УДК 631.39

ЗАЩИТА ОТ ПОМЕХ В СЕНСОРНЫХ СЕТЯХ МОНИТОРИНГА НА ОСНОВЕ НИЗКОПЛОТНОСТНЫХ КОДОВ КОНЕЧНЫХ ГЕОМЕТРИЙ

Т. А. АНДРИЯНОВА, С. Б.САЛОМАТИН

Белорусский государственный университет информатики и радиоэлектроники
П. Бровки, 6, Минск, 220013, Беларусь

Аннотация. Предложены схемы помехоустойчивого кодирования информации в сенсорных сетях мониторинга с использованием сигнатур и алгоритмов кодирования в конечных полях. Даны оценки вычислительной сложности и помехоустойчивости алгоритмов
Ключевые слова: сигнатура, конечное поле, функция следа, кодовые структуры.

Abstract. The schemes of noise-immune encoding of information in sensor monitoring networks are proposed with the use of signatures and coding algorithms in finite fields. Estimates of the computational complexity and noise immunity of algorithms are given
Key words: signature, finite field, trace function, code structures.

Введение

Низкоплотностные LDPC коды, использующие алгоритмы распространения доверия для декодирования обладают хорошими корректирующими возможностями, что обуславливает применение таких кодов в современных системах связи и сенсорных сетях мониторинга [1, 2]. Основная трудность при синтезе проверочной матрицы LDPC кода связана с требованием отсутствия циклов в двудольном графе Таннера [3, 4]. Наличие циклов в двудольном графе кода является основным фактором снижения исправляющей способности алгоритма декодирования. Циклы приводят к зависимости декодируемого значения на текущей итерации от значений декодирования этого же символа на предыдущих итерациях.
Одним из путей решения проблемы циклов является применение конечный геометрий и схем инциденций над конечными полями и декодирования кода алгоритмами с итеративным распространением доверия (IBP, iterative belief-propogation).

Конструкции LDPC кодов на основе схем инциденций

LDPC-коды часто задаются с помощью графа, для которого матрица *H* является матрицей смежности (так называемого графа Таннера). Это двудольный граф, вершины которого делятся на два множества: 1) n символьных вершин (bit nodes), соответствующих столбцам; 2) r проверочных вершин (check nodes), соответствующих строкам проверочной матрицы. Ребра, соединяющие вершины графа, соответствуют ненулевым позициям в матрице H.
Коды, построенные с использованием конечных геометрий [5, 6], не имеют циклов длиной 4. Особый интерес представляют (квази-) циклические LDPC коды с охватом (girth) g = 6.
С конечной геометрией *G* можно ассоциировать $J \times n$ матрицу инциденций $H_G^{(1)} = [h_{i,j}]$, где каждая строка и столбец соответствуют линии и точки, соответственно. Элемент $h_{i,j} = 1$ если и только если *j*-ая точка соответствует i- ой линии в G, в противном случае $h_{i,j} = 0$.
Строки и столбцы веса ρ и γ соответствуют ρ точкам в соответствующей линии и γ линиям, проходящим через соответствующие точки. Если $\rho \ll n$ и $\gamma \ll J$, то матрица $H_G^{(1)}$ может рассматриваться как проверочная матрица LDPC кода. Коды, соответствующие $H_G^{(1)}$, носят название кода геометрии I GLDPC, имеющего блоки длиной n. Взаимно заменяя строки и

столбцы в матрице инциденций, получаем код геометрии II GLDPC с проверочной матрицей $H_G^{(1)} = \left(H_G^{(1)}\right)^T$.

Конструкции LDPC кодов, свободных от циклов длиной 4, могут быть получены на основе балансных блоковых схемах (BIBD). Схемы BIBD имеют параметра (v, k, λ, r, b) и представляют собой упорядоченные пары (X, A) у которых v точек множества X разбивается в семействе A на b подмножеств (блоков), у которых любые две точки определенны λ блоками с k точками в каждом блоке и каждая точка присутствует в r различных блоках.

Так как $bk = vr$ и $\lambda(v-1) = r(k-1)$, то только три из пяти параметров независимые. Поэтому запись $(v, k, \lambda) - BIBD$ соответствует схеме BIBD, построенной на v точках, имеющей блоки размеров k, и индекс λ.

