BULLETPROOFS

FROM ZERO (KNOWLEDGE)
TO BULLETPROOFS

REFERENCES

- Adam Gibson, “From Zero (Knowledge) to Bulletproofs”

REFERENCES

- Groth, Jens. "Linear algebra with sub-linear zero-knowledge arguments." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 2009.
- Bootle, Jonathan, et al. "Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2016.
- Bünz, Benedikt, et al. "Bulletproofs: Short proofs for confidential transactions and more." 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018.

BEFORE READ THE PAPER

- 표기법
  - 정수(스칼라)는 평문 소문자: x
  - 타원 곡선 상의 점은 대문자: X
  - 스칼라배(scalar multiplication)는 기호 생략: xY
  - 벡터는 굵은 글씨: $\mathbf{x}, \mathbf{X}$
  - 행렬은 mathbb: $\mathbb{X}$
  - 식 $a_1G_1 + a_2G_2 + \ldots + a_nG_n$은 두 벡터의 내적 $\mathbf{aG}$으로 표기

INTRODUCTION

INTRODUCTION

- 목표
  - “벡터를 공개하지 않고서, 연루된 커밋(commits)으로부터 알고 있음을 증명”
  - “원본을 숨기며, 커밋된 두 벡터의 내적이 0임을 증명”

INTRODUCTION

- 왜 “벡터들을 공개하지 않고서, 그 벡터들을 알고 있음을 증명”하는 것이 유용한가?
- 왜 하필이면 벡터와 내적인가?
  - 벡터의 집합을 행렬로 인코딩할 수 있음
  - 내적만이 아니라 행렬곱, 역행렬, 아다마르 곱(Hadamard product)으로 확장 가능
  - 삼각행렬(triangular matrix)인 경우에는 각종 기이(!)하고 유용한 연산 가능

COMMITMENT SCHEMES

COMMITMENT SCHEMES

- 암호학적(Cryptographic) 커미트먼트는 강력한 기술
  - 나중에 참이었음을 밝히기 전에,
  - 무엇인가를 지금은 숨긴 상태에서 참인 것으로 확약하고 싶을 때 사용

COMMITMENT SCHEMES

- 커미트먼트를 구현하는 방법 1
- 단방향 함수(one-way function)을 사용
  - 암호학적 해시 함수
  - 값 2를 커밋하고 싶으면, 2에 대한 해시를 전송:
  - “53c234e5e8472b6ac51c1ae1cab3fe06fad053beb8ebfd8977b010655bfdd3c3”

COMMITMENT SCHEMES

- 문제
  - 고작 2에 대해 “53…c3”을 전송해야 함
  - 의미상(semantic) 보안의 결핍
    - 누군가가 2와 “53…c3”의 연관성을 알아낸다면
    - 더 이상 은닉성을 제공하지 못함

COMMITMENT SCHEMES

- 커미트먼트를 구현하는 방법 2
- 무작위 값을 섞어서 커밋
  - (2 + 무작위 값) 조합을 커밋
  - $SHA256(secret\_number || random\_characters)$

COMMITMENT SCHEMES

- 커미트먼트를 구현하는 방법 2
- 무작위 값을 섞어서 커밋
    - 나중 단계에서 “값”과 "무작위 값”의 공개를 요청
    - 방법 1과 동일한 수준의 보안 속성 제공

COMMITMENT SCHEMES

- 방법 2는 커미트먼트 스킴이 가지는 주요한 두 속성을 모두 충족
  - 은닉(Hiding): 커미트먼트 C는 커밋한 값을 드러내지 않음
  - 구속(Binding): 커미트먼트 $C(m)$을 m으로부터 만든 이상,
    다른 메시지 m'으로부터 만들었다고 주장할 수 없음

COMMITMENT SCHEMES

- 문제
  - 해시 함수는 동형(homomorphic)의 특징을 갖지 않음
  - $SHA256(a) + SHA256(b) = SHA256(a+b)$가 성립하지 않음
  - 활용하기 좋은 수단이 아님
- 동형 커미트먼트 스킴은 어떻게 만들 수 있을까?

EC PEDERSEN COMMITMENT

- 타원 곡선 형태의 페더슨 커미트먼트
  - 해시 함수를 사용하는 것 대신에, 타원 곡선을 활용
- 타원 곡선계에서의 단방향 함수
  - 곡선 상의 점에 대한 스칼라배

EC PEDERSEN COMMITMENT

- 타원 곡선 상의 점에 대한 스칼라배
- $C = rH + aG$
  - C : 커미트먼트로 사용할 곡선 상의 점
  - a : 커밋할 값(숫자), r : 은닉을 제공하기 위한 무작위 값
  - G : 생성자 점
  - H : 또 다른 곡선 상의 점,
    아무도 $H = qG$를 만족하는 이산 로그 q를 몰라야 함
    (Nothing Up My Sleeve, NUMS)

EC PEDERSEN COMMITMENT

- 페더슨 커미트먼트는 동형의 특징을 가짐
  - $C(r_1, a_1) + C(r_2, a_2)$
  - $= r_1H + a_1G + r_2H + a_2G$
  - $= (r_1 + r_2)H + (a_1 + a_2)G$
  - $= C(r_1 + r_2, a_1 + a_2)$
- 커미트먼트들의 합은, 각 값의 합과 무작위 값의 합을 커미트먼트한 것과 같음

VECTOR PEDERSEN COMMITMENT

- 벡터 페더슨 커미트먼트
  - $C = rH + aG$ 를 더 강력한 형태로 확장
  - $C = rH + (v_1G_1 + v_2G_2 + ... + v_nG_n) = rH + \mathbf{vG}$
- 간결함에서 주된 효과
  - 벡터는 임의의 큰 숫자의 원소들로 구성될 수 있으나,
  - 오직 단 하나의 점 C에 커밋됨

VECTOR PEDERSEN COMMITMENT

- 2개 또는 더 많은 벡터를 포괄하도록 확장 가능
  - 각 벡터를 위한 N개의 NUMS 기반 곡선 상의 점만 있으면 됨
  - $C = rH + \mathbf{v}\mathbf{G} + \mathbf{w}\mathbf{H}$

ZERO KNOWLEDGE ARGUMENTS

ZERO KNOWLEDGE ARGUMENTS

- 지식 논의(Knowledge arguments)
  - 증명이 오직 계산적임을 내포
  - 지식 증명(proofs)과는 구분되는 기술 용어

ZERO KNOWLEDGE ARGUMENTS

- 영지식 증명은 확률적인 건실성을 제공하지만,
- 영지식 논의는 연산적 건실성을 제공
  - 충분한 연산 능력(매우 클 것이지만)을 가진 상대는
  - 심지어 실제로는 그렇지 않은 경우에도
  - 아마도 그가 비밀 값(들)을 알고 있다고 상대를 납득시킬 수 있음

ZERO KNOWLEDGE ARGUMENTS

- 지식 논의
  - 각자 $N(\neq m)$ 차원을 가지는 m개의 벡터 $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}$
  - 각자의 커미트먼트 집합을 생성하고 나서 논할 수 있음
  - $$C_1 = r_1 H + \mathbf{x_1} \mathbf{G}$$
    $$C_2 = r_2 H + \mathbf{x_2} \mathbf{G}$$
    $$\cdots$$
    $$C_m = r_m H + \mathbf{x_m} \mathbf{G}$$

ZERO KNOWLEDGE ARGUMENTS

- 커미트먼트들은 완벽한 은닉을 제공하므로
  - 이들로부터 벡터를 알아낼 수 없음
- 이 커미트먼트들을 공유
- 이제 이들 모두를 개방할 수 있다는 영지식 관련 논의를 수행할 수 있음

ZERO KNOWLEDGE ARGUMENTS

- 영지식 논의 상호 절차
  - $P \rightarrow V : C_0$ (새롭게 선택한 N차원의 무작위 벡터에 대한 새 커미트먼트)
  - $V \rightarrow P : e$ (무작위 스칼라)
  - $P \rightarrow V : (\mathbf{z}, s)$ (N차원의 벡터와 스칼라)
- 마지막 두 값은:
  - $\mathbf{z} = \Sigma_{i=0}^{m} e^i \mathbf{x}_i$ 와 $s = \Sigma_{i=0}^{m} e^i r_i$ 로 계산
  - 덧셈(Sigma)이 0부터 시작함에 유의

ZERO KNOWLEDGE ARGUMENTS

- 어떻게 벡터 $\mathbf{x}$를 은닉시키는가?
  - $\mathbf{z}$의 한 요소 $\mathbf{z_2} = x_{02} + ex_{12} + \ldots + e^m x_{m2}$ 는
  - 벡터의 값을 덧셈으로 은닉

ZERO KNOWLEDGE ARGUMENTS

- $(\mathbf{z}, s)$를 받으면, 검증자는 이 증명이 유효한지를 다음과 같이 검증
  - 기호 =? 는 등호의 좌변과 우변이 같은지를 확인한다는 의미
  - 만일 같다면 증명은 참
  - 그렇지 않으면 증명은 거짓
- $$\sum_{i=0}^{m} e^i C_i =? \, sH + \mathbf{z}\mathbf{G}$$

ZERO KNOWLEDGE ARGUMENTS

- 완결성(Completeness)
  - 유효성 검사의 우변을 확장: $sH + \mathbf{z}\mathbf{G}$
    $$= \sum_{i=0}^{m} e^i(r_i H) + \sum_{i=0}^{m} e^i \mathbf{x}_i \mathbf{G}$$
    $$= \sum_{i=0}^{m} e^i(r_i H + \mathbf{x}_i \mathbf{G})$$
    $$= \sum_{i=0}^{m} e^i C_i$$
  - 정직한 증명자는 검증자를 납득시킬 수 있음

