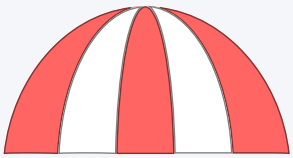
ОТКРЫТЫЙ ДОКУМЕНТ · PUBLIC PAPER



UmbrellaX

Протокол UmbrellaX

Как устроена защита ваших сообщений — простыми словами

ВЕРСИЯ	ДАТА	ЯЗЫК
1.0	10 мая 2026	Русский

АУДИТОРИЯ	АНГЛИЙСКАЯ ВЕРСИЯ
Пользователи, журналисты, юристы, внешние аудиторы	UmbrellaX_protocol_public_en.pdf

Этот документ объясняет простыми словами как работает защита ваших сообщений в UmbrellaX. Без технического жаргона, с живыми примерами и аналогиями. Английская версия — в отдельном файле того же объёма.

СОДЕРЖАНИЕ

01 Добро пожаловать

02 Как работают обычные чаты

03 Секретные чаты

04 Звонки и видео

05 Как защищён ваш профиль и контактная книга

06 Несколько устройств

07 Когда суд требует данные

08 Серьёзный сбой

09 Сравнение с другими мессенджерами

10 Частые вопросы

11 Для внешних аудиторов

12 Модерация — как работают жалобы без лазейки

ГЛАВА ПЕРВАЯ

Добро пожаловать

Пять обещаний, два режима шифрования и принцип «физически не можем» вместо «обещаем не делать».

UmbrellaX — приватный мессенджер, построенный с первого дня так, чтобы ваша приватность была защищена физически, не просто обещаниями.

Наши пять обещаний

1. **Мы не читаем ваши сообщения** — ни ваши личные, ни ваших друзей, ни в публичных каналах.
2. **Мы не сканируем содержимое** — нет PhotoDNA, нет сканирования на вашем устройстве, нет поиска по ключевым словам.
3. **Мы не передаём данные разведкам** — даже «добровольно».
4. **Мы не храним ваш IP-адрес** — ни секунды. Входная защита от массовых атак работает до приложения; в журналах UmbrellaX нет IP, страны или региона пользователя.
5. **Ваш идентификатор — это криптографический ключ**, а не номер телефона или email. Вы для нас — просто уникальный ключ, мы не знаем вашего имени или номера.

Почему это важно

Когда любой другой мессенджер говорит «мы не читаем ваши сообщения» — это обычно обещание на чести. Сотрудники технически могли бы прочитать, но обещают не делать. У UmbrellaX подход другой: наша архитектура устроена так, что **мы физически не можем прочитать даже если захотим**. Это не этика, это физика. В этом документе мы показываем как именно устроена эта физика.

Два режима шифрования

У каждого чата есть один из двух режимов:

- **Обычный (по умолчанию)** — сообщения хранятся на наших серверах зашифрованными особым способом (глава 2). Работает на нескольких устройствах, история синхронизируется, боты работают, группы до 200 тысяч человек, каналы до миллионов.
- **Секретный (по желанию)** — чистое сквозное шифрование. Ключи только на ваших устройствах (глава 3). Для максимальной приватности.

Вы выбираете режим при создании чата. Переключить после нельзя — нужно создать новый чат в другом режиме.

Почему два режима, а не только сквозное

Чистое сквозное шифрование имеет ограничения: не работают боты, не бывает очень больших групп, нет синхронизации между устройствами, если потеряли телефон — потеряли историю. Наш обычный режим решает эти проблемы через систему 5 Сейфов — вы получаете удобство мессенджера с синхронизацией и ботами, и при этом мы физически не можем прочитать ваши сообщения.

ГЛАВА ВТОРАЯ

II

Как работают обычные чаты

Аналогия из Англии 1950-х: запертый ящик у почтальона, ключ разрезан на пять кусочков, пять независимых банковских хранилищ. Вот как устроен наш Cloud-режим.

Представьте такую сцену:

Вы в Англии 1950-х годов. Хотите отправить письмо другу, но боитесь что почтальон прочтёт.

Вы решаете так:

1. Запираете письмо в маленький ящик на ключ.
2. Ключ разрезаете на 5 кусочков. Кладёте в 5 независимых банковских сейфов. Каждый банк держит свой кусочек в хранилище, куда у вас доступа нет — только сам банк может выдать кусочек.
3. Ящик с письмом отправляете через почтальона. Почтальон видит ящик, но не знает как его открыть.
4. Ваш друг хочет прочитать. Он посылает свою подпись и документы 3 из 5 банков. Банки сверяются: «да, это действительно наш друг». Выдают ему 3 кусочка ключа. Трёх достаточно чтобы склеить весь ключ математически.
5. Друг склеивает ключ, открывает ящик, читает письмо.

Кто в этой схеме не может прочитать:

- Почтальон (у него нет ключа).
- Отдельный банк (1 кусочек из 5 — бесполезен).
- Суд (может заставить почтальона передать ящик, но не может заставить 3 независимых банка одновременно выдать кусочки без настоящего друга).
- Правительство (то же что и суд).

Кто может прочитать:

- Вы (создали ключ).
- Ваш друг (через банки).

UmbrellaX устроен аналогично. Только вместо банков — **5 «Сейфов»** (защищённых серверов), вместо почтальона — **«Почтальон»** (наш облачный сервер).

Почтальон — наш облачный сервер

Это обычный сервер, как у любого сайта. Хранит:

- **Зашифрованные сообщения** (никто не может прочитать без ключа).
- **Минимальные метаданные доставки** — идентификатор отправителя A, идентификатор получателя или чата B, дата и время, состояние доставки. Это нужно чтобы сообщение попало к правильному получателю и синхронизировалось между вашими устройствами. IP-адрес, имя, номер телефона, текст сообщения и содержимое вложений туда не входят.
- **Завёрнутые ключи** — ключи от сообщений в зашифрованном виде. Почтальон не умеет их развернуть.

Это обычная наша служба — админы обновляют, масштабируют, ставят заплатки безопасности.

Главное: в обычном облачном режиме у Почтальона лежит только шифротекст сообщения и минимальная запись доставки A → B во времени. Ключ сообщения раскрывается только через Сейфы по правилу **3 из 5**; один сервер, один администратор или один дата-центр не могут превратить шифротекст в открытый текст.

5 Сейфов — наши хранилища ключей

Это 5 независимых защищённых зон. В публичном документе мы сознательно **не раскрываем** провайдеров, страны, регионы, города, дата-центры, количество машин и внутренние маршруты. Такая информация не помогает пользователю проверить криптографию, зато помогает атакующему строить карту нападения.

«Сейф» — это логическая защищённая зона, а не публичная схема размещения. Правило «3 из 5» работает на уровне независимых зон: для сборки ключа нужны 3 разные зоны одновременно. Сколько машин внутри зоны и где они стоят физически — операционная информация, которую нельзя публиковать в открытом документе.

Каждая машина в зоне — **сервер с шифрованием оперативной памяти** (технология AMD SEV-SNP). Даже наши собственные администраторы **не имеют доступа** к содержимому её памяти.

Главный ключ разбит на 5 кусочков математическим способом (схема Шамира), по одному на зону. Для сборки нужно **3 из 5 кусочков**. Ни одна зона не имеет все 5.

Корневой ключ уничтожен и публично подтверждён

После установки мы провели специальную церемонию:

1. Сгенерировали главный ключ на одном компьютере (не подключённом к интернету).
2. Разбили его на 5 кусочков.
3. Отправили каждый кусочек в свой Сейф физически — курьером в запечатанном конверте.
4. **Физически уничтожили** всё что было на том «одном компьютере» — разбили молотком, расщепили диски шредером, уничтожили бумажные копии.
5. Процедуру сняли на видео в присутствии независимого казахстанского нотариуса. Видео выложено публично.

После этого **никто** — ни один основатель, ни один доверенный партнёр, **все они вместе взятые** — не может получить главный ключ. Он разбит на 5 кусочков, каждый Сейф держит свой кусочек внутри защищённого модуля HSM.

