

ВВЕДЕНИЕ

“Мне не оставалось ничего иного, как подслушивать, ведь мы занимались прослушкой”.

“Анонимус”



Если бы мы могли перенестись в начало 1990-х годов, то из-за главы 23, в которой реализуется алгоритм RSA, книгу нельзя было бы вывезти за пределы США. Поскольку сообщения, зашифрованные с помощью алгоритма RSA, не поддавались взлому, экспорт такого рода криптографических систем считался вопросом национальной безопасности и требовал специального разрешения Госдепа США. Фактически на законодательном уровне стойкие криптографические системы подчинялись тем же регуляторным нормам, что и танки, ракеты, огнеметы...

В 1990 году Даниэль Бернштейн, студент Калифорнийского университета в Беркли, захотел опубликовать академическую статью, включающую исходный код его криптосистемы Snuffle. Государственные органы США проинформировали его о том, что прежде, чем он сможет опубликовать исходный код, он должен получить лицензию на торговлю оружием. Ему также было сказано, что в случае подачи им заявки на экспортную лицензию он получит отказ ввиду высокой надежности этой технологии.

В судебном иске против Соединенных Штатов интересы Даниэля Бернштейна представляла молодая правозащитная организация — Фонд электронных рубежей (Electronic Frontier Foundation). Впервые в истории суд постановил, что на программный код распространяется действие первой поправки к Конституции США в части защиты свободы слова и что

ограничение пересылки зашифрованных сообщений законами об экспортном контроле нарушает права Бернштейна.

В наши дни на стойких криптосистемах базируется значительная часть мировой экономики. От них зависят сайты электронной торговли по всему миру, через которые ежедневно совершаются миллионы покупок. Опасения разведслужб относительно того, что зашифрованное программное обеспечение похоронит национальную безопасность, оказались беспочвенными.

Однако в начале 1990-х годов свободное распространение знаний (что и пытается делать автор книги) довело бы вас до тюрьмы за незаконную торговлю оружием.

Для кого предназначена эта книга

Обучению основам криптографии посвящено множество книг. Есть ряд книг, в которых новичков учат взламывать шифры. Но нет ни одной книги, в которой новичков учили бы писать компьютерные программы, способные взламывать шифры. Данная книга восполняет этот пробел.

Книга предназначена для тех, кто интересуется шифрованием, взломом шифров и криптографией. Все шифры, рассматриваемые в данной книге (за исключением криптосистем с открытым ключом, которым посвящены главы 23 и 24), существуют уже много столетий, и для их взлома достаточно вычислительных мощностей любого современного ноутбука. Ни одна организация и ни одно частное лицо уже не пользуется этими шифрами, однако в процессе их изучения вы освоите основы криптографии и узнаете, как хакеры взламывают слабые шифры.

Примечание

Изучение шифров, рассматриваемых в данной книге, станет увлекательным занятием, но помните, что они не обеспечивают подлинную безопасность. Не пытайтесь применять приведенные в книге программы для защиты собственных файлов. И вообще возьмите за правило не доверять шифрам, которые сами создаете. Профессиональные криптосистемы исследуются годами, прежде чем получают одобрение со стороны экспертов в области криптографии.

Книга ориентирована в том числе на новичков в программировании. Здесь излагаются основы программирования на Python — одного из лучших языков для начинающих. Python характеризуется настолько плавной кривой обучения, что освоить его смогут новички любого возраста, а его возможности удовлетворяют запросам даже самых требовательных про-

фессионалов. Python выполняется в среде Windows, macOS, Linux и даже Raspberry Pi, причем он доступен для свободной загрузки и использования (соответствующие указания будут приведены далее).

В литературе по криптографии часто употребляется термин *хакер*. Это слово имеет двойкий смысл. Оно может означать лицо, которое пытается досконально изучить систему (например, пошаговую процедуру шифрования или программный код) для того, чтобы преодолеть ее ограничения и творчески ее модифицировать. В то же время слово *хакер* также может означать преступника, который взламывает компьютерные системы и получает доступ к конфиденциальной информации, нанося ущерб. В данной книге, говоря о хакерах и взломе шифров, мы придерживаемся первого определения. Быть хакером — это круто! А вот преступники — это те, кто самоутверждаются за счет других.

