Содержание

Введение

Системы интернета вещей с каждым днём всё больше входят в повседневную жизнь каждого человека. Одной из областей, в которых широко применяются такие устройства, является сбор метеорологических данных. Полезность и коммерческая ценность таких данных неоспорима – начиная от прогноза погоды и заканчивая обнаружением природных явлений, угрожающих жизни людей. До широкого распространения систем интернета вещей сбор метеоданных производился либо централизованно – компания закупала оборудование, возможно, создавала собственные программные комплексы, и использовала данные в своих нуждах, – либо не систематически – энтузиасты ставили ряд датчиков, и собранную с них информацию публиковали на сайте. С резким увеличением площади покрытия сетей 2G, 3G, 4G/LTE и значительным удешевлением вычислительных устройств и датчиков появились новые способы сбора данных, а также новые способы извлечь из них коммерческую выгоду. Большой проблемой для небольших компаний является высокий порог вхождения, так как компании наподобие «Яндекс» с сервисом Яндекс.Погода занимают фактически монопольное положение на рынке. Для того чтобы решить проблему высокого порога вхождения, требуется принципиально поменять модель, на которой будет основываться компания, что с одной стороны открывает широкие перспективы развития, а с другой – ставит новые задачи по обеспечению безопасности данных, представляющих коммерческий интерес.

На данный момент различные способы защиты информации изучены широко по отдельности, но слабо связаны между собой и обычно применяются для какой-то узкой задачи либо используются под заранее определённый круг устройств, часто имеющих аппаратную поддержку того или иного алгоритма. Кроме тогозачастую при реализации алгоритма предполагается исполнение на доверенных устройствах; подразумевается

невозможность декомпиляции кода, отсутствие влияния на аппаратную часть с целью искажения исполнения программы или получения секретной информации (атака по побочным каналам). Вопрос защиты информации после её продажи стеганографическими методами (в частности, вотермаркинг) предполагает наличие больших мощностей или больших объёмов данных, в которые можно вносить некоторые искажения, не влияющие на ценность данной информации (как в случае с мультимедиа-данными, где размер стеганографического контейнера велик, а искажения не видны людям; так и в случае исходного кода, часть из которого может не использоваться реальной программой, а выступать в качестве сигнатуры).

В рамках представленной работы мы рассмотрим модель защиты сервиса по сбору, обработке и дальнейшей продаже метеорологических данных. Специфика сервиса предполагает отказ от предварительной закупки большого числа оконечных устройств, содержания больших объёмов информации или вычислительных мощностей. Акцент будет сделан на части программного комплекса, обеспечивающего безопасность данных во время их сбора, передачи, обработки, хранения и реализации. Для этого мы построим модель нарушителя, оценим вероятность реализации тех или иных угроз, выберем самые значимые и выберем способ борьбы, основываясь на нахождении баланса между быстродействием, энергетической эффективностью и стоимостью информации с одной стороны, и стоимостью защитных мер с другой. С учётом почти нулевой стоимости одного блока данных может быть применён нестандартный подход, который может частично или полностью удалять, заменять, искажать или добавлять данные, опираясь исключительно на корректность средних показателей в больших массивах данных, а также их дальнейшего использования для прогнозирования или моделирования на основе этих данных.

Такая система с одной стороны позволит значительно увеличить количество собираемых метеоданных, одновременно снижая их стоимость, а

с другой позволит компаниям (в том числе крупным) концентрироваться на анализе этих данных и разработке алгоритмов их обработки, а не многократно проделывать одну и ту же работу. Кроме того, такой подход за счёт рыночного регулирования стоимости информации позволит привлечь больше мощностей в менее исследованные районы, что, в конечном счёте, позволит делать более точные прогнозы именно там, где это востребовано.

Преимущество данного решения заключается в его масштабируемости на различные сферы человеческой деятельности, включая, например, картографию, сбор медицинской статистики и многое прочее.

Результатом работы станет программное решение, закрывающее наиболее опасные уязвимости (с учётом вероятности реализации угрозы и потенциального ущерба).

1 Обзорно-аналитический раздел

В данном разделе будет рассмотрена система в целом, а также соответствующая литература по защите информации на разных этапах её обработки.

1.1 Описание модели системы

Для большей конкретики подробно опишем модель системы, для чего введём ряд понятий:

Данные – некоторая порция сведений о погоде в том или ином месте (области, регионе) за определённый промежуток времени.

Агрегатор – компания, разрабатывающая программный комплекс по сбору и систематизации Данных, а также их продаже.

Конечный потребитель – физическое или юридическое лицо, которое приобрело Данные.

Датчик – конечное устройство, предназначенное для сбора Данных и дальнейшей передачи.

Владелец Датчика – физическое или юридическое лицо, которое приобрело Датчик и установило на него программный комплекс для сбора Данных, сублицензированный Агрегатором.

Вознаграждение – денежное вознаграждение, которое Агрегатор платит Владельцу Датчика за предоставляемые Данные.

Комиссия – денежное вознаграждение, которое Конечный потребитель выплачивает Агрегатору за доступ к Данным.

Метка – стеганографический отпечаток, накладываемый на Данные перед их продажей. Метка определяет, какому именно Конечному потребителю Агрегатор передал Данные.

Обработка – процесс вычисления дополнительной информации из исходных данных, (например, средние показатели по региону в данный промежуток времени).

Хранилище – распределённая система, предназначенная для хранения Данных, включая результат их Обработки.

Путь информации от её получения со стороны Датчика до продажи Конечному потребителю выглядит следующим образом:

Владелец Датчика устанавливает Датчик, исходя из личных побуждений, таких как удобство (например, поставить Датчик дома), коммерческий интерес (установить датчик на такой локации, Данные на которой высоко оцениваются Агрегатором) или личный интерес (например, мониторинг загрязнения на территории собственного предприятия).

Владелец Датчика не становится автоматически владельцем Данных, полученных его Датчиком. В обмен на Данные Владелец получает Вознаграждение, размер которого зависит от таких факторов, как доверие Датчику (качество измерения), качество Данных (соответствие их действительности – если датчик постоянно нагревается Солнцем, его измерения сильно искажены) и ценность Данных (их уникальность – если на какой-то площади много Датчиков, ценность Данных, полученных ими, снижается). Если Владельца Датчика интересуют Данные, он может их приобрести на общих основаниях. Интерес Агрегатора заключается в марже между Комиссией и Вознаграждением; интерес Владельца Датчика заключается в том, что за небольшую комиссию он получает гораздо более точные данные, чем сырые Данные, сгенерированные его Датчиком.

После сбора данные отправляются на Обработку. Из-за большой ресурсоёмкости эти вычисления будут производиться с использованием облачных технологий – а, значит, на этом этапе надо особенно бережно отнестись к безопасности Данных, так как именно на этом этапе обрабатывается весь их объём.

После обработки данные отправляются в Хранилище – здесь важно позаботиться не только о безопасности этих данных, но и об их доступности и целостности. На данном этапе следует использовать распределённые

системы хранения данных. Перед передачей данных Конечному потребителю на данные ставится Метка, чтобы при обнаружении перепродажи Данных можно было выяснить, кто именно стал вторичным распространителем данных.

Общий жизненный цикл выглядит следующим образом:

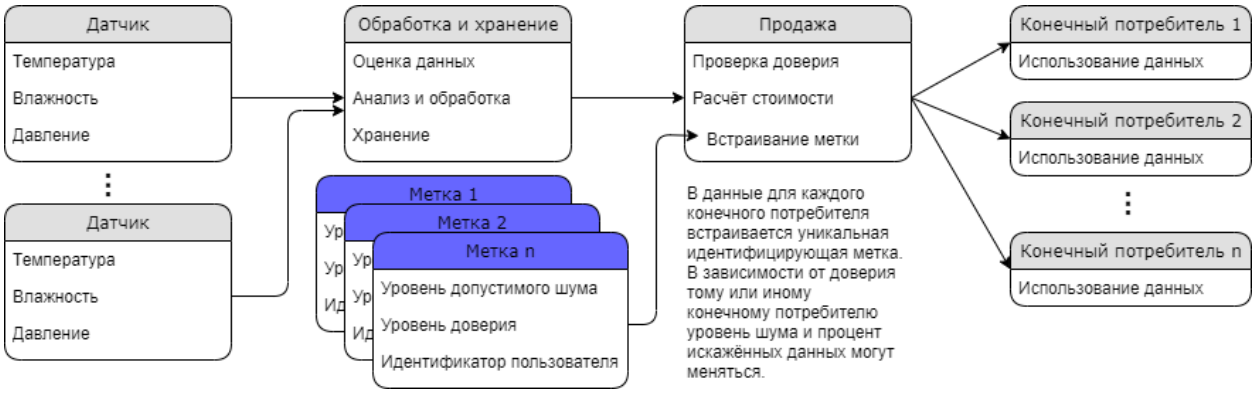

Рисунок 1 – жизненный цикл данных в системе.

1.2 Определение этапов защиты

На начальном этапе требуется защитить Данные, получаемые Датчиком, от перехвата. В первую очередь это решается регулярной криптографией, однако следует помнить о том, что хранимые данные будут доступны не определённому заранее кругу лиц, а также должны обрабатываться без их раскрытия на сторонних серверах. Существует множество способов защиты Данных на этом этапе. Наиболее перспективным направлением можно считать гомоморфное шифрование [1] и перекодирование данных, зашифрованных одним ключом, для другой пары ключей без раскрытия данных и первого ключа в момент перешифрования (proxy re-encryption) [2]. Эти методы на данный момент слабо исследованы; их внедрение требует наложить ограничения на вычисления, производимые над этими Данными. После разработки конкретных алгоритмов оценки и Обработки Данных ожидается переход на новые виды шифрования.

На данный момент следует прибегнуть к стандартным методам криптографии. Основываясь на том, что метод шифрования, используемый

на датчике, должен обладать не только высокой надёжностью, но и энергоэффективностью, стоит обратиться к шифрам-финалистам конкурса AES. Шифр RC6 использует операцию умножения, которая может быть плохо реализована на некоторых аппаратных платформах. Шифр MARS имеет сложную структуру и использует операции умножения 32-битного числа на произвольное число бит. С учётом того, что легальный пользователь будет использовать систему на устройствах без аппаратной поддержки этих операций, применение этих шифров будет неэффективно по времени и энергозатратам.

При рассмотрении оставшихся кандидатов стоит обратить внимание, что шифрование выполняется на машине (Датчике), к которой у злоумышленника есть физический доступ. Это открывает возможность осуществить атаку по побочным каналам.

В Таблице 1 рассмотрим наиболее успешные атаки на каждого оставшегося кандидата.

Таблица 1 – атаки по побочным каналам на шифры-кандидаты AES.

Название	Время	Восстановлено бит ключа	Ссылка
Rijndael	2^{13}	Полное восстановление ключа (128 бит)	[3]
SERPENT	2^{18}	Полное восстановление ключа	[4]
Twofish		96 бит из 128 бит ключа при известном весе Хэмминга ключа, открытом тексте и шифртексте	[5]

Из-за того, что в Twofish используются блоки подстановки (S-box), зависящие от ключа, а также достаточно сложная процедура генерации раундовых ключей (key scheduling), алгебраический анализ с использованием атак по побочным каналам (Algebraic Side-Channel Attack) представляется затруднительным. Стойкость к атакам по побочным каналам, а также

относительная простота реализации, позволяет остановить выбор на Twofish без каких-либо дальнейших модификаций.

