

Согласованные СГ-НЗБ

Процедура вложения

$$C_w(n) = \begin{cases} C(n) & \text{если НЗБ } (C(n)) = b(n) \\ C(n) & \text{с вероятностью } \frac{1}{2} \\ C(n) & \text{с вероятностью } \frac{1}{2} \end{cases} \text{если НЗБ } (C(n)) \neq b(n)$$

Процедура извлечения

$$\begin{aligned} \text{Если } 0, () \text{ четное число, т.е. НЗБ } () (\begin{matrix} 0 \\ w \end{matrix}) = \\ \text{Если } 1, () \text{ нечетное число, т.е. НЗБ } () (\begin{matrix} 1 \\ w \end{matrix}) = \end{aligned}$$

Основная идея использования ± 1 НЗБ вместо обычного НЗБ-замещения состоит в том, что обычное НЗБ обладает некоторой несимметрией. Это в свою очередь, приводит к появлению характерных статистических признаков, позволяющих сделать процедуру обнаружения более надежной.

Для СГ НЗБ ± 1 разработаны достаточно сложные методы стегоанализа, отличные от рассмотренных ранее для обычных СГ-НЗБ:

1. Гистограммный метод;
2. Метод калибровки изображений;
3. Метод использования статистики второго порядка.

Гистограммный метод стегоанализа СГ-±1 НЗБ

Гистограмма изображения называется:

$$h_c(n) = \#(i, j); \{p_c(i, j) = n\},$$

где $p_c(i, j)$ – яркость пикселя с координатами (i, j)

В качестве меры распределения энергии используется **центр масс COG (H)**

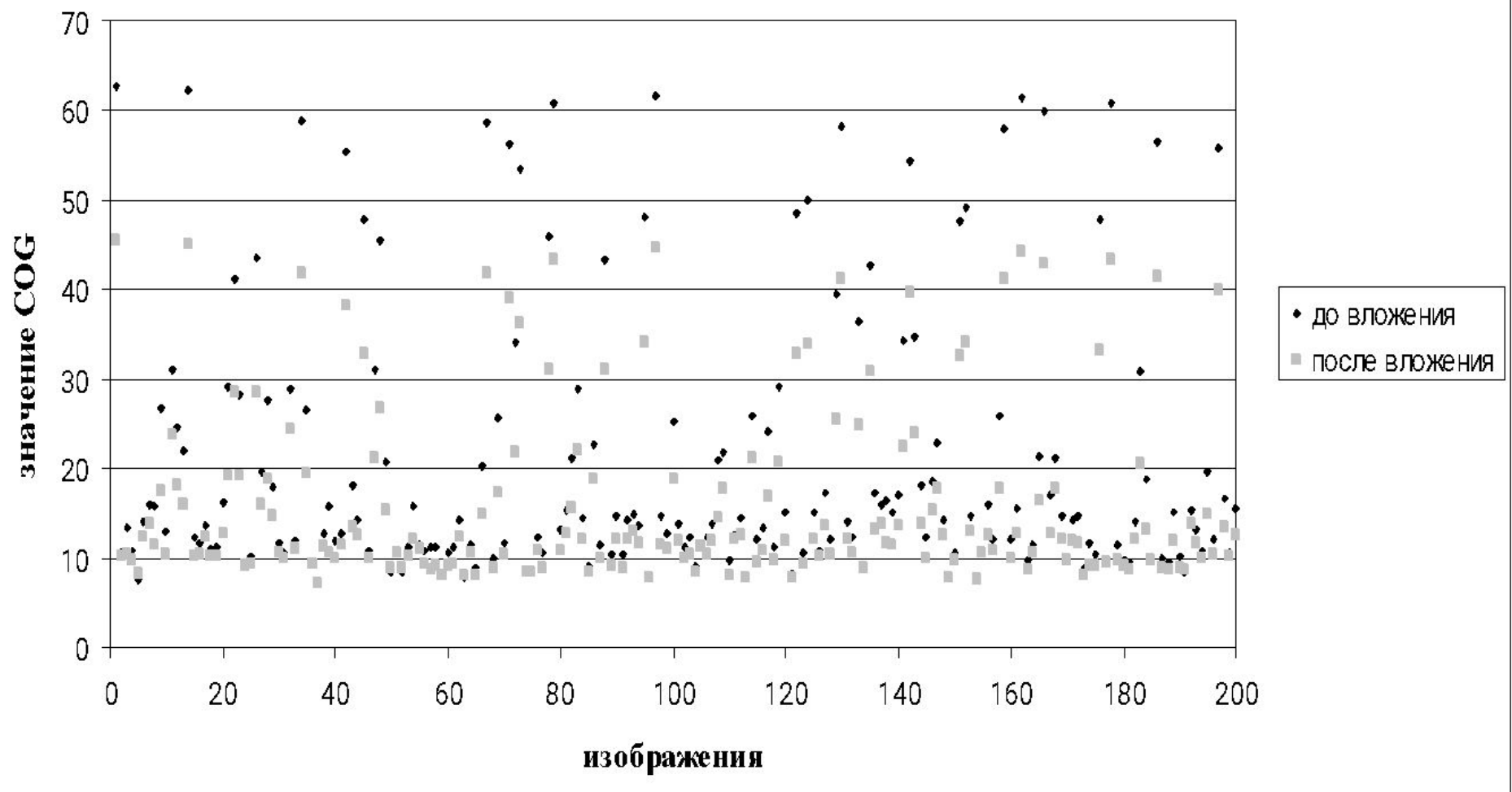
$$COG(H_c) = \frac{\sum_{k=0}^{N/2-1} k |H_c[k]|}{\sum_{k=0}^{N/2-1} |H_c[k]|}, \quad COG(H_s) = \frac{\sum_{k=0}^{N/2-1} k |H_s[k]|}{\sum_{k=0}^{N/2-1} |H_s[k]|},$$

