

① 개인키 / 공개키

개인키 : d (정수)

공개키 : $Q = d \cdot G$ (타원곡선 점)

$G = \text{Secp256k1}$ 기본 생성점

Q 는 타원곡선 위의 (x, y) 좌표

② 메시지 해시 e

메시지를 해시하면 :

$e = \text{keccak256}(\text{message})$

③ 서명 생성 과정 (Signer가 하는 일)

여기서 "난수 k " 가 핵심이다.

1) k 선택

$k \leftarrow \text{random } (1 \leq k \leq n-1)$

2) $R = k \cdot G$ 계산

R 은 타원곡선 점 :

$R = (x(R), y(R))$

3) $r = x(R) \bmod n$

여기서 $x(R)$ 은 R 점의 x 좌표

$\rightarrow r$ 은 "서명값의 일부"

4) s 계산

$S = (e + r \cdot d)^* k^{-1} \bmod n$

여기서

• d = 개인키

• k^{-1} = 모듈러 역원

최종 서명

Signature = (r, s, v)

: v 는 y 좌표의 짝수 / 홀수에 따라 공개키 복구할 때 필요함

결론

• $R = k \cdot G$

• $r = x(R)$

• $s = (e + rd)/k$

이 3개가 하나의 연속되는 공식이다.

서명은 이 3개만으로 끝.

④ 서명 검증 과정 (Verifier가 하는 일)

이제 A가 서명했는지 검증하려면 Signer가 썼던 k, d 를 모르지만 다음 수를 사용해 표명할 수 있음



✓ 검증에서 쓰는 핵심 식

(1) u_1, u_2 계산.

$$u_1 = e * s^{-1} \bmod n$$

$$u_2 = r * s^{-1} \bmod n$$

$$\text{이 두 공식은 } S = (e+r \cdot d)/k$$

의 역식을 정리하면 자연스럽게 얻어진다.

(2) P 계산

$$P = u_1 * G + u_2 * Q$$

$Q = d \cdot G$ 이므로, P를 치환해보면 :

$$P = e \cdot kG^{-1} + r \cdot d \cdot kG^{-1}$$

$$= (e + r \cdot d) k^{-1} G$$

$$= R$$

즉, $P = R$ 이 된다.

(3) 검증 조건

$$r == x(P) \bmod n$$

이게 만족하면

→ 서명은 유효

→ 즉 A의 개인키로 서명한 것이 맞다.

U1, U2는 어디서 오나?

서명 공식을 정리하면 :

$$S \equiv (e + rd) / k$$

양변을 S^{-1} 로 곱하면 :

$$k \equiv (e + rd) * S^{-1}$$

이를 검증식에 접어놓으면

자연스럽게 $u_1 = e/S$, $u_2 = r/S$ 가 등장한다.

즉 u_1, u_2 는 검증식에서 R을 재구성하기 위해 필요한 값들이다.

✓ 정리 : 각 기호의 정확한 의미

기호	의미
d	개인 키
Q	공개 키 = dG
k	nonce (랜덤 값)
R	$kG \rightarrow$ 타원 궤선 점
$x(R)$	점 R의 x좌표
r	$x(R) \bmod n$
e	message hash
S	$(e + rd) / k \bmod n$
v	점 R의 Y좌표 부호(짝/홀) 정보
u_1	e/S
u_2	r/S
P	$u_1 \cdot G + u_2 \cdot Q$ (검증용 점)

① Private key : d
Public key : $Q = dG$

공식의 완전 유도

message hash : e
랜덤 nonce : k

② $R = k \cdot G$ 유도 $\rightarrow R = (x(R), y(R))$

k 는 임의의 정수

\hookrightarrow 타원 곡선 점이므로 (x, y) 가 존재.

G 는 타원 곡선의 생성점.

③ 왜 $r = x(R) \bmod n$?

ECDSA 서명에서 "서명값"의 첫 구성요소 r 은 R 의 x 좌표에서 얻는다.

④ ECDSA의 원래 서명 공식.

$$S \cdot k \equiv e + r \cdot d \pmod{n}$$

좌변 : $S \cdot k$

우변 : 메시지 정보 e + 개인키 관여값 $r \cdot d$

검증 과정을 통해 R 을 재구성할 수 있도록 설계됨.

