**Анонимная сеть «Hidden Lake»**

Коваленко Геннадий Александрович

**Аннотация**. Разработка и проектирование сетей с теоретически доказуемой анонимностью является специфичным, узконаправленным и малоизученным занятием. На данный момент известно лишь две реализации теоретически доказуемых анонимных сетей: Dissent и Herbivore, и обе они созданы на базе проблемы обедающих криптографов (DC-сети). Ранее была известна лишь одна такая проблема, но начиная с 2021 года было открыто ещё две новые проблемы: задача на базе увеличения энтропии (EI-сети) и задача на базе очередей (QB-сети). Последняя представляет собой наибольший интерес в программных реализациях, потому как её архитектурное выражение является наиболее простым и примитивным[[1]](#footnote-2) в сравнении со всеми имеющимися задачами, вследствие чего открывается возможность более лёгкого доказательства в безопасности итогового продукта.

**Ключевые слова**: анонимные сети; децентрализованные сети; теоретически доказуемая анонимность; задача на базе очередей; анонимная сеть «hidden lake»;

**Содержание**

[**1.** Введение 1](#_Toc1)

[**2. Модули** 2](#_Toc2)

[**2.1. Сетевой уровень** 2](#_Toc3)

[**2.2. Анонимизирующий уровень** 3](#_Toc4)

[**2.3. Прикладной уровень** 3](#_Toc5)

[**3. Приложения** 3](#_Toc6)

[**3.1. HLS (сервис)** 3](#_Toc7)

[**3.2. HLT (трафик)** 3](#_Toc8)

[**3.3. HLM (мессенджер)** 4](#_Toc9)

[**4. Композиции** 4](#_Toc10)

[**4.1. HLS+HLM (анонимный мессенджер)** 4](#_Toc11)

[**4.2. HLM+HLT (отложенные сообщения)** 4](#_Toc12)

[**4.3. HLS+HLT (тайные каналы связи)** 4](#_Toc13)

[**5. Заключение** 4](#_Toc14)

[**Список литературы** 4](#_Toc15)

# **1.** Введение

Разработка анонимных сетей является прямым и непосредственным следствием надвигающейся или уже повсеместно существующей централизации. Интенсивное развитие анонимных сетей начинает свидетельствовать лишь о том, что экспансия централизованных систем приближается к своему пределу. Развившаяся до своего предела централизация становится выражением всего множества сетевых коммуникаций, в то время как анонимные сети становятся лишь её придатками [1].

Анонимность, вследствие безукоризненного давления централизацией, начинается разветвляться на несколько возможных уровней. Первым её уровнем становится её же гибель, а точнее смерть её былой видимости в роли симулякра третьего порядка, когда сеть Интернет не была настолько централизованна, и где множество пользователей находились в консенсусе ошибочного мнения о реально существующей анонимности. Вторым же её уровнем становится отрицание действующей централизации, вследствие которого происходит разработка специализированных анонимных сетей с конкретной моделью угроз.

Анонимные сети адаптируются под текущие условия уровня централизации. Если централизация не достигает своего апогея в лице полного сращивания всего возможного множества сетевых коммуникаций, иными словами не обретает свою финальную монопольную форму, то и модель угроз анонимных сетей держится и зиждется на сравнительно низком уровне. Но как только централизация переходит в свою монопольную оболочку, посредством которой она обретает контроль за большим объёмом данных, то анонимные сети начинают подстраиваться под текущие реалии, а их модель угроз начинает быть более строгой и проработанной.

Анонимная сеть Hidden Lake становится одной из таковых форм, придерживающихся строгой модели угроз, в которой предполагается, что монополистическая централизация уже повсеместна была выстроена, а субъекты (участники) самой анонимной сети уже непосредственно находятся в реалиях монополизации сетевых коммуникаций.

# **2. Модули**

Разграничение функций, а также модулей между собой при проектировании анонимных сетей является важным и необходимым качеством в сохранении гибкости и простоты архитектуры. При данном сценарии нужно точно и чётко оценивать границы зон ответственности между одним слоем проектирования и другим.

К сожалению, такое не всегда возможно, когда речь заходит о практической анонимности, в которой одним из важных условий является сохранение высокой пропускной способности трафика. За счёт этого требуется уже непосредственное сращивание нескольких слоёв и уровней проектирования в один монолитный с целью оптимизации и увеличения производительности. Тем не менее, когда речь заходит о теоретически доказуемой анонимности, то сценарии расчёта зон ответственности становятся легче по причине сильного разрыва сетевой и криптографической идентификаций друг от друга, и как следствие, невозможности даже гипотетического сращивания уровней с целью аналогичной оптимизации.

