Криптография: Система PGP

Реферат к индивидуальному докладу

Генералов Даниил, 1032212280

Содержание

Список иллюстраций

Список таблиц

1 Введение

PGP – это приложение, протокол и спецификация системы шифрования. Она была изобретена в 1991 году Филиппом Зиммерманом [1] для шифрования электронной почты. С течением времени она стала использоваться для других задач. В этом докладе мы рассмотрим устройство этой криптосистемы, сравним ее с X.509 (самой популярной криптосистемой, на основании которой работает HTTPS и TLS), и рассмотрим практические применения этой системы в различных задачах.

2 Архитектура

2.1 Криптография

PGP – это гибридная криптосистема: она сочетает в себе инструменты симметричной и асимметричной криптографии.

Первые версии PGP использовали самописные криптографические примитивы (включая симметричный алгоритм BassOmatic), но с выпуском версии 5 они все были заменены на стандартные алгоритмы, вроде RSA, DSA и ElGamal.

Когда пользователь хочет отправить зашифрованное сообщение одному или нескольким получателям, он должен сначала получить их публичные ключи (например с keyserver, см. ниже). После этого создается эвфемерный ключ симметричного шифрования, которым шифруется основное содержимое сообщение. После этого к сообщению прикладываются несколько копий этого эвфемерного ключа – по одной на каждого получателя, зашифрованные публичным ключом этого получателя. (Это сделано так, потому что асимметричное шифрование гораздо медленнее, чем симметричное шифрование, поэтому асимметричное шифрование используется для минимального количества данных.)

Для выполнения электронной подписи, как всегда, используется хеш содержимого сообщения, который затем используется с приватным ключом, чтобы сгенерировать подпись. Любой получатель затем может использовать публичный ключ, связанный с этим приватным ключом, чтобы проверить

авторство этого сообщения. Если нужно отправить сообщение, которое одновременно зашифрованно и подписано, то сначала применяется подпись, а затем шифрование.

2.2 Сертификаты

Электронная подпись проверяет, что содержимое сообщения не было изменено, и что это сообщение действительно создано владельцем связанного приватного ключа. Для того, чтобы определить, кто является владельцем приватного ключа, требуется сертификат – специальное сообщение, которое делает связь между публичным ключом и идентификационной информацией (вроде имени, email и т.д.).

В контексте PGP терминология несколько смешана: это сообщение по сути является сертификатом, но оно называется “публичный ключ” (и имеет в ASCII-представлении заголовок `BEGIN PGP PUBLIC KEY BLOCK`). Сертификат содержит публичный ключ, но не равен ему: у сертификата также есть дополнительная информация.

2.3 Подписи ключей

Сертификат по умолчанию имеет подпись от того ключа, которому он соответствует – он самоподписанный. Для того, чтобы установить доверие, другие пользователи должны использовать свои ключи, чтобы создать подпись для этого сертификата. Присутствие такой подписи выражает доверие информации в сертификате: если Боб подписал сертификат Алисы, это значит, что он верит, что Алиса – настоящий обладатель ключа, который указан в сертификате.

Этот шаг требуется, потому что кто угодно может создать свой собственный сертификат, в котором написано, что это – Алиса. Без подписей пользователи

криптосистемы не могут определить, какой из этих сертификатов настоящий.

2.4 Keyserver

Для того, чтобы пользователи получили информацию о том, что Алиса имеет определенный публичный ключ (и что этому факту верит Боб), нужно иметь какое-то хранилище этой информации. В PGP для этого используются специальные базы данных, которые называются `keyserver`: они позволяют пользователям получить публичный ключ (и сертификат) пользователя по его отпечатку или email-адресу, а также хранят информацию о том, кто подписал этот сертификат, и обо всех возможных событиях, связанных с этим сертификатом (вроде продления или отзыва).