Структура книги

В первых нескольких главах читатели познакомятся с основами Python и азами криптографии. В последующих главах поочередно объясняется, как запрограммировать тот или иной шифр и как его взломать. Кроме того, каждая глава завершается контрольными вопросами, которые помогут проверить, насколько хорошо вы усвоили прочитанный материал.

- **Глава 1 “Инструменты «бумажной» криптографии”.** Обсуждаются простейшие инструменты “бумажных” криптотехнологий и демонстрируются способы шифрования, существовавшие в докомпьютерную эпоху.
- **Глава 2 “Программирование в интерактивной оболочке”.** Объясняется, как применять интерактивную оболочку для построчного выполнения кода Python.
- **Глава 3 “Строковый тип данных и написание программ”.** Рассказывается о том, как писать программы, и вводится строковый тип данных, используемый во всех программах в книге.
- **Глава 4 “Обратный шифр”.** Показано, как написать простую программу шифрования.
- **Глава 5 “Шифр Цезаря”.** Описывается простейший шифр, известный человечеству уже несколько тысяч лет.
- **Глава 6 “Взлом шифра Цезаря методом грубой силы”.** Демонстрируется, как можно дешифровать сообщение, не имея ключа шифрования.

- **Глава 7 “Шифрование с помощью перестановочного шифра”.** Рассматривается программа, шифрующая сообщения с помощью перестановочного шифра.
- **Глава 8 “Дешифрование перестановочного шифра”.** Продолжение темы перестановочного шифра, только на этот раз демонстрируется процедура дешифрования сообщений.
- **Глава 9 “Написание тестов”.** Введение в методики тестирования программ.
- **Глава 10 “Шифрование и дешифрование файлов”.** Объясняется, как писать программы, осуществляющие чтение и запись файлов.
- **Глава 11 “Программное распознавание английских слов”.** Описывается, как с помощью компьютера распознавать предложения на английском языке.
- **Глава 12 “Взлом перестановочного шифра”.** Применяем полученные в предыдущих главах знания для взлома перестановочного шифра.
- **Глава 13 “Аффинное шифрование с помощью модульной арифметики”.** Рассматриваются математические основы аффинного шифрования.
- **Глава 14 “Программирование аффинного шифра”.** Демонстрируется написание программы аффинного шифрования.
- **Глава 15 “Взлом аффинного шифра”.** Объясняется, как написать программу для взлома аффинного шифра.
- **Глава 16 “Программирование простого подстановочного шифра”.** Рассматривается программа, реализующая шифрование на основе подстановочного шифра.
- **Глава 17 “Взлом простого подстановочного шифра”.** Объясняется, как написать программу для взлома простого подстановочного шифра.
- **Глава 18 “Программирование шифра Виженера”.** Описывается более сложный вариант подстановочного шифра — шифр Виженера.
- **Глава 19 “Частотный анализ”.** Исследуется частотная структура английских слов и демонстрируется, как применить эту информацию для взлома шифра Виженера.
- **Глава 20 “Взлом шифра Виженера”.** Рассматривается программа для взлома шифра Виженера.
- **Глава 21 “Одноразовый шифроблокнот”.** Обсуждается концепция одноразовых шифроблокнотов и приводится математическое обоснование того, почему такой блокнот невозможно взломать.

- **Глава 22 “Нахождение и генерирование простых чисел”.** Рассказывается, как написать программу для быстрого определения того, является ли число простым.
- **Глава 23 “Генерирование ключей для криптосистем с открытым ключом”.** Обсуждаются принципы шифрования с открытым ключом и рассматривается программа, генерирующая открытый и закрытый ключи.
- **Глава 24 “Программирование шифров с открытым ключом”.** Объясняется, как написать программу шифрования с открытым ключом, который невозможно взломать обычными методами.
- **Приложение А “Отладка кода Python”.** Демонстрируется, как находить и устранять ошибки в программном коде с помощью отладчика IDLE.
- **Приложение Б “Ответы на контрольные вопросы”.** Содержит ответы на упражнения к главам.

Как пользоваться книгой

Данная книга отличается от остальных книг по программированию тем, что основное внимание в ней уделено исходному коду готовых программ. Вместо того чтобы обучать читателей основам программирования, предлагая им самостоятельно понять, как решать конкретные задачи, автор демонстрирует полноценные программные решения и подробно объясняет, как они работают.

