

03, март 2016

УДК 003.26.004.7.004.9

Экспериментальные исследования стеганографического метода на основе фазового кодирования

***Мессерле А.А.**, студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Информационная безопасность»*

***Осенов Б.О.**, студент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Информационная безопасность»*

*Научный руководитель: Чичварин Н.В., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Информационная безопасность»
iu8@bmstu.ru*

Введение

В настоящее время существует довольно много информации, требующей определенных мер защиты от воровства, подделки или модификации. В частности, чувствительной информацией является проектная документация. Можно использовать шифрование, хэш-функции и контрольные суммы, а также в целях защиты от копирования – цифровые водяные знаки (ЦВЗ), специальные (контрольные) метки и другие методы. Одно из доминирующих современных направлений в стеганографии – скрытие данных в звуковых файлах и стегоанализ таких контейнеров с сообщениями. В данной работе мы рассмотрим внедрение сообщения в звук на основе фазового кодирования (не в последнюю очередь это связано с малой освещенностью самого метода и атак на него как в русскоязычных, так и в иностранных публикациях).

В данной работе рассматриваются возможности:

А) скрытного внедрения в цифровой звуковой файл некой удостоверяющей (проверочной) информации, например, данные о предприятии-изготовителе, дате разработки, контрольные суммы и т.д.

Б) скрытие самого документа (например, изображения) в звуковом файле методом фазового кодирования

В) исследование методов, позволяющих определить наличие скрытых данных в контейнере

В качестве контейнера был выбран звук в формате wav (Waveform Audio File Format), так как звуковые файлы такого формата имеют чрезвычайно малое количество методов, способных детектировать записанное таким образом сообщение, в отличие от тех же стего-изображений, где широко распространен поиск младших битов, гистограммный анализ, RS-анализ. Анализировать звук довольно проблематично – младших битов как таковых чрезвычайно много, инструментов для вейвлет-анализа немного, а используя качественное рассеяние данных в спектрах позволит ликвидировать угрозу спектрального анализа (отличить шум от данных на спектрограммах становится крайне сложной задачей). При использовании уникального звукового файла, злоумышленник, не знающий, что и где искать, будет вынужден опустить руки. Таким образом можно защитить информацию, и, при необходимости, доказать свои права на документ.

В данной статье мы также сделаем обзор возможных атак на контейнер с данными.

Цель проведенных исследований и решенные задачи

Целью исследований являлось, во-первых, определить применимость метода фазового кодирования на практике, во-вторых, испытание контейнеров с записанными данным алгоритмом сообщениями на устойчивость к атакам.

Были решены следующие задачи: написана программа реализующая алгоритм фазового кодирования в звуке, успешно записаны и восстановлены стего сообщения, представляющее из себя конструкторские файлы (тексты и изображения).

Результаты исследований

Теория

Из набора гармоник, образующих звуковой сигнал выделяют два вида спектра – амплитудный и фазовый.

Амплитудный спектр представляет собой набор амплитуд всех гармоник, и представляется в виде вертикальных линий, длины которых пропорциональны амплитудным значениям гармонических составляющих, а место на горизонтальной оси определяется номером гармоники данной гармоники (рис. 1).

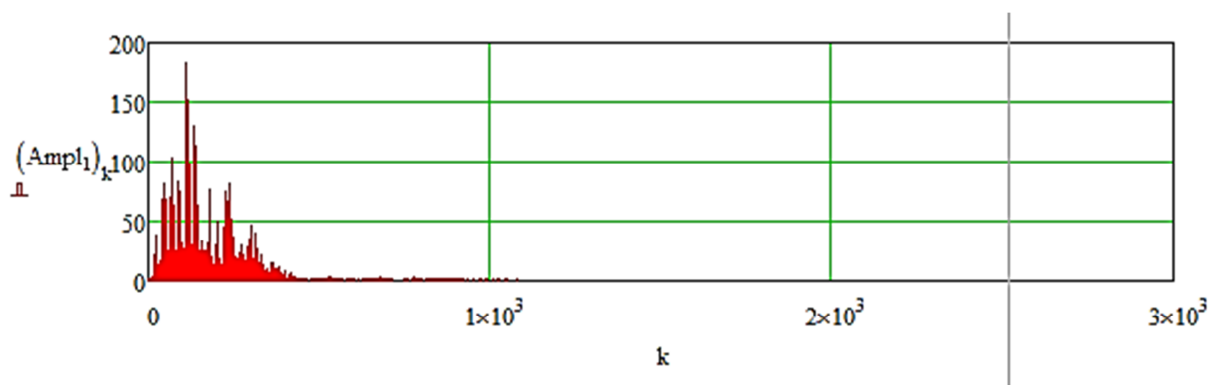


Рис. 1. Пример амплитудного спектра

Фазовый спектр, аналогично амплитудному, является совокупностью начальных фаз всех гармонических составляющих, и отображается также в виде набора вертикальных линий (рис. 2). Начальные фазы принято измерять в пределах от $-\pi$ до π [1]. В данной работе мы будем оперировать дискретными спектрами.

