정수론

비트코인 속으로 들어간 타원곡선 [3]: 다중서명의 세계

2023년 1월 20일

이철희

지금까지 "비트코인 주소"와 "디지털 서명 알고리듬"에서 타원곡선이 비트코인 시스템에서 중요하게 활용되고 있음을 살펴보았다. 비트코인 주소를 만들기 위해서는 비트코인 타원곡선의 한 점을 선택해야 한다는 사실을 알게 되었다. 비트코인의 소유자가 송금을 하려면 타원곡선을 이용해 디지털서명을 만들어내야 한다는 것도 살펴보았다. 이 두 절차는 모두 큰 숫자인 비밀키를 갖는 데서 시작된다.

비밀키를 아무에게도 알려주지 말라는 것은 연재를 시작하며 처음으로 한 말이기도 하다. 잃어버린 비밀키를 찾는 일이 얼마나 어려운 것인지에 대해서도 조금이나마 살펴보았다. 하지만 그 하나의 숫자일 뿐일 비밀키를 보관하는 문제에 대해서는 별다른 이야기를 하지 않았다. 이 글은 바로 그 점에 대한 이야기라고도 할 수 있다.

선택한 숫자를 안전하게 보관하는 일이란 그렇게 어려운 일도 아니고 진지한 이야기의 소재처럼 들리지 않을 수도 있다. 그냥 비밀키에 해당하는 숫자를 종이에 적어 나만 아는 곳에 숨겨두면 된다고 생각할 수도 있을 것이다. 그러나 그것은 그 종이를 잃어버렸을 때의 결과에 대해 진지하게 고민해보지 않았기 때문일지도 모른다. 앞으로의 이야기에서 종이를 하드디스크로, 하드디스크를 종이로 바꿔 읽더라도 크게 달라지는 것은 없다.

한 영국인은 9년 동안 자신이 버린 하드디스크를 찾기 위해 곳곳에 있는 쓰레기장을 돌아다니고 있다.[1] 기사는 그가 2013년에 8,000개의 비트코인이 들어있는 디스크를 내다 버렸다고 전하고 있다. 이런 기사를 볼 때 우리는 이제 사라진 하드디스크에 들어있던 것은 비트코인이 아니라 그에 대한 거래 권한을 주는 비밀키라고 바로잡아 이해할 수 있다. 세상에 비트코인이 들어있는 하드디스크란 없으며 사라지는 비트코인도 없다. 어느 주소가 얼마만큼의 비트코인을 소유하고 있는지 여부는 수만 대의 컴퓨터가 보관하고 있는 장부에 적혀있다. 비트코인이란 본질적으로 이 장부에 있는 숫자를 읽을 때

사용하는 단위일 뿐이다. 장부에 적힌 비트코인의 개수는 누군가의 하드디스크가 사라지는 일과 관계가 없다. 세상에서 사라진 것은 오직 비밀키일 뿐이다. 8,000개의 비트코인이라면 현재의 기준으로 천억 원 단위의 금액으로 거래될 수 있다. 숫자 하나를 적어 몰래 숨겨둔 종이를 잃어버렸을 때의 대가가 그렇게나 큰 금액의 손실이라면, 과연 그 종이의 안위를 걱정하지 않고 발 뻗고 잠을 자거나 집을 비울 수 있을지 한번 생각해 볼 일이다. 문제는 이처럼 비트코인의 비밀키를 잃어버린 사람들의 사연은 매우 흔하게 찾아볼 수 있는 것이라는 점이다.

