**Буквально 8 января был выведен в стадию preview новый сервис Azure Key Vault.**

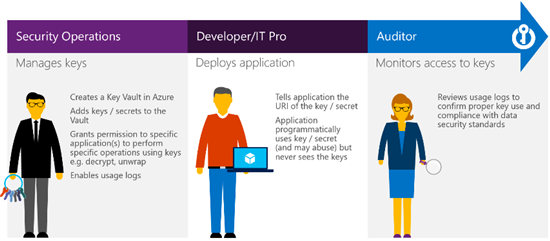
****

Key Vault это - HMS as a Service (Hardware security module <http://en.wikipedia.org/wiki/Hardware_security_module> ). HMS - это **выделенное hardware**, которое позволяет хранить, управлять ключами/секретами и **шифровать/расшифровывать, ставить/проверять подписи** максимально безопасным образом и достаточно быстро (специфичное железо, заточенное под шифрование, по заявлениям работает быстро, но на сколько или какие делались замеры- информации нет).

Azure Key Vault – это менеджмент ключей и секретов через веб-интерфейс, а все остальное предоставляет HMS. Ранее KV был известен как BYOK (bring-your-own-key).

**Теоретическая часть**

Для понимания концепции нужно глянуть на несколько схем и это видео



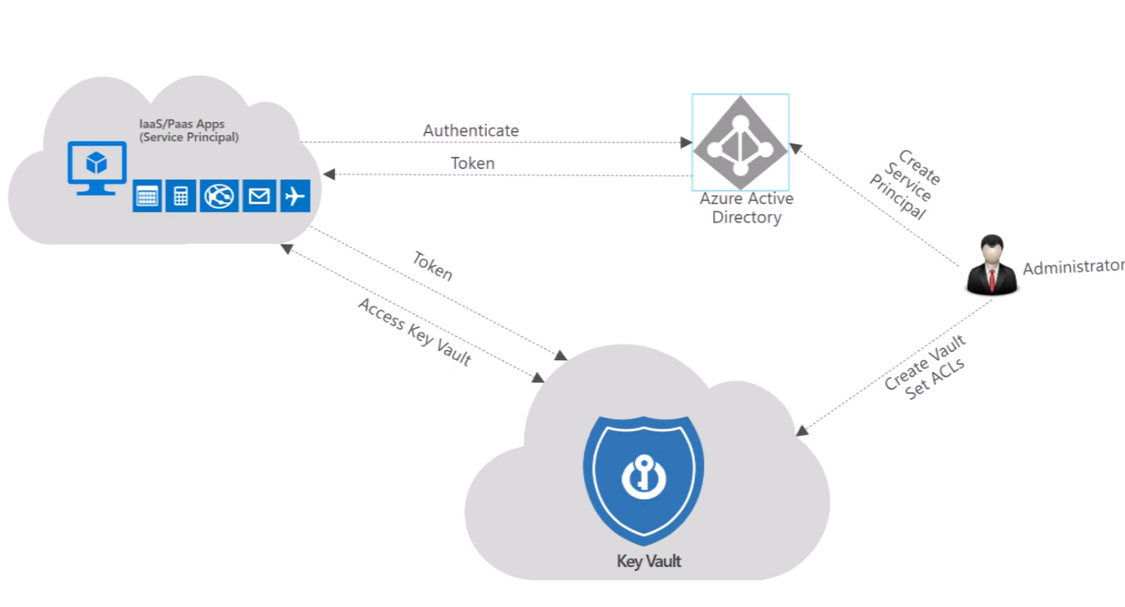
В концепции есть 3 выделенных роли.

1. Те, кто занимаются ключами.
2. Те, кто пишет софт, который использует ключи.
3. Те, кто следит за результатами работы приложений с ключами.