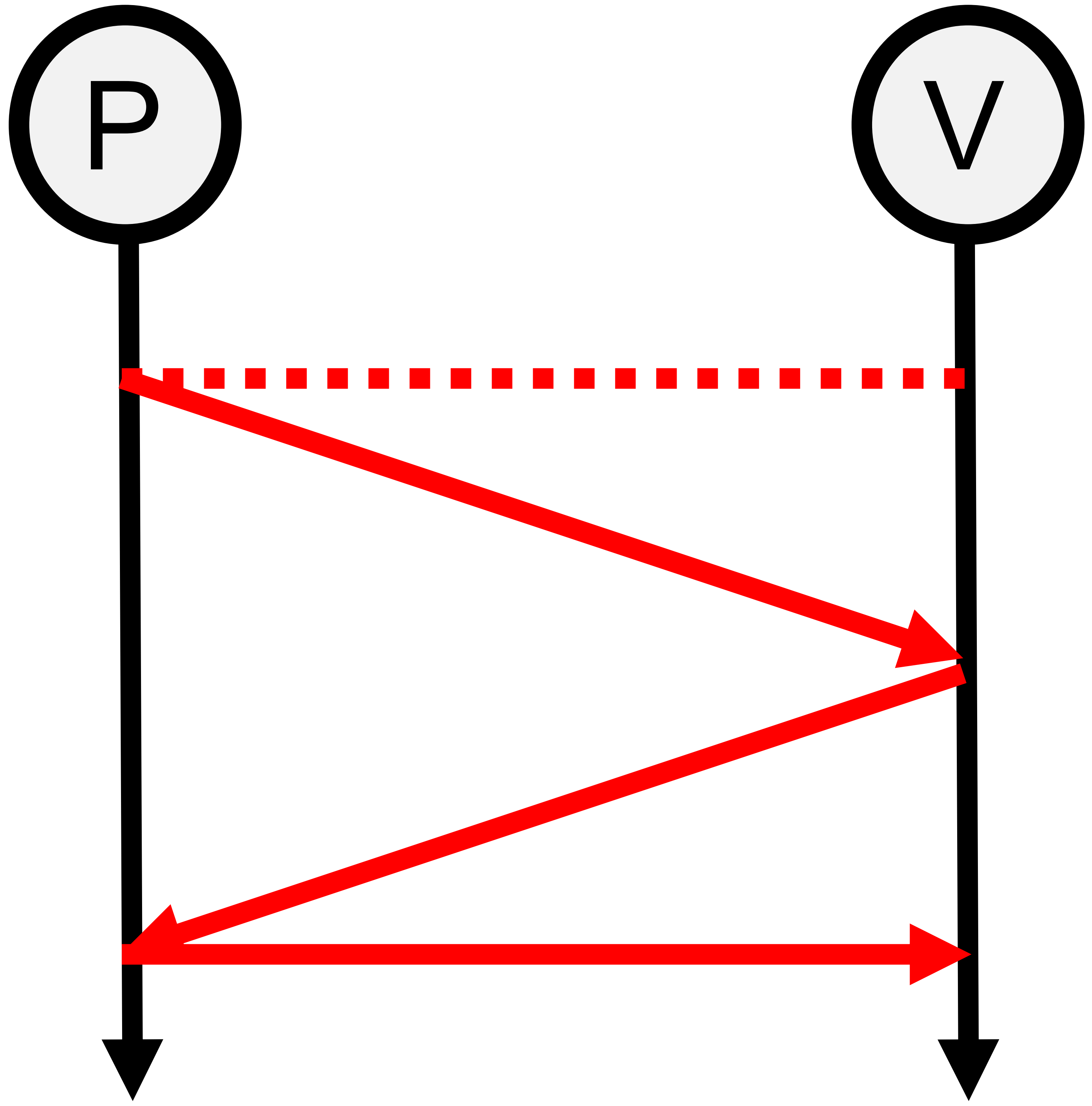
ZERO KNOWLEDGE ARGUMENTS

- 건실성(Soundness)과 영지식성(zero knowledge)
  - 생략

SIGMA PROTOCOL

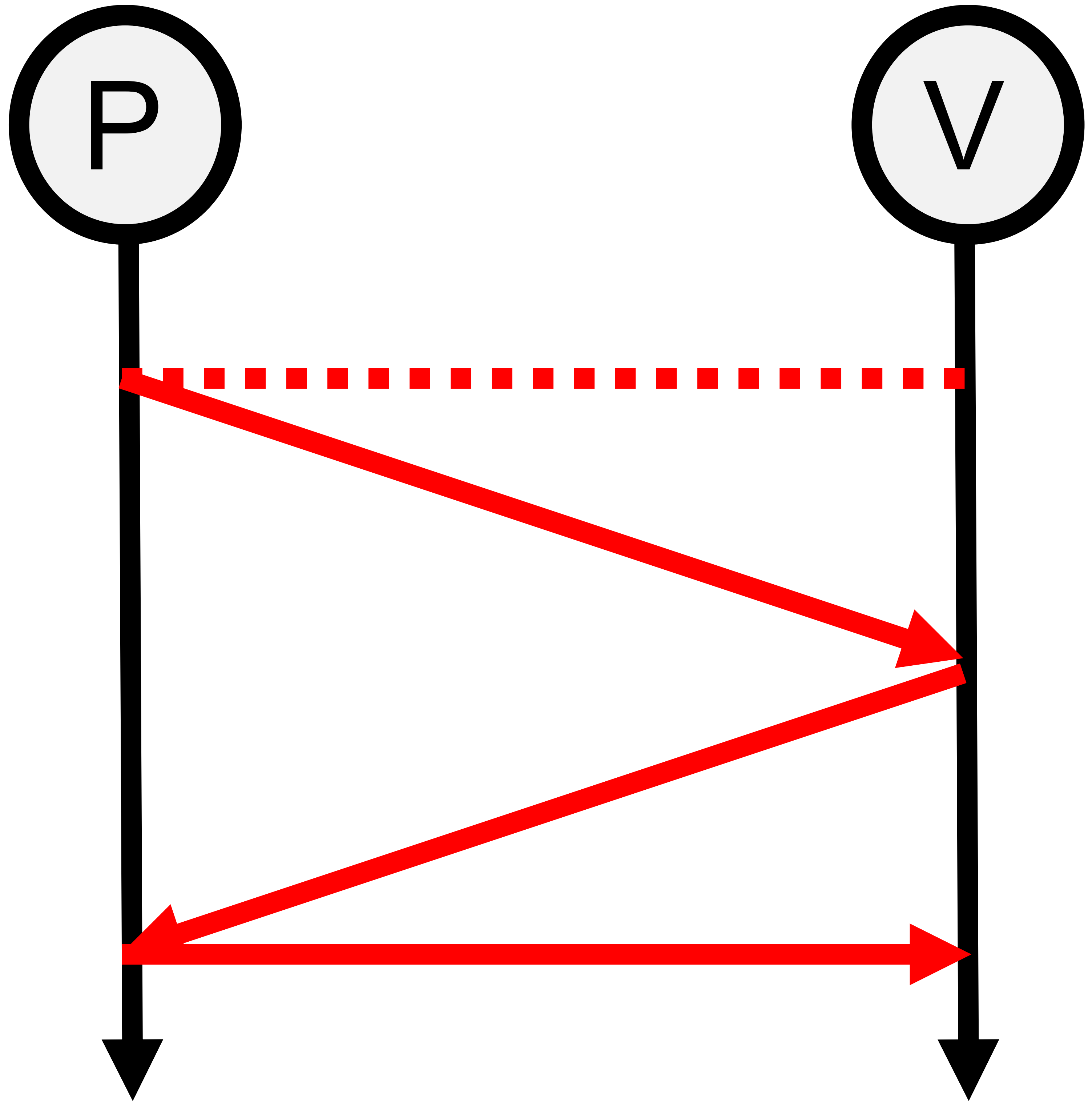
SIGMA PROTOCOL

- 상호 절차가 시그마(Σ)와 유사해 붙여진 이름
  - 증명자 —> 검증자
  - 검증자 —> 증명자
  - 증명자 —> 검증자



SIGMA PROTOCOL

- 시그마 프로토콜의 일반화
  - $P \rightarrow V$: 커미트먼트(Commitment)
  - $V \rightarrow P$: 챌린지(Challenge)
  - $P \rightarrow V$: 응답(Response) 혹은 증명(Proof)



SIGMA PROTOCOL

- 앞서 살펴본 벡터 집합의 지식 증명은
  - 시그마 프로토콜의 한 예
  - $P \rightarrow V : C_0$ (새롭게 선택한 N차원의 무작위 벡터에 대한 새 커미트먼트)
  - $V \rightarrow P : e$ (무작위 스칼라)
  - $P \rightarrow V : (\mathbf{z}, s)$ (N차원의 벡터와 스칼라)

SIGMA PROTOCOL

- 보다 간단한 예시:
- 슈노르 아이덴디티 프로토콜(Schnorr's identity protocol)
  - $P \rightarrow V : R$ (P만이 $R = kG$인 k를 알고 있는 무작위 곡선 상의 점)
  - $V \rightarrow P : e$ (무작위 스칼라)
  - $P \rightarrow V : s$ (P가 $s = ex + k$로 계산한 값)

INNER PROOF

INNER PROOF

- 내적 증명
  - Groth가 제시한 알고리즘
  - BulletProof의 핵심

INNER PROOF

- 내적 증명
  - 증명자가 두 벡터 $\mathbf{x}$와 $\mathbf{y}$를 가지고
  - 두 벡터의 내적 z을 명백히 알고 있는 상황

INNER PROOF

- 세 커미트먼트
  - $C_z = tH + zG$
  - $C_x = rH + \mathbf{x}\mathbf{G}$
  - $C_y = sH + \mathbf{y}\mathbf{G}$
- 증명자가 할 일은 검증자에게 다음을 납득시키는 것:
  - 세 페더슨 커미트먼트들에 대해 $z = \mathbf{x} \cdot \mathbf{y}$를 만족함

INNER PROOF

- 1. 커미트먼트 단계
- 증명자 P는 각각 $\mathbf{x}$와 $\mathbf{y}$에 상응하는
두 논스(nonce) 벡터 $\mathbf{d}_\mathbf{x}$, $\mathbf{d}_\mathbf{y}$의 커미트먼트를 전송해야 함
- 다음 페더슨 커미트먼트를 전송:

$$
\begin{aligned}
A_d &= r_d H + \mathbf{d}_x \mathbf{G} \\
B_d &= s_d H + \mathbf{d}_y \mathbf{G}
\end{aligned}
$$

- r_d와 s_d는 무작위 값

INNER PROOF

- 1. 커미트먼트 단계
- 내적을 증명해야 하므로,
  두 숨겨진 벡터들의 내적의 커미트먼트를 전송해야 함
- 어떤 정보가 필요한가?
  - 내적의 형태가 어떻게 되는가?