При Обработке Данных необходимо предотвратить возможность их дешифрования. Это достаточно легко достигается распределением вычислений и/или нивелируется выбором доверенных сервисов (мощностей). До определения конкретных алгоритмов обработки защита должна производиться организационными мерами.

После продажи данных важно предотвратить вторичную передачу Данных Агрегатору, а также по возможности ограничить их перепродажу. Учитывая, что купленные Данные Конечный потребитель получает в открытом виде (или, что фактически равнозначно, имеет легальную возможность их расшифрования), он может предпринять попытку вторичной продажи полученных данных Агрегатору. Если Агрегатор будет контролировать полное совпадение, злоумышленный Конечный потребитель может внести небольшой шум в Данные. С одной стороны, такие данные пройдут элементарную проверку на вторичность, с другой – будут высоко оценены при анализе соответствия предполагаемым показателям.

Важная проблема заключается в том, что даже без использования специальных вычислительных мощностей Конечный потребитель может выдать себя за Владельца большого числа Датчиков, фактически эмулируя эти Датчики на машине, получающей Данные в режиме он-лайн от Аргегатора. Кроме экономического ущерба, получаемого вследствие выплат фиктивному Владельцу Датчиков, Агрегатор столкнётся с критическим ущербом, вызванным неадекватной оценкой корректности Данных. Фиктивные Датчики будут генерировать Данные, основанные исключительно на прошлой оценке Данных. Это приведёт к достаточно быстрому накоплению ошибки за счёт положительной обратной связи.

1.3 Оценка рисков для системы в целом

После определения ключевых понятий можно перейти к детальному оцениванию рисков.

На каждом этапе будем рассматривать три типа рисков – нелегальное ознакомление (аналогично конфиденциальности), моментальная доступность и чрезмерное искажение (аналогично целостности). При этом риски могут касаться различных объёмов данных. Очевидно, риски более критичны для больших объёмов данных.

Вопросы целостности по большей части относятся к злоумышленным искажениям, так как флуктуации измерений и вычислений свойственны погодным данным.

Стоит отдельно отметить, что отсутствие моментальной доступности является проблемой, но во многих случаях не является критичным фактором. Например, при апостериорной оценке качества информации, а также при уточнении моделей и построении прогнозов, небольшая задержка при получении данных не снижает их ценность.

При выборе системы шифрования особое внимание стоит уделить защите от атак по побочным каналам. Также стоит учитывать, что легальные пользователи будут использовать программное обеспечение, предоставляемое Агрегатором, на устройствах, ограниченных вычислительными мощностями и потреблением электроэнергии. На устройствах, используемых легальными пользователями, не предусматривается аппаратное ускорение шифров, как, например, у Rijndael в микроархитектуре Bulldozer. В то же время злоумышленник может использовать любое аппаратное обеспечение, а, следовательно, основным критерием устойчивости шифра будет являться стоимость взлома, которая должна превышать комиссию, которую Агрегатор уплатит за потенциально сфальсифицированные Данные.

При передаче Данных с датчика на сервера хранения или обработки Данных риск ознакомления с информацией практически полностью отсутствует. Сессионные ключи должны меняться достаточно часто, чтобы стоимость раскрытия одного ключа превышала стоимость информации, зашифрованной данным ключом.

Существует проблема злоумышленного искажения информации перед началом её передачи – даже без ознакомления с Данными их можно контролируемо исказить, если используется блочный шифр в режиме обратной связи по выходу. Впрочем, проблема легко решается с использованием шифров в режиме гаммирования с обратной связью (режим обратной связи по шифртексту, cipher feedback mode, CFB).

Потеря моментальной доступности может быть вызвана перегрузкой каналов в месте нахождения датчиков, что не является темой данной работы.

На данный момент нет конкретного алгоритма анализа, оценки и обработки данных для описываемой системы. По этой причине не представляется возможным оценить ни распределяемость вычислений, ни мощности, требующиеся для них, ни специфические требования для систем шифрования данных (например, в случае использования гомоморфного шифрования, по каким операциям требуется гомоморфизм). Во избежание чрезвычайного усложнения рассматриваемой модели в рамках данной работы будем считать, что вычисления, связанные с анализом и обработкой Данных осуществляются на доверенных мощностях.

На этапе хранения Данных присутствуют все три типа рисков. Если Данные попали в Хранилище, значит, моментальная доступность не является критическим фактором. Целостность Данных обеспечивается грамотной организацией распределённого хранилища. По причине широкой распространённости различных решений такие системы в рамках данной работы отдельно рассматриваться не будут. Вероятность нелегального ознакомления с Данными на этапе хранения достаточно низкая, однако при

реализации угрозы ущерб может быть существенным. Следует бережно относиться к хранению закрытых ключей и позаботиться, чтобы стоимость раскрытия одного ключа была выше стоимости данных, зашифрованных с его помощью. Эти проблемы решаются методами стандартной криптографии, так что не представляют интереса в рамках данной работы.

После продажи данных остаётся проблема нелегального ознакомления с Данными и риск чрезмерного искажения Данных в будущем. Для большей конкретики опишем случай, когда злоумышленник легально приобрёл Данные. Простейший способ извлечь дополнительные выгоды, не предусмотренные Агрегатором, – вторичная продажа этих Данных. Запретить перепродажу Данных фактически невозможно, однако в случае раскрытия предполагаемого факта перепродажи нужен способ достоверного определения источника раскрытия данных с целью прекращения сотрудничества с ним.

Более сложная с технической точки зрения возможность перепродажи основывается на продаже Данных, полученных от Агрегатора, под видом Данных с датчика. В случае перепродажи качество этих Данных будет высоко оценено системой. Учитывая, что погодные данные, полученные с разницей в несколько минут, могут отличаться значительно меньше, чем Данные, получаемые Датчиком и Данные, которые Агрегатор вычисляет с использованием данных с разных датчиков, перепродажа Данных Агрегатора ему же с небольшой задержкой может принести прибыль даже после вычета Комиссии. Более весомой причиной, чтобы бороться с покупкой собственных Данных под видом исходных, является возникающая обратная связь, при которой ошибка, получаемая из-за разницы между реальными погодными данными и Данными, получаемыми с Датчиков, будет достаточно быстро накапливаться. Такой сценарий может нанести системе существенный ущерб, в том числе репутационный, так как поставит под угрозу достоверность любых продаваемых Данных.

1.4 Общая модель нарушителя

Ввиду определённой специфики системы, опишем особенности системы в виде таблицы, в которой, основываясь на сказанном выше, попытаемся оценить наиболее уязвимые места. Оценка разового ущерба основывается на объёме данных; число случаев зависит от числа пользователей, задействованных на данном этапе; вероятность реализации зависит от технической оснащённости злоумышленника.

Таблица 2 – общая модель нарушителя

Этап	**Угроза**	**Уязвимость**	**Ущерб**	**Случаи**	**Реализац**	**Контрмеры**
Сбор данных	Конф.	Атака на шифр	Низкий	Низкое	Низкая	Стандартное шифрование
		Атака по поб. кан.	Низкий	Низкое	Средняя	Шифр без известных атак по поб. каналам
Передача данных	Цел.	Искажение Данных	Низкий	Среднее	Средняя	Шифр в режиме гаммирования
	Дост.	Перегрузка канала связи	Низкий	Высокое	–	Алгор. разрешения конфликтов
Обработка данных	Конф.	Ознак. при обработке	Высокий	Низкое	Неизвест.	Гомоморф. шифр, proxy re-encryption
	Конф.	Раскрытие алгоритма	**Критич.**	Низкое	Неизвест.	Использование доверенных вычислительных мощностей
	Цел.	Искаж. при обработке	Средний	Неизв.	Неизвест.	
		Ошибки обработки	Низкий	Низкое	–	Распределение вычислений
Хранение данных	Конф.	Взлом ключ.	Высокий	Низкое	Низкая	Смена ключей
	Цел.	Потеря данных	Выше среднего	Низкое	–	Использование распределённого хранилища
	Дост.	Отказ в обслуживан.	Средний	Ниже среднего	–	
Продажа данных	Конф.	Перепрод.	Средний	Высокое	Высокая	Применение стеганографии
	Цел.	Продажа Д.	**Критич.**	Среднее	Средняя	

Как видно из таблицы 2, критический ущерб может быть нанесён на этапе обработки и после продажи данных.

1.5 Обзор способов защиты

Ввиду слабой на данный момент распространённости систем, похожих на описанную выше, комплексных решений, предполагающих наличие вероятного злоумышленника на каждом этапе обработки информации, нет. Однако существует множество решений для какой-либо конкретной задачи, частично или полностью соответствующей проблеме, решаемой в рамках рассматриваемой системы.

После анализа модели нарушителя можно сделать вывод, что наиболее критичный ущерб может быть нанесён Агрегатору во время Обработки Данных и в результате нелегитимного использования Данных Конечным потребителем.

Ввиду отсутствия конкретных требований к алгоритмам Обработки Данных нет возможности обоснованно выбрать ту или иную модель гомоморфной системы шифрования и/или распределённых вычислений, а также оценить возможность недопущения декомпиляции алгоритмов обработки. Следовательно, защита Данных на этапе Обработки должна производиться организационными мерами.

При защите информации в момент её сбора и первичной передачи может быть использовано одно из множества существующих решений. Как следует из анализа литературы, одним из наиболее энергоэффективных шифров без известных атак по побочным каналам, способных за разумное время восстановить весь ключ, является Twofish. Во избежание искажения информации без ознакомления с ней следует использовать блочные шифры в режиме гаммирования.

Единственная стадия защиты данных, не имеющая на данный момент готового решения – этап защиты Данных после их продажи Конечному Потребителю. По этой причине основное внимание будет уделено определению принадлежности данных, а именно встраиванию Метки в Данные, передаваемые Конечному потребителю.

На данный момент стеганография широко распространена в области защиты авторских прав на мультимедиа данные (например, фильмы). Специфика контейнеров позволяет вносить метки большого размера, в том числе не влияющие на полезную нагрузку, например, изменяя неиспользуемые байты (dummy bytes).

Кроме этого медиафайлы являются целостным блоком информации, что позволяет внедрять метки большого объёма, заранее зная все характеристики стеганографического контейнера.

К сожалению, такой подход не применим для данных о погоде, поскольку такие данные очень плотно «упакованы», и каждый байт имеет смысловую нагрузку. Более того, данные о погоде передаются небольшими блоками, что не позволяет до внедрения метки собрать статистические свойства контейнера.

Интересной представляется работа [6], предлагающая встраивание метки в реляционные базы данных. В ней рассматривается встраивание фиктивных блоков, позволяющих определить авторство. Предлагаемый подход может использоваться для относительно небольшого количества данных, что необходимо в рамках рассматриваемой системы, так как данные о погоде содержат гораздо меньший объём информации, чем типичные для внедрения стеганографических сообщений контейнеры (например, мультимедиа данные). Эта статья, однако, не предлагает какой-либо реализации, подходящей для использования в описанной системе, и направлена на защиту реляционных баз данных, не предполагающих хранения приблизительно верной информации. Кроме того, алгоритм, рассмотренный в статье, не предполагает возможности искажения настоящих данных. Подход, рассмотренный в данной статье, будет адаптирован к нуждам системы сбора погодных данных, а вставка фиктивных пакетов по необходимости будет рассматриваться как внесение искажений.

