Кэтрин

Оглавление

Содержание

Предисловие

Теме криптографии на эллиптических кривых (ECC) посвящено великое множество книг. Но мало среди них таких, где приводится код или объяснения того, как код работает. Эта книга поможет вам понять и математику, и реализующий ее код. В центре внимания – практические применения, поэтому математических доказательств вы здесь не найдете. Если код работает, то математика, очевидно, правильна. Если код не работает, то в математике, очевидно, есть какой-то дефект. И это единственное доказательство, которое нам нужно.

Математику эллиптических кривых изучают вот уже больше 300 лет, и до сих пор это область активных исследований. Мы, конечно, не можем представить эту область во всей глубине. А рассмотрим только то, что лежит в основе реализации стойкой криптографической безопасности. Хотя термин «вулкан изогений» кажется мне очень образным, а сама теория пленительной, но нам туда не надо.

Криптография на эллиптических кривых используется для обмена ключами и цифровых подписей свыше 30 лет. В последние несколько лет она применялась для формирования множественных цифровых подписей в блокчейне, а также для доказательств с нулевым разглашением. Метод нулевого разглашения тоже существует более 30 лет. Использование доказательств с нулевым разглашением в блокчейне для доказательства транзакции без раскрытия суммы теперь – с помощью спаривания точек на эллиптической кривой – стало возможным с применением гораздо меньшего количества данных.

В 2015 году профессора Коблиц и Менезес опубликовали работу, в которой описали недоразумение, возникшее в связи с высказываниями Агентства национальной безопасности (АНБ) о криптографии на эллиптических кривых (см. библиографию). Причиной путаницы стали комментарии АНБ по поводу способности кванто-

вых компьютеров взломать ECC. В настоящее время представляется крайне преждевременным рассуждать о том, что квантовые компьютеры способны вскрыть криптографию на эллиптических кривых, и я хочу уделить минутку, чтобы объяснить, почему.

В квантовых компьютерах кубиты используются как транзисторы в обычных (классических) компьютерах. Но если транзистор может быть либо включен, либо выключен, то состояние кубита – суперпозиция «вкл» и «выкл». Для создания вентиля необходимо несколько кубитов – точно так же, как для создания вентиля NAND требуется несколько транзисторов. За прошедшие пять лет число кубитов на единичной площади удваивалось каждый год – это лучше, чем закон Мура.

В 2017 году Roetteler et al. опубликовали работу о вскрытии ECC с помощью квантовых компьютеров. Авторы показали, как создать вентили, решающие задачу дискретного логарифмирования на эллиптических кривых с целью нахождения ключа. Их таблица вентилей, преобразованная в степени двойки, приведена в третьем столбце табл. 1. В первом столбце показан классический уровень безопасности в битах, а во втором – длина простого числа в ECC, необходимая для обеспечения того же уровня безопасности.

Таблица 1 Количество кубитов, необходимое для вскрытия ECC

Безопасность	Длина простого числа	Вентилей
80	160	1.7×2^{34}
128	256	1.8×2^{36}
256	512	1.0×2^{40}

Это количественное выражение путаницы, на которое указано в работе Koblitz and Menez (2015). В настоящее время самое большое из известных устройств содержит 1.7×2^8 кубитов. Это примерно 2^6 вентилей. Следовательно, пройдет по меньшей мере 30 лет, прежде чем квантовые компьютеры действительно смогут вскрыть ECC, – и это в предположении, что на протяжении всего этого времени плотность кубитов будет ежегодно удваиваться.

Если вам меньше 40 лет, то пора уже начать изучать, как работают квантовые компьютеры. Вполне возможно, что они окажут влияние на вашу карьеру. Да и вообще, это очень интересная и познавательная тема! Но как минимум ближайшие 20 лет криптография на эллиптических кривых может чувствовать себя в безопасности, так что затраты времени на ее изучение не пропадут втуне.

Рассмотрим первые два столбца табл. 1. Безопасность тесно связана с методом шифрования одного секретного ключа, например алгоритмом Advanced Encryption Standard (AES). Чтобы вскрыть AES, нужно перебрать все возможные ключи. В среднем придется проверить половину всех ключей, т. е. совершить $2^{(\text{безопасность}-1)}$ попыток. Под

длиной понимается количество битов в простом числе, необходимое для достижения такого уровня безопасности при использовании эллиптических кривых. Разница в два раза объясняется методом решения задачи дискретного логарифмирования на эллиптических кривых.

