Blocko OpenKeyChain: *Key-based Authentication in Open Environment*

2015.12.01

Overview

현대의 인터넷 환경에서는 TLS의 보편적인 보급으로 인해 우리는 모두 안전한 통신을 즐기고 있다. 검열을 피해 암호화된 메시지를 주고받고, 도난 당할 걱정 없이 신용카드 번호를 결제에 사용할 수 있으며, 중간에 위,변조될 걱정을 할 필요 없이 안전하게 파일을 다운로드 받아 사용하고 있다.

그러나 TLS 같은 신뢰성 있는 기술도, 우리에게 익숙한 클라이언트 서버 모델을 벗어난 더 복잡한 구조의 상호작용을 모두 만족시키지는 못한다.

클라이언트 서버 모델에서는 값비싸고 번거로울지라도 PKI를 통해 인증서를 발급하고 교환하는 방식이 가능했지만, 상호작용이 필요한 기기와 서비스 수가 기하급수적으로 증가하는 오늘날에는 보다 더 효율적인 수단이 필요하게 되었다.

즉, 소셜 네트워크 서비스뿐만 아니라 냉장고, 디지털 도어락도 중간자 공격으로부터 보호 받을 수 있어야 하고, 사용자의 개입 없이도 상호 인증 및 안전한 통신이 가능해야 한다. 또한 사물 인터넷의 도래가 본격화하면 이러한 요구가 더욱 심화될 것이다.

더욱이 PKI의 신뢰성 자체가 의심 받기 시작하면서, 구글의 Certificate Transparency[1]와 같은 기술이 이를 보완하기 위해 사용되고 있는 실정이다.

이러한 현실에서, 안전한 분산 장부 기술인 블록체인이 해결책으로 대두되고 있으며,[2] 블로코는 OpenKeyChain 기술을 통해 블록체인을 활용하여 당면한 문제를 해결할 수 있는 신뢰성 있는 공개된 표준을 제시하고자 한다.

OpenKeyChain을 활용하면 누구나 애플리케이션에서 필요로 하는 PKI 및 인증서를 생성, 등록, 사용할 수 있다. 또한 TLS 없이도 메시지에 서명하여 위,변조를 방지하고, 서비스와 사용자 간에 암호화된 메시지를 주고 받을 수 있다.

OpenKeyChain은 오픈소스로 공개되어 누구나 구현하여 사용할 수 있으며, 블로코의 CoinStack SDK에는 OpenKeyChain의 reference implementation이 포함되어 있어 모든 서버, 모바일, HTML5 기반 웹브라우저 등에서 이를 활용할 수 있다.

[1] https://www.certificate-transparency.org/
[2] https://en.wikipedia.org/wiki/Public_key_infrastructure#Blockchain-based_PKI

Domains

OpenKeyChain을 사용하기 위해서는 먼저 도메인을 생성해야 한다.

도메인은 하나의 PKI에 해당하며, 각 도메인에 등록된 인증서는 다른 도메인과 호환되지 않는다.
보통 하나의 조직, 또는 애플리케이션을 위해 하나의 도메인을 생성하고 관리한다.

Representation

도메인 자체는 사실상 추상적 개념이며, 실제 각 도메인은 인증서로 표현되고, 관리된다.

Domain Organization

각 도메인 하위에는 다른 도메인 또는 인증서들이 존재할 수 있다.

도메인 하위에 존재하는 도메인은 서브 도메인(subdomain)이라고 하며, 다른 도메인 상위에 존재하는 도메인은 부모 도메인이라고 한다.
다른 도메인 하위에 존재하지 않는 도메인은 루트 도메인(root domain)이라고 한다.

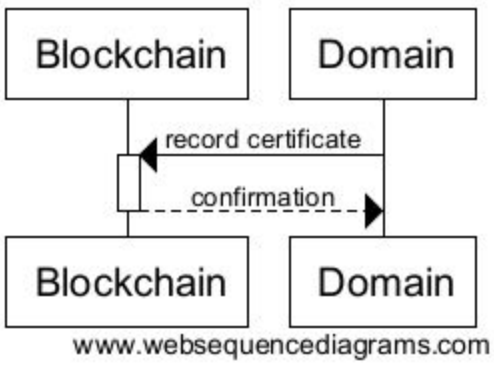
Lifecycle

Registration

도메인을 생성하기 위해서는 먼저 인증서를 생성해야 한다.

OpenKeyChain 인증서는 비공개키와 공개키로 구성되며, 반드시 secp256k1[3] [4] 방식으로 생성해야 한다.
생성된 비공개키와 공개키는 해당 도메인과 연동되며, 연동 관계를 선언하기 위해서는 블록체인에 registration record를 저장해야 한다.

Registration record는 colored coin[5] 방식으로 transaction으로서 블록체인에 저장되며, OpenKeyChain marker output을 포함한 OpenKeyChain record 형식을 준수해야 한다.