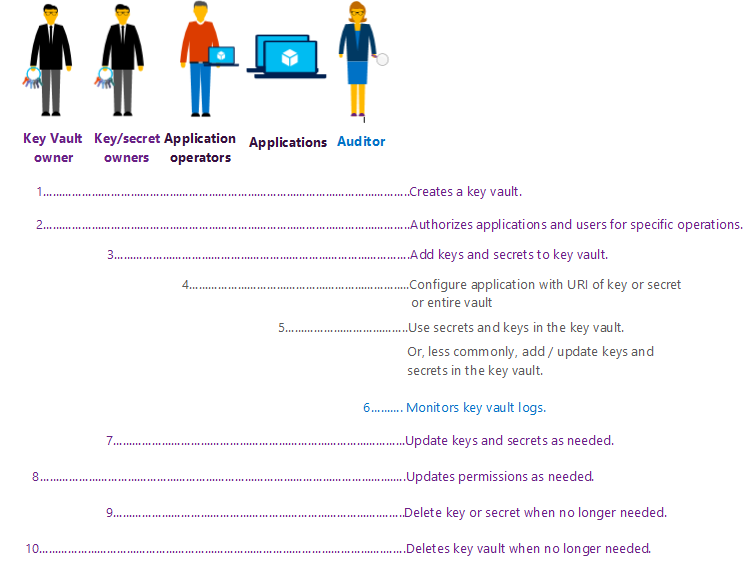
Из моего опыта: первые и третьи в 90% случаях сидят в одном и том-же отделе информационной безопасности, причем третьи смотрят за логами не через призму сопровождения и эксплуатации конкретных приложений, а вообще про использование криптографии. Microsoft выделила 3 роли, т.к. аудитор должен быть независимым от всей остальной компании, чтобы делать независимые оценки, поэтому будем использовать эту концепцию и далее.

**Для понимания механизма работы нам понадобятся 2 схемы**

Схему нужно читать справа налево. Администратор создает в Azure Active Directory приложение, загружает в Key Vault ключи/секреты. Для упрощения считаем, что реальное приложение уже написано. Приложение, аутентифицировавшись в AAD, выполняет работу с шифрованием через Key Vault, используя полученный при авторизации токен.

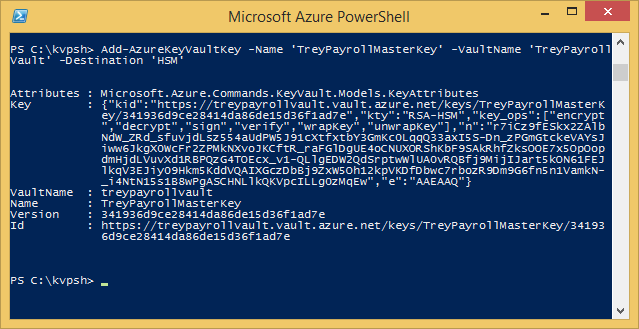


Ниже на схеме показано что может делать каждая роль с key vault.



Как реально все это выглядит через Management Portal можно посмотреть в этой <http://azure.microsoft.com/en-us/documentation/articles/key-vault-get-started/> и этой статье <http://blogs.technet.com/b/kv/archive/2015/01/09/azure-key-vault-step-by-step.aspx> , либо в видео.

Мне не хочется к каждому пункту прикладывать powershell команду, их можно прочесть из статьи оригинала.



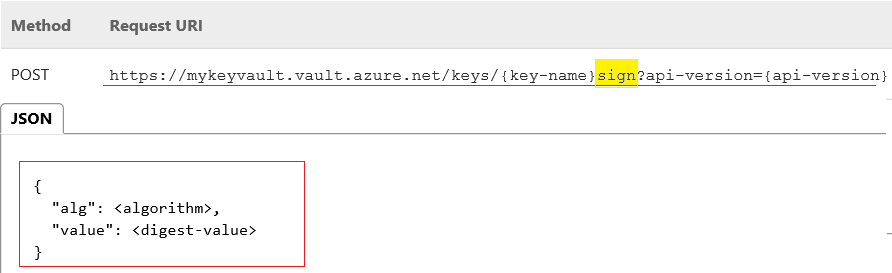
**Использование для шифрования**

При создании Key Vault мы его как-то назвали. На пример mykeyvault.

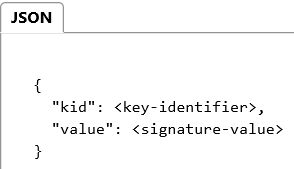
Мы создали ключ, который будем использовать в примере key-name. У ключа может быть несколько версий (старая версия).

