В современном мире, когда шифрование информации используется повсеместно, криптография развивается семимильными шагами. Но несмотря на огромное количество алгоритмов, задача шифрования все еще не решена полностью. Математиками до сих пор не найдено идеального алгоритма шифрования, который был бы устойчив к атакам, от банального перебора всех возможных ключей до более хитрых методов.

Базисные функции являются перспективным направлением в этой области, находя применение в алгоритмах шифрования самой различной информации. Рассмотрим некоторые из них.

Система передачи защищенной речевой информации

Для начала, разберемся, что такое вейвлет-анализ. Это исследование сигнала при помощи базисных функций. Базисы, которые применяются для этого, названы вейвлетами и представляют из себя функции с двумя аргументами – масштаба и сдвига.

$$f_{a,b} = f(\frac{(t-a)}{b})$$

Вейвлет-преобразование представляет сигнал на двумерной плоскости положение-масштаб. Это позволяет выделять крупные и мелкие особенности сигналов, вместе с тем локализуя их на временной оси.

Идея дискретного вейвлет-анализа состоит в том, чтобы представить сигнал как последовательность образов с разной степенью детализации. Это позволяет выявить локальные особенности сигнала. Разложение сигнала осуществляется с помощью двух фильтров и блоков децимации (рис. 1).

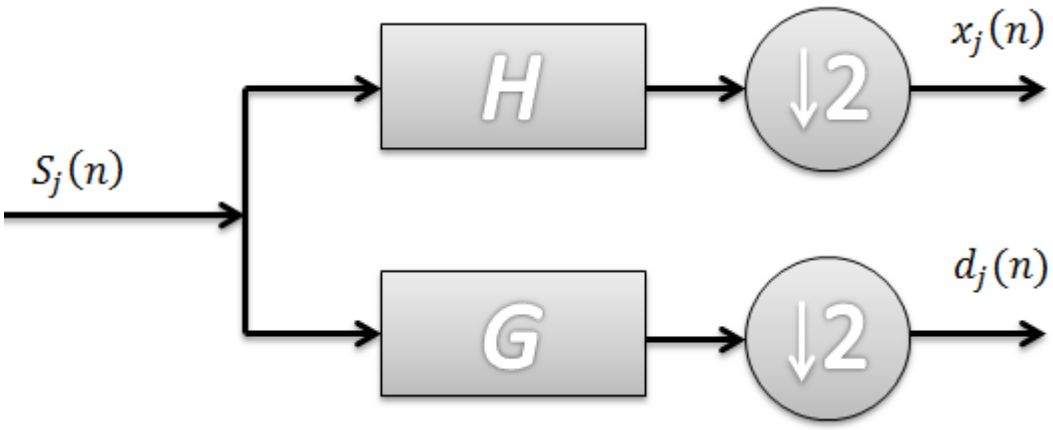

Рис.1. Вейвлет-разложение сигнала.

Исходя из данной схемы, вейвлет-анализ сводится к нахождению коэффициентов $x_j(n)$ и $d_j(n)$ в разложении сигнала $S_j(n)$. Коэффициенты $x_j(n)$ называются аппроксимирующими, коэффициенты $d_j(n)$ – детализирующими.

Для реализации системы передачи речевой информации необходимо, чтобы для АЧХ вейвлет-фильтра $|H(\omega)|$ выполнялись следующие условия:

1. $|H(\omega)|^2 + |H(\pi - \omega)|^2 = 2$
2. $|H(\omega)| \geq 0$
3. $H(\pi) = 0$

Теперь приступим к построению ортогональных вейвлет-фильтров. Пусть имеется некоторый сигнал $S(n)$. Нам нужно, чтобы при разложении на вейвлет-коэффициенты функции $f(n) = IFFT\{S(\omega) * (1 + e^{-j\omega})\}$ все детализирующие коэффициенты должны быть равны нулю. В таком случае вейвлет-преобразования сигнала $S(n)$ будут следующими:

$$\begin{cases} H(\omega)F(\omega) + H(\omega + \pi)F(\omega + \pi) = X(\omega) \\ G(\omega)F(\omega) + G(\omega + \pi)F(\omega + \pi) = D(\omega) \end{cases} \quad (1)$$

$X(\omega)$ и $D(\omega)$ – образы Фурье соответственно аппроксимирующих и детализирующих коэффициентов преобразования, $H(\omega)$ и $G(\omega)$ – частотные характеристики низкочастотного и высокочастотного фильтров. Поскольку существуют нулевые моменты фильтра H, пусть $F(\omega) = S(\omega) * (1 - e^{-j\omega})$. Кроме того, установим соответствие между фильтрами H и G: $G(\omega) = e^{-j\omega} * \overline{H(\omega + \pi)}$. Получаем:

$$G(\omega + \pi) * \overline{G(\omega)} + H(\omega + \pi) * \overline{H(\omega)} = 0, \quad (2)$$

Где $\overline{G(\omega)}$ и $\overline{H(\omega)}$ – частотные характеристики соответствующих фильтров восстановления.