Мы называем это **«корневой ключ уничтожен и публично подтверждён»** — админский доступ к главному ключу физически ликвидирован, подтверждение процедуры заверено нотариусом и видео-запись публична. Такого доступа больше не существует.

> «Мы не можем прочитать» — не обещание, а физика. Чтобы получить главный ключ, нужно одновременно скомпрометировать три независимых Сейфа.

Пример: Алиса пишет Бобу

ПРИМЕР

Пусть Алиса в Москве пишет Бобу в Берлин.

Отправка:

1. Телефон Алисы создаёт одноразовый ключ для этого сообщения (32 случайных байта).
2. Телефон шифрует сообщение этим ключом (алгоритмом AES-256).
3. Телефон обращается к 3 Сейфам из 5: «заверните этот ключ для разговора Алисы и Боба». Сейфы (совместно, математически, не видя ключ напрямую) заворачивают ключ.
4. Телефон отправляет Почтальону: зашифрованное сообщение + завёрнутый ключ + служебная информация (от Алисы к Бобу, время отправки).
5. Почтальон хранит это. Ничего не понимает — у него нет способа развернуть ключ.

Получение (Боб открывает чат):

1. Телефон Боба обращается к 3 Сейфам из 5: «я Боб (вот моя цифровая подпись). Дайте ключ для разговора Алисы и меня».
2. Сейфы сверяются с нашим каталогом авторизованных устройств, где хранится список устройств каждого пользователя. «Да, Боб авторизован».
3. Если 3 из 5 Сейфов согласны — они отдают кусочки ключа **напрямую на устройство Боба** (через зашифрованный канал, минуя Почтальона).
4. Телефон Боба собирает 3 кусочка → получает полный одноразовый ключ.
5. Телефон Боба забирает зашифрованное сообщение от Почтальона.
6. Расшифровывает → показывает Бобу.

Открытый текст сообщения никогда не существует на серверах. Только на устройствах Алисы и Боба.

Что происходит при взломе

КОГО ВЗЛОМАЛИ	ЧТО ПОЛУЧИЛ ВЗЛОМЩИК	МОЖЕТ ПРОЧИТАТЬ?
Только Почтальона	Зашифрованные сообщения + завёрнутые ключи	**Нет.** Без Сейфов ключи не развернуть.
1 Сейф	1 из 5 кусочков главного ключа	**Нет.** Нужно 3 кусочка. 1/5 — бесполезно.
2 Сейфа	2 из 5 кусочков	**Нет.** Всё ещё нужно 3.
3 Сейфа одновременно	3 из 5 кусочков	**Потенциально да.** Катастрофическое событие. Warrant canary перестаёт обновляться.
3 Сейфа + Почтальон	3 кусочка + шифротекст	Да. Но очень сложно: нужно одновременно скомпрометировать несколько независимых защищённых зон.
Все 5 Сейфов + Почтальон	Всё	Да. Но это катастрофическая атака на все независимые зоны сразу.

ГЛАВА ТРЕТЬЯ

III

Секретные чаты

Чистое сквозное шифрование через стандарт MLS. Ключи только на ваших устройствах. Для случаев когда даже мы не должны иметь никакой доли ключа.

Секретный режим — для максимальной приватности. Используется международный стандарт **MLS (IETF RFC 9420)** — современный стандарт сквозного шифрования.

Как устроено

Ключи только на ваших устройствах. Нет Сейфов. Нет серверного хранения ключей. Почтальон хранит только зашифрованное содержимое (как в обычном режиме), но **никаких завёрнутых ключей** — их не существует в принципе.

Аналогия: вы и друг обмениваетесь закрытыми ящиками лично, передаёте друг другу ключи при встрече. Письма отправляете через Почтальона (который просто не знает что там). Даже если кто-то взломает все наши Сейфы — секретный чат **не затрагивается**, потому что его ключей у нас никогда не было.

Пример: журналистка Наргиза

ПРИМЕР

Наргиза — журналистка в Казахстане, расследует коррупцию. Получила угрозы. Переживает что её телефон могут захотеть прочитать.

Что ей подходит: секретный режим. Почему:

- В обычном режиме её сообщения хранились бы на нашем сервере в зашифрованном виде. Мы не можем прочитать, но **ключи существуют** в виде кусочков у Сейфов. Если атакующий одновременно скомпрометирует 3 независимых Сейфа (крайне маловероятно, но теоретически) — сообщения можно прочитать.
- В секретном режиме **ключи физически нигде кроме её телефона и телефонов собеседников**. Даже мы не можем расшифровать ни при каких обстоятельствах.

Что она теряет:

- Если потеряет телефон и не создала секретный чат ещё на другом своём устройстве — история потеряна (у нас ключей нет, восстановить не можем).
- Секретный чат не появится на её ноутбуке автоматически. Если хочет секретный чат с тем же собеседником на ноутбуке — придётся создать новый секретный чат прямо с ноутбука.

Характеристики секретного режима

ОСОБЕННОСТЬ	ПОВЕДЕНИЕ
Шифрование	MLS RFC 9420 — чистое сквозное, совершенная секретность вперёд
Ключи	Только на устройствах участников
Синхронизация между устройствами	**Нет.** Ключи не распространяются.
Поиск по истории	Только локально на устройстве где чат создан
Медиафайлы	Зашифрованы сквозным шифрованием
Боты	Не работают в секретных чатах
Максимум участников	~1000 (практический предел MLS)
Исчезающие сообщения	Да, таймер от 1 часа до 1 недели
Защита от снимков экрана	Усиленная (уведомление отправителю если получатель сделал снимок)

Когда использовать секретный режим

- **Журналисты** в СНГ, Иране, Китае — общение с источниками.
- **Активисты** — координация в странах с жёстким наблюдением.
- **Юристы** — привилегия адвоката с клиентом.
- **Врачи** — медицинская информация о пациентах.
- **Супруги и пары** — интимные разговоры.

Важно понимать

- **Если потеряли телефон** с секретным чатом — содержимое **невосстановимо**. Резервной копии не будет (мы физически не имеем ключей).
- **Если хотите секретный чат на втором устройстве** — нужно создать **новый** секретный чат с того второго устройства.
- **Снимок экрана** в секретном чате физически возможен (мы не контролируем камеры устройств), но отправитель получает уведомление что его сделали.

Сравнение обычного и секретного режима

ПАРАМЕТР	ОБЫЧНЫЙ (CLOUD)	СЕКРЕТНЫЙ
Хранение сообщений	Почтальон (зашифровано)	Почтальон (зашифровано)
Хранение ключей	5 Сейфов (3 из 5 по Шамиру)	Только на устройствах
Могу прочитать я	Да, через свои устройства	Да, на устройстве
Можем прочитать мы	**Нет** (корневой ключ уничтожен)	**Нет** (ключей у нас нет никогда)
Мульти-устройство	Да	Нет
История на новом устройстве	Да	Нет (только на том где создан)
Боты	Да	Нет
Большие группы	До 200 000	До ~1000
Потеря телефона	24 слова; при полной потере устройств — 24+12 слов	Секретный чат потерян

ГЛАВА ЧЕТВЁРТАЯ

IV

Звонки и видео

Всегда сквозные. Три пути связи с автовыбором: напрямую, через наш ретранслятор, облачный запасной для заблокированных стран.

Все голосовые и видео звонки — всегда сквозные. Даже для пользователей в странах где обычный WebRTC-звонок заблокирован (Россия, Иран, Китай).

По умолчанию — через ретранслятор (IP собеседника скрыт)

Когда два устройства соединяются напрямую через интернет для звонка, им нужно **знать адреса друг друга** — это свойство прямого соединения. Шифрование при этом не ломается (никто не слышит разговор), но сам **IP-адрес собеседника** ему виден. IP раскрывает примерное местоположение (страну, город, провайдера), поэтому для приватности мы по умолчанию **не используем прямое соединение**.