где $H_s[k], H_c[k]$ - N - элементное дискретное преобразование Фурье от h_s, h_c

k – номер пикселя, N – количество пикселей в изображении.

Критерий гистограммного метода стегоанализа для СГ-±1НЗБ базируется на следующем неравенстве:

$$COG(H_s[k]) \leq COG(H_c[k])$$



Значения COG (H[k]) до (ромбы) и после (квадраты) вложения для 200 изображений с градациями серого

Замечания:

- эффективность стегоанализа на основе COG зависит от источника ПО.
- невозможно определить универсальный порог
- нельзя полностью автоматизировать,

Гистограммный метод стегаанализа СГ-±1 НЗБ с использованием калибровки изображения

Калибровка:

$$p'_c(i, j) = \left| \frac{\sum_{u=0}^1 \sum_{v=0}^1 p_c(2i+u, 2j+v)}{4} \right|,$$

где $p'_c(i, j)$ – яркость пикселя откалиброванного изображения с координатами (i, j)

Калибровка уменьшает зависимость СОГ от содержания ПО. Поэтому центр масс исходного ПО примерно равен центру масс откалиброванного ПО

$$COG(H'_c[k]) \approx COG(H_c[k])$$

Несмотря на то что вложение вносит шум в откалиброванные изображения и, следовательно, уменьшает $COG(H'_c[k])$, предполагается, что уменьшение происходит в значительно меньшей степени, чем в случае с изначальным ПО, т. е.:

$$COG(H_c[k]) - COG(H_s[k]) > COG(H'_c[k]) - COG(H'_s[k]).$$

Решающее правило

$$\delta = \frac{COG(H(k))}{COG(H'(k))},$$

где $COG(H(k))$ – центр масс исходного (неоткалиброванного) изображения;
 $COG(H'(k))$ – центр масс того же, но откалиброванного изображения

Вероятности пропуска СГ и ложного обнаружения при выборе различных величин порогов, полученные на статистике 10000 изображений

Пороговое значение	Вероятность ложной тревоги	Вероятность пропуска
1,2	0,9856	0,0019
1,1	0,9712	0,0055
1	0,8314	0,0206
0,9	0,5411	0,085
0,85	0,4075	0,3429
0,8	0,2675	0,2528
0,747	0,1439	0,5031
0,7	0,0661	0,7662
0,6	0,0037	0,9898

Оптимальный порог примерно 0,85:

$$P_e = \frac{P_m + P_{fa}}{2} = 0.6571$$

Матричные методы вложения

Определение. Покрывающая функция $\text{cov}(p, n, k)$ – это отображение f множества F_2^k (т.е. всех двоичных цепочек длиной k) в множество F_2^n , $n > k$ такая, что для любого $x \in F_2^n$ и $y \in F_2^k$ найдется такое $x' \in F_2^n$, что $y = f(x')$ и $d_H(x, x') \leq \rho$, где $d_H(x, x')$ – расстояние Хэмминга между цепочками x и x' .

Пусть:

k/n – относительная скорость (максимизируется), ρ/n – относительный радиус покрытия (минимизируется).

Если такая функция существует, то можно погрузить k бит в n пикселей изображения, произведя не более ρ изменений НЗБ в ПО.

Частный случай.

$\text{Cov}(1, 2^k - 1, k)$ строится на основе кодов Хэмминга.

Алгоритм:

. Вычисляется вектор $z = xH^T \oplus y$, где H – проверочная матрица кода Хэмминга. x – вектор ПО, y – вектор вкладываемого сообщения, \oplus – сложение по мод. 2

. Найти столбец l матрицы H , совпадающий с z/l

. Изменить координату x_l в векторе $x = (x_1, x_2, \dots, x_l, \dots, x_k)$ на противоположную для получения x' , то есть вектора стеговложения.