Далее потребуем выполнения свойства 1. Предположим, что $D(\omega) = 0$ и решим систему (1), используя соотношение (2).

$$H(\omega) = \frac{X(\omega) * F(\omega)}{|F(\omega)|^2 + |F(\omega + \pi)|^2}$$

Найдем последнее условие на $X(\omega)$. Учитывая первой свойство и равенство $X(\omega) = X(\omega + \pi)$, имеем

$$|X(\omega)|^2 = 2 * (|F(\omega)|^2 + |F(\omega + \pi)|^2).$$

Пусть $X(\omega)$ – действительная функция. Тогда

$$X(\omega) = \sqrt{2} * \sqrt{|F(\omega)|^2 + |F(\omega + \pi)|^2}.$$

Получили окончательную формулу ЧХ низкочастотных фильтров:

$$H(\omega) = \frac{\sqrt{2} * \overline{F(\omega)}}{\sqrt{|F(\omega)|^2 + |F(\omega + \pi)|^2}}.$$

Особенность данных вейвлет-фильтров в том, что их импульсная характеристика формируется с учетом характеристик обрабатываемого сигнала. В случае построения системы передачи защищенной информации эти характеристики могут быть ключевой последовательностью. Информация о сигнале закладывается в сам фильтр, и разложить сигнал на исходную речь и наложенный шум может только обладатель данных фильтров. Поставив такие фильтры на стороне отправителя и на стороне приемника, получим защищенную систему передачи голосовой информации. Исходный сигнал, проходя через фильтр Hна стороне отправителя, являет собой сумму взвешенных аппроксимирующих функций, где коэффициенты взвешивания – отсчеты сигнала в определенные моменты времени. Маскирующий шум на выходе фильтра G – сумма взвешенных вейвлет-функций, где коэффициенты взвешивания – отсчеты белого гауссовского шума. Посколько аппроксимирующие функции ортогональны вейвлет-функциям, их смесь легко разделить на стороне приемника, если известен ключ, который и определяет форму этих функций.

Универсальный алгоритм шифрования на основе всплесковых базисов

Пусть $(G, \dotplus)$ – нуль-мерная локально компактная Абелева группа с основной цепочкой подгрупп

$$\dots \subset G_n \subset \cdots \subset G_1 \subset G_0 \subset G_{-1} \subset \cdots \subset G_{-n} \subset \cdots, \left(\frac{G_n}{G_{n+1}}\right)^{\#} = p$$

p – простое, а $g_n \in \frac{G_n}{G_{n+1}}$ – базисная последовательность. Любой элемент $x \in G$ можно единственным образом представить в виде суммы ряда

$$x = \sum_{n=-\infty}^{+\infty} x_n g_n \ , \quad x_n = \overline{0, p-1}.$$

Определим отображение $\psi: G \to \mathbb{R}^+$, которое переводит подгруппу G_n на отрезок $x_n = [0, \frac{1}{p^n}]$. Теперь мы можем рассмотреть группу G на полупрямой, топология в которой определяется сдвигами отрезков x_n. Введем $G_n^{\perp}$ - аннуляторы подгруппы G_n. Назовем функциями Радемахера характеры $r_n \in \frac{G_n^{\perp}}{G_{n+1}^{\perp}}$, а значение характера χ на элементе x обозначим как (χ, x).

Пусть $r_{-1} \in \frac{G_0^{\perp}}{G_{-1}^{\perp}}$ - функция Радемахера. $r_{-1} \in G_0^{\perp} \Rightarrow r_{-1}(G_0) = 1$. Т.к. p простое, то (r_{-1}, g_{-1}) – корень из единицы степени p. Иными словами, $(r_{-1}, g_{-1}) = e^{\frac{2\pi i}{p}\nu}$. Элемент g_{-1} возможно подобрать так, что $(r_{-1}, g_{-1}) = e^{\frac{2\pi i}{p}}$, так что $r_{-1}(G_0 \dotplus j g_{-1}) = e^{\frac{2\pi i}{p}j}$, $j = \overline{0, p-1}$.

