KGZ protocol

Commitment Scheme이란 committer가 어떤 값(또는 다항식)에 대해 사전에 공개 약속(commit)하고, 이를 나중에 검증 가능하도록 하는 메커니즘입니다.이러한 scheme는 두가지 특성을 만족해야 합니다.

1. Binding: Committer는 한 번 commit한 값을 나중에 변경할 수 없습니다. 즉, 동일한 commitment에 대응하는 서로 다른 두 값 A와 B를 찾는 것(commit(A) = commit(B), A ≠ B)이 계산상 불가능해야 합니다.

2. Hiding: Commitment 값 자체만으로는 원래의 숨겨진 값에 대한 정보를 노출하지 않아야 합니다. 즉, verifier는 commit한 값만 봐서는 원본 값이 무엇인지 알 수 없어야 합니다.

이러한 scheme는 data availability를 증명할 때 사용됩니다. 실제로 eip 4844에서는 scalability를 위해 blob 데이터를 KGZ protocol을 이용해 DA를 증명하는 방식을 제안했고, L2 가스비가 기존 대비 10~100배 감소했습니다.

KGZ protocol은 크게 두 가지 가정을 기초로 합니다.

1. DL Assumption: p가 매우 큰 소수이고 g,d,p가 주어질 때, 다음을 만족하는 n을 구할 수 없습니다. $g^n \equiv d \pmod{p}$

2. t-SDH Assumption: $< g, g^a, g^{a^2}, \ldots, g^{a^t} >$ 가 주어질 때, c ≠ -a인 모든 c에 대해서 $\langle c, g^{1/(a+c)} \rangle$를 구할 수 없습니다.

committer가 차수가 t인 polynomial $\phi(x)=\sum_{j=0}^{t} \phi_j x^j$를 commit할 때, KGZ 과정은 다음과 같습니다.

1) setup: public key(PK)와 symmetric bilinear pairing e를 생성합니다. 이 때, secret key는 α이며, public key는 $\langle g, g^{\alpha}, \ldots, g^{\alpha^t} \rangle$입니다. 그리고 secret key는 버립니다.

2) Commit(PK, $\phi(x)$): C = $g^{\phi(\alpha)}$로 commit합니다. 이 때, public key만 가지고도 commit할 수 있습니다. C = $\prod_{j=0}^{t}(g^{\alpha^j})^{\phi_j}$ 가 성립하므로, public key와 방정식의 계수만 알면 구할 수 있습니다

3) Open: committer는 자신이 commit한 값 $\phi(x)$를 다른 사람에게 알립니다

4) verify(PK, $\phi(x)$, C): verifier는 commit값과 open된 polynomial이 바르게 commit 되었는지 검증합니다. open한 polynomial을 2번 과정과 똑같이 진행해 기존에 commit된 C와 같은지 확인하면 됩니다.

5) CreateWitness(PK, $\phi(x)$, i): committer는 witness $\langle i, \phi(i), w_i \rangle$를 생성해 verifier에게 나누어줍니다. $\psi_i(x) = \frac{\phi(x) - \phi(i)}{(x - i)}$일 때, $w_i = g^{\psi_i(\alpha)}$입니다.

6) VerifyEval(PK, C, I, $\phi(i)$, w_i): 해당 witness가 올바르게 C를 생성했는지 다음과 과정을 통해 확인합니다. $e(C, g)? = e(w_i, g^\alpha / g^i)e(g, g)^{\phi(i)}$ 만약 C와 witness가 올바르다면 다음과 같은 전개를 통해 식이 성립합니다.

$$\begin{aligned} e(w_i, g^\alpha / g^i)e(g, g)^{\phi(i)} &= e(g^{\psi_i(\alpha)}, g^{(\alpha - i)})e(g, g)^{\phi(i)} \\ &= e(g, g)^{\psi_i(\alpha)(\alpha - i) + \phi(i)} \\ &= e(g, g)^{\phi(\alpha)} = e(C, g) \text{ as } \phi(x) = \psi_i(x)(x - i) + \phi(i) \end{aligned}$$

이제 각 속성이 binding과 hiding을 만족하는지 증명하겠습니다.

‘A를 해결하는 것이 불가능할 때, B를 해결하는 것은 불가능하다’ 라는 명제를 증명하는 패턴은 다음과 같습니다.

1) B를 해결 가능하다고 가정합니다

2) B를 통해서 A를 해결할 수 있음을 증명합니다. 그러면 대우가 성립함으로 증명이 완료됩니다.

1. Binding 증명: t-SDH assumption이 성립할 때, C = $g^{\phi(\alpha)} = g^{\phi'(\alpha)}$ 인 $\phi(x) \neq \phi'(x)$ 를 만들 수 없다

증명: C = $g^{\phi(\alpha)} = g^{\phi'(\alpha)}$ 인 $\phi(x) \neq \phi'(x)$ 를 만들 수 있다고 가정하겠습니다. $\phi''(x) = \phi(x) - \phi'(x)$ 로 정의하면 $g^{\phi''(\alpha)} = 1$ 이므로 $\phi''(\alpha)$ = 0이 됩니다. 따라서 α 값을 직접 구할 수 있으므로, t-SDH 가정이 깨지게 됩니다.

2. Hiding 증명: DL assumption이 성립한다고 하자. verifier가 Commitment C, witness $\langle i, \phi(i), w_i \rangle$ for $i \in \{1,\ldots,t\}$에 대해 알 때, 모든 $i' \notin \{1,\ldots,t\}$인 $\phi(i')$를 알지 못한다. (만약 verifier가 $\phi(i')$ 알게 된다면 Lagrange Interpolation에 의해 polynomial을 복구할 수 있습니다.)

증명: B가 g^a를 받았고 a를 구하고자 합니다. 그러면 B는 Public Key를 $\langle g, g^{\alpha}, \ldots, g^{\alpha^t} \rangle$ 로 구성합니다. 그리고 commit하고자하는 polynomial $\phi(0)$ = a로 가정합니다. 이렇게 가정할 수 있는 이유는 상수항의 값을 몰라도 commitment를 만들 수 있기 때문입니다.그리고 위의 scheme을 바탕으로 총 t+1개의 witness를 받습니다. 그러면 Lagrange Interpolation에 의해 $\phi(x)$를 구할 수 있고, 따라서 a = $\phi(0)$를 구할 수 있습니다.

KGZ는 Feldman VSS, EIP-4844 등 다양한 곳에서 사용되고 있습니다.

Reference

1. A. Kate, G. M. Zaverucha, and I. Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In Proceedings of ASIACRYPT'10, volume 6477 of LNCS, pages 177–194. Springer, 2010.

2. EIP-4844, https://github.com/ethereum/EIPs/blob/master/EIPS/eip-4844.md