INNER PROOF

- 1. 커미트먼트 단계
- 슈노르 아이덴디티 프로토콜을 상기하면:
- 임의의 챌린지 e에 대한 응답이 $e\mathbf{x} + \mathbf{d}_{\mathbf{x}}$, $e\mathbf{y} + \mathbf{d}_{\mathbf{y}}$ 형태일 것임을 직관적으로 예상할 수 있음
  - 슈노르 아이덴디티 프로토콜에서 $ex + k$에 해당

INNER PROOF

- 1. 커미트먼트 단계
- 숨겨진 형태 $e\mathbf{x} + \mathbf{d}_\mathbf{x}$, $e\mathbf{y} + \mathbf{d}_\mathbf{y}$의 내적 $((e\mathbf{x} + \mathbf{d}_\mathbf{x})\cdot(e\mathbf{y} + \mathbf{d}_\mathbf{y}))$은 e에 대한 이차식의 형태
  - 3개의 계수(coefficients)를 가짐
  - 계수에 대한 커미트먼트를 제공해야 함

INNER PROOF

- 1. 커미트먼트 단계
- e^2에 대한 계수는 이미 $C_z = tH + zG$, $z = \mathbf{x}\cdot\mathbf{y}$로 주어져 있음
- 따라서 2개만 더 추가로 제공하면 됨
  - $C_1 = t_1H + (\mathbf{x}\cdot\mathbf{d_y} + \mathbf{y}\cdot\mathbf{d_x})G$
  - $C_0 = t_0H + (\mathbf{d_x}\cdot\mathbf{d_y})G$

INNER PROOF

- 1. 커미트먼트 단계
- 결론적으로 본 커미트먼트 단계에서 증명자는
  - 4개의 무작위 스칼라 r_d, s_d, t_1, t_0 와
  - 2개의 무작위 벡터 $\mathbf{d_x}, \mathbf{d_y}$ 를 생성
- 이들로부터 생성한 4개의 페더슨 커미트먼트
  - A_d, B_d, C_1, C_0 를 전송

INNER PROOF

- 2. 챌린지 단계
- 검증자가 하나의 스칼라 e를 전송

INNER PROOF

- 3. 응답 단계
- 응답 단계에서 증명자는 다음과 같은 값들을 전송
- $$\mathbf{f}_x = e\mathbf{x} + \mathbf{d}_x$$
  $$\mathbf{f}_y = e\mathbf{y} + \mathbf{d}_y$$
  $$r_x = er + r_d$$
  $$s_y = es + s_d$$
  $$t_z = e^2 t + e t_1 + t_0$$

INNER PROOF

- 3. 응답 단계
- 두 벡터의 숨겨진 형태 $\mathbf{f}_x, \mathbf{f}_y$ 에 대해
  검증자는 커미트먼트를 복구하고 C_x, C_y와 일치하는지 검증
- $eC_x + A_d =? r_x H + \mathbf{f}_x \mathbf{G}$
  $eC_y + B_d =? s_y H + \mathbf{f}_y \mathbf{G}$

INNER PROOF

- 3. 응답 단계
- r_x, s_y는 상응하는 페더슨 커미트먼트에 대해
- 무작위 값을 만들기 위해 사용됨

INNER PROOF

- 3. 응답 단계
- 마지막으로, t_z에 대한 검증이 필요
  - 이 검증으로부터 내적이 올바르다는 것을 확인
  - 페더슨 커미트먼트 식의 무작위 값이 올바르다는 것을 확인

INNER PROOF

- 3. 응답 단계
- $\mathbf{f_x}\cdot\mathbf{f_y}$
  $= (e\mathbf{x} + \mathbf{d_x})\cdot(e\mathbf{y} + \mathbf{d_y})$
  $= e^2 z + e(\mathbf{x}\cdot\mathbf{d_y} + \mathbf{y}\cdot\mathbf{d_x}) + \mathbf{d_x}\cdot\mathbf{d_y}$
  - ($z = \mathbf{x}\cdot\mathbf{y}$ 이므로)

INNER PROOF

- 3. 응답 단계
- 페더슨 커미트먼트 $Comm$에 대해 위 식은
- $Comm(\mathbf{f_x} \cdot \mathbf{f_y})$
  $= e^2 C_z + e(Comm((\mathbf{x} \cdot \mathbf{d_y} + \mathbf{y} \cdot \mathbf{d_x})) + Comm(\mathbf{d_x} \cdot \mathbf{d_y})$
- 준비해둔 C_1, C_0, t_z 에 따라:
- $t_z H + (\mathbf{f}_x \cdot \mathbf{f}_y) G =? \, e^2 C_z + e C_1 + C_0$

INNER PROOF

- 완결성
- $e C_x + A_d =? \, r_x H + \mathbf{f}_x \mathbf{G}$ 에 대해:
  $e C_y + B_d =? \, s_y H + \mathbf{f}_y \mathbf{G}$
- $$\begin{aligned} &e C_x + A_d \\ &= e(rH + \mathbf{x}\mathbf{G}) + r_d H + \mathbf{d}_x \mathbf{G} \\ &= (er + r_d)H + (e\mathbf{x} + \mathbf{d}_x)\mathbf{G} \\ &= r_x H + \mathbf{f}_x \mathbf{G} \end{aligned}$$

$$\begin{aligned} &e C_y + B_d \\ &= e(sH + \mathbf{y}\mathbf{G}) + s_d H + \mathbf{d}_y \mathbf{G} \\ &= (es + s_d)H + (e\mathbf{y} + \mathbf{d}_y)\mathbf{G} \\ &= s_y H + \mathbf{f}_y \mathbf{G} \end{aligned}$$

INNER PROOF

- 완결성
- $t_z H + (\mathbf{f}_x \cdot \mathbf{f}_y) G =? \, e^2 C_z + e C_1 + C_0$ 에 대해:

$$\begin{aligned} & e^2 C_z + e C_1 + C_0 \\ & = e^2(tH + zG) + \mathrm{e}\big(t_1 H + (\mathbf{x} \cdot \mathbf{d}_y + \mathbf{y} \cdot \mathbf{d}_x) G\big) + \big(t_0 H + (\mathbf{d}_x \cdot \mathbf{d}_y) G\big) \\ & = (e^2 t + e t_1 + t_0) H + \big(e^2 z + e(\mathbf{x} \cdot \mathbf{d}_y + \mathbf{y} \cdot \mathbf{d}_x) + \mathbf{d}_x \cdot \mathbf{d}_y\big) G \\ & = (e^2 t + e t_1 + t_0) H + (e\mathbf{x} + \mathbf{d}_x) \cdot (e\mathbf{y} + \mathbf{d}_y) G \\ & = t_z H + (\mathbf{f}_x \cdot \mathbf{f}_y) G \end{aligned}$$

- 따라서 정직한 증명자는 검증자를 납득시킬 수 있음

MORE COMPACT INNER PROOF

MORE COMPACT INNER PROOF

- 더 간결한 내적 증명
  - Bootle의 논문에서
- 내적 증명 문제를 해결하기 위한 더 정교하고 현명한 방법 제시
  - 재귀(recursion)의 도입
- BulletProof에서 사용되는 아이디어와 매우 유사

MORE COMPACT INNER PROOF

- 목적은 두 당사자(증명자와 검증자) 사이의
- 데이터 통신의 양을 줄이는 것
  - 피아트-샤미르 휴리스틱(Fiat-Shamir heuristic)을 사용해 비-상호작용 형태로 바꾸면 더 간결한 증명이 됨 (생략)
- 영지식에 대한 새로운 방법이 아닌
- 연산에 이점이 있는 방법을 제시

MORE COMPACT INNER PROOF

- 하나의 벡터에 대한 고려
- 어떠한 벡터의 커미트먼트를 만들고 증명을 전송하는 데 얼마나 많은 데이터가 필요한가?

MORE COMPACT INNER PROOF

- 하나의 벡터에 대한 고려
- 10차원의 벡터 $\mathbf{a} = [a_1, a_2, \ldots, a_{10}]$ 의 예시:
  - 페더슨 커미트먼트를 사용
  - 무작위 값은 없다고 가정
  - $A = a_1 G_1 + a_2 G_2 + \ldots + a_{10} G_{10}$

MORE COMPACT INNER PROOF

- $A = a_1G_1 + a_2G_2 + \ldots + a_{10}G_{10}$
- $[a_1, a_2, \ldots, a_{10}]$을 공개하는 것으로 지식 증명이 가능
  - 정보가 공개됨
  - 10개의 스칼라를 전송해야 함
  - 타원곡선 시나리오에서는 스칼라 각자가 32바이트에 해당함
- 압축시킬 방법은 없을까?

MORE COMPACT INNER PROOF

- $A = a_1G_1 + a_2G_2 + \ldots + a_{10}G_{10}$
- 현명한 방법은 커미트먼트 A를 다른 커미트먼트 A'으로 바꾸는 것
  - 벡터의 원소 개수를 줄임으로써 형성
  - 그럼에도 불구하고 원래의 커미트먼트를 내포해야 함

MORE COMPACT INNER PROOF

- 커미트먼트 단계
- 원본 벡터를 조각으로 나눔
  - $[a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}] = [[a_1, a_2], [a_3, a_4], [a_5, a_6], [a_7, a_8], [a_9, a_{10}]]$
- 동일한 연산을 G_i 들에 대해서도 수행

MORE COMPACT INNER PROOF

- 커미트먼트 단계
- 식의 우변을 5개의 2차원 벡터로 취급:
  - $[a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}] = [[a_1, a_2], [a_3, a_4], [a_5, a_6], [a_7, a_8], [a_9, a_{10}]]$ 에서
  - $[[a_1, a_2], [a_3, a_4], [a_5, a_6], [a_7, a_8], [a_9, a_{10}]] = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5]$
- G에 관한 부분 역시:
  - $[\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4, \mathbf{G}_5]$

MORE COMPACT INNER PROOF

- 커미트먼트 단계
- 본 예제에서는 10=5X2 라서 두 개씩 묶었지만
- 개수는 소인수분해에 따라 달라질 수 있음
  - 가령 21개로 시작했으면 21=7X3이니 세 개씩 묶으면 됨

MORE COMPACT INNER PROOF

- 커미트먼트 단계
- 새로운 배치를 행렬 형태로 시각화 할 수 있음
- $$\begin{pmatrix} \mathbf{a_1 G_1} & \mathbf{a_2 G_1} & \cdots & \cdots & \mathbf{a_5 G_1} \\ \mathbf{a_1 G_2} & \mathbf{a_2 G_2} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \mathbf{a_3 G_3} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \mathbf{a_4 G_4} & \cdots \\ \mathbf{a_1 G_5} & \cdots & \cdots & \cdots & \mathbf{a_5 G_5} \end{pmatrix}$$

MORE COMPACT INNER PROOF

- 커미트먼트 단계
- $$\begin{pmatrix} \mathbf{a}_1\mathbf{G}_1 & \mathbf{a}_2\mathbf{G}_1 & \cdots & \cdots & \mathbf{a}_5\mathbf{G}_1 \\ \mathbf{a}_1\mathbf{G}_2 & \mathbf{a}_2\mathbf{G}_2 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \mathbf{a}_3\mathbf{G}_3 & \cdots & \cdots \\ \cdots & \cdots & \cdots & \mathbf{a}_4\mathbf{G}_4 & \cdots \\ \mathbf{a}_1\mathbf{G}_5 & \cdots & \cdots & \cdots & \mathbf{a}_5\mathbf{G}_5 \end{pmatrix}$$
- 행렬의 주대각성분의 합이 A임
  - $\mathbf{a}_i\mathbf{G}_i = a_{2i-1}G_{2i-1} + a_{2i}G_{2i} \;\; \forall i \in 1..5$

