21 Инфраструктура открытых ключей

21.1 Сертификаты открытых ключей

Пусть A (Алиса), B (Боб), C (Клара),... — абоненты информационной системы. Пусть sk_A, sk_B, sk_C,... — личные ключи абонентов, а pk_A, pk_B, pk_C,... — соответствующие открытые. Для организации связи абоненты должны обменяться открытыми ключами. Причем, как мы видели на примере атаки «противник посередине» на протокол Диффи — Хеллмана, при обмене обязательно должен быть обеспечен контроль целостности и подлинности ключей — ключи должны распространяться по аутентифицированным каналам связи (АКС).

Первоначально вопросам распространения открытых ключей не уделялось большого внимания. Считалось, что открытые ключи будут размещаться в *каком-то* общедоступном справочнике и *каким-то* образом будет обеспечиваться контроль целостности и подлинности данных, загружаемых в справочник и выгружаемых из него. Со временем стало понятно, что все не так просто. Как метко заметил В. Столлингс, «*управление открытыми ключами является ахиллесовой пятой криптографии с открытым ключом*». Возникли вопросы: кто будет хранить справочник? как защищать справочник? как организовать АКС для связи с ним?

Не следует понимать АКС буквально. Это не физический канал связи, а скорее интерфейс, концепция. Для организации АКС Алиса и Боб прибегают к сервисам, которые принято называть услугами доверия (trust services). Сервисы предоставляют специальные службы (центры, серверы) — поставщики услуг доверия.

Совокупность поставщиков услуг доверия называется *инфраструктурой открытых ключей* (ИОК[1]). ИОК могут быть корпоративными, ведомственными, государственными, глобальными. Самая распространенная архитектура ИОК основана на *сертификатах* открытых ключей и регламентируется стандартом X.509, разработанном ITU-T (Международный консультационный комитет по телефонии и телеграфии). Фактически, X.509 — это способ организации распределенного справочника и создания виртуальных АКС.

В X.509 главные поставщики услуг доверия — это удостоверяющие центры (УЦ). В нашей системе имен УЦ — это Трент. Трент вызывает алгоритмы `Setup`/`Gen` и вырабатывает долговременные параметры par_T, личный ключ sk_T и открытый ключ pk_T. Трент использует ключи для выработки и проверки ЭЦП.

Сертификат открытого ключа Алисы — это данные о ней и ее открытом ключе, заверенные ЭЦП Трента. Более точно, сертификат Алисы состоит из следующих полей:

$Cert_T(A) =$

версия сертификата
номер сертификата
Id_T (идентификационные данные Трента)
срок действия сертификата
Id_A (идентификационные данные Алисы)
par_A (долговременные параметры Алисы)
pk_A (открытые ключи Алисы)
расширения сертификата (дополнительные атрибуты)
`Sign`(par_T, sk_T, объединение предыдущих полей)

Боб, располагая сертификатом Алисы $Cert_T(A)$ и открытым ключом Трента pk_T, может проверить подлинность и целостность открытого ключа Алисы pk_A. Открытый ключ Трента может распространяться также в виде сертификата $Cert_{T_1}(T)$, заверенного еще одним удостоверяющим центром T_1. Для проверки открытого ключа T_1 может использоваться еще один сертификат $Cert_{T_2}(T_1)$ и так далее. В конце концов образуются цепочки сертификатов

$$Cert_{T_1}(T), Cert_{T_2}(T_1), \ldots, Cert_{T_{n-1}}(T_n), Cert_{T_n}(T_n).$$

Последнее звено цепочки — это самоподписанный сертификат корневого удостоверящего центра. Такие сертификаты выпускаются известными компаниями, оказывающими услуги в области криптографии с открытым ключом (примеры — VeriSign, DigiCert).

