ZKPs

A SURVEY ON THE ZERO-KNOWLEDGE PROTOCOLS ON BLOCKCHAIN

PAPER

- “블록체인에서의 영지식 프로토콜 현황 조사”
- 박상현(lukepark@altair.snu.ac.kr), 문수묵(smoon@snu.ac.kr)
- 서울대학교 전기°정보공학부

ABSTRACT

ABSTRACT

- 영지식 프로토콜을 사용
  - 정보를 공개하지 않으면서
  - 인코딩된 정보만으로 유효함을 입증

ABSTRACT

- 블록체인의 익명성 향상에 응용
- 블록체인의 확장성 증대에 응용
- 증명을 생성하고 검증하기 위해 추가적인 연산이 필요
- 일부 영지식 프로토콜은 별도의 신뢰 설정 페이즈를 요구

ABSTRACT

- 최신 영지식 프로토콜
  - zk-SNARKs
  - zk- STARKs
  - BulletProofs

INTRODUCTION

INTRODUCTION

- 영지식 프로토콜
  - 어떤 명제가 참임을 증명할 때
  - 그 문장의 참 혹은 거짓 여부를 제외한
  - 다른 어떠한 정보도 노출되지 않는 프로토콜
- 영지식 증명(Zero-Knowledge Proof, ZKP)으로도 칭함

