BULLETPROOFS

FROM ZERO (KNOWLEDGE) TO BULLETPROOFS

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:

- $$\therefore C' = a'G'$$
  $$= (ax + bx^{-1})(x^{-1}G_1 + xG_2)$$
  $$= aG_1 + bG_2 + x^2aG_2 + x^{-2}bG_1$$
  $$= C + x^2L + x^{-2}R$$

  - 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 $\mathbf{a} = [\mathbf{a_1}, \mathbf{a_2}, \ldots \mathbf{a_n}]$의 커미트먼트는 $\mathbf{aG}$
- 이를 반으로 나누면(cut), 그리고 다시 합치면(fold):
  - 벡터 원소의 재정렬
  - $(a_1, a_2, \ldots, a_{n/2}), (a_{n/2+1}, a_{n/2+2}, \ldots, a_n)$: "cut"
    $([a_1, a_{n/2+1}], [a_2, a_{n/2+2}], \ldots, [a_{n/2}, a_n])$: "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:
  - $\mathbf{a}'\mathbf{G}'$
    $= (xa_1 + x^{-1}a_{n/2+1})(x^{-1}G_1 + xG_{n/2+1}), (xa_2 + x^{-1}a_{n/2+2})(x^{-1}G_2 + xG_{n/2+2}),$
    $\ldots, (xa_{n/2} + x^{-1}a_n)(x^{-1}G_{n/2} + xG_n)$
  - $= \mathbf{aG} + x^2(a_1G_{n/2+1} + a_2G_{n/2+2} + \cdots + a_{n/2}G_n) + x^{-2}(a_{n/2+1}G_1 + a_{n/2+2}G_2 + \cdots + a_nG_{n/2})$
    $= \mathbf{aG} + x^2L + x^{-2}R$
- 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 곱선 상의 점을 사용하므로, 이때의 L은:
  - $$L = L_a + L_b$$
    $$= \left(a_1 G_{n/2+1} + a_2 G_{n/2+2} + \cdots + a_{n/2} G_n\right) + \left(b_1 H_{n/2+1} + b_2 H_{n/2+2} + \cdots + b_{n/2} H_n\right)$$
- $R = R_a + R_b$ 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:
  - $C' = C + x^2L + x^{-2}R = zG + \mathbf{aG} + \mathbf{bH} + x^2L + x^{-2}R$
- 문제는 이제 z가 크기를 감축한 벡터 $\mathbf{a}'$과 $\mathbf{b}'$의 내적이 아니라는 것
  - 새 벡터에 대한 내적의 결과는 다음과 같아야 함
  - $$\mathbf{a}' \cdot \mathbf{b}' = \left(xa_1 + x^{-1}a_{n/2+1}, xa_2 + x^{-1}a_{n/2+2}, \ldots, xa_{n/2} + x^{-1}a_n\right) \cdot \left(x^{-1}b_1 + xb_{n/2+1}, x^{-1}b_2 + xb_{n/2+2}, \ldots, x^{-1}b_{n/2} + xb_n\right)$$

BULLETPROOFS

- 새 벡터에 대한 내적의 결과는 다음과 같아야 함
  - $$\begin{aligned}
    &\mathbf{a}' \cdot \mathbf{b}' \\
    &= \left(x a_1 + x^{-1} a_{n/2+1}, x a_2 + x^{-1} a_{n/2+2}, \ldots, x a_{n/2} + x^{-1} a_n\right) \\
    &\quad \cdot \left(x^{-1} b_1 + x b_{n/2+1}, x^{-1} b_2 + x b_{n/2+2}, \ldots, x^{-1} b_{n/2} + x b_n\right) \\
    &= a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \\
    &\quad + x^2 \left(a_1 b_{n/2+1} + a_2 b_{n/2+2} + \cdots + a_{n/2} b_n\right) \\
    &\quad + x^{-2} \left(a_{n/2+1} b_1 + a_{n/2+2} b_2 + \cdots + a_n b_{n/2}\right)
    \end{aligned}$$
- 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
  - $L = L_a + L_b + (a_1 b_{n/2+1} + a_2 b_{n/2+2} + \cdots + a_{n/2} b_n) G$
    $R = R_a + R_b + (a_{n/2+1} b_1 + a_{n/2+2} b_2 + \cdots + a_n b_{n/2}) G$
- 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 y에 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:
  - $z^2 \mathbf{a}_L \cdot \mathbf{2}^n + z(\mathbf{a}_L - \mathbf{1}^n - \mathbf{a}_R) \cdot \mathbf{y}^n + (\mathbf{a}_L \circ \mathbf{a}_R) \cdot \mathbf{y}^n = z^2 v$
- $\mathbf{a_L} \cdot \mathbf{2^n}\ (= v)$
  - 벡터 v에 대한 조건을 의미