연재글

비트코인 속으로 들어간 타원곡선

1. 비트코인 주소
2. 디지털 서명 알고리듬
3. 다중서명의 세계

은행의 계좌에 천억 원을 넣어두고 잘 지내던 어느 날 통장을 확인해보니 그 돈이 모두 사라졌다고 해보자. 당황스럽긴 하겠지만 그래도 정신을 차리고 해야 할 일들을 생각할 수는 있을 것이다. 급한 대로 은행에 전화를 해 볼 수도 있을 것이고, 은행에 직접 찾아갈 수도 있다. 은행의 대응이 미덥지 않다면 경찰서를 찾아가야 할지도 모른다. 그래도 할 수 있는 일들이 있다. 그런데 비트코인의 비밀키를 적어 숨겨둔 종이가 불행히 불에 타버렸다고 하자. 이제는 더 이상 연락해볼 고객센터의 전화번호 같은 것이 없다. 비트코인은 은행과 같은 신뢰할 수 있는 중개자를 제거하려는 의도를 가지고 탄생한 시스템이기 때문이다. 은행이 없으므로 은행의 고객센터도 있을 리가 없다. 정상적인 사고를 하는 사람이 자신의 소중한 자산을 달랑 종이 한 장의 운명에 맡긴다는 것은 사실 상상하기 어렵다. 그러니 비트코인 이야기에서 숫자 하나를 보관하는 문제의 중요성을 이해할 수 있을 것이다. 이것이 사소하다면 비트코인은 그냥 흥미로운 장난에 불과한 것이다.

이 글에서는 위에서와 같은 불행한 사태를 방지할 수 있게 해주는 다중서명multi-signature, multisig에 대해 살펴볼 것이다. 그리고 타원곡선을 이용한 슈노르 서명Schnorr signature의 개념을 소개하고 이것이 어떤 장점이 있는지 살펴볼 것이다. 이를 응용하여 비밀키를 둘로 쪼개서 보관하는 타원곡선 친화적인 방법도 한번 생각해 볼 것이다.

단일 실패점

지난 10월 15일 판교에 있는 카카오의 데이터센터에 화재가 발생했다. 이는 이후 며칠 동안의 서비스 장애로 이어졌다. 수많은 사람들이 사용하고 있는 카카오톡 메신저를 비롯한 카카오의 대다수 서비스가 정지되었다. 통신·결제·교통 등 일상의 수많은 영역을 카카오톡이라는 단일한 플랫폼에 의존하던 사회에서 이용자들은 크고 작은 피해를 경험할 수밖에 없었다. 접수된 장애 피해 사례는 10만 건이 넘는 것으로 알려졌다. 한국 사회의 약점을 찾는 일에 많은 관심이 있는 사람이라면 판교의 좁은 영역에 불을 내는 것만으로도 사회 전체에 큰 피해를 줄 수 있다는 사실을 직접 경험하며 확인했을 것이다. 세상을 더 좋게 만들고 싶은 사람들과 나쁘게 만들고 싶은 사람들 모두에게 이것은 의미 있는 사건일 수 있다. 이러한 사례는 단

일 실패점single point of failure이라는 개념에 대해 돌아볼 계기를 준다. 단일 실패점이란 제대로 동작하지 않으면 전체 시스템이 실패하거나 작동하지 않게 되는 시스템의 구성 요소를 말한다.

비트코인 시스템은 가능한 모든 부분에서 단일 실패점을 없애려는 것을 목표로 설계되어 있다. 전통적인 금융 시스템에서의 거래는 은행과 같은 신뢰할 수 있는 중개자를 통해 수행된다. 지배적인 영향력이 있는 중개자는 단일 실패점이 될 수 있다. 카카오톡이 멈추면 메시지 전달이 되지 않듯이, 은행이 멈추면 송금이 불가능하다. 은행도 서비스 장애를 일으킬 수 있고 더 심각한 경우 고객의 예금을 잘못 관리할 수도 있다. 극단적인 경우를 찾지 않더라도 자정 무렵에는 많은 은행이 시스템 점검이라는 이유로 서비스를 잠시 동안 중단하는 일이 매일 일어난다. 비트코인은 네트워크에 참여하는 모든 노드가 거래를 확인하고 전파하며 장부를 저장한다. 동일한 장부를 모두가 중복해서 저장하는 것은 가장 효율적인 방식은 아닐 것이다. 하지만 하나의 노드가 다운되더라도 다른 노드가 여전히 동일한 장부를 가지고 있기에 네트워크는 시스템 점검을 이유로 하는 서비스 중단없이 계속 작동할 수 있다.