Главы желательно читать по порядку, так как концепции программирования объясняются на основе материала предыдущих глав. Однако язык Python настолько прост для изучения, что уже после прочтения нескольких начальных глав вы, вероятно, сможете перейти к любой из последующих глав и легко понять, как работает приведенный в ней программный код. Если же почувствуете, что нить изложения ускользает от вас, вы всегда сможете вернуться к предыдущим главам.

Ввод исходного кода

Читателям настоятельно рекомендуется *вводить весь исходный код вручную*, так как это определенно поможет лучше понять его.

Вводить начальные номера строк не нужно. Эти номера не являются частью самих программ и служат исключительно для того, чтобы можно было ссылаться на ту или иную строку в тексте книги. А вот собственно программный код, следующий за номерами строк, необходимо вводить

именно в таком виде, в каком он приведен в книге, с учетом регистра букв.

Кроме того, обратите внимание на то, что некоторые строки кода выделяются отступами длиной четыре, восемь или более пробелов. Во избежание ошибок следите за строгим соблюдением количества пробелов в начале каждой строки.

Если не хотите самостоятельно вводить исходный код программ, загрузите его на сайте издательства:

<https://nostarch.com/crackingcodes>

Исходные файлы доступны также на сайте издательства “Диалектика” (в этих файлах исправлен ряд авторских ошибок):

<http://www.williamspublishing.com/Books/978-5-907203-02-0.html>

Исправление опечаток

Самостоятельный ввод исходного кода программ действительно помогает в процессе изучения Python, но это чревато появлением опечаток, которые не всегда легко обнаружить, особенно если программа достаточно большая.

Чтобы облегчить и ускорить поиск ошибок во введенном вами исходном коде, скопируйте и вставьте текст программы в окно онлайн-утилиты Diff, доступной по следующим адресам:

<https://inventwithpython.com/cracking/diff/>

<https://inventwithpython.com/hacking/diff/>

В окне утилиты выделяются любые различия между исходным кодом, приведенным в книге, и кодом, набранным вами.

Правила оформления листингов, принятые в книге

Книга задумана не как справочник, а как пособие для начинающих, поэтому стиль оформления листингов иногда может не соответствовать устоявшейся практике. Это было сделано сознательно, чтобы облегчить изучение программ.

Опытные программисты наверняка смогут предложить немало изменений для повышения производительности кода. Но не забывайте: цель автора заключалась в том, чтобы получить работающую программу с наименьшими усилиями.

Ресурсы в Интернете

На сайте издательства вы найдете множество полезных ресурсов, включая все необходимые программные файлы, а также дополнительные ссылки:

<https://nostarch.com/crackingcodes/>

Загрузка и установка Python

Прежде чем заняться программированием, необходимо установить интерпретатор Python — программу, которая будет выполнять написанные вами инструкции. В дальнейшем при упоминании о Python мы будем подразумевать интерпретатор Python.

Версии Python для Windows, macOS и Linux/UNIX можно бесплатно загрузить по адресу <https://www.python.org/downloads/>. Все приведенные в книге программы должны без проблем выполняться в последней версии Python.

Примечание

Загружайте Python версии 3 (например, 3.6 или 3.8). Приведенные в книге программы рассчитаны именно на эту версию. В случае использования Python 2 они могут выполняться некорректно или вообще не работать.

Windows

В Windows загрузите установщик Python, файл которого должен иметь расширение `.msi`, и выполните двойной щелчок на нем. Установите Python в соответствии с приведенной ниже процедурой, следуя инструкциям, которые будут появляться в окне программы установки.

1. Выберите пункт **Install Now**, чтобы начать установку.
2. По завершении инсталляции щелкните на кнопке **Close**.

macOS

В macOS загрузите файл `.dmg` и дважды щелкните на нем. Установите Python в соответствии с приведенной ниже процедурой, следуя инструкциям, которые будут появляться в окне программы установки.

1. Когда DMG-пакет откроется в новом окне, выполните двойной щелчок на файле `Python.mpkg`. Возможно, вам придется ввести пароль администратора.
2. Щелкайте на кнопках **Continue** в разделе **Welcome**, а затем щелкните на кнопке **Agree**, чтобы принять условия лицензии.

