тем, за кого себя выдает, а сообщение при передаче не подверглось искажению.

При этом стойкость ЭЦП к подделыванию (криптостойкость) определяется теми же факторами, что и криптостойкость алгоритмов зашифрования/расшифрования сообщений: чтобы применение ЭЦП имело смысл, необходимо, чтобы вычисление легитимной подписи без знания закрытого ключа было вычислительно сложным процессом. Решение такой задачи в асимметричных алгоритмах реализации ЭЦП опирается на известные нам вычислительные задачи:

• факторизации, т. е. разложения числа на простые множители;

• дискретного логарифмирования.

**ЭЦП на основе RSA**

Здесь можно рассматривать две ситуации:

• сообщение *Мo* подписывается и передается в открытом (незашифрованном) виде;

• сообщение *Мo*подписывается и передается в зашифрованном виде.

При этом подпись *S* вычисляется на основе соотношения

*S* ≡ (*H*(*Mo*))*dо* mod *no*,

при указанном выше реверсе в отношении ключевой информации; в *dо* и *no* – элементы тайного ключа отправителя. Передаваемое сообщение *М'* = *Мo*||*S*.

Соответственно, операция расшифрования на приемной стороне (получатель анализирует *Мп*||*S*) будет производиться в соответствии с формулой с известной модификацией ключей:

*H*(*Mo*) ≡ (*S*) *ео* mod *no*.

Далее вычисляется *Н*(*Mп*). Если *H*(*Mo*) = *H*(*Mп*), подпись верифицирована.

Если подписываемое сообщение *М*(*М'*) также должно передаваться в зашифрованном виде, то обычно *М'* шифруется на стороне отправителя стандартным образом: с помощью открытого ключа получателя (*е*п и *n*п), который перед основным процессом верификации подписи расшифровывает послание своим тайным ключом: *d*п и *n*п. Далее осуществляются вычисления и анализ, как и в первом случае.

**ЭЦП Эль-Гамаля**

Основное отличие в применении расчетов состоит в том, что результатом зашифрования является только одна пара чисел, а не пара для каждого блока исходного сообщения. Причем в рассматриваемом случае таким сообщением является хеш подписываемого документа: *Н*(*M*о).

Ключевая информация отправителя: открытый ключ: *y*, *g* и *р*; тайный ключ: *х*. Чтобы подписать сообщение *Мо*, обладатель используемых для ЭЦП ключей должен выбрать, как и в предыдущей схеме, случайное число *k*, взаимно простое с (*р* – 1). Затем вычисляется числа *а* и *b*, являющиеся цифровой подписью (*S* = {*a*, *b*}):

*a* ≡ *gk* mod *p*;

для вычисления *b* с помощью расширенного алгоритма Евклида решается уравнение

*Н*(*Mо*) ≡ (*xa* + *kb*) mod (*p* – 1).

Получателю отправляется сообщение *М'* = *Мо*||*S*.

Для верификации подписи вычисляется хеш полученного сообщения *Н*(*М*п) = *h*. Далее нужно убедиться, что выполняется равенство

*ya ab* ≡ *gh* mod *p*.

Если равенство выполняется, подпись верифицируется.

**ЭЦП Шнорра**

Рассматриваемая схема является основой стандарта ЭЦП в Беларуси. Алгоритм ЭЦП К. Шнорра (К. Schnorr) является вариантом алгоритма ЭЦП Эль-Гамаля.

Одной из особенностей ЭЦП Эль-Гамаля является то, что число p должно быть очень большим, чтобы сделать действительно трудной проблему дискретного логарифма. Рекомендуемая длина p должна составлять по крайней мере 1024 бита. Чтобы уменьшить размер подписи, Шнорр предложил новую схему, но с уменьшенным размером подписи.

Ключевая информация: *p* – простое число в диапазоне от 512 до 1024 битов; *q* –160-битное простое число, делитель (*p* – 1); любое число *g* (*g* ≠ 1) такое, что

*gq* ≡ 1 mod *p*.

Числа *p*, *g*, *q* являются открытыми и могут применяться группой пользователей.

Выбирается число *х* < *q* (*х* является тайным ключом) и вычисляется последний элемент открытого ключа:

*y* ≡ *g*–*х* mod *p*.

Секретный ключ имеет длину не менее 160 битов.

Для подписи сообщения *М*о выбирается случайное число *k* (1 < *k* < *q*) и вычисляет параметр *а*:

*а* ≡ *gk* mod *p*.