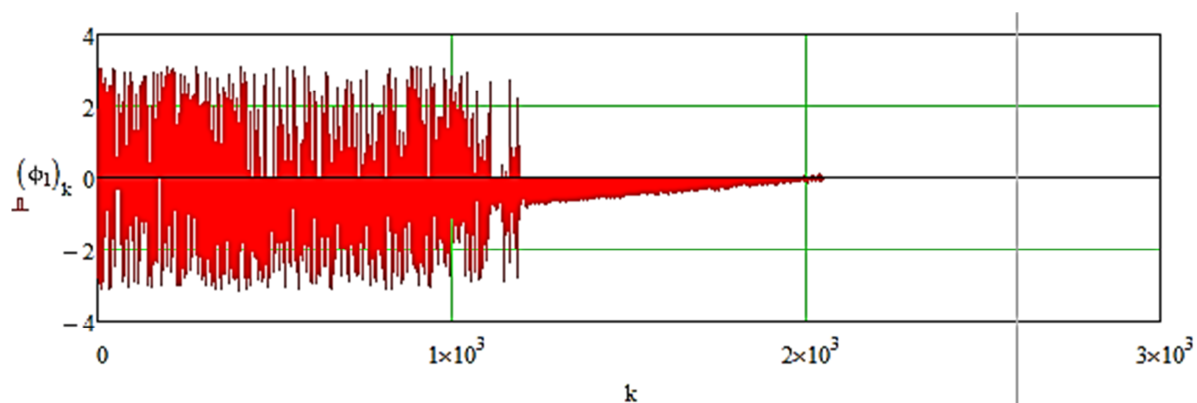


Рис. 2. Пример фазового спектра

Метод фазового кодирования в частотной области

Главный принцип такого метода стеганографии для звуковых файлов заключается в том, что фаза исходного сегмента сигнала заменяется на некоторую другую, опорную фазу. Характер изменения этой опорной фазы и будет отражать информацию, которую мы хотим скрытно передать посредством звукового файла. Также необходимо согласовать фазы сегментов, чтобы сохранить разностную фазу между ними, так как именно к относительной разности фаз человеческое ухо наиболее чувствительно. Более подробное описание метода можно найти в [2].

Данный метод является очень эффективным по соотношению сигнал/шум. Из-за изменения соотношения фаз между каждыми частотными составляющими происходит рассеивание (разброс) фазы. Однако, при малых изменениях фазы, скрытие данных не будет восприниматься на слух (однако также восприятие на слух зависит и от самого

контейнера – возможно появление шумов). [2] Именно поэтому записываемые биты кодируются «0» = $\frac{\pi}{2}$ и «1» = $-\frac{\pi}{2}$.

Рассмотрим внедрение сообщения для звука смеха формата .wav, длительностью 18 секунд и битрейтом 1411 Кб/сек. Скрываемое сообщение: «he laughs best who laughs last, he who laughs last laughs longest» (перевод: хорошо смеется тот, кто смеется последним). Для скрытия одного символа (байта) сообщения достаточно 8 бит звука (так как текст в формате ASCII). Следует отметить, что недостатком метода является тот факт, что длина ключа должна быть степенью двойки. В данной статье модифицированный алгоритм оперирует ключом, который состоит из «классического ключа» - степени двойки и длины скрытого сообщения. В данном случае эти параметры имеют значения 2^{12} и 520 (бит).

Исходная последовательность первого канала разбилась на 199 сегментов. Затем были найдены амплитудный и фазовый спектры и произведено запоминание о разности фаз.

После этих операций осуществили конвертацию данных в последовательность битов и успешно записали эту последовательность в первый сегмент фазы (рис. 3).

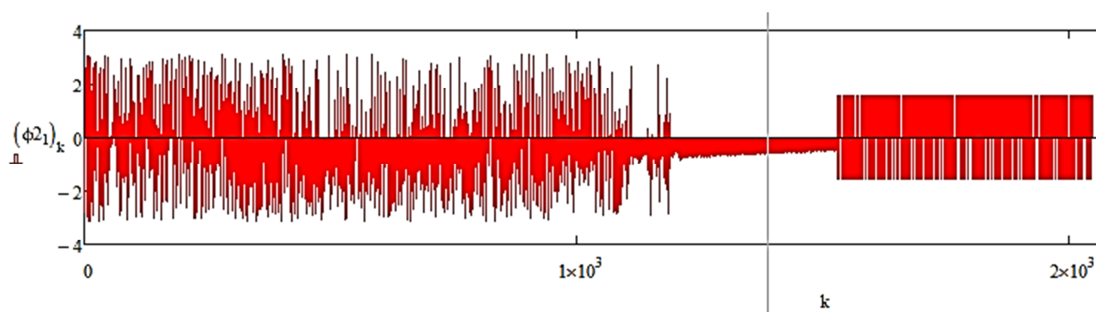


Рис. 3. Встроенные данные

При этом, разности фаз с остальными сегментами сохранены прежними (рис. 4).

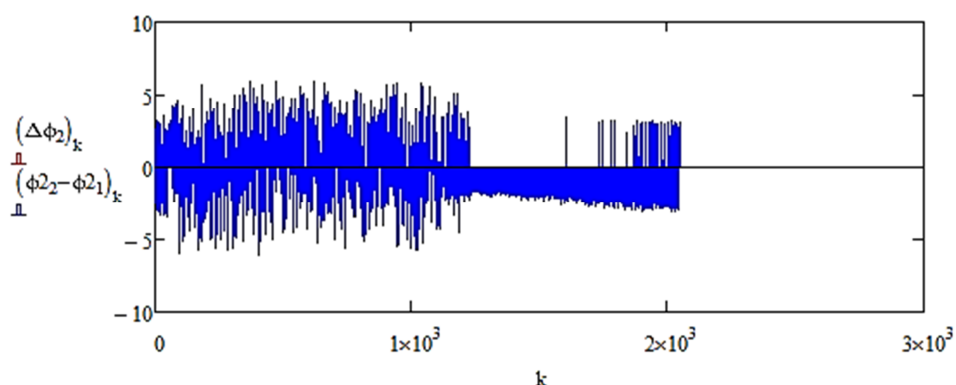


Рис. 3. Идентичность разностей фаз между первым и вторым сегментом

Амплитудные Фурье-образы каналов со стегосообщением и без совпадают (рис. 5).

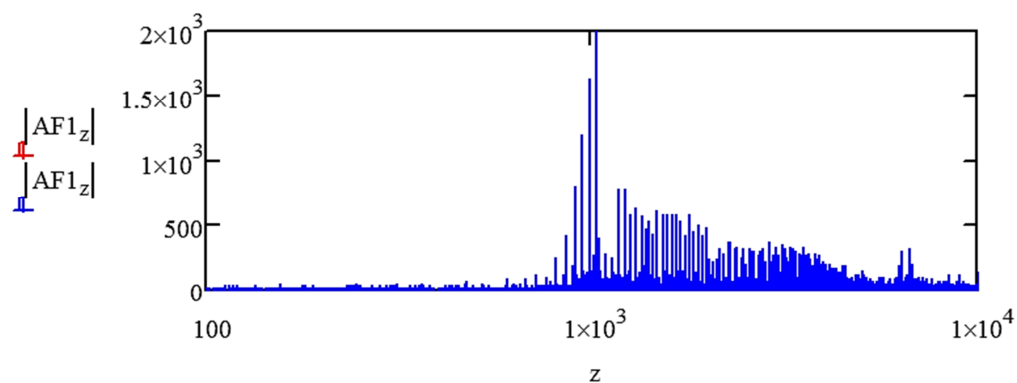


Рис. 4. Амплитудные Фурье-образы канала без стегосообщения и канал с внедренным сообщением

После проведения процедуры восстановления данных, полученное сообщение оказалось эквивалентным исходному. Следует отметить, что восстанавливались биты по допущению «1» = $\frac{\pi}{3}$ и «0» = $-\frac{\pi}{3}$, так как в процессе передачи могли возникнуть помехи.

