KACHINA: Private Smart Contracts

Thomas Kerber
papers@tkerber.org

Aggelos Kiayias
akiayias@ed.ac.uk

Markulf Kohlweiss
mkohlwei@ed.ac.uk

The University of Edinburgh & IOHK

September 23, 2020

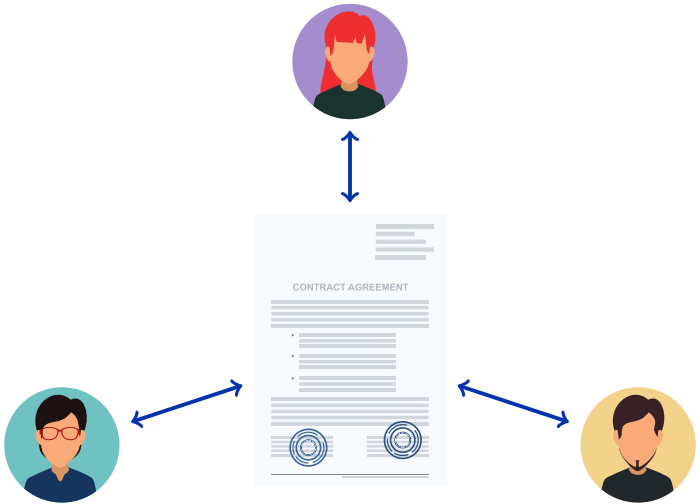
Overview

- What *is* privacy in smart contracts?
- Existing solutions
- Common tradeoffs and issues
- Providing better foundations for contract developers

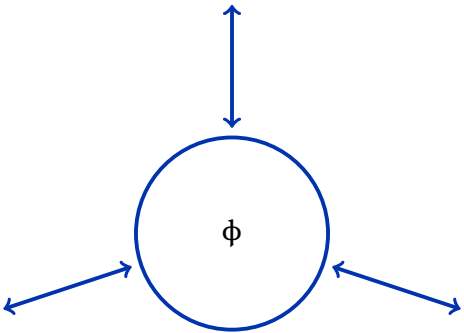
What are smart contracts?

- Smart contracts are **Ethereum**?
  Does not help in figuring out privacy.
- Smart contracts are programmable **legal contracts**?
  Too powerful and vague to be useful.