Дефолтный режим — **через ретранслятор** (путь 2 ниже). Задержка чуть больше (+50-100 мс), но IP не раскрывается. Режим «быстрые прямые звонки» (путь 1) можно включить в настройках с явным предупреждением о раскрытии IP. Для особо чувствительных звонков есть режим **«максимальная приватность»** (цепочка из двух независимых ретрансляторов, см. ниже).

Три способа связи

UmbrellaX пробует последовательно:

Путь 1: Напрямую между устройствами (предпочтительный)

Ваше устройство связывается с устройством собеседника **напрямую через интернет**. Ключи только у вас двоих. Никто посередине.

- **Работает:** свободные страны (без глубокой фильтрации).
- **Задержка:** 50-150 миллисекунд (ощущается как очный разговор).

Путь 2: Через наш ретранслятор

Если напрямую не получилось (например, ваш провайдер использует сложный NAT), трафик идёт через сервис `coturn` — наш ретранслятор. Он **передаёт зашифрованные пакеты, не зная что внутри**. Ключи всё ещё только у вас и собеседника.

- **Работает:** почти везде.
- **Задержка:** 100-250 миллисекунд (заметно, но терпимо).

Путь 3: Облачный запасной (для заблокированных стран)

Если WebRTC **полностью заблокирован** (великий файерволл в Китае, Роскомнадзор в России), ваш клиент отправляет **зашифрованные аудио/видео пакеты через обычный HTTPS** на наш облачный ретранслятор звонков. Он **передаёт зашифрованные пакеты**, вы и собеседник расшифровываете их у себя.

- **Работает:** везде где работает HTTPS.
- **Задержка:** 200-500 миллисекунд (небольшой лаг).

Пример: активист Саид в Иране

ПРИМЕР

Саид — активист в Иране. Обычный WebRTC у него заблокирован на уровне интернет-провайдера. Он хочет позвонить маме в Турцию.

Что происходит:

1. Саид открывает чат с мамой в UmbrellaX, нажимает «Позвонить».
2. Его клиент пробует Путь 1 (напрямую). **Не получается** — провайдер блокирует.
3. Его клиент пробует Путь 2 (через ретранслятор). **Тоже не получается** — сигнальные пакеты блокируются.
4. Его клиент переходит на Путь 3 (облачный запасной). Зашифрованный голос упаковывается в обычный HTTPS-поток и идёт на наш облачный ретранслятор звонков.
5. Мама Саида получает вызов в Турции, отвечает.
6. **Разговор идёт.** Задержка 300-400 миллисекунд (чуть больше чем в обычном звонке, но разговор слышен нормально).
7. **Наш сервер видит только зашифрованный голос** — ключи только у Саида и мамы.

В Telegram такой звонок бы не получился — они не имеют облачного запасного. В UmbrellaX — работает.

Автовыбор пути

Клиент пробует по очереди:

1. Путь 1 (напрямую) — 3 секунды на попытку.

2. Путь 2 (ретранслятор) — 5 секунд.
3. Путь 3 (облачный запасной) — всегда работает.

Итого: звонок начинается за 3-13 секунд в худшем случае. Обычно за секунду (первый путь срабатывает).

Можно ли заставить использовать только один путь

Да. В настройках:

- **Автоматически** (по умолчанию) — 3 пути как выше.
- **Только сквозной через интернет** — отключает облачный запасной. Для параноидальных.
- **Всегда через облако** — сразу в облачный запасной. Полезно в Китае/России чтобы не ждать 8 секунд на попытки.

Исключение — большие голосовые чаты (>32 участников)

Физика: нельзя смешать звук более 32 человек без расшифровки. Большие голосовые чаты и прямые трансляции требуют серверного микширования. В интерфейсе видно предупреждение: «Для сквозного шифрования используйте 1-на-1 или группу до 32».

Всё остальное (1-на-1, группы от 2 до 32 человек) — **всегда сквозное**. Сервер никогда не слышит голос.

Группы от 2 до 32 участников

- Ключ группы MLS обновляется при добавлении или удалении участника (это гарантия «после выхода не слышно»).
- Сервер SFU (Selective Forwarding Unit — избирательный пересылатель) видит только зашифрованные кадры, не может раскодировать.
- SFrame (стандарт IETF) шифрует каждый кадр отдельно.
- **IP участников автоматически скрыт**: в групповых звонках участники подключаются только к SFU, не друг к другу. SFU видит их IP (работает в защищённой зоне без логов), но сами участники IP друг друга никогда не видят — это архитектурное свойство SFU.

Режим «максимальная приватность звонка»

Для журналистов, активистов, юристов и других кому IP собеседника критично скрыть даже от своего же ретранслятора — опция **«Максимальная приватность звонка»** в настройках. При её включении звонок идёт через **два независимых ретранслятора цепочкой**. Первый ретранслятор знает только «звонок от А идёт во второй», второй — «звонок идёт в Б». Ни один из двух не знает полной связки «А звонит Б».

Задержка при этом режиме +100-200 мс поверх обычного ретранслятора (итого ~200-400 мс) — для голоса всё ещё приемлемо. Опция включается глобально или **per-chat** (для

конкретного контакта можно отметить «sensitive contact» — для такого контакта всегда используется двойной ретранслятор, независимо от глобальной настройки).

Во время звонка иконка в интерфейсе показывает текущий режим: через один ретранслятор, через два, или (редко) прямое соединение.

ГЛАВА ПЯТАЯ

V

Как защищён ваш профиль и контактная книга

Ваше видимое имя, аватар и телефон собеседника — последние слабые места после шифрования сообщений и звонков. Здесь мы закрываем и их.

Защита сообщений и звонков — это здорово, но представьте: вы общаетесь с журналистом под псевдонимом, переписка зашифрована, звонок через ретранслятор — а в вашем аккаунте стоит настоящее имя «Иван Петров» и фото вашего лица. Или ваш телефон привязан, и кто-то проверяет «есть такой юзер с этим номером?». Приватность сообщений не помогает если сама ваша личность видна. Поэтому защищаем и это.

Ваш профиль — запечатанная визитка

Визитка — это ваше видимое имя + аватар + статус, то что показывается другим людям когда вы с ними общаетесь.

Наша модель не допускает открытого хранения профиля у нас. Мы не видим, какое имя вы поставили, и у нас нет открытого профиля, который можно было бы кому-то выдать.

Как работает сейчас:

Ваше устройство при создании аккаунта генерирует **личный ключик для визитки** (32 случайных байта). Этот ключик хранится на вашем телефоне вместе с основным ключом аккаунта, восстанавливается теми же 24 словами seed-фразы; при полной потере всех устройств используется дополнительный 12-словный код восстановления для безопасной ротации личности.

Ваша визитка (имя + фото + статус) **заворачивается в конверт** вашим ключиком-от-визитки. Конверт мы называем «запечатанным». Вот его мы и храним в нашем справочнике. Открыть не можем — ключика у нас нет.

Как ваши друзья видят вашу визитку:

Когда вы с Машей становитесь контактами (она вас добавила, или вы вместе в чате, или приняла ваш инвайт) — ваш телефон отправляет Маше копию вашего ключика-от-визитки. Не открыто — а завернув его в ещё один конверт, который может открыть только телефон Маши её собственным ключом аккаунта. Сервер видит только двойной конверт с мусором, сам открыть не может.

Теперь устройство Маши:

1. Скачивает из справочника ваш запечатанный конверт с визиткой.
2. Открывает сохранённым у себя ключиком-от-вашей-визитки.
3. Показывает Маше «Иван» и фото кота.

Когда вы меняете аватар или имя — просто заворачиваете новую визитку тем же старым ключиком. Все ваши контакты автоматически видят обновление при следующем открытии чата.

Работает одинаково в обычных и секретных чатах — потому что визитка живёт отдельно от сообщений. Различается только **способ доставки ключика новому контакту**: в обычных чатах — через отдельный защищённый канал, в секретных — прямо внутри MLS-протокола чата.

> Даже если вы поставите «Иван Петров» — это имя есть только у вас на телефоне и у контактов которых вы сами выбрали. У нас — только запечатанный конверт.