Поскольку все сообщения, которые можно опубликовать на keyserver, имеют цифровые подписи от своих отправителей, доверие к keyserver не требуется. Keyserver не имеет возможности подменить сообщения, которые отправлены на него – единственное, что он может сделать, это отказаться публиковать это сообщение, и в таком случае можно просто использовать другой keyserver. Разные keyserver-а обмениваются информацией между собой, поэтому ключ, опубликованный на один, будет доступен на других спустя некоторое время.

В последнее время законы о персональных данных, вроде GDPR, запрещают публикацию персональной информации без согласия ее владельца. Из-за этого традиционные сети keyserver почти не существуют, а вместо них появились *валидирующие* keyserver. По умолчанию они позволяют отправлять на них только анонимную информацию о криптографических ключах [2], и если вы хотите опубликовать свой сертификат (с именем и email-адресом), то вам потребуется подтвердить владение этим email-адресом.

3 Сравнение с X.509

X.509 – это еще один популярный стандарт криптографической системы с сертификатами. Он используется как основа TLS и HTTPS-шифрования, а также для подписи и шифрования email (с помощью S/MIME) и электронных документов.

В отличии от PGP, X.509 имеет древовидную структуру доверия. Сверху дерева находится сертификат Удостоверяющего Центра (УЦ, англ. Certificate Authority (CA)), доверие к которому приходит аксиоматически путем наличия этого УЦ в списке доверенных. Для многих браузеров и операционных систем такой список предоставляется вендором: так, браузер Firefox использует список УЦ из Mozilla Root CA Program, и пользователи Firefox автоматически доверяют всем УЦ, которым доверяет Mozilla.

Поскольку сертификаты в PGP могут иметь несколько подписей, и эти подписи могут быть двухсторонними, PGP имеет более сложную структуру доверия, называемую *web of trust*. В этой системе доверие между пользователями выражается с помощью цепочки подписей, и при достаточно активном использовании криптосистемы можно ожидать, что сертификат нового контакта будет иметь какую-то связь с существующей сетью контактов (по аналогии с так называемым правилом шести рукопожатий).

Благодаря сетевой архитектуре, PGP более устойчивый к потере ключей: если кто-то потерял контроль над своим ключом, то его подписи перестают иметь значение, но если его контакты имеют другие подписи, то они не потеряют доверие. (Но если потерянный ключ был значимой частью сети доверия, то его

потеря может привести к необычным отдаленным эффектам в общей сети.)

В X.509 сертификаты имеют строгую иерархию значимости: листовые сертификаты не способны создавать подписи на другие сертификаты, поэтому их потеря ничего не значит для общей сети доверия, в то время как утечка или уничтожение ключа УЦ (или даже слухи о том, что он не заслуживает доверия) может привести к банкротству компании.

4 Применения

Изначально PGP был создан для шифрования электронной почты. С течением времени (и особенно после публикации стандарта OpenPGP, RFC 4880) PGP начал использоваться для других задач.

4.1 Подпись репозиториев пакетов

Linux-дистрибутивы используют репозитории пакетов, чтобы распространять ПО. Для того, чтобы их было удобнее кешировать, многие из них доступны по нешифрованному HTTP. Однако это дает возможность злоумышленникам подменить файлы на сервере (или выполнить MITM-атаку), чтобы клиент установил себе вредоносное ПО.

Чтобы предотвратить это, все пакеты в таких репозиториях подписаны ключами разработчиков дистрибутива [3]. Таким образом, можно скачать пакет и сохранить его в кеше, и любой пользователь кеша может подтвердить, что этот пакет не был манипулирован.

4.2 Локальное шифрование

Традиционно, PGP используется для шифрования сообщений, которые идут с одного компьютера на другой. Но его также можно использовать для шифрования сообщений на одном и том же компьютере: например, существуют приложения, которые используют PGP для шифрования диска.

Аналогичное поведение доступно для отдельных файлов: например, KDE Wallet хранит пароли и другие секреты в KDE-системе в файле, который может быть зашифрован с помощью PGP-ключа [4].