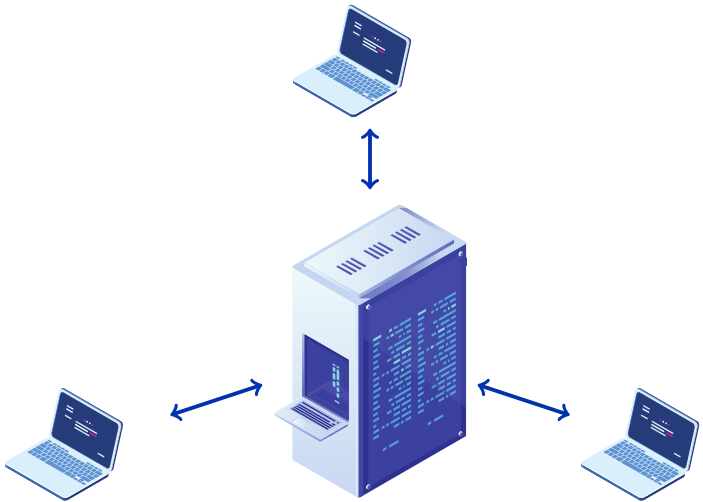
Reactive State Machines



Reactive State Machines



Relation to Client/Server Model



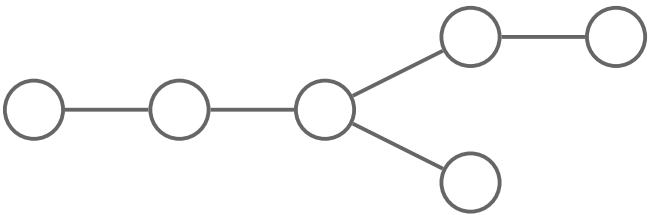

ebay


amazon

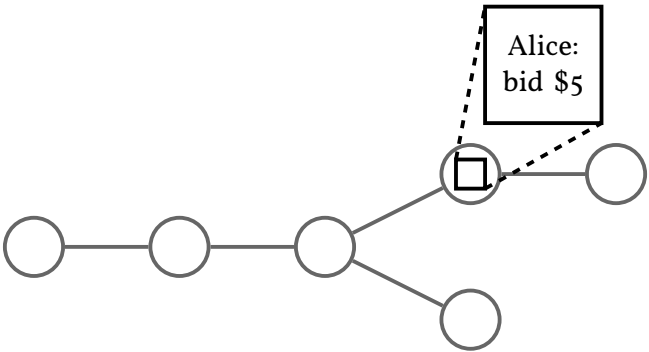

facebook

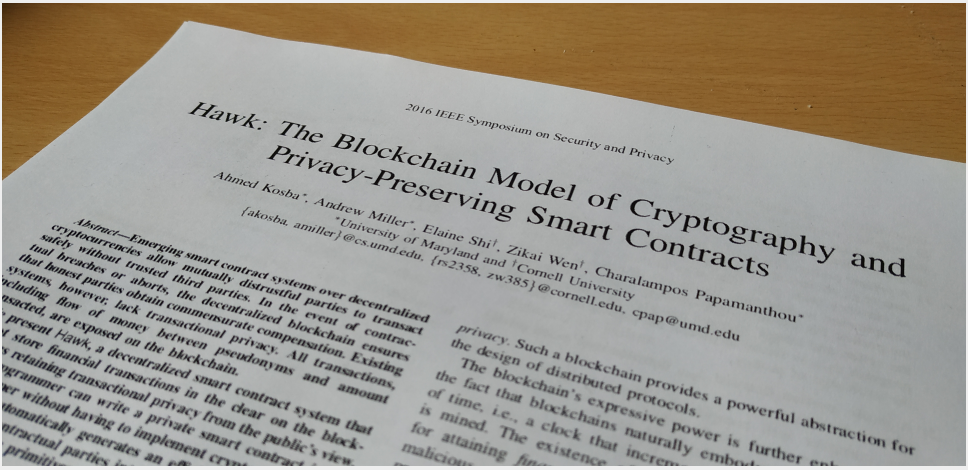
Centralised privacy relies on trust

Blockchains and Smart Contracts



Blockchains and Smart Contracts





2016 IEEE Symposium on Security and Privacy

Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts

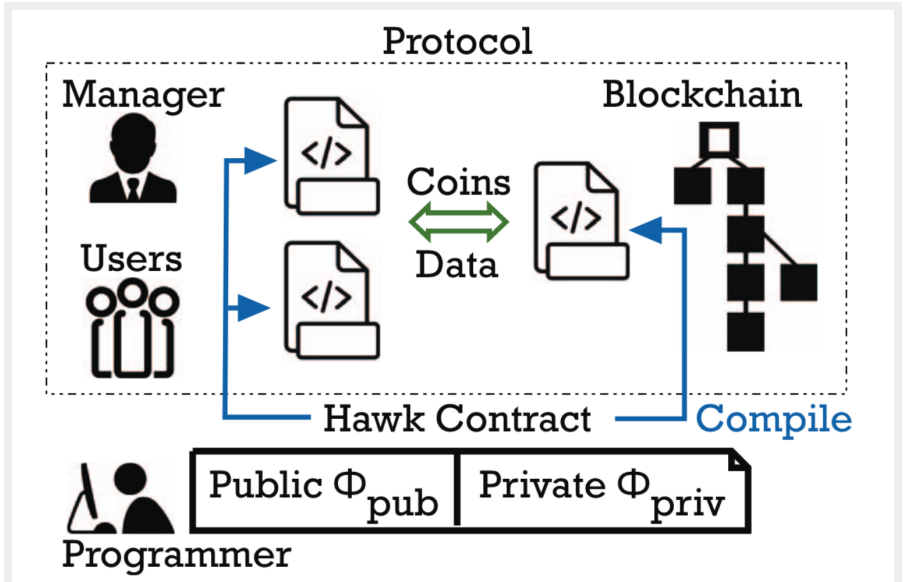
Ahmed Kosba*, Andrew Miller*, Elaine Shi†, Zikai Wen†, Charalampos Papamanthou*
*University of Maryland and †Cornell University
{akosba, amiller}@cs.umd.edu, {rs2358, zw385}@cornell.edu, cpap@umd.edu