Публичный профиль — для тех кто хочет быть видимым

Иногда профиль хочется показывать всем: блогеры, журналисты с узнаваемым именем, бизнесы, каналы. Для них — **опция «Публичный профиль»** в настройках. По умолчанию выключена.

Включая её, юзер подтверждает: «я хочу чтобы моё имя и аватар видели все, включая тех кто может использовать это против меня». При нажатии приложение показывает жёсткое предупреждение с двумя галочками подтверждения.

Технически у публичного юзера **две визитки**: приватная (запечатанная, для контактов — там может быть более личное имя и фото) и публичная (открытая копия в отдельном публичном справочнике — там то что юзер хочет показать миру). Юзер сам решает что куда.

Каналы и боты — всегда публичные. У них нет приватной визитки по природе (канал это публичная сущность, бот — публичная услуга).

Контактная книга — как мы не знаем ваши телефоны

Если вы прикрепили телефон для того чтобы друзья могли вас найти — мы вас **не хотим знать** и сделали чтобы не могли.

Какую простую схему мы не используем: телефон считает отпечаток номера и отправляет его серверу. Такая схема опасна: если кто-то приносит конкретный номер, сервер может повторить расчёт и проверить базу. Поэтому в UmbrellaX так не сделано.

Как сейчас:

Ваш номер превращается в **магическую метку** через специальный математический танец с Сейфами:

1. Устройство делает из номера **запутанный вопрос** (математически шифрует номер случайным числом). Вопрос выглядит как случайный шум, из него номер не восстановить.
2. Вопрос отправляется на наш сервер, который передаёт его **трём из пяти Сейфов**. Три Сейфа вместе применяют к вопросу **секретную математическую печать** — ключ к этой печати тоже разделён на 5 долек по Сейфам, ни один Сейф в одиночку её не знает. Наш сервер посредник ничего не понимает — просто передаёт туда-сюда.
3. Запутанный результат возвращается вашему устройству. Оно **распутывает на своей стороне**, получает финальную **магическую метку** — уникальную короткую строчку для этого номера.
4. Метку отправляет нам. Мы храним: «метка `7e93f4...` = ваш аккаунт». Номер не знаем, отпечатка не знаем — только метку.

Ключевая особенность: превратить номер в метку **без Сейфов никто не может**. Сейфы соглашаются применить печать **только по запросу настоящего приложения на настоящем телефоне** — с подтверждением от Apple или Google что это реальное устройство, а не скрипт. Без этого отказ.

Поиск друзей работает так же как в других мессенджерах:

Устройство Пети для каждого номера из его телефонной книжки делает тот же танец — получает пачку меток. Отправляет нам. Мы сверяем со справочником, отвечаем: «метка №3 = юзер Маша». Устройство Пети сопоставляет со своей локальной книжкой («метка №3 была для номера Маши Ивановны → значит юзер Маша»). Показывает уведомление «Маша теперь в UmbrellaX». Имя «Маша Ивановна» взято из собственной телефонной книжки Пети — мы его не знаем.

Пример: журналистка Наргиза под повесткой

ПРИМЕР

К нам приходит запрос суда: «Есть ли у вас юзер с номером +7 777 123 45 67, и каково его настоящее имя?»

Наш ответ:

1. **Про номер:** «Ваша честь, мы не храним номера. Мы храним только магические метки, которые генерируются Сейфами для зарегистрированных устройств. Чтобы превратить ваш номер в метку, нужно попросить Сейфы его запечатать — но Сейфы выполняют эту операцию только для запросов от физического устройства конкретного юзера с подтверждением подлинности от Apple или Google. У нас такого устройства нет. Имитировать мы его не можем — подписи подлинности производим не мы, а Apple и Google, каждая привязана к конкретному железу. Поэтому физически дать ответ "да/нет, этот номер у нас есть" мы не можем.»
2. **Про имя:** «Видимое имя хранится только в виде запечатанного конверта. Ключик от конверта — у самой юзерки и у выбранных ею контактов. У нас ключика нет и не было. Открытого имени у нас нет, поэтому раскрывать нечего.»

Что мы не можем выдать:

- Номер телефона — мы его не храним.
- Настоящее имя — оно лежит только на устройствах пользователя и выбранных контактов.
- IP-адрес — не храним.
- Содержание сообщений — не имеем ключей для чтения.

Если у пользователя есть публичный псевдоним или публичный профиль, это уже видно всем без запроса к нам. По непубличным данным ответ один: у нас нет таких данных для передачи.

Что остаётся уязвимым — честно

1. **Физический доступ к разблокированному устройству.** Если суд захватил её разблокированный телефон — они сами откроют приложение и увидят настоящее имя. Мы ничего дополнительного не дадим. Это присущий компромисс: защищаем от удалённых запросов, не защищаем от физических.
2. **Массовый сбор невозможен.** «Дайте всех юзеров из страны X» — нет, у нас нет обратного поиска.
3. **Публичный канал или бот.** У них нет приватности по определению — мы выдадим всё что публично (это и так видят все).

4. **Юзер включил публичный профиль.** Тогда имя и аватар уже публичны и видны всем без отдельной передачи с нашей стороны. Это осознанный выбор юзера, и он был предупреждён.

Что нам видно из инфраструктуры — честно

Любой инфраструктурный провайдер, через которого проходят серверные пакеты, технически видит часть сетевой картины: какие наши узлы обмениваются пакетами, когда и какого размера. Это не даёт ему содержимое сообщений и не даёт ему ключи.

Что провайдер видит:

- Сетевые метаданные между нашими серверными узлами — направление пакетов, время и размеры. Это служебная картина инфраструктуры, а не открытый текст сообщений.

Что провайдер НЕ видит:

- Содержимое ваших сообщений — оно шифруется протоколом, а серверные соединения дополнительно закрываются TLS/mTLS.
- Управляющий трафик кластера — отделён от пользовательского горячего пути и защищается отдельным внутренним каналом.
- Ваш IP-адрес — мы его не сохраняем. В рабочей записи сообщения остаются только A, B, дата/время, состояние доставки и сам зашифрованный блок.

Подробнее — в публичной политике приватности, раздел 4.5.

Итог главы

- Визитка (имя + аватар + статус) — **запечатанная**, ключик только у вас и у выбранных вами контактов.
- Два режима видимости — «для контактов» (по умолчанию) и «публичный» (опция для блогеров / каналов / бизнесов).
- Телефонные номера — мы не знаем и физически не можем проверить принадлежность без устройства юзера.
- Работает одинаково в обычных и секретных чатах.
- Код, параметры и события Сейфов публикуются так, чтобы внешние проверяющие могли сверять реализацию, церемонию и журнал прозрачности.
- Инфраструктурный провайдер может видеть сетевую служебную картину между серверными узлами, но не содержимое сообщений и не ключи.

ГЛАВА ШЕСТАЯ

VI

Несколько устройств

QR-код для связывания, до 10 устройств на аккаунт, восстановление по 24 словам и аварийное восстановление через 24+12 слов.

UmbrellaX поддерживает до 10 устройств на один аккаунт. Связывание через QR-код.

Процедура связывания

```
1. На новом устройстве (ПК, планшет, веб) выбираете "Связать с аккаунтом"
2. Новое устройство показывает QR-код
3. На основном устройстве (телефоне) → Настройки → Устройства → "Сканировать QR"
4. Сканируете QR
5. Основное устройство проверяет подпись, отправляет авторизацию на серверы
6. Новое устройство получает ключи от 5 Сейфов → готово
```

Время всей процедуры: 10-15 секунд.

Что синхронизируется

- **Обычные чаты** — синхронизируются на все устройства (ключи от Сейфов выдаются каждому авторизованному устройству).
- **Секретные чаты — не синхронизируются** (физика сквозного шифрования). Останутся только на оригинальном устройстве.
- **Входящие сообщения** — доставляются на все авторизованные устройства одновременно.
- **Исходящие** — синхронизируются обратно через нашу внутреннюю шину событий.

Пример: адвокат Карим

ПРИМЕР

Карим — адвокат в Алматы. Работает на iPhone в пути, на Macbook в офисе, имеет iPad для клиентов. Использует UmbrellaX для общения с клиентами.