Такой разрыв должен обуславливаться одним или сразу несколькими криптографическими протоколами, переводящими среду передачи информации в абстрактную форму, независимую от законов оригинальной среды. За счёт этого события, становятся неважны такие составляющие как уровень централизации сети, количество участников системы и открытость сетевых адресов [2]. Аномниная сеть Hidden Lake, являясь абстрактной, относится именно к такому виду коммуникаций.

## **2.1. Сетевой уровень**

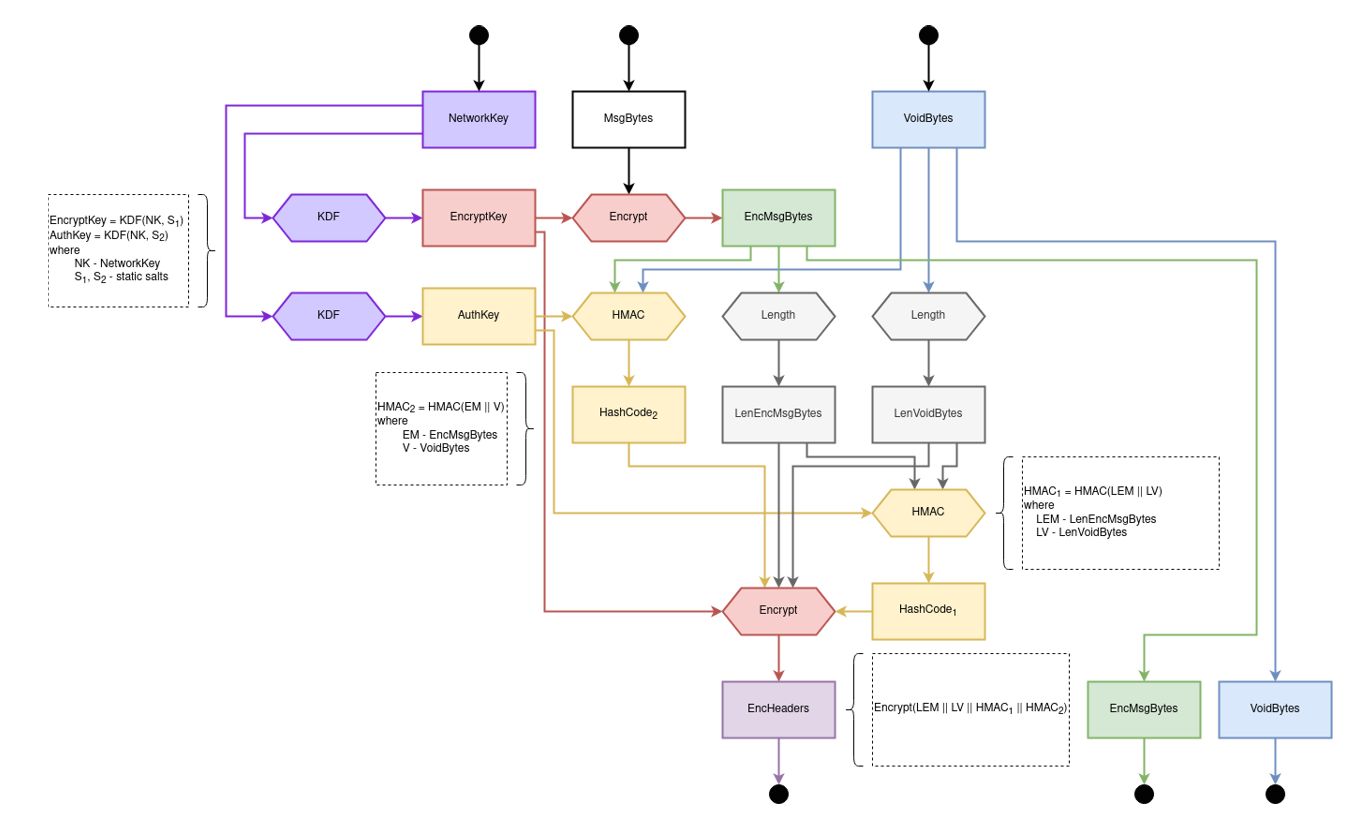
Сетевой уровень анонимной сети Hidden Lake выстраивается за счёт Broadcast коммуникаций, когда любые действия участников сети приводят к широковещательной передаче информации. Данное качество позволяет осуществлять «слепую» (заливочную) маршрутизацию, вследствие которой становится избыточен дополнительный механизм выстраивания коммуникаций от отправителя к получателю. Пакет (информация) в конечном итоге сама начинает находить своего получателя. По умолчанию, вне других уровней, предполагается, что получателем сообщения является вся сеть.

