Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра радиотехнических систем

ИССЛЕДОВАНИЕ МЕТОДОВ ЗАЩИТЫ ИНФОРМАЦИИ В СИСТЕМАХ СБОРА ДАННЫХ

Методические указания
к лабораторной работе по курсам
«Теория кодирования и защита информации»,
«Теория кодирования и основы криптологии»
для студентов радиотехнических специальностей
всех форм обучения

Минск БГУИР 2011

УДК 004.056.55(076.5)
ББК 32.973.26-018.2я73
И85

С о с т а в и т е л и:
С. Б. Саломатин, Д. М. Бильдюк

Исследование методов защиты информации в системах сбора
И85 данных : метод. указания к лаб. работе по курсам «Теория кодирования и защита информации», «Теория кодирования и основы криптологии» для студ. радиотех. спец. всех форм обуч. / сост. С. Б. Саломатин, Д. М. Бильдюк. – Минск : БГУИР, 2011. – 28 с. : ил.

Методические указания содержат теоретические сведения по организации систем защиты информации на базе индуктивных RFID-устройств с использованием алгоритмов криптографического преобразования информации для организации протоколов аутентификации, обеспечения секретности и целостности передаваемой информации. В издании изучается протокол передачи информации в соответствии со спецификациями ISO 14443, алгоритм аутентификации и идентификации, исследуются качественные характеристики ключевого пространства систем защиты информации на базе симметричных шифров.

УДК 004.056.55(076.5)
ББК 32.973.26-018.2я73

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

1 Изучить основные принципы работы индуктивных RFID-устройств.

2 Изучить методы организации систем защиты информации на базе индуктивных RFID-устройств.

3 Изучить принципы организации протоколов аутентификации, обеспечения секретности и целостности передаваемой информации.

4 Исследовать качественные характеристики ключевого пространства систем защиты информации на базе симметричных шифров.

5 Приобрести навыки построения криптографических протоколов и программ для организации систем защиты информации.

2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1 Базовые принципы функционирования систем радиочастотной идентификации

2.1.1 Основные компоненты RFID-систем

Система радиочастотной идентификации состоит из двух основных компонентов (рисунок 2.1):

– транспондер, закрепляемый на объекте, который должен пройти процедуру идентификации;

– считывающее устройство, или ридер, которое, в зависимости от приложения, может не только считывать, но и записывать данные.

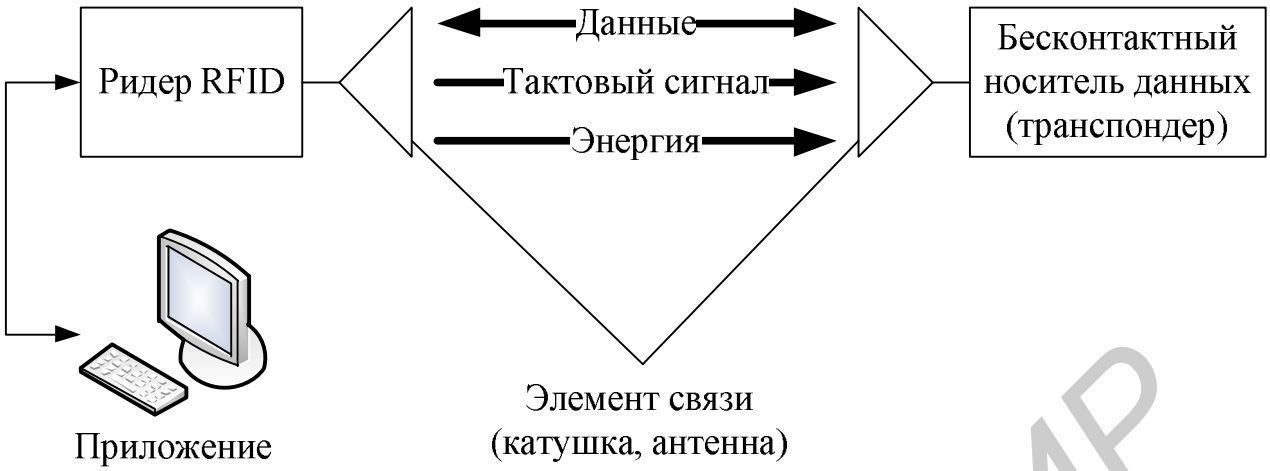


Рисунок 2.1 – Основные компоненты системы радиочастотной идентификации: слева – считывающее устройство, справа – транспондер

Считыватель обычно содержит радиочастотный модуль (передатчик и приемник), блок управления, включающий микропроцессор и память, и элемент связи с транспондером. Кроме того, многие считыватели оборудуются дополнительным интерфейсом (RS 232, RS 485, USB и др.), чтобы иметь возможность передавать принятые данные в другую систему (ПК, систему обработки данных и т. п.).

Транспондер представляет собой носитель данных RFID-системы и обычно включает в себя приемник, передающую схему, антенну и блок памяти для хранения информации. Приемник, передающая схема и память конструктивно выполняются в виде отдельной интегральной схемы (чипа), поэтому внешне кажется, что радиочастотная метка состоит всего из двух частей: многовитковой антенны и интегральной схемы. Иногда в состав конструкции радиочастотной метки включается автономный источник питания.

Процесс радиочастотной идентификации выполняется следующим образом:

– передатчик считывателя через антенну непрерывно (или в заданное время) излучает посылку радиосигнала с принятой в данной системе частотой;

– транспондер, находящийся в зоне действия считывателя, через свою антенну

принимает этот радиосигнал и использует его энергию для электропитания (в этом заключается пассивность идентификатора – ему не требуется источник питания). Транспондер считывает код из своего запоминающего устройства (ЗУ) и модулирует им ответный радиосигнал;

– считыватель принимает ответный сигнал, выделяет заключенный в нем код, проводит, если это предусмотрено, операции криптозащиты и процедуры антиколлизии (последовательной работы с несколькими идентификаторами, одновременно находящимися в зоне действия считывателя) и передает информацию по назначению: в приложение, системе обработки данных или оператору.

2.1.2 Системы RFID с индуктивной связью

Работа индуктивно связанной системы RFID основана на связи двух электрических цепей системы через магнитное поле. Индуктивная связь возможна только в ближней зоне антенны передатчика, где электромагнитное поле еще не отделено от антенны. Системы RFID, использующие индуктивную связь между транспондером и считывателем, работают на частоте ниже 135 кГц или в диапазонах частот 6,78; 13,56 и 27,125 МГц.

Транспондеры с индуктивной связью почти всегда работают в пассивном режиме. Это означает, что вся энергия, необходимая для работы микрочипа, должна подаваться от считывателя (рисунок 2.2).

