Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Отчет по лабораторной работе №10**

**«ИССЛЕДОВАНИЕ АЛГОРИТМОВ ГЕНЕРАЦИИ И ВЕРИФИКАЦИИ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ»**

Выполнил:

Cтудент 3 курса 1 группы

Парибок И. А.

Вариант 5

**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.

Остановимся на важнейших свойствах и иных информационных и фактологических характеристиках ЭЦП. Более подробные сведения из предметной области можно найти в [2, 4, 29, 50]. Понятие «электронная цифровая подпись» было введено в 1976 г. У. Диффи и М. Хеллманом.

После создания RSA разработаны алгоритмы цифровой подписи И. Рабина и Р. Меркле. В 1984 г. Ш. Гольдвассер, С. Микали и Р. Ривест сформулировали требования безопасности к алгоритмам ЭЦП, описали атаки на ЭЦП. Государственный стандарт Республики Беларусь [51] определяет понятие ЭЦП в следующем виде.

**Определение 1.** **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.

Как следует из определения 1, основным компонентом в технологии ЭЦП является ключ. Принадлежность ключа, в предположении, что он известен только законным пользователям, позволяет решать все «возложенные на ЭЦП», сформированную на основе этого ключа, задачи. В соответствии с этим обстоятельством перечисленные выше функции 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).

Первый из перечисленных методом ничем не отличается, например, от DES.

Во втором случае создаются две симметричные системы: между отправителем и посредником и между посредником и получателем. Причем посредник выдает двум сторонам различный тайный (для иных субъектов системы) ключ.

В последнем случае сообщение, отправляемое получателю, шифруется тайным ключом отправителя. Отправитель же верифицирует подпись (в данном случае – устанавливает авторство, используя для расшифрования публичный ключ отправителя, и получает гарантию в защищенности переданного сообщения от подделок, если после расшифрования формат и содержание документа имеют логическую стройность) с помощью открытого ключа отправителя.

Таким образом, в этом случае, как и в первых двух случаях, ЭЦП, как отдельный, самостоятельный, присоединенный к исходному документу элемент получаемого сообщения, отсутствует. Кроме того, в отличие от классической асимметричной криптографии, где используется ключевая информация получателя, в нашем случае используется ключевая информация отправителя: открытый ключ – для зашифрования, тайный – для расшифрования.

С учетом изложенного можем сформулировать определение ЭЦП в несколько ином виде.

**Определение 2. Электронная цифровая подпись** – бинарная (или в ином виде) последовательность символов, являющаяся реквизитом электронного документа, зависящая от содержания этого документа и предназначенная для подтверждения целостности и подлинности электронного документа.

Классическая технология использования ЭЦП предусматривает подписание не самого сообщения (обозначим его здесь Мо), 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 – *М*о – *М'* – представляет собой, как правило, конкатенацию этого документа и ЭЦП S. Кроме этих двух элементов, интегральный документ может содержать некоторую служебную информацию (дата, время отправки или различные данные об отправителе), как это схематично показано на рис. 1.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 1 – Пояснение к процедуре формирования ЭЦП и структуре подписанного документа

Важное свойство цифровой подписи заключается в том, что ее может проверить (верифицировать) каждый, кто имеет доступ к открытому ключу ее автора. На рис. 2 показан в общем виде порядок процесса верификации (без учета использования служебной информации). Заметим, что в общем случае версии исходного документа (*М*о) и полученного (*М*п) могут отличаться.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Рисунок 2 – Пояснение к процедуре верификации ЭЦП

Если в результате устанавливается равенство хешей: Н(*М*п) = Н(*М*о), то принимается решение о подлинности подписи и целостности документа *М*п, т. е. это также означает, что *М*п = *М*о. Из приведенных на рис. 1 и рис. 2 последовательных преобразований можно сделать следующие общие выводы:

– 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. На основе первой задачи строится алгоритм RSA, на основе второй – алгоритмы, например, Эль-Гамаля, DSA, Шнорра. Здесь остановимся на кратком описании математических основ алгоритмов.

 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 указанном выше реверсе в отношении ключевой информации; в (10.1) 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 (8.6) с известной модификацией ключей:

*H*(*M*o) ≡ (*S*)еоmod*n*o.

Далее вычисляется *Н*(*M*п).

Если *H*(*M*o) = *H*(*M*п), 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.