Заливочная маршрутизация безусловно имеет негативные свойства, одним из которых является линейная нагрузка всей сети от количества передаваемых сообщений. Тем не менее, таковая маршрутизация в категориях теоретически доказуемой анонимности имеет ряд положительных сторон. Во-первых, заливочная маршрутизация является простым механизмом транслирования трафика, и как следствие, примитивным, легко переносимым и отказоустойчивым в плане передачи информации. Поверх «слепой» маршрутизации может быть создана совершенно любая логика дальнейшей идентификации субъектов, и как следствие, возможность её использования на более прикладных уровней. Во-вторых, заливочная маршрутизация полезна в анонимных коммуникациях за счёт способности смешивать инициатора соединения (отправителя информации) со всей сетью, вследствие чего появляется сложность отделения / разделения одного участника от оставшийся системы. Данное качество не позволяет расщеплять всю сеть на малые подсети с целью сужения групп участников и их деанонимизации.

Сетевой уровень в своей реализации обладает также шифрованием информации между непосредственно существующими соединениями. Такое шифрование приносит две положительные функции:

1. Расщепление сетей на несколько несращиваемых. Данное свойство полезно, когда необходимо отделять сети друг от друга и не позволять им сращиваться между собой в одну целую. Это может быть использовано в тех случаях, когда одна сеть становится слишком большой, вследствие чего начинает генерироваться огромное количество трафика, нагружающего устройства всех пользователей. Чтобы снизить нагрузку, может быть полезным применить разные ключи сети, чтобы одна сеть не могла получать сообщения из другой и наоборот. В таком случае произойдёт «форк» (разделение) сети с невозможностью последющего «мёржа» (слияния).

2. Шифрованный пакет начинает скрывать размер истинной информации от внешнего глобального наблюдателя посредством случайной (пустой) добавочной величины. Такое свойство становится необходимым, чтобы скрыть принадлежность сети к анонимизации трафика, часто имеющего однотипный размер передаваемой информации. Помимо прочего, само такое шифрование, в отрыве от анонимизирующего механизма, позволяет создавать группы / каналы участников для передачи широковещательной информации, что может быть удобно при кооперациях.



**Рисунок 1.** Шифрование информации на сетевом уровне

Криптографический протокол сетевого уровня может быть изображён на *Рисунке 1* и описан полной функцией действий[[2]](#footnote-3) со стороны отправителя. В данной схеме немаловажным критерием является существование ключа сети и его разделение на два ключа – для шифрования и для аутентификации. Оба данных ключа должны генерироваться разными способами / методами, или одним и тем же с разной солью.

Hidden Lake применяет в качестве KDF алгоритм PBKDF2 с хеш-функцией SHA256. Для шифрования используется AES256-CFB, для аутентификации – HMAC-SHA256. Режим шифрования CFB был выбран с учётом следующих деталей:

1. Режим шифрования должен оставаться блочным и не переходить в свою поточную форму на примере OFB, CTR. Связано это в первую очередь с тем, чтобы у злоумышленника не существовало возможности дешифровывать сообщения основываясь на повторении ранее сгенерированной гаммы. В это же самое время повторённый вектор инициализации может привести к повторению блоков данных, что также является крайне нежелательным свойством, но само по себе оно обладает менее выраженной деструктивной функцией.

2. Режим шифрования не должен создавать дополнительного количества информации, как это работает с режимами ECB, CBC. Данное свойство является нежелательным уже со стороны анонимизирующего уровня, когда генерируемый пакет должен обладать статичной величиной. Тем не менее, удобным является тот фактор, когда одна и та же спецификация алгоритма используется повсеместно, что несомненно будет уменьшать спектр возможных действий злоумышленника.

Применяемые соли при создании ключей аутентификации и шифрования являются статичными и имеют следующий вид (в качестве значения солей были выбраны первые разряды числа пи):

1. Соль для ключа аутентификации:

«1415926535\_8979323846\_2643383279\_5028841971\_6939937510»

2. Соль для ключа шифрования:

«5820974944\_5923078164\_0628620899\_8628034825\_3421170679»

Ключ сети, с учётом существования анонимизирующего уровня, не является сильно секретной информацией и может быть разглашён публично. Такое действие возможно с учётом того, что генерируемый трафик активно не блокируется провайдерами связи. В ином случае, ключ сети должен быть конфиденциальным, чтобы предотвращать анализ размера передаваемых данных и их структуру.

## **2.2. Анонимизирующий уровень**

Анонимизирующий уровень основан на двух составляющих: монолитном криптографическом протоколе [3] и очередях. В то время как первая составляющая создана непосредственно для реконструкции идентификации субъектов на криптографическом уровне, вторая же составляющая ссылается на абстрактно анонимизирующую модель очередей. Данная модель сводится к генерации шифрованных пакетов с периодом в 5 секунд.