Обычные чаты (переписка с коллегами):

- Видит их на всех 3 устройствах. История синхронизируется.
- Когда отправляет сообщение с Macbook — оно появляется через секунду на iPhone.
- Поиск по истории работает на всех устройствах.

Секретные чаты (с клиентами — адвокатская тайна):

- Создал секретный чат с клиентом на Macbook.
- На iPhone и iPad этот чат **не виден** — это физика (ключей нет).
- Если хочет вести тот же разговор с клиентом со iPhone — придётся создать **новый** секретный чат прямо с iPhone.

Удаление устройства

Настройки → Устройства → выбрать → «Удалить». Устройство **больше не получает ключи**. При следующем запросе — «Сессия истекла, пожалуйста переподключитесь».

Восстановление при потере телефона

Есть два разных случая.

Обычное восстановление: если потеряли основное устройство, но у вас есть 24 слова seed-фразы и/или другое уже привязанное устройство, новое устройство восстанавливает личность и получает доступ к обычной истории после проверки.

Безусловное аварийное восстановление: если потеряны все устройства, используется связка **24 слова + дополнительные 12 слов кода восстановления**. Эти 12 слов создаются отдельно и должны храниться отдельно от 24 слов. Вместе они детерминированно выводят новую личность аккаунта, старые устройства отзываются, а в журнале ключей появляется запись ротации. Это защищает от сценария, где злоумышленник нашёл только одну фразу.

Если потеряли и 24 слова, и дополнительные 12 слов — аккаунт не восстановим. Как в криптокошельках. Это сознательный выбор ради приватности — никого не можем восстановить даже по судебному приказу.

ГЛАВА СЕДЬМАЯ

VII

Когда суд требует данные

Юридические запросы получают юридический ответ, но личные данные, содержимое и IP мы не передаём, потому что не храним их как раскрываемые данные. Разведкам не передаём ничего.

Казахстанский судебный приказ (наша юрисдикция)

Мы отвечаем на такие запросы юридически, но не передаём личные данные пользователя, потому что они не хранятся в раскрываемом виде.

Что мы не храним и не можем передать:

- **Содержание сообщений** — физически не можем прочитать (см. главу 2).
- **IP-адреса** — не храним вообще никогда.
- **Email** — не храним.
- **Номер телефона** — не храним.
- **Настоящее имя и аватар** — храним только запечатанный конверт без ключа.
- **Конвертная информация** (адреса отправителя/получателя/время старых сообщений) — храним только до момента доставки + получения «прочитано», обычно минуты, потом удаляем. Запросы про «кто с кем общался месяц назад» физически не имеют ответа.
- **Само содержимое визитки** — только зашифрованный конверт без ключа.
- **Доли ключей в Сейфах** — мы не имеем доступа.

Публичный псевдоним или публичный профиль, если пользователь сам их включил, и так видны всем. Непубличные данные мы не раскрываем, потому что у нас нет открытого содержимого и ключей.

Вместо передачи личных данных по таким запросам уходит юридический ответ: запрошенные непубличные данные не хранятся в открытом виде, ключей у нас нет, содержание сообщений и профиль раскрыть невозможно.

Иностранные судебные приказы (США, ЕС, другие страны)

Мы НЕ выполняем автоматически. Требуем **MLAT** (межгосударственное соглашение о правовой помощи) между Казахстаном и страной запроса. Если есть соглашение — через казахстанский суд.

Если соглашения нет — отказ.

Правоохранительные запросы без судебного приказа

Отказываем. Даже в экстренных ситуациях (самоповреждение, похищение).

Исключение: проверенная жертва (см. ADR-19c). Жертва детской порнографии или revenge porn может подать заявление через `/victim-portal`, пройти проверку личности нашим юридическим отделом, и мы передадим отчёт соответствующему агентству (NCMEC для детской порнографии, StopNCII, GIFCT, Polaris и другие). **По запросу жертвы, не правительства.**

Чего мы никогда не делаем

- **Автоматическая отправка отчётов в NCMEC** (без запроса жертвы).
- **Добровольная передача данных** разведкам.
- **Закладки (backdoor)** в приложении.
- **Передача ключей Сейфов** (у нас их нет — корневой ключ уничтожен).
- **Автоматический скан содержимого.**

Warrant Canary

На странице `https://umbrellax.io/canary` публичные заявления:

> Мы **НЕ** получали:
>
> - Письмо национальной безопасности за последний квартал.
> - Судебный запрет разглашения за последний квартал.
> - Требование установить закладку.
> - Требование передать ключи шифрования.

Обновляется примерно 19-го числа каждого месяца.

Если warrant canary перестаёт обновляться дольше 35 дней — значит что-то изменилось. Мы не можем сказать что, но молчание — сигнал.

ГЛАВА ВОСЬМАЯ

VIII

Серьёзный сбой

Что произойдёт если взломают Почтальона, один, два или три Сейфа. И что будет с вашей приватностью в каждом случае.

Сценарий 1: Почтальон взломан

Взломщик получил наш обычный облачный сервер. Видит **зашифрованные сообщения**.

Что могут прочитать: ничего. Ключей у Почтальона нет (они в Сейфах).

Что произойдёт: восстановление из резервной копии (24-48 часов), смена всех паролей базы, публичный разбор полётов.

Для вас: возможно небольшая задержка доставки сообщений. Приватность не страдает.

Сценарий 2: 1-2 Сейфа взломаны

Система продолжает работать: порог 3-из-5 — нужно ещё 1-2 Сейфа для полной компрометации.

Что произойдёт: изоляция скомпрометированных Сейфов, церемония нового Сейфа (2-4 недели), смена главного ключа.

Для вас: возможна небольшая задержка в некоторых операциях. Приватность **не страдает**.

Сценарий 3: 3 и больше Сейфов взломаны (КРИЗИС)

Главный ключ потенциально скомпрометирован.

Что произойдёт:

- **Warrant canary перестаёт обновляться** (первый публичный сигнал).
- Новые сообщения в обычном режиме **блокируются**.
- Полная церемония новых 5 Сейфов + новый главный ключ.
- **Все активные пользователи** повторно шифруют свои чаты на стороне клиента.
- Полный внешний аудит.
- Публичный разбор полётов со всеми подробностями.

Для вас: прерывание работы на несколько недель. Сообщения в момент инцидента могут быть потеряны. Но **приватность исторически защищена** — только шифротекст утёк, а ключи уже сменены.

Сценарий 4: Экстренный взлом криптографии (маловероятно до 2040 года)

Найдена уязвимость в AES-256 или ML-KEM (крайне маловероятно, но на всякий случай план есть).

Что произойдёт:

- Экстренное обновление приложения (новая криптография).
- **Ваше устройство автоматически мигрирует** старые чаты на новый алгоритм (на стороне клиента, не сервера).
- Льготный период 6 месяцев для неактивных пользователей.
- **Запись в warrant canary:** «Экстренная миграция криптографии. Активирован Путь А. Церемонии на Сейфах не проводилось.»

Общий принцип — «никакой тайной компрометации»

Любой реальный инцидент публично задокументирован в warrant canary + журнал прозрачности + отчёт внешнего аудита. Мы **не скрываем** инциденты компрометации.

Это отличает нас от моделей «мы обещаем» (Telegram), где при утечке информация может замалчиваться. У нас архитектурная ответственность.

ГЛАВА ДЕВЯТАЯ

IX

Сравнение с другими

Signal, WhatsApp, Telegram — и UmbrellaX в двух режимах. Где мы проигрываем, где выигрываем, и что уникально.