1.6 Выводы раздела

На каждом этапе работы с Данными существуют свои риски для Агрегатора. Определена общая модель нарушителя и предложены контрмеры по каждому из этапов. Критический ущерб может быть нанесён Агрегатору на этапах Обработки и после продажи Данных в случае их злоумышленной перепродажи Агрегатору. Для защиты Данных после передачи их Конечному потребителю не существует готовой системы защиты, следовательно, требуются более подробное описание модели атакующего на данном этапе и разработка системы контрмер.

2 Архитектура программного комплекса

В данном разделе мы рассмотрим модели злоумышленника, модели атак, которые он может осуществить, оговорим целесообразность осуществления атак, идею доказательства принадлежности данных Агрегатору, а их цифровой копии – соответствующему Конечному потребителю, и рассмотрим идею алгоритма защиты Данных.

2.1 Модели атак

Рассмотрим следующие модели атакующего:

1. Атакующим является Конечный потребитель: он покупает данные у Агрегатора и, чтобы снизить затраты, понесённые на оплату Комиссии, перепродаёт эти данные третьей стороне.
2. Атакующим является группа из некоторого числа Конечных потребителей и злоумышленников, выдающих себя за Владельцев Датчиков: вместо того, чтобы устанавливать Датчики и передавать Агрегатору Данные, один или несколько Конечных потребителей перепродают Агрегатору его же Данные для многократного получения за них вознаграждения, превышающего сумму уплаченной Комиссии.

В первом случае для установления факта принадлежности Данных Агрегатору, а также выяснения, какому именно Конечному потребителю эти Данные были переданы, достаточно извлечь Метку из Данных. Злоумышленник может попытаться стереть Метку за счёт добавления шума в данные. Возможность коалиции исключается, так как экономически не целесообразно дважды платить Комиссию, для перепродажи Данных по более низкой цене (если цена не будет ниже, чем у Агрегатора, то третья сторона приобретёт данные напрямую у Агрегатора).

Во втором случае появляется возможность создания коалиции – если одни и те же Данные продавать Агрегатору несколько раз, это может окупить

стоимость двойной или тройной Комиссии. Стоит, однако, учитывать, что при большом предложении Данных Агрегатор соразмерно снизит вознаграждение, выплачиваемое Владельцам Датчиков, что не позволит окупить сколь угодно большое число приобретаемых копий Данных.

Кроме этого злоумышленник может попытаться построить регрессию на имеющихся Данных, чтобы заплатить Комиссию за небольшой фрагмент данных, а затем длительное время продавать эти же данные под видом Данных с Датчика. Учитывая, что регрессию можно точно так же построить на собственных данных, проблема борьбы с такого рода фальшивыми данными не может быть решена при помощи встраивания Меток или любого другого способа защиты информации Агрегатором. По этой причине проблему следует рассматривать в рамках создания алгоритма оценивания качества Данных, что не является объектом рассмотрения данной работы.

2.2 Модель шума

Агрегатор оценивает соответствие Данных ожидаемым результатам. Чем больше Данные отличаются от предварительной оценки, тем ниже их качество (а, следовательно, меньше Вознаграждение, выплачиваемое за эти данные), особенно если данные систематически отклоняются в какую-то сторону. Напротив, если Данные идентичны прогнозу, построенному при помощи какой-либо модели (включая модель, используемую самим Агрегатором), это резко снижает доверие к Датчику. Исходя из этого, злоумышленник, скорее всего, будет использовать шум, по распределению близкий к нормальному. Математическое ожидание шума будет нулевым. Дисперсия может быть различной: при нулевой дисперсии шум фактически будет равен нулю, слишком большая дисперсия приведёт к серьёзным искажениям, что значительно снизит качество Данных, а, следовательно, и Вознаграждение.

2.3 Коалиционная модель атакующего

Атакующие могут предпринять попытки избавиться от Метки при помощи усреднения данных, либо выбора одной из версий данных. Учитывая малую стоимость одного фрагмента Данных, Агрегатор может передавать Конечным потребителям массив Данных, в которых некоторое число пакетов вырезано, а небольшая часть – фальшивые. Чтобы бороться с заполнением уничтоженных пакетов или уничтожением фальшивых пакетов, предоставленных Агрегатором, часть уничтоженных, искажённых и/или добавленных пакетов следует делать одинаковой у разных Конечных потребителей. При этом стоит позаботиться, чтобы искажения, вносимые Агрегатором, не снижали ценность данных, одновременно сохраняя устойчивость к попыткам уничтожения Метки.

2.4 Целесообразность атаки

При оценке стратегии, выбираемой злоумышленником, стоит учитывать, что один пакет Данных имеет как низкую себестоимость, так и небольшое Вознаграждение за его передачу Агрегатору. Применение сложных вычислений более затратно, чем честная установка Датчиков, поэтому в стратегиях искажения Данных, поступающих от Агрегатора, злоумышленник будет выбирать вычислительно простые способы.

При наложении шума злоумышленником может использоваться сумма нескольких случайно распределённых случайных чисел в заданном диапазоне. Близость распределения к нормальному будет достигаться за счёт центральной предельной теоремы. Дисперсия будет определяться числом сгенерированных величин.

Если коалиция злоумышленников располагает несколькими версиями одного пакета Данных, выбор версии для передачи Агрегатору должен осуществляться алгоритмом, не требующим существенных вычислительных затрат. Для выбора версии пакета может применяться один из известных

алгоритмов, например, вычисление среднего арифметического, выбор наиболее часто встречающегося (мажоритарная атака), взвешенная мажоритарная атака и др.

Отдельно стоит отметить, что злоумышленник не будет оценивать корректность Данных, поступающих от Агрегатора. В обратном случае ресурсы, затраченные злоумышленником, будут превышать теоретические выгоды от осуществления атаки. Это делает анализ данных Агрегатора нецелесообразным. По этой причине единственное, чем ограничен Агрегатор при выборе искажений – это общая достоверность данных для легального Конечного потребителя.

2.5 Доказательство права собственности

Очевидно, что один пакет Данных не может быть однозначным доказательством ни против одного злоумышленника, ни, тем более, против группы.

При легальном использовании системы следует ожидать отклонение Данных от прогноза как по причине погрешности в измерениях, так и вследствие несоответствия модели фактическим данным. Потеря пакетов Данных также свойственна такого рода системам.

Наиболее подозрительным будет получение от Владельца Датчика некоторого фальшивого пакета, переданного какому-либо Конечному потребителю. Отличить фальшивые Данные от ошибки в измерении может быть проблематично, если Данные искажены не слишком сильно. В таком случае значительно искажённые фальшивые пакеты можно передавать тем Конечным потребителям, которые подозреваются в перепродаже Данных: это позволит обнаружить корреляцию между Данными Агрегатора, передаваемые Конечному потребителю, и Данными, передаваемыми третьей стороне или Агрегатору. Такая стратегия, однако, может не работать для коалиции злоумышленников.

2.6 Алгоритм защиты

Метка должна обладать робастностью – то есть быть устойчивой к уничтожению шумом или другим воздействием. Кроме алгоритма выкалывания и добавления фальшивых пакетов Данных, который пригоден для борьбы с одиночными злоумышленниками, требуется вносить искажения в Данные таким образом, чтобы они не оказывали на Данные большого влияния, но на больших объёмах были заметны. Ввиду борьбы с коалициями искажения не должны интерферировать, чтобы было возможно определить каждого злоумышленника. В то же время искажения, вносимые в каждый пакет, должны быть минимальными, то есть не влиять существенно на качество Данных, передаваемых Конечному потребителю. Другими словами, наличие Метки не должно быть заметно Конечному потребителю – а, значит, следует использовать некоторый шумоподобный сигнал. С учётом описанного выше, наиболее подходящим представляется наложение на Данные некоторой шумоподобной функции. Вносимые искажения не должны существенно влиять на качество Данных, но на большом их объёме должны позволить почти безошибочно определить, содержат ли эти Данные Метку и если да, то какую (какому Конечному Потребителю она принадлежит).

2.7 Выводы раздела

В данном разделе была описана модель атакующего, включая модель шума и модель коалиционной атаки. Выбраны основные направления защиты, основанные на имеющихся предположениях и свойствах системы.

На свойства системы может влиять число злоумышленников в коалиции, а также конкретные параметры шума, выбранные злоумышленником.

При создании системы надо учитывать экономическую целесообразность – как для злоумышленника, так и для Агрегатора.

3 Реализация программного комплекса

3.1 Выбор метки

Опираясь на сказанное в предыдущих разделах, для встраивания метки в данные можно использовать любое семейство ортогональных периодических функций с нулевым математическим ожиданием. Примером такого семейства служит функция Уолша (кроме нулевой функции). Удобство её использования объясняется возможностью в любой момент ввести в систему новых пользователей, для которых соответственно будут созданы новые метки.

Нулевая функция Уолша не может использоваться с целью определения принадлежности цифровой копии Данных, однако может служить для общего определения принадлежности Данных Агрегатору.

Метка внедряется в данные в качестве аддитивного шума с нулевым математическим ожиданием. При описываемом подходе очень просто изменить уровень вносимого шума (с увеличением искажений повышается робастность Метки): сами искажения представляются элементарным произведением функции Уолша в конкретный момент времени на заранее определённую константу. Кроме прочего, такой подход позволяет менять робастность Метки для конкретного пользователя, не разрушая другие свойства системы. В случае коалиционных атак подобное изменение коэффициентов позволяет убедиться в составе коалиции и избежать ложных обвинений.

3.2 Алгоритм внедрения метки

Если злоумышленнику станет известно, в какие пакеты Данных встраивается Метка, то он с лёгкостью избавится от неё. Так, не стоит внедрять данные в каждый пакет: путём несложного перебора злоумышленник, вероятно, сможет обнаружить, какие именно модификации были внесены в определённый блок данных, а ввиду их периодичности с

лёгкостью от них избавиться. Поэтому важно, чтобы Конечному потребителю не было заведомо известно, в какие именно пакеты встраивается Метка. В то же время должен существовать детерминированный алгоритм, позволяющий Агрегатору обнаруживать пакеты, содержащие Метку. Для этих целей разумно использовать генератор псевдослучайной последовательности (ГПСП), в основе которого лежит любая односторонняя функция с лазейкой, например, криптографическая.

Для начала потребуется задать зерно. Таким могут служить, например, координаты, ограничивающие область, данные о которой передаются Конечному потребителю. Далее вычисляется некоторая криптографическая функция от этого зерна с использованием секретного ключа, принадлежащего Агрегатору. Секретный ключ может отличаться для каждого региона, однако это вовсе не является обязательным по причине малого числа шифруемых данных. Несмотря на это, однако, представляется разумным регулярно менять такие ключи. Выход криптографической функции используется как вектор инициализации для шифра в режиме обратной связи по выходу (Output feedback, OFB). Из выработанной гаммы затем последовательно забирается некоторое число бит, определяющее, в какой именно сегмент Данных будет встроена Метка: на N подряд идущих блоков Данных приходится n бит гаммы, где $N = 2^n$.