MORE COMPACT INNER PROOF

- 커미트먼트 단계
- 전송해야 하는 커미트먼트들의 개수는
  - 이미 알고있는 주대각성분 A를 포함해 2X5-1 = 9개
- 대각성분에 관한 수식:
  - $-4 \leq k \leq 4$ 인 정수 k에 대해 $A_k = \sum_{\max(1,1-k)}^{\min(5,5-k)} \mathbf{a}_{i+k}\mathbf{G}_i$
- 커미트먼트의 수를 9개로 줄임

MORE COMPACT INNER PROOF

$$\begin{pmatrix} \mathbf{a}_1\mathbf{G}_1 & \mathbf{a}_2\mathbf{G}_1 & \mathbf{a}_3\mathbf{G}_1 & & \mathbf{a}_5\mathbf{G}_1 \\ \mathbf{a}_1\mathbf{G}_2 & \mathbf{a}_2\mathbf{G}_2 & \mathbf{a}_3\mathbf{G}_2 & \cdots & \mathbf{a}_5\mathbf{G}_2 \\ \mathbf{a}_1\mathbf{G}_3 & \mathbf{a}_2\mathbf{G}_3 & \mathbf{a}_3\mathbf{G}_3 & & \mathbf{a}_5\mathbf{G}_3 \\ & \vdots & & \ddots & \vdots \\ \mathbf{a}_1\mathbf{G}_5 & \mathbf{a}_2\mathbf{G}_5 & \mathbf{a}_3\mathbf{G}_5 & \cdots & \mathbf{a}_5\mathbf{G}_5 \end{pmatrix}$$

$$\begin{aligned}
A_{-4} &= \mathbf{a}_1\mathbf{G}_5 \\
A_{-3} &= \mathbf{a}_1\mathbf{G}_4 + \mathbf{a}_2\mathbf{G}_5 \\
A_{-2} &= \mathbf{a}_1\mathbf{G}_3 + \mathbf{a}_2\mathbf{G}_4 + \mathbf{a}_3\mathbf{G}_5 \\
A_{-1} &= \mathbf{a}_1\mathbf{G}_2 + \mathbf{a}_2\mathbf{G}_3 + \mathbf{a}_3\mathbf{G}_4 + \mathbf{a}_4\mathbf{G}_5 \\
A_0 &= \mathbf{a}_1\mathbf{G}_1 + \mathbf{a}_2\mathbf{G}_2 + \mathbf{a}_3\mathbf{G}_3 + \mathbf{a}_4\mathbf{G}_4 + \mathbf{a}_5\mathbf{G}_5 \\
A_1 &= \mathbf{a}_2\mathbf{G}_1 + \mathbf{a}_3\mathbf{G}_2 + \mathbf{a}_4\mathbf{G}_3 + \mathbf{a}_5\mathbf{G}_4 \\
A_2 &= \mathbf{a}_3\mathbf{G}_1 + \mathbf{a}_4\mathbf{G}_2 + \mathbf{a}_5\mathbf{G}_3 \\
A_3 &= \mathbf{a}_4\mathbf{G}_1 + \mathbf{a}_5\mathbf{G}_2 \\
A_4 &= \mathbf{a}_5\mathbf{G}_1
\end{aligned}$$

MORE COMPACT INNER PROOF

- 챌린지 단계
- 검증자가 하나의 스칼라 x를 전송

MORE COMPACT INNER PROOF

- 압축 단계
- 실질적인 압축이 일어나는 지점
- 다음과 같은 새로운 2차원 벡터들을 형성:

$$\mathbf{a}' = \sum_{i=1}^{5} x^{i}\mathbf{a}_i \,, \qquad \mathbf{G}' = \sum_{i=1}^{5} x^{-i}\mathbf{G}_i$$

MORE COMPACT INNER PROOF

- 압축 단계
- $\mathbf{G}'$ 은 명시적으로 다음과 같음:

$$\begin{aligned}\mathbf{G}' &= (G_1', G_2') \\ &= [x^{-1}G_1, x^{-1}G_2] + [x^{-2}G_3, x^{-2}G_4] + [x^{-3}G_5, x^{-3}G_6] + [x^{-4}G_7, x^{-4}G_8] + [x^{-5}G_9, x^{-5}G_{10}] \\ &= [(x^{-1}G_1 + x^{-2}G_3 + x^{-3}G_5 + x^{-4}G_7 + x^{-5}G_9), (x^{-1}G_2 + x^{-2}G_4 + x^{-3}G_6 + x^{-4}G_8 + x^{-5}G_{10})]\end{aligned}$$

- $\mathbf{a}'$ 에 대해서 x의 지수가 양수인 점만 제외하면 유사:

$$\begin{aligned}&\mathbf{a}' \\ &= [(xa_1 + x^2a_3 + x^3a_5 + x^4a_7 + x^5a_9), (xa_2 + x^2a_4 + x^3a_6 + x^4a_8 + x^5a_{10})]\end{aligned}$$

MORE COMPACT INNER PROOF

- 압축 단계
- 커미트먼트를 이 새로운 좌표계로 치환:

$$A' = \mathbf{a}'\mathbf{G}'$$
$$= (xa_1 + x^2a_3 + x^3a_5 + x^4a_7 + x^5a_9)G_1' + (xa_2 + x^2a_4 + x^3a_6 + x^4a_8 + x^5a_{10})G_2'$$

  - G_1', G_2'는 2차원 벡터 $\mathbf{G}'$의 두 원소
- 위 식의 곱셈항을 전개해 행렬꼴로 나타낼 수 있음

MORE COMPACT INNER PROOF

- 압축 단계
- 곱셈항을 전개해 행렬꼴로 나타낼 수 있음

$$\begin{pmatrix} a_1G_1 + a_2G_2 & x(a_3G_1 + a_4G_2) & x^2(a_5G_1 + a_6G_2) & x^3(a_7G_1 + a_8G_2) & x^4(a_9G_1 + a_{10}G_2) \\ x^{-1}(a_1G_3 + a_2G_4) & a_3G_3 + a_4G_4 & x(a_5G_3 + a_6G_4) & x^2(a_7G_3 + a_8G_4) & x^4(a_9G_3 + a_{10}G_4) \\ \dots & \dots & a_5G_5 + a_6G_6 & \dots & \dots \\ \dots & \dots & \dots & a_7G_7 + a_8G_8 & \dots \\ x^{-4}(a_1G_9 + a_2G_{10}) & \dots & \dots & \dots & a_9G_9 + a_{10}G_{10} \end{pmatrix}$$

  - 주대각성분의 합이 $A = a_1G_1 + a_2G_2 + \ldots + a_{10}G_{10}$ 과 같음
  - 대각성분은 같은 지수를 가진 x 항들이 모여있음

MORE COMPACT INNER PROOF

- 압축 단계
- A와 A'를 연관짓기 위해 검증자는 전체 항을 검증해야 함
- 다음을 계산:
  - $A' = \sum_{k=-4}^{4} x^k A_k$
  - $A' = \sum_{k=-4}^{4} x^k A_k = \sum_{k=-4}^{4} \sum_{\max(1,1-k)}^{\min(5,5-k)} x^k \mathbf{a}_{i+k} \mathbf{G}_i$

MORE COMPACT INNER PROOF

- 재귀 단계
- 증명자와 검증자 모두 A' 및 G'을 구성할 수 있음이 핵심
  - 예시에서 10차원의 벡터 $\mathbf{a}$로 시작했으며
  - 현재 2차원의 벡터 $\mathbf{a}'$을 다루고 있음에 주목
  - 현재 벡터 $\mathbf{a}'$ 그 자체는 증명자만이 알고 있음

MORE COMPACT INNER PROOF

- 재귀 단계
- 만일 600차원의 벡터로 시작했다면?
  - 60 X 10 으로 간주해 $\mathbf{G}'$ 은 60차원을 가짐
- 재귀적으로, 60차원의 $\mathbf{G}'$ 을
  - 6 X 10 으로 간주해 압축
  - 6차원의 벡터 10개

MORE COMPACT INNER PROOF

- 재귀 단계
- $A \leftarrow A', G \leftarrow G'$ 으로
  - 커미트먼트 단계
  - 챌린지 단계
  - 압축 단계를 반복

MORE COMPACT INNER PROOF

- 공개 단계
- 여러 재귀 단계를 거쳐 마지막 단계에 다다르면
  - 증명자는 해당 벡터를 공개
  - 예시에서는 두 개의 숫자로 구성된 벡터 $\mathbf{a}'$ 를 공개
    - $[(xa_1 + x^2a_3 + x^3a_5 + x^4a_7 + x^5a_9), (xa_2 + x^2a_4 + x^3a_6 + x^4a_8 + x^5a_{10})]$

MORE COMPACT INNER PROOF

- 요약) 공시된 커미트먼트 A에 대한 상호작용의 패턴
  - 커미트먼트 단계: P가 대각성분 커미트먼트를 전송
  - 챌린지 단계: V가 x를 전송
  - 압축 단계: 양 측이 $A', \mathbf{G}'$을 계산
  - 재귀 단계: (1) $A \leftarrow A', \mathbf{G} \leftarrow \mathbf{G}'$. (2) P가 새로운 대각성분 커미트먼트를 전송 . (3) V가 x'을 전송 . (4) (1)~(3)의 반복 .
  - 공개 단계: P가 벡터 $\mathbf{a}'$을 공개

MORE COMPACT INNER PROOF

- 얼마나 (압축에) 도움이 되는가?
- 10차원의 경우
  - 주대각성분의 커미트먼트 A를 제외하고
  - 추가적인 8 = 9 - 1 개의 커미트먼트를 전송해야 함
  - 마지막에 2개의 스칼라를 공개
  - 대략 10개의 스칼라 전송과 비슷한 수준

MORE COMPACT INNER PROOF

- 얼마나 (압축에) 도움이 되는가?
- 600차원의 경우
  - 60차원의 벡터 10개로 압축, 6차원의 벡터 10개로 압축
  - 마지막에는 단지 6개의 스칼라
  - 각 압축 단계에서 2 X 10 - 1 = 19 커미트먼트가 발생
  - 총 2 X 19 + 6 - 1 = 43 개만 전송하면 됨
- 600차원을 그대로 전송하려면 600개의 스칼라를 전송해야 했음

MORE COMPACT INNER PROOF

- 지금까지는 하나의 벡터 $\mathbf{a}$에 대해서만 고려함
  - 내적 증명으로 확장
  - 일반적인 지식 증명: $z = \mathbf{a} \cdot \mathbf{b}$인 $\mathbf{a}$, $\mathbf{b}$, z에 관해 적용