Практическое задание

1. Разработать авторское оконное 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;

• оценку времени выполнения указанных процедур при реальных (требуемых) ключевых параметрах.

2. Для выполнения необходимых операций передачи (по сети)/верификации информации обменяться открытой ключевой информацией с получателем подписанного сообщения для каждого исследуемого алгоритма (по согласованию с преподавателем).

**Генерация и верификация ЭЦП на основе алгоритма RSA**

RSA (Rivest-Shamir-Adleman) — это криптографический алгоритм, использующий асимметричное шифрование. Он основан на задаче факторизации больших целых чисел. Алгоритм RSA включает в себя генерацию ключей, шифрование и расшифровку сообщений, а также генерацию и верификацию цифровой подписи.

**Генерация ключей RSA:**

1. Генерация простых чисел p и q большой длины.

2. Вычисление произведения n = p \* q.

3. Вычисление функции Эйлера от n: φ(n) = (p-1) \* (q-1).

4. Выбор целого числа e, взаимно простого с φ(n), такого что 1 < e < φ(n).

5. Вычисление секретного числа d, обратного по модулю φ(n) к числу e: d \* e ≡ 1 (mod φ(n)).

6. Пара (n, e) является открытым ключом, а пара (n, d) - секретным ключом.

**Шифрование RSA:**

1. Представление сообщения в виде целого числа m, меньшего n.

2. Вычисление зашифрованного сообщения c = m^e (mod n), где e - открытый ключ.

**Расшифровка RSA:**

1. Получение зашифрованного сообщения c.

2. Вычисление расшифрованного сообщения m = c^d (mod n), где d - секретный ключ.

**Генерация и верификация ЭЦП RSA:**

1. Вычисление хеша сообщения.

2. Подпись сообщения: Вычисление ЭЦП подписи s = H(m)^d (mod n), где H(m) - хеш сообщения, d - секретный ключ.

3. Верификация подписи: Вычисление хеша сообщения и проверка условия s^e ≡ H(m) (mod n), где s - ЭЦП подписи, e - открытый ключ.

Реализация RSA включает использование библиотеки `cryptography`. В приведенном выше коде используются функции из этой библиотеки для генерации ключей, шифрования, расшифровки и генерации/верификации ЭЦП RSA.

- Функция `generate\_rsa\_keys` генерирует открытый и секретный ключи RSA.

- Функция `sign\_rsa` вычисляет ЭЦП сообщения с использованием секретного ключа RSA.

- Функция `verify\_rsa` выполняет проверку подписи сообщения с использованием открытого ключа RSA.

На рисунке 1 приведен пример использования алгоритма RSA, включая генерацию ключей, шифрование/расшифровку сообщения и генерацию/верификацию ЭЦП. Выводятся само сообщение, его хеш, время выполнения операций, результат проверки и время выполнения проверки для RSA. С полным листингом программы можно ознакомится в приложение А.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1- Результат выполнения ЭЦП RSA

**Генерация и верификация ЭЦП на основе алгоритма Эль-Гамаля**

Алгоритм Эль-Гамаля - это криптографический алгоритм, основанный на задаче дискретного логарифмирования в конечных полях. Он также использует асимметричное шифрование. Алгоритм Эль-Гамаля включает в себя генерацию ключей, шифрование и расшифровку сообщений, а также генерацию и верификацию цифровой подписи.

**Генерация ключей Эль-Гамаля:**

1. Генерация большого простого числа p.

2. Выбор целого числа g, являющегося первообразным корнем по модулю p.

3. Выбор случайного целого числа x, такого что 1 < x < p-1.

4. Вычисление y = g^x (mod p).

5. Пара (p, g, y) является открытым ключом, а число x - секретным ключом.

**Шифрование Эль-Гамаля:**

1. Представление сообщения в виде целого числа m, меньшего p.

2. Выбор случайного целого числа k, такого что 1 < k < p-1 и НОД(k, p-1) = 1.

3. Вычисление первой части зашифрованного сообщения: a = g^k (mod p).

4. Вычисление второй части зашифрованного сообщения: b = y^k \* m (mod p).

**Расшифровка Эль-Гамаля:**

1. Получение зашифрованного сообщения a и b.

2. Вычисление обратного элемента к a^x (mod p).

3. Расшифровка сообщения: m = (b \* (a^x)^-1) (mod p).