Стоит обратить внимание на то, чтобы пакет, в который встроена Метка, был корректно определён. Ошибка может возникнуть вследствие задержки, а также в результате попытки злоумышленника несколько перемешать данные, поступающие от Агрегатора. Так, одним из алгоритмов, позволяющих решить эту проблему, является встраивание одного сегмента Метки в несколько подряд идущих пакетов – это же снизит вероятность обнаружения Метки на фоне остальных данных и её устранение, например, усреднением подряд идущих пакетов. Применение такого алгоритма зависит

от таких свойств системы, как допустимая погрешность, качество используемых Датчиков, частота передачи и пр.

Алгоритм встраивания метки может быть описан в виде блок-схемы:

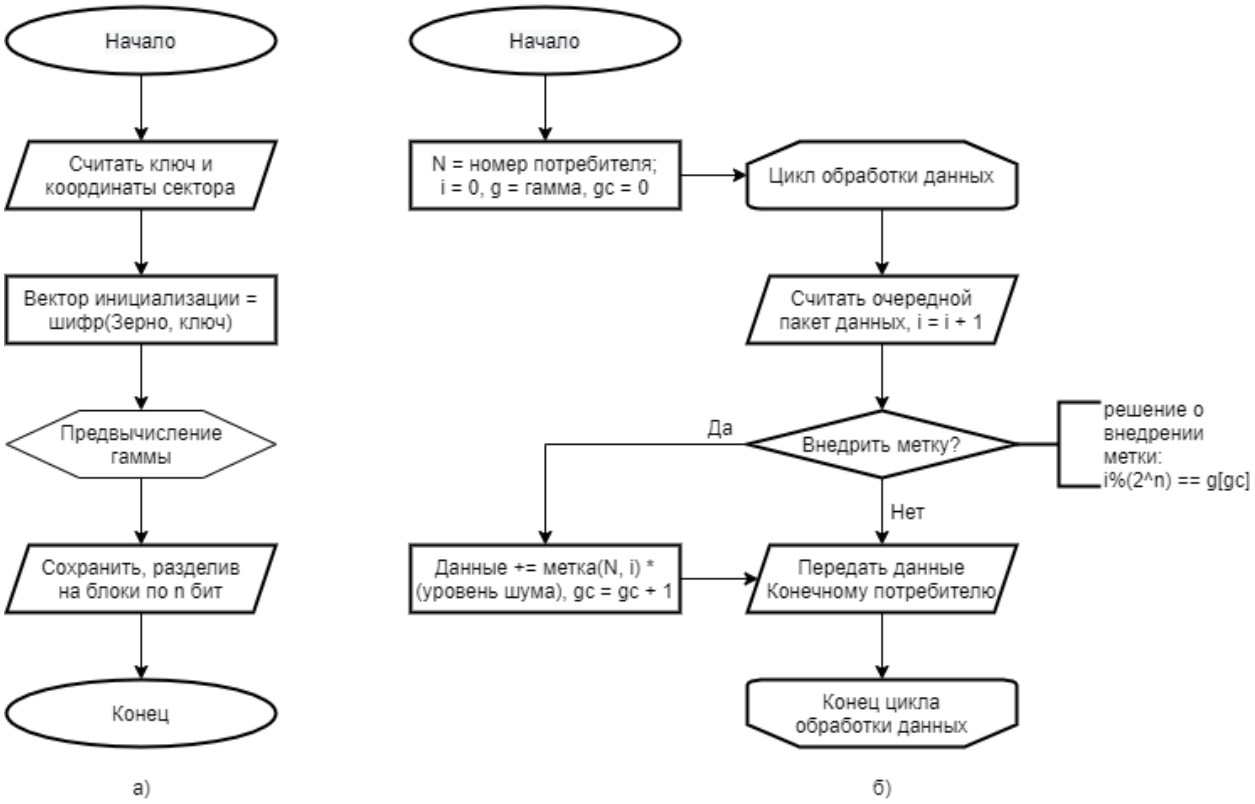

Рисунок 2 – блок-схема алгоритма встраивания метки: вспомогательный алгоритм генерации для выбора искажаемого пакета (а) и алгоритм встраивания метки в пакеты данных для одного пользователя (б)

На данной блок-схеме функция вычисления метки возвращает значение, равное плюс или минус единице, а под уровнем шума (далее – delta) имеется в виду некоторое значение искажения, считающееся допустимым (см. раздел 3.1 – уровень допустимого шума). Решение о внедрении Метки можно принимать различным способом, здесь предлагается использовать предвычисленную гамму.

3.3 Алгоритм извлечения метки

Очевидно, один пакет данных не может адекватно оцениваться на вторичность или, тем более, на принадлежность цифровой копии конкретному пользователю.

Минимальный объём данных, пригодный для оценивания — блок данных, содержащий L искажённых пакетов, где L — длина функции Уолша, используемая в системе.

Длина функции Уолша, используемая в системе, может быть вычислена по следующей формуле:

$$L = 2^{\lfloor log_2 N \rfloor + 1},$$

где L - длина функции, N – число пользователей, $\lfloor\rfloor$ – округление вниз до целого.

Вычисление и использование гаммы аналогично использованию в алгоритме встраивания метки.

Далее под пакетом данных будут подразумеваться исключительно Данные, в которые предположительно внедрена Метка, если обратное не оговорено отдельно.

Алгоритм извлечения метки может быть описан в виде блок-схемы:

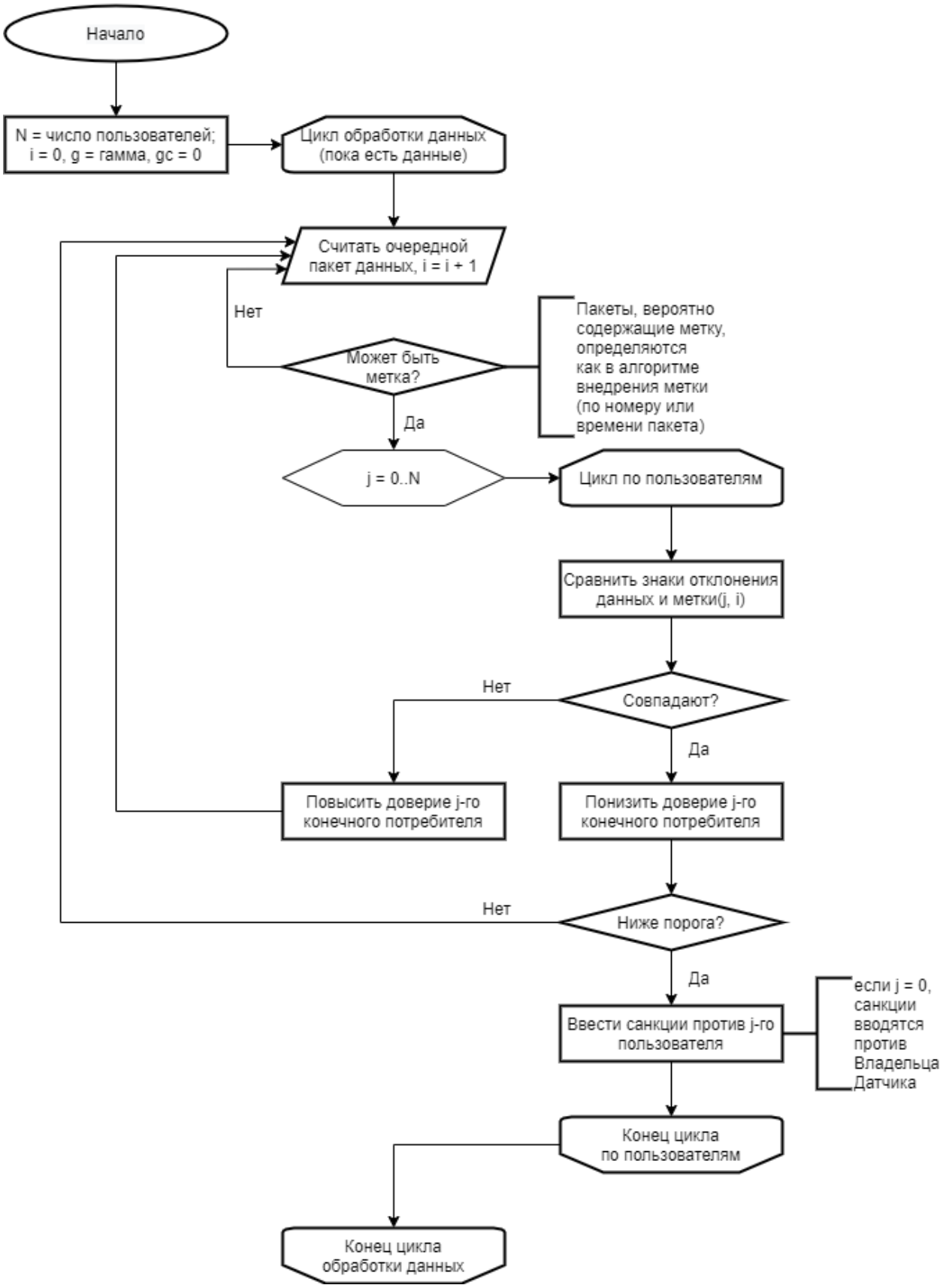

Рисунок 3 – Схема извлечения метки

3.4 Методы оценки Данных

Введём несколько методов оценки данных. Каждый из методов оценки рассматривает Данные с одного Датчика на предмет возможной вторичности, а именно принадлежности одному пользователю или группе пользователей. При этом оценке подлежат как сами Данные, так и доверие Конечным потребителям. Доверие самим Данным выражается через доверие «нулевому пользователю». Недоверие Данным предполагает систематическое накопление в них ошибки, являющейся следствием неудачной установки Датчика (например, в тени или в закрытом помещении) либо вторичности этих Данных.

Моментальное доверие Конечному потребителю (МДКП). В рамках одного пакета Данные могут отличаться от ожидаемых (то есть полученных в результате обработки) Данных в большую или меньшую сторону. Требуется рассмотреть корреляцию такого отклонения и Метки, внедрённой в цифровую копию данных каждого пользователя. Если отличие данных имеет тот же знак, что и моментальное значение функции Уолша для данного пользователя, пакет понижает степень доверия пользователю; в обратном случае повышает. Таким образом, моментальное доверие может быть как положительным, так и отрицательным.

МДКП пользователя u в момент времени i вычисляется по формуле:

$$МДКП_u[i] = sig(u[i] - o[i]) * sig(wal_u[i]),$$

где sig – функция вычисления знака, $u[i]$ – данные пользователя u, $o[i]$ – ожидаемые (являющиеся результатом обработки) данные, а $wal_u[i]$ – значение функции Уолша для пользователя u в момент времени i.

МДКП принимает значения {-1, 1} для каждого пакета данных. По очевидным причинам МДКП не может служить причиной для применения к пользователю санкций.

Удобным инструментом оценки сегмента данных служит *Нормальное доверие Конечному потребителю* (НДКП), которое вычисляется для N

подряд идущих искажённых пакетов. НДКП является минимальной осмысленной единицей измерения доверия к пользователю. На основе НДКП можно вменить Владельцу Датчика небольшие санкции, например, отказать в оплате сегмента Данных, для которых НДКП близок к минус 1.