Я уже дважды употребил фразу «задача дискретного логарифмирования на эллиптических» кривых, но что она означает? Деталями мы займемся в главе 4, а пока скажу только, что если имеется точка P на эллиптической кривой и мы умножим ее на некоторую константу k, то получится новая точка на кривой. Если вы знаете только точки P и Q, то задача дискретного логарифмирования заключается в нахождении k. В лучших известных на сегодня алгоритмах для этого требуется число попыток, равное квадратному корню из простого числа. Если длина простого числа – 256 бит, то длина квадратного корня из него – 128 бит, так что в среднем для нахождения k необходимо 2^{128} попыток. Это то же самое, что 128-битовая безопасность в AES.

Существует масса способов повысить эффективность и скорость описанных в книге вычислений. Но важнее сначала понять базовые операции, а затем, если ситуация того требует, изучить, какие есть варианты для уменьшения скорости реакции. В этой книге основной упор делается на безопасность. То, что делает ваш код быстрее, может облегчить жизнь противнику. Чтобы не дать развернуть плацдарм для атаки, в нашем коде будут использоваться простые числа на каждом уровне. Это медленно, зато очень безопасно.

Благодарности

Когда Трой Дрейер связался со мной на предмет написания еще одной книги по криптографии на эллиптических кривых для издательства Manning, я только-только уволился с прежней работы. Я потратил много времени на изучение самой разной математики, большая часть которой не попала в эту книгу. К счастью, мое предложение о будущей книге попало в руки знающих специалистов, которые указали, что одна из основных тем, которую я собирался обсудить, утратила актуальность буквально за неделю до подачи предложения. Поэтому я сократил план и отправил Трою новое предложение. Оно выдержало испытание, и я потратил довольно много времени на сочинение сколько-нибудь пригодного кода и текста.

Мне также повезло с редактором, Мариной Майклз, которая все время просила меня не быть настолько лаконичным. Хотя в наших разговорах часто возникала тема истории, Марина приложила немало усилий, чтобы объяснить мне, как нужно писать. Надеюсь, я чему-то научился.

Марк Биссен, технический редактор книги и заслуженный исследователь-рационализатор в области приборостроения в Висконсинском университете в Мэдисоне, очень помог мне в плане математической теории, задавая уйму отличных вопросов. Есть много способов описать одну формулу, но некоторые понятнее других.

Я благодарен д-ру Маню Дрийверсу, исследователю-криптографу в компании Dfinity Foundation, объяснившему мне нотацию, использованную в его статье, на которую я ссылаюсь в главе 18 (Boneh et al., 2018). Это определенно помогло мне написать код, в котором я больше уверен.

Джанлуиджи Спаньоло, корректор, проверил весь мой код и внес много предложений по улучшению как математического описания, так и форматирования. Замечание всех рецензентов были так или иначе учтены в книге. Я высоко ценю время, потраченное ими на

чтение и комментирование черновых вариантов текста. Спасибо вам, Грег Маклин, Клиффорд Т., Серджио Арбео, Марко Массенцио, Адриан Кукош, Рани Шарим, Дэвид Романо, Ален Кунио, Юрген Хётцель, Максим Волгин, Дотри Микаэль, Джанпьеро Гранателла, Джузеппе Денора, Грегори Пикколи, Тим ван Дёрзен, Питер Мэйхон, Роман Жужа, Рави Киран Бамиди и Ник Декроос. Ваши предложения помогли сделать книгу лучше. Весь коллектив издательства Manning принимал участие в подготовке этой книги в многочисленных современных форматах и последующей рекламе.

Огромное спасибо всем – я высоко ценю помощь каждого.

Об этой книге

Эллиптические кривые над конечным полем – сложная тема. Прийти к пониманию спаривания точек на эллиптических кривых можно разными путями. Мне эта книга представляется тропой, вьющейся по склону горы. С нее открывается великолепный вид на сотни других вершин, которые вы можете покорить. Но если не добраться до смотровой площадки, то ни на одну из них не подняться, да что там – даже не увидеть. В библиографии есть указания на другие пути, но тропа к пониманию спаривания, выбранная в этой книге, узка.

В книге приводится введение в математику спаривания на эллиптической кривой, ориентированное на программистов-криптографов. Имея в активе базовые знания линейной алгебры и годичный опыт программирования, читатель сможет понять описанные подпрограммы и перевести их на свой любимый язык. Хотя знание математики над конечным полем было бы исключительно полезно, в книге приведено пошаговое введение в эту дисциплину и ее связи с эллиптическими кривыми.

Если вы не обладатель степени PhD по математике, то, чтобы в полной мере понять и оценить примеры приложений в последних двух главах, вам придется прочесть книгу целиком. В каждой главе информация из предыдущих глав используется для написания кода, который понадобится в последующих.

Структура книги

Глава 1 содержит более подробное введение в книгу в целом. Последующий текст разделен на три части.

Часть I включает главы 2–6. Начав с простых полей в главе 2, мы напишем функции, используемые в последующих главах.

