# Keep Secrets Right

# WHOAMI

## Артем Кулаков
## Android Lead в Redmadrobot

- пишу Android приложения
- интересуюсь безопасностью приложений и серверов
- могу что-нибудь взломать на досуге ;)
- в свободное время помогаю OpenSource проектам

Twitter: @Fi5t
Telega: https://t.me/android_guards

# План доклада

- Хранение секретной информации
- Проблемы шифрования
- Генерация надежных ключей
- Где хранить ключи?
- Итоги

# Соглашения

**Sensitive information == “чувствительная информация” == конфиденциальная информация == секретная информация**

# Виды секретов

- Персональные данные пользователя
- Токены доступа
- Служебные файлы приложения
- Придумайте свой вариант...

# Хранение секретов
# (популярный подход)

## Хранение секретов
## (правильный подход)

- Не хранить вообще
- Хранить на backend/middleware
- Шифровать

# Как не хранить секреты

- Не хардкодить аккаунты, “суперсистемные” токены и прочую секретную информацию
- Не сохранять персональные данные пользователя на диске
- Не использовать “секретные конфиги” вида disableSSL=false

# Как не хранить секреты

**Вывод:** понять какие данные являются секретными для вашей доменной области и отказаться от их хранения.

# Как хранить на сервере

- Весь обмен с сервером только по HTTPS + SSL Pinning
- Все секретные данные лежат на сервере и запрашиваются в realtime и не кэшируются
- Токены имеют адекватно-короткое время жизни и обновляются

# Как хранить на сервере

**Вывод:** приложение становится максимально тонким клиентом.

# Как шифровать

# Шифрование. Племенная мифология.

- Это сложно (spoiler: да)
- Нужно знать матан, дифуры и квантовую физику
- Медленно и дорого
- Никто не знает где хранить ключи

