



# Оглавление

<b>Предисловие</b> .....	<b>18</b>
Целевая аудитория .....	18
Условные обозначения в книге.....	18
Примеры исходного кода.....	19
Использование примеров исходного кода .....	20
Ссылки на компании и продукты.....	20
Адреса и транзакции в этой книге .....	20
Как с нами связаться .....	20
Связь с Андреасом.....	20
Связь с Рене.....	21
Связь с Олаолувой Осунтокун.....	21
Признательности от Андреаса.....	21
Признательности от Рене.....	21
Признательности от Олаолувы Осунтокун .....	22
Участники проекта .....	22
Источники .....	23
<b>Об авторах</b> .....	<b>24</b>
<b>Об иллюстрации на обложке (Колофон)</b> .....	<b>25</b>
<b>ЧАСТЬ I. ПОНИМАНИЕ СЕТИ LIGHTNING</b> .....	<b>27</b>
<b>Глава 1. Введение</b> .....	<b>28</b>
Базовые понятия сети Lightning.....	28
Доверие в децентрализованных сетях.....	30
Справедливость без центральной власти .....	31
Доверительные протоколы без посредников .....	32
Протокол справедливости в действии .....	33
Примитивы безопасности как строительные блоки.....	34
Пример протокола справедливости .....	35
Мотивация для сети Lightning .....	36
Масштабирование блочных цепей.....	36
Определяющие признаки сети Lightning.....	38
Примеры использования сети Lightning, пользователи и их истории.....	39
Вывод.....	40

<b>Глава 2. Приступаем к работе .....</b>	<b>41</b>
Первый кошелек Lightning Алисы .....	41
Узлы Lightning.....	42
Проводники Lightning .....	42
Кошельки Lightning .....	43
Тестовая сеть Bitcoin.....	46
Уравновешивание сложности и контроля .....	47
Скачивание и инсталляция кошелька Lightning .....	48
Создание нового кошелька .....	49
Ответственность за хранение ключей .....	49
Мнемонические слова.....	49
Безопасное хранение мнемоники.....	50
Загрузка биткойна в кошелек .....	51
Приобретение биткойна .....	51
Получение биткойна .....	52
Из системы Bitcoin в сеть Lightning.....	56
Каналы сети Lightning.....	56
Открытие канала Lightning.....	58
Покупка чашки кофе с помощью сети Lightning.....	61
Кофейня Боба.....	61
Счет Lightning .....	62
Вывод.....	64
<b>Глава 3. Как работает сеть Lightning .....</b>	<b>65</b>
Что такое платежный канал? .....	66
Основы платежного канала .....	66
Маршрутизирование платежей по каналам .....	67
Платежные каналы .....	68
Мультиподписной адрес .....	69
Финансовая транзакция.....	69
Пример плохой процедуры открытия канала .....	70
Фиксационная транзакция .....	70
Обман с предыдущим состоянием.....	72
Объявление канала.....	75
Заккрытие канала .....	75
Взаимное закрытие (хороший путь) .....	76
Принудительное закрытие (плохой путь) .....	77
Нарушение протокола (уродливый путь) .....	78
Счета.....	79
Платежный хеш и прообраз.....	80
Дополнительные метаданные .....	81
Доставка платежа.....	82
Эпидемический протокол обмена сообщениями между одноранговыми узлами.....	82
Отыскание пути и маршрутизация .....	83
Отыскание пути на основе источника .....	84
Луковичная маршрутизация .....	85
Алгоритм пересылки платежей .....	87

Шифрование однорангового обмена сообщениями .....	88
Мысли о доверии .....	89
Сравнение с системой Bitcoin.....	89
Адреса против счетов, транзакции против платежей.....	89
Выбор выходов против отыскания пути.....	90
Выходы со сдачей в Bitcoin против отсутствия сдачи в Lightning .....	91
Майнинговые комиссионные против маршрутизационных комиссионных .....	91
Комиссионные, варьирующиеся в зависимости от трафика, против объявленных комиссионных .....	91
Публичные Bitcoin-транзакции против частных платежей Lightning ...	92
Ожидание подтверждений против денежного расчета Lightning.....	93
Отправка произвольных сумм против ограничений по емкости.....	93
Стимулы для крупных платежей против малых платежей.....	94
Использование блочной цепи в качестве реестра против судебной системы.....	94
Офлайн против онлайн, на асинхронность против синхронности.....	94
Сатоши против миллисатоши .....	95
Общие черты сетей Bitcoin и Lightning.....	95
Денежная единица.....	95
Необратимость и окончательность платежей .....	96
Доверие и риск контрагента.....	96
Безразрешительная работа.....	96
Открытый исходный код и открытая система .....	96
Вывод.....	96

## **Глава 4. Программное обеспечение узла Lightning..... 97**

Среда разработки Lightning .....	98
Использование командной строки .....	98
Скачивание репозитория книги.....	99
Docker-контейнеры .....	100
Bitcoin Core и regtest.....	102
Сборка контейнера Bitcoin Core .....	102
Взаимодействие с контейнером bitcoin core .....	103
Проект c-lightning узла Lightning .....	105
Сборка c-lightning в качестве Docker-контейнера .....	105
Настройка сети Docker .....	106
Оперирование контейнерами bitcoind и c-lightning .....	107
Инсталлирование c-lightning из исходного кода .....	108
Инсталлирование необходимых библиотек и пакетов.....	108
Копирование исходного кода c-lightning.....	109
Компилирование исходного кода c-lightning.....	109
Проект демона узла сети Lightning .....	111
Docker-контейнер LND .....	111
Оперирование контейнерами bitcoind и LND .....	112
Инсталлирование LND из исходного кода.....	114
Копирование исходного кода LND .....	115
Компилирование исходного кода LND.....	115

Проект узла Lightning Eclair .....	116
Docker-контейнер Eclair .....	116
Оперирование контейнерами bitcoind и Eclair .....	117
Инсталлирование Eclair из исходного кода .....	118
Копирование исходного кода Eclair .....	119
Компилирование исходного кода Eclair.....	119
Сборка полной сети из разнообразных узлов Lightning.....	120
Использование docker-compose для оркестрирования Docker- контейнеров .....	120
Конфигурация docker-compose .....	121
Запуск образца сети Lightning.....	121
Открытие каналов и маршрутизирование платежа .....	122
Вывод.....	124

## **Глава 5. Оперирование узлом сети Lightning..... 125**

Выбор своей платформы.....	126
Почему для оперирования узлом Lightning важна надежность? .....	126
Типы аппаратных узлов Lightning .....	127
Оперирование в «облаке» .....	127
Оперирование узлом дома .....	128
Какое оборудование требуется для работы узла Lightning? .....	129
Переключение серверной конфигурации в облаке.....	130
Постоянное хранилище данных (накопитель) .....	131
Использование инсталлятора или помощника .....	131
RaspiBlitz .....	131
myNode .....	133
Umbrel.....	133
BTCPay Server .....	134
Узел Bitcoin или облегченный узел Lightning.....	135
Выбор операционной системы.....	136
Выбор имплементации узла Lightning.....	136
Инсталлирование узла Bitcoin или Lightning .....	137
Фоновые службы.....	138
Изоляция процесса.....	138
Запуск узла.....	139
Конфигурирование узла.....	140
Конфигурирование сети.....	141
Это просто работает!.....	142
Автоматическая переадресация портов с использованием UPnP .....	143
Использование Tor для входящих соединений .....	144
Ручная переадресация портов.....	145
Безопасность вашего узла.....	146
Безопасность операционной системы.....	146
Доступ к узлу.....	147
Резервное копирование узла и каналов.....	148
Риск со стороны горячего кошелька .....	150

Зачистка средств.....	150
Внутрицепная зачистка .....	151
Внецепная зачистка .....	151
Зачистка на основе подводного свопа .....	151
Подводные свопы с помощью петли.....	152
Время безотказной работы и доступность узла Lightning.....	153
Допускайте неисправности и автоматизируйте .....	154
Мониторинг доступности узла .....	154
Сторожевые вышки .....	155
Управление каналами .....	156
Открытие исходящих каналов.....	157
Автопилот .....	157
Получение входящей ликвидности .....	160
Закрытие каналов.....	161
Перебалансировка каналов.....	161
Комиссионные за маршрутизацию.....	162
Управление узлом.....	164
Ride The Lightning .....	164
Indmon .....	164
ThunderHub .....	165
Вывод.....	165

## **ЧАСТЬ II. СЕТЬ LIGHTNING В ДЕТАЛЯХ ..... 167**

### **Глава 6. Архитектура сети Lightning ..... 168**

Комплект протоколов сети Lightning.....	168
Lightning в деталях.....	169

### **Глава 7. Платежные каналы..... 171**

Другой способ использования системы Bitcoin .....	172
Владение биткойном и контроль над ним.....	173
Разнообразие форм (независимого) владения и мультиподпись.....	174
Совместное владение без независимого контроля.....	174
Предотвращение «привязанности» и нерасходуемости биткойна .....	174
Строительство платежного канала.....	175
Приватный и публичный ключи узла .....	175
Сетевой адрес узла .....	175
Идентификаторы узлов.....	176
Соединение узлов в качестве прямых одноранговых участников сети.....	176
Строительство канала .....	177
Одноранговый протокол для управления каналами .....	177
Поток сообщений об установлении канала .....	177
Сообщение open_channel .....	179
Сообщение accept_channel .....	180
Финансовая транзакция.....	181
Генерирование мультиподписного адреса .....	181
Сборка финансовой транзакции .....	181

Удерживание подписанных транзакций без широковещательной передачи .....	182
Возврат средств до финансирования .....	182
Сборка предварительно подписанной возвратной транзакции .....	183
Выстраивание транзакций в цепь без широковещательной передачи .....	183
Решение проблемы деформируемости (сегрегированный свидетель) .....	184
Сообщение <code>funding_created</code> .....	185
Сообщение <code>funding_signed</code> .....	186
Широковещательная передача финансовой транзакции .....	186
Сообщение <code>funding_locked</code> .....	187
Отправка платежей по каналу .....	187
Разделение остатка .....	187
Конкурирующие фиксации .....	188
Обман со старыми фиксационными транзакциями .....	189
Отзыв старых фиксационных транзакций .....	189
Асимметричные фиксационные транзакции .....	190
Задержанное (привязанное ко времени) расходование выхода <code>to_self</code> .....	191
Отзывные ключи .....	192
Фиксационная транзакция .....	193
Продвижение состояния канала вперед .....	195
Сообщение <code>commitment_signed</code> .....	196
Сообщение об отзыве и возврате .....	196
Отзыв и рефиксация .....	197
Обман и наказание на практике .....	197
Резерв канала: обеспечение личной заинтересованности .....	200
Заккрытие канала (кооперативное закрытие) .....	200
Сообщение <code>shutdown</code> .....	201
Сообщение <code>closing_signed</code> .....	202
Транзакция кооперативного закрытия .....	202
Вывод .....	203