ДВЕ ПРАВЫЕ КОЛОНКИ — НАШИ РЕЖИМЫ

ОСОБЕННОСТЬ	SIGNAL	WHATSAPP	TELEGRAM	НАШ «ОБЫЧНЫЙ»	НАШ «СЕКРЕТНЫЙ»
Содержимое зашифровано по умолчанию	Да	Да	Нет (обычные чаты серверные)	Да (MLS + Сейфы 3 из 5)	Да (чистый MLS)
Ключи только на устройствах	Да	Да	Нет	Нет (облачный режим 3 из 5)	Да
Мульти-устройство	Да	Да	Да	Да	Нет
Большие группы	до 1000	до 1024	до 200 000	до 200 000	до ~1000
Боты	Нет	Нет	Да	Да	Нет
Сервер может прочитать	Нет	Нет	Да	Нет (корневой ключ уничтожен)	Нет
Ключи живут	Устройства	Устройства	Серверы	Устройства + 5 Сейфов	Устройства
Регистрация	Телефон	Телефон	Телефон	Крипто-ключ + 24 слова; 12 слов для аварийного восстановления	Крипто-ключ
Постквантовая защита	Нет (Kyber частично)	Нет	Нет	Гибрид с MVP 0	Гибрид с MVP 0
Сквозные звонки	Да	Да	Да (1-на-1)	Да	Да
Звонки в России и Иране	Может не работать	Может не работать	Не работает	Работает (облачный запасной)	Работает
Хранение IP-адреса	90 дней иногда	Facebook-логи	Неизвестно	Никогда	Никогда
Warrant canary	Нет	Нет	Нет	Да, ежемесячно	Да, ежемесячно
Код для проверки	Да	Нет	Частично	Да	Да

Что уникально у UmbrellaX

1. **Архитектура с уничтоженным корневым ключом** — единственный мессенджер где админский доступ к главному ключу физически уничтожен и это публично подтверждено (не «мы обещаем не заглядывать»).
2. **Постквантовый гибрид с первого дня** — другие мессенджеры только планируют на 2027-2030 годы.
3. **5 Сейфов в 5 юрисдикциях** — никто из конкурентов не имеет такой географически + юридически распределённой системы хранения ключей.
4. **Ноль хранения IP** — Signal хранит 90 дней иногда, WhatsApp ещё больше.
5. **Регистрация через крипто-ключ** — без телефона и email (Signal требует телефон).
6. **Журнал прозрачности на каждый запрос** — уровень Apple и Cloudflare по прозрачности (другие мессенджеры публикуют только квартальные PDF).

ГЛАВА ДЕСЯТАЯ

X

Частые вопросы

Ответы на двенадцать вопросов которые мы чаще всего слышим — от «вы правда не можете прочитать?» до «как вы зарабатываете?».

Вопрос: Вы действительно не можете прочитать мои обычные сообщения?

Да. Главный ключ разбит на 5 кусочков в 5 независимых Сейфах (защита AMD SEV-SNP, админский доступ уничтожен). Наша процедура уничтожения корневого ключа подтверждена нотариусом, видео публично. Без **одновременной компрометации 3 из 5 Сейфов** собрать ключ невозможно. Это физически невыполнимо для любого взломщика (включая государство).

Вопрос: А если я хочу чтобы вообще никто не мог — только я?

Используйте **секретный режим**. Ключи MLS только на ваших устройствах. Никаких Сейфов, никакой синхронизации. Плата за это: не работает на нескольких устройствах, максимум около 1000 участников в группе.

Вопрос: Что будет с моими сообщениями если я потеряю телефон?

Обычный режим: если есть 24 слова seed-фразы → восстанавливаете стандартный доступ. Если потеряны все устройства, нужен полный аварийный набор **24+12 слов**: 24 слова личности и отдельные 12 слов кода восстановления. Если нет этих фраз и нет второго привязанного устройства → история потеряна (мы не можем восстановить — корневой ключ уничтожен).

Секретный режим: если нет того самого устройства где был секретный чат → история потеряна безусловно (даже с 24 словами).

Вопрос: Почему вы не используете просто обычное сквозное везде?

Telegram использует серверное по умолчанию именно потому что чистое сквозное ломает фичи: боты не могут работать, группы больше 1000 невозможны, синхронизация между устройствами невозможна.

Наш обычный режим — **гибрид лучшего**: преимущества Telegram (фичи) + физическая невозможность прочитать (уничтоженный корневой ключ + Сейфы).

Вопрос: Можно ли выключить Сейфы и работать только серверным режимом?

Нет. Это противоречит нашему основополагающему обязательству. Сейфы — **архитектурная глубокая защита**, не опция.

Вопрос: А что если Apple или Google заставит вас добавить закладку?

Архитектурно мы **не можем добавить закладку в обычный режим** без церемонии нового главного ключа (который потребует все 5 Сейфов + ротацию). Церемония видна в публичном журнале прозрачности — тайком не получится.

Секретный режим — ключи только на устройствах, мы их вообще не видим.

Если нас заставили бы изменить код (чтобы клиент отправлял открытый текст разработчикам) — это возможно только через обновления в App Store. **Воспроизводимые сборки + проверки прозрачности** защищают от этого (внешние аудиторы проверяют что сборка соответствует исходникам).

При настоящем принуждении: warrant canary перестаёт обновляться. **Это ваш сигнал.**

Вопрос: Что такое «постквантовый гибрид»?

Квантовые компьютеры (теоретически через 15-25 лет) смогут ломать классическую криптографию (X25519). Атака «собери сейчас — расшифруй потом» уже реальна — собирают зашифрованный контент сегодня, расшифруют через 20 лет.

UmbrellaX **с MVP 0** использует **гибрид**: одновременно классический X25519 + постквантовый ML-KEM-768 (стандарт NIST 2024 года). Если один слой сломается — второй защитит. Рассчитано на 40+ лет безопасности.

Вопрос: Что с номером телефона при регистрации?

Опционально. Можете добавить для поиска по контактам (ваши друзья находят вас через номер). Хранится как **хэш SHA-256 + соль**, не сам номер. Можно удалить в любой момент.

Основной идентификатор — криптографический ключ Ed25519, генерируется на устройстве.

Вопрос: Кто владеет UmbrellaX?

UmbrellaX LLP — казахстанская компания (BIN 260440006927). Зарегистрирована в городе Орал (Западный Казахстан). Основана в 2026 году.

Расположение инфраструктуры: публичный документ не раскрывает провайдеров, страны, регионы, города, дата-центры, количество машин и маршруты. Это операционная безопасность: такая карта не нужна пользователю для проверки криптографии, но полезна атакующему.

Вопрос: Как вы зарабатываете?

- **Premium $4.99 в месяц** — VPN на весь телефон, файлы до 4 ГБ, группы до 10 тысяч.
- **VIP $1000 в месяц** — для публичных лиц, без верификации (только факт оплаты).

- **UMX-Coin** — внутренняя валюта для подарков.
- **Реклама** — только в публичных пространствах (паблики, блоги), **не в чатах**, контекстная (не поведенческая).

Вопрос: Код доступен для проверки?

- **Клиенты** (iOS, Android, веб, десктоп) — доступны для проверки безопасности.
- **Критические протокольные части** — доступны для аудита, криптотестирования и ответственного анализа по публичным условиям доступа.
- **Боевая инфраструктурная карта** — не публикуется: провайдеры, регионы, маршруты и количество машин скрыты намеренно.
- **Процедуры церемонии** — описаны публично без раскрытия данных, которые помогают атаковать размещение.

Вопрос: Почему вы не публикуете где стоят Сейфы?

Потому что это было бы приглашением к атаке. Публично проверяется правило **3 из 5**, код, форматы, тесты, церемония и журнал прозрачности. Точные провайдеры, регионы, маршруты и количество машин остаются закрытой операционной информацией.

ГЛАВА ОДИННАДЦАТАЯ

XI

Для внешних аудиторов

Формализация протокола, параметры Шамира, цепочка проверки AMD SEV-SNP, модель угроз и контакты для ответственного раскрытия уязвимостей.

Этот раздел для исследователей безопасности, аудиторов, криптографов. Здесь более техничные подробности, но по-прежнему на русском.

10.1 Формализация протокола Sealed Boxes

Обозначения:

- `M` — главный ключ разговора (256 бит).
- `(K_i)_{i=1..5}` — кусочки по Шамиру, порог t=3.
- `pk_device, sk_device` — пара ключей Ed25519 устройства.
- `pk_account, sk_account` — пара ключей Ed25519 аккаунта (выведена из BIP-39 seed).
- `E_k(m)` — шифрование с проверкой целостности (AEAD) сообщения `m` под ключом `k`. В текущей реализации облачного wrap это ChaCha20-Poly1305; для звонков SFrame отдельно использует AES-256-GCM-SHA512-128.