Пример 1. Пусть $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, и *A* семейство 12 трех-элементных блоков:

$$A = \{(1,2,3), (4,5,6), (7,8,9), (1,4,7), (2,5,8), (3,6,9), (1,5,9), (2,6,7), (3,4,8), (1,6,8), (2,4,9), (3,5,7)\},$$

пара (X, A) является (9, 3, 1)-BIBD [30].

Структура *(v, k, λ)*-BIBD с $k = 3$ и $\lambda = 1$ называется системой тройки Штейнера. Системы троек Штейнера существуют для всех $v = 1, 3 \bmod 6$.

Если множество блоков являются разбиениями точек множества *X* в BIBD, то множество блоков образуют параллельные классы. Разрешение BIBD структур оценивается через возможность разбиения семейства на параллельные классы. Разрешение дает точно *r* параллельных классов. BIBD с, как минимум, одним разрешением.

Пример 2. Для (9, 3, 1)-BIBD в системе тройки Штейнера множество A может быть разбито на четыре параллельных класса следующим образом:

$$\{(1,2,3), (4,5,6), (7,8,9)\}, \{(1,4,7), (2,5,8), (3,6,9)\}, \{(1,5,9), (2,6,7), (3,4,8)\}, \{(1,6,8), (2,4,9), (3,5,7)\}.$$

Определим блок-точку матрицы инциденций $H = [h_{i,j}]_{v \times b}$, где $h_{i,j} = 1$, если *i* -ая точка в X является в j-ом блоке A, и $h_{i,j} = 0$ в противном случае.

Если каждая точка $(v, k, \lambda) - BIBD$ с $k \geq 2$ и $\lambda = 1$, относится к проверочному уравнению, и каждый блок является битом линейного блокового кода, тогда *H* есть проверочная матрица размером $v \times b$ строки которой имеют вес *r*, а столбцы – вес *k*. Если $r \ll k$ и $k \ll b$, *H* является разреженной проверочной матрицей LDPC кода.

Пример 3. Проверочная матрица (9, 3, 1)-BIBD кода имеет следующий вид:

$$H = \begin{bmatrix} 1&0&0&1&0&0&1&0&0&1&0&0 \\ 1&0&0&0&1&0&0&1&0&0&1&0 \\ 1&0&0&0&0&1&0&0&1&0&0&1 \\ 0&1&0&1&0&0&0&0&1&0&1&0 \\ 0&1&0&0&1&0&1&0&0&0&0&1 \\ 0&1&0&0&0&1&0&1&0&1&0&0 \\ 0&0&1&1&0&0&0&1&0&0&0&1 \\ 0&0&1&0&1&0&0&0&1&1&0&0 \\ 0&0&1&0&0&1&1&0&0&0&1&0 \end{bmatrix}$$

Если в проверочной матрице *H* на двух столбцах и соответствующих им строках располагаются две единицы, то в ассоциированном с такой матрицей графе Таннера образуется цикл длиной 4. Для блок-матрицы инциденций *H* , построенной на *t*-схеме с $\lambda = 1$, такой вариант исключается. Поэтому для неё граф Таннера кода свободен от цикла длиной 4.

Декодирования LDPC кода с итеративным распространением доверия.

Алгоритм декодирования, соответствующий IBP, вычисляет точные апостериорные вероятности после некоторого количества итераций, если граф Таннера для данного кода не содержит циклов. Ниже предлагается передача двоичных сигналов по каналу АБГШ. Модулированные символы $m(v_i)$ передаются по Гауссовскому каналу и принимаются в виде $r_i = m(v_i) + w_i$, где w_i – гауссовская случайная величина с нулевым средним и дисперсией $N_0/2,\ 1 \le i \le N$.

1. Алгоритм с жестким принятием решения и процедурой суммирования-умножения (sum product algorithm). Пусть $\mathbf{h} = \{h_0,\dots,h_{N-1}\}$ любая строка из проверочной матрицы кода с низкой плотностью проверок на четность. Будем говорить, что проверка для некоего принимаемого вектора $\mathbf{y} = \{y_0,\dots,y_{N-1}\}$ выполняется тогда, когда скалярное произведение вектора y на проверку дает ноль. Элемент y_i принятого вектора **y** участвует в проверке с $\mathbf{h} = \{h_0,\dots,h_{N-1}\}$ тогда, когда соответствующий элемент проверки h_i не равен нулю.