## **Глава 8. Маршрутизация в сети платежных каналов..... 205**

Маршрутизирование платежа .....	205
Маршрутизация против отыскания пути .....	207
Создание сети платежных каналов .....	207
Физический пример «маршрутизирования» .....	208
Протокол справедливости .....	214
Имплементирование атомарных бездоверительных многопереходных платежей .....	214
Возвращаясь к примеру с донатами .....	215
Внутрицепное и внецепное улаживание HTLC-контрактов .....	216
Контракты с привязкой к хешу и времени .....	216
HTLC-контракты на Bitcoin Script .....	217
Платежный прообраз и верификация хеша .....	218
Распространение HTLC-контрактов от Алисы к Дины .....	219
Обратное распространение секрета .....	220
Привязка подписи: предотвращение кражи HTLC-контрактов .....	222

Оптимизация хеша.....	223
Кооперативный отказ и отказ тайм-аута по HTLC-контракту.....	225
Декрементирование привязок ко времени.....	226
Вывод.....	227
<b>Глава 9. Работа канала и пересылка платежей.....</b>	<b>228</b>
Локальный (один) канал против маршрутизируемых (многочисленных) каналов.....	229
Пересылка платежей и обновление фиксаций с помощью HTLC-контрактов.....	229
HTLC-контракт и поток фиксационных сообщений.....	230
Пересылка платежей с помощью HTLC-контрактов.....	230
Добавление HTLC-контракта.....	231
Сообщение update_add_HTLC.....	231
HTLC-контракт в фиксационных транзакциях.....	232
Новая фиксация с выходом из HTLC-контракта.....	233
Алиса фиксирует.....	234
Боб признает новую фиксацию и отзывает старую.....	235
Боб фиксирует.....	238
Несколько HTLC-контрактов.....	239
Исполнение HTLC-контракта.....	240
Распространение HTLC-контракта.....	240
Дина исполняет HTLC-контракт с Чаном.....	240
Боб улаживает HTLC-контракт с Алисой.....	241
Удаление HTLC-контракта из-за ошибки или истечения срока.....	244
Осуществление локального платежа.....	245
Вывод.....	245
<b>Глава 10. Луковичная маршрутизация.....</b>	<b>246</b>
Физический пример, иллюстрирующий луковичную маршрутизацию.....	247
Выбор пути.....	247
Сборка слоев.....	248
Отслаивание слоев.....	250
Введение в луковичную маршрутизацию на основе HTLC-контрактов.....	251
Алиса выбирает путь.....	251
Алиса конструирует полезные грузы.....	253
Полезный груз для Дины в заключительном узле.....	253
Переходный полезный груз для Чана.....	254
Переходный полезный груз для Боба.....	255
Окончательные полезные грузы переходов.....	256
Генерация ключей.....	256
Сеансовый ключ Алисы.....	257
Детали генерации ключей.....	258
Генерация совместных секретов.....	258
Обертывание луковичных слоев.....	260
Луковицы фиксированной длины.....	260
Обертывание луковицы (в общих чертах).....	261



Обертывание переходного полезного груза Дины .....	262
Луковично-маршрутизация защита от повторного воспроизведения и его обнаружение .....	265
Обертывание переходного полезного груза Чана .....	266
Обертывание переходного полезного груза Боба .....	267
Заключительный луковичный пакет .....	268
Отправка луковицы .....	269
Сообщение <code>update_add_htlc</code> .....	269
Алиса отправляет луковицу Бобу .....	269
Боб проверяет луковицу .....	270
Боб генерирует заполнитель .....	270
Боб распутывает свой переходный полезный груз .....	271
Боб извлекает внешний НМАС для следующего перехода .....	272
Боб удаляет свой полезный груз и сдвигает луковицу влево .....	272
Боб конструирует новый луковичный пакет .....	273
Боб верифицирует детали HTLC-контракта .....	273
Боб отправляет <code>update_add_htlc</code> Чану .....	274
Чан пересылает луковицу .....	274
Дина получает заключительный полезный груз .....	275
Возвращение ошибок .....	275
Сообщения о сбоях .....	276
Застрявшие платежи .....	278
Спонтанные платежи <code>keysend</code> .....	279
Конкретно-прикладные луковичные TLV-записи .....	279
Отправка и получение платежей <code>keysend</code> .....	280
Платеж <code>keysend</code> и конкретно-прикладные записи в приложениях Lightning .....	280
Вывод .....	280
<b>Глава 11. Сплетни и каналный граф .....</b>	<b>281</b>
Обнаружение одноранговых узлов .....	283
Самозагрузка P2P-узлов .....	284
Самозагрузка адресов DNS-серверов .....	284
Рабочий поток самозагрузки нового однорангового узла .....	285
Опции SRV-запроса .....	288
Канальный граф .....	289
Ориентированный граф .....	289
Сообщения эпидемического протокола .....	290
Сообщение <code>node_announcement</code> .....	291
Структура сообщения <code>node_announcement</code> .....	291
Валидация объявлений узла .....	292
Сообщение <code>channel_announcement</code> .....	292
Необъявленные (приватные) каналы .....	293
Локализация канала в блочной цепи Bitcoin .....	293
Короткий ID канала .....	294
Структура сообщения <code>channel_announcement</code> .....	294
Валидация объявления канала .....	296
Сообщение <code>channel_update</code> .....	296

Текущее сопровождение канального графа .....	297
Вывод.....	298
<b>Глава 12. Отыскание пути и доставка платежа.....</b>	<b>299</b>
Отыскание пути в рамках комплекта протоколов Lightning.....	299
Где же BOLT? .....	300
Отыскание пути: какую задачу мы решаем?.....	300
Выбор наилучшего пути.....	301
Отыскание путей в математике и информатике .....	302
Емкость, остаток, ликвидность.....	302
Неопределенность остатков .....	303
Сложность отыскания пути.....	304
Без лишних сложностей .....	304
Отыскание пути и процесс доставки платежа.....	305
Построение канального графа.....	305
Неопределенность в канальном графе .....	308
Неопределенность ликвидности и вероятность.....	309
Комиссионные и другие метрики канала .....	310
Отыскание кандидатных путей.....	312
Доставка платежа (цикл проб и ошибок) .....	312
Первая попытка (путь №1) .....	313
Учеба на ошибках .....	313
Вторая попытка (путь № 4).....	313
Учеба на успехах .....	314
Застоявшиеся знания? .....	314
Многокомпонентные платежи .....	314
Использование MPP .....	315
Разбивка платежей .....	315
Метод проб и ошибок в течение нескольких «раундов».....	316
Вывод.....	318
<b>Глава 13. Проводной протокол: фреймирование</b>	
<b>и расширяемость.....</b>	<b>319</b>
Слой обмена сообщениями в рамках комплекта протоколов Lightning .....	319
Проводное фреймирование .....	320
Высокоуровневое фреймирование.....	320
Кодировка типа.....	321
Расширения «Тип–длина–значение для сообщений» .....	322
Протокол буферизует формат сообщения .....	322
Прямая и обратная совместимости.....	323
Формат «Тип–длина–значение» .....	323
Целочисленная кодировка BigSize .....	324
Ограничения TLV-кодирования .....	325
Каноническое TLV-кодирование .....	325
Биты функциональностей и расширяемость протокола .....	325
Биты функциональностей как механизм обеспечения	
обнаруживаемости модернизаций .....	326

TLV для прямой и обратной совместимостей.....	327
Таксономия механизмов модернизации.....	328
Модернизации внутренней сети.....	328
Сквозные модернизации.....	328
Модернизации уровня строительства канала.....	329
Вывод.....	329

## **Глава 14. Шифрованный транспорт сообщений Lightning ..... 330**

Шифрованный транспорт в рамках комплекта протоколов Lightning.....	330
Введение.....	330
Канальный граф как децентрализованная инфраструктура публичных ключей.....	331
Почему не TLS?.....	332
Каркас криптосвязи на основе протокола Noise.....	333
Шифрованный транспорт Lightning в деталях.....	333
Noise_XK: рукопожатие Noise в сети Lightning.....	333
Нотация рукопожатия и поток протокола.....	334
Высокоуровневый обзор.....	334
Рукопожатие в трех действиях.....	335
Инициализация состояния сеанса рукопожатия.....	337
Акты рукопожатия.....	337
Акт первый.....	338
Акт второй.....	339
Акт третий.....	340
Шифрование транспортных сообщений.....	342
Ротация ключей сообщений Lightning.....	343
Вывод.....	343

## **Глава 15. Платежные запросы Lightning..... 345**

Счета в комплекте протоколов Lightning.....	345
Введение.....	345
Платежные запросы Lightning против Bitcoin-адресов.....	346
ВОЛТ #11: сериализация и интерпретация платежных запросов Lightning.....	347
Кодирование платежного запроса на практике.....	347
Человекочитаемый префикс.....	347
bech32 и сегмент данных.....	348
Тегированные поля счета.....	349
Вывод.....	350

## **Глава 16. Безопасность и конфиденциальность сети Lightning ..... 351**

Почему важна конфиденциальность?.....	351
Определения конфиденциальности.....	351
Процесс оценивания конфиденциальности.....	352
Анонимностное множество.....	353

Различия между сетями Lightning и Bitcoin с точки зрения конфиденциальности .....	354
Атаки на Lightning .....	356
Наблюдение за суммами платежей .....	356
Связывание отправителей и получателей .....	356
Раскрытие остатков каналов (прощупывание) .....	358
Отказ в обслуживании .....	360
DoS в Bitcoin .....	360
DoS в Lightning .....	361
Известные DoS-атаки .....	361
Закливание фиксаций .....	362
Запирание ликвидности канала .....	362
Межслоевая деанонимизация .....	362
Внутрицепная кластеризация Bitcoin-сущностей .....	363
Контрмеры .....	364
Внецепная кластеризация узлов Lightning .....	364
Контрмеры .....	364
Межслоевое связывание: узлы Lightning и Bitcoin-сущности .....	365
Граф Lightning .....	365
Как выглядит граф Lightning в реальности? .....	365
Граф Lightning сегодня .....	366
Атаки на основе топологии .....	366
Темпоральность сети Lightning .....	367
Централизация в сети Lightning .....	368
Экономические стимулы и графовая структура .....	368
Практические советы пользователям по защите их конфиденциальности .....	369
Необъявленные каналы .....	369
Соображения по маршрутизации .....	370
Принятие каналов .....	371
Вывод .....	372
Справочные материалы и дальнейшее чтение .....	372
Конфиденциальность и атакиощупыванием .....	372
Атаки переполнением .....	372
Соображения по маршрутизации .....	372
<b>Глава 17. Заключение .....</b>	<b>373</b>
Децентрализованные и асинхронные инновации .....	373
Инновации в Bitcoin-протоколе и в Bitcoin Script .....	374
Инновация в протоколе Lightning .....	374
Расширяемость TLV .....	375
Строительство платежного канала .....	375
Сквозные функциональности в порядке выбора .....	375
Lightning-приложения (LApps) .....	376
На старт, внимание, марш! .....	377