Для извлечения сообщения, получатель следует по тому же пути, что и отправитель и читает ρ бит сообщения, вычисляя произведение матрицы H на значение наименее значащих бит блоков, длины которых равны $2^{\rho} - 1$

Пример, для кода Хэмминга (7,4), имеющего проверочную матрицу

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

Предположим, что нам необходимо произвести вложение сообщения с относительной скоростью $v = 3/7$. Это значит, что 3 бита нашего секретного сообщения будут вложены в $2^3 - 1 = 7$ пикселей.

Пусть блок значений яркости пикселей исходного сообщения

$$g = (11, 10, 15, 17, 13, 21, 19),$$

секретное сообщение $m = (0, 0, 1)$.

Отправитель сначала должен преобразовать значения яркости пикселей в последовательность НЗБ:

$$x = g \bmod 2 = (1, 0, 1, 1, 1, 1, 1)$$

и вычислить синдром

$$Hx = (0, 1, 0).$$

Вычислим также $Hx - m = (0, 1, 1)$

Таким образом, для вложения сообщения, отправителю необходимо изменить значение НЗБ третьего пикселя, то есть изменить $g[3] = 15$ на $g[3] = 14$ или $g[3] = 16$.

$$g = (11\ 10\ 15\ 17\ 13\ 21\ 19)$$

$$x = (1\ 0\ 1\ 1\ 1\ 1\ 1)$$

$$\begin{pmatrix} 0001111 \\ 0110011 \\ 1010101 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$H \qquad x \qquad Hx \qquad m$

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

$Hx \qquad m \qquad Hx - m$

Извлечение:

$$\begin{pmatrix} 0001111 \\ 0110011 \\ 1010101 \end{pmatrix} \times \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

H y m

Требования для использования матричного метода:

Вложение p бит информации производится в кодовый блок длиной $2^p - 1$.

(Например, для вложения $p = 3$ бит информации нам потребуется блок длиной $2^p - 1 = 7$ пикселей.)

Преимущество:

Использование кодов Хэмминга значительно уменьшает количество изменений, которое требуется для вложения секретного сообщения и, таким образом, увеличивает эффективность вложения.

Пример использования различных кодов Хэмминга для вложения секретных сообщений в неподвижные сообщения размером 256×256 пикселей

Общее количество информационных бит	Общее количество изменений покрывающего сообщения	Количество бит в одном блоке (p)	Длина блока ($2^p - 1$)
65536	65536	1	1
43690	21845	2	3
28086	9362	3	7
17476	4369	4	15
10570	2114	5	31
6240	1040	6	63
3612	516	7	127
2056	257	8	253
1152	128	9	511
640	64	10	1023
352	32	11	2047
192	16	12	4095
104	8	13	8191
56	4	14	16383
30	2	15	32767
16	1	16	65535

Пусть изображение $256 \times 256 = 65536$ пкс, вложение с вероятностью 0,01.

Тогда усов.СГ-НЗБ, $m \approx 655$ бит, не более 655 изменений НЗБ ПО.

матричный метод вложения на кодах Хэмминга с параметром $p=9$, то $m \approx 1152$ бита, изменений не более чем 128 НЗБ ПО

Реальные СГС-НЗБ.

1. Реальные стегосистемы с вложением в НЗБ.

(программы для этих СГС распространяются свободно по Интернету)

1.1 Jsteg. В качестве ПО используется цветное изображение в формате JPEG.

Вложение производится в НЗБ частотных коэффициентов (за исключением нулевых и единичных), выбираемых по псевдослучайному пути, который задается стегоключом (паролем).

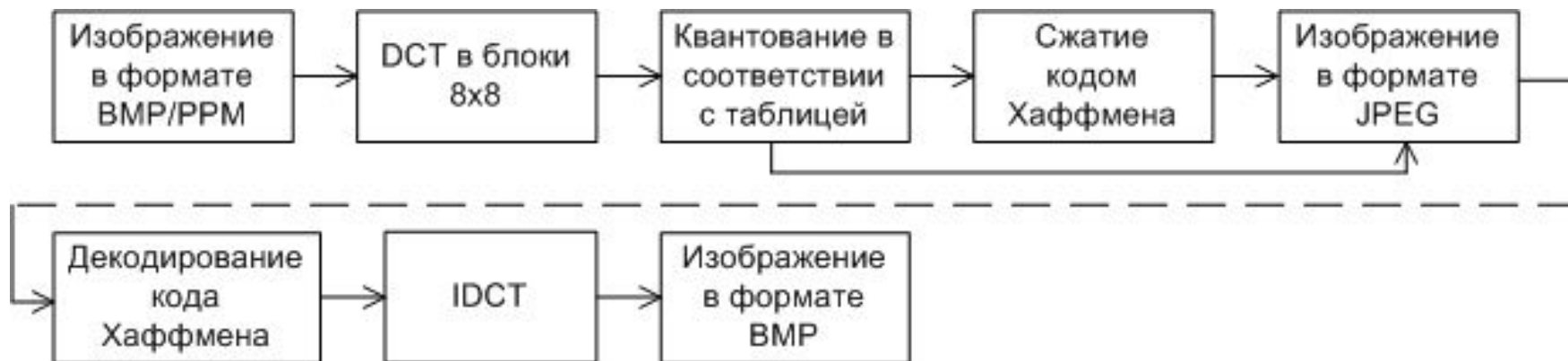


Рис. 1. Алгоритм сжатия изображения в формате JPEG.