Одна итерация «жесткого» декодирования инвертированием битов производится следующим образом.

1. Для принятого вектора вычисляются все проверки.
2. Если некоторый бит принятого вектора участвовал более чем в половине не выполнившихся проверок, бит инвертируется.
3. После такого анализа всех символов принятого вектора вектор проверяется на принадлежность коду. Если вектор является кодовым словом, декодирование заканчивается, в противном случае выполняется следующая итерация алгоритма.

Фазы шагов итеративного процесса.

В первой фазе первого шага итеративного процесса *n*-й символьный узел, $n = 0,1,\dots,N-1$, посылает логарифмическое отношение правдоподобия $z_{n,l}^{(0)} = z_n^{(0)}$, называемое «сообщением», связанным с ним проверочным узлам $l \epsilon \mathcal{L}(n)$. Затем *i*-ый проверочный узел, получивший K сообщений $z_{n,l}^{(0)}, n \in \mathcal{N}(l)$, формирует K логарифмических отношений правдоподобия $y_{l,n}^{(1)}, n \in \mathcal{N}(l)$,

$$y_{l,n}^{(1)} = \log\frac{P(v_n = 0|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\})}{P(v_n = 1|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\})},$$

где $P(v_n = 0|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\})$ и $P(v_n = 1|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\})$ – условные вероятности того, что $v_n = 0, v_n = 1$, соответственно, при условии, что известно множество принятых символов $\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\}$.

При этом

$$P(v_n = 0|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\}) = \frac{1}{2}\left(\prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \left(P(v_{n'} = 1|r_{n'}) + P(v_{n'} = 0|r_{n'})\right) + \prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \left(P(v_{n'} = 1|r_{n'}) - P(v_{n'} = 0|r_{n'})\right)\right)$$

и

$$P(v_n = 1|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\})$$
$$= \frac{1}{2}\left(\prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \left(P(v_{n'} = 1|r_{n'}) + P(v_{n'} = 0|r_{n'})\right)\right.$$
$$\left. - \prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \left(P(v_{n'} = 1|r_{n'}) - P(v_{n'} = 0|r_{n'})\right)\right).$$

Логарифмическое отношение правдоподобия $y_{l,n}^{(1)}$ может быть выражено как функция (K – 1) логарифмических отношений правдоподобия $z_{n'}^{(0)}, n' \in \mathcal{N}(l)\backslash\{n\}$. Используя равенство

$$\frac{\left(e^{z_{n'}^{(0)}} - 1\right)}{\left(e^{z_{n'}^{(0)}} + 1\right)} = \tanh\left(\frac{z_{n'}^{(0)}}{2}\right)$$

получим

$$y_{l,n}^{(1)} = \log\frac{P(v_n = 0|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\})}{P(v_n = 1|\{r_{n'}, n' \in \mathcal{N}(l)\}\backslash\{n\})} = \log\frac{1 + \prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \tanh\left(\frac{z_{n',l}^{(0)}}{2}\right)}{1 - \prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \tanh\left(\frac{z_{n',l}^{(0)}}{2}\right)}.$$

Вычисление статистик $y_{l,n}^{(1)}, l = 0,1, \ldots, L-1, n \in \mathcal{N}(l)$ завершает первую фазу первого шага процесса итеративного декодирования.

Во время выполнения второй фазы первого шага итеративного процесса декодирования *l*-ые, $l = 0,1, \ldots, L-1$, проверочные узлы шлют логарифмические отношения правдоподобия $y_{l,n}^{(1)}, n \in \mathcal{N}(l)$, соответствующим смежным символьным узлам. Затем *n*-ый символьный узел, получивший сообщения от смежных проверочных узлов, вычисляет J логарифмических отношений правдоподобия $z_{n,l}^{(1)}, l \in \mathcal{L}(n)$,

$$z_{n,l}^{(1)} = z_n^{(0)} + \sum_{l' \in \mathcal{L}(n)\backslash\{l\}} y_{l',n}^{(1)} = z_n^{(0)} + \sum_{l' \in \mathcal{L}(n)\backslash\{l\}} \log\frac{1 + \prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \tanh\left(\frac{z_{n',l}^{(0)}}{2}\right)}{1 - \prod_{n' \in \mathcal{N}(l)\backslash\{n\}} \tanh\left(\frac{z_{n',l}^{(0)}}{2}\right)}.$$

Здесь $y_{l',n}^{(1)}, l' \in \mathcal{L}(n)\backslash\{l\}$, означают сообщения, называемые внешней информацией (extrinsic information), которые *l*-ый символьный узел получил от смежного проверочного узла.