<b>Приложение А. Обзор основных принципов системы Bitcoin .....</b>	<b>378</b>
Ключи и цифровые подписи.....	378
Приватные и публичные ключи .....	379
Хеши .....	380
Цифровые подписи .....	382
Типы подписей .....	383
Транзакции Bitcoin .....	383
Входы и выходы.....	383
Транзакционные цепочки .....	385
TxID: идентификаторы транзакций.....	386
Выходные точки: выходные идентификаторы .....	387
Bitcoin Script.....	388
Работа языка Bitcoin Script.....	388
Привязывающие и отвязывающие скрипты .....	390
Привязывание к публичному ключу (подписи) .....	390
Привязывание к хешу (секрету) .....	391
Мультиподписные скрипты.....	392
Скрипты привязки ко времени .....	393
Скрипты с несколькими условиями.....	394
Использование управления потоком в скриптах.....	395
<b>Приложение В. Базовая инсталляция и использование Docker .....</b>	<b>397</b>
Инсталляция Docker .....	397
Базовые команды Docker .....	398
Сборка контейнера .....	398
Оперирование контейнером .....	398
Исполнение команды в контейнере .....	398
Остановка и запуск контейнера .....	398
Удаление контейнера по имени .....	399
Выведение списка оперируемых контейнеров .....	399
Выведение списка Docker-образов .....	399
Вывод.....	399
<b>Приложение С. Сообщения проводного протокола.....</b>	<b>400</b>
Типы сообщений.....	400
Структура сообщения.....	402
Сообщения об установлении соединения .....	402
Сообщение init.....	402
Сообщения об ошибке.....	403
Сообщение error .....	403
Оживленность соединения .....	404
Сообщение ping .....	404
Сообщение pong .....	404

---

Финансирование канала .....	405
Сообщение open_channel .....	405
Сообщение accept_channel .....	406
Сообщение funding_created .....	406
Сообщение funding_signed .....	407
Сообщение funding_locked .....	407
Заккрытие канала .....	407
Сообщение shutdown.....	408
Сообщение closing_signed .....	408
Операция канала .....	408
Сообщение update_add_htlc .....	408
Сообщение update_fulfill_htlc.....	409
Сообщение update_fail_htlc .....	409
Сообщение commitment_signed.....	409
Сообщение revoke_and_ack.....	410
Сообщение update_fee .....	410
Сообщение update_fail_malformed_htlc .....	410
Объявление канала.....	411
Сообщение channel_announcement .....	411
Сообщение node_announcement .....	411
Сообщение channel_update .....	412
Сообщение announce_signatures.....	412
Синхронизация канального графа.....	413
Сообщение query_short_chan_ids .....	413
Сообщение reply_short_chan_ids_end.....	413
Сообщение query_channel_range.....	413
Сообщение reply_channel_range.....	414
Сообщение gossip_timestamp_range .....	414
<b>Приложение D. Источники и уведомления о лицензиях.....</b>	<b>415</b>
Источники .....	415
Сервер BTCPay Server .....	416
Lamassu Industries AG .....	416
<b>Глоссарий.....</b>	<b>417</b>
<b>Предметный указатель .....</b>	<b>436</b>

# Предисловие

Сеть Lightning (Lightning Network, аббр. LN, или сеть-молния) – это второслойная одноранговая сеть, которая позволяет совершать платежи Bitcoin «вне цепи», то есть без фиксации их в качестве транзакций в блочной цепи (блокчейне) системы Bitcoin.

Сеть Lightning предоставляет безопасные, дешевые, быстрые и гораздо более приватные платежи Bitcoin, даже для очень малых платежей.

Основываясь на идее платежных каналов, впервые предложенной изобретателем системы Bitcoin Сатоши Накамото, сеть Lightning представляет собой маршрутизируемую сеть платежных каналов, в которой платежи делают «прыжки» вдоль пути платежных каналов от отправителя к получателю.

Первоначальная идея сети Lightning была предложена в 2015 году в новаторской статье Джозефа Пуна (Joseph Poon) и Тадеуша Дриджа (Thaddeus Dryja) «Сеть Lightning в рамках системы Bitcoin: масштабируемые мгновенные платежи вне цепи» (The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments). К 2017 году в интернете была запущена «тестовая» сеть Lightning, по мере того разные группы строили совместимые имплементации и координировали работу, чтобы установить какие-нибудь стандарты совместимости. В 2018 году сеть Lightning заработала, и потекли платежи.

В 2019 году Андреас М. Антонопулос, Олаолува Осунтокун и Рене Пикхардт согласились сотрудничать в написании этой книги. И похоже, мы добились успеха!

## ЦЕЛЕВАЯ АУДИТОРИЯ

Эта книга в основном предназначена для технически подкованных читателей, имеющих представление об основах системы Bitcoin и других открытых блочных цепей.

## УСЛОВНЫЕ ОБОЗНАЧЕНИЯ В КНИГЕ

В книге используются следующие типографические условные обозначения:

*курсивный шрифт*

обозначает новые термины, URL-адреса, адреса электронной почты, имена файлов и расширения файлов;

моноширинный шрифт

используется для листингов программ, а также внутри абзацев для ссылки на элементы программ, такие как переменные или имена функций, базы данных, типы данных, переменные среды, инструкции и ключевые слова;

**жирный моноширинный шрифт**

показывает команды либо другой текст, который должен быть набран пользователем;

*моноширинный шрифт курсивом*

показывает текст, который должен быть заменен значениями, передаваемыми пользователем, либо значениями, определяемыми по контексту.



Данный элемент обозначает подсказку или совет.



Данный элемент обозначает общее замечание.



Данный элемент обозначает предупреждение или предостережение.

## ПРИМЕРЫ ИСХОДНОГО КОДА

Примеры проиллюстрированы на языках Go, C++, Python и с использованием командной строки Unix-подобной операционной системы. Все фрагменты исходного кода доступны в репозитории на GitHub в подкаталоге *code*. Сделайте ответвление исходного кода книги, попробуйте примеры кода или отправьте исправления через GitHub<sup>1</sup>.

Все фрагменты исходного кода могут быть воспроизведены в большинстве операционных систем с минимальной инсталляцией компиляторов, интерпретаторов и библиотек для соответствующих языков.

Там, где это необходимо, мы предоставляем базовые инструкции по инсталляции и пошаговые примеры результата работы этих инструкций.

Некоторые фрагменты исходного кода и результаты его работы были переформатированы для печати. Во всех таких случаях строки были разделены символом обратной косой черты (`\`), за которым следует символ новой строки. При транскрибировании примеров удалите эти два символа и снова соедините строки, и вы должны увидеть результаты, идентичные тем, которые показаны в примере.

Во всех фрагментах исходного кода, там, где это возможно, используются реально существующие значения и вычисления, так что вы можете переходить

<sup>1</sup> См. <https://github.com/lnbook/lnbook>.



от примера к примеру и видеть одни и те же результаты в любом программном коде, который вы пишете для вычисления одних и тех же значений. Например, все приватные ключи и соответствующие им публичные ключи и адреса реально существуют.

## ИСПОЛЬЗОВАНИЕ ПРИМЕРОВ ИСХОДНОГО КОДА

Если у вас возникли технические вопросы или проблемы с использованием примеров исходного кода, то, пожалуйста, отправьте электронное письмо по адресу [bookquestions@oreilly.com](mailto:bookquestions@oreilly.com).

## СЫЛКИ НА КОМПАНИИ И ПРОДУКТЫ

Все ссылки на компании и продукты предназначены для образовательных, демонстрационных и справочных целей. Авторы не поддерживают ни одну из упомянутых компаний или продуктов. Мы не тестировали работу или безопасность ни одного из продуктов, проектов или сегментов исходного кода, показанных в этой книге. Используйте их на свой страх и риск!

## АДРЕСА И ТРАНЗАКЦИИ В ЭТОЙ КНИГЕ

Адреса, транзакции, ключи, QR-коды и данные блочной цепи Bitcoin, используемые в этой книге, по большей части реальны. Это означает, что вы можете просматривать блочную цепь, просматривать предлагаемые в качестве примеров транзакции, извлекать их с помощью ваших собственных скриптов или программ и т. д.

Однако обратите внимание, что приватные ключи, использованные для создания адресов, напечатанных в этой книге, были «сожжены». Это означает, что если вы отправите деньги на любой из этих адресов, то деньги будут либо потеряны навсегда, либо (что более вероятно) присвоены, поскольку любой, кто читает книгу, может забрать их, используя напечатанные здесь приватные ключи.



**НЕ ОТПРАВЛЯЙТЕ ДЕНЬГИ НИ НА ОДИН ИЗ АДРЕСОВ, УКАЗАННЫХ В ЭТОЙ КНИГЕ.** Ваши деньги будут взяты другим читателем или потеряны навсегда.

## КАК С НАМИ СВЯЗАТЬСЯ

Информация о книге «Освоение сети Lightning», а также открытое издание и переводы доступны по адресу <https://lnbook.info>.

# Об авторах

**Андреас М. Антонопулос** – автор бестселлеров, оратор, преподаватель и очень востребованный эксперт по системе Bitcoin и открытым технологиям на основе блочных цепей. Он известен тем, что облегчает понимание сложных тем и подчеркивает как положительное, так и отрицательное воздействие, которое эти технологии могут оказывать на наши глобальные общества.

Андреас написал еще два технических бестселлера для программистов с O'Reilly Media, «Освоение системы Bitcoin» (Mastering Bitcoin) и «Освоение системы Ethereum» (Mastering Ethereum). Он также опубликовал серию книг «Интернет денег», в которых основное внимание уделяется социальному, политическому и экономическому значению и последствиям этих технологий. Андреас еженедельно выпускает бесплатный образовательный контент на своем канале YouTube и проводит виртуальные семинары на своем веб-сайте. Узнайте больше по адресу [aantonop.com](http://aantonop.com).