Считыватель выполняет две основные задачи:

– генерацию электромагнитных колебаний, чтобы подать энергию на транспондер и привести его в действие;

– передачу данных и команд к транспондеру и получение данных от транспондера.

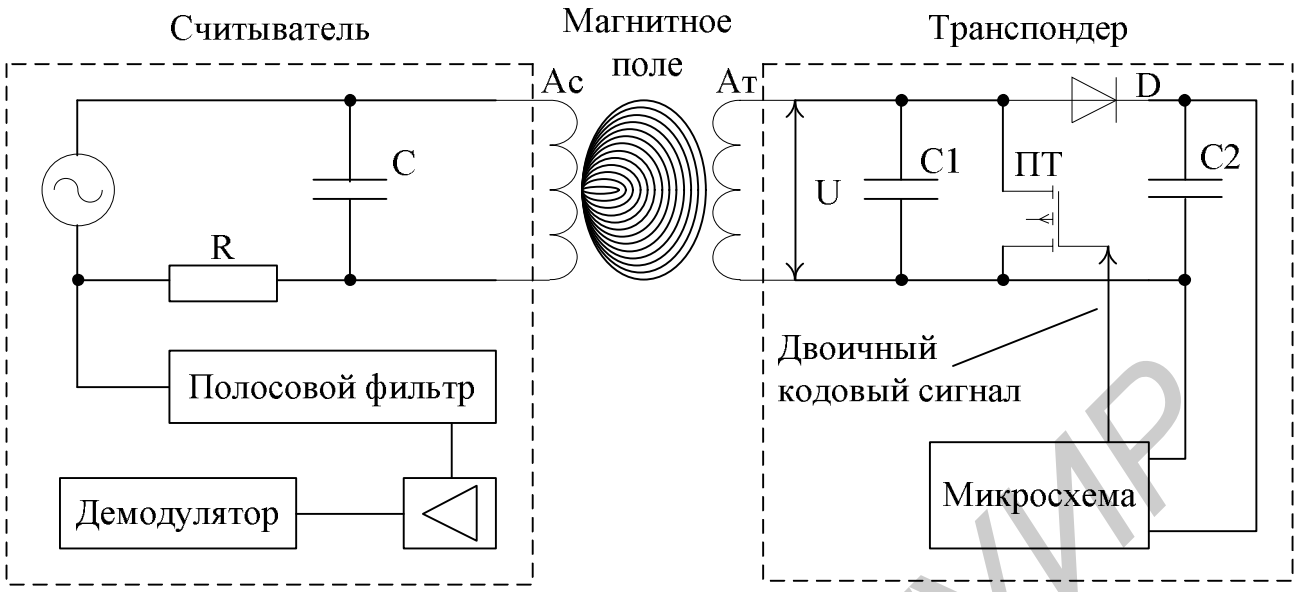


Рисунок 2.2 – Схема системы RFID с индуктивной связью

2.2 Обеспечение безопасности данных в системах RFID

Системы RFID все чаще используются в ответственных приложениях, требующих достаточного уровня безопасности. При передаче данных с использованием бесконтактной технологии очень вероятно, что могут возникнуть помехи, вызывающие нежелательные изменения в передаваемых данных и, соответственно, ведущие к ошибкам передачи. На системы RFID, работающие в таких приложениях, могут негативно воздействовать как случайные помехи, так и преднамеренные атаки злоумышленников, которые пытаются обмануть систему RFID, чтобы добиться неавторизованного доступа.

Критериями безопасности данных в системах RFID являются их целостность, конфиденциальность и доступность.

Для обеспечения безопасности данных необходимо поддерживать три основные функции:

– целостность данных;

– защиту передаваемых или хранимых в памяти данных от несанкционированного доступа;

– аутентификацию транспондеров и считывателей при установлении соединения.

Типичная структура транспондера, поддерживающего описанные выше функции, представлена на рисунке 2.3.

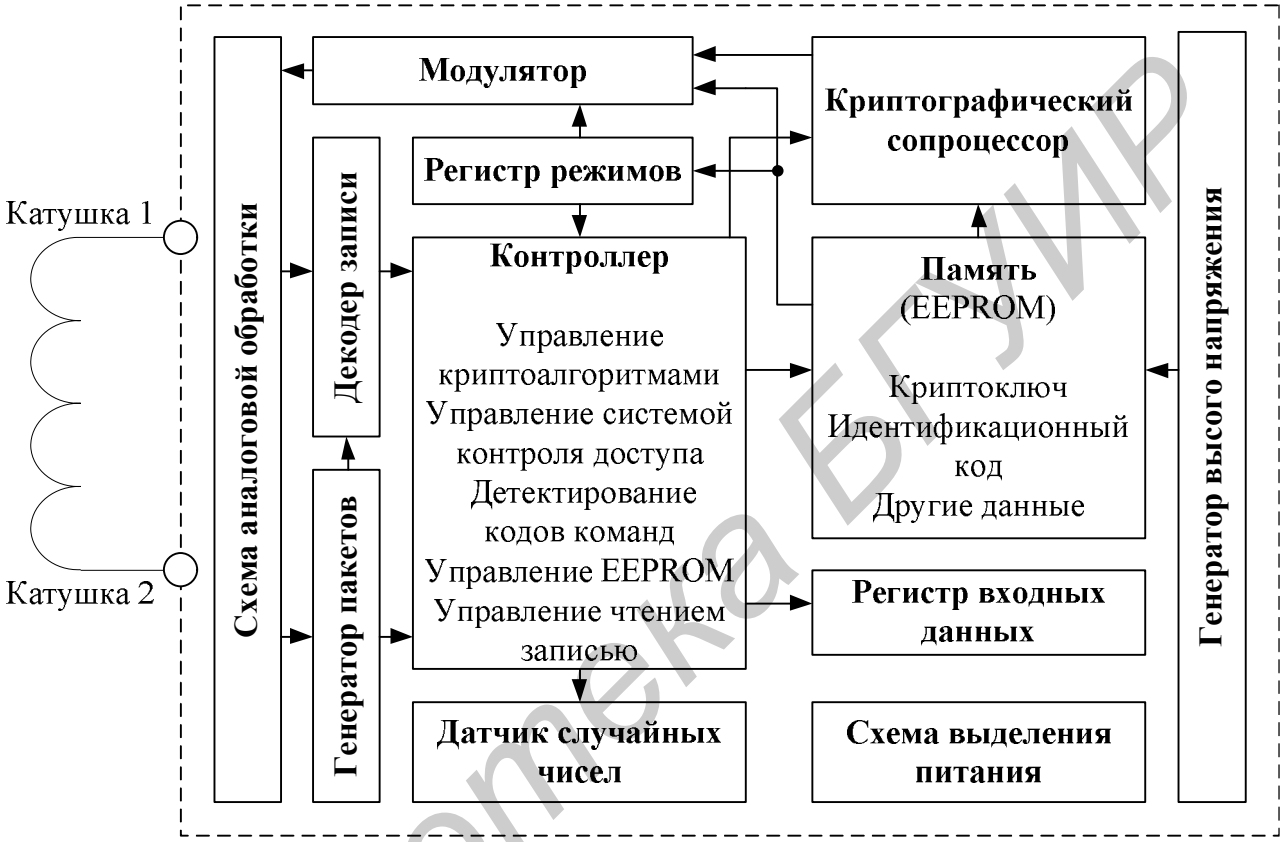


Рисунок 2.3 – Блок-схема транспондера

2.2.1 Обеспечение целостности данных

Под угрозой нарушения целостности понимается любое случайное или умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую.