Jsteg не обнаруживается визуальной атакой, но легко обнаруживается с использованием статистики χ^2 и анализа пар выборок.

Алгоритм Jsteg

Длина 5 бит- выражает длину поля В (в битах)	0-31 бит. Выражает длину скрываемого сообщения (в байтах)	Скрываемая информация
--	--	-----------------------

A	B...BBBB	C...CCCCCC
---	----------	------------

Исходное изображение
16778 байт



Изображение со скрытым
текстом 17344 байт



Исходное изображение	Изображение со скрытым текстом	
Квантованные коэффициенты DCT	Квантованные коэффициенты DCT	Младшие биты квантованных коэффициентов DCT
D6 69 13 05 03 15 F2 EB FF 04 01 00 FA FB F9 FF 06 02 FE FF 00 00 00 FF 01 03 02 01 01 FF 00 00 01 00 00 00 00 00 00 00 00 FF FF 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00	D6 69 12 05 03 15 F3 EA FE 04 01 00 FA FB F8 FE 06 03 FE FF 00 00 00 FE 01 02 03 01 01 FE 00 00 01 00 00 00 00 00 00 00 00 FE FF 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00	00 01 00 01 01 01 01 00 00 00 01 00 00 01 00 00 00 01 00 01 00 00 00 00 01 00 01 01 01 00 00 01 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00

Поле A: 00 01 00 01 01 или 01011 , т.е. 11 бит в поле B

Поле B: 11 бит для декодирования 01 01 00 00 00 00 01 00 00 00 01 или 11000010001, т.е. 1553 – размер скрытого сообщения...

Поле C: 1553*8 бит для декодирования, начиная с: 00 01 00 00 01 00 00 01 или 01001001 в двоичном или 73 в десятичной системе, что в соответствии с таблицей ASCII является буквой «I» в верхнем регистре, и соответствует первой букве скрытого стихотворения

1.2 Outguess. В качестве ПО используется цветное изображение в формате JPEG. Алгоритм реализован под операционную систему FreeBSD на языке C++. В лабораторной работе N1 портирован с помощью эмулятора cygwin. Работает из командной строки. Требуется задания паролей (стегоключей) вложения и извлечения. Алгоритм вложения разработан для обеспечения защиты от атаки обнаружения χ^2 .

Вложение происходит в два прохода: первый – по псевдослучайному пути, определяемому стегоключом (паролем), как в Jsteg, а второй – с изменением коэффициентов не затронутых первым проходом, с целью приближения гистограммы СГ-изображения к гистограмме ПО, что затрудняет χ^2 -атаку.

Однако, обнаружение Outguess сказывается возможным, для чего используется факт увеличения “неоднородности” в блоках 8x8, которые сравниваются с неоднородностью исходного ПО, полученного при помощи оценки стегоизображения (см. далее: обнаружение СГС-ШПС и “слепой” стегоанализ).

1.3 F5. В качестве ПО используется цветное изображение в формате JPEG. Однако, в отличие от Jsteg и Outguess, это не чистая СГС-НЗБ. Основной принцип F5: при заданном числе вкладываемых бит информации минимизировать количество изменяемых бит ПО.

Пример. $x_1, x_2 \in \{0,1\}$ - биты вкладываемой информации. Обычное НЗБ требует изменение 2^x бит ПС. Модифицированное вложение (где a_1, a_2, a_3 – биты ПС, которые можно изменять):

$$x_1 = a_1 \quad a_3 \oplus x_2 = a_2 \quad a_3 = 0 \Rightarrow \text{ничего не изменять,}$$

$$x_1 \neq a_1 \quad a_3 \oplus x_2 = a_2 \quad a_3 = 1 \Rightarrow \text{изменить } a_1,$$

$$x_1 = a_1 \quad a_3 \oplus x_2 \neq a_2 \quad a_3 = 0 \Rightarrow \text{изменить } a_2,$$

$$x_1 \neq a_1 \quad a_3 \oplus x_2 \neq a_2 \quad a_3 = 1 \Rightarrow \text{изменить } a_3.$$

Во всех случаях изменяется не более одного бита. По заданным a_1, a_2, a_3 однозначно восстанавливаются x_1, x_2 .

Правило извлечения:

$$x_1 = a_1 \oplus a_3, \quad x_2 = a_2 \oplus a_3$$

Алгоритм F5 реализован с помощью JavaScript и использует обобщение данного подхода (матричный) $(1, n, k)$ -код, где n – число позиций, которые могут меняться, k – число вкладываемых бит, 1 – максимальное число изменений при вложении k бит.