다중서명

비트코인의 디자인에 담긴 이런 속성과 별개로 개인 사용자의 입장에서는 비트코인의 비밀키가 단일 실패점의 성격을 가질 수도 있다. 비밀키에 대한 접근 권한이 사라지면, 비트코인을 움직일 방법도 사라진다. 비밀키를 통해 지켜지고 있는 가치를 너무나 쉽게 잃게 될 수 있다는 점은 키의 소유자에게 불안을 안겨준다. 그렇기 때문에 많은 사람은 비트코인을 신뢰할 수 있는 거래소에 보관하는 방법을 선택하게 된다. 그리고 그런 거래소는 물론 고객을 배신할 수 있다. 이 글을 준비하던

지난 11월에도 세계 3위의 암호화폐 거래소였던 FTX가 고객의 예치금을 모두 허공에 날리고 파산 신청을 하는 일이 발생했다. 신뢰할 수 있는 중개자를 없애려는 시도가 다시 신뢰할 수 있는 중개자를 불러오고, 그 중개자는 다시 신뢰를 배반하는 일이 끝없이 반복된다.

"

비트코인 시스템은 가능한 모든 부분에서 단일 실패점을 없애려는 것을 목표로 설계되어 있다.

"

이러한 비밀키라는 단일 실패점의 문제를 해결하는 데 활용할 수 있는 개념이 바로 다중서명이다. 다중서명이란 여러 서명자가 단일한 메시지에 공동의 서명을 생성하는 방식을 말한다. 여러 개의 열쇠가 필요한 자물쇠에 비유할 수 있다. 비트코인 시스템에서는 다중서명 방식으로 작동하는 주소를 생성하여 사용할 수 있다. 이 경우는 비트코인 타원곡선의 한 점을 공개키로 쓰는 것이 아니라, 타원곡선의 여러 점을 선택하여 공개키로 사용한다. 타원곡선의 여러 점을 선택하더라도 점들의 좌표를 모아서 해시함수를 적용할 수 있으므로 여러 개의 공개키를 가지고 하나의 비트코인 주소를 생성할 수 있다. 각각의 공개키에 대한 비밀키 중에서 몇 개 이상을 사용해 서명할 것인지도 설정할 수 있다. 하나의 주소에 들어있는 비트코인을 옮기기 위해 2개, 3개 또는 그 이상의 비밀키를 이용해 서명해야 한다는 점에서 이는 비밀키가 가진 단일 실패점의 성격을 완화해준다. 설정된 개수 이상의 서명을 제출하면 다른 사람들이 이를 공개키의 목록과 비교해 각각의 서명이 올바른지 그리고 다중서명의 조건이 충족되는지를 검증할 수 있다.

이러한 다중서명은 다양한 방식으로 활용이 가능하다. 예를 들어 가족 구성원 두 명이 두 개의 공개키를 가지고 하나의 주소를 생성한 뒤에 각자 비밀키를 보관하는 방식을 생각할 수 있다. 적은 금액을 함께 관리하며 사용한다면 둘 중 하나의 비밀키만 사용해 서명해도 거래가 되는 방식을 채택할 수 있을 것이다. 장기적으로 보관하려는 큰 금액이라면 두 개의 비밀키를 모두 사용해 두 개의 서명을 제출해야만 하는 방식을 생각할 수 있다. 한 명에게 사고가 생기는 경우를 대비해, 다른 신뢰할 수 있는 사람을 추가해 세 개의 비밀키 중에서 두 개 이상을 사용해 서명하는 방식도 생각해 볼 수 있다. 큰 조직에서 관리하는 공동의 자금이라면 다수의 관리자 중에서 일정 비율 이상의 서명을 얻어야만 거래가 되도록 만들 수도 있다. 이러한 방식이 가능하다면 비밀키의 단일 실패점으로서의 성격은 크게 줄어들 수 있다.

슈노르 서명 타원곡선을 이용한 또다른 서명 알고리듬

다중서명에 대해 이야기할 때, 앞으로 더 많이 등장할 수 있는 개념은 슈노르 서명이라 불리는 것이다. 슈노르 서명은 1990년 독일의 암호학자인 클라우스 슈노르에 개발된 서명 알고리듬이다.[2,3] 타원곡선을 이용해 서명을 생성한다는 점에서는 지난 글에서 살펴본 타원곡선 디지털서명 알고리듬, ECDSA와 유사하다. 하지만 ECDSA에 비해 다중서명의 관점에서 큰 장점을 가지고 있다. 이는 2021년 11월의 비트코인 탭루트 업그레이드를 계기로 비트코인 시스템에 도입되었다.

