Mikks 지갑 구조

문서변경이력

버전 번호	작성일	변경사항 요약	작성자	확인자
0.01	9/18/2018	Initial Release	류준선	

목차

개요

일반적으로 블록체인의 지갑 구조는 크게 두 파트로 이루어진다.
- BIP-0039: 랜덤 엔트로피를 이용하여 니모닉을 생성한 후 바이너리 씨드 로 변경하는 부분.
- BIP-0032: 이 바이너리 씨드를 이용하여 deterministic wallets을 생성하는 부분.

믹스체인은 이 두 파트 중 첫 번째 파트인 BIP-0039의 Mnemonic들을 자가 인지적인 문장, 단어, 그리고 숫자의 조합으로 대신하는 구조를 갖고 있다. 자가 인지적이라 함은 사용자가 문장이나 단어 혹은 숫자를 보고 시각화 혹은 추상화가 가능함을 말한다.

또한 믹스체인은 기존 타 블록체인에서 사용하는 secp256k1에 추가하여 sepc256r1을 선택하여 비밀키와 공개키를 생성한다.

동기

BIP-0039의 Mnemonic 은 기억하기 쉬운 단어들로 이루어져 있어 deterministic wallets을 생성하는 데 용이하다고 알려져 있다. 하지만 이 단어들은 랜덤 엔트로피로부터 파생되며 각 단어들의 연관성 또한 현저히 떨어진다. 즉, 적게는 12개 단어, 최대 24개 단어를 머리속에 외우기는 쉽지 않다.

단어의 조합이 아닌 사용자가 지정한 문장을 이용하는 방법이 그 대안이 될 수 있다. 하지만 지금까지 이러한 "사용자 지정 문장"은 높은 엔트로피 시드를 파생할 수 없다고 알려져 있다. 그 이유는 "사용자 지정 문장"이 "희귀" 또는 "무작위"로 보일 수 있으나 실제로는 그렇지 않다는데 있다. brainwallets 이 그 예이며 공격의 대상이다.

그러므로 "사용자 지정 문장"을 이용하기 위해서는 무엇보다 먼저 높은 엔트로피 시드를 생성할 수 있는 방법이 필요하다.

[MIP-0039] : BIP-0039 (Mnemonic) 대체 알고리즘

최종 Seed를 생성하는 함수는 BIP39에서 채택한 HMAC-SHA512 PBKDF2() 함수를 사용한다. "사용자 정의 문장"과 "사용자 정의 암호"의 Endian 형식은 Big Endian을 사용한다. "사용자 정의 문장"과 "사용자 정의 암호"에 랜덤 넘버 2 Bytes 값을 XOR 하여 보다 높은 엔트로피를 생성한다.

- "사용자 정의 문장" XORing
"사용자 정의 문장"의 MSB 2 Bytes를 랜덤 넘버 2 Bytes와 XOR 한 후 그 Output 값을 다음 MSB

2 Bytes와 다시 XOR 한다. 이렇게 마지막 Byte까지 XOR 한다. 만약 "사용자 정의 문장"의 길이가 홀수라면, LSB 1 Byte와 전 MSB 2 Bytes에서 XOR 되어 나온 Output 값의 MSB 1 Byte를 XOR 한다. MSB 2 Bytes에서 XOR 되어 나온 Output 값의 LSB 1 Byte는 그대로 최종 값이 된다.

- "사용자 정의 암호" XORing
"사용자 정의 문장"과 랜덤 넘버 2 Bytes를 XOR 하여 나온 최종 Output 값 2 Bytes를 "사용자 정의 암호"의 MSB 2 Bytes와 XOR 한다. 그 후 "사용자 정의 문장" XORing 과 같은 방식을 취한다.

PBKDF2 함수는 다음과 같다.
DIGEST = PBKDF2(PRF, Password, Salt, c, DLen), where
PRF: 난수, HMAC-SHA512 사용
Password: "사용자 정의 문장"
Salt: "사용자 정의 암호"
c: 원하는 iteration 반복 수, 2048 고정 값
DLen: 원하는 다이제스트 길이, 64 Bytes 고정 값

PBKDF2() 함수의 인자로 들어가는 문자열은 UTF-8 NFKD 형식을 사용한다.

PBKDF2() 함수의 PRF는 HMAC-SHA512를 사용한다.

PBKDF2() 함수의 Password 인자는 니모닉 대신 "사용자 정의 문장"과 랜덤 함수로부터 얻은 2 Bytes 값을 XOR 하여 사용한다. 사용자 정의 문장 규약은 다음과 같다. (Mandatory)
문자열 기준 최소 32자 이상이어야 한다.
문자열 기준 최대 256자 이하이어야 한다.
자가 인지적 문자 혹은 문자들이 중복적으로 문장에 사용될 수 있다.

PBKDF2() 함수의 Salt 인자로 "mnemonic + 사용자 정의 암호" 대신 "사용자 정의 암호"과 랜덤 함수로부터 얻은 2 Bytes 값을 XOR 하여 사용한다. (Mandatory)
문자열 기준 최소 10자 이상 이어야 한다.
문자열 기준 최대 256자 이하이어야 한다.
사용자 정의 문장과 같이 문장을 암호로 사용할 수 있다.

PBKDF2() 함수의 c 는 2048 iteration을 고정 값으로 사용한다.

PBKDF2() 함수의 DLen은 64 Bytes 고정 값을 사용한다.

PBKDF2() 함수의 최종 Output 인 DIGEST는 HD Wallet에서 사용되는 최종 Seed 값이다.