Определим масштабирующую функцию $\varphi(x)$:

$$\varphi(x) = \frac{1}{p} 1_{G_{-1}}(x) \sum_{j=0}^{p-1} \mu_j (r_{-1}, x)^j.$$

Элементы μ_j – комплексные числа. Наложим на них два ограничения: $\mu_0 = 1$ и $|\mu_j| = 1$. Функция $\varphi(x)$ обладает следующими свойствами:

1. $\varphi(x) = 0$ вне G_{-1}.
2. $x \in G_0 \dot{+} \lambda_{g_{-1}} => \varphi(x) = \varphi(\lambda_{g_{-1}}), \varphi(x \dot{-} \nu g_{-1}) = \varphi(\lambda_{g_{-1}} \dot{-} \nu g_{-1})$. Иными словами, сдвиг $\varphi(x \dot{-} \upsilon g_{-1})$ – функция, принимающая на смежных классах $G_0 \dot{+} j g_{-1}$ значения $\varphi_{\lambda,\nu} = \varphi(\lambda g_{-1} \dot{-} \nu g_{-1})$.
3. Смежные классы $G_0 \dot{+} j g_{-1}$ можно рассматривать как отрезки $[j, j+1]$.
4. $\varphi(x)$постоянна на смежных классах $G_0 \dot{+} j g_{-1}$.

Можно также проверить, что любая функция, принимающая постоянные значения на смежных классах $G_0 \dot{+} \lambda g_{-1} = [\lambda, \lambda + 1]$ – линейная комбинация сдвигов $\varphi(x \dot{-} \nu g_{-1})$[2].

Шифруем сообщение по следующему алгоритму:

1. Выбираем такую последовательность $(\mu_j)_{j=0}^{p-1}$, что $\mu_0 = 1$ и $|\mu_j| = 1$ при $j = \overline{1, p-1}$.
2. Строим функцию $\varphi(x)$ по формуле, указанной выше.
3. Образуем матрицу $\Phi = (\varphi(\lambda g_{-1} \dot{-} \nu g_{-1}))_{\lambda,\nu}$.
4. Для сообщения $X = (x_0, \dots, x_{p-1})^T$ находим коэффициенты $C = (c_0, \dots, c_{p-1})^T$ и передаем их вместо исходного текста.

Метод близок к классическим (например, к методу Хилла), но плюс данного метода в том, что ключей, которыми в нашем случае являются последовательности (μ_j), может быть бесконечно много. При шифровании сообщения длиной больше p следует разделить его на блоки длиной p и шифровать каждый из них отдельно.

Применение решеток в криптографии и криптоанализе

Для начала выясним, что такое решетки. Возьмем $A = \{a_1, \dots, a_n\}$ – набор линейно-независимых векторов в $\mathbb{R}^m$. Решеткой размерности n будет называться набор линейных комбинаций a_i с целочисленными коэффициентами:

$$L = \mathbb{Z}a_1 + \mathbb{Z}a_2 + \cdots + \mathbb{Z}a_n.$$

Один из простейших примеров: возьмем листок в клетку, примем за единицу длину ребра клетки и возьмем $A = \{(1,0), (0,1)\}$. Тогда решеткой L будет множество всех векторов с целочисленными координатами. Набор A является базисом решетки. Кроме того, другой набор $A = \{(5,1), (6,1)\}$ тоже будет являться ее базисом.