За НДКП[j] будет обозначаться НДКП для j-го сегмента данных. Каждый сегмент данных содержит N искажённых (помеченных) пакетов.

НДКП пользователя u для сегмента j вычисляется так:

$$НДКП_u[j] = (\sum_{i=j*L}^{(j+1)*L-1} МДКП_u[i]) / L$$

НДКП принимает значения [-1, 1] с дискретом 1/N. В этой и следующих формулах подразумеваются только те пакеты Данных, в которых ожидается наличие Метки. Математическое ожидание НДКП для легального пользователя равно нулю.

Кумулятивное доверие Конечному потребителю (КДКП). Вычисляется как сумма НДКП за некоторый промежуток времени. Если доверие Конечного потребителя опускается ниже некоторого порогового значения, он начинает подвергаться санкциям. Прежде может производиться проверка: повышается робастность Метки, увеличивается плотность искажённых пакетов (вплоть до 100%), Данные могут быть заменены на полностью фиктивные. При обнаружении соответствующих изменений во входных данных блокируется как Конечный потребитель, который перепродавал данные, так и Владелец Датчика, который поставлял их.

КДКП вычисляется по следующей формуле:

$$КДКП_u[q] = \sum_{j=0}^{q} НДКП_u[j]$$

Число q не ограничено, потому КДКП может принимать любые значения. При пересечении некоторого порогового значения пользователь с низким рейтингом КДКП начинает подвергаться санкциям.

3.5 Ошибки первого и второго рода

Рейтинги доверия обладают рядом нетривиальных свойств, характерных при коалиционных атаках.

При определённых обстоятельствах некоторые пользователи могут получить высокую степень доверия, тогда как математическое ожидание кумулятивного доверия легального пользователя равно нулю. Этот артефакт свидетельствует о недостоверности входных данных, но не свидетельствует против пользователя с высоким индексом кумулятивного доверия.

При достаточно большом числе злоумышленников может быть невозможно определить, кто именно продал данные, однако их вторичность всё равно будет определяться. Это выразится в высокой степени недоверия «нулевому пользователю».

Теоретически, при подобранной специальным образом коалиции можно добиться ложного обвинения других пользователей. На практике такая атака маловероятна по ряду причин: кроме того, что она требует значительного числа участников, эти участники должны быть специальным образом выбраны. Подобрать их можно, например, составив некоторую случайную достаточно большую коалицию, а затем включить в неё пользователей, имеющих высокий индекс доверия (см. артефакт, описанный выше). Практической реализации атаки мешает отсутствие данных об индексе доверия у Конечных потребителей, невозможность произвольного включения участников в коалицию, отсутствие данных о соответствии реального пользователя и его номера функции и присвоение пользователям разных номеров для разных потоков данных (температура, влажность и т.д.). Строго говоря, внутри данных даже одного типа можно выделить сколь угодно много потоков.

При некоалиционной атаке обвинение невиновного Конечного потребителя невозможно.

При неудачной установке Датчика Данные могут систематически отличаться в какую-то сторону от прогнозируемых. Это будет выражаться как высокая степень недоверия «нулевому пользователю», однако никакой легальный пользователь, кроме Владельца неудачно установленного Датчика, обвинён не будет. С целью предотвращения интерпретации корректных данных как ошибочных в результате ошибки построения модели, допустимо принимать данные с ненулевым коэффициентом доверия «нулевому пользователю» в том случае, если другие признаки перепродажи данных отсутствуют, а по абсолютному значению эти данные не слишком отличаются от прогнозируемых. Вопрос определения допустимого отклонения лежит на системе оценивания данных, а не их защиты.

3.6 Выводы раздела

В данном разделе были разработаны и описаны алгоритмы защиты информации соответственно моделям и направлениям, описанным в разделе 2. Рассмотрены конкретные меры, принимаемые для защиты, а также параметры, влияющие на свойства системы. Необходимость таких мер проиллюстрирована на примере ряда практических или теоретических атак.

4 Оценка эффективности разработанного программного обеспечения

В этом разделе будут рассмотрены такие параметры, как необходимое число искажённых пакетов для каждого вида атаки и критический уровень искажения, вносимый злоумышленником, приводящий к уничтожению метки.

В вопросе принятия Данных могут рассматриваться ошибки первого (ложное отклонение) и второго (ложное принятие) рода; в вопросе обвинения Конечных потребителей также рассматриваются ошибки первого (ложное обвинение) и второго (необнаружение злоумышленника) рода.

Стоит обратить внимание, что ложное обвинение Конечного потребителя может произойти как в случае реальной атаки (обвинён неправильный пользователь), так и в случае её отсутствия (ложное срабатывание).

4.1 Некоалиционные атаки

В этом подразделе описано поведение системы при проведении атаки одним Конечным потребителем. Список таких атак довольно ограничен, однако они являются наиболее вероятными.

4.1.1 Прямая перепродажа данных

Строго говоря, такая атака гарантированно обнаруживается ровно за один сегмент. Ложное отклонение невозможно. В случае атаки как необнаружение злоумышленника, так и ложное обвинение легального Конечного Потребителя невозможно. Единственная ошибка, которая может возникнуть при обнаружении такого рода атаки – ложное обвинение (и, соответственно, ложное отклонение) в случае отсутствия атаки.

Рассмотрим случай, когда атака не осуществляется. Требуется избежать ложного обвинения, следовательно, надо найти такое число

сегментов, при котором вероятность ложного обвинения будет пренебрежительно мала. Очевидно, что обвинение не зависит ни от силы шума, ни от его характера, так как сравнивается только знак отклонения со знаком фрагмента Метки. В данной модели будем использовать равномерно распределённый аддитивный шум с нулевым математическим ожиданием.

Сравним вероятность ложного срабатывания системы для разного числа пороговых значений и разного числа пакетов данных. (При различном числе пользователей число пакетов в сегменте может отличаться; свойства системы, однако, не меняются от разбиения сегментов по пакетам). Очевидно, что при повышении абсолютного значения порога вероятность ошибки первого рода снижается, но слишком высокий порог может вызывать необнаружение злоумышленника при других атаках.

Минимальное число пакетов равняется двум, что обусловлено периодом самой короткой (первой) функции Уолша. Очевидно, в системах с N пользователями потребуется использование не менее N пакетов.

Таблица 3 – Вероятность ложного срабатывания

Пакет\порог	0.5	0.75	0.875	0.9375	1
2	0.25	0.25	0.25	0.25	0.25
4	0.31	0.06	0.06	0.06	0.06
8	0.15	0.03	4e-3	4e-3	4e-3
16	0.04	2e-3	2e-4	2e-5	2e-5
32	4e-3	1e-5	0	0	0
64	4e-5	0	0	0	0
128	0	0	0	0	0

Кажущиеся аномалии при малом числе пакетов легко объясняются, если обратиться к свойствам МДКП. Действительно, при наличии всего двух пакетов МДКП для них может принимать значения {-1, -1}, {-1, 1}, {1, -1},

{1, 1}. Соответственно ВКДКП принимает значения {-1} с вероятностью 0.25, {0} с вероятностью 0.5 и {+1} с вероятностью 0.25. При четырёх пакетах уже пять вариантов из 16 вызовут срабатывание при вероятности 0.5; при этом четыре из них (три пакета с МДКП {-1} и один с {+1}) ложатся ровно на границу 0.5.

4.1.2 Перепродажа зашумлённых данных

Как говорилось ранее, Данные, систематически отличающиеся от ожидаемых в какую-либо сторону, считаются системой менее достоверными, так как это может свидетельствовать о неудачной установке датчика. Так, злоумышленник будет использовать некоторый аддитивный шум с нулевым математическим ожиданием и некоторой дисперсией. Для простоты будет использоваться взвешенная сумма равномерно распределённых величин (вычисление логарифмов и других математических функций – достаточно дорогая операция, а злоумышленник не заинтересован в расходовании вычислительных мощностей). Согласно центральной предельной теореме (ЦПТ) сумма достаточного числа слабо зависимых случайных величин имеет распределение, близкое к нормальному. Дисперсия равномерного распределения равна $(b-a)^2/12$, где a, b – нижняя и верхняя границы соответственно. Согласно ЦПТ дисперсия среднего арифметического суммы из k равномерно распределённых величин будет равняться $((b-a)^2/12)/k$.

Вероятность ошибки первого рода в плане принятия (отклонения) Данных, как и характеристика работы системы при отсутствии реального осуществления атаки, такая же, как и при обнаружении прямой перепродажи данных. Далее в этом разделе под ошибкой первого рода будет подразумеваться ложное обвинение (вместо злоумышленника обвинён легальный пользователь), а второго рода – отсутствие какого-либо обвинения при осуществляемой атаке.

В действительности вероятность ошибки первого рода не выше, чем в таблице 3, так как добавление метки одного пользователя к данным не делает обвинение другого более вероятным. По этой причине интерес для исследования представляет только ошибка второго рода.

Опираясь на данные таблицы 3, рассматривать число пакетов меньше 16 не имеет смысла из-за высокой вероятности ошибок первого рода. В таблице указывается величина d – отношение дисперсии к delta, где delta – коэффициент, на который помножается функция Уолша при вставке в пакет метки (см. раздел 3.1 – уровень допустимого шума).

Таблица 4 – Ошибки второго рода при пороговом значении 0.5

Пакет\d	0.333	0.375	0.75	1	1.125	1.5	2.667
16	0	1e-3	0.11	0.11	0.14	0.27	0.49
32	0	5e-5	0.07	0.07	0.10	0.26	0.57
48	0	0	0.05	0.05	0.07	0.25	0.62
64	0	0	0.03	0.03	0.05	0.24	0.66
96	0	0	0.01	0.01	0.03	0.22	0.71
128	0	0	4e-3	5e-3	0.01	0.19	0.75
192	0	0	8e-4	1e-3	4e-3	0.16	0.81
256	0	0	2e-4	3e-4	1e-3	0.13	0.85

Можно заметить, что при достаточно большом шуме оценка одного сегмента данных не позволяет с достаточной вероятностью обнаружить атаку. Более того, при шуме, превышающем некоторое пороговое значение, увеличение сегмента не приводит к повышению корректности определения злоумышленника.

Вообще говоря, данные с таким количеством шума можно просто отсекать как непригодные. Однако если такие данные не отсекаются, значит ли это, что система не способна справиться с высоким уровнем вносимого

шума? Для ответа на этот вопрос понадобится более подробно рассмотреть графики доверия Данным.

Для простоты визуального восприятия будет рассмотрен случай с системой, рассчитанной на трёх пользователей, однако все полученные выводы будут корректны и для систем с большим числом пользователей. Рассмотрим случай, описанный в крайнем правом столбце таблицы 4.

На приведённых графиках злоумышленником является пользователь номер 3 (линия со знаком квадрата).