Вычисление статистик $z_{n,l}^{(1)}, l \in \mathcal{L}(n)$, завершает вторую фазу первого шага процесса итеративного декодирования.

Рассмотрим *i*-ый, $i = 0,1, \ldots, I-1$ шаг итеративного процесса декодирования. Во время первой фазы *i*-ой итерации -ый символьный узел посылает сообщение $z_{n,l}^{(i-1)}$, которое было вычислено им на предыдущем шаге, смежному проверочному узлу $l \in \mathcal{L}(n)$.

Затем i -ый проверочный узел $l = 0,1, \ldots, L-1$, получивший K сообщений $z_{n,l}^{(i-1)}, n \in \mathcal{N}(l)$, от смежных символьных узлов, вычисляет статистики $y_{l,n}^{(i)}, n \in \mathcal{N}(l)$

$$y_{l,n}^{(i)} = \log \frac{1 + \prod_{n' \in \mathcal{N}(l) \setminus \{n\}} \tanh\left(\frac{z_{n',l}^{(i-1)}}{2}\right)}{1 - \prod_{n' \in \mathcal{N}(l) \setminus \{n\}} \tanh\left(\frac{z_{n',l}^{(i-1)}}{2}\right)}.$$

Во время второй фазы i -ого итеративного процесса декодирования, $i = 0,1, \ldots, I-1$, l-ый проверочный узел $l = 0,1, \ldots, L-1$, посылает сообщение $y_{l,n}^{(i)}$ n-ому символьному узлу $n \in \mathcal{N}(l)$. Затем n-ый символьный узел вычисляет статистики

$$z_{n,l}^{(i)} = z_n^{(0)} + \sum_{l' \in \mathcal{L}(n) \setminus \{l\}} y_{l',n}^{(i)} = z_n^{(0)} + \sum_{l' \in \mathcal{L}(n) \setminus \{l\}} \log \frac{1 + \prod_{n' \in \mathcal{N}(l) \setminus \{n\}} \tanh\left(\frac{z_{n',l}^{(i-1)}}{2}\right)}{1 - \prod_{n' \in \mathcal{N}(l) \setminus \{n\}} \tanh\left(\frac{z_{n',l}^{(i-1)}}{2}\right)},$$

которые данный символьный узел будет рассылать как сообщения на следующем шаге итеративного процесса. Этим завершается вторая фаза i-ого шага процесса итеративного декодирования.

Последнее выражение используется для вычисления статистики $z_{n,l}^{(i)}$ на всех шагах итеративного процесса кроме последнего, i -го , шага. На последнем шаге декодер вычисляет для -ого символьного узла только одно логарифмическое отношение правдоподобия

$$z_n^{(I)} = z_n^{(0)} + \sum_{l \in \mathcal{L}(n)} y_{l,n}^{(I)} = z_n^{(0)} + \sum_{l \in \mathcal{L}(n)} \log \frac{1 + \prod_{n' \in \mathcal{N}(l) \setminus \{n\}} \tanh\left(\frac{z_{n',l}^{(I-1)}}{2}\right)}{1 - \prod_{n' \in \mathcal{N}(l) \setminus \{n\}} \tanh\left(\frac{z_{n',l}^{(I-1)}}{2}\right)}.$$

Затем он принимает жесткое решение $\hat{v}_n$ о символе v_n, используя правило

$$\hat{v}_n = \begin{cases} 0, & \text{если } z_n^I > 0, \\ 1, & \text{если } z_n^I < 0. \end{cases}$$

Если $z_n^I = 0$, декодер бросает монету, выбирая $\hat{v}_n = 0$ или $\hat{v}_n = 1$ с вероятностью $1/2$.
Поскольку алгоритм может быть описан как чередование вычислений сумм и произведений, его иногда называют алгоритм сумма-произведение (sun-product algorithm).
Сложность одной итерации «жесткого» декодирования инвертированием бит является линейной, количество итераций декодирования обычно выбирается около $\log_2 N$, где N – длина кодового слова.