В системах RFID для обеспечения целостности передаваемых данных используются процедуры проверки данных на четность, процедуры контроля целостности и исправления ошибок с помощью циклических избыточных кодов, а также процедуры с использованием однонаправленных функций хэширования.

2.2.2 Обеспечение конфиденциальности передаваемых данных

Различают два основных типа атак: пассивные и активные. Пассивный атакующий, не вмешиваясь в процесс передачи данных, старается ее подслушать и записать, чтобы раскрыть конфиденциальную информацию в противоправных целях. Активный атакующий, напротив, старается манипулировать передаваемыми данными и изменить их для своей выгоды.

Для защиты конфиденциальности передаваемых данных от пассивных и активных атак используются криптографические процедуры. В целях защиты передаваемых или хранимых в памяти данных (открытый текст) от несанкционированного доступа они могут быть зашифрованы до передачи таким образом, чтобы потенциальный атакующий не мог определить истинное содержание этих данных (открытого текста).

Следует отметить, что в RFID-системах преимущественно используются процедуры симметричного шифрования, в которых применяются одинаковые ключи для шифрования и расшифрования.

2.2.3 Аутентификация транспондера и считывателя на базе симметричных алгоритмов шифрования

Когда транспондер впервые входит в зону опроса считывателя, необходимо убедиться, что и транспондер, и считыватель принадлежат одному и тому же приложению. С точки зрения считывателя, необходимо защитить приложение от манипуляций, используя фальсифицированные данные. С позиций транспондера, существует необходимость защитить хранимые данные от неавторизованного чтения или перезаписи. Таким образом, необходима взаимная аутентификация считывателя и транспондера.

Взаимная аутентификация с использованием секретного криптоключа.

Взаимная аутентификация между считывателем и транспондером основывается на принципе взаимной аутентификации согласно ISO/IEC 9798-2,

при которой оба участника коммуникации проверяют знание другой стороной некоторого секрета (секретного криптоключа).

В этой процедуре все транспондеры и приемники, образующие некоторый сегмент приложения, владеют одним и тем же секретным криптоключом K для симметричного алгоритма шифрования.

Процедура взаимной аутентификации начинается с отправления считывателем транспондеру запроса (начать аутентификацию).

Отвечая, транспондер генерирует случайное число r_A (отклик) и отправляет его обратно на считыватель (рисунок 2.4).

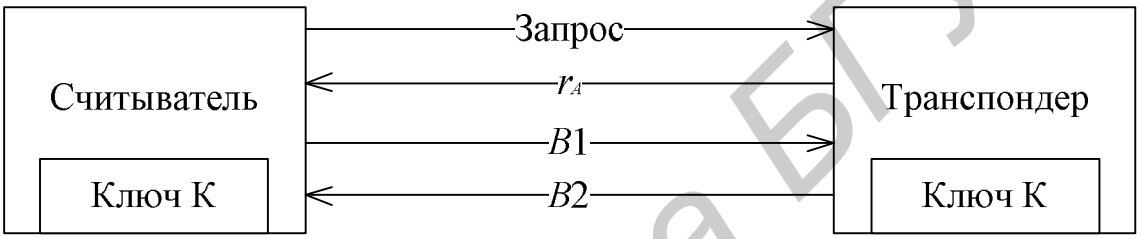


Рисунок 2.4 – Процедура взаимной аутентификации
между транспондером и считывателем

Теперь считыватель генерирует случайное число r_B. Используя общий секретный ключ K и общий алгоритм шифрования E_{K}, считыватель вычисляет блок шифрованных данных B_1, который содержит как случайные числа, так и дополнительные контрольные данные $Data_1$, и отправляет этот блок данных транспондеру: $B_1 = E_{\text{K}}(r_A, r_B, ID_A, Data_1)$.

В транспондере полученный блок данных B_1 расшифровывается и случайное число r'_A, содержащееся в восстановленном тексте, сравнивается с ранее переданным числом r_A. Если эти два числа совпадают, тогда транспондер убеждается, что у считывателя и него одинаковый общий ключ. В транспондере генерируется еще одно случайное число r_{A2}, оно используется для вычисления

блока шифрованных данных B_2, который содержит также r_B и контрольные данные $Data_2$. Блок B_2 отправляется от транспондера к считывателю: $B_2 = E_{\mathrm{K}}(r_{A2}, r_B, Data_2)$.

Считыватель расшифровывает блок B_2 и проверяет, соответствует ли только что полученное число r'_B ранее присланному числу r_B. Если эти два числа совпадают, тогда считыватель убеждается, что использование общего ключа доказано.

Таким образом, транспондер и считыватель удостоверились, что они владеют одним и тем же секретным криптоключом K и, следовательно, принадлежат одной и той же системе. Поэтому между этими двумя сторонами правомочно осуществлять дальнейший обмен информацией.

Существенным недостатком данной процедуры аутентификации является использование одинакового криптографического ключа для защиты всех транспондеров, входящих в одно приложение. Применение общего криптоключа является потенциальным источником опасности для приложений с очень большим количеством транспондеров.

Данный недостаток преодолевается при использовании для защиты транспондеров различных криптоключей.

Взаимная аутентификация с использованием выведенных криптоключей

Процедуру аутентификации транспондеров можно значительно усовершенствовать, защищая каждый транспондер своим криптографическим ключом. Чтобы добиться этого, во время производства транспондера считывается его идентификационный номер ID. Используя этот идентификационный номер ID, криптографический алгоритм и мастер-ключ $\mathrm{K_M}$, можно вычислить криптоключ K_X транспондера. Затем этот транспондер инициализируется. Таким образом, каждый транспондер получает криптоключ K_X, связанный с его собственным идентификационным номером ID и мастер-ключом $\mathrm{K_M}$.