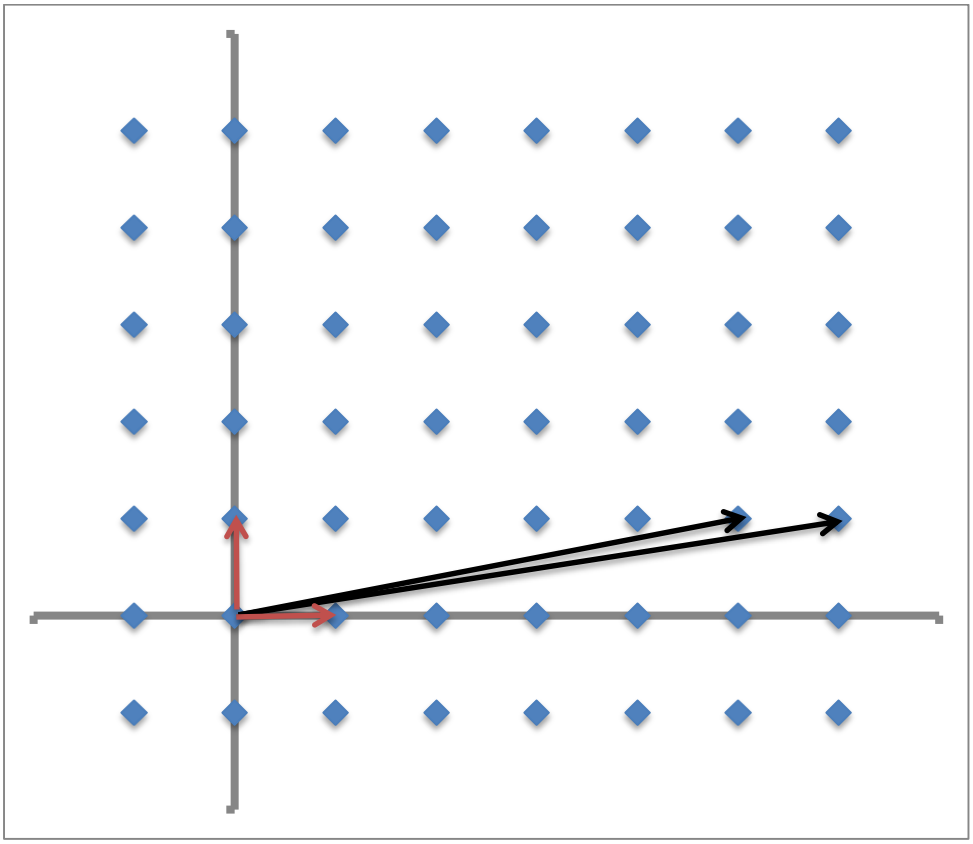

Рис. 2. Пример решетки ***L*** и двух базисов на ней.

Длины векторов второго базиса значительно больше длин векторов первого. Чтобы найти применение решеткам в криптографии, нам необходимо решить задачу о поиске кратчайшего вектора.

Возьмем к примеру базис $A = \{(1,0),(1,1)\}$. Прежде всего нам нужно ортогонализировать его. Это можно сделать по методу Грама-Шмидта. Для начала, определим

$$\mu_{ij} = \frac{\langle a_i, a_j^* \rangle}{\langle a_j^*, a_j^* \rangle}, 1 \leq j < i \leq m.$$

После этого последовательно находим все вектора нового ортогонального базиса:

$$a_i^* = a_i - \sum_{j=1}^{i-1} \mu_{ij} a_j^*, 1 \leq i \leq n.$$

Тогда $\langle a_i^*, a_i \rangle = \langle a_i^*, a_i^* \rangle$ и $\forall j < i$: $\langle a_i^*, a_j \rangle = 0$.

Матрица перехода от базиса A^* обратно к базису A будет выглядеть так:

$$\begin{pmatrix} ||a_1^*|| & \mu_{21}||a_1^*|| & \cdots & \mu_{(m-1)1}||a_1^*|| & \mu_{m1}||a_1^*|| \\ 0 & ||a_2^*|| & & \mu_{(m-1)1}||a_2^*|| & \mu_{m1}||a_2^*|| \\ & \vdots & \ddots & \vdots & \\ & 0 \quad 0 & & & \\ & 0 \quad 0 & \cdots & ||a_m^*|| & \end{pmatrix}$$

За нижнюю оценку на размер кратчайшего вектора решетки примем $\lambda_1(L) \geq min_j \|a_j^*\|$.

Базис называется $\delta - LLL$ – редуцированным, если

$$\forall\, 1 \leq j < i \leq m\colon |\mu_{ij}| \leq \frac{1}{2}\ ,$$

$$\forall 1 \leq i < m\colon \delta\|a_i^*\|^2 \leq \ \left\|\mu_{i+1,i}a_i^* + a_{i+1}^*\right\|^2$$

Если записать базис A в ортонормированном базисе, получаемом из A^*, то, учитывая условие $|\mu_{ij}| \leq \frac{1}{2}$, получим:

$$\begin{pmatrix} ||a_1^*|| & \leq \frac{1}{2}||a_1^*|| & \cdots & \leq \frac{1}{2}||a_1^*|| & \leq \frac{1}{2}||a_1^*|| \\ 0 & ||a_2^*|| & & \leq \frac{1}{2}||a_2^*|| & \leq \frac{1}{2}||a_2^*|| \\ & \vdots & \ddots & \vdots & \\ & 0 \quad 0 & & & \\ & 0 \quad 0 & \cdots & \leq \frac{1}{2}||a_{m-1}^*|| & \leq \frac{1}{2}||a_{m-1}^*|| \\ & & & 0 & ||a_m^*|| \end{pmatrix}$$