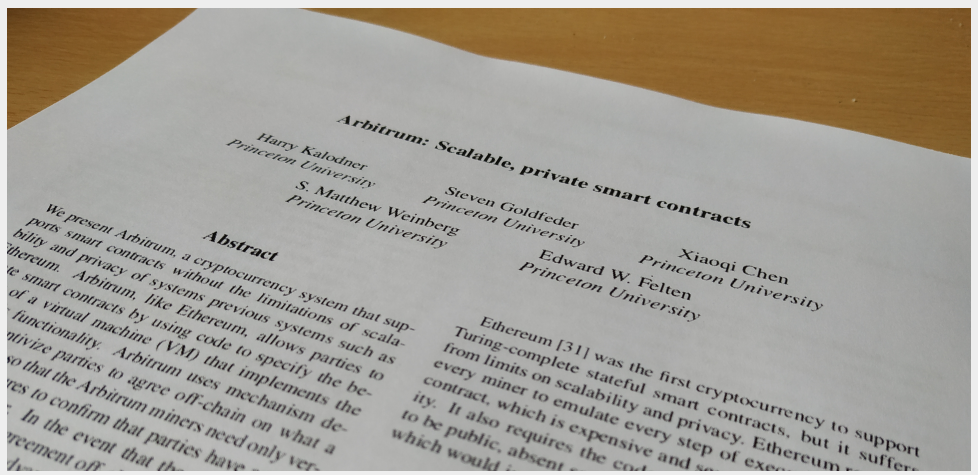
Abstract—Emerging smart contract systems over decentralized cryptocurrencies allow mutually distrustful parties to transact safely without trusted third parties. In the event of contractual breaches or aborts, the decentralized blockchain ensures that honest parties obtain commensurate compensation. Existing systems, however, lack transactional privacy. All transactions, including how much money between pseudonyms and amount asacted, are exposed on the blockchain.

present Hawk, a decentralized smart contract system that store financial transactions in the clear on the blockretaining transactional privacy from the public's view. grammer can write a private smart contract ner without having to implement cryp domatically generates an effi ntractual parties in primitive

privacy. Such a blockchain provides a powerful abstraction for the design of distributed protocols.

The blockchain's expressive power is further en the fact that blockchains naturally embod of time, i.e., a clock that increm is mined. The existence for attaining fin malicious


Protocol
Manager
Blockchain
Coins
Data
Users
Hawk Contract
Compile
Public Φpub
Private Φpriv
Programmer



Arbitrum: Scalable, private smart contracts

Harry Kalodner
Princeton University

Steven Goldfeder
Princeton University

Xiaoqi Chen
Princeton University

S. Matthew Weinberg
Princeton University

Edward W. Felten
Princeton University

Abstract

We present Arbitrum, a cryptocurrency system that supports smart contracts without the limitations of scalability and privacy of systems previous systems such as Ethereum. Arbitrum, like Ethereum, allows parties to make smart contracts by using code to specify the behavior of a virtual machine (VM) that implements the functionality. Arbitrum uses mechanism design to incentivize parties to agree off-chain on what a VM would do so that the Arbitrum miners need only verify enough to confirm that parties have agreed correctly. In the event that the parties have

Ethereum [31] was the first cryptocurrency to support Turing-complete stateful smart contracts, but it suffers from limits on scalability and privacy. Ethereum requires every miner to emulate every step of execution of every contract, which is expensive and slow. It also requires the code to be public, absent special measures, which would

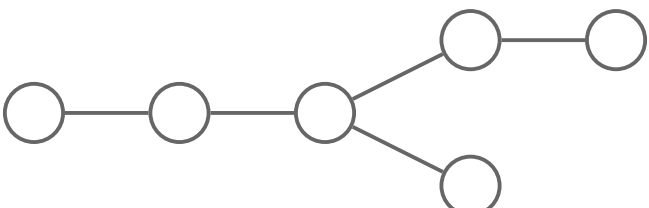

PLUNDER
VOLT

Perfect Privacy

- The same reactive state machine
- No leakage
- Decentralised implementation

- Multi-party computation (MPC) achieves this!
- Run a committee-based chain (e.g. Algorand)
- Have the same committee run MPC for each contract call
- Prohibitively expensive

Decentralisation?