В URL мы должны указать тип операции, которую мы делаем. В примере sign- подпись.

Затем в теле сообщения в виде json передаем 2 параметра: алгоритм и текст, который мы хотим обработать.



В ответ мы получим идентификатор ключа и подписанный текст.



**API VERSION**

API Version - это параметр, его текущее значение - «2014-12-08-preview».

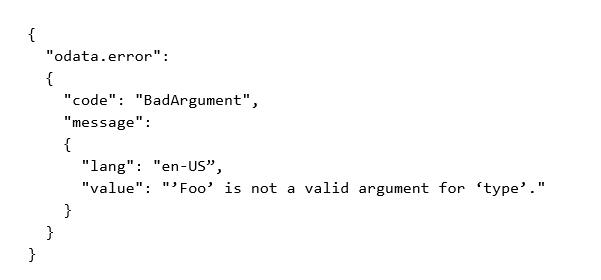
Для тех, кто часто работает с API, необходимость этого параметра и проблемы при его отсутствии очевидны.

Как нетрудно догадаться, в мире нет ничего совершенного, и версия будет меняться, а сам API - становиться несовместимым. Разработчик, единожды написав код, планирует, что он будет работать вечно и далее редко следит за обновлениями сервиса, если по коду нет задач на доработку. Явное указание версии позволит контролировать эти изменения и не споткнуться об автообновление, когда мы ничего не меняли, а нас обвинят в поломке приложения.

**Ошибки**

О произошедших ошибках можно будет понять 3 способами.

* HTTP код ответа. Тут как всегда- 2\*\* - хорошо, 3\*\* - стоит обратить внимание, 4\*\* - проблемы с авторизацией, 5\*\* - azure плохо.
* В возвращаемом json может быть описание проблемы



* аудит

**Аудит**

Смысл аудита в том, чтобы видеть откуда какое приложение и какие ключи использует, когда оно это делало, а также ошибки.

Команда KV заявляет, что скоро будет доступна возможность мониторинга и аудита использования ключей, быстрый анализ на основе Hadoop... НО в данный момент этого нет.

**Железо**

Какие конкретно железки стоят в azure не раскрывается, однако, подчеркивается, что они соответствуют американскому стандарту FIPS\_140-2 (есть 3 версии стандарта от 2009 года) level 2 (всего 4 уровня, 4 самый жесткий.). <http://en.wikipedia.org/wiki/FIPS_140>

**API**

Про API, как всегда, особо разговаривать нечего. Microsoft всегда выпускает API ко всем своим сервисам, при этом сам Azure портал использует этот API для доступа.

У Вас есть 2 варианта работылибо самим написать клиент используя соответствующую документацию , либо использовать готовый .net клиент <http://msdn.microsoft.com/library/azure/dn903628> и powershell командлеты <http://msdn.microsoft.com/library/dn868052.aspx> (сам .net клиент понятное дело в powershell не нуждается). Есть кое-какие примеры работы <http://www.microsoft.com/download/details.aspx?id=45343>

**В чем я вижу огромный плюс этого сервис для разработчика**

Работа с криптографией вынесена из вашего приложения в отдельный, доступный через web-api сервис, не на вашем сервере. Key Vault потенциально снимет кучу головной боли с разработчика (если вы можете использовать его в своем приложении по техническим характеристикам, и это не противоречит политическим мотивам).

Те, кто занимался криптографией на .net в Российских реалиях, не дадут соврать, что достаточно часто шифрование - это какие-нибудь подключаемые нативные библиотеки, которые работают часто только под 32-битной операционной системой, не старше чем какой-нибудь windows xp sp3. И тут либо ты сам выносишь шифрование в виде веб-сервиса и получаешь полный комплект радости от новой сущности (ее же надо мониторить, обучить работать с нею сопровождение, получить еще одну потенциальную точку отказа), либо вынужден согласиться использовать в 2014 году xp sp3 на 32-битной OS.

Еще хуже, когда шифрование предоставлено в виде программы .exe с командным интерфейсом, и приходится еще думать, как с ней взаимодействовать правильно, чтобы при этом выдерживать SLA.