И еще одно условие: в подматрице

$$\begin{pmatrix} ||a_i^*|| & \mu_{i+1,i}||a_i^*|| \\ 0 & ||a_{i+1}^*|| \end{pmatrix}$$

длина первого и второго столбцов практически равны, т.е.,

$$\delta||a_i^*||^2 \leq \mu_{i+1}^2\left||a_i^*|\right|^2 + ||a_{i+1}^*||^2.$$

LLL – редуцированный базис можно вычислить по алгоритму L^3 (от первых букв фамилий создателей: Lenstra, Lenstra, Lovasz):

1. Вычислить $a_1^*, \dots, a_n^*$
2. Для каждого $2 \leq i \leq m$ и для каждого $i - 1 \geq j \geq 1$:

2.1. $\quad a_i = a_i - c_{ij}b_j; \quad c_{ij} = \left\lfloor \frac{\langle a_i, a_j^* \rangle}{\langle a_j^*, a_j^* \rangle} \right\rceil$

3. Обмен значениями:

3.1. Если существует такое i, для которого $\delta||a_i^*||^2 > \left\|\mu_{i+1,i}a_i^* + a_{i+1}^*\right\|^2$, то меняем их местами и возвращаемся к первому шагу.

4. Выдаем результат: $a_1, \dots, a_m$

Если $\delta - LLL$- редуцированный базис найден, то задача поиска кратчайшего вектора будет решена с точностью $(\delta - 0.25)^{-\frac{n-1}{2}}$.

Одно из применений алгоритма L^3 – поиск корней многочленов. Рассмотрим многочлен степени m в $f \in \mathbb{Z}_N[x], N \in \mathbb{Z}$. Не зная разложения N на множители, мы тем не менее можем найти все корни f в $\mathbb{Z}_N$, такие что $|x| \leq N^{\frac{1}{m}}$. Число $N^{\frac{1}{m}}$ для удобства обозначим константой B.

Рассмотрим многочлен $f(x) = x^m + a_{m-1}x^{m-1} + \cdots + a_1x + a_0$. Если для всех его коэффициентов выполняется равенство

$$\left|a_iB^i\right| < \frac{N}{m+1}, 0 \leq i \leq m-1,$$

то тогда для $|x| \leq B$

$$|f(x)| \leq \sum_{i=0}^{d}\left|a_iB^i\right| < N.$$

Такие корни несложно найти. В ином же случае мы можем подобрать такой многочлен g, что он будет иметь маленькие коэффициенты, но те же корни, что и у f.

Рассмотрим множество многочленов

$$F = \{N, Nx, Nx^2, \dots, Nx^{m-1}, f(x)\}.$$

Поскольку мы ищем корни по модулю N, все корни f являются также корнями любой их линейной комбинации. Теперь рассмотрим решетку из столбцов матрицы

$$M = \begin{pmatrix} N & 0 & 0 & & 0 & a_0 \\ 0 & BN & 0 & \cdots & 0 & Ba_1 \\ 0 & 0 & B^2N & & 0 & B^2a_2 \\ & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & & B^{m-1}N & B^{m-1}a_{m-1} \\ 0 & 0 & 0 & \cdots & 0 & B^m \end{pmatrix}.$$

Применив алгоритм L^3, мы найдем такой v, что $\left||v|\right| \leq O\left(\lambda_1(M)\right) \leq O((\det M)^{\frac{1}{m+1}})$.

Определитель матрицы M равен $N * BN * \ldots * B^{m-1}N * B^m = N^m B^{\frac{m(m+1)}{2}}$. Подставив значение определителя в предыдущую формулу, получим

$$\left|\left|v\right|\right| \leq O\left(\lambda_1(M)\right) \leq O\left(N\frac{B^{\frac{m}{2}}}{N^{\frac{1}{m+1}}}\right).$$

Константы под O зависят только от размера решетки. Для $B < c_1(m)N^{\frac{2}{m(m+1)}}$ это уже меньше, чем $\frac{N}{m+1}$.

Теперь о практическом применении. Вспомним алгоритм RSA. Если Алиса отправит одинаковые m трем Бобам с различными N_i и экспонентой e^i, то получится система уравнений $m^3 \equiv c_i (mod\ N_i)$. По китайской теореме об остатках, $m^3 \equiv c (mod\ N_1 N_2 N_3)$. А т.к. $m < N_i$, то последний корень можно извлечь.