Из-за того, что PGP позволяет отправить сообщение, имея доступ только к публичному ключу, это иногда используется для безопасного хранения файлов, которые не должны часто использоваться. Например, можно отправлять все важные пароли для серверов итд. в один почтовый ящик, шифруя их ключом, который хранится в сейфе за стеклом. Тогда, в случае экстренной ситуации, можно разбить стекло и открыть сейф, чтобы получить доступ ко всем этим паролям.

4.3 Подпись git-коммитов

Git – система контроля версий, в которой ревизии файлов в репозитории называются коммиты.

По умолчанию коммиты не имеют криптографической подписи: кто угодно может сделать коммит с любым текстом и автором (и даже существуют утилиты, которые позволяют удобно подделывать коммиты таким образом). Поэтому есть шанс, что злоумышленник, получивший доступ к серверу, сможет добавить вредоносный код в какой-то коммит, и это не будет замечено.

Чтобы защититься от этого, можно использовать PGP, чтобы подписать коммит. Это удостоверяет, что владелец этого ключа согласен с содержимым коммита, включая его код и текст. Более того, поскольку коммиты в Git выстраиваются в направленный граф и имеют связь по хешу, это также подтверждает, что все коммиты до текущего не были изменены.

4.4 Подпись исполняемых файлов

В Microsoft Windows и macOS есть механизм, который позволяет подписывать исполняемые файлы, который работает на основании X.509. Разработчик получает сертификат от вендора, который затем используется для создания цифровой подписи. При запуске программы сначала проверяется подпись, и если она не сходится, или сделана недоверенным сертификатом, то пользователю показывается предупреждение о том, что эта программа недоверенна (или, в зависимости от настроек системы, можно просто запретить этот запуск).

В Linux есть похожий механизм, который вместо этого работает на основе PGP, который работает на основании модуля ядра `digsig-ng` [5]. Перед запуском исполняемого файла в ELF-формате, одна из секций этого файла читается, чтобы извлечь его цифровую подпись. Если секции кода и текста не соответствуют этой подписи, или подпись была сделана неизвестным ключом, то ядро блокирует запуск этой программы.

Разработка модуля `digsig-ng` была заброшена в 2014 году, однако Astra Linux Special Edition предоставляет этот модуль для реализации Замкнутой Програмной Среды [6] (и скорее всего предоставляет поддержку этого модуля для новых версий ядра Linux). Из-за этого сертифицированное ПО, которое нужно запускать на Astra Linux, также требуется подписывать ключом разработчика – но, в отличии от схемы в MS Windows/macOS, ключ разработчика может импортировать конечный пользователь системы.

5 Выводы

В этом докладе мы рассмотрели устройство системы PGP, которая используется для многих задач, которые требуют децентрализованной системы доверия (web of trust); PGP является первым и самым распространенным примером такой системы. Мы также рассмотрели практические употребления этой системы: в последнее время она меньше используется для шифрования данных, и больше – для создания и проверки криптографических подписей.

Список литературы

1. Zimmermann P. Why I Wrote PGP [Электронный ресурс]. 1991. URL: https://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html.
2. OpenPGP. Launching a new keyserver! [Электронный ресурс]. 2019. URL: https://keys.openpgp.org/about/news#2019-06-12-launch.
3. Debian. Securing Debian: Package signing in Debian [Электронный ресурс]. 2022. URL: https://www.debian.org/doc/manuals/securing-debian-manual/deb-pack-sign.en.html.
4. Linux A. KDE Wallet [Электронный ресурс]. 2024. URL: https://wiki.archlinux.org/title/KDE_Wallet.
5. digsig-ng. linux-digsig [Электронный ресурс]. 2014. URL: https://github.com/digsig-ng/linux-digsig.
6. Левдонский А. Astra Linux: Режим замкнутой программной среды [Электронный ресурс]. 2024. URL: https://wiki.astralinux.ru/pages/viewpage.action?pageId=41190634.