## **2.3. Прикладной уровень**

Вg

# **3. Приложения**

Анонимная сеть Hidden Lake не является монолитным приложением, скорее напротив оно является микросервисным, где каждая отдельная задача выносится на выполняющего её представителя. Таким образом, сеть Hidden Lake необходимо рассматривать через призму множества приложений, каждое из которых выполняет свою частную задачу, но в совокупности выполяет общую цель. Hidden Lake можно условно разделить на три типа / класса приложения:

1. Анонимизирующее. К данному типу относится приложение HLS (Hidden Lake Service), выполняющее непосредственно анонимизацию трафика. HLS можно считать ядром анонимной сети Hidden Lake, потому как именно посредством него создаётся криптографическая идентификация пользователей сети.

2. Транслирующее. К данному типу относится приложение HLT (Hidden Lake Traffic), выполняющее роль перераспределения сгенерированного ранее трафика. Вследствие этого, HLT может исполнять две основные функции – ретрансляция и сохранение трафика. В одном сценарии HLT становится лишь обычным ретранслятором зашифрованных сообщений, в другом случае — становится хранилищем шифрованных сообщений.

3. Прикладное. К данному типу относится приложение HLM (Hidden Lake Messenger), выполняющее роль отправления и получения сообщений с графической оболочкой и дизайном обычных мессенджеров. В отличие от других типов приложений, прикладные становятся легко заменяемыми. Иными словами, вместо HLM могут существовать гипотетические HLES (Hidden Lake Email Service), HLF (Hidden Lake Forum), HLFS (Hidden Lake File Service), HLSN (Hidden Lake Social Network) и т.д.

## **3.1. HLS (сервис)**

Вg

## **3.2. HLT (трафик)**

Вg

## **3.3. HLM (мессенджер)**

Вg

# **4. Композиции**

В массе своей, использование Hidden Lake приводит к необходимости запуска сразу нескольких приложений в виде своих композиций. Связано это в первую очередь с микросервисной архитектурой, где каждая отдельная программа является определённого рода сервисов, ограниченным лишь и только своими функциями. Лишь сочитая несколько приложений в одно мы можем достигать определённо лучших результатов.

## **4.1. HLS+HLM (анонимный мессенджер)**

Программы HLS и HLM в отдельности друг от друга мало применимы. HLS лишь генерирует анонимизирующий трафик и даёт возможность применять своё API, но пользоваться им достаточно трудозатратно, если не использовать для этого прикладные приложения. HLM в свою очередь, являясь прикладным приложением, представляет графический интерфейс с возможностью ввода и чтения информации, но при этом не способно отправлять безопасно и анонимно сообщения. Объединяя воедино HLS и HLM мы достигаем положительной композиции, когда мы можем легко использовать HLS и при этом создавать за счёт него анонимные коммуникации в HLM.

## **4.2. HLM+HLT (отложенные сообщения)**

Вg

## **4.3. HLS+HLT (тайные каналы связи)**

Вg

# **5. Заключение**

Вg

# **Список литературы**

1. Коваленко, Г. Теория строения скрытых систем [Электронный ресурс]. — Режим доступа: <https://github.com/number571/go-peer/blob/master/docs/theory_of_the_structure_of_hidden_systems.pdf> (дата обращения: 26.09.2023).
2. Коваленко, Г. Абстрактные анонимные сети [Электронный ресурс]. — Режим доступа: <https://github.com/number571/go-peer/blob/master/docs/abstract_anonymous_networks.pdf> (дата обращения: 26.09.2023).
3. Коваленко, Г. Монолитный криптографический протокол [Электронный ресурс]. — Режим доступа: <https://github.com/number571/go-peer/blob/master/docs/monolithic_cryptographic_protocol.pdf> (дата обращения: 26.09.2023)

1. Существует реализация QB-сети в 200 строк кода на языке Go, представляющая собой приложение-чат: <https://github.com/number571/go-peer/tree/master/cmd/micro_anon>. [↑](#footnote-ref-2)
2. Преобразование входной информации *M* с генерацией пустого сообщения *V* и последующими функциями шифрования *E,* аутентификации *H*.

   *C = E( LEM || LV || H(LEM||LV) || H(EM||V) ) || EM || V,*

   *где LEM = L(EM),*

   *LV = L(V),*

   *EM = E(M),  
    E –* функция шифрования (применяет ключ шифрования)*,*

   *H –* функция аутентификации (применяет ключ аутентификации),

   *L –* функция вычисления длины*,*

   *M –* сообщение*,*

   *V –* пустое сообщение. [↑](#footnote-ref-3)