**Олаолува Осунтокун** является соучредителем и техническим директором компании Lightning Labs, а также ведущим разработчиком Ind, одной из главных имплементаций сети Lightning. Он получил степень бакалавра и магистра в области информатики в UCSB и в 2019 году был членом класса Forbes 30 для молодых специалистов до 30 лет. Во время учебы в аспирантуре он сосредоточился на прикладной криптографии, в частности на шифрованном поиске. Более пяти лет он активно участвовал в разработке системы Bitcoin и является автором нескольких предложений по совершенствованию системы Bitcoin (BIP-157 и 158). В наши дни его основное внимание сосредоточено на строительстве, конструировании и разработке частных масштабируемых автономных протоколов блочной цепи, таких как Lightning.

**Рене Пикхардт** – опытный математик и консультант по науке о данных, который использует свои статистические знания для проведения исследований с NTNU по отысканию путей, конфиденциальности, надежности платежей и соглашениям об уровне обслуживания в сети Lightning. Рене ведет технический и ориентированный на разработчиков канал YouTube<sup>2</sup> о сети Lightning и уже ответил примерно на половину вопросов о работе сети Lightning на бирже Bitcoin, что делает его одним из лучших консультантов для всех новых разработчиков, которые хотят присоединиться к этому пространству. Рене провел множество публичных и частных семинаров о сети Lightning, в том числе обучал студентов резидентуры Chaincode Labs 2019 года вместе с другими ключевыми разработчиками сети Lightning.

---

<sup>2</sup> См. <https://www.youtube.com/renepickhardt>.

# Часть I

# Понимание сети Lightning

Обзор сети Lightning, подходящий для всех, кто заинтересован в понимании базовых концепций и использовании сети Lightning.

# Глава 1

## Введение

Добро пожаловать в книгу «Освоение сети Lightning»!

Сеть Lightning (часто сокращенно LN от англ. Lightning Network) меняет способ обмена стоимостями в интернете, и это одно из самых захватывающих достижений в истории системы Bitcoin. Сегодня, в 2021 году, сеть Lightning все еще находится в зачаточном состоянии. Сеть Lightning – это протокол для использования системы Bitcoin разумным и неочевидным способом. Это технология, накладываемая вторым слоем поверх системы Bitcoin.

Концепция сети Lightning была предложена в 2015 году, а первая ее имплементация была запущена в 2018 году. Начиная с 2021 года мы только начинаем видеть возможности, которые сеть Lightning предоставляет системе Bitcoin, включая улучшенную конфиденциальность, скорость и масштабирование. Обладая базовыми знаниями о сети Lightning, вы сможете сформировать будущее данной сети, одновременно создавая возможности для себя.

Мы исходим из того, что у вас уже есть некоторые базовые знания о системе Bitcoin, но если нет, то не волнуйтесь – в приложении А мы объясним наиболее важные концепции системы Bitcoin, которые вы должны знать, чтобы понять сеть Lightning. Если вы хотите узнать о системе Bitcoin больше, то можете прочитать книгу Андреаса М. Антонопулоса «Освоение системы Bitcoin», 2-е издание (Andreas M. Antonopoulos, Mastering Bitcoin, 2nd edition, O'Reilly), доступную бесплатно онлайн<sup>3</sup>.

Хотя большая часть данной книги написана для программистов, первые несколько глав написаны так, чтобы быть доступными любому, независимо от технического опыта. В этой главе мы начнем с некоторой терминологии, затем перейдем к рассмотрению концепции доверия и ее применения в этих системах и, наконец, обсудим историю и будущее сети Lightning. Давайте начнем.

### БАЗОВЫЕ ПОНЯТИЯ СЕТИ LIGHTNING

Когда мы займемся разведывательным анализом того, как на самом деле работает сеть Lightning, то столкнемся с некоторой технической терминологией, которая поначалу может немного дезориентировать. Хотя все эти понятия и термины будут подробно объяснены по мере продвижения по книге и определены в глоссарии, ознакомление с несколькими базовыми определениями

<sup>3</sup> См. <https://github.com/bitcoinbook/bitcoinbook>.

сейчас облегчит понимание концепций в следующих двух главах. Если вы еще не понимаете всех слов в этих определениях, то не беда. Вы будете понимать все больше по мере продвижения по тексту.

### *Блочная цепь, блокчейн*

Распределенный реестр транзакций, создаваемый сетью компьютеров. Bitcoin, например, – это система, которая создает блочную цепь. Сеть Lightning сама по себе не является блочной цепью и не создает блочную цепь. Это сеть, которая опирается на существующую внешнюю блочную цепь для обеспечения своей безопасности.

### *Цифровая подпись*

Цифровая подпись – это математическая схема для верифицирования подлинности цифровых сообщений или документов. Валидная цифровая подпись дает получателю основание полагать, что сообщение было создано известным отправителем, что отправитель не может отрицать отправку сообщения и что сообщение не было изменено при передаче.

### *Хеш-функция*

Криптографическая функция хеширования – это математический алгоритм, который соотносит данные произвольного размера с битовой строкой фиксированного размера (хешем) и предназначен для однопутной функции, то есть функции, которую невозможно инвертировать.

### *Узел*

Компьютер, который участвует в сети. Узел Lightning – это компьютер, который участвует в сети Lightning. Узел Bitcoin – это компьютер, который участвует в сети Bitcoin. Обычно пользователь LN выполняет узел Lightning и узел Bitcoin.

### *Внутри цепи и вне цепи*

Платеж происходит внутри цепи, если он зарегистрирован как транзакция в сети Bitcoin (или другой базовой) блочной цепи. Платежи, отправляемые по платежным каналам между узлами Lightning и которые не видны в базовой блочной цепи, называются платежами вне цепи. Обычно в сети Lightning единственными транзакциями внутри цепи являются транзакции, используемые для открытия и закрытия платежного канала Lightning. Существует третий тип транзакции, модифицирующий канал, именуемый склеиванием (splicing), который можно использовать для добавления/удаления суммы средств, зафиксированных в канале.

### *Платеж*

Когда стоимость обменивается в сети Lightning, мы называем это «платежом» по сравнению с «транзакцией» в блочной цепи Bitcoin.

### *Платежный канал*

Финансовая взаимосвязь между двумя узлами в сети Lightning, обычно имплементируемая с помощью мультиподписных Bitcoin-транзакций, которые имеют совместный контроль над биткойном между двумя узлами Lightning.

*Маршрутизация по сравнению с отправкой*

В отличие от системы Bitcoin, где транзакции «отправляются» путем их широковещательной передачи всем, Lightning – это маршрутизированная сеть, в которой платежи «маршрутизируются» по одному или нескольким платежным каналам по пути от отправителя к получателю.

*Транзакция*

Структура данных, которая регистрирует передачу контроля над некоторыми средствами (например, некоторыми биткойнами). Сеть Lightning опирается на транзакции Bitcoin (или транзакции другой блочной цепи) для осуществления контроля над средствами.

Более подробные определения этих и многих других терминов можно найти в глоссарии. На протяжении всей книги мы будем объяснять смысл этих терминов и то, как на самом деле работают данные технологии.



На протяжении всей этой книги вы будете встречать слово «Bitcoin» с заглавной первой буквой, которое относится к системе Bitcoin и является именем собственным. Вы также будете встречать слово «биткойн» со строчной буквой «б», которое относится к денежной единице. Каждый биткойн далее подразделяется на 100 миллионов единиц, каждая из которых называется «сатоши» (в единственном и множественном числе на русском пишется одинаково).

Теперь, когда вы знакомы с этими базовыми терминами, давайте перейдем к понятию, которое вам уже знакомо: доверие (trust).

## ДОВЕРИЕ В ДЕЦЕНТРАЛИЗОВАННЫХ СЕТЯХ

Вы часто будете слышать, как люди называют систему Bitcoin и сеть Lightning «бездоверительными» (trustless). На первый взгляд это сбивает с толку. В конце концов, разве доверие – это не хорошо? Банки даже используют его в своих названиях! Разве «бездоверительная» система, система, лишенная доверия, не является чем-то плохим?

Использование слова «бездоверительный» предназначено для того, чтобы передать идею способности работать, не нуждаясь в доверии к другим участникам системы. В такой децентрализованной системе, как Bitcoin, вы всегда можете заключить сделку с тем, кому доверяете. Однако система также гарантирует, что вас не обманут, даже если вы не можете доверять другой стороне транзакции. Доверие – это не обязательное, а приятное свойство системы.

Сравните это с традиционными системами, такими как банковское обслуживание, где вы должны доверять третьей стороне, поскольку она контролирует ваши деньги. Если банк нарушит ваше доверие, то вы можете обратиться в регулирующий орган или суд, но это потребует огромных затрат времени, денег и усилий.

Бездоверительность не означает отсутствие доверия. Она означает, что доверие не является необходимым условием для всех транзакций и что вы можете совершать транзакции даже с людьми, которым вы не доверяете, потому что система предотвращает обман.

Прежде чем мы перейдем к тому, как работает сеть Lightning, важно понять одну базовую концепцию, которая лежит в основе системы Bitcoin, сети Lightning и многих других подобных систем: то, что мы называем протоколом справедливости. Протокол справедливости – это способ достижения справедливых исходов между участниками, которым не нужно доверять друг другу, без необходимости в центральном органе власти, и он является основой децентрализованных систем, таких как Bitcoin.

## СПРАВЕДЛИВОСТЬ БЕЗ ЦЕНТРАЛЬНОЙ ВЛАСТИ

Когда у людей есть конкурирующие интересы, то как они могут установить достаточное доверие, чтобы участвовать в каком-то совместном или транзакционном поведении? Ответ на этот вопрос лежит в основе нескольких научных и гуманистических дисциплин, таких как экономика, социология, поведенческая психология и математика. Некоторые из этих дисциплин дают нам «мягкие» ответы, которые зависят от таких понятий, как репутация, справедливость, мораль и даже религия. Другие дисциплины дают нам конкретные ответы, которые зависят только от допущения о том, что участники этих взаимодействий будут действовать рационально, руководствуясь своими личными интересами в качестве главной цели.

В общих чертах, существует несколько способов обеспечивать справедливые исходы во взаимодействии между людьми, которые могут иметь конкурирующие интересы:

### *Требовать доверия*

Вы взаимодействуете только с теми людьми, которым уже доверяете, благодаря предыдущим взаимодействиям, репутации или семейным отношениям. Это достаточно хорошо работает в малых масштабах, в особенности в семьях и небольших группах, что является наиболее распространенной основой для кооперативного поведения. К сожалению, это не масштабируется и страдает от трайбалистского предубеждения (внутри группы).