При снижении данных порогов точность восстановления повышается.

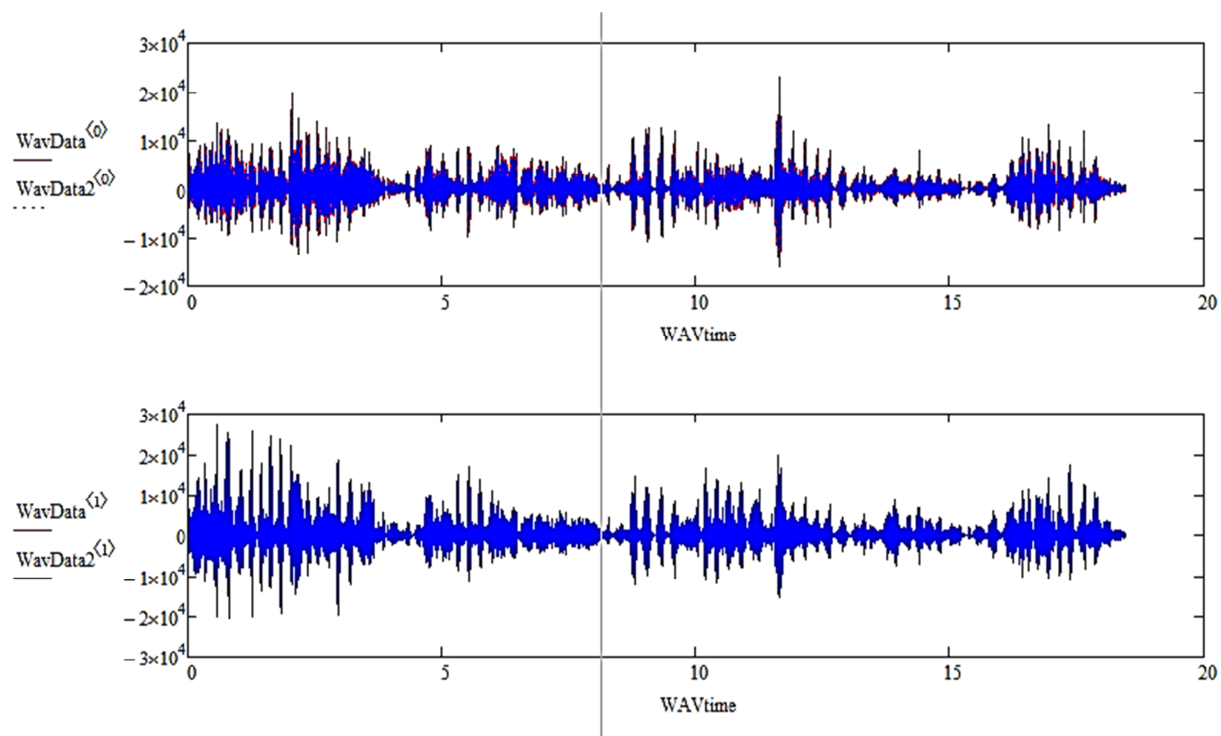


Рис. 5. Сравнение исходного сигнала и сигнала со скрытым сообщением «he laughs best who laughs last, he who laughs last laughs longest»

Очевидно, что второй канал полностью совпал с исходным (рис. 6), а первый канал отличается от исходного, однако визуально повторяет его форму (рис. 6; рис.7).

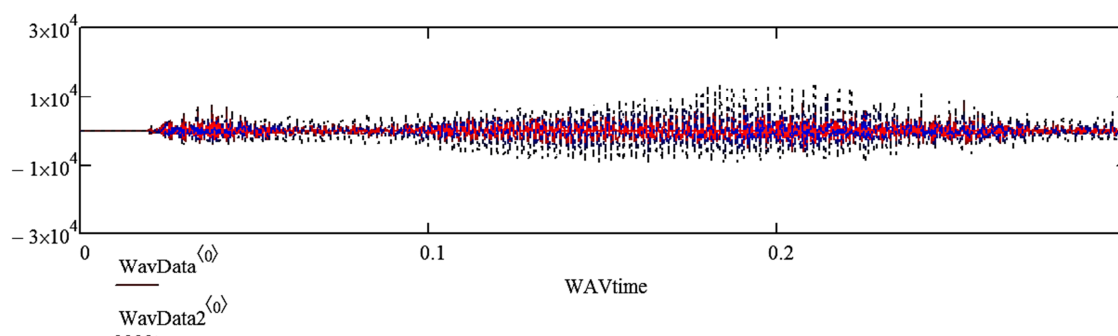


Рис. 7. Сравнение исходного сигнала и сигнала со скрытым сообщением «he laughs best who laughs last, he who laughs last laughs longest» в первом канале, увеличение

Таким образом можно выполнять «подписание» аудиозаписей – реализовано внедрение цифрового водяного знака (ЦВЗ). Преимущества фазового кодирования в этом случае в том, что при малом объеме записываемой информации, длительность записи практически не изменяется, а звуковые искажения не регистрируются на слух. Проверить ЦВЗ можно только зная ключ (это справедливо и для удаления ЦВЗ). Если злоумышленник захочет «очистить» имеющуюся у него в распоряжении аудиозапись, то ему придется определять количество сегментов, их длину, длину сообщения, а потом соединять запись обратно, что не является легкой задачей. Также данный способ внедрения позволяет скрывать и другие данные, которые пользователь хотел бы скрытно передать.