INTRODUCTION

- 영지식성의 활용
  - 트랜잭션의 항목을 숨기면서도 유효함을 입증할 수 있음
  - 블록체인 익명성 향상의 핵심 기술로 주목

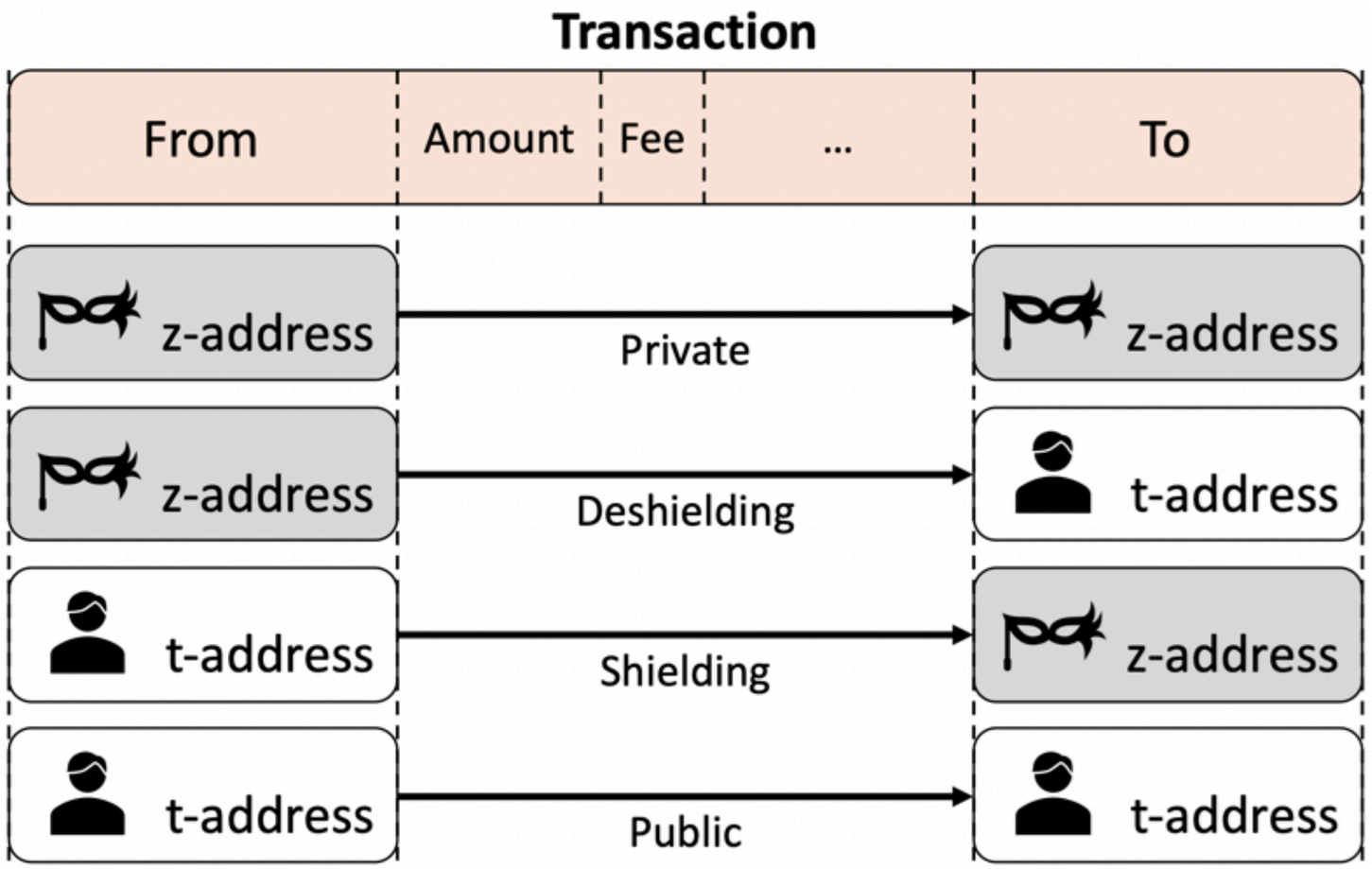
INTRODUCTION

- 지캐시(Zcash)
  - 영지식 논의인 zk-SNARKs를 활용한 블록체인

INTRODUCTION

- 개인 주소(z-address)와 공개 주소(t-address)
- 공개(public) 거래
  - 일반적인 비트코인에서의 거래와 동일
- 개인(private) 거래
  - 주소 및 금액이 영지식성을 통해 숨겨짐

INTRODUCTION



INTRODUCTION

- 영지식 프로토콜을 사용하려면
  - 증명자가 인코딩된 정보인 증명(proof)을 생성해야 함
  - 증거 또는 증명의 크기가 너무 크거나
  - 검증자의 연산에 추가적인 시간 및 공간 비용이 요구

INTRODUCTION

- 한계를 극복하고자 여러 종류의 영지식 프로토콜이 제시
  - zk-SNARKs
  - zk- STARKs
  - BulletProofs

ZERO KNOWLEDGE PROOFS

ZERO KNOWLEDGE PROOFS

- 증명자(Prover)
  - 영지식 증명에서 명제가 참임을 증명하고자 하는 참여자
- 검증자(Verifier)
  - 입증할 상대방

ZERO KNOWLEDGE PROOFS

- 영지식 증명은 다음의 세 성질을 만족해야 함
- 완전성(completeness)
  - 어떤 명제가 참이면,
    정직한 증명자는 정직한 검증자에게 이 사실을 납득시킬 수 있어야 함

ZERO KNOWLEDGE PROOFS

- 영지식 증명은 다음의 세 성질을 만족해야 함
- 건실성(soundness)
  - 어떤 명제가 거짓이면,
    어떠한 부정직한 증명자라도 정직한 검증자에게 명제가 참이라고 납득시킬 수 없음

ZERO KNOWLEDGE PROOFS

- 영지식 증명은 다음의 세 성질을 만족해야 함
- 영지식성(zero-knowledgeness)
  - 검증자는 어떤 명제의 참 혹은 거짓 여부 외에는 아무것도 알 수 없어야 함

ZERO KNOWLEDGE ARGUMENTS

ZERO KNOWLEDGE ARGUMENTS

- 영지식 논의(Zero-Knowledge Argument)
  - 증명이 오직 계산적이라는 의미를 내포
- 영지식 증명은 확률적인 건실성을 제공
- 영지식 논의는 연산적 건실성을 제공