MORE COMPACT INNER PROOF

- 10차원 벡터의 예로, 벡터 $\mathbf{b}$를 고려
  - 위의 방법을 그대로 적용
  - 단, $\mathbf{G}$를 다른 곡선 상의 점들을 대표하는 $\mathbf{H}$로 대체
  - 챌린지 x를 역수(inverse)인 x^{-1}로 적용
- $$B' = \sum_{k=-4}^{4} x^{-k} B_k$$

MORE COMPACT INNER PROOF

- $z = \mathbf{a} \cdot \mathbf{b}$ 에서
  - 두 벡터 모두 압축된 형태
  - 즉, $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5]$, $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5]$
  - $\mathbf{a} \cdot \mathbf{b}$를 행렬 꼴로 나타내면: $\begin{pmatrix} \mathbf{a}_1\mathbf{b}_1 & \mathbf{a}_2\mathbf{b}_1 & \dots & \dots & \mathbf{a}_5\mathbf{b}_1 \\ \mathbf{a}_1\mathbf{b}_2 & \mathbf{a}_2\mathbf{b}_2 & \dots & \dots & \dots \\ \dots & \dots & \mathbf{a}_3\mathbf{b}_3 & \dots & \dots \\ \dots & \dots & \dots & \mathbf{a}_4\mathbf{b}_4 & \dots \\ \mathbf{a}_1\mathbf{b}_5 & \dots & \dots & \dots & \mathbf{a}_5\mathbf{b}_5 \end{pmatrix}$

MORE COMPACT INNER PROOF

- $\mathbf{a}' = \Sigma_{i=1}^{5} x^i \mathbf{a_i}$ 와 마찬가지로 $\mathbf{b}'$ 은:
  - 챌린지를 x^{-1}로 적용했음에 유의
  - $$\mathbf{b}' = \sum_{i=1}^{5} x^{-i} \mathbf{b}_i$$

MORE COMPACT INNER PROOF

- z'과 z_k를 구성함:
  - $$z' = \sum_{k=-4}^{4} z_k x^k, \qquad z_k = \sum_{\max(1,1-k)}^{\min(5,5-k)} \mathbf{a}_{i+k} \cdot \mathbf{b}_i \quad (\text{for } -4 \le k \le 4)$$

MORE COMPACT INNER PROOF

- $z' = \mathbf{a}' \cdot \mathbf{b}'$ 임을 보이기는 쉬움
  - $\mathbf{a}' = [(xa_1 + x^2a_3 + x^3a_5 + x^4a_7 + x^5a_9), ((xa_2 + x^2a_4 + x^3a_6 + x^4a_8 + x^5a_{10}))]$
    $\mathbf{b}' = [(x^{-1}b_1 + x^{-2}b_3 + x^{-3}b_5 + x^{-4}b_7 + x^{-5}b_9), ((x^{-1}b_2 + x^{-2}b_4 + x^{-3}b_6 + x^{-4}b_8 + x^{-5}b_{10}))]$
  - $\mathbf{a}' \cdot \mathbf{b}'$
    $= (xa_1 + x^2a_3 + x^3a_5 + x^4a_7 + x^5a_9)(x^{-1}b_1 + x^{-2}b_3 + x^{-3}b_5 + x^{-4}b_7 + x^{-5}b_9) +$
    $(xa_2 + x^2a_4 + x^3a_6 + x^4a_8 + x^5a_{10})(x^{-1}b_2 + x^{-2}b_4 + x^{-3}b_6 + x^{-4}b_8 + x^{-5}b_{10})$
    $= a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 + a_5b_5 + x^{-4}(\dots) + \cdots + x^4(\dots)$
    $= z'$

MORE COMPACT INNER PROOF

- 요약) 벡터 각각은 물론이고 내적에 대해서도 간결한 증명 작업을 수행할 수 있음
  - 벡터 $\mathbf{G}, \mathbf{H}$는 이미 설정되었다고 가정
  - 증명자가 초기에 (A, B, z)를 공시
  - A와 B는 커미트먼트이며, $z = \mathbf{a} \cdot \mathbf{b}$를 만족
  - 벡터는 임의의 차원 $n = m_1 \times m_2 \times \ldots$

MORE COMPACT INNER PROOF

- 커미트먼트 단계: P가 인수 m_2에 대해 대각성분 커미트먼트 $(A_k, B_k, z_k \forall k \in (1-m)..(m-1))$을 전송
- 챌린지 단계: V가 x를 전송
- 압축 단계: 양 측이 $A', B', z', \mathbf{G}', \mathbf{H}'$ 을 계산
- 재귀 단계: (1) $A \leftarrow A', B \leftarrow B', z \leftarrow z', \mathbf{G} \leftarrow \mathbf{G}', \mathbf{H} \leftarrow \mathbf{H}'$
  (2) P가 다른 인수 m_3에 대해 새로운 대각성분 커미트먼트를 전송
  (3) V가 x'을 전송 . 반복 .
- 공개 단계: P가 벡터 $\mathbf{a}'$ 과 $\mathbf{b}'$ 을 공개함. 이 둘의 내적은 z' 임

BULLETPROOF

BULLETPROOFS

- 불렛프루프 논문은 기본적으로 다음을 언급:
  - (1) 내적 지식 논의의 더욱 더 간결한 형태
  - (2) 그러한 지식 논의를 사용해 간결한 범위 증명(range proof)을 만드는 방법
  - (3) 이 아이디어를 일반 산술 회로(general arithmetic circuits)로 일반화하는 방법
- (1)-(2)를 집중적으로 다뤄볼 것

BULLETPROOFS

- 더욱 더 간결한 내적 증명
  - 두 스칼라에 대한 고려로부터 시작
  - 하나의 벡터
  - 병렬적인 두 벡터
  - 내적까지 확장

BULLETPROOFS

- 두 스칼라에 대한 고려
- 무작위 값을 제외하고 생각하면
    - 두 스칼라 a와 b를 동시에 하나의 커미트먼트 $C = aG_1 + bG_2$ 로 만들 수 있음
    - 만일 이 커미트먼트를 단 하나의 스칼라만으로 개방하고자 한다면,
    - a와 b를 어떠한 방법으로든 통합할 필요가 있음

BULLETPROOFS

- Naïve한 통합은 구속 속성을 상실
  - 가령 $a + b$에 대한 커미트먼트는 $(a + \alpha)(b - \alpha)$의 커미트먼트가 될 수도 있음
  - 아무런 도움이 되지 않음

BULLETPROOFS

- 사용자별 챌린지에 따른 x를 통해
  - $(ax + b)$에 대한 커미트먼트를 생각할 수 있음
  - 꽤 괜찮아 보임
- 그러나 어떻게 검증자가 커미트먼트를 검증할 수 있는가?

BULLETPROOFS

- 필요한 것은 두 함수
  - 값 a, b, x로부터 하나의 스칼라 a'을 구하는 함수 $f(a, b, x)$
  - 기저들 G_1, G_2와 값 x로부터 새로운 기저 G'을 구하는 함수 $g(G_1, G_2, x)$
- $a' = ax + bx^{-1}$
  $G' = x^{-1}G_1 + xG_2$

BULLETPROOFS

- 두 함수로부터 검증자측은 다음과 같은 연산을 수행할 수 있음:
- $$\begin{aligned} \therefore C' &= a'G' \\ &= (ax + bx^{-1})(x^{-1}G_1 + xG_2) \\ &= aG_1 + bG_2 + x^2aG_2 + x^{-2}bG_1 \\ &= C + x^2L + x^{-2}R \end{aligned}$$
  - C는 원본 커미트먼트 $C = aG_1 + bG_2$

BULLETPROOFS

- 커미트먼트 C를 개방하기 위해
  - a, b를 대신해 a', L, R을 사용할 수 있음
- 더 많은 값을 전송해야하니 비효율적으로 보임
  - 스칼라 대신 벡터를 사용하면?

BULLETPROOFS

- 하나의 벡터에 대한 고려
- 벡터 $\mathbf{a} = [\mathbf{a}_1, \mathbf{a}_2, ... \mathbf{a}_n]$의 커미트먼트는 $\mathbf{aG}$
- 이를 반으로 나누면(cut), 그리고 다시 합치면(fold):
  - 벡터 원소의 재정렬
  - $(a_1, a_2, \dots, a_{n/2}), (a_{n/2+1}, a_{n/2+2}, \dots, a_n)$: "cut"
    $([a_1, a_{n/2+1}], [a_2, a_{n/2+2}], \dots, [a_{n/2}, a_n])$: "fold"

BULLETPROOFS

- 검증자로부터 x를 전송받으면 다음을 계산할 수 있음:
  - $\mathbf{a}' = \left(xa_1 + x^{-1}a_{n/2+1}, xa_2 + x^{-1}a_{n/2+2}, \ldots, xa_{n/2} + x^{-1}a_n\right)$
- 동일한 작업을 기저들 $\mathbf{G}$에 대해 수행:
  - $\mathbf{G}' = \left(x^{-1}G_1 + xG_{n/2+1}, x^{-1}G_2 + xG_{n/2+2}, \ldots, x^{-1}G_{n/2} + xG_n\right)$
  - x가 아닌 역수를 적용함에 유의

BULLETPROOFS

- 따라서 $\mathbf{a}'\mathbf{G}'$ 은:
  - $\mathbf{a}'\mathbf{G}'$
    $= (xa_1 + x^{-1}a_{n/2+1})(x^{-1}G_1 + xG_{n/2+1}), (xa_2 + x^{-1}a_{n/2+2})(x^{-1}G_2 + xG_{n/2+2}),$
    $\dots, (xa_{n/2} + x^{-1}a_n)(x^{-1}G_{n/2} + xG_n)$
  - $= \mathbf{aG} + x^2(a_1G_{n/2+1} + a_2G_{n/2+2} + \cdots + a_{n/2}G_n) + x^{-2}(a_{n/2+1}G_1 + a_{n/2+2}G_2 + \cdots + a_nG_{n/2})$
    $= \mathbf{aG} + x^2L + x^{-2}R$
- L과 R의 계산에는 x가 영향을 끼치지 않음

BULLETPROOFS

- 상호작용
  - 증명자가 L과 R을 계산해 전송
  - 검증자가 x를 전송
  - 양 측에서 $\mathbf{G}'$을 구성해 최종 $C' = \mathbf{a}'\mathbf{G}'$을 계산
- 통신 횟수를 n에서 $n/2 + 2$로 줄임
  - ($\mathbf{a}$에서 n) vs. ($\mathbf{a}'$에서 $n/2$, L과 R 전송에 2)

BULLETPROOFS

- 반복적으로 적용한다면
  - $2\log_2 n + 2$
    - 총 $\log_2 n$ 단계마다의 새로운 L^*과 R^*에 2
    - 마지막에 공개하는 스칼라 a와 b에 2