Следует отметить, что после внедрения сообщения, основные параметры записи (такие как битрейт, частота дискретизации) остаются неизменными, а при малом сообщении для сокрытия – длина и размер также не меняются.

Приведем пример сокрытия изображения схемы из САПР в формате PNG (portable network graphics). Дополнительно перед записыванием изображение было сжато. Исходное и восстановленные изображения (рис. 8) полностью идентичны.

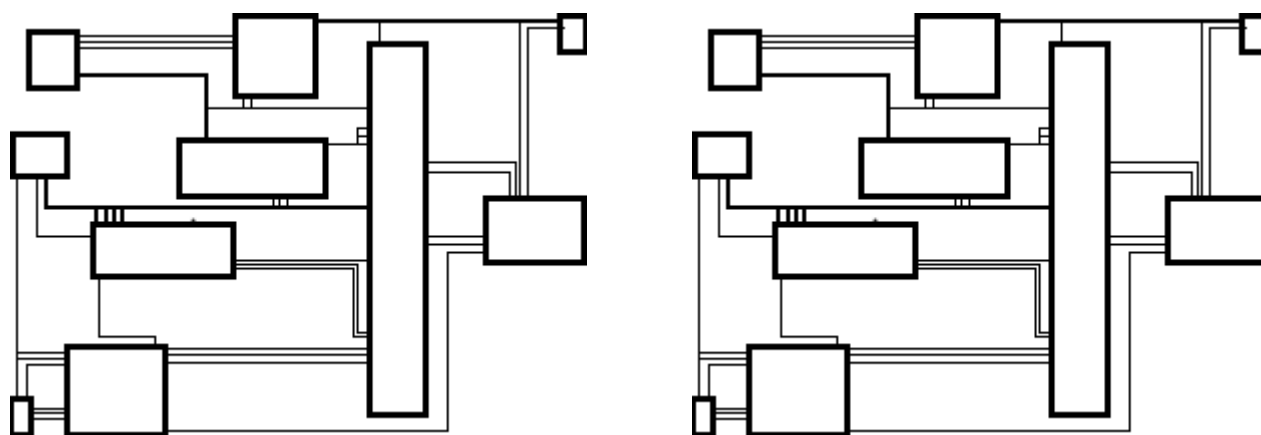


Рис. 8. Исходное (слева) и восстановленные изображения

Проведенные исследования

После получения нескольких контейнеров с сообщениями были исследованы методы атак по следующим направлениям:

- Определение максимальной пропускной способности
- Определение лучших частот для контейнера
- Устойчивость контейнера к фильтрации
- Устойчивость контейнера к шумам
- Визуальные изменения спектрограмм
- Устойчивость контейнера к перекодированию с потерями
- Анализ коэффициента сжимаемости контейнера
- Устойчивость сообщения к удалению части контейнера

Определение максимальной пропускной способности

В данном исследовании мы оценим длины сообщений в различных контейнерах, которые внедряются методом фазового кодирования. Результаты с комментариями для разных длин контейнеров отражены в таблицах 1-3. В таблицах 1-3 длина сообщения выражена в символах, однако поскольку 1 символ кодируется одним байтом, то можно оценить объем скрываемого сообщения.

Таблица 1

Результаты внедрения для короткого файла

Продолжительность (сек)			18.469
Размер звукового файла (байт)			3 257 934
Длина сообщения (символов)	Итоговый размер (байт)	Итоговая длительность (сек)	Результат восстановления
1023	3 276 846	18.576	Не восстановлено
511	3 276 846	18.576	Восстановлено полностью
255	3 276 846	18.576	Восстановлено полностью
55	3 260 462	18.483	Восстановлено полностью
31	3 260 462	18.483	Не восстановлено
Минимальное - максимальное количество символов			33-511 (длины зависят от степени ключа)

Таблица 2

Результаты внедрения для файла средней длины

Продолжительность (сек)			60.291
Размер звукового файла (байт)			10 635 490
Длина сообщения (символов)	Итоговый размер (байт)	Итоговая длительность (сек)	Результат восстановления
4098	11 534 382	65.387	Частично восстановлено (99%)
2049	11 010 094	62.415	Восстановлено полностью
2047	10 747 950	60.929	Частично восстановлено (97%)
1023	10 747 950	60.929	Восстановлено полностью
257	10 682 414	60.558	Частично восстановлено (64%)
Минимальное - максимальное количество символов			513 -4096 (с ошибками восстановления при длине 1025-2048)

Таблица 3

Результаты внедрения для длинного файла

Продолжительность (сек)			93.2
Размер звукового файла (байт)			16 440 558
Длина сообщения (символов)	Итоговый размер (байт)	Итоговая длительность (сек)	Результат восстановления
1029	16 515 118	93.623	Частично восстановлено (50%)
2049	16 777 262	95.109	Восстановлено полностью
4098	16 777 262	95.109	Восстановлено полностью
8196	16 777 262	95.109	Частично восстановлено (99%)
16392	16 777 262	95.109	Частично восстановлено (66%)
Минимальное - максимальное количество символов			2049-8187

Файлы продолжительностью свыше 95 секунд не обрабатываются из-за ограничений среды реализации алгоритма (Mathcad 15).

Таким образом, алгоритм позволяет записать в среднем от 33 до 8187 символов, в зависимости от продолжительности аудиоконтейнера, и извлечь сообщение с помехами в

малом количестве случаев. Отметим, что полученное в результате экспериментов значение в 8187 символ (байт) является достаточно большим, по сравнению с другими алгоритмами, в том числе коммерческими, реализующими, например, внедрение ЦВЗ в аудиозапись.

Определение лучших частот для контейнера

Попытаемся определить, в каких областях частот метод работает лучше, в том числе определим, где искажения могут быть значительны. Будем записывать сообщение с длиной в 513 байт в звуки различных частот. Результаты отражены в таблице 4.