Далее вычисляется хеш от канкатенации сообщения *Мо* и числа *а*: *h* = *H*(*Mo*||*a*). Обратим внимание, что хэш-функция непосредственно не применяется к сообщению. Создается хеш-образ подписываемого сообщения, спереди присоединенного к числу *а*. Далее вычисляется значение *b*:

*b* ≡ (*k* + *xh*) mod *q*.

Получателю отправляются *М'* = *Мо*||*S*; *S* = {*h*, *b*}.

Для проверки подписи получатель вычисляет

*Х* ≡ *gb yh* (mod *p*).

Затем он проверяет выполнение равенства: *h* = *Н*(*M*п||*Х*). Подпись достоверна, если равенство выполняется.

Основные вычисления для генерации подписи могут производиться предварительно. Порядок величин *х* и *h* – около 140 двоичных разрядов, порядок числа *k* – около 70–72 разрядов. С учетом этого сложность операций умножения можно считать ничтожно малой по сравнению с модульным умножением в схеме RSA.

**Практическое задание**

Разработать авторское приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться результатами выполнения предыдущих лабораторных работ, а также доступными библиотеками либо программными кодами.

Приложение должно реализовывать следующие операции:

• генерацию и верификацию ЭЦП на основе алгоритмов RSA, Эль-Гамаля и Шнорра;

Допустим дано сообщение «v».

Подпишем данное сообщение по алгоритму RSA.

Для генерации ЭЦП этого сообщения необходимо найти его хеш. Воспользуемся алгоритмом SHA-256. Хеш сообщения будет иметь вид «4c94485e0c21ae6c41ce1dfe7b6bfaceea5ab68e40a2476f50208e526f506080».

Для осуществления этого действия в программе воспользуемся библиотекой *hashlib*.

def get\_hash(self, m: str):

        return int.from\_bytes(sha256(m.encode('utf-8')).digest(), 'big') % self.n

Листинг 1 – Генерация хеша сообщения

Сгенерируем ключевую информацию в соответствии с алгоритмом. Пусть *p* = 3, *q* = 3. Тогда, *n* = *p* \* *q* = 9, *e* = 3, *d* = 3.

Подпись *S* вычисляется на основе соотношения

*S* ≡ (*H*(*Mo*))*dо* mod *no*

Для возведения хеша в степень приведем его к числовому виду используя функцию *int.from\_bytes*. Получим следующую числовую последовательность:

*H*(*Mo*) = 34637769319165005229284612346681296385034709032735588699641438084944242958464

Используя сгенерированные выше параметры, по формуле вычислим подпись сообщения. Получим значение *S* = 8.

Передаваемое сообщение *М'* = *Мo*||*S*. В нашем случае *М'* = «v8».

Верифицируем данное сообщение по алгоритму RSA.

Соответственно, операция расшифрования на приемной стороне (получатель анализирует *Мп*||*S*) будет производиться в соответствии с формулой с известной модификацией ключей:

*H*(*Mo*) ≡ (*S*)*е* mod *no*.

*H*(*Mo*) ≡ (*8*)3 mod 9 = 8

Далее вычисляется *Н*(*Mп*).

*Н*(*Mп*) = 34637769319165005229284612346681296385034709032735588699641438084944242958464.

Найдем *Н*(*Mп*) mod *n* = 8.

По условию, если *H*(*Mo*) = *H*(*Mп*), подпись верифицирована.

8 = 8 => подпись верифицирована.

Программно данные действия реализованы следующим образом:

class RSA:

    def \_\_init\_\_(self, p, q, e):

        self.p = p

        self.q = q

        self.n = p \* q

        self.e = e

        phi = (p - 1) \* (q - 1)

        self.d = gcd\_extended(e, phi)[1]

        if self.d < 0:

            self.d = phi + self.d

        self.public\_key = (self.n, self.e)

        self.private\_key = (self.n, self.d)

    def get\_hash(self, m: str):

        return int.from\_bytes(sha256(m.encode('utf-8')).digest(), 'big') % self.n

    def get\_signature(self, m):

        return (self.get\_hash(m) \*\* self.d) % self.n

    def check\_signature(self, m, s):

        return self.get\_hash(m) == ((s \*\* self.e) % self.n)

Листинг 2 – ЭЦП на основе алгоритма RSA

Можно проверить приведенные вычисления в приложении.