Решетки применяются не только при анализе криптографических алгоритмов, но и при их построении. Например, хеш-функции Ajtai, идея конструкции которых представлена ниже.

Рассмотрим m векторов $a_1, a_2, \ldots, a_m \in \mathbb{Z}_q^n$. Определим функцию $f_{a_1,\ldots,a_m}(b_1, \ldots, b_m) = \sum_{i-1}^{m} b_i a_i \ \ (mod\ q)$. Нам нужно построить семейство хэш-функций, в качестве которого мы возьмем набор $F = \left\{f_{a_1,\ldots,a_m} \middle| a_1, \ldots, a_m \in \mathbb{Z}_q^n\right\}$. Чтобы семейство было стойким, необходимо, чтобы для случайно взятой функции f из F никакой полиномиальный алгоритм не мог бы с большой вероятностью найти такие $x \neq y$, что $f(x) = f(y)$. Нам нужно, чтобы не существовало такого алгоритма, который по случайным $a_{1,\ldots,}a_m \in \mathbb{Z}_q^n$ мог бы находить с большой вероятностью их линейную комбинацию $b_1, \ldots, b_m \in \{0, \pm 1\}$, такую что $\sum_{i=1}^{m} b_i a_i \equiv 0 \quad (mod\ q)$.

Рассмотрим решетку как распределение вероятностей и добавим в нее нормально распределенный шум. При добавлении достаточного количества шума решетку будет очень сложно отличить от равномерного распределения. Допустим, мы нашли такой параметр P, с которым распределение решетки достаточно близко к равномерному. Возьмем несколько векторов по этому нормальному распределению и приведем их к параллелепипеду решетки. Распределенные в параллелепипеде точки будут распределены практически равномерно. Теперь разобьем все решетку на q^n ячеек, в соответствие каждой из них поставим свой элемент из $\mathbb{Z}_n^q$. Наши исходные векторы теперь соответствуют числам из $\mathbb{Z}_n^q$.

Криптосистема Ajtai-Dwork основана на этом алгоритме. Сама конструкция криптосистемы не очень сложна. Возьмем достаточно большое N. Секретный ключ – просто число $h \in [\sqrt{N}, 2\sqrt{N})$. Публичным ключом будет являться набор из $m = O(\log N)$ чисел $0 \leq a_1, \ldots, a_m \leq N - 1$, которые достаточно близки к числам, кратным $\frac{N}{h}$. N при этом не обязательно нацело делится на h. К тому же, одно из чисел a_{i_0} должно

быть достаточно близко к нечетному кратному $\frac{N}{h}$. Кодом нуля в данном случае является сумма случайного подмножества $\{a_i\}$, кодом единицы – сумма подмножества $\{a_i\}$ плюс $\lfloor a_{i_0}/2 \rfloor$. Алгоритм декодирования: делим число на $\frac{N}{h}$, и если получилось мало, выдаем 0, если много – 1. Поскольку все a_i достаточно близки к тому, чтобы делиться на $\frac{N}{h}$, а $\frac{a_{i_0}}{2}$ не близко, декодирование корректно.

Как мы видим, алгоритмы шифрования информации на основе базисных функций имеют ряд любопытных свойств, которые выгодно отличают их от иных алгоритмов. Помимо криптографии, базисные функции используются также при сжатии изображений, видео и во многих других областях, где тоже демонстрируют отличные результаты, но это уже лежит за пределами тематики данной статьи.

Список использованных источников и литературы

1. *Савватин А.И., Новоселов С.А.* Метод построения цифровой системы защищенной передачи цифровой информации // Электронный журнал ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ. Теория и практика URL: http://network-journal.mpei.ac.ru/cgi-bin/main.pl?l=ru&n=17&pa=11&ar=3

2. *Лукомский Д.С., Лукомский С.Ф.* Всплесковые базисы и криптография // Сб. науч. тр. Механика. Математика, Саратов: Изд-во Сарат. ун-та, 2011. С. 55-58

3. *Danielle Micciancio, Oded Regev* Lattice-based Cryptography // Post-Quantum Cryptography, 2009, pp 147-191

4. *Шокуров А.В., Кузюрин Н.Н, Фомин С.А* Решетки, алгоритмы и современная криптография URL: http://discopal.ispras.ru/images/2/23/Решетки,_алгоритмы_и_современная_криптография.pdf