3. Выберите HD Macintosh (или имя своего жесткого диска) и щелкните на кнопке Install.

Загрузка файла `pyperclip.py`

Почти все приведенные в книге программы импортируют авторский модуль `pyperclip.py`. Он содержит функции, позволяющие выполнять операции копирования и вставки шифротекстов с использованием буфера обмена. Модуль не входит в комплект поставки Python, поэтому его необходимо загрузить по следующему адресу:

<https://inventwithpython.com/pyperclip.py>

Файл модуля должен находиться в той же папке, что и файлы других программ. Если это не так, то при попытке запустить программу появится следующее сообщение об ошибке:

```
ImportError: No module named pyperclip
```

Теперь, когда вы знаете, как загрузить и установить интерпретатор Python и модуль `pyperclip.py`, познакомимся со средой, в которой нам предстоит писать программы.

Запуск IDLE

Интерпретатор Python — это специальная программа, предназначенная для запуска пользовательских приложений. Для написания программ требуется *интегрированная среда разработки* (Integrated Development and Learning Environment — IDLE), во многом напоминающая текстовый процессор. Она устанавливается вместе с Python. Для запуска IDLE необходимо выполнить следующие действия.

- Если вы работаете в Windows 7 или выше, щелкните на значке Пуск в левом нижнем углу экрана, введите **IDLE** в поле поиска и выберите в списке элемент IDLE (Python 3.6 64-bit).
- Если вы работаете в macOS, откройте окно программы Finder, щелкните на пункте меню Applications, затем — на элементе Python 3.6 и далее — на значке IDLE.
- Если вы работаете в Ubuntu, выберите пункты меню Applications⇒Accessories⇒Terminal, а затем введите **idle3**.

Какую бы операционную систему вы ни использовали, окно IDLE будет выглядеть примерно одинаково (рис. 1), разве что текст заголовка

окна может несколько отличаться в зависимости от используемой версии Python.

Это окно называется *интерактивной оболочкой*. Оболочка представляет собой программу, которая позволяет непосредственно вводить инструкции подобно тому, как это делается в окне командной строки Windows. Вводимые здесь инструкции будут сразу же выполняться интерпретатором Python.

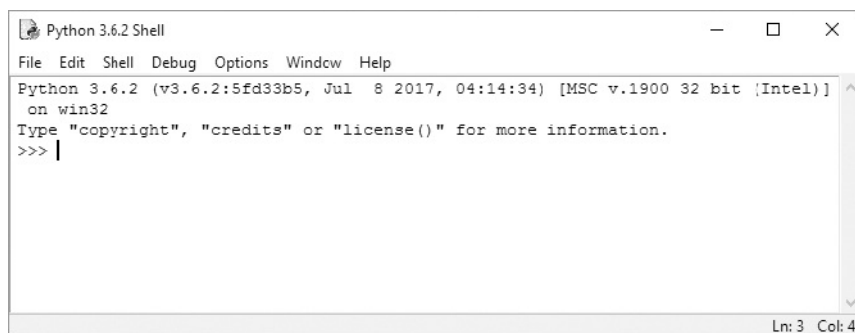


Рис. 1. Окно IDLE

Например, перейдите в окно интерактивной оболочки и введите сразу вслед за приглашением `>>>` следующую команду:

```
>>> print('Hello, world!')
```

Нажмите клавишу `<Enter>`, и интерактивная оболочка немедленно отреагирует на это, выдав следующий текст:

```
Hello, world!
```

Резюме

В былые времена взламывать шифры приходилось с помощью карандаша и бумаги. И хотя с появлением компьютеров многие классические шифры стали уязвимыми для атак, их изучение по-прежнему представляет интерес. Кроме того, написание программ криптоанализа, способных взламывать подобные шифры, — отличный способ научиться программировать.

В главе 1 вы познакомитесь с простейшими криптографическими инструментами, позволяющими шифровать и дешифровать сообщения, не прибегая к помощи компьютеров.

Ждем ваших отзывов!

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересны любые ваши замечания в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам электронное письмо либо просто посетить наш сайт и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Отправляя письмо или сообщение, не забудьте указать название книги и ее авторов, а также свой обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию новых книг.

Наши электронные адреса:

E-mail: info@dialektika.com

WWW: <http://www.dialektika.com>