Параметры F5: $n = 2^k - 1$, - длина блоков, плотность изменений - $1/2^k$, скорость погружения – $k/n = k/2^k - 1$.

(Допустимые для вложения биты определяются ПСП, задаваемой стегоключом (паролем)). Уменьшение плотности изменений позволяет уменьшить вероятность обнаружения.

Однако СГС-F5 может быть обнаружена при помощи сравнения гистограммы выбранных DCT коэффициентов СГС и гистограммы таких же коэффициентов для оценки исходного ПС:

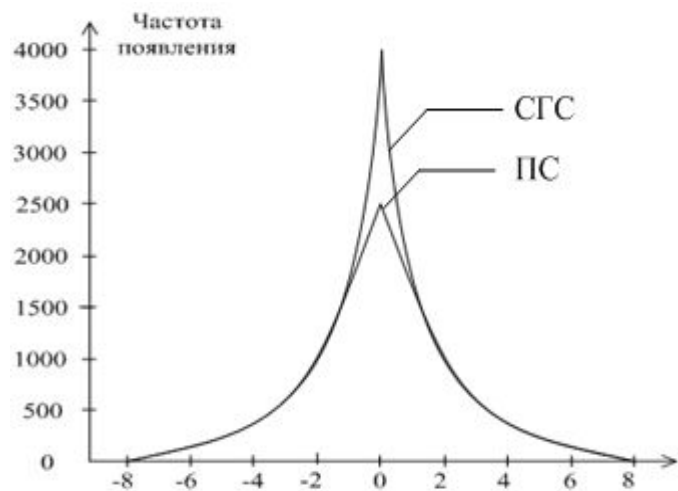


Рис. 2. Гистограмма DCT-(2,1) коэффициентов F5 и оценки действительного ПО. Видно, что СГС-F5 может быть обнаружена.

2. СГС-ШПС.

Все СГС-НЗБ не выдерживают атаки по удалению вложенных сообщений даже при сохранении при этом высокого качества ПО.

Эта атака реализуется при помощи рандомизации НЗБ во временной или частотной области.

Для защиты от такой атаки необходимо использовать широкополосные сигналы (ШПС-СГ):

$$C_W(n) = C(n) + \alpha(-1)^b \pi(n), n = 1, 2 \dots N, \quad (1)$$

где α – коэффициент вложения, $\pi(n)$ – псевдослучайная (± 1) последовательность (ПСП), вырабатываемая по секретному стегоключу, N – длина ПСП, на которой вкладывается один и тот же бит ($b=1$ или 0) информации.

Выделение информации при неизвестном ПО (“слепой” декодер):

$$\sum_{n=1}^N (C'_W(n) - m_c) \pi(n) \begin{cases} > 0 \rightarrow b = 0 \\ < 0 \rightarrow b = 1 \end{cases}, \quad (2)$$

где атака производится аддитивным шумом:

$$C'_W(n) = C_W(n) + \varepsilon(n), n = 1, 2 \dots N, \quad (3)$$

и где $m_c = E\{C(n)\}$.

Поскольку $\pi(n)$ при атаке неизвестна, то при выборе достаточно больших N и малых искажениях $C(n)$, атака не является успешной для любой статистики шума.

Действительно, рассмотрим вероятность ошибки для легитимного пользователя, который знает $\pi(n)$, $n = 1, 2 \dots N$.

$$p(1/0) = P\{\Lambda \leq 0 / b = 0\}, p(0/1) = P\{\Lambda > 0 / b = 1\} \quad (4)$$

$$\Lambda = \sum_{n=1}^N ((C'_w(n) - m_c) \pi(n))$$

При $N \rightarrow \infty$, $\Lambda \sim N(E(\Lambda), Var(\Lambda))$ (ЦПТ теории вероятностей)

$$E\{\Lambda\} = E\left\{\sum_{n=1}^N (C(n) - m_c + \alpha(-1)^b \pi(n) + \varepsilon(n)) \pi(n)\right\} = \alpha(-1)^b N \quad (5)$$

$$\begin{aligned} Var\{\Lambda\} &= \sum_{n=1}^N E\{((C(n) - m_c + \varepsilon(n)) \pi(n))^2\} = \\ &= NE\{(C(n) - m_c)^2 + 2\varepsilon(n)\pi(n)(C(n) - m_c) + \varepsilon^2(n)\} = N(\sigma_c^2 + \sigma_\varepsilon^2), \end{aligned} \quad (6)$$

где $\sigma_c^2 = Var\{C(n)\}$, $\sigma_\varepsilon^2 = Var\{\varepsilon(n)\}$.

Если мы положим $m_c = 0$ в (3), то получим вместо (6):

$$Var\{\Lambda\} = N(E\{C^2(n)\} + \sigma_\varepsilon^2) = N(Var\{C(n)\} + m_c^2 + \sigma_c^2) = N(\sigma_c^2 + m_c^2 + \sigma_\varepsilon^2) \quad (7)$$

$$Var\{\Lambda\} \geq Var\{\Lambda\}$$

Положим сначала $b = 0$. Тогда

$$p(1/0) = p\{\Lambda \leq 0 / b = 0\} = Q(E\{\Lambda\} / \sqrt{Var\{\Lambda\}}),$$

где $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt$.