Таблица 4

Результаты внедрения в звуки разных частот

Частота звука	Описание изменений, воспринимаемых на слух	Результат восстановления
95 Гц	Ощутимые трески	Не восстановлено
250 Гц	Несколько малоощутимых тресков	Частично восстановлено
1 кГц	Малозаметные потрескивания	Восстановлено полностью
2 кГц	Искажений не выявлено	Восстановлено полностью
4 кГц	Искажений не выявлено	Восстановлено полностью
10 кГц	Ощутимые негромкие трески	Не восстановлено
15 кГц	Трески	Не восстановлено
18 кГц	Искажений не выявлено	Восстановлено полностью
1 – 20 кГц	Искажения в соответствии с указанными выше зависимостями	Не восстановлено

Отметим, что в случае последнего контейнера (с записью в полном диапазоне частот), особенно сильно искажения проявились в области 10,5 – 15 кГц. Отсюда можно сделать вывод, что лучше всего использовать контейнеры с доминирующими средними частотами.

Устойчивость контейнера к фильтрации

Исследуем устойчивость контейнера к фильтрации. Применим фильтр к записи с внедренным сообщением «Published by BMSTU "Audio Stego Records", September 2015». Функция фильтра:

$$h_n = e^{-\frac{t_n^2}{r^2}}$$

Примем параметр $r=0.001$ для первого случая и $r=1$ для второго. Результаты восстановления отражены на рис. 9.

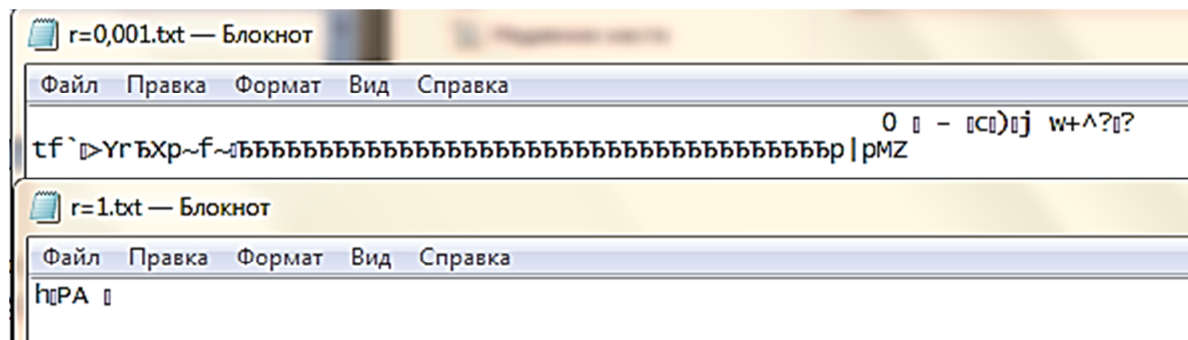


Рис. 9. Результаты восстановления при разных параметрах нелинейного фильтра

Очевидно, что сообщение после такой фильтрации восстановить не удалось. Проверим устойчивость к обычным искажениям, используемым в программах звуковой обработки (таких, как Adobe Audition или Sound Forge и их аналогов)(таблица 5).

Таблица 5

Восстановление из отфильтрованных сигналов

Название фильтра, описание	Результат восстановления
Эхо	Восстановлено без искажений
Экстрактор центрального канала (удаление голоса)	Восстановлено без искажений
Инверсия значений	Не восстановлено
Параметрическое нелинейное искажение	Восстановлено верно 12% сообщения
Постепенное затухание	Восстановлено без искажений
Усиление	Восстановлено без искажений
Нелинейный фильтр на основе преобразования Фурье	Не восстановлено
Сдвиг фазы	Восстановлено верно 40% сообщения (Published by BMSTU "Aud! @q@t-b)

Можно заключить, что аудио контейнер с сообщением устойчив к некоторым типам преобразований, особенно линейным, или не связанным с изменением фазовой составляющей сигнала.

Устойчивость контейнера к шумам

Добавим в канал со стегосообщением различные типы шумов и проследим изменения в восстановленных данных (таблицы 6-7).

Таблица 6

Восстановление после добавления шума (аддитивный белый шум)

Среднеквадратическое отклонение	Процент восстановления
0,0025	100%
0,255	100%
2,55	100%
4,53	100%
6,4	100%
7,18	98,2% (Published by BMSTU "Audio!Stego Records", September 2015)
8,06	94,7% (Publishad by BMSTU0"Audio STego Records", September 2015)
10,15	89,4%
14,33	66,6%
25,5	35%
45,34	15,7%
45,34 (с порогом восстановления $\pm \frac{\pi}{99}$)	22,8% (A]zOq!N`ad#arBTTm!gIwTcTb&7 (uO6CchE\ds m Kaxtember 2015)

Последняя строка таблицы показывает, что даже при высокой степени зашумления существует возможность восстановить часть данных путем понижения порога восстановления.

Таблица 7

Восстановление после добавления шума (аддитивный равномерный шум)

Максимальное значение шума	Процент восстановления
20	100%
23	96,5%
24	98,2%
25	96,5%
30	94,7%
40	75,4%
40 (с порогом восстановления $\pm \frac{\pi}{99}$)	94,7%
60	71,9%
60 (с порогом восстановления $\pm \frac{\pi}{99}$)	70,1%
70	35%
70 (с порогом восстановления $\pm \frac{\pi}{99}$)	63,1%
80	33,3%
80 (с порогом восстановления $\pm \frac{\pi}{99}$)	36,8%

90	22,8%
90 (с порогом восстановления $\pm \frac{\pi}{99}$)	29,8%
100	21%
100 (с порогом восстановления $\pm \frac{\pi}{99}$)	29,8%
110	19,2%
110 (с порогом восстановления $\pm \frac{\pi}{99}$)	24,5%

Визуальные изменения спектрограмм

Рассмотрим спектрограммы оригинального (рис. 10) и содержащего сообщения (рис. 11) сигналов.