**Генерация и верификация ЭЦП Эль-Гамаля:**

1. Выбор случайного целого числа k, такого что 1 < k < p-1 и НОД(k, p-1) = 1.

2. Вычисление первой части ЭЦП подписи: r = g^k (mod p).

3. Вычисление второй части ЭЦП подписи: s = (H(m) - x \* r) \* k^-1 (mod p-1), где H(m) - хеш сообщения, x - секретный ключ, k^-1 - обратный элемент к k по модулю p-1.

4. Пара (r, s) является ЭЦП подписью.

**Верификация ЭЦП Эль-Гамаля:**

1. Получение хеша сообщения H(m).

2. Вычисление левой части проверки: v1 = (y^r \* r^s) (mod p).

3. Вычисление правой части проверки: v2 = g^H(m) (mod p).

4. Проверка условия v1 = v2. Если условие выполняется, то подпись считается верифицированной.

Реализация Эль-Гамаля также может включать использование библиотеки `cryptography`. В приведенном коде используются функции из этой библиотеки для генерации ключей, шифрования, расшифровки и генерации/верификации ЭЦП Эль-Гамаля.

- Функция `generate\_elgamal\_keys` генерирует открытый и секретный ключи Эль-Гамаля.

- Функция `sign\_elgamal` вычисляет ЭЦП сообщения с использованием секретного ключа Эль-Гамаля.

- Функция `verify\_elgamal` выполняет проверку подписи сообщения с использованием открытого ключа Эль-Гамаля.

На рисунке 2 представлено использования алгоритма Эль-Гамаля, включая генерацию ключей, шифрование/расшифровку сообщения и генерацию/верификацию ЭЦП. Выводятся само сообщение, его хэш, время выполнения операций, результат проверки и время выполнения проверки для Эль-Гамаля.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2 - Результат выполнения ЭЦП Эль-Гамаля

**Генерация и верификация ЭЦП на основе алгоритма Шнорра**

Алгоритм Шнорра - это криптографический алгоритм, который используется для создания цифровой подписи. Он является асимметричным алгоритмом, основанным на проблеме вычисления дискретного логарифма в конечных полях.

Алгоритм Шнорра включает в себя генерацию ключей, создание подписи и верификацию подписи.

**Генерация ключей Шнорра:**

1. Генерация большого простого числа p.
2. Выбор целого числа q, которое является простым и является делителем числа p-1.
3. Выбор случайного целого числа g, такого что g^q mod p = 1 и g != 1.
4. Выбор случайного секретного числа x, такого что 0 < x < q.
5. Вычисление открытого ключа y = g^x mod p.
6. Пара (p, q, g, y) является открытым ключом, а число x - секретным ключом.

**Создание подписи Шнорра:**

1. Получение сообщения m.
2. Выбор случайного целого числа k, такого что 0 < k < q.
3. Вычисление точки R = g^k mod p.
4. Вычисление хеша сообщения e = H(m), где H() - хеш-функция.
5. Вычисление значения s = (k - x \* e) mod q.
6. Пара (R, s) является цифровой подписью сообщения m.

**Верификация подписи Шнорра:**

1. Получение подписи (R, s) и сообщения m.
2. Вычисление хеша сообщения e = H(m).
3. Вычисление точки V = (g^s \* y^e) mod p.
4. Проверка условия R = V. Если условие выполняется, то подпись считается верифицированной.

Реализация алгоритма Шнорра может включать использование библиотеки **cryptography**. В приведенном коде используются функции из этой библиотеки для генерации ключей, создания и верификации подписи Шнорра.

* Функция **generate\_schnorr\_keys** генерирует открытый и секретный ключи Шнорра.
* Функция **sign\_schnorr** создает подпись сообщения с использованием секретного ключа Шнорра.
* Функция **verify\_schnorr** выполняет проверку подписи сообщения с использованием открытого ключа Шнорра.

На рисунке 3 также демонстрирует использование алгоритма Шнорра, включая генерацию ключей, создание подписи и верификацию подписи. Выводятся само сообщение, его хеш, время выполнения операций, результат проверки и время выполнения проверки для Шнорра.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 3 - Результат выполнения ЭЦП Шнорра

**Вывод**

В ходе выполнения работы были реализованы алгоритмы RSA, Эль-Гамаля и Шнорра в приложении на языке Python. Каждый из алгоритмов обеспечивает возможность генерации и верификации цифровой подписи, а также выполнения операций шифрования и расшифрования сообщений.