Подставляя (5) и (6) в (8), получим

$$p(1/0) = Q\left(\frac{\alpha}{\sqrt{N/(\sigma_c^2 + \sigma_\varepsilon^2)}}\right) \quad (9)$$

(Легко проверить, что аналогичное выражение получается и для случая $b = 1$, т.е. $p(1/0) = p(0/1) = p$).

Введем обозначения:

$$\eta_w = \frac{\sigma_c^2}{\alpha^2} \quad \text{- (отношение сигнал/шум после погружения WM),} \quad (10)$$

$$\eta_a = \frac{\sigma_z^2}{\alpha^2 + \sigma_\varepsilon^2} \quad \text{(отношение сигнал/шум после атаки).} \quad (11)$$

Подставляя (10) и (11) в (9), получим

$$p = Q\left(\sqrt{N\eta_a / (\eta_a\eta_w + \eta_w - \eta_a)}\right) \quad (12)$$

Типичным является случай, когда $\eta_w \geq \eta_a \gg 1$.

Тогда для (12) получаем приближение

$$p = Q\left(\sqrt{N/\eta_w}\right) \quad (13)$$

Рассмотрим теперь случай *информированного декодера*, когда принятие решения о вложении информации выполняется по правилу:

$$\Lambda' \begin{cases} \geq 0 \rightarrow b = 0 \\ < 0 \rightarrow b = 1, \end{cases} \quad (14)$$

где
$$\Lambda' = \sum_{n=1}^N (C'_w(n) - C(n))\pi(n) \quad (15)$$

Используя ЦПТ получаем:

$$p' = P\{\Lambda' < 0 \mid b = 0\} = Q\left(\frac{E\{\Lambda'\}}{\sqrt{Var\{\Lambda'\}}}\right) \quad (16)$$

$$E\{\Lambda'\} = E\left\{\sum_{n=1}^N (C(n) + \alpha\pi(n) + \varepsilon(n) - C(n))\pi(n)\right\} = \alpha N$$

$$Var\{\Lambda'\} = Var\left\{\sum_{n=1}^N \varepsilon(n)\pi(n)\right\} = N \overset{(18)}{Var\{\varepsilon(n)\}} Var\{\pi(n)\} = N\sigma_\varepsilon^2$$

Подставляя (17), (18) в (16) и используя (10), (11), получим

$$p' = Q\left(\frac{\alpha\sqrt{N}}{\sigma_\varepsilon}\right) = Q\left(\sqrt{\frac{N}{(\eta-1)}}\right), \quad (19)$$

где
$$\eta = \eta_w / \eta_a$$

Сравнивая p по (12) и p' по (19) мы видим, что $p \geq p'$.
Действительно, выбирая $p = p'$, но разные N и N' получаем

$$N' / (\eta - 1) = N / \eta_w \Rightarrow N / N' = \eta_w / (\eta - 1) \quad (20)$$

Пример. Положим $\eta_w = 120$, $\eta_a = 100$. Тогда $N/N' = 600$.

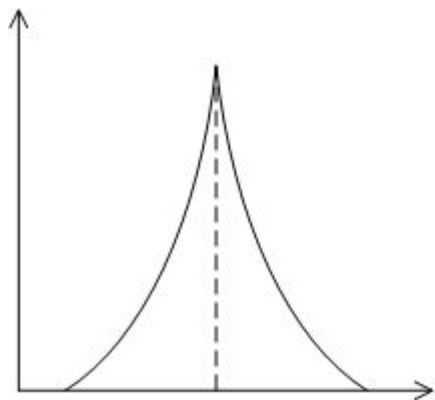
Это означает, что для “слепого” декодера скорость вложения будет в 600 раз меньше, чем для информированного!

Чтобы уменьшить эту разницу (для неизвестного у декодера ПС) используют *информированный кодер* (метод погружения), который отличается от (1).

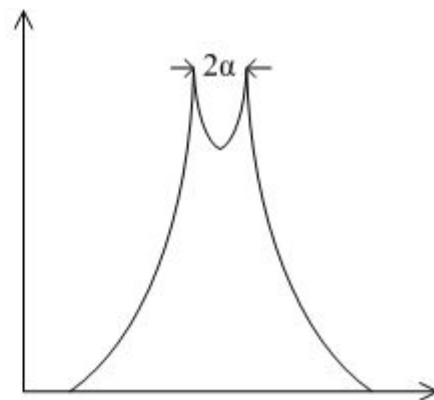
Однако это может привести к лучшему обнаружению СГС и поэтому он используется обычно для ЦВЗ (см. далее).