**В чем отделу информационной безопасности(ИБ) выгода?**

На мой взгляд, отделу ИБ будет на порядок проще управлять ключам шифрования (создавать, отзывать), если они не будут раскиданы по десяткам серверов, а будут собраны в едином Key Vault. Как минимум, ключи нужно периодически менять (0.5-1 года) и сделать это на куче серверов - это обезьянья работа.

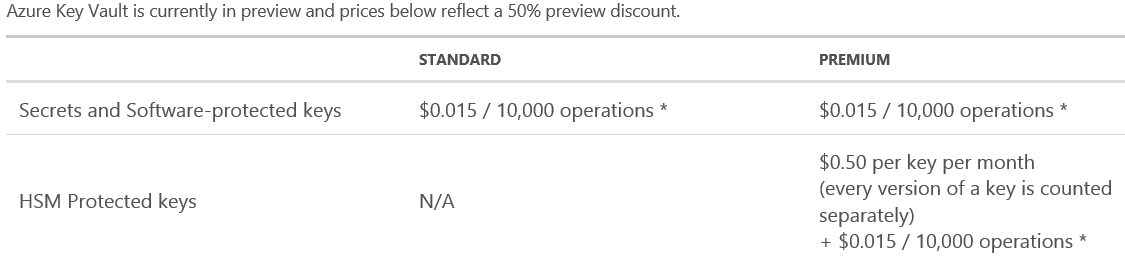
На уровне key vault можно ограничивать использование приложениями/пользователями разных ключей и типы операций, которые они могут делать.

Пример: пришел нам из Центрального банка архив, мы знаем, что на нем должна быть подпись, а на каждом файле в архиве - шифрование и авторизация. Соответственно, мы даем приложению, которое работает с этими файлами, права на расшифровку и проверку подписи, но не даем права на шифрование этим ключом и установку подписи и, тем самым, снижаем вероятность неверного использования.

**Уверен, что отдел ИБ найдет десяток причин не использовать Key Vault если захочет, но это их работа - никому не доверять.**

**Цены:**

Более детально можно узнать из статьи <http://azure.microsoft.com/en-us/pricing/details/key-vault/> , особенно прочитав FAQ в конце статьи, а я расскажу основную мысль.

Есть 2 тарифных плана: **standard**, **premium** и есть всего 2 фичи, за которые берутся деньги. 1- это операции с использованием ключей, а 2- это хранение ключей в HSM. В итоге 2 плана отличаются тем, что в плане standard нет хранения ключей в HSM. 

«Что такое операция?» - это первый возникающий вопрос! Процитируем:

“Every successfully authenticated REST API call counts as one operation. Examples of operations for keys: create, import, get, list, backup, restore, delete, update, sign, verify, wrap, unwrap, encrypt and decrypt. Examples of operations for secrets: create/update, get, list.” <http://msdn.microsoft.com/en-us/library/azure/dn903630.aspx>

Говоря русским языком, это операции с ключами и секретами (создание ключей, обновление, шифрование, проверка подписей и т.д.).

И еще один хороший, вопрос: «А если ключи выпустил не я, а другой подписчик Azure, то кто платит?» Ответ: тот, кто выпустил, а не тот, кто пользуется.

**Ссылки:**

* Стартовая http://azure.microsoft.com/en-us/services/key-vault/
* Стартовая для документации <http://azure.microsoft.com/en-us/documentation/services/key-vault/>
* Блог команды авторов <http://blogs.technet.com/b/kv/>
* Видео о том, что это такое. <http://channel9.msdn.com/Blogs/Windows-Azure/Azure-Key-Vault-Developer-Quick-Start>
* Forum <https://social.msdn.microsoft.com/forums/azure/en-US/home?forum=AzureKeyVault>
* Использование Key Vault в SQL сервер <http://blogs.technet.com/b/kv/archive/2015/01/12/using-the-key-vault-for-sql-server-encryption.aspx> <http://msdn.microsoft.com/library/dn198405.aspx>
* Стартовая страница документации на MSDN <http://msdn.microsoft.com/en-us/library/azure/dn903625.aspx>