BULLETPROOFS

- 병렬적인 두 벡터
- 두 벡터를 하나의 커미트먼트로 만드는 방법은 자명:
  - $C = \mathbf{aG} + \mathbf{bH}$

BULLETPROOFS

- 두 벡터 모두 차원이 2의 거듭제곱수인 n이라 하면
- $\mathbf{a}$와 $\mathbf{b}$가 서로 다른 곡선 상의 점을 사용하므로, 이때의 L은:
  - $L = L_a + L_b$
    $= (a_1 G_{n/2+1} + a_2 G_{n/2+2} + \cdots + a_{n/2} G_n) + (b_1 H_{n/2+1} + b_2 H_{n/2+2} + \cdots + b_{n/2} H_n)$
- $R = R_a + R_b$ 역시 동일하게 계산 가능

BULLETPROOFS

- 증명자가 커미트먼트 $\mathbf{aG} + \mathbf{bH}$로 시작
- 반복의 각 단계마다 L과 R의 통합된 형태를 전송
- 명시적으로 새로운 커미트먼트는 다음과 같은 형태를 가짐
  - $C' = \mathbf{a}'\mathbf{G}' + \mathbf{b}'\mathbf{H}' = C + x^2(L_a + L_b) + x^{-2}(R_a + R_b)$

BULLETPROOFS

- 내적의 재도입
- 증명자가 커미트먼트 $C = zG + \mathbf{aG} + \mathbf{bH}$를 주장
  - $z = \mathbf{a} \cdot \mathbf{b}$
- 검증자가 내적이 올바른지에 대해 검증할 수 있는 구조를 생각해야 함
  - 반복의 각 단계마다 $z' = \mathbf{a}' \cdot \mathbf{b}'$을 보장해야 함

BULLETPROOFS

- 아무런 수정 없이 기존의 방법을 적용하면
- 새 커미트먼트는:
  - $C' = C + x^2 L + x^{-2} R = zG + \mathbf{aG} + \mathbf{bH} + x^2 L + x^{-2} R$
- 문제는 이제 z가 크기를 감축한 벡터 $\mathbf{a}'$과 $\mathbf{b}'$의 내적이 아니라는 것
  - 새 벡터에 대한 내적의 결과는 다음과 같아야 함
  - $\mathbf{a}' \cdot \mathbf{b}'$
    $= (xa_1 + x^{-1}a_{n/2+1}, xa_2 + x^{-1}a_{n/2+2}, \ldots, xa_{n/2} + x^{-1}a_n) \cdot (x^{-1}b_1 + xb_{n/2+1}, x^{-1}b_2 + xb_{n/2+2}, \ldots, x^{-1}b_{n/2} + xb_n)$

BULLETPROOFS

- 새 벡터에 대한 내적의 결과는 다음과 같아야 함

  - $$\begin{aligned}&\mathbf{a}' \cdot \mathbf{b}' \\ &= \left(xa_1 + x^{-1}a_{n/2+1}, xa_2 + x^{-1}a_{n/2+2}, \ldots, xa_{n/2} + x^{-1}a_n\right) \\ &\quad \cdot \left(x^{-1}b_1 + xb_{n/2+1}, x^{-1}b_2 + xb_{n/2+2}, \ldots, x^{-1}b_{n/2} + xb_n\right) \\ &= a_1b_1 + a_2b_2 + \cdots + a_nb_n \\ &\quad + x^2\left(a_1b_{n/2+1} + a_2b_{n/2+2} + \cdots + a_{n/2}b_n\right) \\ &\quad + x^{-2}\left(a_{n/2+1}b_1 + a_{n/2+2}b_2 + \cdots + a_nb_{n/2}\right)\end{aligned}$$

- 마치 C'이 C를 포함하듯, $\mathbf{a}'\cdot\mathbf{b}'$이 z를 포함하고 있음

BULLETPROOFS

- 마치 C'이 C를 포함하듯, $\mathbf{a}' \cdot \mathbf{b}'$이 z를 포함하고 있음
- L과 R을 약간 변형해 나머지 항을 포함하도록 함
  - $$L = L_a + L_b + \left(a_1 b_{n/2+1} + a_2 b_{n/2+2} + \cdots + a_{n/2} b_n\right) G$$
    $$R = R_a + R_b + \left(a_{n/2+1} b_1 + a_{n/2+2} b_2 + \cdots + a_n b_{n/2}\right) G$$
- 이제 커미트먼트를 잘 감축할 수 있음:
  - $C' = z'G + \mathbf{a}'\mathbf{G} + \mathbf{b}'\mathbf{H} = C + x^2 L + x^{-2} R, \qquad z' = \mathbf{a}' \cdot \mathbf{b}'$

BULLETPROOFS

- 본 절차 역시 반복적으로 적용 가능
  - 길이 n인 벡터로부터 시작
  - $C = zG + \mathbf{aG} + \mathbf{bH}$ 를 L 및 R 과 함께 검증자에게 전송
  - 검증자는 챌린지 x를 전송
  - 양 측은 C' 을 계산
- 마지막 단계까지 반복

BULLETPROOFS

- 마지막 단계
  - 스칼라 a, b를 공개
  - 검증자는 $C^* = abG + aG_1 + bH_1$ 으로 검증

BULLETPROOFS

- 주어진 z가 커밋된 두 벡터의 내적으로 변경되려면
  - 검증자가 초기 챌린지 x를 제공해야 함
  - 초기 $C = z(xG) + \mathbf{aG} + \mathbf{bH}$

RANGE PROOF

RANGE PROOF

- 지금까지 벡터의 내적에 대한 지식 증명을 논의
    - 적용에 대해서는 고려한 바 없음
- 어떻게 불렛프루프가
    - 다항식들을 이용해
    - 내적의 형태로
    - 커밋된 숫자에 대한 범위 증명을 수행하는가?

RANGE PROOF

- 숫자/값 v에 대한 커미트먼트로 시작
  - $V = \gamma H + vG$
  - 값 v에 대해
  - 무작위 값 γ를 사용한
  - 기본적인 은닉 페더슨 커미트먼트
- 원하는 것은 v가 특정 범위 $v \in 2^n \,..\, 2^{n-1}$ 에 있다는 증명

RANGE PROOF

- 범위 증명을 어디에 쓸 수 있을까?
- Maxwell이 제안한 비트코인의 기밀 트랜잭션
  - 값에 대한 페더슨 커미트먼트를 만들고
  - 커미트먼트의 동형 특성을 이용해 잔액에 대해 보증

RANGE PROOF

- 이루고자 하는 목표
  - (1) 가능한 증명을 간결하게 만들어, 통신에 필요한 데이터 양을 줄임
  - (2) 그러면서도 동시에 여러 조건들에 대한 증명을 제공
- (1)을 위해, 내적을 포함하는 외부 다항식을 구성
- (2)를 위해, 다항식을 평가하기 위한 챌린지를 사용

RANGE PROOF

- 추가적인 표기법
- $\mathbf{k^n} = [1, k, k^2, \ldots, k^n]$
  - 정수의 지수승의 벡터
  - 모든 정수는 $\mod p$를 취함

RANGE PROOF

- 추가적인 표기법
- $\mathbf{a} \circ \mathbf{b} = [a_1 b_1, a_2 b_2, \ldots, a_n b_n]$
  - 두 벡터의 아다마르 곱(Hadamard product)
  - 아다마르 곱의 결과는 벡터임에 유의

RANGE PROOF

- 단계 1
- 값 v를 비트(bit) 표현으로 인코딩
  - $\mathbf{a_L}$은 $\mathbf{a_L} \cdot 2^n = v$인 벡터
  - 쉽게 말해 v의 이진법 표현

RANGE PROOF

- 단계 2
- v가 범위 안에 있다는 것을 증명하기 위해
- $\mathbf{a_L}$의 각 원소는 0 또는 1이어야만 한다는 조건을 더함
  - 이를 위해 보수 벡터(complementary vector) $\mathbf{a_R} = \mathbf{a_L} - 1^n$ 을 구성
  - $\mathbf{a_L} \circ \mathbf{a_R} = 0^n$ 이어야 함

RANGE PROOF

- 단계 3
- 주어진 문제는 세 관점에서 바라볼 수 있음
- 그 중 두 관점을 먼저 살펴볼 것
    - (1) 단계 2에서 언급한 조건들 (1만큼 차이나는 두 벡터, 아다마르 곱이 0) 각각에 대해 챌린지 y를 사용해 내적을 구성
    - (2) 이들 조건에 v에 대한 조건(이진법 표현)을 더한 세 조건을 챌린지 z에 대한 방정식의 형태로 표현

RANGE PROOF

- 단계 3
- 방정식의 형태:
  - $z^2\mathbf{a}_L \cdot \mathbf{2}^n + z(\mathbf{a}_L - \mathbf{1}^n - \mathbf{a}_R) \cdot \mathbf{y}^n + (\mathbf{a}_L \circ \mathbf{a}_R) \cdot \mathbf{y}^n = z^2 v$

RANGE PROOF

- 단계 3
- 방정식의 형태:
  - $z^2\boxed{\mathbf{a}_L \cdot \mathbf{2}^n} + z(\mathbf{a}_L - \mathbf{1}^n - \mathbf{a}_R) \cdot \mathbf{y}^n + (\mathbf{a}_L \circ \mathbf{a}_R) \cdot \mathbf{y}^n = z^2\boxed{v}$
- $\mathbf{a_L} \cdot \mathbf{2^n}\ (= v)$
  - 벡터 v에 대한 조건을 의미

RANGE PROOF

- 단계 3
- 방정식의 형태:
  - $z^2\mathbf{a}_L\cdot\mathbf{2}^n + z\boxed{(\mathbf{a}_L-\mathbf{1}^n-\mathbf{a}_R)}\cdot\mathbf{y}^n + (\mathbf{a}_L\circ\mathbf{a}_R)\cdot\mathbf{y}^n = z^2v$
- $\mathbf{a_L}-\mathbf{1^n}-\mathbf{a_R}\ (=0)$
  - 벡터 $\mathbf{a_L}$과 $\mathbf{a_R}$의 차이가 1이라는 조건을 의미

RANGE PROOF

- 단계 3
- 방정식의 형태:
  - $z^2\mathbf{a}_L \cdot \mathbf{2}^n + z(\mathbf{a}_L - \mathbf{1}^n - \mathbf{a}_R) \cdot \mathbf{y}^n + (\boxed{\mathbf{a}_L \circ \mathbf{a}_R}) \cdot \mathbf{y}^n = z^2 v$
- $\mathbf{a}_\mathrm{L} \circ \mathbf{a}_\mathrm{R}\ (=0)$
  - 벡터 $\mathbf{a}_\mathrm{L}$과 $\mathbf{a}_\mathrm{R}$의 아다마르 곱이 0이라는 조건을 의미

