Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кузбасский государственный технический университет
имени Т.Ф. Горбачева»

Кафедра информационных и автоматизированных производственных систем

Составитель
Г. А. Алексеева

СЕРТИФИКАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания к самостоятельной работе

Рекомендовано цикловой методической комиссией
специальности СПО 09.02.07 Информационные системы
и программирование в качестве электронного издания
для использования в образовательном процессе

Кемерово 2018

Рецензенты:
Ванеев О. Н. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем
Чичерин И. В. – кандидат технических наук, доцент кафедры информационных и автоматизированных производственных систем

Алексеева Галина Алексеевна

Сертификация информационных систем: методические указания к самостоятельной работе [Электронный ресурс] для обучающихся специальности СПО 09.02.07 Информационные системы и программирование очной формы обучения / сост. Г. А. Алексеева; КузГТУ. – Электрон. издан. – Кемерово, 2018.

Приведено содержание самостоятельной работы, материал, необходимый для успешного изучения дисциплины.

Назначение издания – помощь обучающимся в получении знаний по дисциплине «Сертификация информационных систем» и организация самостоятельной работы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью самостоятельной работы обучающихся является получение новых знаний по дисциплине «Сертификация информационных систем».

Самостоятельная работа необходима для формирования у обучающихся способности самостоятельно решать задачи профессиональной деятельности, формирования умения и навыков планирования времени, формирования стремления развиваться и совершенствоваться.

Виды самостоятельной работы обучающихся указаны в таблице 1.

Таблица 1 – Виды самостоятельной работы

№ п/п	Вид самостоятельной работы
1	Выполнение заданий на самостоятельную работу, приведенных в данных методических указаниях
2	Оформление отчетов по практическим занятиям

Далее приведены задания на самостоятельную работу.

1 ФОРМИРОВАНИЕ И ИЗУЧЕНИЕ СВОЙСТВ МОДЕЛИ БЕЛЛА-ЛАПАДУЛА

1.1 ЦЕЛЬ РАБОТЫ

Цель работы – овладеть методикой формирования мандатных моделей доступа, исследовать возможности мандатных моделей доступа.

1.2 ОСНОВНЫЕ РАЗДЕЛЫ ИЗУЧАЕМОГО МАТЕРИАЛА

Дискреционные модели управления доступом позволяют пользователям без ограничений передавать свои права другим пользователям. Это и используется в троянских программах, которые перераспределяют права без уведомления их владельца. Модели мандатного управления доступом лишены подобного недостатка. Классической моделью данного типа является модель Белла-ЛаПадула.

Идеи, лежащие в основе модели Белла-ЛаПадула, взяты из мира бумажных документов, в котором каждому субъекту и объекту ставится в соответствие уровень безопасности – от нулевого (unclassified) до совершенно секретного (top secret). В этом случае для предотвращения утечки информации пользователю с низким уровнем безопасности не разрешается читать документ с более высоким уровнем секретности. Кроме того, запрещено помещать информацию в объекты, чей уровень безопасности ниже, чем у субъекта. Например, право чтения совершенно секретного файла никаким образом не может быть передано неклассифицированному пользователю.

Описание модели Белла-ЛаПадула содержит множества субъектов *S*, объектов *O*, прав доступа *A* и матрицу доступа *M*. Однако, в этой модели множества *S* и *O* не изменяются при переходе системы из состояния в состояние, и множество *A* содержит два права доступа:

– *read*;

– *write*.

В модели Белла-ЛаПадула используется решетка уровней безопасности L и функция $F:SUO \rightarrow L$, которая в данном состоянии системы сопоставляет субъекту или объекту уровень безопасности.

Множество V-состояний системы – это множество упорядоченных пар (F, M). Система определяется начальным состоянием v_0, определенным набором запросов R и функцией переходов $T:(VxR) \rightarrow V$, которая переводит систему из состояния в состояние при выполнении запроса. Однако самое важное отличие модели Белла-ЛаПадула заключается во введении ряда определений, которые являются необходимыми и достаточными условиями безопасности системы:

Определение 1. Правило «нет чтения вверх» NRU (no read up) или простая безопасность. Состояние (F, M) безопасно по чтению тогда и только тогда, когда для $\forall s \in S$ и для $\forall o \in O$ и $read \in W[s, o]$ $F(s) \geq F(o)$.