ECDSA를 이해하는데 약간의 노력을 들인 이후라면, 슈노르 서명도 큰 어려움없이 받아들일 수 있다. 그러니 여기서 ECDSA의 절차를 복기해보도록 하자. 비트코인은 고정된 소수 $p = 2^{256} - 2^{32} - 977$와 유한체 $\mathbb{F}^p$ 위의 타원곡선 $y^2 = x^3 + 7$을 사용하고 있다. 점 $G = (x_1, y_1)$는 이 타원곡선의 특별한 점이며 그 좌표 (x_1, y_1)는 지난 글에서 확인할 수 있다. 이 점을 반복하여 더하면 비트코인 타원곡선에 있는

$$n = 115792089237316195423570985008687907852837564279074904382605163141518161494337$$

개의 점을 차례로 얻을 수 있다. 비트코인 주소의 비밀키는 n보다 작은 자연수 k, 공개키는 타원곡선의 점 $P = kG$라는 사실은 그동안 여러 번 반복하였다. ECDSA에서의 서명은 다음과 절차를 따른다:

1. 1부터 $n - 1$사이의 임의의 자연수 ℓ을 선택 (남에게 공개하면 안 되는 정보)
2. 선택한 자연수 ℓ을 이용해 타원곡선의 점 $R = \ell G = (r, y)$을 계산 (남에게 공개하는 정보)
3. 전파하려는 메시지 m에 해시함수 h를 적용해 자연수 $z = h(m)$를 계산 (남에게 공개하는 정보)

이제 타원곡선 위의 세 점 G, P, R과 자연수 z, r이 얻어졌다. 여기서 미지수를 s로 갖는 ECDSA의 핵심 방정식

$$zG + rP = sR \quad \cdots \quad (1)$$

을 생각할 수 있다. 메시지의 해시 z에 대한 서명 (R, s)을 만들어내려면 이 방정식을 성립하도록 만드는 자연수 s를 찾아야 한다. 비밀키 k와 임의의 자연수 ℓ을 이미 알고 있는 서명자는

$$s = \ell^{-1}(z + rk) \mod n \quad \cdots \quad (2)$$

를 손쉽게 찾을 수 있다.

슈노르 서명은 ECDSA의 절차를 약간 변화시켜 얻을 수 있다. 구체적으로는 다음과 같다:

1. 1부터 $n - 1$사이의 임의의 자연수 ℓ을 선택 (남에게 공개하면 안 되는 정보)
2. 선택한 자연수 ℓ을 이용해 타원곡선의 점 $R = \ell G$을 계산 (남에게 공개하는 정보)
3. 공개키 P, 위에서 선택한 점 R, 전파하려는 메시지 m에 해시함수 h를 적용해 자연수 $z = h(P, R, m)$를 계산 (남에게 공개하는 정보)

이렇게 타원곡선 위의 세 점 G, P, R과 자연수 z를 갖게 되었다. ECDSA의 방정식(1)에 대응되는 슈노르 서명의 방정식은 미지수를 s로 갖는 방정식

$$sG = R + zP \quad \cdots \quad (3)$$

이다.

ECDSA의 경우처럼 s에 대한 방정식(3)은 G, P, R과 z를 알고 있다고 해서 풀 수 있는 것이 아니다. 하지만 비밀키의 소유자가 알고 있는 $P = kG$와 $R = \ell G$라는 사실을 이용하면

$$sG = \ell G + zkG$$

가 되어, 서명에 필요한 수

$$s = \ell + zk \mod n \quad \cdots \quad (4)$$

를 쉽게 찾을 수 있다.

식 (1)와 (3), 그리고 (2)와 (4)를 각각 비교해보면 ECDSA에서보다 슈노르 서명에서의 식이 더 간단하다는 것을 알 수 있다. 이는 계산에 필요한 비용을 줄여준다. 계산상의 이점이 보안성을 희생해서 얻어진 것도 아니므로, 이것은 좋은 점이라고 할 수 있다. 하지만 슈노르 서명에는 더 놀라운 점이 있다.