Алгоритм RSA основан на факторизации больших целых чисел, алгоритм Эль-Гамаля - на задаче дискретного логарифмирования, а алгоритм Шнорра - на проблеме вычисления дискретного логарифма в конечных полях. Каждый из этих алгоритмов обладает своими уникальными свойствами и применяется в различных областях криптографии.

Оценка времени выполнения операций позволяет сравнить производительность каждого алгоритма. В приведенном приложении были замерены времена выполнения генерации ключей, шифрования/расшифрования сообщений и генерации/верификации цифровой подписи для каждого из алгоритмов.

По результатам замеров времени выполнения можно сделать следующие выводы:

* Алгоритм RSA обладает самым длительным временем выполнения операций, особенно при генерации ключей и шифровании/расшифровании сообщений. Это связано с вычислительной сложностью факторизации больших чисел.
* Алгоритм Эль-Гамаля имеет более быстрое время выполнения операций по сравнению с RSA, особенно при генерации ключей и шифровании/расшифровании сообщений. Однако он требует больше вычислительных ресурсов при генерации и верификации цифровой подписи.
* Алгоритм Шнорра обеспечивает сравнительно быстрые операции генерации и верификации цифровой подписи, а также выполнения шифрования/расшифрования сообщений. Однако он требует предварительной генерации большого простого числа и выбора подходящих параметров.

В целом, выбор алгоритма криптографии должен основываться на требованиях безопасности, производительности и конкретных потребностях приложения. Алгоритм RSA обеспечивает широкую поддержку и надежность, но требует больше вычислительных ресурсов. Алгоритмы Эль-Гамаля и Шнорра имеют свои преимущества и могут быть предпочтительными в определенных сценариях, где требуется более эффективное выполнение операций подписи и шифрования.

**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.

**2. Охарактеризовать основные функции ЭЦП.**

• аутентифицирование лица, подписавшего сообщение;

• контроль целостности подписанного сообщения;

• защита сообщения от подделок;

• доказательство авторства лица, подписавшего сообщение, если это лицо отрицает свое авторство.

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

Сходства: (см. вопрос 2).

Различия:

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

• указанная бинарная последовательность зависит от содержания подписываемого сообщения.

**4. Охарактеризовать основные способы реализации ЭЦП.**

• на основе симметричных систем (с тайным ключом);

• на основе симметричных систем и посредника;

• на основе асимметричных систем (с открытым ключом).

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

Да, есть различия в использовании ключевой информации при передаче зашифрованных сообщений и при передаче подписанных сообщений (ЭЦП).

При передаче зашифрованных сообщений используется общий ключ шифрования, который используется как отправителем, так и получателем сообщения. Этот ключ должен быть известен только обеим сторонам и должен быть передан по надежному каналу связи, чтобы никто другой не мог получить доступ к зашифрованным данным.

При передаче подписанных сообщений используется открытый ключ отправителя и закрытый ключ получателя. Отправитель создает цифровую подпись, используя свой закрытый ключ, а получатель может проверить цифровую подпись, используя открытый ключ отправителя. В этом случае, открытый ключ отправителя может быть общедоступным и не требует передачи по надежному каналу связи, в отличие от общего ключа шифрования при передаче зашифрованных сообщений.

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

Криптостойкость ЭЦП (электронной цифровой подписи) на основе RSA, схемы Эль-Гамаля, схемы Шнорра и DSA основывается на сложности математических задач, которые являются основой для этих алгоритмов.

RSA – это асимметричный алгоритм, который основан на трудности факторизации больших целых чисел. При использовании RSA для создания ЭЦП, приватный ключ используется для подписи сообщения, а публичный ключ используется для верификации подписи. Сложность факторизации больших чисел делает алгоритм RSA криптостойким, однако при использовании коротких ключей или плохой реализации алгоритма, RSA может стать уязвимым к атакам.

Схема Эль-Гамаля – это асимметричный алгоритм, который основан на вычислительной сложности задачи дискретного логарифмирования. При использовании схемы Эль-Гамаля для создания ЭЦП, приватный ключ является случайным числом, а публичный ключ является результатом вычисления степени от базового числа и приватного ключа. Дискретное логарифмирование считается сложной задачей, что делает схему Эль-Гамаля криптостойкой.