Определение 2. Правило «нет записи вниз» NWD (no write down), или *-свойство. Состояние (F, M) безопасно по чтению тогда и только тогда, когда для $\forall s \in S$ и для $\forall o \in O$ и $write \in M[s, o]$ $F(o) \geq F(s)$.

Определение 3. Состояние системы безопасно тогда и только тогда, когда оно безопасно по чтению и по записи.

Безопасность по чтению запрещает доступ низкоуровневого пользователя к чтению высокоуровневых объектов. *-свойство защищает высокоуровневые файлы от копирования информации высокоуровневым троянским конем в файлы, к которым имеет доступ по чтению низкоуровневый пользователь.

Белла и ЛаПадула доказали Основную теорему безопасности.

Теорема: система (v_0, R, T) безопасна при следующих условиях:

1. Состояние v_0 безопасно.

2. Матрица T такова, что любое состояние v, достижимое из v_0 при выполнении конечной последовательности запросов из R, также безопасно.

3. Если $T(v, c) = v^*$, где $v = (F, M)$ и $v\ ^*= (F^*, M^*)$, и переходы (T)-системы из состояния в состояние подчиняются следующим ограничениям (для $\forall s \in S$ и для $\forall o \in O$):

– если $read \in M^*[s, o]$ и read $M[s, o]$, то $F^*(s) \geq F^*(o)$;

– если $read \in M[s, o]$ и $F^*(s) < F^*(o)$, то $read\ M^*[s, o]$;

– если $write \in M^*[s, o]$ и $write\ M[s, o]$, то $F^*(o) \geq F^*(s)$;

– если $write \in M[s, o]$ и $F^*(o) < F^*(s)$, то $write\ M^*[s, o]$.

Достоинства модели Белла-ЛаПадула:

– понятность и простота реализации;

– решение проблемы троянских программ.

Недостатками этих моделей являются:

– разрешение доступа, не указанного в модели;

– проблема системы *Z*;

– нарушение безопасности доверенными субъектами;

– скрытые каналы утечки информации.

Хотя модель Белла-ЛаПадула содержит набор A прав доступа из двух операций: $read \in A$ и $write \in A$, ничто не мешает нарушителю использовать такие виды доступа, как *delete* или *create*. Это не приводит к утечке информации, но может послужить причиной ее уничтожения или потери.

Кроме того в модели Белла-ЛаПадула возникает следующая проблема: переход системы из состояния в состояние должен гарантировать не только безопасность последующих состояний, но и безопасный способ их достижения. В качестве примера можно привести так называемую систему *Z*, в которой при запросе субъектом доступа к объекту все сущности деклассифицируются до самого низкого уровня, и тем самым доступ разрешается. Это приводит к потере секретности и к деградации системы, хотя состояния по-прежнему безопасны, и *Z*-система удовлетворяет требованиям модели Белла–ЛаПадула и «Основной теореме безопасности».

Для решения проблемы *Z*-систем были введены требования сильного и слабого спокойствия. Правило сильного спокойствия гласит, что уровни безопасности субъектов и объектов никогда не меняются в ходе системной операции. Очевидный недостаток применения этого метода – потеря гибкости системы при выполнении операций.

Правило слабого спокойствия требует следующее условие: уровни безопасности сущностей никогда не меняются в ходе системной операции таким образом, чтобы нарушить заданную политику безопасности. Например, уровень объекта не должен меняться, когда к нему обращается некоторый субъект.

Отметим, что в описании модели не дается характеристика субъектов. Компьютерные системы обычно имеют администратора. Доверенные субъекты могут функционировать в интересах этого администратора или исполнять некоторые критические службы. В системе с реализованной моделью Белла-ЛаПадула доверенные субъекты не смогут работать, не нарушая правил безопасности.

Кроме того, серьезную проблему представляют скрытые каналы утечки информации. Рассмотрим систему, в которой существуют монитор безопасности и низкоуровневый субъект. Субъект пытается записать информацию в несуществующий высокоуровневый файл, периодически создаваемый и удаляемый высокоуровневым злоумышленником. Если нарушитель успеет создать файл к моменту записи, то субъект не узнает, кому попала его информация, так как он не сможет ее прочитать. Несмотря на то, что скрытые каналы могут быть выявлены другими методами, подобные тесты обычно проводятся на завершающей стадии разработки компьютерных систем, когда внесение изменений оборачивается огромной тратой материальных и интеллектуальных средств.