양변 k^{-1} 을 곱하면

$$S \equiv (e + r \cdot d) * k^{-1} \pmod{n} \rightarrow \text{ECDSA의 } S \text{이다.}$$

$$R = kG$$

$$r = x(R) \bmod n$$

$$S = (e + rd) / k \bmod n$$

$$\text{Signature} = (r, s, v)$$

이 3개 식이 전부이며, 여기서 k 와 d 는 오직 서명자만 알 수 있다.

공개키 복구 과정 (V가 왜 필요한가?)

: 서명과 메시지 해시가 주어졌을 때 “이 서명이 어떤 공개키 으로부터 만들어졌는가?”
를 구할 수 있는 절차가 존재.

이 기능이 바로

$ecrecover(\text{hash}, r, s, v)$

둘은 같은 원리.

$\text{ECDSA}.\text{recover}(\text{hash}, \text{signature Bytes})$

65 bytes

① 검증자(혹은 recover 함수)는 무엇을 알고 있는가?

- e
- S
- r
- v (y 좌표의 Parity : 짝수인지 홀수인지)
- 곡선 파라미터 G, n

② R점을 재구성해야 한다.

$R = kG$ 이고 $r = x(R)$ 즉, r은 x좌표만 알려준다.

하지만 점 R은 x좌표가 같고 y좌표는 두로 2개 알 수 있다.

$$R = (x, y)$$

$R' = (x, p-y)$ 즉 x값만으로는 R을 완전히 확장지울 수 없음.

그래서 v가 필요한 것이다.

③ V는 무엇인가?

v는 다음 중 하나 :

- R의 y좌표가 짝수이면 $v=27$ 또는 0
- R의 y좌표가 홀수이면 $v=28$ 또는 1

즉 v는 y의 짝/홀 정보 (Parity bit)를 제공한다.

이 한 비트 덕분에 “두후보 중 어떤 y가 진짜인지” 결정해서 R을 완전히 재구성 가능.

④ R을 복구하면 Q도 복구할 수 있다.

검증식은 다음과

$$U_1 = e/s$$

$$U_2 = r/s$$

$$P = U_1 \cdot G + U_2 \cdot Q$$

서명자가 만들었을 때 :

$$P = R$$

즉 $R = U_1 \cdot G + U_2 \cdot Q$ 를 만족하는 Q는 하나 밖에 없음.

그래서 Q를 역연산할 수 있는 것이다.

이걸 자동으로 해주는 것이 recover 함수이다.

QH U_1, U_2 를 "역으로 사용"하면 서명을 위조할 수 있는가?

핵심 아이디어

$$P = U_1 \cdot G + U_2 \cdot Q,$$

$$r = x(P)$$

이 조건을 만족하면 서명은 유효한 것으로 판단됨.

이제 이를 거꾸로 이용할 수 있음.

① U_1, U_2 를 내가 먼저 정하면 $\rightarrow P$ 가 자동으로 정해짐.

② P 의 x좌표가 r 이 됨.

③ $s = r/u_2$

④ $e = r * (U_1/U_2)$

즉, 검증식이 만족되도록

(r, s, e) 의 세 값을 "내가 조작해서" 만들 수 있다.

공격자는 d (개인키)를 모르지만 검증식은 k 나 d 를 뭍지 않기 때문에

U_1, U_2 만 조작해도 "일관된 구조를 지닌 valid signature" 가 만들어짐.

왜 가능한가?

검증은 다음 조건만 확인한다.

$$u_1 = e/s$$

$$u_2 = r/s$$

$$P = u_1 \cdot G + u_2 Q$$

$$r = x(P)$$

이 4줄이 전부이다.

이 조건만 맞으면

개인키 "d"를 사용하지 않아도

검증기는 서명을 유효하다고 본다.

결론 공격이 가능한 이유

- 검증기는 k 나 d 를 알 필요없이 $P=R$ 만 확인한다.
- 따라서 u_1, u_2 기반으로 R 을 구성하면,
- 검증식은 개인키를 전혀 사용하지 않고도 통과할 수 있다.

즉 ECDSA 서명 수식에는 '검증자가 모르는 자유도 2개 (u_1, u_2)' 가 존재하기 때문에

이 자유도를 공격자가 이용하면 "일관된 r, s, e "를 조작할 수 있다.

0) 공격을 막으려면 반드시 필요한 4가지 검증

=> 이 중 단 1개만 넣었어도 공격은 절대 안통했다.

- ① 공격자는 임의 (r, s, e) 를 만들어내는데, 컨트랙트는 그 e 가 "정말 서명자가 의도한 메시지인지"
전혀 검증하지 않았다.