Взаимная аутентификация начинается с запроса считывателем у транспондера номера *ID* (рисунок 2.5). Затем в специальном блоке безопасности (ББ) считывателя, основанном на выведенных ключах с использованием мастер-ключа K_M, для данного транспондера вычисляется криптоключ К*x*, который можно использовать для выполнения процедуры аутентификации.

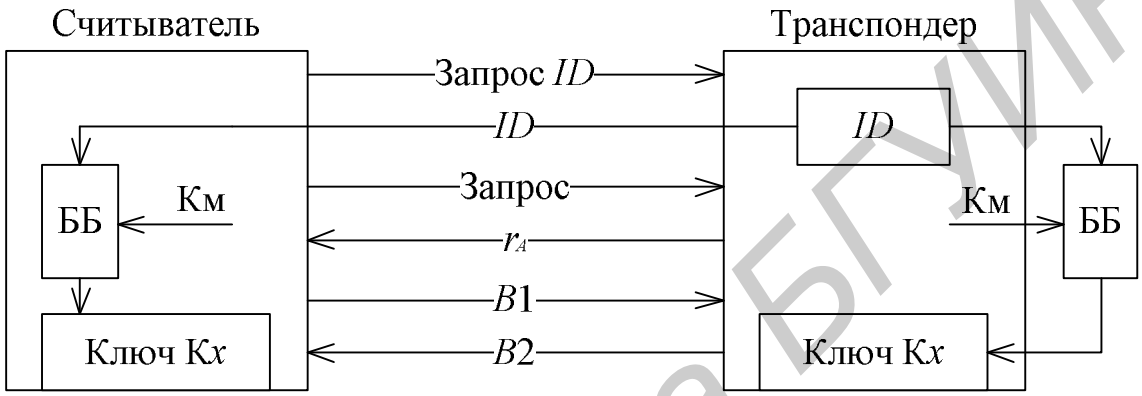


Рисунок 2.5 – Процедура взаимной аутентификации
с использованием выведенных криптоключей

2.2.4 Формирование ключа аутентификации на базе генераторов псевдослучайных чисел

Алгоритм взаимной аутентификации транспондера и считывателя реализует процедуру доказательства участниками знания секрета (общего ключа) без его разглашения. Очевидно, что знание злоумышленником ключа аутентификации дает ему возможность обмануть транспондер или считыватель. Поэтому стойкость алгоритма аутентификации зависит не только от используемого симметричного шифра, но и от качества алгоритма формирования ключа аутентификации. Если криптоаналитик может предсказать значение ключа, то он может успешно пройти процедуру аутентификации.

Для формирования ключа обычно используют генераторы псевдослучайных чисел. Генератор должен обладать следующими свойствами: иметь высокую периодичность, генерировать независимые (равновероятные) числа.

Определение. Генератор псевдослучайных чисел (ПСЧ) определяется через функцию $F: Z_m \to Z_n^L$, где m строго меньше, чем n^L. Входное начальное значение $x_0 \in Z_m$ называется зерном. Выходная последовательность из L чисел $x_1, x_2, ..., x_L$ называется последовательностью псевдослучайных чисел. Если $n = 2$, то функция F формирует псевдослучайные биты.

Линейный конгруэнтный генератор. Генераторы ПСЧ часто используют модулярную линейную конгруэнтную функцию. Пусть n будет положительным целым, $a, b \in Z_n$. Для $k > 0$ определим $x_k = (ax_{k-1} + b) \bmod n$.

В таком генераторе следующее формируемое число является линейной функцией от предыдущих чисел. Если число n выбрано простым, a является примитивным элементом в Z_n, а число $x_0 \neq 0$, то последовательность ПСЧ имеет период $\pi = n - 1$. Обычно такой генератор используется в программных средах и обозначается как *rand*.

Генератор на основе регистра сдвига с линейной обратной связью. Генератор на основе регистра сдвига с линейной обратной связью (РСЛОС). В основе генератора лежит рекуррентное соотношение. Пусть m будет положительным целым и пусть $c_0, c_1, ..., c_{m-1} \in Z_2$, $c_0 = 1$. Для $k \geq 0$ определим $x_{k+n} = c_0 x_k + c_1 x_{k+1} + ... + c_{m-1} x_{k+m-1} \mod 2$.

Положим, что не все элементы зерна $x_0, x_1, ..., x_{k-1} \in Z_2$ равны 0 и полином $m(y) = c_0 + c_1 y + ... + c_m y^{m-1} + y^m$ является примитивным в $Z_2[y]$. Полином $m(y)$ – неприводимый, а y является примитивным в поле $Z_2[y] / m(y)$. При соблюдении этих условий период псевдослучайного генератора бит равен $2^m - 1$.

Линейный конгруэнтный генератор и генератор на РСЛОС не удовлетворяют криптографическим требованиям. Причина состоит в том, что

криптоаналитик, имея достаточное число значений x_t , способен определить значения n, a, b и, следовательно, предсказать все значения x_t с вероятностью равной 1. Для криптографического генератора ПСЧ криптоаналитик может предсказать значения с вероятностью незначительно выше, чем $1/n$.

Покажем возможность взлома линейного конгруэнтного генератора. Заметим, что если $b = 0$, то

$$x_i = ax_{i-1} \bmod n = a^2 x_{i-2} \bmod n = x_0 a^i \bmod n .$$

Предположим, что криптоаналитик имеет значения x_1, x_2, x_3, x_4, x_5, используя которые можно составить следующие соотношения

$$x_1 x_4 \equiv x_0^2 a^5 \equiv x_2 x_3 \bmod n .$$

Из данного соотношения получаем, что $n \mid x_1 x_4 - x_2 x_3$. Кроме того, $n \mid x_1 x_3 - x_2^2$. Отсюда число

$$g = \text{НОД}((x_1 x_4 - x_2 x_3), (x_1 x_3 - x_2^2))$$

будет множителем n. Так как n – простое число и $0 < x_i x_j < n^2$, то n является простым наибольшим числом, делящим g. Следовательно, n может быть получено путем факторизации числа g. Процедура факторизации для больших значений n является достаточно трудоемкой. Для снижения сложности алгоритма можно использовать НОД для значений, разнесенных на большее число позиций. При этом g будет наименьшим множителем числа n.

Если известна величина модуля n, достаточно просто вычисляется значение a:

$$a = x_2 x_1^{-1} \bmod n .$$

Зная n и a, можно предсказать значение x_{L+1}. Программа подобных вычислений для функции rand программного обеспечения Maple приведена в приложении 1.

Генератор на РСЛОС может быть вскрыт с помощью алгоритма Берлекэмпа-Месси, который, используя не менее чем $\log L$ принятых символов, может вычислить полином обратной связи $m(y)$.