### *Верховенство закона*

Установить правила взаимодействия, которые будут соблюдаться учреждением. Это масштабируется лучше, но не может масштабироваться глобально из-за различий в обычаях и традициях, а также неспособности масштабировать институты правоприменения. Одним из неприятных побочных эффектов этого решения является то, что по мере своего роста институты становятся все более и более мощными, а это может привести к коррупции.

### *Доверенные третьи стороны*

Поставить посредника в каждом взаимодействии, чтобы обеспечивать справедливость. В сочетании с «верховенством закона», обеспечивающим надзор за посредниками, это лучше масштабируется, но страдает от того же дисбаланса власти: посредники становятся очень влиятельными и могут привлечь коррупцию. Концентрация власти приводит к системному риску и системному провалу («слишком большой, чтобы позволить ему обанкротиться»).

*Теоретико-игровые протоколы справедливости*

Эта последняя категория возникает в результате сочетания интернета и криптографии и является предметом данного раздела. Давайте посмотрим, как она работает и в чем ее преимущества и недостатки.

**Доверительные протоколы без посредников**

Криптографические системы, такие как Bitcoin и сеть Lightning, – это системы, позволяющие совершать транзакции с людьми (и компьютерами), которым вы не доверяете. Такую работу часто называют «бездоверительной», хотя на самом деле она не является бездоверительной. Вы должны доверять выполняемому программному обеспечению и должны верить, что протокол, имплементированный этой программой, приведет к справедливым исходам.

Большое различие между подобного рода криптографической системой и традиционной финансовой системой заключается в том, что в традиционных финансах у вас есть доверенная третья сторона, например банк, для обеспечения справедливости исходов. Существенная проблема с такими системами заключается в том, что они передают слишком много власти третьей стороне, а также уязвимы перед единой точкой отказа. Если доверенная третья сторона сама нарушает доверие или пытается обмануть, то основание для доверия нарушается.

Изучая криптографические системы, вы заметите определенную закономерность: вместо того чтобы полагаться на доверенную третью сторону, эти системы пытаются предотвратить несправедливые исходы, используя систему положительных и отрицательных стимулов. В криптографических системах вы доверяете протоколу, фактически представляющему собой систему с набором правил, которые при правильной разработке будут правильно применять желаемые положительные и отрицательные стимулы. Преимущество такого подхода двоякое: вы не только избегаете доверия третьей стороне, но и уменьшаете необходимость обеспечения справедливых исходов. До тех пор, пока участники подчиняются согласованному протоколу и остаются в рамках системы, механизм стимулирования в этом протоколе обеспечивает справедливые исходы без контроля за его исполнением.

Использование положительных и отрицательных стимулов для достижения справедливых исходов является одним из аспектов раздела математики, именуемого теорией игр, предметом которого является изучение «моделей стратегического взаимодействия между лицами, принимающими рациональные решения»<sup>4</sup>. Криптографические системы, которые контролируют финансовые взаимодействия между участниками, такие как Bitcoin и сеть Lightning, в значительной степени опираются на теорию игр, чтобы предотвращать обман участников и позволять участникам, которые не доверяют друг другу, добиваться справедливых исходов.

Хотя теория игр и ее использование в криптографических системах на первый взгляд могут показаться запутанными и незнакомыми, скорее всего, вы уже знакомы с этими системами в своей повседневной жизни; вы просто еще

<sup>4</sup> Статья в Википедии ([https://en.wikipedia.org/wiki/Game\\_theory](https://en.wikipedia.org/wiki/Game_theory)), посвященная теории игр, содержит дополнительную информацию.



не узнаете их. В следующем далее разделе мы задействуем простой пример из детства, который поможет определить базовую закономерность. Как только вы поймете базовую закономерность, вы увидите ее повсюду в пространстве блочной цепи и научитесь распознавать ее быстро и интуитивно.

В этой книге мы называем эту закономерность протоколом справедливости, определяемым как процесс, в котором используется система положительных и/или отрицательных стимулов в целях обеспечения справедливых исходов для участников, которые не доверяют друг другу. Соблюдение протокола справедливости необходимо только для того, чтобы участники не могли избежать положительных или отрицательных стимулов.

## Протокол справедливости в действии

Давайте рассмотрим пример протокола справедливости, с которым вы, возможно, уже знакомы.

Представьте себе семейный обед с родителем и двумя детьми. Дети привередливы в еде, и единственное, что они согласятся съесть, – это жареную картошку. Родитель приготовил чашу жареного картофеля («картофель фри» или «чипсы», в зависимости от того, какой английский диалект вы используете). Брат и сестра должны разделить между собой тарелку с чипсами. Родитель должен обеспечить справедливое распределение чипсов между каждым ребенком; в противном случае родителю придется выслушивать постоянные жалобы (возможно, весь день), и всегда есть вероятность того, что несправедливая ситуация перерастет в насилие. Что должен делать родитель?

В этом стратегическом взаимодействии между двумя детьми, которые не доверяют друг другу и имеют конкурирующие интересы, существует несколько разных способов достижения справедливости. Наивный, но часто используемый метод заключается в том, что родители используют свою власть в качестве доверенной третьей стороны: они делят миску с чипсами на две порции. Это похоже на традиционные финансы, где банк, бухгалтер или юрист выступают в качестве доверенной третьей стороны, чтобы предотвратить любой обман между двумя сторонами, которые хотят совершить сделку.

Проблема этого сценария заключается в том, что он возлагает большую власть и ответственность на доверенную третью сторону. В этом примере родитель несет полную ответственность за равное распределение чипсов, а стороны просто ждут, наблюдают и жалуются. Дети обвиняют родителей в том, что они занимают сторону любимчика и несправедливо распределяют чипсы. Они дерутся из-за чипсов, крича, что «этот чипс больше!», втягивая родителя в свою перепалку. Это звучит ужасно, не так ли? Должен ли родитель кричать громче? Убрать все чипсы? Угрожать никогда больше не делать чипсы и отправить детей из-за стола голодными?

Существует гораздо более оптимальное решение: детей учат играть в игру под названием «дели и выбирай». За каждым обедом один ребенок делит чашу чипсов на две порции, а *другой* выбирает ту порцию, которую он хочет. Почти сразу же дети начнут понимать динамику этой игры. Если один из них совершает ошибку или пытается обмануть, то другой может его «оштрафовать», выбрав чашу побольше. Играть честно – в интересах обоих детей, но в особенности того, кто делит чашу. В этом сценарии проигрывает только обманщик.

Родителю даже не нужно использовать свою власть или обеспечивать справедливость. Родителю лишь нужно *обеспечивать соблюдение протокола*; до тех пор, пока дети не смогут избежать назначенных им ролей «делющего» и «выбирающего», протокол сам по себе обеспечивает справедливый результат без необходимости какого-либо вмешательства. Родитель не может занимать сторону любимчика или исказить исход.



Хотя печально известные битвы за чипы 1980-х годов четко иллюстрируют эту точку зрения, любое сходство между описанным выше сценарием и реальным детским опытом любого из авторов со своими двоюродными братьями совершенно случаен ... или нет?

## Примитивы безопасности как строительные блоки

Для того чтобы подобный протокол справедливости работал, должны существовать определенные гарантии, или *примитивы безопасности*, которые можно комбинировать для обеспечения соблюдения. Первый примитив безопасности – это строгая *временная упорядоченность/последовательность*: действие «деление» должно происходить до действия «выбор». Это не сразу очевидно, но если вы не сможете гарантировать, что действие А произойдет до действия В, тогда протокол развалится. Второй элемент безопасности – это *обязательство без возможности отказа*. Каждый ребенок должен определиться со своим выбором роли: либо делющий, либо выбирающий. Кроме того, как только деление завершено, делющий привязан к созданному им делению – он не может отказаться от этого выбора и повторить попытку.

Криптографические системы предлагают ряд примитивов безопасности, которые могут объединяться разными способами для строительства протокола справедливости. В дополнение к упорядоченности и обязательству мы также можем использовать целый ряд других инструментов:

- хеш-функции для генерирования отпечатков данных как форма обязательства или как основа для цифровой подписи;
- цифровые подписи для аутентификации, неразглашения и подтверждения права собственности на секрет;
- шифрование/дешифрование для ограничения доступа к информации только авторизованным участникам.

Это лишь небольшой список целого «зверинца» используемых средств обеспечения безопасности и криптографии. Все время изобретаются более простые примитивы и комбинации.

В нашем примере из реальной жизни мы увидели одну из форм протокола справедливости под названием «дели и выбирай». Это всего лишь один из множества различных протоколов справедливости, которые могут быть построены путем комбинирования строительных блоков примитивов безопасности различными способами. Но базовая закономерность всегда одна и та же: два или более участника взаимодействуют, не доверяя друг другу, выполняя ряд шагов, которые являются частью согласованного протокола. Шаги протокола устанавливают положительные и отрицательные стимулы для обеспечения

того, чтобы, если участники рациональны, обман был контрпродуктивным, а справедливость – автоматическим исходом.

Контроль за соблюдением не является необходимым для получения справедливых исходов – он необходим только для того, чтобы участники не нарушали согласованный протокол.

Теперь, когда вы понимаете эту базовую закономерность, вы начнете видеть его повсюду в системе Bitcoin, сети Lightning и многих других системах. Давайте рассмотрим несколько конкретных примеров далее.

## Пример протокола справедливости

Наиболее ярким примером протокола справедливости является консенсусный алгоритм системы Bitcoin под названием *Доказательство работы* (Proof of Work, аббр. PoW). В системе Bitcoin майнеры соревнуются за верификацию транзакций и их агрегирование в блоки. В целях обеспечения того, чтобы майнеры не обманывали без наделения их полномочиями, в Bitcoin используется подсистема положительных и отрицательных стимулов. Майнеры должны использовать электричество и выделять оборудование для выполнения «работы», которая встроена в качестве «доказательства» внутрь каждого блока. Это достигается благодаря свойству хеш-функций, при котором выходное значение случайно распределяется по всему диапазону возможных выходов. Если майнерам удастся произвести валидный блок достаточно быстро, то они получают вознаграждение, зарабатывая блочное вознаграждение за этот блок. Принуждение майнеров использовать много электроэнергии до того, как сеть рассмотрит их блок, означает, что у них есть стимул правильно проверять транзакции в блоке. Если они обманывают или совершают какую-либо ошибку, то их блок отклоняется, и электричество, которое они использовали, чтобы «доказать» это, тратится впустую. Никому не нужно заставлять майнеров производить валидные блоки; вознаграждение и наказание стимулируют их к этому. Протоколу лишь нужно обеспечить, чтобы принимались только валидные блоки с доказательством работы.