This setting limits privacy

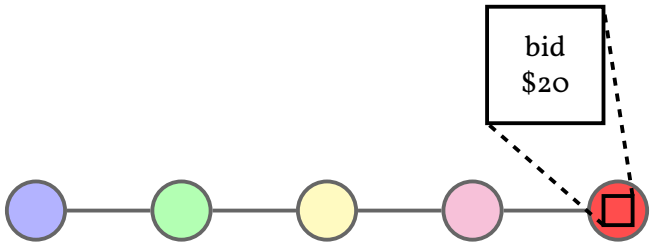
Example: The King of Ether

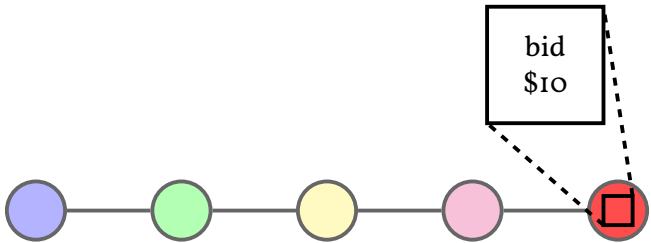
- The “throne” can be bought
- The price increases exponentially
- The previous king gets the proceeds
- A fee is paid for each attempt

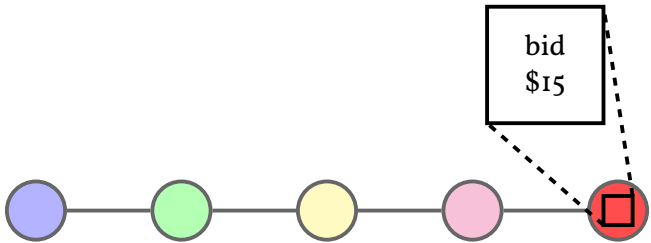
A private variant would hide:

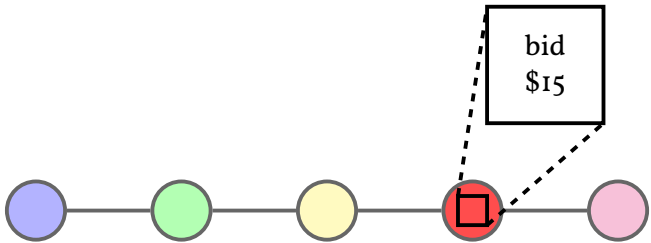
- The value of the throne
- Who holds it
- When it was obtained

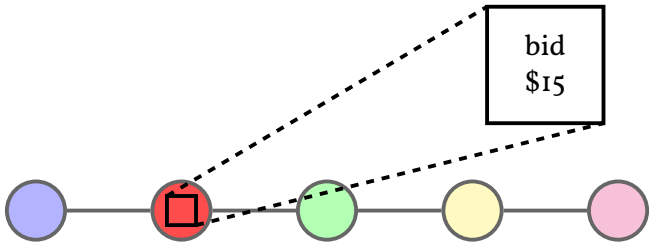
It cannot


bid
$20

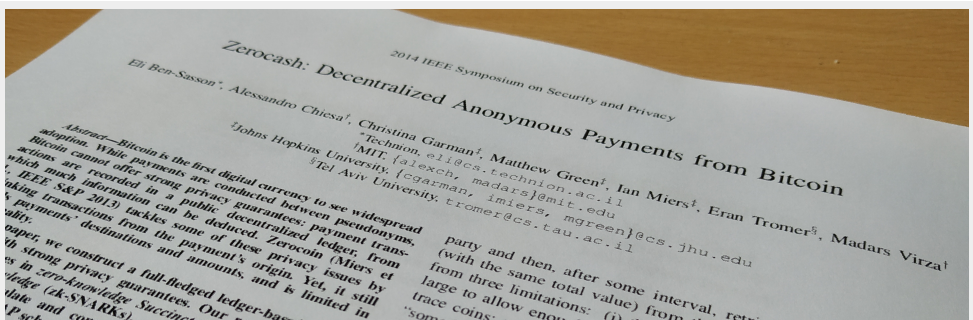

bid
$10


bid
$15


bid
$15


bid
$15

There is Hope!