ZERO KNOWLEDGE ARGUMENTS

- 다음 항목들이 바로 영지식 논의 프로토콜들
  - zk-SNARKs
  - zk- STARKs
  - BulletProofs

ZK-SNARKS

ZK-SNARKS

- 2011년 Nir Bitansky 등
- 간결한 지식 논의 프로토콜 SNARK를 정의
  - Succinct Non-interactive adaptive ARgument of Knowledge
- 이를 영지식 논의에 적용한 것이 zk-SNARK

ZK-SNARKS

- 지캐시 구현체에 적용되며 널리 알려짐
- 영지식 논의를
  - 비-상호적이고(non-interactive)
  - 간결하게(succinct)
  - 변형한 형태

ZK-SNARKS

- 비-상호적
  - 프로토콜을 진행하면서 증명자와 검증자의 실시간 소통이 필요치 않음
- 간결
  - 증명의 크기가 유의미하게 작음

ZK-SNARKS

- 공개 매개변수를 설정하기 위한
  - 별도의 신뢰 설정 페이즈(trusted setup phase)를 요구
  - 공개 매개변수를 생성하는 무작위 과정
- 이 과정에 공격자가 개입하면 문제가 발생
  - 공격자는 유효하다고 판단되는 거짓 증명을 만들 수 있음

ZK-SNARKS

- 임의의 연산을 zk-SNARK에서의 증명 형태로 변환하는 과정
  - 연산 → 회로 → $R1CS \rightarrow QAP \rightarrow zk\text{-}SNARK$

ZK-SNARKS

- 연산 → 회로 → $R1CS$ → QAP → $zk\text{-}SNARK$
- 연산을 가능한 작은 산술 회로(arithmetic circuit)로 치환

ZK-SNARKS

- 연산 → 회로 → $R1CS$ → QAP → $zk\text{-}SNARK$
- 회로를 R1CS(Rank 1 Constraint System) 형태로 변환

ZK-SNARKS

- 연산 → 회로 → $R1CS \rightarrow QAP \rightarrow zk\text{-}SNARK$
- R1CS 형태로 표현된 회로를
  다항식의 집합인 이차 산술 프로그램(Quadratic Arithmetic Program, QAP)으로 변환

ZK-SNARKS

- 연산 → 회로 → $R1CS$ → QAP → $zk\text{-}SNARK$
- 다항식을 무작위 시프트(random shifts)해
  동일성을 만족하면서 영지식성을 확보

ZK-SNARKS

- 연산 → 회로 → $R1CS$ → QAP → $zk\text{-}SNARK$
- 다항식의 집합인 이차 산술 프로그램
- 슈바르츠-지펠 보조정리(Schwartz-Zippel Lemma)
  - 서로 다른 다항식은 대부분의 점에서 서로 다른 값을 가짐
  - 임의의 점에 대해 다항식-연산-들이 같은 값을 가짐을 보이면,
    높은 확률로 다항식-연산-들이 같음을 보일 수 있음

ZK-SNARKS

- 공개 매개변수로부터
  - 평가에 사용할 QAP에서의 점이
  - 암호화된 형식으로 제공
- 공개 매개변수가 정직하게 형성되면
  - 검증자나 증명자 누구도 이 점에 대해 알지 못함
  - 부정한 다항식 및 증명을 구성할 수 없음

ZK-STARKS

ZK-STARKS

- Succinct Transparent ARguments of Knowledge
  - 간결한(Succinct) 대신에
  - 확장가능한(Scalable)이라고도 함

ZK-STARKS

- 연산의 크기가 증가해도 소요시간 증가는 거의 없음
  - SNARks는 더 많은 소요시간을 필요로 함
- 증명의 크기가 SNARKs나 BulletProofs 대비 큼

ZK-STARKS

- 신뢰 설정에 의존하지 않음
  - 신뢰 설정은 SNARKs의 주요 취약점

ZK-STARKS

- 더 간단한 암호학적 가정에 기반
  - 타원 곡선의 필요에서부터 벗어남
  - 오직 해시와 정보 이론(information theory)에만 의존
- 양자 내성(Post-Quantum)을 확보