Хаотичный генератор случайных чисел. Поведение хорошего генератора случайных чисел является хаотичным.

Наиболее своеобразной характеристикой хаотических систем является бифуркация, т. е. дивергенция траекторий для смежных начальных точек. Уровень бифуркации измеряется с помощью экспоненты Ляпунова λ, которая для непрерывных систем определяется как

$$\left|f'(S_0+\varepsilon)-f'(S_0)\right|=\varepsilon e^{t\lambda},$$

где S – состояние дескриптора,

f – функция перехода из одного состояния в другое,

t – число итераций,

ε – малое приращение начального состояния.

Формально, экспонента Ляпунова описывается через предел λ для $\varepsilon\to 0$ и $t\to\infty$:

$$\lambda(S_0)=\lim_{t\to\infty}\lim_{\varepsilon\to 0}\frac{1}{t}\log_e\left|\frac{f'(S_0+\varepsilon)-f'(S_0)}{\varepsilon}\right|.$$

Применительно к дискретным системам положим, что состояние дескриптора S может быть записано через двоичный код $S=(S_{n-1},S_{n-2},...,S_0)_2=\sum_{i=0}^{n-1}2^iS_i$.

Тогда для оценки различия между двумя состояниями S,S' можно применить расстояние Хэмминга $d_H(S,S')$.

Так как наименее возможное значение ε равно 1, экспоненту Ляпунова для дискретных систем можно определить как

$$\lambda(S)=\frac{1}{t}\log_e d_H(f'(S),f'(S')),$$

где $d_H(S,S')=1$,

t выбираются из области, где расстояние Хэмминга между двумя траекториями растет экспоненциально.

Наибольшее значение расстояния Хэмминга равно n. Среднее значение расстояния между случайными состояниями равно (n/2).

Для формирования хаотичной последовательности ПСЧ можно воспользоваться следующими алгоритмами, использующими рекуррентное соотношение Фибоначчи с преобразованием двоичных кодов чисел:

1 $X_n = \text{rot}\, r\{(X_{n-j} + X_{n-k}) \bmod 2^b\}$;

2 $X_n = (\text{rot}\, r_1\{X_{n-j}\} + \text{rot}\, r_2\{X_{n-k}\}) \bmod 2^b$;

3 $X_n = (\text{rot}\, r_1\{X_{n-j}\} + \text{rot}\, r_2\{X_{n-j}\} + \text{rot}\, r_3\{X_{n-k}\}) \bmod 2^b$;

4 $X_n = Y_n + Z_n 2^{b/2}$, $Z_n = (\text{rot}\, r_3\{Y_{n-j}\} + \text{rot}\, r_1\{Y_{n-k}\}) \bmod 2^{b/2}$,

5 $Y_n = (\text{rot}\, r_4\{Z_{n-j}\} + \text{rot}\, r_2\{Z_{n-k}\}) \bmod 2^{b/2}$;

где X_n – бинарное число из n бит, Y_n и Z_n состоят из b/2 бит;

i, j и k – различные целые числа, $0 < i < j < k$;

для первых трех соотношений $r \in [0, b)$, для четвертого варианта $r \in [0, b/2)$.

Оператор rot r выполняет циклический сдвиг кода на r позиций вправо. Например (rot $3\{00001111_2\} = 11100001_2$.

Определение. Псевдослучайный генератор, формирующий последовательность бит, удовлетворяет требованиям криптографической безопасности, если для упорядоченной выходной последовательности генератора из L бит $x_i, x_{i+1}, \ldots, x_{i+L-1}$ невозможно с помощью вычислений определить значения x_{i-1} или x_{i+L} с вероятностью незначительно выше, чем 1/2.

Генератор BBS. Пусть n образуется как произведения двух больших случайных простых чисел p и q. Причем выполняются соотношения $p \equiv 3 \bmod 4$ и $q \equiv 3 \bmod 4$. Произведение чисел p и q дает число Блюма $n = pq$. Выберем другое случайное целое число x, взаимно простое с n. Вычислим $x_0 = x^2 \bmod n$.

Это начальное значение генератора (зерно). Теперь можно начать вычислять биты.

Алгоритм BBS

Вход: зерно x_0 выбирается как квадратичный вычет в Z_n.

Выход: последовательность $z_1, z_2, ..., z_L$.

1 Последовательное вычисление $x_i = x_{i-1}^2 \bmod n$ для $1 \le i \le L$.

2 Последовательное преобразование $z_i = x_i \bmod 2$ для $1 \le i \le L$.

Генератор BBS строится на основе сложностно-теоретического подхода. Теория генератора BBS использует квадратичные вычеты по модулю n.

Свойство генератора BBS. Для получения i-го бита не нужно вычислять $i-1$ предыдущих битов. Если известны значения p и q, можно вычислить i-й бит напрямую из соотношения

$$x_i = x_o^{(2^i) \bmod((p-1)(q-1))}.$$

Устойчивость схемы к вскрытию основана на сложности разложения n на множители. Вы можете опубликовать n так, что кто угодно может генерировать биты с помощью генератора. Однако пока криптоаналитик не сумеет разложить n на множители, он не сможет предсказать выход генератора. Строго говоря, генератор BBS непредсказуем влево и непредсказуем вправо. Криптоаналитик, получив последовательность, сформированную генератором, не сможет предсказать ни следующий, ни предыдущий бит последовательности.

Генераторы истинно случайных последовательностей. Лучший способ генерации множества случайных битов – извлечение их из естественно случайных событий реального мира. Например, атмосферный шум, тепловой шум полупроводникового диода, шум радиоактивного распада и др. Главный недостаток подобных схем – возможные закономерности в генерируемой последовательности. Используемые процессы могут быть и случайными, но между физическим процессом и устройством применения находятся различные измерительные инструменты, которые могут привести к появлению таких проблем как смещение и корреляция.

Для устранения смещения можно использовать операцию суммирования по модулю 2 нескольких битов друг с другом. Так, если случайный бит смещен к 0 на величину e, то вероятность 0 можно записать как $P(0) = 0,5 + e$. Операция суммирования над двумя такими битами дает $P(0) = (0,5 + e)^2 + (0,5 - e)^2 = 0,5 + 2e^2$. Те же вычисления с 4 битами дают $P(0) = 0,5 + 8e^4$. Операция суммирования над m битами экспоненциально ведет к равной вероятности 0 и 1.

Один из методов формирования случайных чисел – нахождение множества с виду случайных событий и извлечение из них случайности. Для этого можно использовать быстрые однонаправленые хэш-функции.