Закономерность протокола справедливости также можно найти во многих разных аспектах сети Lightning:

- те, кто финансирует каналы, обеспечивают, чтобы у них была подписана транзакция возврата средств, прежде чем опубликовать финансовую транзакцию;
- всякий раз, когда канал переводится в новое состояние, старое состояние «отзывается», обеспечивая, что если кто-либо попытается выполнить его широковещательную передачу, то он потеряет весь остаток средств и будет оштрафован;
- те, кто пересылают платежи, знают, что если они подтверждают/фиксируют пересылку средств вперед, то могут либо получить возврат, либо получить оплату от узла, предшествующего им.

Снова и снова мы видим эту закономерность. Соблюдение справедливых исходов не обеспечивается никакими органами власти. Они возникают как естественное следствие протокола, который поощряет справедливость и штрафует обман, протокола справедливости, в котором задействуются личные интересы, направляя их на справедливые исходы.

Система Bitcoin и сеть Lightning являются имплементациями протоколов справедливости. Тогда зачем нужна сеть Lightning? Разве системы Bitcoin недостаточно?

## МОТИВАЦИЯ ДЛЯ СЕТИ LIGHTNING

Bitcoin – это система, которая регистрирует транзакции в глобально реплицируемом публичном реестре. Каждая транзакция видна, валидируется и хранится каждым участвующим компьютером. Как нетрудно себе представить, это генерирует большой объем данных, и его трудно масштабировать.

По мере роста системы Bitcoin и спроса на транзакции число транзакций в каждом блоке увеличивается, пока в конечном итоге не достигает предельного размера блока. Как только блоки «заполняются», избыточные транзакции остаются ждать в очереди. Многие пользователи будут увеличивать сумму комиссионных, которые они готовы платить, чтобы купить место для своих транзакций в следующем блоке.

Если спрос продолжает опережать емкость сети, то все большее число транзакций пользователей остаются неподтвержденными. Конкуренция за комиссионные также увеличивает стоимость каждой транзакции, делая многие транзакции с меньшей стоимостью (например, микротранзакции) совершенно неэкономичными в периоды особенно высокого спроса.

В целях решения этой проблемы мы могли бы увеличить лимит на размер блока, чтобы создать пространство для большего числа транзакций. Увеличение в «предложении» блочного пространства приведет к более низкому ценовому равновесию для транзакционных комиссий.

Однако увеличение размера блока перекладывает затраты на операторов узлов и требует, чтобы они тратили больше ресурсов на валидирование и хранение блочной цепи. Поскольку блочные цепи являются эпидемическими протоколами обмена сообщениями, каждый узел должен знать и валидировать каждую отдельную транзакцию, которая происходит в сети. Более того, после валидации каждая транзакция и блок должны быть распространены на «соседей» узла, умножая потребность в емкости. Таким образом, чем больше размер блока, тем выше потребность в емкости, обработке и хранении каждого отдельного узла. Увеличение емкости транзакций в таком ключе приводит к нежелательному эффекту централизации системы за счет сокращения числа узлов и операторов узлов. Поскольку операторы узлов не получают компенсации за работу узлов, если работа узлов обходится очень дорого, то только несколько хорошо финансируемых операторов узлов будут продолжать выполнять узлы.

## Масштабирование блочных цепей

Как показывают несколько расчетов, побочные эффекты увеличения размера блока или уменьшения времени блока по отношению к централизации сети являются серьезными.

Предположим, что использование системы Bitcoin растет настолько, что сеть должна обрабатывать 40 000 транзакций в секунду, что является приблизительным уровнем транзакционной обработки сети Visa во время пиковой используемости.

Исходя из того, что на транзакцию в среднем приходится 250 байт, это приведет к потоку данных 10 мегабайт в секунду (МБ/с) или 80 мегабит в секунду (Мбит/с) только для того, чтобы иметь возможность получать все транзакции. Сюда не входят накладные расходы на трафик, связанные с пересылкой информации о транзакции другим одноранговым узлам. Хотя 10 МБ/с и не кажутся экстремальными в контексте высокоскоростных оптоволоконных и мобильных скоростей 5G, это фактически исключило бы любого, кто не может удовлетворить данное требование, из работающего узла, в особенности в странах, где высокопроизводительный интернет недоступен или недоступен широко.

У пользователей также есть много других требований к пропускной способности, и нельзя ожидать, что они будут тратить столько лишь на получение транзакций.

Более того, хранение этой информации локально привело бы к 864 гигабайтам в день. Это примерно один терабайт данных, или размер жесткого диска.

Верифицирование 40 000 подписей алгоритма цифровой подписи на основе эллиптической кривой (Elliptic Curve Digital Signature Algorithm, аббр. ECDSA) в секунду также едва выполнимо (см. соответствующую статью на StackExchange<sup>5</sup>), что делает скачивание начального блока (initial block download, аббр. IBD) блочной цепи Bitcoin (синхронизирование и верифицирование всего, начиная с генезисного (первичного) блока) практически невозможным без очень дорогого оборудования.

В то время как 40 000 транзакций в секунду кажутся большими, они достигают паритета с традиционными финансовыми платежными сетями только в пиковые моменты. Инновации в межмашинных платежах, микротранзакциях и других приложениях, скорее всего, повысят спрос на много порядков.

Проще говоря: масштабировать блочную цепь для валидирования транзакций по всему миру децентрализованным способом невозможно.

*Но что, если бы каждый узел не был бы обязан знать и проверять каждую отдельную транзакцию? Что, если бы существовал способ удерживать масштабируемые транзакции вне цепи без потери безопасности сети Bitcoin?*

В феврале 2015 года Джозеф Пун (Joseph Poon) и Тадеуш Дриджа (Thaddeus Dryja) предложили возможное решение проблемы масштабируемости системы Bitcoin, опубликовав работу под названием «Сеть Lightning в рамках системы Bitcoin: масштабируемые мгновенные платежи вне цепи» (The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments)<sup>6</sup>.

В (ныне устаревшем) техническом документе Пун и Дриджа подсчитали, что для того, чтобы система Bitcoin достигла 47 000 транзакций в секунду, обрабатываемых системой Visa на пике, потребуется 8 ГБ блоков. Это сделало бы опе-

<sup>5</sup> См. <https://bitcoin.stackexchange.com/questions/95339/how-many-bitcoin-transactions-can-be-verified-per-second>.

<sup>6</sup> Джозеф Пун и Тадеуш Дриджа. Сеть Lightning в рамках системы Bitcoin: масштабируемые мгновенные платежи вне цепи. ЧЕРНОВАЯ версия 0.5.9.2. 14 января 2016 года (Joseph Poon and Thaddeus Dryja. "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments". DRAFT Version 0.5.9.2. January 14, 2016). <https://lightning.network/lightning-network-paper.pdf>.

рирование узлом совершенно неприемлемым для кого бы то ни было, кроме крупных предприятий и операций промышленного уровня. Результатом стала бы сеть, в которой только несколько пользователей могли бы фактически валидировать состояние реестра. Система Bitcoin опирается на то, что пользователи сами валидируют реестр, не доверяя третьим лицам в явной форме, чтобы оставаться децентрализованной. Взимание комиссии с пользователей за оперирование узлами вынудило бы среднестатистического пользователя доверять третьим сторонам, чтобы узнавать состояние реестра, что в конечном итоге нарушило бы доверительную модель в системе Bitcoin.

Сеть Lightning предлагает новую сеть, второй слой, где пользователи могут осуществлять платежи друг другу в одноранговом режиме без необходимости публикации транзакции в блочной цепи системы Bitcoin для каждого платежа. Пользователи могут платить друг другу в сети Lightning столько раз, сколько захотят, без создания дополнительных транзакций Bitcoin или взимания комиссии внутри цепи. Они используют блочную цепь системы Bitcoin только для первоначальной загрузки биткойна в сеть Lightning и для расчетов, то есть для удаления биткойна из сети Lightning. Как следствие гораздо больше платежей Bitcoin может происходить вне цепи, при этом только транзакции начальной загрузки и окончательного расчета должны валидироваться и храниться узлами Bitcoin. Помимо снижения нагрузки на узлы, платежи в сети Lightning дешевле для пользователей, потому что им не нужно платить комиссию за блочную цепь, и более приватны для пользователей, потому что они не публикуются для всех участников сети и, кроме того, не хранятся постоянно.

Хотя сеть Lightning изначально была задумана для системы Bitcoin, она может быть имплементирована в любой блочной цепи, которая отвечает некоторым базовым техническим требованиям. Другие блочные цепи, такие как Litecoin, уже поддерживают сеть Lightning. Кроме того, несколько других блочных цепей разрабатывают аналогичные второслойные решения, чтобы помочь им масштабироваться.

## ОПРЕДЕЛЯЮЩИЕ ПРИЗНАКИ СЕТИ LIGHTNING

Сеть Lightning – это сеть, которая работает как протокол второго слоя поверх системы Bitcoin и других блочных цепей. Сеть Lightning обеспечивает быстрые, безопасные, приватные, бездоверительные и не требующие разрешения платежи. Вот несколько признаков сети Lightning:

- пользователи сети Lightning могут маршрутизировать платежи друг другу по низкой цене и в реальном времени;
- пользователям, которые обмениваются стоимостями через сеть Lightning, не нужно ждать подтверждения блока для платежей;
- как только платеж в сети Lightning завершен, обычно в течение нескольких секунд он является окончательным и не может быть отменен. Как и транзакция в системе Bitcoin, платеж в сети Lightning может быть возвращен только получателем;
- в отличие от внутрицепных транзакций Bitcoin, которые передаются широкоэмитально и проверяются всеми узлами сети, платежи, маршрути-

зируемые в сети Lightning, передаются между парами узлов и не видны всем, что обеспечивает гораздо большую конфиденциальность;

- в отличие от транзакций в сети Bitcoin, платежи, маршрутизируемые в сети Lightning, не нуждаются в постоянном хранении. Таким образом, Lightning потребляет меньше ресурсов и, следовательно, дешевле. Это свойство также имеет преимущества для конфиденциальности;
- сеть Lightning использует луковичную маршрутизацию, аналогичную протоколу, используемому сетью Tor (от англ. The Onion Router, луковичный маршрутизатор), работающей на основе технологии обеспечения конфиденциальности, в результате чего даже участвующие в маршрутизации платежа узлы напрямую знают только о своем предшественнике и преемнике в маршруте платежа;
- при использовании поверх системы Bitcoin сеть Lightning использует настоящий биткойн, который всегда находится во владении (хранении) и полностью контролируется пользователем. Lightning – это не отдельный токен или монета, это действительно биткойн.

## ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ СЕТИ LIGHTNING, ПОЛЬЗОВАТЕЛИ И ИХ ИСТОРИИ

В целях более глубокого понимания того, как на самом деле работает сеть Lightning и почему люди ее используют, мы проследим за рядом пользователей и их историями.

В наших примерах некоторые люди уже систему Bitcoin использовали, а другие в ней абсолютные новички. Каждый человек и его приведенная ниже история иллюстрируют один или несколько конкретных вариантов применения. Мы будем возвращаться к ним на протяжении всей этой книги:

### *Потребитель*

Алиса – пользователь системы Bitcoin, который хочет совершать быстрые, безопасные, дешевые и приватные платежи за небольшие розничные покупки. Она покупает кофе за биткойны, используя сеть Lightning.

### *Торговец*

Боб владеет кафе «Кофейня Боба». Внутрицепные платежи Bitcoin не масштабируются для малых сумм, таких как чашка кофе, поэтому он использует сеть Lightning для приема платежей Bitcoin почти мгновенно и за низкие комиссионные.

### *Бизнес по предоставлению программно-информационных услуг*

Чан – китайский предприниматель, который продает информационные услуги, связанные с сетью Lightning, а также системой Bitcoin и другими криптовалютами. Чан продает эти информационные услуги через интернет, осуществляя микроплатежи через сеть Lightning. Кроме того, Чан имплементировал службу поставщика ликвидности, которая арендует емкость входящего канала в сети Lightning, взимая небольшие комиссионные в биткойне за каждый период аренды.

*Игрок*

Дина – геймер-подросток из России. Она играет во много разных компьютерных игр, но ее любимые – это те, в которых есть «внутриигровая экономика», основанная на реальных деньгах. Играя в игры, она также зарабатывает деньги, приобретая и продавая виртуальные внутриигровые предметы. Сеть Lightning позволяет ей совершать небольшие сделки по внутриигровым предметам, а также зарабатывать небольшие суммы за выполнение квестов.

**Вывод**

В этой главе мы поговорили о фундаментальной концепции, которая лежит в основе как системы Bitcoin, так и сети Lightning: протоколе справедливости.

Мы рассмотрели историю сети Lightning и мотивы, лежащие в основе технических решений для масштабирования на втором слое для системы Bitcoin и других сетей, основанных на блочной цепи (блокчейне).

Мы изучили базовую терминологию, включая узел, платежный канал, транзакции внутри цепи и платежи вне цепи.

Наконец, мы познакомились с Алисой, Бобом, Чаном и Диной, за которыми будем следить на протяжении всей остальной части книги. В следующей главе мы встретимся с Алисой и рассмотрим ее мыслительный процесс, когда она выбирает кошелек Lightning и готовится совершить свой первый платеж Lightning, чтобы купить чашку кофе в кофейне Боба.



# Глава 2

## Приступаем к работе

В этой главе мы начнем с того, с чего начинает большинство людей, впервые сталкиваясь с сетью Lightning, – с выбора программного обеспечения для участия в экономике LN. Мы рассмотрим варианты выбора двух пользователей, которые представляют распространенный вариант использования сети Lightning, и будем учиться на примере. Алиса, посетительница кофейни, будет использовать кошелек Lightning на своем мобильном устройстве, чтобы купить кофе в кофейне Боба. Боб, торговец, будет использовать узел Lightning и кошелек для работы системы кассовых расчетов в своей кофейне, чтобы иметь возможность принимать платежи через сеть Lightning.

### Первый кошелек Lightning Алисы

Алиса – давний пользователь системы Bitcoin. Впервые мы встретились с Алисой в главе 1 книги «Освоение системы Bitcoin»<sup>7</sup>, когда она купила чашку кофе в кофейне Боба, используя транзакцию Bitcoin. Если вы еще незнакомы с тем, как работают транзакции Bitcoin, или нуждаетесь в освежении своих знаний, то, пожалуйста, ознакомьтесь с книгой «Освоение системы Bitcoin» или кратким описанием в приложении А.

Алиса узнала, что кофейня Боба недавно начала принимать платежи LN! Алисе не терпится узнать о сети Lightning и поэкспериментировать с ней; она хочет стать одним из первых клиентов LN Боба. Для этого Алиса сначала должна выбрать кошелек Lightning, который соответствует ее потребностям.

Алиса не хочет доверять хранение своего биткойна третьим лицам. Она узнала о криптовалюте уже достаточно, чтобы знать, как пользоваться кошельком. Ей также нужен мобильный кошелек, чтобы она могла использовать его для небольших платежей на ходу, поэтому она выбирает кошелек Eclair, популярный неопекаемый (некустодиальный)<sup>8</sup> мобильный кошелек Lightning. Давайте узнаем больше о том, как и почему она сделала этот выбор.

<sup>7</sup> *Андреас М. Антонопулос*. Освоение системы Bitcoin. 2-е изд., глава 1 (Andreas M. Antonopoulos, Mastering Bitcoin, 2nd Edition, Chapter 1, O'Reilly, <https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch01.asciidoc>).

<sup>8</sup> Неопекаемым, или некастодиальным (noncustodial), можно считать криптовалютный кошелек, сохраняющий за пользователем возможность полностью контролировать ключи и свои средства. К данной категории можно отнести аппаратные, мобильные, бумажные, настольные и веб-кошельки. – *Прим. перев.*

## Узлы LIGHTNING

Доступ к сети Lightning осуществляется посредством программных приложений, которые могут обмениваться данными по протоколу LN. Узел сети Lightning (узел LN или просто узел) – это программное приложение, обладающее тремя важными характеристиками. Во-первых, узлы Lightning являются кошельками, поэтому они отправляют и получают платежи через сеть Lightning, а также в сети Bitcoin. Во-вторых, узлы должны взаимодействовать на одноранговой основе с другими узлами Lightning, создавая сеть. Наконец, узлам Lightning также необходим доступ к блочной цепи Bitcoin (или другим блочным цепям для других криптовалют) с целью защиты средств, используемых для платежей.

Пользователи имеют наивысшую степень контроля, управляя своим собственным узлом Bitcoin и узлом Lightning. Однако узлы Lightning также могут использовать облегченный клиент Bitcoin, обычно именуемый упрощенной верификацией платежей (simplified payment verification, аббр. SPV), в целях взаимодействия с блочной цепью Bitcoin.

## Проводники LIGHTNING

Проводники LN – это полезные инструменты для показа статистики узлов, каналов и емкости сети.

Ниже приведен неисчерпывающий список:

- проводник 1ML<sup>9</sup> по сети Lightning;
- проводник ACINQ<sup>10</sup> по сети Lightning с причудливой визуализацией;
- проводник Amboss Space<sup>11</sup> по сети Lightning с метриками сообщества и интуитивно понятными визуализациями;
- проводник Fiatjaf<sup>12</sup> по сети Lightning с многочисленными диаграммами;
- проводник hashXP<sup>13</sup> по сети Lightning.

<sup>9</sup> См. <https://1ml.com/>.

<sup>10</sup> См. <https://explorer.acinq.co/>.

<sup>11</sup> См. <https://amboss.space/>.

<sup>12</sup> См. <https://ln.bigsun.xyz/>.

<sup>13</sup> См. <https://hashxp.org/lightning/node>.



Обратите внимание, что при использовании проводников Lightning, как и в случае с другими проводниками по блокам, конфиденциальность может быть проблемой. Если пользователи проявляют небрежность, то веб-сайт может отслеживать их IP-адреса и собирать данные об их поведении (например, интересующие пользователей узлы).

Также следует отметить, что поскольку нет глобального консенсуса относительно текущего графа Lightning или текущего состояния какой-либо существующей политики канала, пользователям никогда не следует полагаться на проводников по сети Lightning для получения самой актуальной информации. Кроме того, по мере того как пользователи открывают, закрывают и обновляют каналы, граф будет меняться, и отдельные проводники по сети Lightning могут быть не в курсе последних событий. Используйте проводники Lightning для визуализации сети или сбора информации, но не в качестве авторитетного источника того, что происходит в сети Lightning. Для того чтобы иметь авторитетное представление о сети Lightning, запустите свой собственный узел Lightning. Он построит граф каналов и соберет различную статистику, которую вы сможете просматривать с помощью веб-интерфейса.

## КОШЕЛЬКИ LIGHTNING

Термин «*кошелек Lightning*» несколько двусмыслен, поскольку он может описывать широкий спектр компонентов в сочетании с некоторым пользовательским интерфейсом. Наиболее распространенные компоненты программного обеспечения кошелька Lightning включают:

- хранилище ключей, в котором хранятся секреты, такие как приватные ключи;
- узел LN (узел Lightning), который взаимодействует в одноранговой сети, как описано ранее;
- узел Bitcoin, который хранит данные блочной цепи и взаимодействует с другими узлами Bitcoin;
- «карту» узлов и каналов, генерируемую на основе базы данных, которые объявлены в сети Lightning;
- менеджер каналов, который может открывать и закрывать каналы LN;
- систему в увеличенном масштабе, которая может отыскивать путь подсоединенных каналов от источника платежа до места назначения платежа.

Кошелек Lightning может содержать все эти функции, действуя как «полноценный» кошелек, не полагаясь на какие-либо сторонние службы. Либо один или несколько из этих компонентов могут полагаться (частично или полностью) на сторонние службы, которые выполняют эти функции.

*Ключевое* различие (преднамеренный каламбур) заключается в том, является функция хранилища ключей внутренней или переданной на аутсорсинг. В блочных цепях контроль над ключами определяет характер опекунства над средствами, о чем свидетельствует фраза «ваши ключи – ваши монеты; не ваши ключи – не ваши монеты». Любой кошелек, который передает управление ключами на аутсорсинг, называется *опекаемым*, или кустодияльным, кошельком, потому что третья сторона, действующая в качестве опекуна, конт-

ролирует средства пользователя, а не самого пользователя. Неопекаемый, или *самоопекаемый*, кошелек, для сравнения, – это кошелек, в котором хранилище ключей является частью кошелька, а ключи контролируются непосредственно пользователем. Термин «неопекаемый кошелек» просто подразумевает, что хранилище ключей является локальным и находится под контролем пользователя. Однако один или несколько других компонентов кошелька могут передаваться или не передаваться на аутсорсинг и полагаться на доверенные третьи стороны.