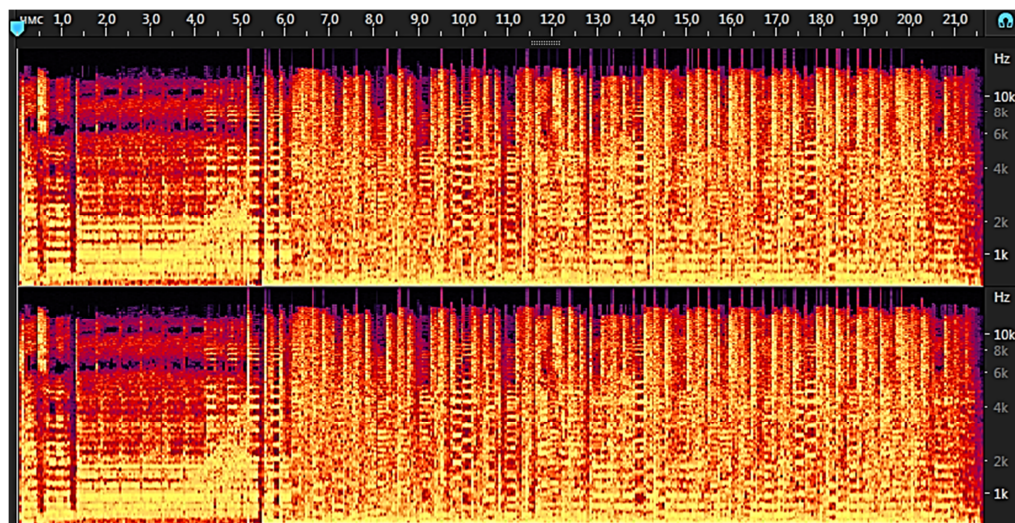


Рис. 10. Спектрограмма оригинального сигнала

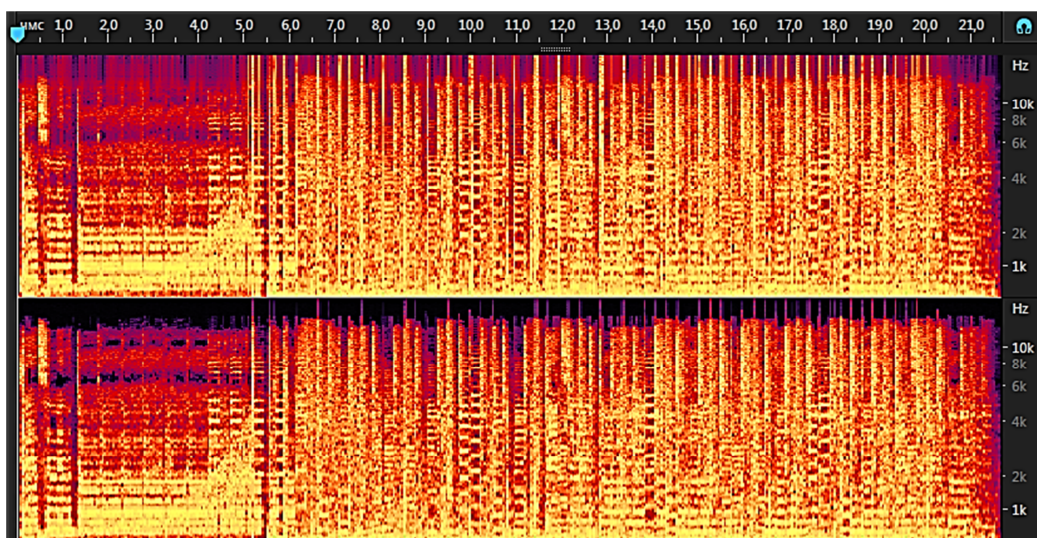


Рис. 61. Спектрограмма сигнала со скрытым сообщением

Очевидно, что внедрение сообщения в первый канал аудиозаписи привело к появлению на спектрограмме (рис. 11, вверху) характерных искажений, которые особенно заметны в области от 10 кГц и выше, при этом второй канал, в который данные не заносились, остался без изменений. Можно заключить, что опыт анализа различных спектрограмм позволяет визуально определять наличие скрытых данных в аудиоконтейнере.

Заметим, что на спектрограммах звуков различной частоты с внедренными сообщениями проявляются признаки из пункта «Определение лучших частот для контейнера», то есть, например, спектрограмма для 95 Гц имеет сплошной вид, в то время как спектрограмма для звука частотой 2 кГц имеет дискретный характер. Спектрограммы звуков в пределах 10,5 – 15 кГц имеют плотный дискретный характер, а после 18 кГц опять становятся дискретными.

Устойчивость контейнера к перекодированию с потерями

В этой части проверяется устойчивость контейнера к перекодированию в другие (более популярные) форматы с потерями. Результаты исследований отражены в таблице 8.

Таблица 8

Восстановление после перекодирования

Формат	Результат восстановления
MP3 (MPEG-1 Audio Layer 3)	f_Тhgx dq2-(SuzuDmjer 01
OGG	Published by BMSTU "Audio Stego Records", September 2015
WMA (Windows Media Audio)	@u ТhТh z2,0\е}uember 2015

В данном случае, перекодирование в распространенный формат MP3 привело к практически полной потере данных (успешно восстановлено менее 10%); в формате WMA восстановилось лишь 17,5% данных, однако при перекодировании в свободный формат OGG внедренные данные полностью сохранились. Поэтому метод фазового кодирования лучше не использовать в таких распространенных форматах как WMA и MP3.

Анализ коэффициента сжимаемости контейнера

Отдельным классом атак является исследование коэффициентов сжатия данных [4], поскольку данный метод стегоанализа не зависит от алгоритма внедрения сообщений. Данный метод основан на том, что контейнер с данными сжимается хуже, чем пустой

контейнер. Зная значение порога, можно сделать предположение о наличии в контейнере скрытых данных.

Рассмотрим различия в степенях сжатия различными форматами оригинального аудиосигнала и контейнером с данными различной длины. Для этого мы воспользуемся программами 7z и WinRAR и применим следующие форматы сжатия: ZIP (алгоритм Deflate), 7z (алгоритм LZMA), RAR и отобразим результаты эксперимента на рис. 12-14.

