УДК 004.021

ШИФРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ДАННЫХ В БАЗЕ ДАННЫХ REALM ДЛЯ ОПЕРАЦИОННОЙ СИСТЕМЫ IOS

Д.А. МЕДЮХО
(Представлено: Т.С. РУДЬКОВА)

Анализируются теоретические сведения о методах защиты пользовательских данных в базе данных Realm. Рассмотрен практический подход использования шифрования и защищенного хранения ключей шифрования.

Каждую секунду пользователи со всего мира вводят личную информацию в установленные приложения, которые неявно обязуются хранить эту информацию в безопасности. Большое количество компаний и разработчиков по-прежнему относятся к безопасности, как аспекту, которым занимаются после реализации основных функций приложения, а не как к основному важнейшему компоненту их программного продукта.

Метод шифрования. В Realm имеется возможность шифрования файлов. Используя 64-байтовый ключ шифрования методом AES-256+SHA-2, Realm становится настолько безопасным, что многие банки начали использовать его в своих клиентских приложениях [1].

AES (Advanced Encryption Standard) – симметричный алгоритм блочного шифрования, принятый правительством США в качестве стандарта в результате конкурса, проведенного между технологическими институтами [2]. Он заменил устаревший DES (Data Encryption Standard), который больше не соответствовал требованиям сетевой безопасности, усложнившимся в XXI веке. Считается, что используемый в AES ключ длиной в 256 бит – достаточно надежная защита против лобовой атаки, то есть с чисто математической точки зрения подобрать один правильный пароль из всех возможных – трудно осуществимая задача. Несмотря на некоторые недостатки AES, взломать защищенную с помощью этого алгоритма информацию практически нереально. Схема алгоритма показана на рисунке 1.

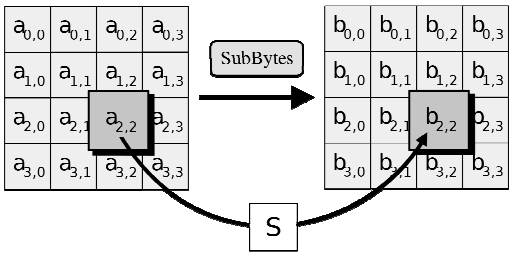

Рисунок 1. – Алгоритм AES Винсента Рэймена и Йоана Даймена

Хэш-функции семейства SHA-2 пользуются высокой популярностью в приложениях, связанных с систематизацией, поиском и защитой информации [3]. Криптографические хэш-функции – это математические операции, выполняемые с цифровыми данными; сравнивая хеш (результат выполнения алгоритма) с известным значением хеша, человек может определить целостность данных. Ключевым аспектом криптографических хеш-функций является их устойчивость к взлому: никто не должен иметь возможность найти два разных входных значения, которые приводят к одному и тому же результату хэш-функции. SHA-2 справляется с этим на отлично и был выбран как стандарт хеширования данных. Схема алгоритма представлена на рисунке 2.

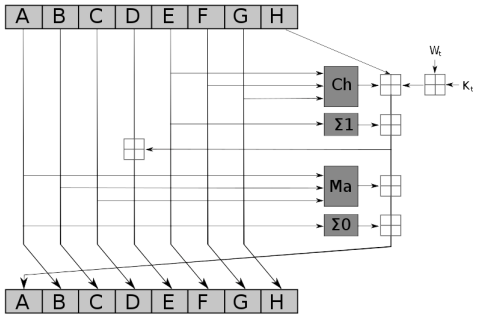

Рисунок 2. – Алгоритм SHA-2

Реализация. Включение шифрования в Realm настолько просто, что требуется только одна дополнительная настройка в настройках конфигурации Realm. Для того чтобы включить шифрование для файлов Realm, необходимо определить свойство *encryptionKey* при создании Configuration базы данных, а затем использовать созданную конфигурацию при создании объекта Realm [2]. Чтобы создать файл с шифрования, необходимо придерживаться следующих шагов:

1) генерация 64 байта случайных данных для ключа шифрования;
2) создание объекта конфигурации с ключом шифрования;
3) ткрытие файла Realm с созданной конфигурацией;
4) бработка исключений при открытии файла с неверным ключом;
5) спользование объекта для дальнейших запросов.

После создания файла Realm с ключом шифрования, этот ключ будет необходим при каждом новом открытии. При попытке открыть файл с неверным ключом, Realm просто вернет ошибку, указав, что база данных недействительна.

Ограничения. Ключ шифрования для определенного файла Realm всегда только один и его изменение невозможно. Чтобы изменить ключ шифрования, можно экспортировать копию файла Realm, используя *writeCopyToPath*(*_:encryptionKey:*) и установить новый ключ на копию [2]. Также важно заметить, что шифрование в Realm увеличивает вашу базу данных на небольшой размер из-за необходимых операций шифрования и дешифрования.

Принцип защиты ключа шифрования. Хотя сами данные безопасно защищены Realm, самая важная проблема безопасности – защита ключа шифрования. Ответственность за хранение ключа шифрования несет разработчик. К этому вопросу должны относиться с тем же вниманием, что и к обработке пароля учетной записи пользователя. Самым безопасным вариантом хранения ключа шифрования является его сохранение Keychain. Keychain – надежно защищенный стандарт хранения паролей на устройствах пользователя [5].

Когда зашифрованный файл Realm скопирован на новое устройство, ключ шифрования также должен быть доступен на новом устройстве. Существует много способов, например, сохранить ключ шифрования в Keychain и синхронизировать Keychain через iCloud или через ваш собственный сервис при входе в систему [6].

Заключение

Процесс шифрования считается трудоемким и сложным, но с помощью Realm защита пользовательских данных становится простым шагом. Благодаря тому, что шифрование настолько легко интегрируется, это должно побудить разработчиков правильно защитить данные своих пользователей и свести к минимуму утечки данных в будущем.

ЛИТЕРАТУРА

1. Realm // Wikipedia [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Realm. – Дата доступа: 17.09.2017.
2. Realm [Электронный ресурс]. – Режим доступа: https://realm.io/docs/objc/latest/. – Дата доступа: 17.09.2017.
3. Advanced Encryption Standard [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Advanced_Encryption_Standard. – Дата доступа: 17.09.2017.
4. SHA-2 [Электронный ресурс]. – Режим доступа https://ru.wikipedia.org/wiki/SHA-2. – Дата обращения 17.09.2017.
5. Adding Capabilities. iCloudKeychain Sharing [Электронный ресурс]. – Режим доступа: https://developer.apple.com/library/content/documentation/IDEs/Conceptual/AppDistributionGuide/Adding Capabilities/Adding Capabilities.html. – Дата доступа: 18.09.2017.
6. KeychainAccess [Электронный ресурс]. – Режим доступа: https://support.apple.com/en-us/HT204085. – Дата доступа: 18.09.2017.