RANGE PROOF

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

RANGE PROOF

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

RANGE PROOF

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

RANGE PROOF

- 단계 3
- 방정식의 형태:
  - $z^2 \mathbf{a}_L \cdot \mathbf{2}^n + z(\mathbf{a}_L - \mathbf{1}^n - \mathbf{a}_R) \cdot \mathbf{y}^n + (\mathbf{a}_L \circ \mathbf{a}_R) \cdot \mathbf{y}^n = z^2 v$
- 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 $\mathbf{a_L}, \mathbf{a_R}$가 아닌, 다음과 같이 정의되는 $\mathbf{l}$ 과 $\mathbf{r}$
  - $\mathbf{l}(x) = \left((\mathbf{a}_L + \mathbf{s}_L x) - z\mathbf{1}^n\right) = \boxed{(\mathbf{a}_L - z\mathbf{1}^n) + \mathbf{s}_L x}$
    $\mathbf{r}(x) = \mathbf{y}^n \circ \left((\mathbf{a}_R + \mathbf{s}_R x) + z\mathbf{1}^n\right) + z^2\mathbf{2}^n = \mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n + \mathbf{s}_R x) + z^2\mathbf{2}^n$
    $t(x) = \mathbf{l}(x) \cdot \mathbf{r}(x) = t_0 + t_1 x + t_2 x^2$
- 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 $\mathbf{r}$
  - $\mathbf{l}(x) = \left((\mathbf{a}_L + \mathbf{s}_L x) - z\mathbf{1}^n\right) = (\mathbf{a}_L - z\mathbf{1}^n) + \mathbf{s}_L x$
    $\mathbf{r}(x) = \mathbf{y}^n \circ \left((\mathbf{a}_R + \mathbf{s}_R x) + z\mathbf{1}^n\right) + z^2\mathbf{2}^n = \mathbf{y}^n \circ (\mathbf{a}_R + z\mathbf{1}^n + \mathbf{s}_R x) + z^2\mathbf{2}^n$
    $t(x) = \mathbf{l}(x) \cdot \mathbf{r}(x) = t_0 + t_1 x + t_2 x^2$
- 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 $\mathbf{a_L}, \mathbf{a_R}$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 $\mathbf{a_L}, \mathbf{a_R}$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
  - $\hat{t}G + \tau_x H =? z^2 V + \delta G + xT_1 + x^2 T_2$
- G에 대해: $\mathbf{l}(x) \cdot \mathbf{r}(x) = z^2 V + \delta + xt_1 + x^2 t_2$
- 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:
  - $\mathbf{H}' = y^{-n}\mathbf{H}$
    $P = A + xS - zG + (z\mathbf{y}^n + z^2\mathbf{2}^n)\mathbf{H}'$
    $P =? \mu H + \mathbf{lG} + \mathbf{rH}'$
- $\mathbf{H}'$의 도입은 $\mathbf{y}^\mathbf{n}$과의 아다마르 곱을 사용하는 $\mathbf{r(x)}$의 정의와,
- A의 $\mathbf{a_R}$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
- $V, n, \mathbf{G}, \mathbf{H}, G, H$ (shared in advance)

Prover		Verifier
A, S	$\rightarrow$	
	$\leftarrow$	y, z
T_1, T_2	$\rightarrow$	
	$\leftarrow$	x
$\tau_x, \mu, \hat{t}, \mathbf{l}, \mathbf{r}$	$\rightarrow$	
		verify

  - 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