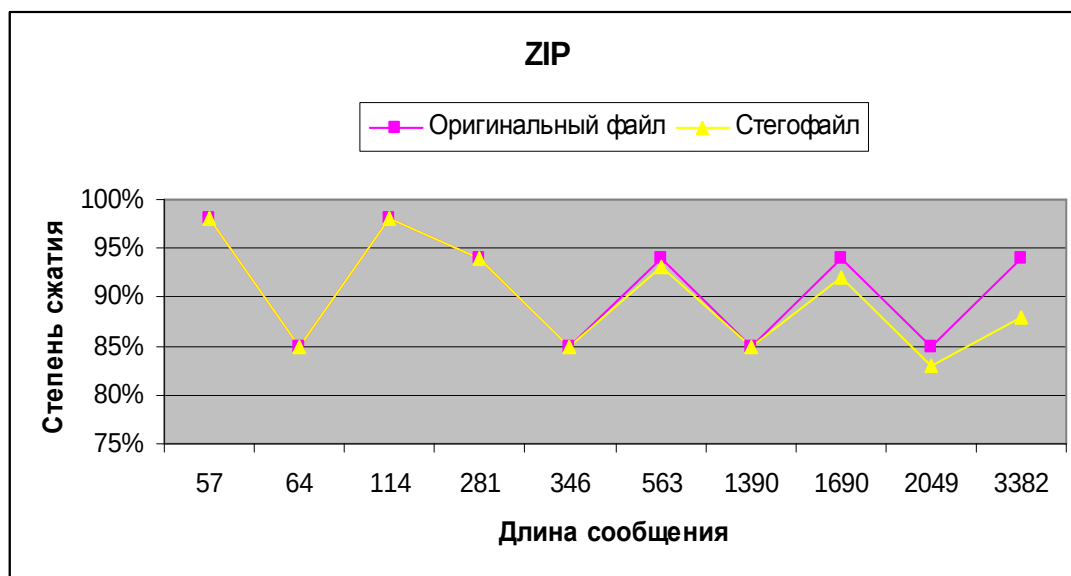


Рис. 12. Степени сжатия алгоритмом Deflate

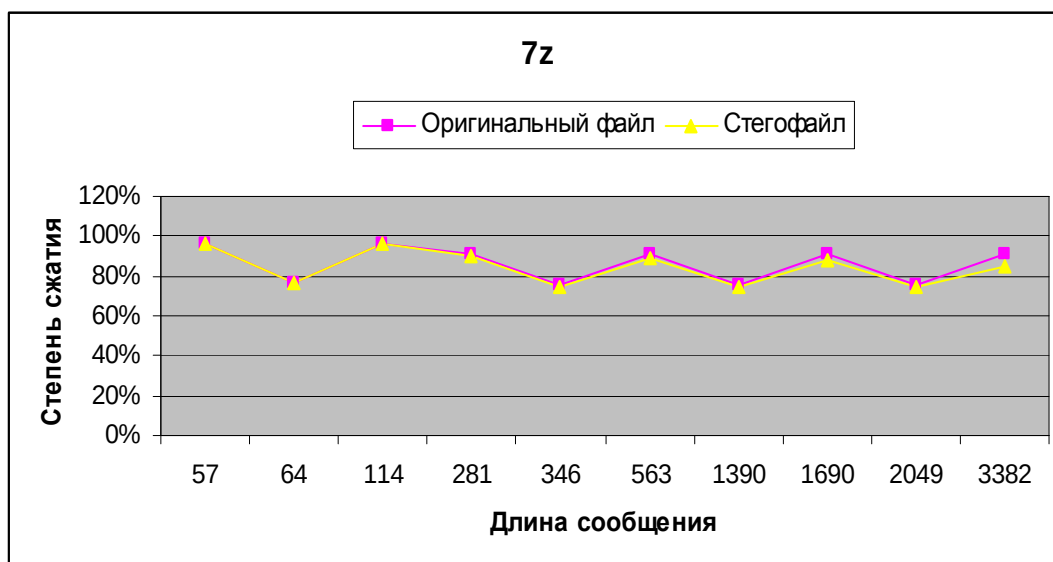


Рис. 13. Степени сжатия алгоритмом LZMA

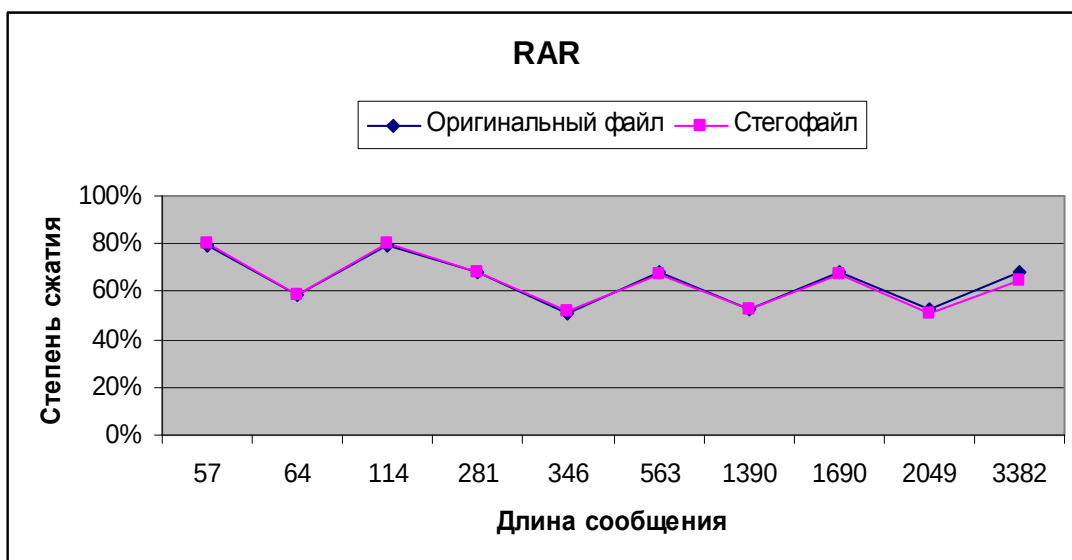


Рис. 7. Степени сжатия RAR

Отметим, что при малой длине сообщения (менее 250 символов) оба файла сжимаются практически одинаково, в нескольких случаях файл с сообщением сжимается лучше. С ростом объема сообщения, как и ожидалось, стегофайл сжимается хуже, что особенно заметно при длине сообщения свыше 346 символов. Форматы 7z и ZIP показывают более хорошие результаты для анализа, и могут служить основными для определения порогов сжатия, при которых можно считать, что в контейнер внесены данные.