В главе 3 дается введение в эллиптические кривые над конечным полем и обсуждается, почему медленный код лучше с точки зрения

безопасности, особенно во встраиваемых системах. Я также объясню, как произвольную информацию можно преобразовать в точку на эллиптической кривой.

В главах 4 и 5 мы рассмотрим основные алгоритмы обмена ключами и цифровых подписей.

В главе 6 описывается выбор кривой и программы для нахождения хороших кривых. Большинство этих программ нужно выполнить только один раз для инициализации безопасной системы. Получив хорошие кривые, их уже не нужно изменять.

Часть II включает главы 7–12. В каждой главе рассматривается одна математическая тема и связанные с ней подпрограммы. В этих главах закладываются основы, необходимые для понимания математики спаривания. Одна из основных причин, почему спаривания не используются более широко, – сложность стоящей за ними математики. Спору нет, математика действительно глубокая. Но большая часть этой глубины не имеет прямого отношения к тому, что нам нужно знать, а мы сосредоточимся на тех деталях, которые помогут сделать работу.

Часть III включает главы 13–19. В главе 13 мы будем заниматься расширениями конечных полей. С нашей точки зрения, это просто полиномы с кучей полезных свойств. В своем изложении идей я воспользуюсь очень малым простым числом для создания модельной кривой и перечислю все точки на этой кривой и ее расширении. Так мы проясним многие термины, которые математики считают само собой разумеющимися.

Обычно мы стремимся сделать «степень вложения» очень большим числом, чтобы помешать противнику преобразовать эллиптическую кривую в теоретико-числовую задачу дискретного логарифмирования. В случае пригодных для спаривания кривых мы требуем, чтобы кривая имела низкую степень вложения. В главе 14 обсуждается математика того, как можно создать расширение поля с низкой степенью вложения, используя метод комплексного умножения.

В главе 15 мы узнаем о математике спаривания точек на эллиптических кривых. Операция спаривания принимает две точки на эллиптической кривой и образует значение, не являющееся точкой. На самом деле это корень степени n из единицы.

Далее в главах 16 и 17 обсуждаются различные виды спаривания. У спаривания Вейля есть свойства, одновременно полезные и вызывающие проблемы в разных приложениях. То же самое верно в отношении спаривания Тейта. Есть много других видов спаривания, более эффективных, но, зная эти два, вы без труда разберетесь в остальных методах.

В главе 18 мы рассмотрим одно из основных применений спаривания в блокчейне: множественные цифровые подписи. Множест-

венная цифровая подпись – это расширение подписи BLS, которое допускает одношаговую проверку многих подписей документа.

Наконец, в главе 19 рассматривается современное использование спаривания в методе SNARK с нулевым разглашением. Акроним выглядит весьма уместно, если вспомнить поэму Льюиса Кэрролла «Охота на Снарка». Способность доказать знание транзакции, не раскрывая, кто выполнил перевод и что именно было переведено, позволяет поддерживать корректность анонимных блокчейнов. Это очень действенный математический инструмент.

В приложениях объясняется, как получить библиотеки программ, используемые в этой книге, а также приводятся подробные сведения о гильбертовых полиномах классов из главы 14. Весь код можно найти на GitHub по адресу https://github.com/drmike8888/Elliptic-curve-pairings. В репозитории есть дополнительные программы и их результаты, особенно относящиеся к главам 13 и 16 и помогающие лучше понять расширения полей и спаривание.

Почему C?

Одно из основных применений криптографические протоколы на основе эллиптических кривых находят во встраиваемых системах, таких как смарткарты и аппаратные активационные ключи. Небольшие мощные процессоры и программируемые пользователем вентильные матрицы (ППВМ) стоят дешево, но и память у них ограничена. Обычно эти устройства программируются на языках низкого уровня типа C и Verilog. Я в основном занимался встраиваемыми системами и потому накопил большой опыт работы на C. У языка C много преимуществ в части работы с битами и способности изменять семантику указателя – на строку или на число. В этой книге используются стандартные приемы программирования на C в различных функциях, где участвуют указатели и массивы. Я считаю их преимуществом, поскольку они позволяют эффективно контролировать вычисления.

Весь приведенный в книге код можно откомпилировать с помощью GNU gcc. Версия компилятора не играет роли. Всё было написано в Ubuntu Linux и должно компилироваться и компоноваться любой версией gcc для Linux. Проверьте, что версии библиотек GMP и PARI, с которыми вы будете компоновать программы, совместимы с вашим компилятором. Код очень общий, в нем нет никаких специализированных системных вызовов.

О примерах кода

В этой книге много примеров кода как в пронумерованных листингах, так и в виде небольших фрагментов прямо в тексте. В обоих слу-