1.3 ПРИМЕР ВЫПОЛНЕНИЯ

1. Исходные данные

Рассмотрим систему, содержащую два субъекта и два объекта:

– S_1 с уровнем безопасности «секретно»;
– S_2 с уровнем безопасности «несекретно»;
– O_1 с уровнем секретности «несекретно»;
– O_2 с уровнем секретности «секретно».

Таким образом, мы можем построить решетку уровней:

1 ---------- S_1---------- O_2----------«секретно»
0 ---------- S_2---------- O_1----------«несекретно»

Очевидно, что S_1 может читать O_1 и O_2, а S_2 – только O_1. Записывать информацию разрешено S_1 в O_2, а S_2 в O_1 и O_2. Тогда файл описания модели будет выглядеть следующим образом:

```
// листинг файла инициализации b_l_m.ini
//1 – секретно; 0 – несекретно
S(1,1,0,0,0,0,0,0,0);
S(2,0,0,0,0,0,0,0,0);
O(1,0,0,0,0,0,0,0,0);
O(2,1,0,0,0,0,0,0,0);
ATTRNAME seclevelS IS ATTRS(1);
ATTRNAME seclevelO IS ATTRO(1);
RULES
READO IF(seclevelS[THISS]>=seclevelO[THISO])
WRITEO IF(seclevelO[THISO]>=seclevelS[THISS])
ENDRULES
```

Prefile и postfiie не используются.

2. В ходе работы с реализованной моделью Белла-ЛаПадула была проведена проверка на доступ к объектам по чтению и записи. Данные операции выполнялись в соответствии с правилами NRU и NWD: *S*1 смог читать *O*1 и *O*2, а *S*2 – только *O*1. Записывать информацию было разрешено *S*1 в *O*2 , а *S*2 – в *O*1 и *O*2. Операция, не определенная в модели Белла-ЛаПадула (например, смена атрибутов объекта), разрешена, что нарушает защищенность информации в системе.

3. Затем в файл описания модели были внесены изменения, чтобы реализовать проблему *Z*-системы. Для простоты реализован вариант *Z*-системы, в котором у запросившего доступ субъекта изменяется уровень безопасности до уровня объекта, и доступ тем самым разрешается. В этом варианте, так как меняется уровень не всех сущностей, а только одного субъекта

```
// новый вид файла инициализации – b_I_m-z.ini
S(1,1,0,0,0,0,0,0,0);
S(2,0,0,0,0,0,0,0,0);
O(1,0,0,0,0,0,0,0,0);
O(2,1,0,0,0,0,0,0,0);
ATTRNAME seclevelS IS ATTRS(1);
ATTRNAME seclevelO IS ATTRO(1);
RULES
READO
// если второй субъект хочет прочитать второй объект, то его уровень повышается до необходимого
```

```
    if(THISS==2 && THISO==2)
    {
    make_secret();  //  прописываем  новые  атрибуты
субъекта
    seclevelS[THISS]=AS(1,2), // заносим изменения в
массив первого атрибута. Это разрешает чтение второго
объекта
    }
    IF(seclevelS[THISS]>=seclevelO[THISO])
    WRITEO
    // после чтения второй субъект хочет записать ин-
формацию в первый объект. Для этого он понижает свой
уровень секретности до уровня объекта № 1
    if(THISS==2)
    {
    make_nonsecret();
    seclevelS[THISS]=AS(1,2);
    }
    IF(seclevelO[THISO]>=seclevelS[THISS])
    ENDRULES
    Содержимое файла prefile:
    void make_secret(void)
    {
    S(2, 1, 0,0,0,0,0,0,0);
    }
    void make_nonsecret(void)
    {
    S(2, 0, 0,0,0,0,0,0,0);
    }
```

Заполнение массива первых атрибутов субъектов *seclevelS*[] происходит в файле b_l_m-z.ini, так как данный массив не является глобальной переменной. Его изменения допустимы только в файле описания модели. Дополнительный файл postfile ничего не содержит.

После внесенных изменений была произведена проверка работы определенной таким образом Z-системы. Второй субъект повысил свой уровень секретности: ему было разрешено читать информацию, хранящуюся во втором объекте, затем он понизил степень доверия, и ему было разрешено произвести запись в объект № 1. Такая последовательность действий не противоречит правилам модели Белла-ЛаПадула, но приводит к утечке информации.