슈노르 서명의 응용 비밀키를 두 개로 쪼개 사용하기

비밀키를 생성해 보관하는 일은 어려운 일이다. 그에 대해 이제 이러한 방법을 생각해 보자. 숫자 두 개 k_1, k_2를 임의로 선택해 비밀키는 그 둘의 합 $k = k_1 + k_2$으로 택하는 것이다. 그리고 두 조각의 비밀키 k_1, k_2를 두 대의 컴퓨터에서 따로 관리한다. 이렇게 하면 k를 하나의 컴퓨터에 보관하는 것보다는 해킹이나 도둑맞을 위험으로부터 더 안전하다. 물론 k_1나 k_2 중에서 하나만 분실해도 비밀키를 복구하지 못한다는 점은 여전히 문제이므로, 두 조각의 비밀키 k_1과 k_2에 대한 복사본을 만들어 두는 것이 좋을 것이다. 이것은 k에 대한 복사본을 만드는 것보다 훨씬 안전하다. 이렇게 단일 실패점을 더 제거할 수 있게 된다.

흥미로운 것은 비밀키 k를 직접 사용하지 않고도 이에 대한 비트코인 주소를 얻어낼 수 있다는 점이다. 왜냐하면 비밀키 k에 대한 공개키 kG는 두 조각의 비밀키 각각에 대한 공개키 k_1G, k_2G를 더한 $k_1G + k_2G$와 같은 것이기 때문이다. 그렇기 때문에 비밀키 k에 대한 비트코인 주소는 비밀키 k 없이도 두 공개키의 합 $k_1G + k_2G$을 구한 다음 해시함수를 적용해 얻어낼 수 있다. 타원곡선의 덧셈은 어떤 컴퓨터도 k를 직접 손대지 않은 상태에서 쪼개진 비밀키만을 이용해 k에 대한 비트코인 주소를 생성하는 길을 열어준다.

그러나 이렇게 두 조각으로 나뉘어 보관되는 비밀키에는 문제가 있다. 왜냐하면 비밀키는 궁극적으로 서명을 하기 위해 존재하는 숫자이기 때문이다. 비밀키 k를 이용해 ECDSA를 통해 서명을 하려면 결국 따로 보관하던 k_1과 k_2를 더한 다음에 절차를 진행해야 한다. 서명을 하는 순간에는 둘을 더해 얻어지는 본래의 비밀키 k를 컴퓨터에 불러와야만 한다. 그리고 그 순간 단일 실패점이 생겨난다. 운이 없어 그때 해킹을 당한다면 비밀키를 탈취당할지도 모른다.

여기서 슈노르 서명을 활용하는 방법을 생각해보자. 쪼개진 비밀키를 보관하는 두 대의 컴퓨터를 이용해 다음과 같은 절차를 따른다:

1. 컴퓨터1, 2는 각각 두 조각의 비밀키 k_1, k_2를 보관 (두 대가 서로 공유하지 않는 정보)

2. 컴퓨터1, 2에서 각각 두 조각의 공개키 $P_1 = k_1 G, P_2 = k_2 G$를 계산하여 공유
3. 컴퓨터1, 2에서 각각 임의의 자연수 ℓ_1, ℓ_2를 선택 (두 대가 서로 공유하지 않는 정보)
4. 컴퓨터1, 2에서 각각 점 $R_1 = \ell_1 G, R_2 = \ell_2 G$을 계산하여 공유
5. 컴퓨터1, 2에서 각각 두 공개키의 합 $P = P_1 + P_2$, 두 임의의 점의 합 $R = R_1 + R_2$, 전파하려는 메시지 m에 해시함수 h를 적용해 자연수 $z = h(P, R, m)$를 계산
6. 컴퓨터1, 2에서 각각 슈노르 서명 (4)의 조각 $s_1 = \ell_1 + zk_1, s_2 = \ell_2 + zk_2$ 을 계산하여 공유

두 대의 컴퓨터에서 이런 절차를 수행하면, 타원곡선 위의 세 점 G, P, R과 자연수 z, 그리고 두 대의 컴퓨터가 각각의 조각 비밀키로 만들어낸 슈노르 서명의 조각 s_1, s_2가 얻어지게 된다.

좀 복잡하게 들릴 수도 있지만 핵심은 지금까지 어디에도 본래의 비밀키 k가 사용되지 않았다는 점이다. 그리고 서로 공유되는 정보를 이용하더라도 컴퓨터1에서는 비밀키 k_2를 알아낼 수 없고 컴퓨터2도 k_1을 알아내지 못한다. 이제 이 슈노르 서명의 두 조각 s_1, s_2를 더하여 얻어지는 수를 s로 두면, (R, s)는 다름 아닌 본래의 비밀키 k를 이용해 메시지 해시 z에 대해 생성한 슈노르 서명과 같다. 왜냐하면,

$$s = \ell + zk = (\ell_1 + \ell_2) + z(k_1 + k_2) = s_1 + s_2$$

이기 때문이다.