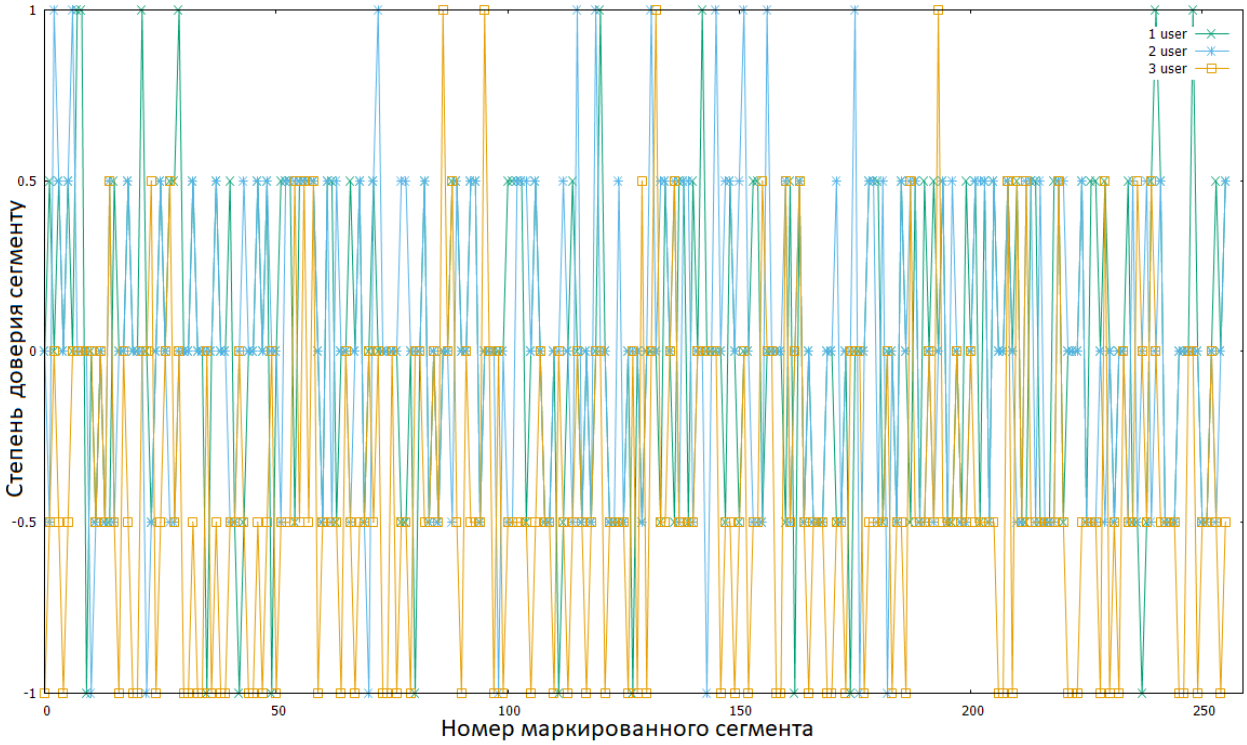

Рисунок 4 – НДКП для системы с тремя пользователями, злоумышленник №3

На этом графике хорошо видно, что пользователь, являющийся злоумышленником, имеет более низкий уровень доверия, чем остальные пользователи. Другими словами, сегменты злоумышленника чаще остальных пользователей имеют отрицательную оценку.

Рассмотрим более подробно оценку системой сегментов конкретно злоумышленного пользователя.

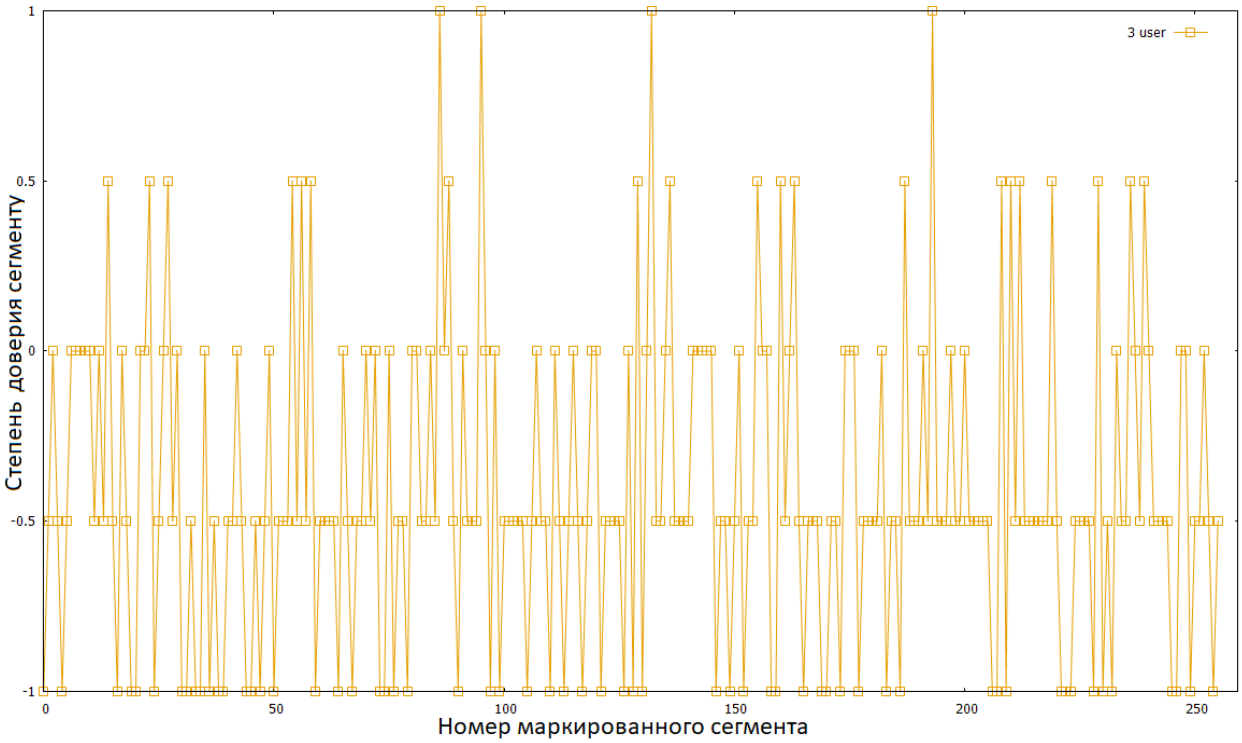

Рисунок 5 – НДКП для злоумышленного пользователя

Видно, что среднее доверие сегменту злоумышленного пользователя систематически находится ниже нуля. Проблема заключается в том, что из-за большой зашумлённости данных многие пакеты находятся выше границы принятия решения о том, что сегмент является недостоверным.

Существует ли формальный способ принять решение о злоумышленности некоторого Конечного потребителя в таком случае? Да, для этого существует КДКП. Этот рейтинг основывается на НДКП большого числа сегментов. Кроме того, на этот рейтинг не влияет пороговое значение НДКП, что позволяет снизить (исключить при некоалиционной атаке) вероятность ошибок второго рода.

Рассмотрим график КДКП для описанного выше случая:

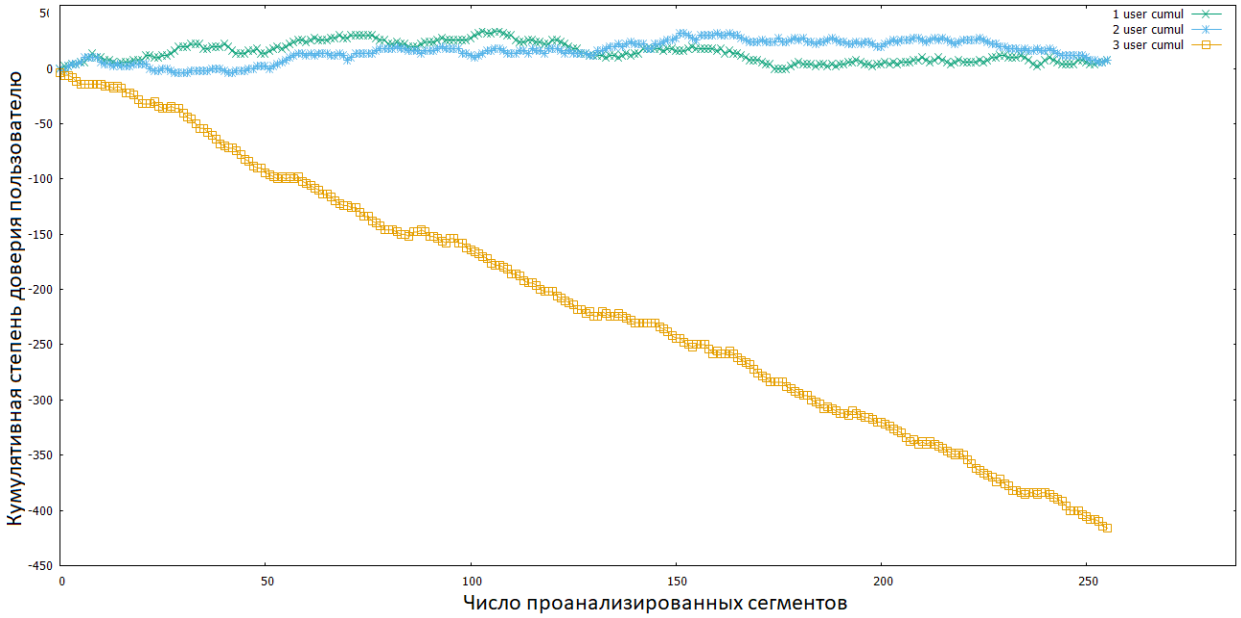

Рисунок 6 – КДКП для системы с тремя пользователями, злоумышленник №3

Когда КДКП становится ниже некоторого значения, к злоумышленнику можно применять санкции: прекратить продажу Данных Конечному потребителю с соответствующим номером, а также прекратить покупку Данных, в которых был обнаружен подлог. До введения санкций можно перепроверить предположение о наличии в Данных Метки, например, увеличив показатель delta.

Пороговое значение для КДКП можно выбрать произвольно, так как злоумышленник рано или поздно будет обнаружен.

Чтобы продемонстрировать способность системы обнаруживать Метку в данных, в которых добавленный шум сколь угодно превышает искажения, внесённые Меткой, рассмотрим систему, в которой уровень силы шума в сто раз превышает уровень силы Метки. Разумеется, в реальной системе такие большие искажения, скорее всего, означают, что погрешность, вносимая шумом, в несколько раз превышает полезную нагрузку, а значит, такие данные будут отклонены.