Обнаружение СГС-ШПС

1. По одномерной статистике (гистограмме)



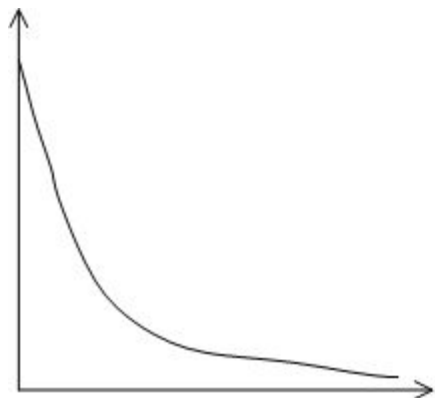
а) ПС



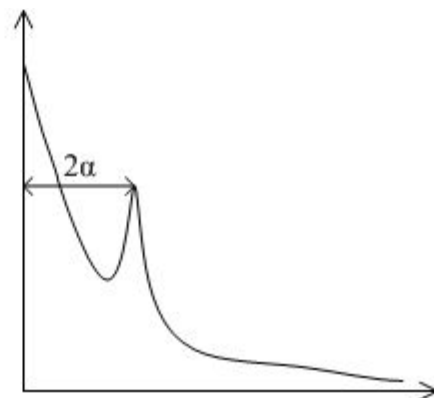
б) СГС-ШПС

2. По статистике второго порядка

(По гистограммам модулей разностей яркостей смежных пикселей, $|C(n+1)-C(n)|$)



а) ПС



а) СГС-ШПС

α	№ изображения	P	X^2
1	1	1	51248
		0,5	15885
		0,1	45781
		0	58211
	2	1	40026
		0,5	3629
		0,1	43308
		0	60000
2	1	1	54824
		0,5	47791
		0,1	54739
		0	58211
	2	1	20386
		0,5	22046
		0,1	50453
		0	60000
3	1	1	49463
		0,5	50267
		0,1	55932
		0	58211
	2	1	39683
		0,5	44964
		0,1	55701
		0	60000

3. Использование критерия X^2 (см. Лекцию 2)

Можно сделать вывод, что этот метод работает для изображений высокого качества (без цифрового шума).

Лучшие результаты мы получаем для вероятности встраивания $P = 0.5$.

4. ПВА .

Экспериментальные результаты расчета \tilde{P} по методу ПВА ля метода СГ-ШПС

α	P	№ изображения				
		1	2	3	4	5
1	1	0,00306	0	0,03965	0,00525	0,00603
	0,5	0,03307	0	0,10544	0,01854	0,02475
	0,1	0	0	0,00580	0,00482	0,00468
2	1	0,00030	0	0	0,00005	0
	0,5	0	0,05735	0	0	0,00397
	0,1	0	0,00211	0	0	0,00050
3	1	0,00050	0,00335	0,01936	0,00215	0
	0,5	0,00061	0	0	0,00183	0
	0,1	0,00009	0,00397	0	0,00012	0
0		0	0	0	0	0,00005

Видно, что этот метод работает не очень хорошо, но он может быть использован в сочетании с другими методами.

5. Метод, основанный на подсчете нулей в гистограмме
Количество нулей в гистограмме СО всегда меньше, чем в ПО

Результаты подсчета количества нулей гистограммы для 5 различных изображений

α	P	№ изображения				
		1	2	3	4	5
1	1	165	95	154	138	161
	0,5	139	5	128	93	127
	0,1	141	6	134	100	131
2	1	153	68	133	131	157
	0,5	134	36	120	109	130
	0,1	138	36	127	120	136
3	1	165	95	154	138	161
	0,5	139	5	127	93	127
	0,1	140	5	132	95	132
0		201	162	193	174	200

Видно, что метод работает, однако не для всех изображений. Лучшие результаты при $P=0.5$.

6. По статистике суммы квадратов разностей яркостей соседних пикселей

$$\Gamma = \frac{1}{2N_0\sigma_c^2} \sum_{n=1}^{N_0} (C_W(n+1) - C_W(n))^2, \quad (21)$$

где $\sigma_c^2 = \frac{1}{N_0} \sum_{n=1}^{N_0} C_W^2(n).$

N_0 – общее число пикселей изображения.

Метод обнаружения СГС-ШПС:

$\Gamma > \gamma_0 \Rightarrow$ СГС присутствует,

$\Gamma \leq \gamma_0 \Rightarrow$ СГС отсутствует.

(22)

Действительно, для ПО:

$$E\{\Gamma\} = \frac{N_0}{2N_0\sigma_c^2} E\{C^2(n+1) + C^2(n) - 2C(n+1)C(n)\} = 1 - R_c(n, n+1), \quad (23)$$

где $R_c(n, n+1)$ – нормированный коэффициент корреляции между яркостями соседних пикселей.