Операция завёртывания (Алиса отправляет):

```
Алиса:
  K := random_32_bytes()  # одноразовый AEAD-ключ для сообщения
  c := E_K(m)

  Отправляет запрос завёртывания 3 из 5 Сейфов:
    request = { conversation_id, participants, K }

  Каждый Сейф (3 из 5):
    проверяет что запрос авторизован (через подпись нашего каталога устройств)
    вычисляет кусочек завёрнутого K через пороговую криптографию
    возвращает share_i

  Клиент собирает wrapped_K из 3 кусочков через интерполяцию Шамира

  Отправляет Почтальону: { c, wrapped_K, metadata }
```

Операция развёртывания (Боб читает):

```
Боб:
  Получает шифротекст + wrapped_K от Почтальона

  Для 3 Сейфов из 5:
    Отправляет запрос развёртывания:
      request = { conversation_id, pk_device, challenge, sig_device(challenge) }

    Каждый Сейф:
      проверяет pk_device в authorized_devices[account_id]
      проверяет sig_device
      расшифровывает свой кусочек M по Шамиру
      выводит кусочек K из wrapped_K с помощью своего кусочка
      отправляет кусочек K Бобу (напрямую через TLS, не через Почтальона)

  Боб собирает K из 3 кусочков (интерполяция Шамира)
  Боб расшифровывает c с ключом K → открытый текст m
```

Свойство безопасности: ни один Сейф в одиночку не имеет полезной информации. Атакующий, скомпрометировавший менее 3 Сейфов, имеет теоретико-информационно ноль знаний о M.

10.2 Интеграция с MLS RFC 9420

Групповой MLS для каждого чата:

- **Эпоха 0:** начальное создание группы. Участники добавляются через `KeyPackage`. Дерево ratchet выводит начальный `group_secret`.
- **Эпоха N:** смена состава триггерит новый `commit`. Вращение дерева выводит новый `group_secret`. Совершенная секретность вперёд + постскомпрометационная безопасность.

- **Каждое сообщение:** ratchet отправителя выводит уникальный ключ AEAD для каждого сообщения (совершенная секретность вперёд).

Наш набор шифров (основной MVP 0): `MLS_256_XWING_CHACHA20POLY1305_SHA256_Ed25519`.

Гибридный KEX через X-Wing draft-10:

```
shared = HKDF-Extract(
    salt = "UmbrellaX-MLS-xwing-v1",
    ikm = XWing.Decapsulate(ct, sk)  # X25519 + ML-KEM-768
)
group_secret = HKDF-Expand(shared, info = "MLS-group-secret-v1", L = 32)
```

10.3 Справки AMD SEV-SNP

Каждый Сейф публикует справку каждые 24 часа:

```
Поля справки:
- HARDWARE_ID: уникальный идентификатор чипа
- POLICY: зашифрованная память, нет отладки, нет миграции
- MEASUREMENT: SHA384(загрузчик || ядро || образ VM || бинарник программы Сейфа)
- REPORT_DATA: версия программы Сейфа + хэш конфигурации
- TIMESTAMP: Unix-время
- SIGNATURE: корневой ключ AMD (ECDSA P-384)
```

Цепочка проверки:

```
AMD Root CA (открытый ключ AMD известен)
  → AMD SEV Signing CA (для каждого семейства)
    → AMD SEV VCEK (Versioned Chip Endorsement Key для каждого чипа)
      → Подпись справки
```

Любой может получить справку с `attestation.umbrellax.io/<сейф-id>/<timestamp>` и проверить:

1. Цепочку подписи до корня AMD.
2. Что measurement совпадает с опубликованным хэшем открытого кода.
3. Что политика не разрешает отладку.
4. Что метка времени свежая (меньше 24 часов).

10.4 Параметры схемы Шамира

- **Поле:** GF(256) для байт-уровневой эффективности, умножение через таблицу поиска.
- **Секрет:** M = 256 бит = 32 байта.
- **Кусочки:** 5 штук, порог t=3.
- **Полином:** P(x) = M + r_1·x + r_2·x^2 (полином степени 2, случайные коэффициенты).
- **Кусочек i:** (i, P(i)) для i = 1..5.

- **Восстановление:** интерполяция Лагранжа на любых 3 кусочках.

Теоретико-информационная безопасность: с менее чем 3 кусочками, M равномерно случаен с точки зрения взломщика (даже с неограниченной вычислительной мощностью).

10.5 Модель угроз

Рассматриваемые противники:

1. **Сетевой противник** — может наблюдать и портить TLS-трафик. Митигация: TLS 1.3 + закрепление сертификата + mTLS.
2. **Компрометация Почтальона** — взломщик получает корневой доступ облачного стека. Видит шифротекст + завёрнутые ключи, но не может расшифровать.
3. **Компрометация одного Сейфа** — взломщик получает 1/5 кусочка Шамира. Недостаточно.
4. **Компрометация нескольких Сейфов** — взломщик получает 3+ Сейфа. **Считается катастрофическим**, warrant canary сигнализирует.
5. **Инсайдер или админ** — после церемонии нет доступа к Сейфам (корневой ключ уничтожен).
6. **Государственное принуждение** — судебный приказ требует данные которых у нас нет.
7. **Цепь поставок** — компрометация чипа AMD. Митигация: несколько поставщиков (AMD + Intel TDX рассматривается).
8. **Компрометация на стороне клиента** — в стиле Pegasus. Вне области действия протокола (уровень операционной системы).

10.6 Внешний аудит

- **Ежегодный независимый аудит** — часть производственной процедуры UmbrellaX. Целевые исполнители: Cure53, Trail of Bits или сопоставимые независимые команды.
- Охват: реализация протокола, обзор кода, процедуры церемонии, согласованность журнала прозрачности, схема 3 из 5, устойчивость к принуждению и ошибкам эксплуатации.
- Отчёты публикуются на `audit.umbrellax.io`.
- Публичное вознаграждение за ответственное раскрытие уязвимостей.

10.7 Контакт для аудиторов

- **Отчёты об уязвимостях:** `security@umbrellax.io`
- **Программа вознаграждений:** `https://umbrellax.io/bounty` или `security@umbrellax.io`
- **Журнал прозрачности:** `https://transparency.umbrellax.io`
- **Warrant canary:** `https://umbrellax.io/canary`
- **Портал для жертв:** `https://umbrellax.io/victim-portal`

ГЛАВА ДВЕНАДЦАТАЯ

XII

Модерация — как работают жалобы без лазейки для расшифровки

Частый вопрос: если модераторы могут удалять сообщения и банить аккаунты — значит ли это что у них есть ключи? Нет. Рассказываем подробно как это устроено и почему это не дыра в приватности.

Короткий ответ

Модераторы видят **только те сообщения, которые им явно пожаловались другие пользователи**. Они не имеют ключей от Сейфов. Они не могут прочитать историю. Они не могут выгрузить чужой чат. Они не могут «посмотреть что пишет Иванов» без действия со стороны самого получателя Иванова. Бан аккаунта — это отзыв авторизации устройств в нашем каталоге устройств, а не извлечение ключей.

> Модератор видит ровно одно сообщение — то которое получатель сам переслал как жалобу. Ничего сверх этого.

11.1 Откуда у модератора вообще появляется текст жалобы

Представьте: Ивану приходит мерзкое сообщение от Петра. Иван открывает его, читает (его устройство уже расшифровало содержимое — у Ивана есть ключ в кэше, он ведь участник чата). Иван делает долгое нажатие → «Пожаловаться» → выбирает категорию (оскорбление, детская порнография, угроза и т.д.).

