**Издательство C**

В этой короткой главе обсуждается, как опубликовать ваше приложение Android, создав версию приложения, которой можно поделиться с другими. В частности, объясняется, как [криптографически подписать ваше приложение](https://developer.android.com/studio/publish/app-signing.html) и собрать его так, чтобы его могли установить люди, не использующие Android Studio.

Прежде чем вы фактически распространите свое приложение, вы должны убедиться, что оно полностью готово к публикации. Google предоставляет [отличный список](https://developer.android.com/studio/publish/preparing.html) того, что нужно сделать перед выпуском приложения, а также более подробный [контрольный список](https://developer.android.com/distribute/best-practices/launch/launch-checklist.html) для выпуска приложения в Play Store.

* К наиболее распространенным задачам относятся: удаление лишних команд ведения журнала, а также проверка доступности и локализации.

После выполнения этих шагов вы будете готовы создать и подписать свое приложение.

**C.1 Подписание приложения**

Как описано в [лекции 1](https://info448.github.io/introduction.html#building-apps) , создание приложения Android включает в себя компиляцию кода Java и XML в байт-код DVM, а затем упаковку этого кода (вместе с ресурсами и графикой) в .apkфайл.

Но для того, чтобы установить этот .apkфайл на устройство, его необходимо **криптографически подписать** , включив [**сертификат открытого ключа**](https://en.wikipedia.org/wiki/Public-key_cryptography) . Этот [сертификат](https://en.wikipedia.org/wiki/Digital_signature) соответствует *закрытому ключу* (секретному коду), который есть у вас (разработчика). Поскольку каждый сертификат открытого ключа связан с кодом, который знаете только вы как разработчик (похож на пароль), он может выступать в качестве идентификационной *подписи* для вашего приложения: только вы знаете секретный пароль, поэтому только вы можете предоставить этот конкретный сертификат. Таким образом, *подписывая* своей .apkподписью, вы отмечаете пакет как разработанный вами, а не кем-то другим — как подпись на чеке. Android использует эти подписи в качестве функции безопасности, чтобы гарантировать, что все будущие обновления исходят от одного и того же человека (никаких вредоносных обновлений приложений!), а также для проверки источника установленного пакета.

* Секретные *закрытые ключи* хранятся на вашем компьютере в .keystoreфайлах (представьте себе: базу данных закрытых ключей). На вашем компьютере может быть несколько различных хранилищ ключей.

По умолчанию, когда вы создаете и запускаете приложение в Android Studio, IDE автоматически генерирует **отладочный сертификат** , с помощью которого вы подписываете свое приложение. Этот сертификат не защищен (это автоматически сгенерированный пароль!), поэтому не заслуживает доверия для магазинов приложений (например, Play Store)… но его достаточно для установки и запуска вашего приложения через Android Studio.

* По умолчанию ключи хранятся в ~/.androidпапке на Mac и Linux, и C:\Users\USER\_NAME\.android\папке на Windows. Вы можете просмотреть ключ отладки (например, на Mac) с помощью команды:

keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey -storepass android -keypass android

* + Это -aliasимя конкретного сертификата, а аргументы -storepassи keypassявляются буквальными паролями, связанными с хранилищем (базой данных) и сертификатом (в базе данных) соответственно. Тот факт, что это хранилище защищено паролем, делает его секретным и доступным только разработчику.
  + Прокрутите страницу вниз, чтобы увидеть «Отпечатки пальцев сертификата»; например, SHA1сертификат используется при получении [ключа API Google Карт](https://developers.google.com/maps/documentation/android-api/signup) .
* Важно отметить, что каждый компьютер, на котором запущена Android Studio, создаст свой собственный уникальный *сертификат отладки* . Это означает, что «подпись», идентифицирующая ваше приложение, будет отличаться для каждого компьютера: даже если у него тот же пакет и тот же исходный код, Android будет считать его «другой» программой, поскольку она была собрана (в режиме отладки) на другой машине. Это особенно важно, когда такие вещи, как ключи API (например, для Google Maps), связаны с определенной цифровой подписью; это означает, что *каждая* машина разработки должна иметь свою уникальную подпись, связанную с API!

В дополнение к автоматически сгенерированным *сертификатам отладки* вы можете подписывать приложения с помощью собственного сгенерированного **сертификата выпуска** . Это сертификат, который не создается автоматически Android Studio, но все еще связан с секретным «паролем», который знаете только вы. Эти сертификаты также хранятся в файле .keystore, который создается и защищается паролем разработчиком. Поскольку это хранилище ключей хранится в секрете и заблокировано, только разработчик может подписать сборку .apkс помощью проверяемой подписи, тем самым гарантируя, что любые обновления приложения должны исходить от этого разработчика.

Мне нравится думать об *отладочных сертификатах* как о дешевых, легко воспроизводимых ручках Bic, а *о сертификатах выпуска* как о причудливых золотых перьях. Когда над приложением работают несколько разработчиков, каждый из них будет подписывать свои тестовые версии своими собственными дешевыми ручками Bic. Но когда дело доходит до выпуска проекта, вам нужно достать дорогое золотое перо, чтобы сделать подпись. Действительность приложения зависит от того, какая «ручка» используется для его подписи. (В этой метафоре файл .keystore— это пенал).

**Выпускать.apk**

Чтобы создать совместный **релиз.apk** , вам необходимо создать *сертификат релиза* , чтобы подписать приложение, а затем создать и подписать приложение .apkс помощью этого сертификата.

Android studio упрощает [подписание сборки релиза](https://developer.android.com/studio/publish/app-signing.html#release-mode) (перейдите по ссылке для получения более подробной информации и примеров). Короче говоря, выберите Build > Generate Signed APKиз меню и следуйте инструкциям мастера!

* Вам будет предложено указать местоположение файла .keystoreдля использования (например, где хранить ваши ручки релиза). Я рекомендую создать файл где-нибудь в домашнем каталоге вашего пользователя (например, ~/android-release.jks). Обратите внимание, что вы можете использовать один и тот же закрытый ключ (находится в хранилище ключей) для нескольких приложений.
* Также можно настроить Android Studio на [автоматическую](https://developer.android.com/studio/publish/app-signing.html#sign-auto) подпись вашего приложения при сборке для выпуска. Убедитесь, что вы [удалили информацию о подписи](https://developer.android.com/studio/publish/app-signing.html#secure-shared-keystore) из ваших файлов сборки, чтобы ваши пароли не загружались на GitHub!

Собранный и подписанный .apkфайл будет создан в выбранной вами папке назначения. Затем этот файл можно будет использовать [совместно](https://developer.android.com/studio/publish/index.html#publishing-release) : загрузить в Google Play Store, разместить на веб-странице или отправить по электронной почте напрямую кому-либо для установки.

* Обратите внимание, что установка .apkфайлов не из Play Store (даже подписанных) требует от пользователя [согласия на использование приложений из неизвестных источников](https://developer.android.com/studio/publish/index.html#unknown-sources) .