Схема Шнорра – это асимметричный алгоритм, который также основан на задаче дискретного логарифмирования. При использовании схемы Шнорра для создания ЭЦП, приватный ключ является случайным числом, а публичный ключ является результатом вычисления степени от базового числа и приватного ключа. Алгоритм Шнорра является криптостойким, так как решение задачи дискретного логарифмирования является вычислительно сложной.

DSA – это асимметричный алгоритм, который также основан на задаче дискретного логарифмирования. При использовании DSA для создания ЭЦП, приватный ключ является случайным числом, а публичный ключ является результатом вычисления степени от базового числа и приватного ключа. Алгоритм DSA считается криптостойким и используется, например, в стандарте Digital Signature Algorithm для защиты информации в правительственных и коммерческих организациях.

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

RSA:

- Публичный ключ, который состоит из модуля и открытой экспоненты

- Приватный ключ, который состоит из модуля и закрытой экспоненты

Схема Эль-Гамаля:

- Публичный ключ, который состоит из открытого базового числа, открытого случайного числа и открытого значения, полученного в результате возведения базового числа в степень открытого случайного числа

- Приватный ключ, который представляет собой закрытое случайное число

Схема Шнорра:

- Публичный ключ, который состоит из открытого базового числа, открытого значения, полученного в результате возведения базового числа в степень закрытого случайного числа и открытого значения, полученного в результате применения функции хеширования к подписываемым данным

- Приватный ключ, который представляет собой закрытое случайное число

DSA:

- Публичный ключ, который состоит из открытого базового числа, открытого модуля, открытого значения, полученного в результате возведения базового числа в степень закрытого случайного числа и параметров, определяющих длину ключа

- Приватный ключ, который представляет собой закрытое случайное число

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

Криптостойкость: Все эти алгоритмы считаются криптостойкими, однако при использовании коротких ключей или плохой реализации алгоритма, они могут стать уязвимыми к атакам.

Безопасность: RSA и DSA подвержены атакам, если секретный ключ будет скомпрометирован, тогда как схема Эль-Гамаля и схема Шнорра могут быть уязвимы, если случайное число, используемое для создания подписи, будет использоваться несколько раз.

Скорость: RSA быстрее схемы Эль-Гамаля и схемы Шнорра при создании подписи, но медленнее при проверке подписи. DSA медленнее всех при создании подписи, но быстрее схемы Эль-Гамаля и схемы Шнорра при проверке подписи.

Длина ключа: Длина ключа RSA должна быть больше, чем у других алгоритмов, для обеспечения такой же степени безопасности. Длина ключа схемы Эль-Гамаля и схемы Шнорра должна быть такой же, как у RSA для обеспечения такой же степени безопасности. Длина ключа DSA может быть меньше, чем у RSA, но должна быть достаточно большой для обеспечения криптостойкости.

Использование: RSA и DSA широко используются в промышленности и в правительственных учреждениях. Схема Эль-Гамаля и схема Шнорра не так широко распространены, но используются в определенных приложениях, например, в криптовалюте.

**9. Охарактеризовать особенности государственного стандарта ЭЦП в Республике Беларусь.**

Использование алгоритма ГОСТ Р 34.10-2001 (или его модификации) для создания ключей и формирования ЭЦП. Этот алгоритм основан на криптографических примитивах, разработанных в России, и является обязательным для использования в государственных учреждениях.

Обязательное наличие сертификатов ключей ЭЦП. Сертификаты выдаются аккредитованными удостоверяющими центрами, которые имеют право осуществлять проверку подлинности электронных подписей.

Установление требований к формату электронной подписи, включая способы представления данных и форматы документов, которые могут быть подписаны.

Правовое признание электронной подписи в качестве равнозначной ручной подписи. Это значит, что электронная подпись имеет такую же юридическую силу, как и обычная подпись на бумажном документе.

Разработка процедур проверки подлинности электронной подписи, включая проверку целостности и подлинности данных, использованных для создания подписи, а также проверку правильности использования алгоритмов ЭЦП.

Установление требований к удостоверяющим центрам, включая требования к их компетенции, методам работы, стандартам безопасности и сертификации.

Обязательство организаций, которые используют электронные подписи, соблюдать определенные правила и процедуры, включая управление ключами, обеспечение безопасности информации, установление полномочий и ответственности сотрудников и т.д.