В этот момент:

1. Приложение Ивана **локально, на его устройстве** делает копию этого конкретного сообщения в открытом виде. Оно уже расшифровано — сам Иван его только что прочитал.
2. Приложение подписывает копию ключом устройства Ивана.
3. Приложение отправляет пакет жалобы в нашу службу модерации (отдельный изолированный сервис).
4. Содержимое жалобы: конкретное сообщение (1 штука), идентификатор отправителя Петра, категория жалобы, подпись Ивана.

Ключевое: plaintext в нашу службу модерации попадает **только потому что Иван сам его отправил**. Мы не вытащили это сообщение из Сейфов — мы получили его от Ивана как добровольный repost в рамках жалобы. Это принципиальная разница.

11.2 Что видит модератор

Модератор первого уровня, когда берёт жалобу из очереди, видит:

- **Метаданные:** кто жаловался (Иван), на кого (Пётр), когда, какая категория.
- **Содержимое скрыто за кнопкой «Показать».** Нажимает — видит **только это одно сообщение** (или несколько последних сообщений контекста если категория того требует, например «травля», но контекст тоже присылается получателем, не извлекается с Сейфов).

Чего модератор НЕ видит и не может получить:

- Историю переписки Петра с Иваном (кроме того что Иван сам добровольно приложил к жалобе).
- Сообщения Петра в других чатах с другими людьми.
- Список чатов Петра.
- Контакты Петра.
- Ключи шифрования.

Модератор работает в отдельном внутреннем интерфейсе модерации, который технически не имеет сетевого доступа к Сейфам. Запросы вида «дай мне всё что писал Пётр» не существуют как опция — их нет в API. Архитектура исключает такой запрос не через политику, а через отсутствие кода.

11.3 Разделение ролей: модератор / старший проверяющий / юрист

- **Модератор первого уровня** — видит конкретную жалобу, решает «ложная жалоба» или «эскалировать».
- **Старший проверяющий** — утверждает удаление и бан. Один человек не может забанить в одиночку.

- **Юрист компании** — проверяет ордера судов, решает о передаче по закону, но даже у него нет ключей.
- **Системный администратор** (devops Почтальона) — имеет доступ к серверам Почтальона, но **не** к Сейфам. Может увидеть шифротекст в базе, который для него бесполезен.
- **Никто** из перечисленных не имеет admin-доступа к Сейфам. После церемонии уничтожения корневого ключа такого доступа не существует.

11.4 Что физически невозможно сделать админу

Вот список вещей, которые **архитектурно невозможны** в UmbrellaX (не «запрещены политикой», а **физически невозможны** потому что кода нет и ключей нет):

- Прочитать сообщение которое никто не пожаловался.
- Получить историю чата по user_id.
- Выгрузить все сообщения пользователя для передачи в правоохранительные органы.
- Расшифровать сохранённый ранее шифротекст задним числом.
- Добавить новое «админское устройство» в чужой аккаунт. Для этого нужна подпись с приватного ключа владельца аккаунта, которого у нас нет.
- Расшифровать Secret-чаты ни при каких обстоятельствах.

11.5 Как происходит удаление контента

Когда старший проверяющий подтверждает нарушение:

1. В Почтальоне запись этого сообщения **помечается как удалённая** — через фоновое уплотнение базы она физически исчезает.
2. Для медиа-файла — блоб удаляется из нашего файлового хранилища, а его **перцептивный хэш** заносится в базу хэшей запрещённого контента. Это позволяет предотвратить повторную загрузку того же файла, **не** добавляя возможность сканировать существующие.
3. Событие удаления попадает в журнал аудита (неизменяемое дерево Меркла) — любой внешний аудитор может проверить что удаление было обоснованным (ID тикета жалобы).

Ключи Сейфов при этом не трогаются. Сейфы вообще не знают про удаление конкретного сообщения — они оперируют ключами разговоров, не самими сообщениями.

11.6 Как происходит бан аккаунта

Когда старший проверяющий подтверждает бан пользователя:

1. Наш каталог устройств помечает все авторизованные устройства этого аккаунта как заблокированные.

2. Сейфы подписаны на события отзыва — при следующем запросе `unwrap` от забаненного устройства они откажут (подпись устройства больше не в списке авторизованных).
3. Забаненный пользователь при следующем запуске приложения видит «ваш аккаунт заблокирован, обжалование по адресу appeals@umbrellax.io».

Что важно:

- **Ключи не разглашаются.** Мы просто перестаём их выдавать этому устройству.
- **Старые сообщения других людей не затрагиваются.** Если Мария переписывалась с Петром до бана, у Марии в устройстве уже есть кэшированные ключи — она сможет прочитать историю.
- **Бан обратим.** Если обжалование удовлетворено, каталог устройств возвращает авторизацию, Сейфы снова начинают выдавать ключи. Никаких «мы потеряли ваши данные потому что банили» — данные были и остаются в хранилище, просто доступ был временно отключён.

11.7 Почему модерация не является лазейкой

Суммируем по четырём критериям:

НЕТ МАССОВОГО ДОСТУПА К СОДЕРЖИМОМУ

- Модератор видит только то что Иван сам прислал как жалобу.
- Нет API «дай всё что писал Пётр».
- Нет способа «проверить активность пользователя».

КЛЮЧИ ОСТАЮТСЯ В СЕЙФАХ

- Модератор не имеет сетевого доступа к Сейфам.
- Бан — отзыв авторизации, не извлечение ключей.
- Уничтоженный корневой ключ никто не трогает.

КАЖДОЕ ДЕЙСТВИЕ В АУДИТЕ

- Неизменяемое дерево Меркла для всех модерационных решений.
- Внешние аудиторы ежегодно проверяют.
- Пользователи могут запросить историю действий над своим аккаунтом.

ПОЛЬЗОВАТЕЛЬ ВСЕГДА ИНИЦИАТОР

- Без жалобы — ни одна строка текста не попадает в модерацию.
- Никакого упреждающего сканирования.
- Никакого PhotoDNA, никакого ML-сканирования содержимого.

11.8 А что со злоупотреблениями внутри самой команды модерации

Реальный вопрос: что если модератор — недобросовестный, и хочет нагадить пользователю (например, по заказу)? Можно ли злоумышленно забанить Ивана, если он ничего не нарушал?

Ответ — защищаемся технически и процедурно:

- **Два человека для бана.** Модератор не может забанить в одиночку — решение проходит через старшего проверяющего.
- **Случайное распределение жалоб.** Модератор не может выбрать «к кому в жалобу попаду» — очередь распределяется случайно.
- **Обжалование доступно сразу.** Забаненный пользователь в течение 14 дней может подать обжалование через `appeals@umbrellax.io`. Обжалование рассматривает **другая команда** (не та, что банила).
- **Публичный журнал аудита.** Все бан-решения со статистикой по категориям публикуются в `transparency.umbrellax.io`.
- **Психологическая защита модераторов.** Максимум 8 сложных жалоб в день на человека, ротация каждые 3 месяца в другую функцию, обязательный психолог — чтобы снизить риск ошибок из-за выгорания.

11.9 Модерация публичных каналов и пабликов (уточнение)

Публичные каналы — это отдельный сценарий. Они **по дизайну** открыты всем подписчикам — как веб-страница. Наш индексатор для поиска имеет свой ключ устройства в Сейфах для **публичных** (не приватных) чатов, чтобы индексировать их. Это обосновано: если любой может зарегистрироваться и читать канал, то наш индексатор делает то же самое.

Для приватных 1-на-1, групп и секретных чатов у индексатора нет доступа. Никакого «всё равно пройдусь по всему». Sealed боксы проверяют — публичный это канал или приватный, и только для публичных отдают ключ.

11.10 Если вам всё равно тревожно — используйте секретный режим

Секретный режим убирает последнюю гипотетическую точку доверия: в нём ключи **вообще никогда** не попадают в Сейфы. Даже если бы наши модераторы вдруг нашли способ скомпрометировать 3 Сейфа одновременно (это невозможно, но гипотетически) — секретные чаты не читаемы, потому что их ключей там не существует.

Секретный режим — **выбор для журналистов, активистов, юристов, врачей, и любого кто хочет исключить даже гипотетический внутренний сговор**.