# Главная проблема криптографии

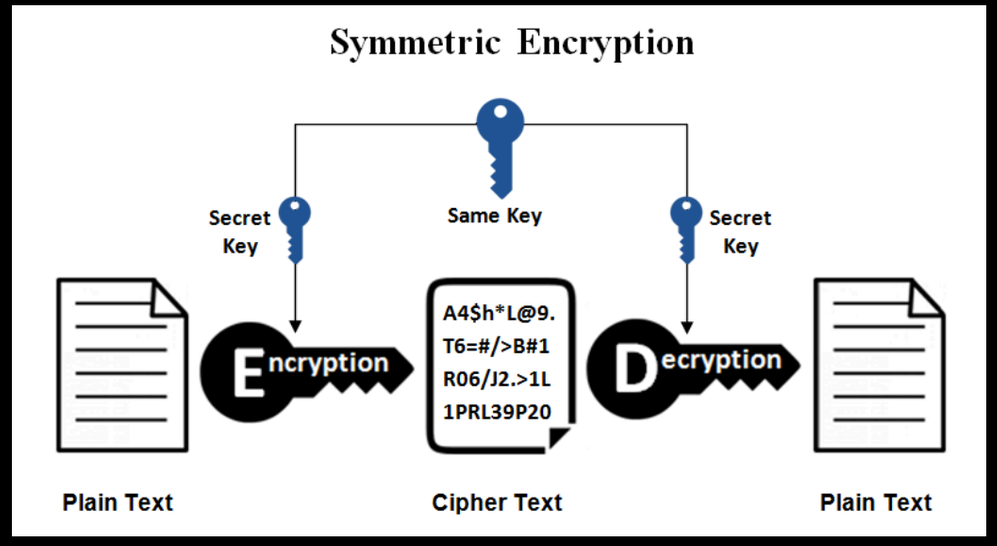
## Управление ключами

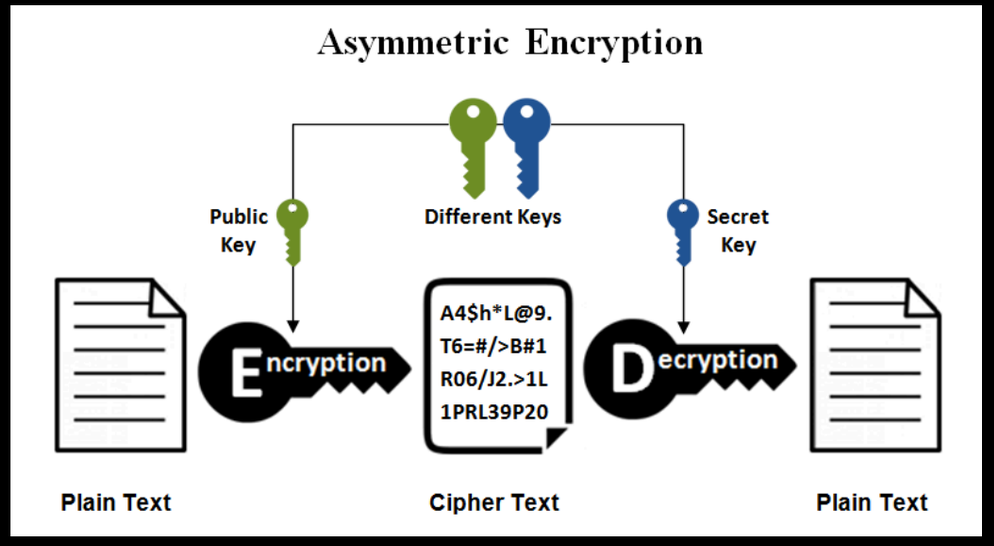
**Стоимость руководителя контрразведки ЦРУ вместе с женой не превысила двух миллионов долларов. Это намного дешевле, чем создавать крупные компьютеры для взлома и нанимать гениальных криптоаналитиков.**

**Б. Шнайер**

# Виды ключей

- симметричные (AES, DES)
- асимметричные (RSA, Diffie-Hellman)



# Создание надежных ключей

1. Взять за основу пользовательский ввод
2. Взять соль и/или перец (salt + pepper)
3. Смешать все вместе и скормить хэш-функции
4. ??????
5. PROFIT!!!

# Создание надежных ключей

**PBKDF2**

- HMAC-SHA1 в качестве PRF
- 64 бита соль
- 10k итераций

**Альтернативы:** bcrypt/scrypt, Argon2

# Где хранить ключи?

**Там, откуда их сложно извлечь без паяльника на 100W ;)**

- В голове пользователя (PBKDF2 and etc.)
- В железе (TEE, SEP)

# Android way

- Trusted Execution Environment
- Secure Element
- Keystore & KeyChain

# KeyChain vs Keystore


- KeyChain: общесистемное хранилище
- Keystore: индивидуальное хранилище для каждого приложения

# KeyChain

- Android 4.0+
- Можно хранить PKCS#12 (private key + X.509 CA)
- Можно хранить в железе (Android 4.3+)
- Нельзя грабить корованы и хранить симметричные ключи =(

# Keystore

- Android 4.3+
- Нельзя извлечь из памяти приложения
- Можно хранить в железе
- Поддерживаются асимметричные ключи и симметричные ключи (Android 6+)
- Поддержка Fingerprint auth API

# Проблемы Keystore

- Удаление ключей при изменении типа блокировки экрана (баг или фича?)
- Key blob можно расшифровать (но это неточно)
- Ключи остаются на устройстве после удаления приложения если в его манифесте было allowBackup=true

# Итоги

- Нельзя ничего хранить на устройстве
- Хочешь хранить? Шифруй.
- Хочешь шифровать? Не храни ключи!
- Хочешь хранить? Страдай…

# Полезные ссылки

- Android Keystore System - https://goo.gl/wMCoHC
- Keystore redesign in Android M - https://goo.gl/Qdd1BV
- The Forgetful Keystore - https://goo.gl/x6Cz6w
- Unifying Key Store Access in ICS - https://goo.gl/2nXydS
- Applied Cryptography - https://goo.gl/mNwKrQ
- Android Security Internals - https://goo.gl/xEL1Pd

# Any questions?

REDMADROBOT