4. Далее, чтобы избежать проблему Z-системы, файл описания был изменен следующим образом:

```
S(1,1,0,0,0,0,0,0,0);
S(2,0,0,0,0,0,0,0,0);
O(1,0,0,0,0,0,0,0,0);
O(2,1,0,0,0,0,0,0,0);
ATTRNAME secievelS IS ATTRS(1);
ATTRNAME seclevelO IS ATTRO(1);
RULES
READO
if(THISS==2 && THISO==2) // для чтения секретного объекта субъект № 2 повышает степень доверия
{
make_secret();
seclevelS[THISS]=AS(1,2);
}
IF(seclevelS[THISS]>=seclevelO[THISO])
WRITEO
// субъект № 2 понижает уровень секретности для записи в несекретный
// объект № 1
if(THISS==2)
{ make_nonsecret();
seclevelS[THISS]=AS(1,2); }
// правило слабого спокойствия: если запись в несекретный объект № 1, то вернуть субъект на уровень «секретно» и таким образом не разрешить запись
if(THISS==2 &&THISO==1)
{
rnake_secret();
seclevelS[THISS]=AS(1,2);
}
IF(seclevelO[THISO]>=seclevelS[THISS])
ENDRULES
```

Файлы prefile и postfile остались прежними.

1.4 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

1. Получить доступ к набору служебных программ «Монитор безопасности», используя следующие программы:

– expert.exe (программа преобразования файла описания моделей в exe.файл);

– security.exe (основная программа «Монитора безопасности»).

2. Создать файл описания модели Белла-ЛаПадула при помощи языка описания. Исходные данные для его создания задаются преподавателем. Следует определить только права доступа по чтению и записи.

3. Используя программу expert.exe, получить exe-файл описания.

4. Запустить программу security.exe.

5. Выбрать в меню FILE созданный exe-файл.

6. Убедиться в защите, предоставляемой моделью Белла-ЛаПадула. Для этого необходимо выполнить следующие операции:

– чтения и записи для пар «субъект–объект», которые удовлетворяют правилам NRU и NWD. Убедиться в возможности выполнения таких операций;

– чтения-записи с нарушением правил модели Белла-ЛаПадула.

Убедиться в невыполнимости такого рода действий.

7. Выполнить одно из неописанных в файле описания модели действий, например, delete object. Убедиться, что оно разрешается моделью. Исследовать модель подобным образом для различных операций.

8. Выйти из программы security.exe и вернуться к редактированию файла описания модели.

9. Реализовать в файле описания *Z*-систему, позволяющую нарушить защиту модели Белла-ЛаПадула. На данном этапе возможно использование дополнительных файлов prefile и postfile.

10. Повторить выполнение пунктов 3–5.

11. Убедиться в возможности обхода защиты, предоставляемой мандатной моделью Белла-ЛаПадула, в системе *Z*. Для этого необходимо:

– снять атрибуты субъектов и объектов;

– выполнить чтение или запись высокоуровневого объекта;

– снова снять атрибуты сущностей.

12. Выйти из программы security.exe и продолжить редактирование файла описания модели.

13. Реализовать правило слабого спокойствия.

14. Повторить действия пунктов 3–5.

15. Убедиться защищенности информации и невозможности реализации проблемы *Z*-системы. Для этого необходимо повторить действия пункт 11.

16. Выйти из программы security.exe.

1.5 СОДЕРЖАНИЕ ОТЧЕТА

В отчете требуется привести следующие сведения:

1. Исходные данные:

– наборы субъектов и объектов, а также соответствующие им уровни секретности;

– файл описания модели Белла-ЛаПадула;

– содержимое файлов prefile и postfile, если таковые применялись.

2. Результаты, подтверждающие защищенность информации по чтению и записи в системе с реализованной моделью Белла-ЛаПадула и ее незащищенность при использовании других операций.

3. Последовательность действий при обходе защиты модели Белла-ЛаПадула в *Z*-системе и файл описания, который реализует угрозу.

4. Результаты применения правила слабого спокойствия для *Z*-системы.

5. Выводы по лабораторной работе.

1.6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. На основе каких идей построена модель Белла-ЛаПадула?

2. Какое правило позволяет решить проблему «троянских коней»?

3. Какие критические замечания предъявляют модели Белла-ЛаПадула?