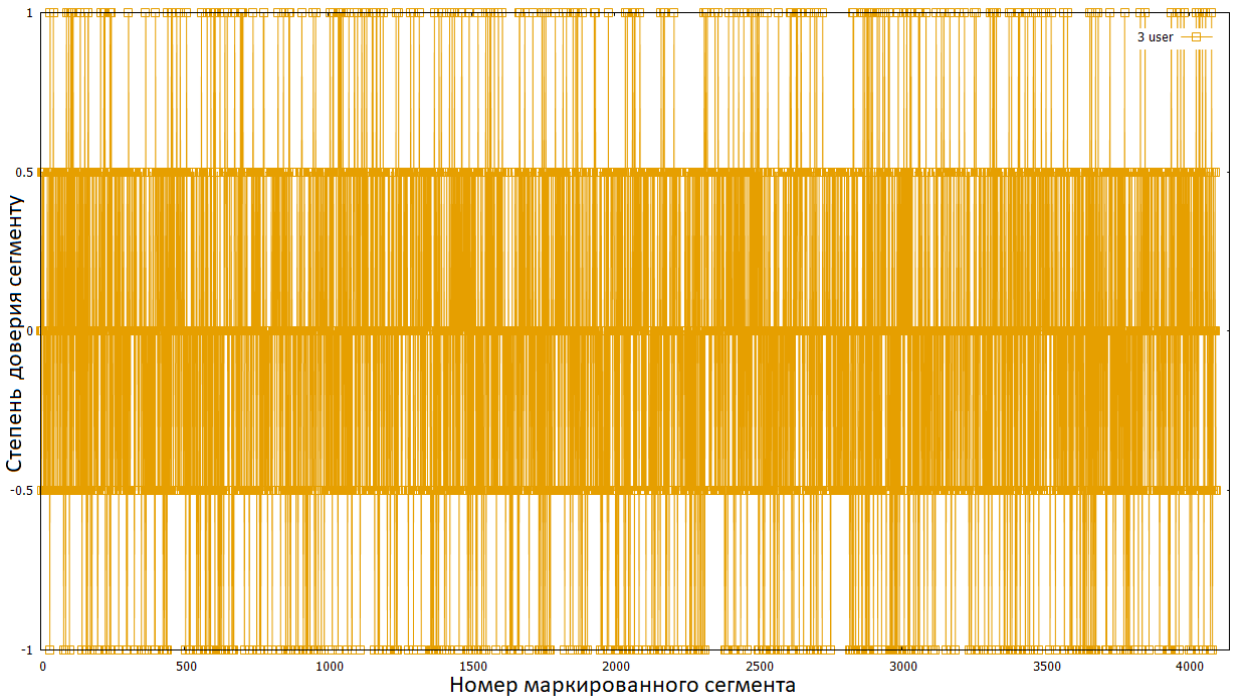

Рисунок 7 – НДКП для помеченных данных с большим шумом

Из данного графика создаётся видимость, что степень доверия злоумышленнику стремится к нулю (как для легального пользователя). Эта проблема была продемонстрирована в таблице 4.

Тем не менее, выявление злоумышленника в условиях сверхвысокой зашумлённости возможно с использованием оценки методом КДКП.

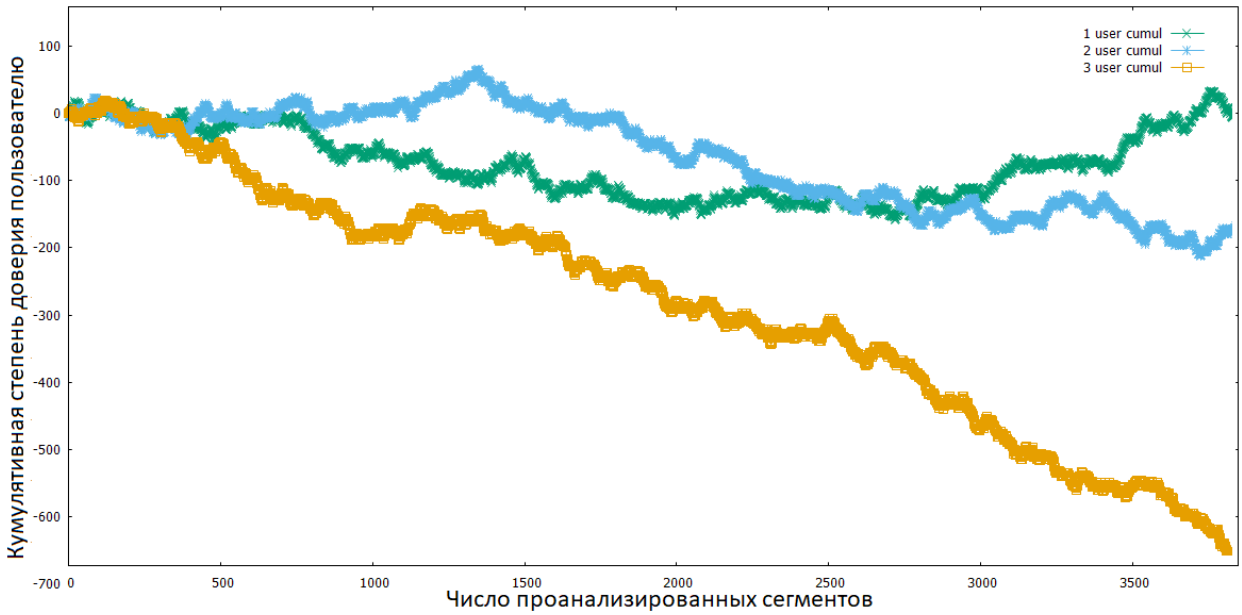

Рисунок 8 – КДКП системы с тремя пользователями, сильный шум, злоумышленник №3

Из графика КДКП наглядно видно, что при рассмотрении достаточного числа сегментов определить принадлежность цифровой копии возможно, даже если шум, добавленный к цифровой копии, настолько велик, что фактически приводит к уничтожению самих данных.

Важно убедиться, что на реальных данных КДКП легальных пользователей не вызывает ложные срабатывания. Рассмотрим график КДКП для Данных, полученных от легальных Владельцев Датчиков:

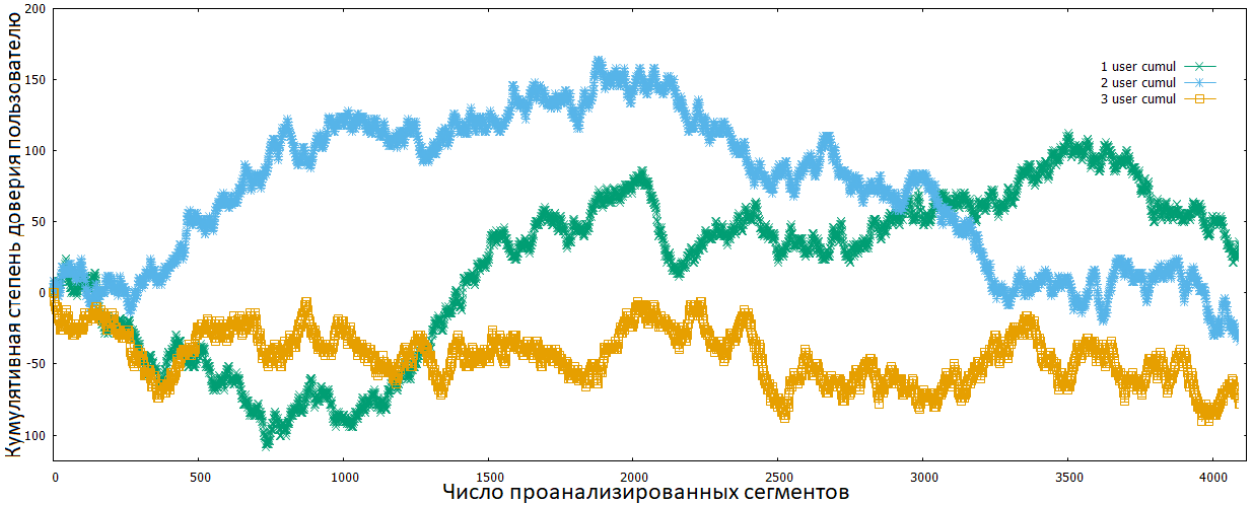

Рисунок 9 – КДКП легальных пользователей для системы с тремя пользователями.

Из графика видно, что КДКП обладает плавающим характером (флуктуации вызваны вероятностными свойствами шума).

Как было наглядно продемонстрировано, один злоумышленник не может осуществить успешную атаку на систему. При грамотно подобранном пороговом значении КДКП вероятность ошибок первого рода стремится к нулю. Обнаружение единичного злоумышленника является вопросом времени: вероятность обнаружения злоумышленника повышается с увеличением числа исследованных сегментов. Так, при увеличении абсолютного значения порога снижается вероятность ошибки первого рода, но увеличивается число необходимых для анализа сегментов.

4.2 Коалиционные атаки

В этом подразделе описано поведение системы при проведении атаки группой Конечных потребителей.

4.2.1 Атака усреднением

Атакующими вычисляется среднеарифметическое всех версий одного пакета. Вычисленное значение выдаётся за Данные с Датчика и передаётся Агрегатору.

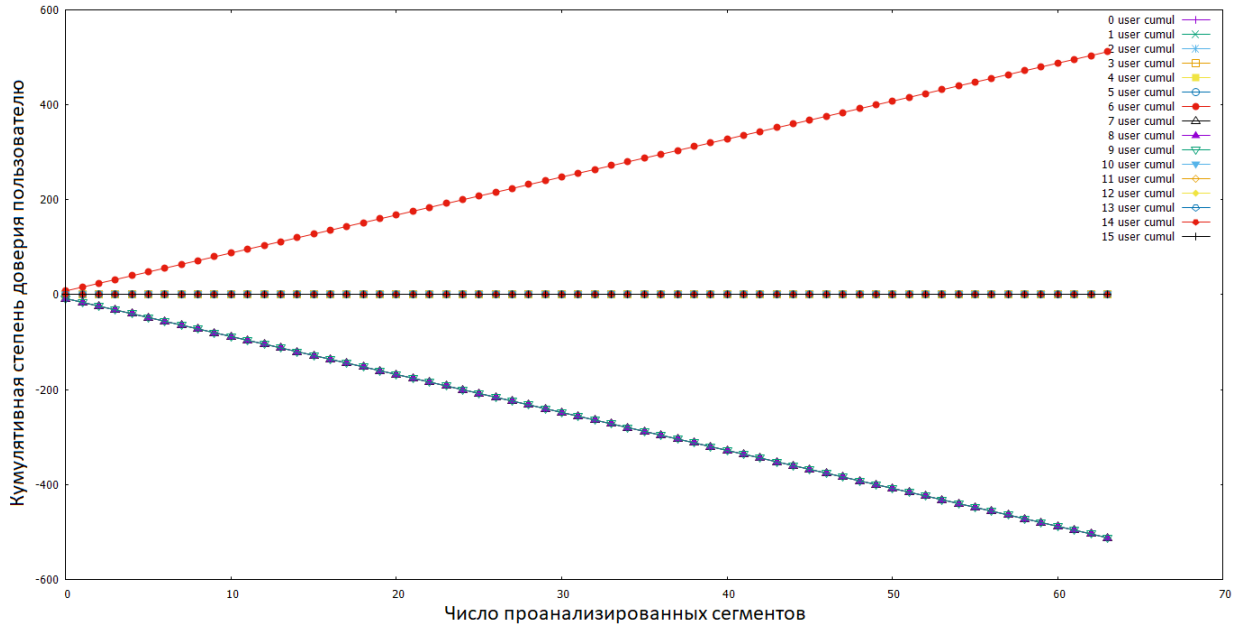

Рисунок 10 – КДКП при атаке усреднением

На рисунке 10 рассмотрена атака со стороны трёх пользователей – с номерами 7, 8, 9

Все три атакующих определены корректно (отрицательная степень доверия). Ни один легальный пользователь не обвинён.

У пользователя с номером 6 наблюдается артефакт в виде высокой степени доверия, он будет рассмотрен позднее.