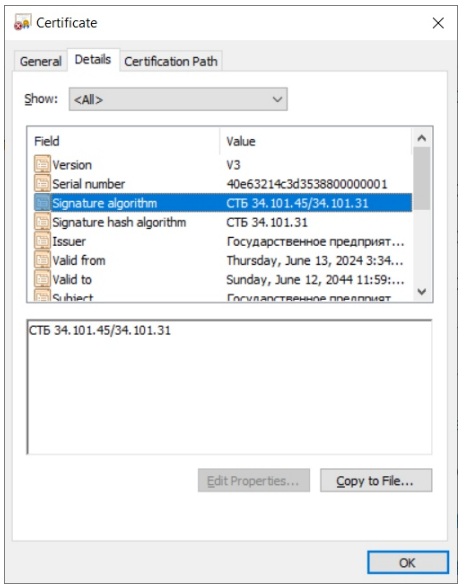
[1] PKI, Public Key Infrastructure

Сертификаты корневых удостоверящих центров включатся в состав поставки операционных систем, «прошиваются» в браузерах и т.д. Например, список сертификатов браузера Firefox можно найти вот здесь: `https://wiki.mozilla.org/CA/Included_Certificates`.

21.2 Инфраструктура РБ

В нашей стране государственная ИОК называется ГосСУОК — Государственная система управления открытыми ключами. Она функционирует с 2014 года. Ее оператором является Национальный центр электронных услуг (НЦЭУ). На сайте НЦЭУ (`https://nces.by`) до недавнего времени обновлялась информация о числе изданных сертификатов: ≈ 2500000 на 10 марта 2024 года.

На следующем рисунке представлен действующий сертификат КУЦ. Это самоподписанный сертификат — он выпущен КУЦ для КУЦ, причем открытый ключ сертификата совпадает с открытым ключом подписи. Сертификат выпущен на 20 лет, ближе к окончанию срока действия он будет заменен на новый. Новый сертификат будет уже не самоподписанным, а самовыпущенным: новый открытый ключ будет подписан на старом личном ключе.



В сертификате указан идентификатор алгоритмов выработки подписи. Это алгоритмы `bign-with-hbelt`, определенные в СТБ 34.101.45. В алгоритмах используются долговременные параметры `bign-curve256v1`. Параметры определяет группу точек эллиптической кривой астрономически большого порядка $q = 115792089237316195423570985008687907853218625362177255134934233029367509640711$. Кривая выбрана так, что для компрометации подписи (например, определения личного ключа по открытому) требуется выполнить порядка $\sqrt{q} \approx 2^{128}$ операций. Такой объем вычислений по одним оценкам потребует не менее 30 лет работы крупных государственных криптоаналитических агенств, а по другим — вообще недостижим (не хватит всей энергии Вселенной).

Личный ключ КУЦ не покидает границ специального аппаратного устройства (HSM), которое выпускает сертификаты. Сразу после генерации ключ был разделен на 5 частей с помощью алгоритмов СТБ 34.101.60. Полученные в результате частичные секреты хранятся в разных местах. В случае сбоя HSM личный ключ может быть восстановлен по любым 3 частям из 5. При этом любые 2 частичных секрета не позволяют получить никакой информации о ключе.

Как видим, в ГосСУОК активно используются государственные стандарты серии СТБ 34.101. Номера релевантных стандартов и их краткое содержание представлены в следующей таблице.

Номер	Содержание
31	блочный шифр Belt, режимы шифрования, хэширование, имитозащита, аутентифицированное шифрование, шифрование с сохранением формата, дисковое шифрование
45	ЭЦП, транспорт ключа
47	генерация псевдослучайных чисел, одноразовые пароли
60	разделение секрета
65	TLS 1.2 с собственными криптонаборами
66	протоколы типа Диффи — Хеллмана
77	хэширование на основе sponge-функции
17, 19, 26	ИОК типа X.509
23, 50, 80	форматы криптографических данных
81, 82	службы
78	профиль ИОК
79	криптографические токены (id-карты)
87	аутентификация

Подробную информацию о стандартах можно найти вот здесь: `http://apmi.bsu.by/resources/std`.