Наиболее интересные результаты можно получить, анализируя сжимаемость в звуках разных частот. Рассмотрим несколько случаев с форматами сжатия RAR, RAR5 и ZIP, с учетом результатов раздела «Определение лучших частот для контейнера» (табл. 9-11).

Таблица 9

Степени сжатия звуков одной частоты форматом RAR

Частоты	Степень сжатия, %	
	Оригинального файла	Файла со стего (513 символов)
1 кГц	64	63
2 кГц	59	59
4 кГц	25	43

Таблица 10

Степени сжатия звуков одной частоты форматом RAR5

Частоты	Степень сжатия, %	
	Оригинального файла	Файла со стего (513 символов)
1 кГц	53	53
2 кГц	21	32
4 кГц	25	43

Таблица 11

Степени сжатия звуков одной частоты форматом ZIP

Частоты	Степень сжатия, %	
	Оригинального файла	Файла со стего (513 символов)
1 кГц	98	97
2 кГц	55	68
4 кГц	63	78

Можно отметить, что если комбинировать несколько методов сжатия, то можно получить значительные расхождения в уровнях сжатия контейнеров, что значительно облегчит выработку порога определения содержания внедренных данных. Наглядным примером служит степень сжатия звука частотой 2 кГц, которая значительно меняется при использовании разных форматов сжатия.

Устойчивость сообщения к удалению части контейнера

В данном тесте мы проверим, что случится при обрезке контейнера – удалении различных частей аудиосигнала (табл. 12). Исходный сигнал имел продолжительность 1,00,290 мин., а сигнал с внедренным стего сообщением длиной в 2049 символов - 1,02,415 мин. (в конце образовался беззвучный участок). Тогда длина первого сегмента (в котором сокрыты наши данные) составит около 2,85 секунд.

Таблица 12

Восстановление после удаления фрагментов аудиоконтейнера

Удаленные данные	Результат восстановления
2 секунды с начала	Ничего не восстановилось (удален 1 сегмент)
4 секунды в центре	Восстановлено полностью
2 секунды в конце	Восстановлено полностью
Оставили первые 10 секунд	Восстановлено полностью
Оставили первые 4 секунд	Восстановлено полностью
Оставили первые 3 секунд	Восстановлено полностью
Оставили первые 2 секунд	Ошибка программы (длина сегмента

Следовательно, если в начале аудиофайла присутствует беззвучный участок (например, задержка при записи), то такой файл лучше не использовать в качестве контейнера, поскольку некоторые пользователи и программы производят обрезку аудиозаписей не только в конце, но и в начале, что может привести к потере скрытых данных.

Заключение

Следует отметить, что в данной работе был исследован простейший случай кодирования информации, при котором 1 бит сообщения заменялся на 1 значение фазы. Данный способ можно модифицировать, кодируя дуплеты битов – например «00» = $\frac{\pi}{4}$, «01» = $\frac{3\pi}{4}$, «10» = $-\frac{\pi}{4}$, «11» = $-\frac{3\pi}{4}$ и восстанавливать в соответствующих значениям интервалах. Это позволит повысить емкость контейнера в 2 раза, однако может привести к тому, что изменения станут ощущаться на слух. Также, пропускную способность контейнера можно увеличить в два раза, используя оба звуковых канала вместо одного. В результате проведенных исследований можно сделать следующие выводы:

1. Метод фазового кодирования позволяет внедрить в аудиоконтейнер данные различного типа, в том числе и файлы САПР, объемом от 33 до 8187 байт, что, однако, зависит от системы реализации алгоритма, длительности аудиозаписи.
2. Внесенные искажения в области средних частот человеческим ухом не воспринимаются, однако искажения в областях низших и высоких частот могут быть восприняты как помехи (искажения).
3. Контейнер сохраняет данные после некоторых распространенных аудио-фильтров, однако после применения нелинейных фильтров (или фильтров, выполняющих манипуляции с фазовой составляющей сигнала) скрытое сообщение может исказиться или исчезнуть.
4. Данный метод обеспечивает хорошую устойчивость к шумам различного типа, восстанавливая данные в том числе с заниженным порогом восстановления.
5. Имея опыт, можно на основе спектрограмм определять наличие скрытых данных в контейнере.
6. Данные в контейнере не сохраняются при перекодировании звука в такие распространенные форматы как MP3 и WMA, однако успешно восстанавливаются при конвертации в формат OGG.

7. Наличие скрытых методом фазового кодирования данных можно обнаружить, анализируя степени сжатия различных контейнеров.
8. Контейнер имеет критическую область - первые несколько секунд (зависит от длины первого сегмента), при удалении которых сообщение восстановить уже не удастся. За исключением этого кусочка, можно удалить весь оставшийся объем контейнера и успешно восстановить скрытые данные после этого.

Метод фазового кодирования не является лучшим выбором, если требуется скрыть большой объем данных (существуют более подходящие для этого алгоритмы [3]), однако предоставляет возможности для внедрений ЦВЗ. Используя результаты исследований, представленных в данной статье, можно разработать более изощренные атаки, направленные на скрытые методом фазового кодирования данные. Полученные данные обладают некоторой научной новизной, поскольку сам метод и атаки на него мало освещены в публикациях, и актуальностью (данный метод может быть применен как для технологий ЦВЗ, так и для скрытия САПР-документации).

Список литературы

- [1]. Спектры периодических сигналов. Режим доступа: http://ets.ifmo.ru/osipov/os1/2_1.htm (дата обращения 31.08.2015).
- [2]. Коханович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. М.: МК-ПРЕСС, 2006. 288 с.
- [3]. Чичварин Н.В. Сопоставительный анализ областей применения характерных стеганографических алгоритмов // III Всероссийская научно-техническая конференция «Безопасные информационные технологии»: сб. трудов / под. ред. В. А.Матвеева. М.: НИИ радиоэлектроники и лазерной техники, 2012. С. 174-179.
- [4]. Жилкин М.Ю., Меленцова Н.А., Рябко Б.Я. Метод выявления скрытой информации, базирующийся на сжатии данных // Вычислительные технологии. 2007. Т. 12. Спец. вып. 4. С. 26-31.