Блочные цепи, в особенности открытые блочные цепи, такие как Bitcoin, пытаются свести к минимуму или ослабить доверие к третьим сторонам и расширить возможности пользователей. Такой подход нередко именуется «бездоверительным», хотя больше подходит термин «минимально доверительный». В таких системах пользователь доверяет правилам программного обеспечения, а не третьим лицам. Поэтому вопрос контроля над ключами является принципиальным соображением при выборе кошелька Lightning.

Любой другой компонент кошелька Lightning предполагает аналогичные соображения по поводу доверия. Если все компоненты находятся под контролем пользователя, то степень доверия к третьим лицам сводится к минимуму, обеспечивая максимальную власть пользователю. Разумеется, это приводит к прямому компромиссу, потому что с этой властью приходит соответствующая ответственность за управление сложным программным обеспечением.

Каждый пользователь должен учитывать свои собственные технические навыки, прежде чем решать, какой тип кошелька Lightning использовать. Те, кто обладают сильными техническими навыками, должны использовать кошелек Lightning, который ставит все компоненты под непосредственный контроль пользователя. Тем, у кого меньше технических навыков, но есть желание контролировать свои средства, следует выбрать нестандартный кошелек Lightning. Часто доверие в этих случаях связано с конфиденциальностью. Если пользователи решают передать некоторые функции на аутсорсинг третьей стороне, то они обычно отказываются от некоторой конфиденциальности, поскольку третья сторона будет узнавать о них некоторую информацию.

Наконец, те, кто ищет простоту и удобство, даже в ущерб контролю и безопасности, могут выбрать опекаемый кошелек Lightning. Это наименее сложный с технической точки зрения вариант, но он подрывает доверительную модель криптовалюты и поэтому должен рассматриваться только как ступенька к большему контролю и самообеспечению.

Кошельки можно характеризовать или классифицировать несколькими способами. Наиболее важные вопросы, которые следует задать о конкретном кошельке, таковы:

1. Имеет ли этот кошелек Lightning полноценный узел Lightning или же он использует сторонний узел Lightning?
2. Имеет ли этот кошелек Lightning полноценный узел Bitcoin или же он использует сторонний узел Bitcoin?
3. Хранит ли этот кошелек Lightning свои собственные ключи под контролем пользователя (самоопека) или ключи хранятся сторонним опекуном?



Если кошелек Lightning использует сторонний узел Lightning, то именно этот сторонний узел Lightning решает, как взаимодействовать с системой Bitcoin. Следовательно, использование стороннего узла Lightning подразумевает, что вы также используете сторонний узел Bitcoin. Только когда кошелек Lightning использует свой собственный узел Lightning, существует выбор между полноценным узлом Bitcoin и сторонним узлом Bitcoin.

На самом высоком уровне абстракции вопросы 1 и 3 являются самыми элементарными. Из этих двух вопросов можно вывести четыре возможные категории. Эти четыре категории можно поместить в квадрант, как показано в табл. 2-1. Но помните, что это всего лишь один из способов классифицирования кошельков Lightning.

**Таблица 2-1.** Квадрант кошельков Lightning

	Полноценный узел Lightning	Сторонний узел Lightning
Самоопекаемый	Вопрос 1: высокие технические навыки, наименьшее доверие к третьим сторонам, наибольшее число разрешений	Вопрос 2: технические навыки ниже среднего, доверие к третьим сторонам ниже среднего, требуются некоторые разрешения
Опекаемый	Вопрос 3: технические навыки выше среднего, доверие к третьим сторонам выше среднего, требует некоторых разрешений	Вопрос 4: низкие технические навыки, высокое доверие к третьим сторонам, наименьшее число разрешений

Квадрант 3 (вопрос 3), где используется полный узел Lightning, но ключи хранятся у опекуна, в настоящее время не распространен. Будущие кошельки из этого квадранта могут позволить пользователю беспокоиться об операционных аспектах своего узла, но затем делегировать доступ к ключам третьей стороне, которая в основном использует холодное хранилище.

Кошельки Lightning могут устанавливаться на различных устройствах, включая ноутбуки, серверы и мобильные устройства. Для оперирования полноценным узлом Lightning вам потребуется использовать сервер или настольный компьютер, поскольку мобильные устройства и ноутбуки обычно недостаточно мощны с точки зрения емкости, обработки, времени автономной работы и подключаемости.

Категория сторонних узлов Lightning снова может быть подразделена:

#### *Облегченный*

Это означает, что кошелек не управляет узлом Lightning и, следовательно, должен получать информацию о сети Lightning через интернет от чужого узла Lightning.

#### *Никакой*

Это означает, что не только узел Lightning управляется третьей стороной, но и большая часть кошелька управляется третьей стороной в облаке. Этот кошелек предназначен для хранения средств, в котором кто-то другой опекает средства.

Эти подкатегории используются в табл. 2-2.

Другие термины, которые нуждаются в пояснении в табл. 2-2 в столбце «Узел Bitcoin», таковы:

#### *Neutrino*

Этот кошелек не управляет узлом Bitcoin. Вместо этого доступ к узлу Bitcoin, управляемому кем-то другим (третьей стороной), осуществляется по протоколу Neutrino.

#### *Electrum*

Этот кошелек не управляет узлом Bitcoin. Вместо этого доступ к узлу Bitcoin, управляемому кем-то другим (третьей стороной), осуществляется по протоколу Electrum.

#### *Bitcoin Core*

Это (референтная) имплементация узла Bitcoin.

#### *btcd*

Это еще одна имплементация узла Bitcoin.

В табл. 2-2 мы видим несколько примеров популярных в настоящее время приложений узлов и кошельков Lightning для различных типов устройств. Список отсортирован сначала по типу устройства, а затем в алфавитном порядке.

**Таблица 2-2.** Примеры популярных кошельков Lightning

Приложение	Устройство	Узел Lightning	Узел Bitcoin	Хранилище ключей
Blue Wallet	Мобильное	Никакой	Никакой	Опекаемый
Breez Wallet	Мобильное	Полноценный узел	Neutrino	Самоопекаемый
Eclair Mobile	Мобильное	Облегченный	Electrum	Самоопекаемый
Lntxbot	Мобильное	Никакой	Никакой	Опекаемый
Muun	Мобильное	Облегченный	Neutrino	Самоопекаемый
Phoenix Wallet	Мобильное	Облегченный	Electrum	Самоопекаемый
Zeus	Мобильное	Полноценный узел	Bitcoin Core/btcd	Самоопекаемый
Electrum	Настольное	Полноценный узел	Bitcoin Core/ Electrum	Самоопекаемый
Zap Desktop	Настольное	Полноценный узел	Neutrino	Самоопекаемый
c-lightning	Сервер	Полноценный узел	Bitcoin Core	Самоопекаемый
Eclair Server	Сервер	Полноценный узел	Bitcoin Core/ Electrum	Самоопекаемый
lnd	Сервер	Полноценный узел	Bitcoin Core/btcd	Самоопекаемый

## Тестовая сеть Bitcoin

Система Bitcoin предлагает альтернативную цепь для целей тестирования, именуемую *testnet*, в отличие от «обычной» цепи Bitcoin, которая называется *mainnet*. В *testnet* валютой является *testnet-биткойн* (tBTC), который представляет собой бесполезную копию биткойна, используемую исключительно для тестирования. Каждая функция системы Bitcoin воспроизводится в точности, но деньги ничего не стоят, так что вам буквально нечего терять!

Некоторые кошельки Lightning также могут работать в *testnet*, позволяя совершать платежи Lightning с помощью биткойна *testnet*, не рискуя реальными

средствами. Это отличный способ безопасно поэкспериментировать с сетью Lightning. Кошелек Eclair Mobile, который Алиса использует в этой главе, является одним из примеров кошелька Lightning, поддерживающего работу testnet.

Немного tBTC для игры можно получить с помощью вентиля биткойнов testnet, который выдает бесплатные tBTC по запросу. Вот несколько вентиляей testnet:

<https://coinafaucet.eu/en/btc-testnet>

<https://testnet-faucet.mempool.co>

<https://bitcoinafaucet.uo1.net>

<https://testnet.help/en/btcfacuet/testnet>

Все примеры в этой книге можно точно воспроизвести в testnet с tBTC, так что если пожелаете, то можете последовать по пятам, не рискуя реальными деньгами.

## УРАВНОВЕШИВАНИЕ СЛОЖНОСТИ И КОНТРОЛЯ

Кошельки Lightning должны соблюдать тщательное равновесие между сложностью и контролем со стороны пользователя. Те, которые дают пользователю наибольший контроль над своими средствами, высочайшую степень конфиденциальности и наибольшую независимость от сторонних служб, с неизбежностью являются более сложными и трудными в эксплуатации. По мере развития технологий некоторые из этих компромиссов станут менее очевидными, и пользователи смогут получать больший контроль без дополнительных сложностей. Однако на данный момент разные компании и проекты занимаются разведкой разных позиций в этом спектре сложности управления, надеясь найти «золотую» середину для пользователей, на которых они ориентированы.

Выбирая кошелек, имейте в виду, что даже если вы не видите этих компромиссов, они все равно существуют. Например, многие кошельки будут пытаться снимать бремя управления каналами со своих пользователей. Для этого они вводят центральные хабовые узлы, к которым автоматически подключаются все их кошельки. Хотя этот компромисс упрощает пользовательский интерфейс и пользовательский опыт, он вводит единую точку отказа (point of failure, аббр. SPoF), поскольку эти хабовые узлы становятся незаменимыми для работы кошелька. Более того, использование такого «хаба» может снизить конфиденциальность пользователя, так как хаб знает отправителя и потенциально (при построении маршрута платежа от имени пользователя) также получателя каждого платежа, производимого кошельком пользователя.

В следующем далее разделе мы вернемся к нашему первому пользователю, Алисе, и рассмотрим ее первую настройку кошелька Lightning. Она выбрала кошелек, который является более изоциренным, чем более простые опекаемые кошельки. Это позволяет нам показать некоторые основные сложности и представить некоторые внутренние механизмы продвинутого кошелька. Вы, возможно, обнаружите, что ваш первый идеальный кошелек ориентирован на простоту использования, принимая некоторые компромиссы между контролем и конфиденциальностью. Или же, вероятно, вы более опытный пользователь и хотите выполнять свои собственные узлы Lightning и Bitcoin как часть технического решения для вашего кошелька.