2. Алгоритм «мягкого» декодирования. Пусть $h_{i,j}$ элемент i-й строки и -го столбца проверочной матрицы H. Обозначим множество кодовых позиций, участвующих в i-ом проверочном уравнении как $\zeta(m) = \{l: h_{m,l} = 1\}$ и примем $\mu(l) = \{m: h_{m,l} = 1\}$ как множество проверочных позиций, в которых участвует кодовая позиция l.

Алгоритм итеративно вычисляет два типа условных вероятностей:
- $q_{m,l}^x$ – вероятность того, что l-ый бит вектора $\mathbf{v}$ имеет значение x по информации, полученной от всех проверочных вершин (графа Таннера) кроме вершины m;
- $r_{m,l}^x$ – вероятность того, что уравнение, соответствующее проверочной вершине m, удовлетворяется, если значение l-го бита равно x, а остальные биты независимы с вероятностями $q_{m,l'}, l' \in \zeta(m) \setminus l$.

Начальные установки
Для $l \in \{1,2,\ldots,N\}$ установить априорные вероятности кодовых вершин (графа Таннера) равными

$$p_l^1 = \frac{1}{1+\exp\left(r_l \frac{4}{N_0}\right)},$$

где $p_l^0 = 1 - p_l^1$ для каждой пары $(l.m)$ такой, что $h_{m,l} = 1, q_{m,l}^0 = p_l^0,\ q_{m,l}^1 = p_l^1$.
Обработка сообщения
Шаг 1. Снизу вверх (по горизонтали):
Для каждой пары $(l.m)$ вычислить

$$\delta r_{m,l} = \prod_{l' \in \zeta(m)\setminus l} \left(q_{m,l}^0 - q_{m,l'}^1\right)$$

и $r_{m,l}^0 = \left(1+\delta r_{m,l}\right)/2, \quad r_{m,l}^1 = \left(1-\delta r_{m,l}\right)/2.$
Шаг 2. Сверху вниз (по вертикали):

$$q_{m,l}^0 = p_l^0 \prod_{m' \in \mu(l)\setminus l} r_{m',l}^0, \qquad q_{m,l}^1 = p_l^1 \prod_{m' \in \mu(l)\setminus m} r_{m',l}^1,$$

результат нормируется с множителем $\alpha = \frac{1}{\left(q_{m,l}^0 + q_{m,l}^1\right)}$, где $q_{m,l}^0 = \alpha q_{m,l}^0, \quad q_{m,l}^1 = \alpha q_{m,l}^1$.
Для каждого l вычислить апостериорные вероятности

$$q_l^0 = p_l^0 \prod_{m \in \mu(l)} r_{m,l}^0, \qquad q_l^1 = p_l^1 \prod_{m \in \mu(l)} r_{m,l}^1,$$

результат нормируется их множителем $\alpha = \frac{1}{\left(q_l^0 + q_l^1\right)}$, где $q_l^0 = \alpha q_l^0, \quad q_l^1 = \alpha q_l^1$.
Этап декодирования и формирования мягких выходов.
Для $i = 1,2,\ldots,N$ вычислить $v_i = sgn(q_i^o)$. Если $\hat{\mathbf{v}}\mathbf{H} = 0$, то оценкой кодового слова и мягкими выходами являются

$$\Lambda(v_i) = \log\left(q_i^1\right) - \log\left(q_i^0\right), \quad 1 \le i \le N$$

процесс завершается.
Иначе процесс возвращается к шагу 2. Если число итераций превышает заранее установленный порог, то фиксируется отказ от декодирования (ошибка). На выход выдаются принятые значения символов. Процесс завершается.
На рисунке 1 приведены зависимости вероятности ошибки от отношения сигнал/шум в канале для алгоритмов декодирования низкоплотностного кода.