4.2.2 Мажоритарная атака

За счёт принципа вычисления МДКП мажоритарная атака обладает свойствами, аналогичными атаке усреднением. При мажоритарной атаке Данные отличаются от ожидаемых в большей степени, чем в случае атаки усреднением, следовательно, будут ниже оценены системой оценки Данных.

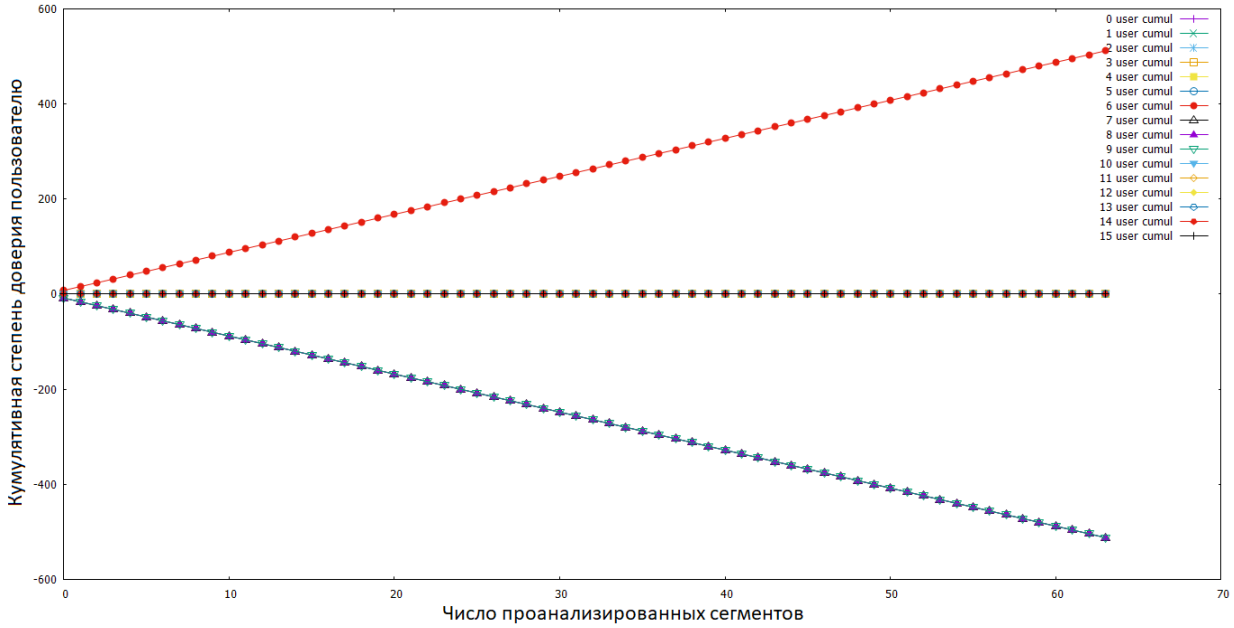

Рисунок 11 – КДКП при мажоритарной атаке

Из рисунков 10 и 11 видно, что поведение системы не различается при атаке усреднением и мажоритарной атаке.

4.2.3 Взвешенная мажоритарная атака

Рассмотрим атаку, в которой пользователь номер 7 имеет вес 0.5, пользователи 8 и 9 – по 0.25.

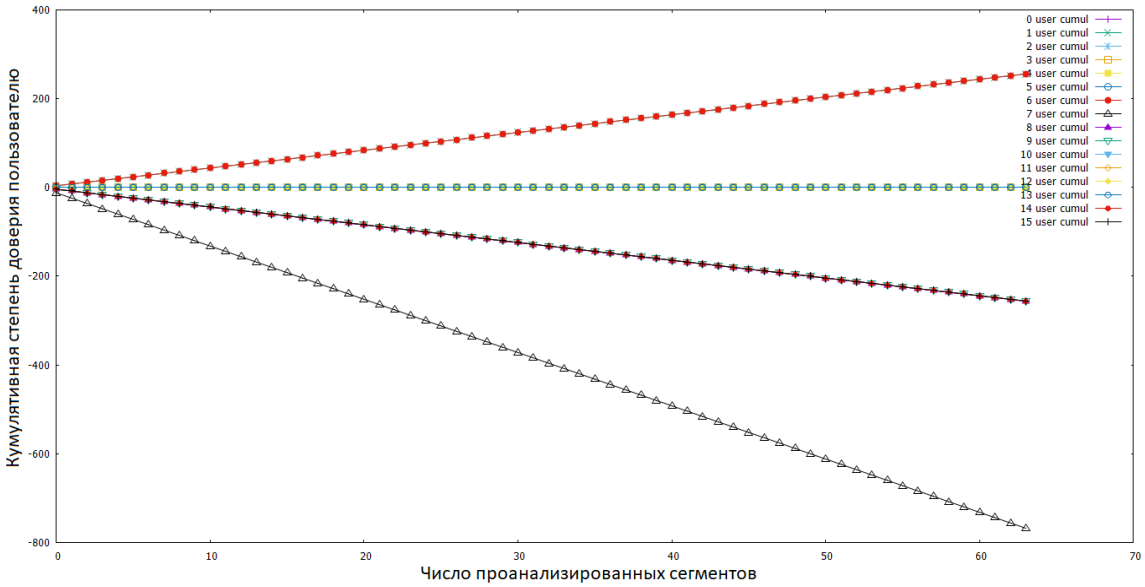

Рисунок 12 – КДКП при взвешенной мажоритарной атаке

Злоумышленник с более высоким вкладом определяется быстрее других. Остальные злоумышленники корректно определяются. Ни один легальный пользователь не обвинён. Артефакты системы сохраняются.

4.2.4 Атака выбором атакующих

Не реализуемая на практике атака, в которой выбор атакующих осуществляется сознательно на основании данных о доверии.

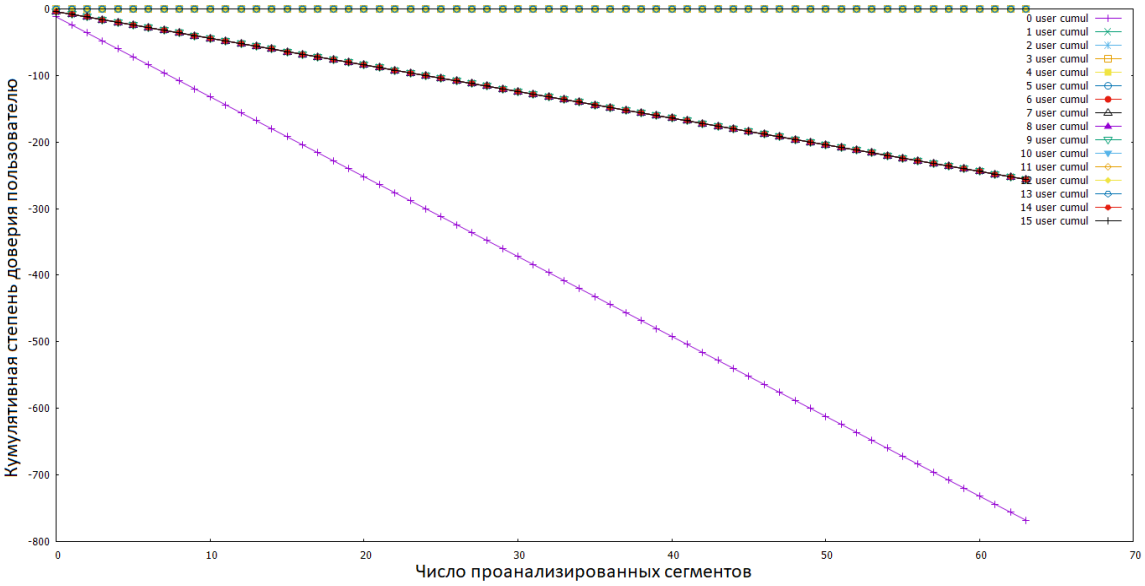

Рисунок 13 – КДКП при атаке с выбором атакующих

Подбор атакующих осуществляется на основании данных о доверии пользователям. Если при некоторой коалиционной атаке доверие определённому пользователю выше нуля, то при включении его в коалицию удастся добиться ложного обвинения легальных пользователей. На рисунке 13 показана атака, произведённая пользователями 6, 7, 8, 9. В результате оценки полученных данных были обвинены все злоумышленники (6, 7, 8, 9), а также ряд легальных пользователей (1, 15, 16). Высокое недоверие фиктивному «нулевому пользователю» свидетельствует о том, что данные содержат набор цифровых копий Данных, приобретённых у Агрегатора.

Такая атака не имеет практической ценности: она не отводит подозрений от злоумышленников, а в силу присвоения пользователям разных номеров в разных потоках данных исключение ложно обвинённых осуществляется тривиально: злоумышленник будет обвинён всюду, а легальный пользователь нет.

4.3 Выводы

В представленной работе рассмотрена система сбора и обработки метеорологических данных. Предложены меры по защите информации на каждом этапе жизненного цикла информации.

Разработана архитектура защитного комплекса.

Реализованы программные средства, защищающие систему от нарушения целостности, приводящего к критическому экономическому и репутационному ущербу.

Подробно рассмотрено поведение системы, методы оценивания данных, а также алгоритмы борьбы с ошибками первого и второго рода.

Заключение

В рамках данной работы были выполнены поставленные задачи:

- произведён обзор литературы;
- представлен общий обзор системы;
- разработана архитектура программного комплекса;
- реализовано программное обеспечение для защиты наиболее уязвимых мест;
- произведена оценка системы;
- даны рекомендации по использованию системы.

Разработанная система позволяет безошибочно определять единичного злоумышленника. Коалиция злоумышленников определяется при рассмотрении большого числа помеченных пакетов с данными. Уровень ошибок первого рода стремится к нулю при корректно подобранных параметрах системы.

Описанная система может применяться в других областях, где требуется стеганографическая защита неспецифичных данных, в которые можно вносить небольшие искажения. Система позволяет внедрять метку в неизвестные заранее данные.

Внедрение метки не требует ни хранения больших объёмов данных, ни ресурсоёмких вычислений.

Список использованных источников

[1] Gentry, C. A Fully Homomorphic Encryption System / C. Gentry, 2009. – URL: https://crypto.stanford.edu/craig/craig-thesis.pdf

[2] Green, M. Identity-Based Proxy Re-encryption / Matthew Green, Giuseppe Ateniese, 2006. – URL: https://eprint.iacr.org/2006/473.pdf

[3] Bonneau, J. Cache-Collision Timing Attacks Against AES / Joseph Bonneau, Ilya Mironov, 2006. – URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2006/10/aes-timing.pdf

[4] Dinur, I. Side Channel Cube Attacks on Block Ciphers / Itai Dinur, Adi Shamir, 2009. – URL: https://eprint.iacr.org/2009/127.pdf

[5] Ma, C. Algebraic Side-Channel Attack on Twofish / Journal of Internet Services and Information Security (JISIS), volume: 7, number: 2 (May 2017), pp. 32-43. – URL: https://pdfs.semanticscholar.org/30f1/f720195b4f6be9b870b6cf9bd9dd1c6e6526.pdf

[6] Chathuranga, M. Watermarking Technology for Copyright Protection of Relational Databases / Mihiran Chathuranga, 2019. – URL: https://www.researchgate.net/publication/331208232_Watermarking_Technology_For_Copyright_Protection_Of_Relational_Databases