3 ЛАБОРАТОРНЫЙ МАКЕТ

3.1 Структура лабораторного макета

Лабораторный макет состоит из 4-х основных компонентов (рисунок 3.1):

– RFID-смарт-карта;

– бесконтактный карт-ридер;

– RFID-сканер;

– персональный компьютер (ПК) с программным приложением.

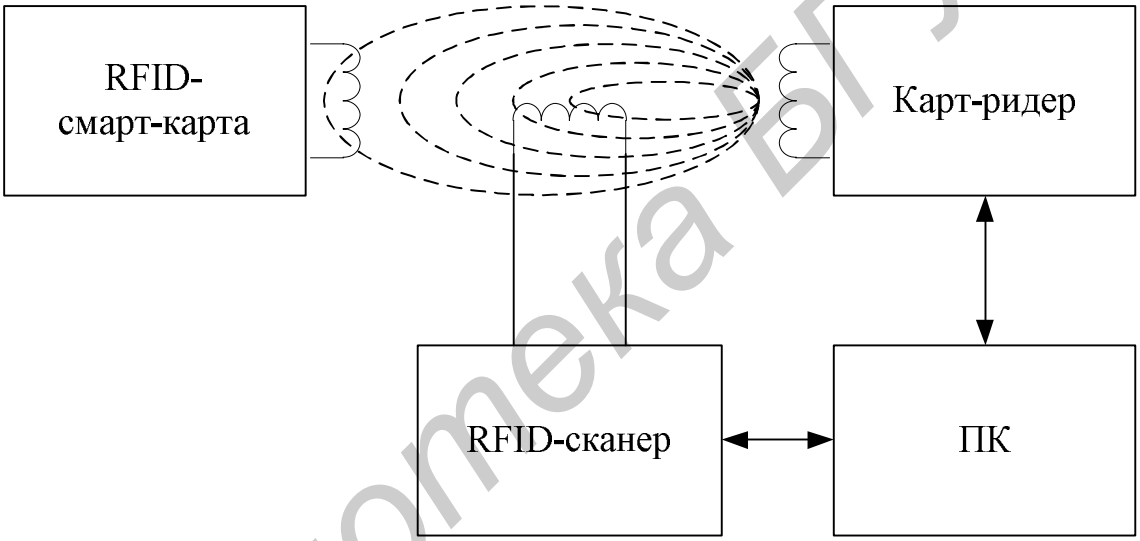


Рисунок 3.1 – структурная схема лабораторного макета

Смарт-карта представляет собой RFID-устройство с индуктивной связью стандарта ISO 14443. На уровне операционной системы смарт-карты реализовано приложение, исполняющее следующие функции:

– поддержка протокола взаимной аутентификации на базе симметричного алгоритма шифрования ГОСТ 28147;

– генерация случайных последовательностей на базе физического датчика случайных чисел.

Карт-ридер представляет собой устройство обмена данными с радиочастотными идентификаторами стандарта ISO 144443, управляемое при помощи программных приложений (функционирующих на ПК) с использованием интерфейса PC/SC.

RFID-сканер по своему функционалу подобен карт-ридеру, однако его основным назначением является прослушка канала «Транспондер-Считыватель» и передача соответствующей информации в приложение, функционирующее на ПК (используя интерфейс USB).

Программные приложения, функционирующие на ПК, позволяют:

– отправлять команды на карт-ридер;

– получать ответы на отправленные команды;

– осуществлять криптографическое преобразование данных;

– отображать данные, перехваченные RFID-сканером.

3.2 Интерфейс пользователя программных приложений

3.2.1 Приложение BCR_SPU.exe

BCR_SPU – программный симулятор карт-ридера соответствующего спецификациям PC/SC.

Программный симулятор обеспечивает подачу команд и прием ответных сообщений при информационном взаимодействии со смарт-картами.

Программный симулятор может работать как с программным симулятором смарт-карты, так и с аппаратной моделью.

Основное окно

Основное окно BCR_SPU (рисунок 3.2) разделено на следующие области:

– вверху – меню и панель инструментов;

– слева – область протокола исполнения;

– справа – область команд;

– внизу – строка состояния.

Размеры областей можно менять по своему усмотрению. Переключение между областями производится мышью.

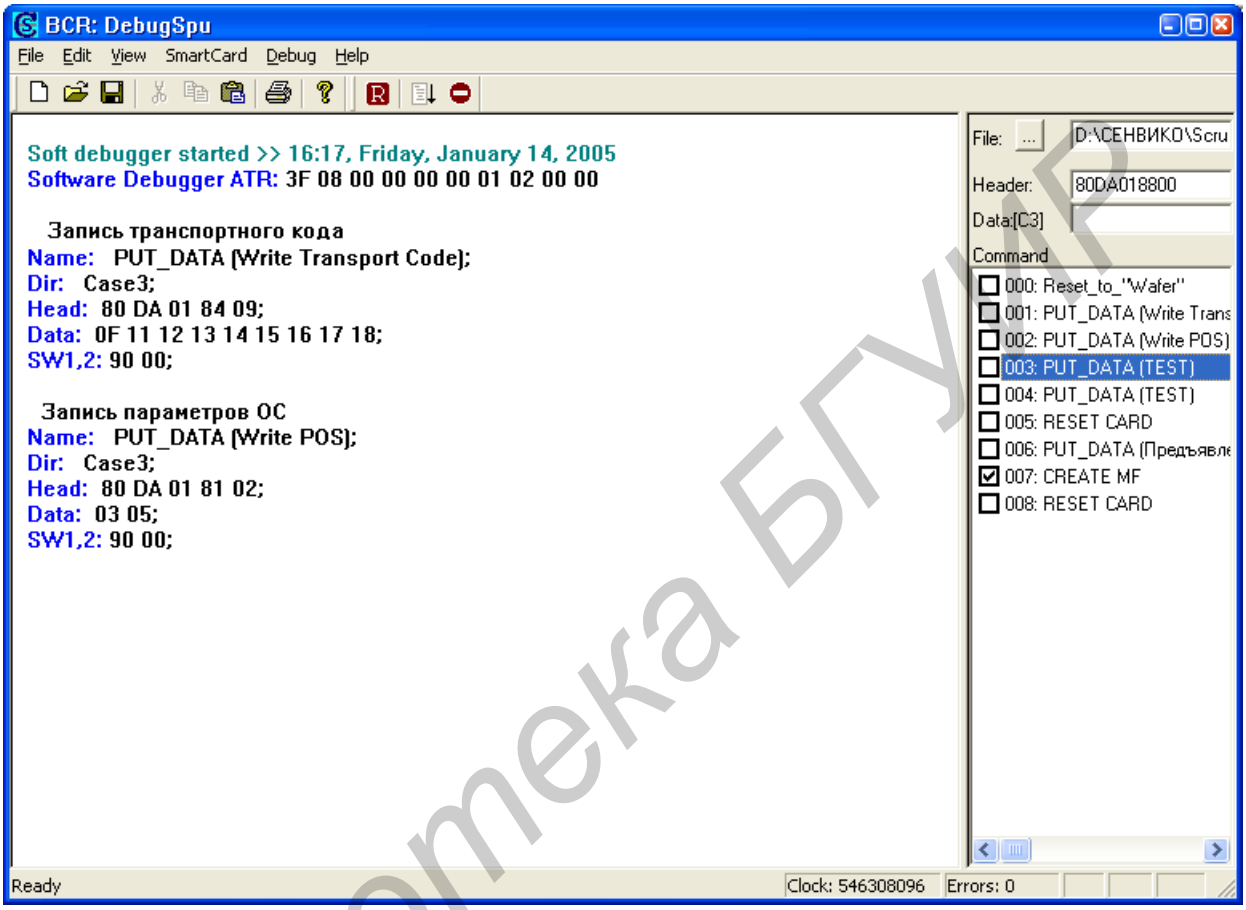
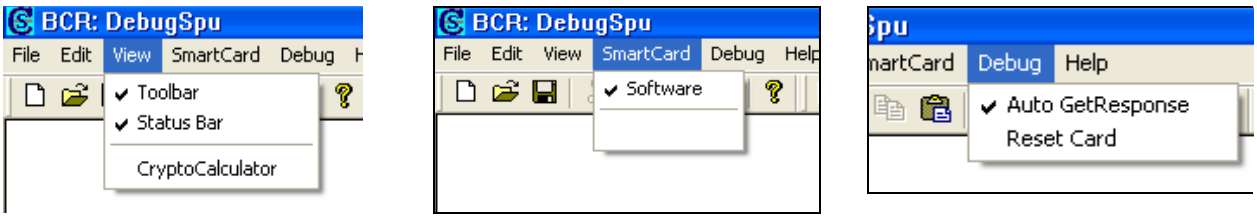