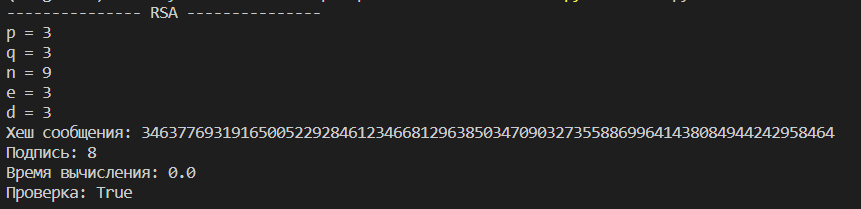


Рисунок 1 – Результат генерации и верификации ЭЦП на основе RSA

Подпишем то же сообщение по алгоритму Эль-Гамаля.

Сгенерируем ключевую информацию по алгоритму.

Пусть *p* = 3, *g = 2, y* = 1, *x* = 2.

Чтобы подписать сообщение *Мо*, сгенерируем случайное число *k*, взаимно простое с (*р* – 1). Пусть *k* = 2.

Затем вычислим числа *а* и *b*, являющиеся цифровой подписью (*S* = {*a*, *b*}):

*a* ≡ *gk* mod *p*;

*а* = 22 mod 3 = 2.

для вычисления *b* с помощью расширенного алгоритма Евклида решается уравнение

*Н*(*Mо*) ≡ (*xa* + *kb*) mod (*p* – 1).

*b* = *k*–1(*Н*(*Mо*) – *xa*) mod (*p* – 1) = 0.

Подпись будет иметь вид *S* = {2, 0}.

Получателю отправляется сообщение *М'* = *Мо*||*S*. *М'* = «v(2, 0)».

Верифицируем подпись по алгоритму Эль-Гамаля.

Для верификации подписи вычисляется хеш полученного сообщения *Н*(*М*п) = *h*. Далее нужно убедиться, что выполняется равенство

*ya ab* ≡ *gh* mod *p*.

Если равенство выполняется, подпись верифицируется.

12 · 20= 2*h* mod 3

1 = 1

Подпись верифицирована.

Программно данный алгоритм реализован следующим образом:

class Elgamal:

    def \_\_init\_\_(self, p, g, x):

        self.p = p

        self.g = g

        self.x = x

        self.y = (g \*\* x) % p

        self.public\_key = (self.p, self.g, self.y)

        self.private\_key = (self.p, self.g, self.x)

    def get\_hash(self, m: str):

        return int.from\_bytes(sha256(m.encode('utf-8')).digest(), 'big') % self.p

    def get\_signature(self, m):

        h = self.get\_hash(m)

        k = 2

        while gcd\_extended(k, self.p - 1)[0] != 1:

            k += 1

        k\_r = gcd\_extended(k, self.p - 1)[1]

        a = (self.g \*\* k) % self.p

        b = ((h - self.x \* a) \* k\_r) % (self.p - 1)

        return (a, b)

    def check\_signature(self, m, s):

        h = self.get\_hash(m)

        a = s[0]

        b = s[1]

        return ((self.g \*\* h) % self.p) == ((self.y \*\* a) \* (a \*\* b)) % self.p

Листинг 3 – ЭЦП на основе алгоритма Эль-Гамаля

Можно проверить приведенные вычисления в приложении.

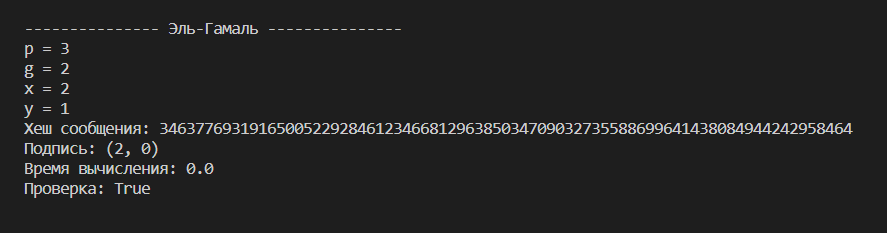


Рисунок 2 – Результат генерации и верификации ЭЦП на основе Эль-Гамаля

Подпишем то же сообщение по алгоритму Шнорра.

Сгенерируем ключевую информацию:

*p* – простое число. Пусть *p* = 48731.

*q* – простое число, делитель (*p* – 1). Пусть *q* = 443.

*g* (*g* ≠ 1) такое, что

*gq* ≡ 1 mod *p*.

Пусть *g* = 11444.

Выбирем число *х* < *q*. Пусть *х*  *=* 357*.* И вычислим последний элемент открытого ключа:

*y* ≡ *g*–*х* mod *p*.