2014 IEEE Symposium on Security and Privacy

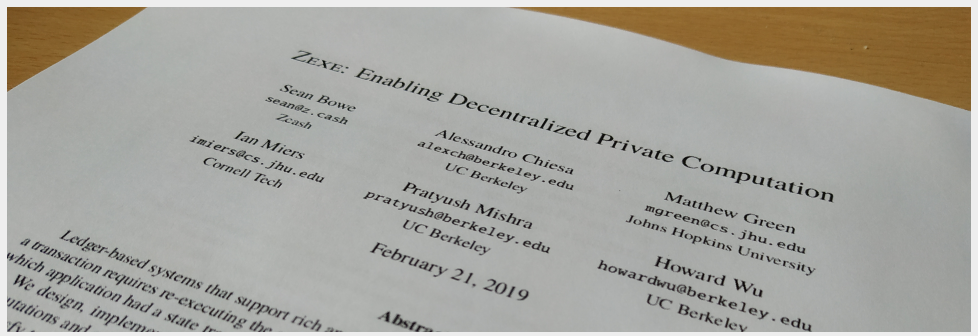
Zerocash: Decentralized Anonymous Payments from Bitcoin

Eli Ben-Sasson*, Alessandro Chiesa†, Christina Garman‡, Matthew Green‡, Ian Miers‡, Eran Tromer§, Madars Virza†
*Technion, eli@cs.technion.ac.il
†MIT, {alexch, madars}@mit.edu
‡Johns Hopkins University, {cgarman, imiers, mgreen}@cs.jhu.edu
§Tel Aviv University, tromer@cs.tau.ac.il

Abstract—Bitcoin is the first digital currency to see widespread adoption. While payments are conducted between pseudonyms, Bitcoin cannot offer strong privacy guarantees: payment transactions are recorded in a public decentralized ledger, from which much information can be deduced. Zerocoin (Miers et al., IEEE S&P 2013) tackles some of these privacy issues by unlinking transactions from the payment's origin. Yet, it still requires payments' destinations and amounts, and is limited in scalability.

In this paper, we construct a full-fledged ledger-based anonymous payment system that offers strong privacy guarantees. Our system builds on recent cryptographic techniques, namely zero-knowledge Succinct Non-interactive ARguments of Knowledge (zk-SNARKs).

There is Hope!



ZEXE: Enabling Decentralized Private Computation

Sean Bowe
sean@z.cash
Zcash

Ian Miers
imiers@cs.jhu.edu
Cornell Tech

Alessandro Chiesa
alexch@berkeley.edu
UC Berkeley

Pratyush Mishra
pratyush@berkeley.edu
UC Berkeley

Matthew Green
mgreen@cs.jhu.edu
Johns Hopkins University

Howard Wu
howardwu@berkeley.edu
UC Berkeley

February 21, 2019

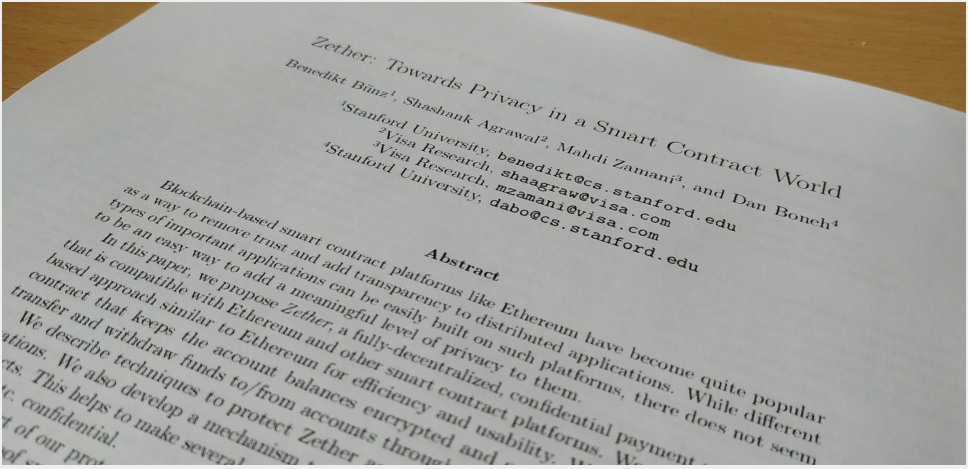
Ledger-based systems that support rich
a transaction requires re-executing the
which application had a state
We design, implem
tations and

There are perfect solutions
for individual problems

Contract Modularity

- Developing distributed systems used to be hard.
- Components interacting is very powerful!
  - Tokens provide assets to write contracts *about*
  - Wallet contracts allow complex access control
  - Basic functionality can be reused

Privacy extensions often harm modularity!