이것은 정말 놀라운 일이다. 이 글을 준비하며 생각을 하는 동안 나는 이것이 충격적으로 우아하다는 생각을 열 번도 넘게 한 것 같다. 이러한 절차를 따르면 어느 컴퓨터도 k라는 숫자를 본 적이 없고 k를 알아낼 수도 없지만, 그 와중에도 우리는 k에 대한 비트코인 주소를 만들고 서명을 생성할 수 있다. 이것은 타원곡선이 가지고 있는 덧셈의 구조, 그리고 슈노르 서명의 식 (4)가 가진 선형성이 복합적으로 만든 결과이다. 숫자 하나를 안전하게 보관하여 활용하는 일의 어려움에 대해 고민해 본 이후라면 이런 일이 가능하다는 것은 쉽게 믿기지 않는다. 내가 느낀 놀라움을 이 글을 읽는 이들도 경험할 수 있기를 소망해본다.

서명을 이런 방식으로 생성해서 외부에 공개할 때, 다른 사람들은 이 서명 (R, s)가 실제로 k를 직접 이용해 얻어졌는지 k_1, k_2를 이용해 얻었는지 알지 못한다. 다시 말해 밖에서는 다중서명이 사용되었는지 아닌지 알 수 없다. 이는 슈노르 서명을 통한 다중서명이 가져다주는 프라이버시 상의 이점이다. 또한 위의 과정을 검토해보면 비밀키를 두 조각으로 쪼개든 백 조각으로 쪼개든 이를 적용하는 데는 큰 차이가 없음을 알 수 있다. 이것은 비트코인에 다중서명의 활용을 넓히는 데도 도움이 된다. 처음에 설명한 고전적인 방식의 다중서명이라면 여러 명이 서명할 때 여러 개의 서명을 모두 장부에 기록해야 하는데, 이제는 하나의 서명만 장부에 기록해도 되기 때문이다. 장부를 모든 참여자가 중복해서 저장하는 비트코인의 속성상 이러한 데이터의 절약은 분산화를 더 폭넓게 실현하는 수단이 된다. 슈노르 서명을 비트코인에 응용하는 방법에 대한 더 세심하고 진전된 논의가 궁금하다면 [4]를 참고할 수 있다.

맺음말

타원곡선에 있는 덧셈 구조를 설명하며 시작한 비트코인에 숨겨진 타원곡선 이야기는 이 정도로 마무리하려 한다. '타원곡선에 대해서는 끝없이 쓸 수 있다'고 했던 서지 랭의 말은 아마 비트코인에 대해서도 적용되는 말일 것이다. 비트코인은 최근의 발명이지만 그 뒤에는 넓은 영역에 걸쳐있는 기나긴 지식의 계보가 존재하고 있다. 그동안 나의 짧은 식견을 동원해 작성한 글이 타원곡선 덕후들과 비트코인에 관심 있는 이들 사이의 머나먼 세계를 서로에게 소개해주는 정도의 역할이나마 했기를 바란다. 자칫 정체불명이 될 수도 있었던 소재에 흔쾌히 지면을 허락해준 호라이즌의 편집위원들께도 감사의 인사를 드린다.

참고문헌

1. 닉 허틀리, 비트코인 2400억원어치 저장된 컴퓨터 버린 남성... 9년째 쓰레기장 수색, BBC 코리아, 2022-8-3
2. Schnorr, C.P. Efficient Identification and Signatures for Smart Cards. (1990). In Brassard, G. (eds) Advances in Cryptology — CRYPTO' 89 Proceedings. CRYPTO 1989. Lecture Notes in Computer Science, vol 435.
3. Schnorr, C.P. Efficient Signature Generation by Smart Cards. (1991). Journal of Cryptology 4(3), 161–174
4. Maxwell, G., Poelstra, A., Seurin, Y. et al. Simple Schnorr multi-signatures with applications to Bitcoin. (2019). Des. Codes Cryptogr. 87, 2139–2164