Рисунок 3.2 – Основное окно BCR_SPU

Панель инструментов и строку состояния (Status Bar) можно убирать и восстанавливать на экране. Для этого нужно установить флажки в списке меню View для тех областей, которые должны быть на экране, как показано на рисунке 3.3, а. В этом же меню можно вызвать крипто-калькулятор.

Для перехода в режим аппаратной модели нужно снять флажок в списке меню SmartCart для Software (рисунок 3.3, б).



а – меню View б – меню SmartCard в – меню Debug

Рисунок 3.3 – Списки меню

Для отмены режима автоматической подачи команды Get Response нужно снять флажок в списке меню Debug для Auto Get Response (рисунок 3.3, в).

Файл команд

Для работы с программным симулятором необходимо создать файл команд с расширением .txt (например Script.txt), содержащий набор команд. Формат данных, используемый для записи команд в файле, представлен на рисунок 3.4.

```
Name:  Наименование команды;
Desc:  Описание;
Dir:   Направление (Case);
Head:  Заголовок команды CLA, INS, P1, P2, Lc/Le;
Data:  Данные;
```

Рисунок 3.4 – Формат данных, используемый для записи команд в файле Script.txt

Пользовательская среда дает возможность: загрузить файл с необходимым набором команд; изменить некоторые данные команды; подать одну команду или блок команд; произвести перезапуск микроконтроллера; получить протокол исполнения команды; сохранить протокол исполнения в файле.

Загрузка файла команд

Нажмите кнопку  в области команд. На экране откроется диалоговое окно Open, (рисунок 3.5). Найдите и откройте необходимый файл.

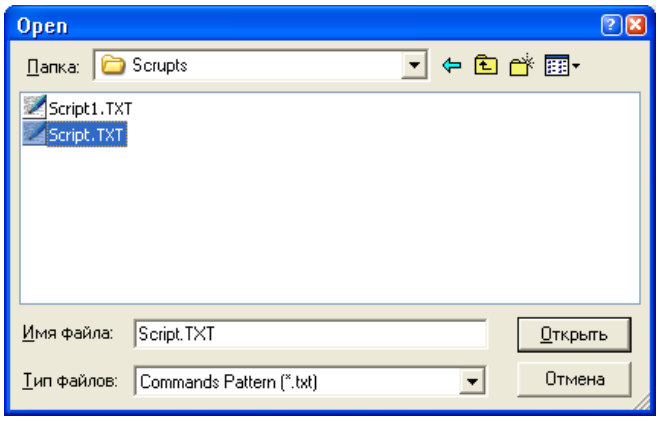


Рисунок 3.5 – Выбор файла команд

После загрузки файла в тестовом поле File появится его полное имя, а в поле Command список имен команд, содержащихся в файле.

Коррекция заголовка и данных выбранной команды

После выбора команды в списке имен команд в поле Command заголовок и данные команды появятся соответственно в полях Header и Data. Значения в этих полях можно изменить. Изменения будут сохраняться до конца сеанса. Содержимое файла команд останется неизменным.

Формат протокола исполнения

В процессе исполнения команд формируется протокол исполнения, который отражает поданные команды и полученные ответные сообщения. Формат данных протокола исполнения и примеры представлены на рисунках 3.6, 3.7.

```
Name:  Наименование команды;
Dir:   Направление (Case);
Head:  Заголовок команды CLA, INS, P1, P2, Lc/Le;
Data:  Данные;
SW1,2:  Статус завершения;
```

Рисунок 3.6 – Формат данных протокола исполнения

```
Software Debugger ATR: 3F 04 20 80 00 01

Name:   SELECT MF;
Dir:   Case1;
Head:  00 A4 00 04 00;
SW1,2: 61 1A;

Name:   GET RESPONSE;
Dir:   Case2;
Head:  00 C0 00 00 1A;
Data:  62 18 82 01 38 81 02 00 00 83 02 00 3F 8A 01 05 85 08 01 00 02 00 00 00 00;
```

Рисунок 3.7 – Фрагмент протокола исполнения

Сохранение протокола исполнения

Сохранить протокол исполнения можно с помощью команды *Save* меню File, либо с помощью кнопки *Save* на панели инструментов. Протокол можно сохранить в любом файле *.rcm. При работе с файлом *.rcm допустимо выполнение всех команд из меню File и Edit.

3.2.2 Библиотека PRNG.mpl

Библиотека PRNG.mpl при ее использовании в программной среде Maple реализует основные операции формирования случайных последовательностей для генерации ключевого пространства при взаимной аутентификации транспондера и считывателя.