Revocation

생성된 도메인을 폐기하기 위해서는 이미 생성된 인증서를 사용하여 revocation record를 블록체인에 저장한다. 서브 도메인의 경우 부모 도메인에서 이 작업을 대신 수행할 수 있다.

[3] http://www.secg.org/sec2-v2.pdf
[4] https://en.bitcoin.it/wiki/Secp256k1
[5] https://en.bitcoin.it/wiki/Colored_Coins

Certificate

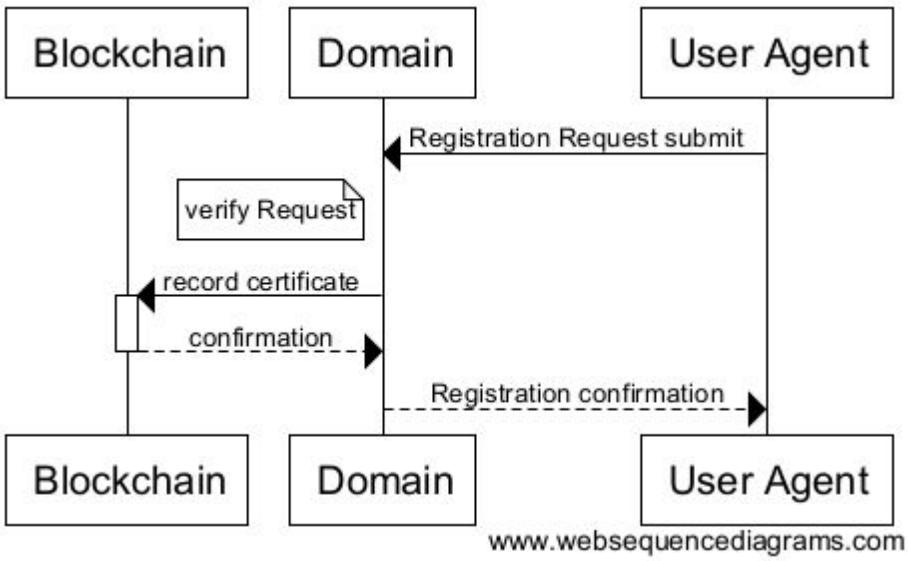
도메인 또는 서브 도메인을 생성한 후에는 단말기 또는 사용자가 사용할 수 있는 인증서를 생성하고 등록할 수 있다.

Lifecycle

Registration

단말기 또는 사용자가 사용할 수 있는 인증서는 도메인을 위한 인증서와 마찬가지로 비공개키와 공개키로 구성되며, 반드시 secp256k 방식으로 생성해야 한다.
생성된 인증서는 특정 도메인 또는 서브 도메인과 연관되어야 하며, 이를 위해 블록체인에 registration record를 저장해야 한다.

Registration record는 도메인을 위한 인증서와 마찬가지로 colored coin방식으로 transaction으로 블록체인에 저장되며, 부모 도메인 또는 서브 도메인에 의해 생성된다.



Revocation

생성된 인증서를 폐기하기 위해서는 이미 생성된 인증서를 사용하여 revocation record를 블록체인에 저장한다. 인증서의 폐기는 인증서를 생성한 도메인 또는 서브도메인에 의해서도 가능하다.

Record Format

OpenKeyChain에서 사용하는 registration record는 colored coin[6] 방식으로 transaction으로서 블록체인에 저장되며, 다음 조건을 만족해야 한다.

- OpenKeyChain marker output을 포함한다.
- Marker output은 아래 형식을 따른다.

Field	Description	Size
Marker	해당 output이 OpenKeyChain marker output이라는 사실을 확인하는 매직 바이트. 항상 0x4B43 이어야 한다.	2 바이트
Version	Major, Minor 버전 정보. 현재 0x0100 이다 (1.0)	2 바이트
OP code	해당 output에서 선언하는 operation 정보. ● Domain 등록: OP_DOMAIN_REG (0x0001), ● Domain 폐기: OP_DOMAIN_REV (0x0002), ● Subdomain 등록: OP_SUBDOMAIN_REG (0x0003), ● Subdomain 폐기: OP_SUBDOMAIN_REG (0x0004), ● 인증서 등록: OP_CERT_REG (0x0101), ● 인증서 폐기: OP_CERT_REV (0x0102)	2 바이트
Payload	해당 OP code 관련 추가 정보 또는 메타데이터	가변

- 서브 도메인 또는 인증서를 다른 도메인이나 서브 도메인 하위에 생성하는 경우, 생성하려는 새로운 서브 도메인이나 인증서를 기존 도메인 또는 서브 도메인 인증서를 사용해서 blessing 한다.
    - 이 경우 생성하려는 새로운 서브 도메인이나 인증서의 public key를 address 방식으로 표현하여 marker output 바로 이전의 colored output으로 포함시켜야 한다. 만약 서브 도메인 등록/폐기, 또는 인증서 등록/폐기를 수행하는 registration record에 이러한 colored output이 포함되지 않으면, 이는 유효하지 않은 registration record로 취급된다.
    - Marker output 바로 다음에 오는 output들은 모두 무시되며, uncolored output으로 취급된다.