4. В чем заключается проблема системы *Z*?

5. Как можно избежать недостатков *Z*-системы?

2 ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ

2.1 ЦЕЛЬ РАБОТЫ

Цель работы – изучить общие сведения о криптографических средствах защиты информации и электронной цифровой подписи.

2.2 ОСНОВНЫЕ РАЗДЕЛЫ ИЗУЧАЕМОГО МАТЕРИАЛА

2.2.1 Криптографические методы защиты информации

Важнейшим механизмом защиты компьютерной информации и документов является шифрование информации. Классические шифры представляют собой преобразование информации, основанное на математической или иной зависимости, причем существует особенность такой зависимости, позволяющая гарантированно восстановить, пошагово или единовременно, исходную информацию. Такая особенность называется ключом.

Системы шифрования должны подчиняться принципу, который требует полной открытости системы для изучения: стойкость криптографического преобразования должна зависеть только от ключа.

Криптография и ее приложения составляют направление науки криптологии, изучающей создание и модификацию алгоритмов шифрования. Другое направление – криптоанализ – изучает стойкость алгоритмов шифрования с позиций их взлома.

Простые шифры представляют собой криптографические преобразования, основанные на однократном применении элементарной математической или иной операции над исходным текстом.

К таким шифрам относятся:

– подстановка, или замена;

– перестановка;

– аналитическое преобразование;

– гаммирование;

– комбинированные.

При использовании простых шифров можно оценить стойкость алгоритма по известным классическим методам дешифрования. Поскольку такие шифры не применяются в реальных системах передачи данных, исследование проводится только в качестве модельной задачи.

При использовании сложных, реально используемых на практике алгоритмов шифрования используются математически более сложные методы, основанные на принципах теории вероятности, математической логики и алгебры.

Криптосистемы разделяются на два основных вида:

– симметричные;

– асимметричные (с открытым ключом).

В симметричных криптосистемах и для шифрования, и для дешифрования используется один и тот же ключ.

Порядок использования систем с секретным ключом следующий:

– безопасно создается, распространяется и сохраняется секретный ключ системы симметричного шифрования;

– отправитель использует быстрый симметричный алгоритм вместе с секретным ключом для получения зашифрованного текста, при этом производится аутентификация;

– происходит передача зашифрованного текста и секретного ключа, при этом секретный ключ никогда не передается по незащищенным каналам связи;

– получатель использует аналогичный механизм шифрования для восстановления исходного текста.

В системах с открытым ключом используются два ключа – открытый и закрытый, которые математически связаны друг с другом.

Информация шифруется с помощью открытого ключа, который доступен всем желающим, а расшифровывается с помощью закрытого ключа, известного только получателю сообщения.

Порядок использования систем с открытыми ключами следующий:

– безопасно создаются и распространяются открытые ключи, предназначенные для шифрования текста;

– создается секретный ключ, предназначенный для расшифрования текста;

– секретный ключ передается центром выдачи сертификатов его владельцу;

– открытый ключ размещается в базе данных и администрируется центром выдачи сертификатов;

– создается секретный ключ симметричного шифрования, используемый как сеансовый;

– с помощью этого ключа шифруется открытый текст;

– отправителем с помощью открытого ключа, полученного из базы данных центра выдачи сертификатов, зашифровывается сеансовый ключ;

– проверяется следующее условие: каждый открытый ключ должен иметь сертификат, подписанный центром выдачи сертификатов;

– сеансовый ключ пересылается получателю;

– получатель расшифровывает полученный ключ с помощью секретного ключа асимметричной системы;

– расшифровывается открытый текст.

Такая система имеет комплексную структуру. Текст в ней шифруется симметричным алгоритмом, но при пересылке ключа дополнительно используется асимметричный алгоритм.

В качестве основного может использоваться и открытый ключ.

Тогда сеансовые ключи не применяются, а основным механизмом, защищающим открытые ключи от подмены, становится электронная цифровая подпись.

2.2.2 Электронная цифровая подпись

Электронная цифровая подпись (ЭЦП) используется для подтверждения целостности и авторства данных. Как и в случае асимметричного шифрования, в данном методе применяются двухключевые алгоритмы с таким же простым вычислением открытого ключа из секретного и лабораторной невозможностью обратного вычисления. Однако назначение ключей ЭЦП совершенно иное. Секретный ключ применяется для вычисления ЭЦП, открытый ключ необходим для ее проверки. При соблюдении

правил безопасного хранения секретного ключа никто, кроме его владельца, не в состоянии вычислить ЭЦП какого-либо электронного документа.