*y* = 11444–357 mod 48731 = 7355.

Для подписи сообщения *М*о выберем случайное число *k* (1 < *k* < *q*). Пусть *k* = 2.

Вычислим параметр *а*:

*а* ≡ *gk* mod *p*.

*а* = 114442 mod 48371 = 33180.

Далее вычисляется хеш от канкатенации сообщения *Мо* и числа *а*: *h* = *H*(*Mo*||*a*).

Далее вычисляется значение *b*:

*b* ≡ (*k* + *xh*) mod *q*.

*b* = 408.

Подпись будет иметь вид *S* = {9561, 408}.

Получателю отправляются *М'* = *Мо*||*S*.

Верифицируем подпись по алгоритму Шнорра.

Для проверки подписи вычислим

Х ≡ *gb yh* (mod *p*).

X = 11444408 · 73559561 mod 443 = 33180.

Затем проверим выполнение равенства: *h* = *Н*(*M*п||*Х*). Подпись достоверна, если равенство выполняется.

33180 = 33180

Подпись верифицирована.

Программно данный алгоритм реализован следующим образом:

class Schnorr:

    def \_\_init\_\_(self, p, q, g, x):

        self.p = p

        self.q = q

        self.g = g

        self.x = x

        self.y = (g \*\* (q - x)) % p

        self.public\_key = (self.p, self.q, self.g, self.y)

        self.private\_key = (self.p, self.q, self.g, self.x)

    def get\_hash(self, m: str):

        return int.from\_bytes(sha256(m.encode('utf-8')).digest(), 'big') % self.p

    def get\_signature(self, m):

        k = 2

        while gcd\_extended(k, self.p - 1)[0] != 1:

            k += 1

        a = (self.g \*\* k) % self.p

        h = self.get\_hash(m + str(a))

        b = (k + self.x \* h) % self.q

        return (h, b)

    def check\_signature(self, m, s):

        h = s[0]

        b = s[1]

        X = ((self.g \*\* b) \* (self.y \*\* h)) % self.p

        return h == self.get\_hash(m + str(X))

Листинг 4 – ЭЦП на основе алгоритма Шнорра

Можно проверить приведенные вычисления в приложении.

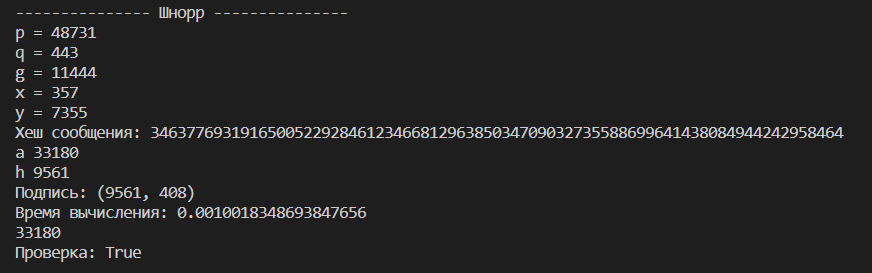


Рисунок 3 – Результат генерации и верификации ЭЦП на основе Шнорра

Вывод: были изучены алгоритмы генерации и верификации электронной цифровой подписи и приобретение практических навыков их реализации.

**Контрольные вопросы**

1. Дать определение ЭЦП.

Электронная цифровая подпись – контрольная характеристика сообщения, которая вырабатывается с использованием личного ключа, проверяется с использованием открытого ключа, служит для контроля целостности и подлинности сообщения и обеспечивает невозможность отказа от авторства.

1. Охарактеризовать основные функции ЭЦП

ЭЦП выполняет те же функции, что и собственноручная (поставленная «от руки») подпись:

• аутентифицирование лица, подписавшего сообщение;

• контроль целостности подписанного сообщения;

• защита сообщения от подделок; • доказательство авторства лица, подписавшего сообщение, если это лицо отрицает свое авторство.

1. В чем заключаются сходства и различия между собственноручной и электронной подписью?

Основное сходство между собственноручной и электронной подписью – это функции, выполняемые подписью.

Важнейшие отличительные особенности ЭЦП:

• ЭЦП представляет собой бинарную последовательность, в отличие от графического образа, каковым является подпись от руки;

• указанная бинарная последовательность зависит от содержания подписываемого сообщения.

1. Охарактеризовать основные способы реализации ЭЦП.

ЭЦП могут быть реализованы на основе классических методов зашифрования/расшифрования:

• на основе симметричных систем (с тайным ключом);

• на основе симметричных систем и посредника;

• на основе асимметричных систем (с открытым ключом).