[6] https://en.bitcoin.it/wiki/Colored_Coins

Channels

인증서 등록이 완료되면 해당 인증서의 identity를 활용하여 서버 클라이언트 또는 P2P 환경에서 메시지를 서명하는 데 사용할 수 있다.

OpenKeyChain에서의 기본적인 메시지 서명은 ECDSA 방식으로 이루어지며, 원본 문서와 서명이 존재하면 해당 identity가 서명한 문서임을 확인할 수 있다.

해당 서명을 검증하는 시점에는 블록체인의 registration record를 참고하여 특정 domain과의 연관관계, 부여 받은 권한, 또는 폐기 여부를 확인할 수 있다.

전자 서명을 지속적으로 생성할 필요가 있거나, IoT 기기처럼 성능이 낮은 환경에서는 ECDSA 방식으로 전자 서명을 생성하거나 검증하는 데 필요한 컴퓨팅 자원이 부담스러울 수 있다. 또한, 경우에 따라 전자 서명을 통한 메시지 검증 뿐만 아니라, 메시지 암호화가 필요할 수도 있다.

그러한 경우에는 ECDSA를 활용하여 shared secret을 공유하고, 이러한 shared secret을 통해 실시간 데이터 스트림에 대한 전자 서명을 생성하고 검증하거나, 암호화를 수행할 수 있는 data channel을 구성할 수 있다.

Initialization

Channel을 열기 위해서는 먼저 shared secret을 공유하기 위한 키 교환이 이루어져야 한다. 키 교환을 위해서는 몇 가지 알고리즘을 활용할 수 있다.

Diffie-Hellman Key Exchange[7]

디피-헬만 키 교환을 이용하면 서로의 공개키 교환만으로도 새로운 shared secret 생성이 가능하다. Forward secrecy를 위해서는, OpenKeyChain을 사용해 등록한 인증서의 공개키를 곧바로 사용하기 보다는 해당 channel에서 사용할 임시 인증서를 새로 생성하고, 등록된 인증서로 ECDSA 서명하여 인증한다. 이후 임시 인증서의 공개키를 교환하고, shared secret을 생성하여 channel을 구성한다. shared secret 생성에 사용한 임시 인증서는 즉시 파기한다.

RSA[8]

RSA와 같은 비대칭 암호 시스템을 channel용 shared secret 교환을 위해 사용할 수 있다. 이 경우 해당 channel에서 사용할 새로운 RSA 비공개키를 channel을 여는 쪽에서 생성하고, 공개키를 OpenKeyChain을 사용해 등록한 인증서로 서명하여 상대에게 전달한다. Channel을 여는 상대편은 shared secret을 생성한 후 전달 받은 공개키를 사용하여 암호화하고, OpenKeyChain으로 등록한 인증서로 서명하여 상대에게 전달한다. 이렇게 교환된 shared secret을 사용하여 channel을 열 수 있다. 이후 shared secret 공유에 사용한 RSA 인증서는 즉시 파기한다.

[7] https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange
[8] https://en.wikipedia.org/wiki/RSA_(cryptosystem)

Usage

일단 Shared secret이 공유된 이후에는, 필요한 컴퓨팅 자원이 ECDSA보다 훨씬 적은 효율적인 암호화 알고리즘을 활용하여 메시지 교환이 가능하다.

Signature

Shared secret을 활용하여 HMAC 방식으로 메시지에 대한 전자 서명을 생성하고 검증할 수 있다. CoinStack reference implementation에서는 HMAC-SHA-256을 지원한다.

Encryption

Shared secret을 활용하여 대칭 키 암호화 방식으로 메시지를 암호화하여 전달할 수 있다. CoinStack reference implemenation에서는 AES-256을 지원한다.

Conclusion

OpenKeyChain을 활용하면 막대한 비용이 소요되는 사설 PKI를 구축하지 않아도 블록체인을 활용하여 애플리케이션 또는 조직에서 필요한 PKI를 최소한의 투자로 구축할 수 있다.

또한 개발 환경 또는 구축 환경에 따라 잘못 운용되기 쉬운 TLS에 전적으로 의존하기 보다는, 메시지 계층의 보안을 강구하여 보다 심층적인 보안 체계를 구축할 수 있다는 장점이 있다. 더불어 애플리케이션에서 필요로 하는 기능을 제공하는 사설 인증서와 PKI를 구축함으로서 기존 TLS의 한계를 극복할 수 있는 이점도 있다.