Замечание. Вложение ШПС-СГС по правилу (1) не обеспечит секретности, если при атаке известна $Var\{C(n)\} = \sigma_c^2$, поскольку тогда

$$Var\{C_w(n)\} = \sigma_c^2 + \alpha^2 > Var\{C(n)\}$$

Для секретной ШПС-СГС выполняется вложение по модифицированному правилу

$$C_w(n) = \beta C(n) + \alpha(-1)^b \pi(n), n = 1, 2, \dots, N, \quad (24)$$

где $\beta = \sqrt{1 - \frac{\alpha^2}{\sigma_w^2}}$.

Тогда $Var\{C_w(n)\} = Var\{C(n)\} = \sigma_c^2$ (это можно легко проверить).

$$\begin{aligned} E\{\Gamma'\} &= \frac{N_0}{2N_0\sigma_c^2} E\{(C_w(n+1) - C_w(n))^2\} = \\ &= \frac{N_0}{2N_0\sigma_c^2} E\left\{\left[\beta C(n+1) + \alpha(-1)^b \pi(n+1) - (\beta C(n) + \alpha(-1)^b \pi(n))\right]^2\right\} \end{aligned} \quad (25)$$

После преобразования (25) получим

$$E\{\Gamma'\} = 1 - \beta^2 R_c(n, n+1).$$

Поскольку $\beta < 1$, то $E\{\Gamma'\} \geq E\{\Gamma\}$, причем эта разница тем больше, чем больше $R_c(n, n+1)$ и обуславливает возможность обнаружения ШПС-СГС.

Проверим обнаруживаемость СГС-ШПС для 20 различных изображений размером ~ 300x200 с градациями серого при $\alpha = 1$.

№	Изображение	Значение Γ для ПС	Значение Γ' для СГС-ШПС
1	17	447 01720	447 20436
2	20	706 15072	706 82968
3	24	73 34841	73 30683
4	25	73 74376	74 12018
5	27	428 02232	428 67656
6	29	235 21896	235 48856
7	32	822 25568	823 40832
8	35	628 16864	628 25872
9	37	123 20428	123 74226
10	38	303 16216	303 88040
11	41	643 62788	643 89600
12	43	200 28484	200 27456
13	44	135 42418	135 94816
14	45	738 28064	738 52400
15	47	1012 11280	1012 68328
16	49	746 53152	746 84432
17	50	51 07258	51 53188
18	51	217 11080	217 88028
19	52	1036 45688	1036 95648
20	53	376 28880	377 21680

Как видно из таблицы, типично изменяются последние пять цифр. Выберем для них порог $\lambda_0 = 46000$.

Тогда для выбранных ПС получаем:

- верно определена СПС-ШПС в 30 случаях
- получены ложные обнаружения в 3-х случаях
- пропущена СГС-ШПС в 7 случаях.

Из таблицы видно, что ПО и СГ различимы, но возникает проблема – как выбрать порог? Таким образом, данный подход применим в случае, когда необходимо различать, какой из двух образов ПО или СГ.

В действительности имеются и более эффективные методы обнаружения СГС-ШПС (см. далее “слепой” стегоанализ).

Пример реальных программ:

- DeepSound – в качестве исходных контейнеров используются файлы формата WAV (только несжатый, PCM), MP3, CDA, WMA, APE, FLAC, при этом на выходе, после внедрения информации, могут получиться файлы только форматов WAV, APE, FLAC. Для вложения информации в данной программе используется алгоритм вложения ШПС. Данная программа позволяет предварительно зашифровать скрываемую информацию криптографическим алгоритмом AES;
- Hallucinate – в качестве контейнеров используются файлы формата BMP, PNG. В качестве стенографического алгоритма используется алгоритм НЗБ;
- JHide – в качестве контейнеров используются файлы формата BMP, PNG, TIFF. В качестве стенографического алгоритма используется алгоритм НЗБ;
- OpenPuff – в качестве контейнеров используются медиафайлы с неподвижными картинками, видеопотоком или аудиозаписями (например, MP4, MPG, VOB). В качестве стенографического алгоритма используется алгоритм НЗБ. Скрываемая информация также защищается криптографическим стойким генератором псевдослучайных чисел (CSPRNG – Cryptographically secure pseudorandom number generator);
- OpenStego – в качестве исходных контейнеров используются файлы формата MP, PNG, JPG, GIF, WBMP. Есть возможность использования шифрования типа AES;
- QuickStego – в качестве исходных контейнеров используются файлы формата BMP, JPG, GIF, при этом на выходе, после внедрения информации, могут получиться файлы только формата BMP. В платной версии программы возможно использовать файлы формата WAV, MP3. В качестве стенографического алгоритма используется алгоритм НЗБ;
- Steghide – в качестве контейнеров используются файлы формата JPG, BMP, WAV и AU. В качестве стенографического алгоритма используется алгоритм НЗБ;
- Hide'N'Send – в качестве контейнеров используются файлы только формата JPEG. В качестве стенографических алгоритмов используются алгоритмы НЗБ и F5. Для шифрования вкладываемой информации криптографическими методами используются алгоритмы AES³⁰, RC4, RC2.