1. Имеется ли различие в использовании ключевой информации при передаче зашифрованных сообщений и при передаче подписанных (ЭЦП) сообщений?

Да, есть различие в использовании ключевой информации при передаче зашифрованных сообщений и при передаче подписанных сообщений с использованием ЭЦП.

При передаче зашифрованных сообщений используется пара ключей: открытый ключ для шифрования сообщения и соответствующий ему закрытый ключ для расшифровки. Отправитель использует открытый ключ получателя для зашифрования сообщения, и только получатель, обладающий соответствующим закрытым ключом, может расшифровать и прочитать сообщение. Здесь цель состоит в обеспечении конфиденциальности сообщения.

При передаче подписанных сообщений с использованием ЭЦП также используется пара ключей: закрытый ключ для создания цифровой подписи и соответствующий ему открытый ключ для проверки подписи. Отправитель использует свой закрытый ключ для создания цифровой подписи, которая является уникальной для данного сообщения. Получатель может использовать открытый ключ отправителя для проверки подписи и убедиться в том, что сообщение не было изменено после создания подписи и что оно действительно было отправлено отправителем. Здесь цель состоит в обеспечении целостности сообщения и подтверждении его авторства.

6. Охарактеризовать криптостойкость ЭЦП на основе RSA, схемы Эль-Гамаля, схемы Шнорра, а также на основе DSA.

Криптостойкость ЭЦП на основе RSA зависит от сложности факторизации больших чисел на простые множители. Для достижения достаточной криптостойкости, длина ключа RSA должна быть не менее 2048 бит.

Схема Эль-Гамаля основана на сложности дискретного логарифмирования. Криптостойкость ЭЦП на основе этой схемы также зависит от длины ключа. Рекомендуется использовать ключи длиной не менее 2048 бит.

Схема Шнорра также основана на сложности дискретного логарифмирования. Рекомендуется использовать ключи длиной не менее 256 бит.

DSA основана на сложности дискретного логарифмирования в конечных полях. Рекомендуется использовать ключи длиной не менее 2048 бит.

7. Какие элементы составляют ключевую информацию алгоритмов реализации ЭЦП, перечисленных в вопросе 6?

RSA: открытый ключ получателя – *е*п и *n*п, тайный ключ – *d*п и *n*п.

DSA: открытый ключ – *p*, *q*, *v*, *y*, закрытый – *х.*

Эль-Гамаля: открытый ключ – *y*, *g* и *р*; тайный ключ – *х.*

Шнорра: откртый ключ – *p*, *q*, *g*, *y,* закрытый – *х.*

8. Дать сравнительные характеристики схемам ЭЦП, перечисленным в вопросе 6.

RSA:

* Прост в реализации и широко используется;
* Быстрый алгоритм шифрования и расшифрования;
* Длина ключа может быть очень большой, что обеспечивает высокий уровень безопасности;
* Использование RSA для ЭЦП требует большой вычислительной мощности.

Схема Эль-Гамаля:

* Меньше вычислительной сложности, чем RSA;
* Имеет более высокую стойкость к атакам на основе квантовых вычислений, чем RSA;
* Длина ключа должна быть больше, чем в RSA, для обеспечения такого же уровня безопасности.

Схема Шнорра:

* Характеризуется высокой эффективностью и высоким уровнем безопасности;
* Обладает сравнительно небольшой длиной подписи и ключа, что делает его полезным для систем с ограниченными ресурсами.

DSA:

* Быстрый алгоритм подписи и проверки подписи;
* Меньшая длина ключа, чем RSA и Эль-Гамаля, при сохранении высокого уровня безопасности;
* Не поддерживает шифрование, только подпись.

Выбор конкретной схемы ЭЦП зависит от требований к безопасности и эффективности, а также от особенностей конкретного применения.

9. Охарактеризовать особенности государственного стандарта ЭЦП в Республике Беларусь.

Существующий стандарт ЭЦП в Республике Беларусь (СТБ 34.101.45-2013) основан на схеме Шнорра (а также на эллиптических кривых).

СТБ 34.101.45-2013 определяет конкретные алгоритмы и параметры, которые должны быть использованы при реализации системы ЭЦП в Беларуси. Это включает в себя спецификации алгоритмов Шнорра и/или эллиптической криптографии, требования к длине ключей, методы генерации ключей, процедуры подписи и проверки подписи, а также требования к безопасности и защите ключевой информации.