RANGE PROOF

- 단계 3
- 방정식의 형태:
  - $z^2 \mathbf{a}_L \cdot \mathbf{2}^n + z(\mathbf{a}_L - \mathbf{1}^n - \mathbf{a}_R) \cdot \boxed{\mathbf{y}^n} + (\mathbf{a}_L \circ \mathbf{a}_R) \cdot \boxed{\mathbf{y}^n} = z^2 v$
- $\mathbf{y}^{\mathbf{n}}$
  - 벡터가 n차원임을 검사하기 위한
  - n차 다항식을 형성하기 위해 필요

RANGE PROOF

- 단계 3
- 방정식의 형태:
  - $\boxed{z^2}\mathbf{a}_L \cdot \mathbf{2}^n + z(\mathbf{a}_L - \mathbf{1}^n - \mathbf{a}_R) \cdot \mathbf{y}^n + (\mathbf{a}_L \circ \mathbf{a}_R) \cdot \mathbf{y}^n = \boxed{z^2}v$
- z^2
  - 조건 세 개를 판단하기 위한
  - 2차 방정식을 구성하기 위해 필요

RANGE PROOF

- 단계 4
- 수학적 절차를 통해 도출한 방정식을 하나의 내적의 형태로 변환
  - 이러한 일종의 인수분해는 일부 항을 남길 수 있음
  - 중요한 것은 $\mathbf{a}_L$, $\mathbf{a}_R$ 항을 내적에 포함하고자 하는 것
  - 이 $\mathbf{a}_L$, $\mathbf{a}_R$은 검증자에게 공유되지 않음에 유의

RANGE PROOF

- 단계 4
- 수학적 절차를 통해 도출한 방정식을 하나의 내적의 형태로 변환:
  - $(\mathbf{a}_L - z\mathbf{1}^n) \cdot (\mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n) + z^2\mathbf{2}^n) = z^2 v + \delta(y, z)$
- δ는 공시된 챌린지들의 함수
  - 댕글링(dangling) 항
  - $(z - z^2)(\mathbf{1}^n \cdot \mathbf{y}^n) - z^3(\mathbf{1}^n \cdot \mathbf{2}^n)$

RANGE PROOF

- 단계 5
- 비로소 단계 3에서 언급 했던 세 번째 관점에 대한 얘기
  - 아직 영지식이 아니라는 점!

RANGE PROOF

- 단계 5
- 이를 해결하기 위해 벡터 $\mathbf{a_L}$, $\mathbf{a_R}$를 숨길 필요가 있음
    - 증명자로부터 생성된 두 무작위 벡터 $\mathbf{s_L}$, $\mathbf{s_R}$를 도입
    - 이들 벡터에 대한 숨겨진 벡터 페더슨 커미트먼트를 전송:

$$A = \alpha H + \mathbf{a}_L \mathbf{G} + \mathbf{a}_R \mathbf{H}$$
$$S = \rho H + \mathbf{s}_L \mathbf{G} + \mathbf{s}_R \mathbf{H}$$

RANGE PROOF

- 단계 6
- 내적에 관련된 식은 숨김 효과를 위해
  - 새로운 챌린지 x를 이용해
  - 새로운 다항식으로 구성될 필요가 있음
- 마치 내적 증명에서 비밀 벡터 $\mathbf{x}$를 전송하기 위해
  - 논스 벡터 $\mathbf{d}$에 대해 $e\mathbf{x} + \mathbf{d}$를 구성한 것과 유사
- 이렇게 하면 $\mathbf{a_L}, \mathbf{a_R}$을 숨기면서도 논의 증명이 가능

RANGE PROOF

- 단계 6
- 전송해야 하는 데이터는 $\mathbf{a_L}$, $\mathbf{a_R}$가 아닌, 다음과 같이 정의되는 $\mathbf{l}$ 과 $\mathbf{r}$
  - $$\mathbf{l}(x) = \left((\mathbf{a}_L + \mathbf{s}_L x) - z\mathbf{1}^n\right) = (\mathbf{a}_L - z\mathbf{1}^n) + \mathbf{s}_L x$$
    $$\mathbf{r}(x) = \mathbf{y}^n \circ \left((\mathbf{a}_R + \mathbf{s}_R x) + z\mathbf{1}^n\right) + z^2\mathbf{2}^n = \mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n + \mathbf{s}_R x) + z^2\mathbf{2}^n$$
    $$t(x) = \mathbf{l}(x) \cdot \mathbf{r}(x) = t_0 + t_1 x + t_2 x^2$$
- 기존의 식과 비교:
  - $$(\mathbf{a}_L - z\mathbf{1}^n) \cdot (\mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n) + z^2\mathbf{2}^n) = z^2 v + \delta(y, z)$$

RANGE PROOF

- 단계 6
- 전송해야 하는 데이터는 $\mathbf{a}_\mathrm{L}$, $\mathbf{a}_\mathrm{R}$가 아닌, 다음과 같이 정의되는 $\mathbf{l}$ 과 $\mathbf{r}$
  - $$\mathbf{l}(x) = \left((\mathbf{a}_L + \mathbf{s}_L x) - z\mathbf{1}^n\right) = \boxed{(\mathbf{a}_L - z\mathbf{1}^n) + \mathbf{s}_L x}$$
    $$\mathbf{r}(x) = \mathbf{y}^n \circ \left((\mathbf{a}_R + \mathbf{s}_R x) + z\mathbf{1}^n\right) + z^2\mathbf{2}^n = \mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n + \mathbf{s}_R x) + z^2\mathbf{2}^n$$
    $$t(x) = \mathbf{l}(x) \cdot \mathbf{r}(x) = t_0 + t_1 x + t_2 x^2$$
- 기존의 식과 비교:
  - $$\boxed{(\mathbf{a}_L - z\mathbf{1}^n)} \cdot (\mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n) + z^2\mathbf{2}^n) = z^2 v + \delta(y, z)$$

RANGE PROOF

- 단계 6

- 전송해야 하는 데이터는 $\mathbf{a_L}$, $\mathbf{a_R}$가 아닌, 다음과 같이 정의되는 $\mathbf{l}$ 과 $\mathbf{r}$
  - $$\mathbf{l}(x) = \left((\mathbf{a}_L + \mathbf{s}_L x) - z\mathbf{1}^n\right) = (\mathbf{a}_L - z\mathbf{1}^n) + \mathbf{s}_L x$$
    $$\mathbf{r}(x) = \mathbf{y}^n \circ \left((\mathbf{a}_R + \mathbf{s}_R x) + z\mathbf{1}^n\right) + z^2\mathbf{2}^n = \boxed{\mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n + \mathbf{s}_R x) + z^2\mathbf{2}^n}$$
    $$t(x) = \mathbf{l}(x) \cdot \mathbf{r}(x) = t_0 + t_1 x + t_2 x^2$$

- 기존의 식과 비교:
  - $$(\mathbf{a}_L - z\mathbf{1}^n) \cdot \boxed{(\mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n) + z^2\mathbf{2}^n)} = z^2 v + \delta(y, z)$$

RANGE PROOF

- 단계 6

- 전송해야 하는 데이터는 $\mathbf{a_L}$, $\mathbf{a_R}$가 아닌, 다음과 같이 정의되는 $\mathbf{l}$ 과 $\mathbf{r}$
  - $$\mathbf{l}(x) = \left((\mathbf{a}_L + \mathbf{s}_L x) - z\mathbf{1}^n\right) = (\mathbf{a}_L - z\mathbf{1}^n) + \mathbf{s}_L x$$
    $$\mathbf{r}(x) = \mathbf{y}^n \circ \left((\mathbf{a}_R + \mathbf{s}_R x) + z\mathbf{1}^n\right) + z^2\mathbf{2}^n = \mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n + \mathbf{s}_R x) + z^2\mathbf{2}^n$$
    $$t(x) = \boxed{\mathbf{l}(x) \cdot \mathbf{r}(x)} = t_0 + t_1 x + t_2 x^2$$

- 기존의 식과 비교:
  - $$\boxed{(\mathbf{a}_L - z\mathbf{1}^n) \cdot (\mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n) + z^2\mathbf{2}^n)} = z^2 v + \delta(y, z)$$

RANGE PROOF

- 단계 6
- 전송해야 하는 데이터는 $\mathbf{a_L}$, $\mathbf{a_R}$가 아닌, 다음과 같이 정의되는 $\mathbf{l}$ 과 $\mathbf{r}$
  - $\mathbf{l}(x) = \left((\mathbf{a}_L + \mathbf{s}_L x) - z\mathbf{1}^n\right) = (\mathbf{a}_L - z\mathbf{1}^n) + \mathbf{s}_L x$
    $\mathbf{r}(x) = \mathbf{y}^n \circ \left((\mathbf{a}_R + \mathbf{s}_R x) + z\mathbf{1}^n\right) + z^2\mathbf{2}^n = \mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n + \mathbf{s}_R x) + z^2\mathbf{2}^n$
    $t(x) = \mathbf{l}(x) \cdot \mathbf{r}(x) = t_0 + t_1 x + t_2 x^2$
  - 내적의 두 항에 x^1의 계수를 가진 $\mathbf{s_L}$, $\mathbf{s_R}$이 포함됨에 주목
  - 두 선형 방정식 $\mathbf{l}$ 과 $\mathbf{r}$이 하나의 x에 대한 2차 방정식으로 결합됨에 주목
  - t는 내적의 결과이기 때문에 스칼라 값

RANGE PROOF

- 이 6 단계가 증명자와 검증자의 상호작용을 전부 포괄하지는 않았으나
    - 핵심적인 부분임

RANGE PROOF

- 요약) 범위 증명을 내적의 형태로 표현하는 방법
  - 값의 실질적 데이터 표현인 $\mathbf{a_L}, \mathbf{a_R}$을 가지고 있음
  - 첫 번째 챌린지인 y를 사용해 벡터의 비트 수를 고정
  - 챌린지 z를 사용해 세 조건(모두 내적의 형태)을 하나의 방정식으로 강제
  - 세 항을 하나의 내적으로 표현하고, 공시된 데이터들로만 구성된 댕글링 항을 허용
  - 마지막으로 무작위 벡터 $\mathbf{s_L}, \mathbf{s_R}$을 원본 벡터와 결합하고
  - 세 번째 챌린지 x를 사용해 구성한 내적을 상수항으로 포함하는 방정식을 만듦