3.2.3 Приложение Crack.exe

Данное приложение позволяет производить автоматическую взаимную аутентификацию транспондера и считывателя с ключом аутентификации в поле «Значение ключа» (рисунок 3.8).

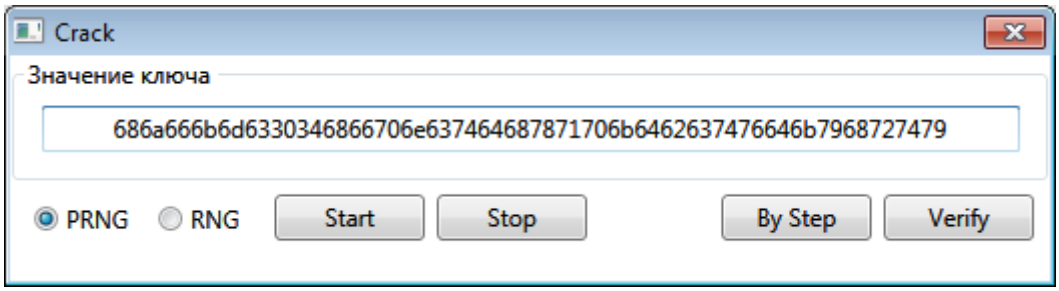


Рисунок 3.8 – Графический интерфейс приложения Crack.exe

Селективными кнопками «PRNG» и «RNG» выбирается используемый при автоматическом подборе генератор ключа аутентификации. «PRNG» – генератор, реализованный в библиотеке PRNG.mpl, «RNG» – генератор, реализованный в карте.

Кнопка «Start» запускает процесс аутентификации с автоматическим подбором ключа, кнопка «Stop» останавливает этот процесс.

Кнопки «By Step» и «Verify» позволяют выполнить процесс подбора ключа в пошаговом режиме («By Step» – выполняет один шаг, «Verify» – выполняет взаимную аутентификацию транспондера и считывателя).

4 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

4.1 Лабораторное задание

4.1.1 Изучение работы системы

1 С разрешения преподавателя подключите к ПК RFID-сканер и карт-ридер.

2 Расположите RFID-сканер над считывающей поверхностью карт-ридера.

3 Расположите бесконтактную карту над антенной сканера.

4 Запустите приложение BCR_SPU.exe.

5 Загрузите файл команд Script.txt.

6 Пошагово выполните команды:

- сброса карты;
- получения случайного числа;
- вычисления хэш-функции.

7 Запустите приложение RFIDTapping.exe, включите сканирование.

8 Повторите пункт 6, при этом зафиксируйте данные, принятые со сканера.

4.1.2 Аутентификация транспондера и считывателя

1 Оставьте над поверхностью сканера одну карту.

2 Загрузите файл команд Aauthenticate.txt в приложении BCR_SPU.exe.

3 Используя команды файла, сгенерируйте случайное число.

4 Используя библиотеку PRNG.mpl в программной среде Maple, сгенерируйте ключ аутентификации, употребив в качестве начального значения полученное случайное число.

5 Полученный ключ загрузите в карту, используя команды загруженного файла в BCR_SPU.

6 Проведите алгоритм аутентификации, используя при этом криптокалькулятор согласно пунктам 2.2.3 и 3.2.1.

7 Переключите приложение BCR_SPU в режим программной модели.

8 Запустите приложение Crack.exe и запустите процедуру подбора ключа (см. пункт 3.2.3).

9 Зафиксируйте время поиска.

10 Повторите пункты 3–9 девять раз.

11 Используя команды файла, сгенерируйте ключ аутентификации.

12 Повторите пункты 5–8.

13 Повторите пункты 11–12 девять раз, зафиксируйте результат.

5 СОДЕРЖАНИЕ ОТЧЕТА

1 Формулировка цели работы.

2 Предварительное задание.

3 Результаты изучения работы системы (команды, поданные на карту, полученные ответы, данные с RFID-сканера, комментарии к ним).

4 Аутентификация транспондера и считывателя (алгоритмы выработки ключа аутентификации, процесс взаимной аутентификации (команды и ответы), гистограммы времени поиска (подбора) ключа аутентификации).

5 Выводы.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1 Назовите основные компоненты RFID-систем, объясните их функциональное назначение.

2 Назовите основные особенности RFID-систем с индуктивной связью.

3 Назовите основные критерии и методы обеспечения безопасности в системах RFID.

4 Объясните суть взаимной аутентификации транспондера и считывателя.

5 Дайте определение генератору ПСП.

6 Поясните организацию и назовите основные свойства генератора на РСЛОС, хаотичного генератора, генератора BBS, генератора истинно случайных последовательностей.

ЛИТЕРАТУРА

1 Дшхунян, В. Л. Электронная идентификация. Бесконтактные электронные идентификаторы и смарт-карты / В. Л. Дшхунян, В. Ф. Шаньгин. – М. : ООО «Издательство АСТ» : Издательство «НТ Пресс», 2004.

2 Шнейер, Б. Прикладная криптография / Б. Шнейер. – М. : Триумф, 2002.

3 Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных : учеб. пособие для вузов / П. Ю. Белкин [и др.] – М. : Радио и связь, 1999.

4 Разработка и производство радиочастотных идентификационных средств. Инновационный проект. – М. : ОАО «Ангстрем», 2002.

5 Скородумов, Б. Безопасность союза интеллектуальных карточек и персональных компьютеров / Б. Скородумов // Мир карточек. – 2002. – №5–6.

6 Стандарт ISO-7816. Идентификационные карты – карты с микросхемой с контактами.

7 Тимофеев, П. А. Защита информации от несанкционированного доступа в современных компьютерных системах / П. А. Тимофеев // Конфидент. – 1998. – №5. – С. 55–59.

8 Тимофеев, П. А. Принципы защиты информации в компьютерных системах / П. А. Тимофеев // Конфидент. – 1998. – №3. – С. 72–76.

Св. план 2010, поз. 72

Учебное издание

Исследование методов защиты информации в системах сбора данных

Методические указания
к лабораторной работе по курсам
«Теория кодирования и защита информации»,
«Теория кодирования и основы криптологии»
для студентов радиотехнических специальностей
всех форм обучения

С о с т а в и т е л и:
Саломатин Сергей Борисович
Бильдюк Денис Михайлович

Редактор И. П. Острикова
Корректор Е. Н. Батурчик
Компьютерная верстка А. В. Тюхай

Подписано в печать 9.03.2011.	Формат 60x84 1/16.	Бумага офсетная.
Гарнитура «Таймс».	Отпечатано на ризографе.	Усл. печ. л. 1,74.
Уч.-изд. л. 1,5.	Тираж 150 экз.	Заказ 531.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровки, 6