Кроме того, существуют виды электронной цифровой подписи, основанные на использовании открытого ключа. В таком случае пользователи не используют секретный ключ, но и доказательство авторства становится проблематичным.

Такая подпись может строиться либо на основе сети доверия, либо на основе инфраструктуры открытых ключей.

Возможные сферы применения криптоалгоритмов делятся на две большие категории (рисунок 1) по признаку расположения: на ПК пользователя или в сети. В первом случае происходит защита данных, хранящихся внутри ПК, во втором – защита межсетевого обмена данными.

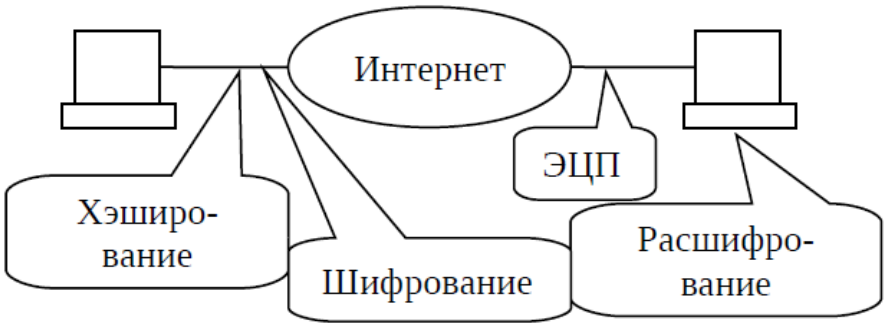

Рисунок 1 – Сферы применения ЭЦП

В первом случае пользователю нужно обеспечить два свойства важной информации – ее конфиденциальность, т. е. недоступность для всех тех, кому явным образом не разрешено ознакомление с данной информацией, и целостность, т. е. неизменность информации в процессе ее хранения.

Конфиденциальность данных достигается путем применения симметричного или асимметричного шифрования. Симметричное шифрование работает с одним и тем же секретным ключом для шифрования и для расшифрования данных, следовательно, если зашифровывает данные один пользователь, а расшифровывает – другой, то один из них (тот, кто создал ключ шифрования) должен передать ключ другому. Причем передача должна происходить «из рук в руки» или любым другим способом, ис-

ключающим возможность получения значения этого ключа посторонним лицом – иначе он сможет расшифровать не предназначенную ему информацию. Однако, если пользователь шифрует данные (т. е. он же и расшифрует их впоследствии), проблемы с передачей ключей не возникает.

Асимметричное шифрование существенно более ресурсоемко, чем симметричное. Иными словами, скорость асимметричного шифрования несравнимо ниже скорости симметричного шифрования при одинаковых ресурсах. Другая проблема асимметричного шифрования – необходимость защиты открытых ключей от подмены. Если злоумышленник подменит открытый ключ иным, то он сможет расшифровывать информацию вместо легального пользователя. По сложности решения данная проблема сравнима с необходимостью конфиденциальной передачи ключей симметричного шифрования.

Для проверки целостности информации применяются хэширование и электронная подпись, при этом:

– только электронная подпись позволяет определить авторство информации;

– электронная подпись требует предварительного хэширования данных, после чего вычисляется собственно подпись, что требует серьезных вычислительных ресурсов.

При защите целостности относительно большого количества данных маленького объема хэширование происходит на несколько порядков быстрее электронной подписи.

Алгоритм шифрования ГОСТ 28147–89, являющийся основным симметричным алгоритмам в отечественных средствах криптографической защиты информации, имеет также встроенный режим генерации имитовставки, которая может быть аналогом электронной цифровой подписи и применяться для аутентификации в автоматизированных системах.

2.3 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Данная самостоятельная работа предполагает выполнение следующих этапов:

– изучить методические указания;
– ответить на контрольные вопросы.

2.4 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается суть принципов создания электронной цифровой подписи?
2. Как реализовано управление ключами в системе?
3. Чем отличается электронная цифровая подпись от хэш-значения?
4. Какие выделяют области применения электронной цифровой подписи?
5. Чем отличается ассиметричное и симметричное шифрование?