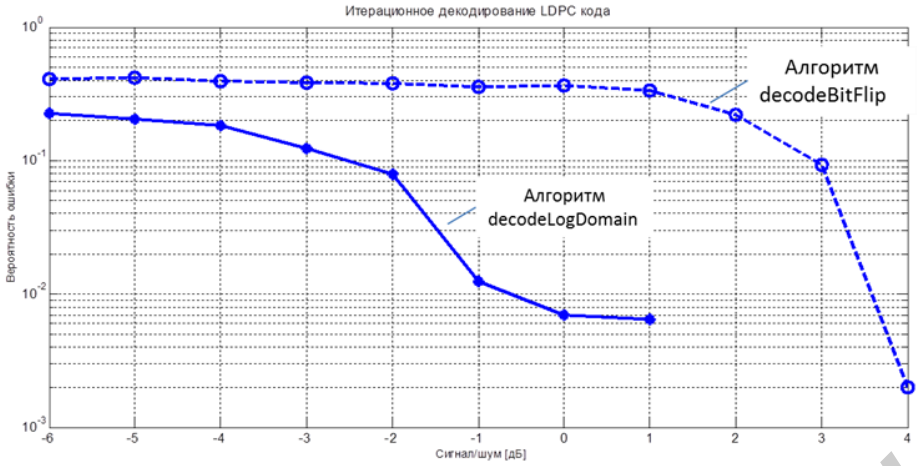

Рис. 1 Зависимости оценки помехоустойчивости низкоплотностного кода

Заключение

Сложность «мягкого» алгоритма выше, чем сложность «жесткого» декодирования инвертированием битов, однако качество декодирования повышается за счет использования дополнительной информации на выходе канала. Однако качество работы такого алгоритма зависит от инициализации: чем точнее она произведена, тем точнее будет конечный результат. Для канала с гауссовским шумом инициализация может быть произведена при помощи информации о дисперсии шума в канале. Для других распределений шума в канале или при неизвестных характеристиках шума точная инициализация алгоритма может являться сложной задачей.

Литература

1. Stojmenovic, I. Handbook of sensor networks: algorithms and architectures/ I. Stojmenovic. – John Wiley & Sons, 2005, vol. 49.
2. MacKay, D. Near Shannon limit performance of low-density parity-check codes/ D. MacKay, R. M. Neal // IEEE Transactions on Information Theory. Vol. 47. Feb. 2001.
3. Richardson, T. J. Efficient encoding of low-density parity- check codes/ T. J. Richardson, R. L. Urbanke // IEEE transactions on information theory. Vol. 47. Feb. 2001.
4. Kou, Y. Low-density parity-check codes based on finite geometries: A rediscovery and new results/ Y. Kou, S. Lin. P. Fossorier // IEEE transactions on information theory. Vol. 47. Nov. 2001.
6. Fossorier, M. P. C.Reduced complexity iterative decoding of low&density parity&check codes based on belief propagation/ M. P. C. Fossorier, M. Mihaljevic, H. Imai // IEEE transactions on communications. Vol. 47. May 1999.
8. Lin, S. Shortened finite geometry codes/ S. Lin // IEEE transactions on information theory. Vol. 18. Sept. 1972. P. 692–696.

Сведения об авторах

Андриянова Т.А., аспирант Белорусского государственного университета информатики и радиоэлектроники. Окончила Белорусский государственный университет информатики и радиоэлектроники по специальности «Радиоэлектронная защита информации» в 2009 году, магистратуру по специальности «Методы и системы защиты информации,

Information about the authors

Andryianava T.A., postgraduate student of Belarusian state university of informatics and radioelectronics. Graduated from Belarusian state university of informatics and radioelectronics «Radioelectronic information security» 2009, Master of Technical Sciences «Methods and systems of information security, information security» 2010.

информационная безопасность» в 2010/

Саломатин С.Б., к.т.н., доцент Белорусского государственного университета информатики и радиоэлектроники.

Salomatin S.B., Ph.D., associate professor of Belarusian state university of informatics and radioelectronics.

Адрес для корреспонденции

220013, Республика Беларусь, Минск, ул. П.Бровки, 6, Белорусский государственный университет информатики и радиоэлектроники.
тел: +375293436560;
e-mail: rezistka@gmail.com
Андриянова Татьяна Александровна
тел: +375296714732;
e-mail: kafsiut@bsuir.by
Саломатин Сергей Борисович

Address for correspondence

220013, Republic of Belarus, Minsk, P. Brovka st., 6, Belarusian state university of informatics and radioelectronics,
tel. +375293436560;
e-mail: rezistka@gmail.com
Andryianava Tatsiana Alexandrovna
tel. +375296714732;
e-mail: kafsiut@bsuir.by
Salomatin Sergei Borisovich