RANGE PROOF

- 이 내적을 검증하는 것으로 v가 범위 안에 있음을 검증할 수 있음

RANGE PROOF

- 다항식 $t(x)$를 구성하고 난 후
  - 내적 증명을 제외하고,
  - 증명자가 값이 범위 안에 있음을 검증자에게 납득시키기 위해
  - 필요한 것이 무엇인가?
- 기본적으로 증명자는 방정식 $t(x)$를 정직하게 구성했음을
  - 검증자에게 납득시켜야 함

RANGE PROOF

- 이를 위해
  - 증명자는 챌린지 x를 받기 전에
  - 계수 t_1과 t_2에 대한 페더슨 커미트먼트들
  - T_1과 T_2를 전송
- $T_1 = \tau_1 H + t_1 G$
  $T_2 = \tau_2 H + t_2 G$

RANGE PROOF

- 챌린지 x를 받은 후에
  - x-다항식의 커밋된 형태로 τ_1과 τ_2를 결합한 커미트먼트
  - $\tau_x = \tau_2 x^2 + \tau_1 x + z^2 \gamma$ 를 전송
- 또한, 내적 $t(x) = \mathbf{l}(\mathbf{x}) \cdot \mathbf{r}(\mathbf{x}) = \hat{t}$ 를 전송
- 이미 전송한 커미트먼트 A, S에 대한 무작위 값에 연관된 값
  - $\mu = \alpha + \rho x$ 를 전송

RANGE PROOF

- 첫 번째 검증
- 값에 대한 페더슨 커미트먼트 $V = \gamma H + vG$ 에 대한 검증은 결국
  - $t(x)$의 상수항이 다음과 같음을 보이는 것:
  - $\hat{t}G + \tau_x H =? z^2 V + \delta G + xT_1 + x^2 T_2$

RANGE PROOF

- 첫 번째 검증
- 방정식의 양 변을 페더슨화된 형태로 검증할 수 있음
  - $\hat{t}G + \tau_x H =? z^2V + \delta G + xT_1 + x^2T_2$
- G에 대해: $\mathbf{l}(x) \cdot \mathbf{r}(x) = z^2V + \delta + xt_1 + x^2t_2$
- H에 대해: $\tau_x = \tau_2 x^2 + \tau_1 x + z^2\gamma$

RANGE PROOF

- 두 번째 검증
- 검증자는 커미트먼트 A, S가 올바른지에 대한 검증도 수행해야 함
  - 달리 말해, 검증자는 증명자가 챌린지 x, y, z를 수신하기 전에 만든 $\mathbf{a_L}, \mathbf{a_R}$에 대한
  - 숨겨진 커미트먼트를 검증해야 함

RANGE PROOF

- 두 번째 검증
- 다음과 같은 요구사항이 있음:
  - $$\mathbf{H}' = y^{-n}\mathbf{H}$$
    $$P = A + xS - zG + (z\mathbf{y}^n + z^2\mathbf{2}^n)\mathbf{H}'$$
    $$P =? \, \mu H + \mathbf{l}\mathbf{G} + \mathbf{r}\mathbf{H}'$$
- $\mathbf{H}'$의 도입은 $\mathbf{y}^{\mathbf{n}}$과의 아다마르 곱을 사용하는 $\mathbf{r}(\mathbf{x})$의 정의와,
- A의 $\mathbf{a}_{\mathbf{R}}$에 대한 커미트먼트를 연관짓기 위한 대수적 기법

RANGE PROOF

- 세 번째 검증
- 마지막으로, 검증자는 내적이 정확한지를 검증해야 함
    - $\hat{t} =? \mathbf{l} \cdot \mathbf{r}$

RANGE PROOF

- 요약) 지금까지 언급한 증명자와 검증자 사이의 상호작용
- $V, n, \mathbf{G}, \mathbf{H}, G, H \quad \text{(shared in advance)}$

$$
\begin{array}{ccc}
\underline{\text{Prover}} & & \underline{\text{Verifier}} \\
A, S & \rightarrow & \\
 & \leftarrow & y, z \\
T_1, T_2 & \rightarrow & \\
 & \leftarrow & x \\
\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r} & \rightarrow & \\
 & & \text{verify}
\end{array}
$$

  - 검증(verify)는 앞서 언급한 세 가지 검증을 수행함을 의미

RANGE PROOF

- 요약) 지금까지 언급한 증명자와 검증자 사이의 상호작용
- 이 절차는 본질적으로 영지식에 해당
  - 그러나 간결하지는 않음
- 어떻게 간결함을 줄 것인가?

COMPACT $O(\log n)$ INNER PRODUCT PROOF

RANGE PROOF

- 불렛프루프에서 제공하는 간결한 내적 증명
  - 오직 모든 것을 통합한 커미트먼트 C만을 공시하고 내적에 관해 주장함
- 마지막 검증 $\hat{t} = ?\mathbf{l} \cdot \mathbf{r}$과 커미트먼트 P에 대한 검증을
  - 하나의 내적 증명으로 변환하는 방법

RANGE PROOF

- 사전에 기저들 $G, H, \mathbf{G}, \mathbf{H}$과 차원 n은 공유되었다고 가정
- 이제 재귀 단계에서 기저들을 치환하는 것만 필요
- 공개적으로(public):
  - $C \leftarrow P - \mu H$
    $\mathbf{H} \leftarrow \mathbf{H}'(= y^{-n}\mathbf{H})$
    $\mathbf{G} \leftarrow \mathbf{G}$
    $z \leftarrow \hat{t}$

RANGE PROOF

- 사전에 기저들 $G, H, \mathbf{G}, \mathbf{H}$과 차원 n은 공유되었다고 가정
- 이제 재귀 단계에서 기저들을 치환하는 것만 필요
- 개인적으로(private):
    - $\mathbf{a} \leftarrow \mathbf{l}(x)$
      $\mathbf{b} \leftarrow \mathbf{r}(x)$

RANGE PROOF

- 통신에 많은 비중을 차지하는 벡터 $\mathbf{l}, \mathbf{r}$을 전송하는 대신
  - 증명자는 C로서 $P - \mu H$를 전송
- $\hat{t}$ 에 대한 전송에는 변화 없음

RANGE PROOF

- 확장성
  - 오직 통신하는 데이터의 양에만 초점을 둠
  - 연산 비용 또한 중요하지만, 여기서는 무시함
  - (조만간 언급할) 비-상호 증명을 사용해, 챌린지 값 x, y, z 대신 증명자 혼자서 도출한 계산된 값을 사용한다고 가정

RANGE PROOF

- 증명자는
  - 곡선 상의 점 A, S, T_1, T_2와
  - 스칼라 $\tau_x, \mu, \hat{t}$
  - 재귀 단계 별 L, R과
  - 마지막 두 개의 스칼라 a, b를 전송해야 함
- 따라서 공시된 증명의 총 크기는
  - (곡선 상의 점의 크기) X $(4 + 2\log_2 n)$ + (스칼라의 크기) X 5

RANGE PROOF

- 곡선 상의 점의 크기와 스칼라는 각각 32 바이트로 인코딩할 수 있음
  - n이 범위의 비트 수라 할 때,
  - 대략 $32 \times (9 + 2\log_2 n)$ 크기를 가짐
- 가령 32 비트 범위에서는 608 바이트 정도를
- 64 비트 범위에서는 672 바이트 정도가 요구됨

RANGE PROOF

- 이는 엄청난 수준의 절감
- 초기 기밀 트랜잭션의 구현체에 사용된 보로미안 고리(Borromean ring) 서명에 기반한 범위 증명이
  - 32 비트 범위에 2,500 바이트를 요구한 것과는 대조적
  - 보로미안 고리 서명은 $\log n$ 스케일링(scailing)을 가지지 않으므로
  - 64 비트에 대해서는 더욱 더 극적일 것

RANGE PROOF

- 통합(Aggregation) 아이디어
  - 내적 증명이 $O(\log n)$ 스케일링을 가지므로
  - 여러 값을 하나로 연결(concatenate)하는 것이 유용하다는 것

RANGE PROOF

- 가령, 비트코인 트랜잭션에서 여러 출력값을 연결할 수 있음
- 여러 값 v_j에 대해
  - 각각의 커미트먼트가 V_j이고
  - 범위 안에 들어있음을 증명해야 함

RANGE PROOF

- 이 값들을 하나로 연결하면
  - 가령 첫 번째 값이 (십진법으로) 10이고 두 번째가 3
  - $n = 4$라면 $v_1 = [1,0,1,0]$이고 $v_2 = [0,0,1,1]$
  - 이들의 연결은 단지 8비트 표현인 $[10100011]$ 임

RANGE PROOF

- 두 개의 값 v_1과 v_2에 대해 64 비트 범위 증명을 하는 상황
  - 단순하게 두 값 각각에 대해 증명을 수행하면
  - $2 \times (32 \times (9 + 2\log_2 64)) = 1344$ 바이트가 요구
- 대신 값을 연결해 사용하면
  - $32 \times (9 + 2\log_2 128) = 736$ 바이트만 요구

RANGE PROOF

- 이 아이디어로부터 여러 범위 증명을 로그 스케일로 처리할 수 있음
- 사실은 값 δ, τ_x와 커미트먼트 P가
  - 여러 값의 존재를 반영하기 위해 업데이트되어야 함
  - 자세한 구현은 생략

NON-INTERACTIVE PROOFS

NON-INTERACTIVE PROOFS

- 지금까지의 프로토콜들은 검증자와 증명자 사이에 여러 상호작용이 있었음
  - 이는 실제 적용에서는 어려운 일
- 가령 비트코인에서 매 트랜잭션마다
  - 수 차례의 상호작용을 수행해야 한다고 생각: 매우 큰 오버헤드

NON-INTERACTIVE PROOFS

- 이는 이러한 종류의 상호작용을 포함한
  - 암호학적 프로토콜들에게서 널리 사용되는 기본적인 기술인
  - 피아트-샤미르(Fiat-Shamir) 발견법(發見法, heuristic)을 사용해 해결할 수 있음
- 이 발견법은 랜덤 오라클 모델(Random Oracle Model, ROM)에 기반
  - 이 프로토콜이 안전하다는 결론에 이르기 위해 추가적인 가정이 필요하다는 것

NON-INTERACTIVE PROOFS

- 기본적으로 랜덤 오라클은
  - 입력에 대해 결정론적으로 예상할 수 없는
  - 무작위 값을 출력하는 블랙박스로 간주
  - 같은 입력에 대해서는 항상 같은 출력을 얻음

NON-INTERACTIVE PROOFS

- 피아트-샤미르 발견법에서의 입력은
  - 특히 해당 지점까지의 상호작용의 기록임에 유의
- 가령, 슈노르 서명에서 챌린지 e는
  - (서명할 메시지, 논스 점 R, 공개키 P)의 해시로 대체

BULLETPROOFS

FROM ZERO (KNOWLEDGE)
TO BULLETPROOFS