ZK-STARKS

- 여러 STARKs 프로토콜들이 활발히 연구 중
  - FRI-STARKs
  - PLONK(FRI)
  - SuperSonic
  - Fractal
  - 등

ZK-STARKS

- FRI 프로토콜을 사용한 STARKs
  - 증명자의 산술 복잡도를 엄밀한 선형(strictly linear)으로,
  - 검증자의 산술 복잡도를 엄밀한 로그(strictly logarithmic)로 달성한
  - 최초의 솔루션이다

BULLETPROOFS

BULLETPROOFS

- 벡터를 공개하지 않으면서도 알고 있음을 간결히 보이는 영지식 논의
- 내적 증명(inner proof)
  - 어떠한 공개하지 않은 스칼라값이 공개하지 않은 벡터들의 내적임을 증명

BULLETPROOFS

- 내적 증명(inner proof)
  - Groth가 논문에서 제시한 핵심 알고리즘에서부터 시작
  - Bootle의 재귀(recursion) 개념을 도입한 더 간결한 내적 증명으로 발전
  - Bünz 외 연구자들이 제시한 더 간결한 형태인 BulletProofs에 도달

BULLETPROOFS

- 간결함
  - 상호 통신하는 데이터의 크기가 작다는 의미
  - 연산의 단순함이나 소요 시간과는 관계가 없음

BULLETPROOFS

- 증명자와 검증자의 연산에 소요되는 시간
  - zk-SNARKs와 zk-STARKs 대비 큼

BULLETPROOFS

- 내적 지식 논의의 예: 범위 증명(range proof)
  - 값 v가 n 비트의 특정 범위 $v \in [0, 2)$ 에 있음을 보임
- 임의의 문제에 대한 일반 산술 회로(General arithmetic circuits)로 일반화

SUMMARY

SUMMARY

- 상세한 수치는 각 영지식 프로토콜을 구현한 방법과 환경에 따라 달라질 수 있음

	zk-SNARKs		zk-STARKs		BulletProofs	
증명자 연산 복잡도 (시간)	$O(N * \log(N))$	2.3 s	$O(N * \text{poly-}\log(N))$	~1.6 s	$O(N * \log(N))$	~30 s
검증자 연산 복잡도 (시간)	$\sim O(1)$	10 ms	$O(\text{poly-}\log(N))$	~16 ms	$O(N)$	~1100 ms
증명 크기 (바이트)	$\sim O(1)$	~200 Bytes	$O(\text{poly-}\log(N))$	45-200 KB	$O(\log(N))$	~1.3 KB
신뢰 설정	필요함		필요 없음		필요 없음	
양자 내성	없음		있음		없음	

SUMMARY

- zk-SNARKs
  - 증명 크기가 수백 바이트 정도로 작음
  - 검증에 필요한 연산 복잡도도 낮음
- 연산량 제한이 있는 스마트 컨트랙트(smart contract)에서도 활용 가능
- 별도의 신뢰 설정 페이즈를 요구

SUMMARY

- zk-STARKs
  - 신뢰 설정이 필요 없으나 증명 크기가 상대적으로 큼
  - 양자 내성

SUMMARY

- BulletProofs
  - 신뢰 설정이 필요 없음
  - 증명 크기도 작음
  - 연산 복잡도가 높음

CONCLUSION

CONCLUSION

- 영지식 프로토콜에 따라 특성이 다름
  - 증명자 및 검증자의 연산 복잡도
  - 증명 크기
  - 신뢰 설정 페이즈의 유무
  - 양자 내성 등의 특성

CONCLUSION

- 영지식 프로토콜에 따라 특성이 다름
- 이러한 특성들을 고려하여
  - 블록체인 프로토콜에
  - 스마트 컨트랙트에
- 가장 적합한 프로토콜을 선택해야 함

ZKPs

A SURVEY ON THE ZERO-KNOWLEDGE PROTOCOLS ON BLOCKCHAIN