Zether: Towards Privacy in a Smart Contract World

Benedikt Bünz[1], Shashank Agrawal[2], Mahdi Zamani[3], and Dan Boneh[4]

[1]Stanford University, benedikt@cs.stanford.edu
[2]Visa Research, shaagraw@visa.com
[3]Visa Research, mzamani@visa.com
[4]Stanford University, dabo@cs.stanford.edu

Abstract

Blockchain-based smart contract platforms like Ethereum have become quite popular as a way to remove trust and add transparency to distributed applications. While different types of important applications can be easily built on such platforms, there does not seem to be an easy way to add a meaningful level of privacy to them.

In this paper, we propose *Zether*, a fully-decentralized, confidential payment that is compatible with Ethereum and other smart contract platforms. We describe techniques to protect Zether

zkay: Specifying and Enforcing Data Privacy in Smart Contracts

Samuel Steffen
ETH Zurich, Switzerland
samuel.steffen@inf.ethz.ch

Benjamin Bichsel
ETH Zurich, Switzerland
benjamin.bichsel@inf.ethz.ch

Mario Gersbach
ETH Zurich, Switzerland
gmario@student.ethz.ch

Noa Melchior
ETH Zurich, Switzerland
noame@student.ethz.ch

Petar Tsankov
ETH Zurich, Switzerland
petar.tsankov@inf.ethz.ch

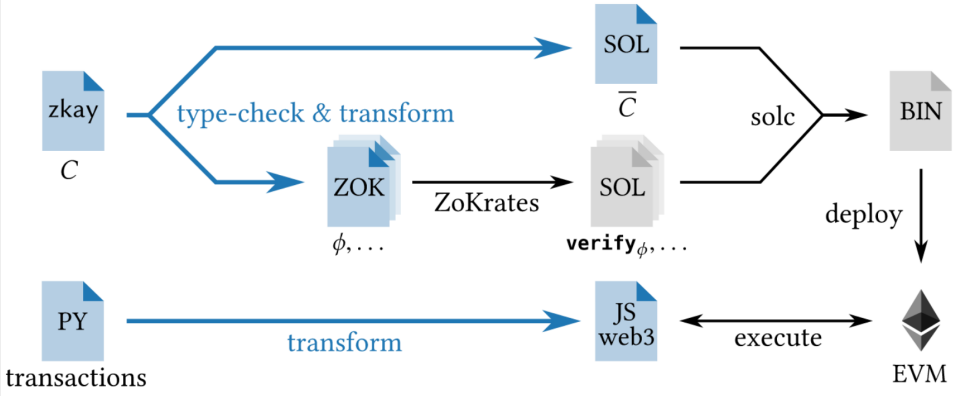
Martin Vechev
ETH Zurich, Switzerland
martin.vechev@inf.ethz.ch

ABSTRACT

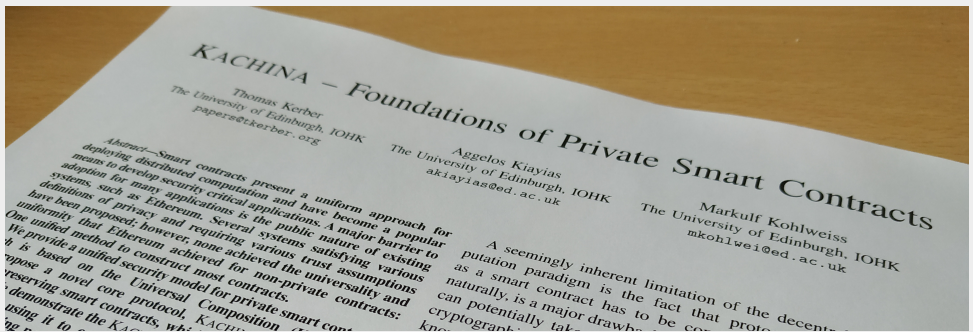
Privacy concerns of smart contracts are a major roadblock prevent-
ing their wider adoption. A promising approach to protect private
data is hiding it with cryptographic primitives and then enforcing
correctness of state updates by Non-Interactive Zero-Knowledge
(NIZK) proofs. Unfortunately, NIZK statements are less expressive
than smart contracts, forcing developers to keep some functionality
in contract. This results in scattered logic, split across contract
and NIZK statements, with unclear privacy guarantees.
To these problems, we present the zkay language, which
privacy types defining owners of private values. zkay
fully type checked to (i) ensure they are realizable
and (ii) prevent unintended information leaks.
zkay contracts is easy to follow by just
enforce zkay contracts, we automatically
is equivalent in terms of privacy and
public blockchains.
a proof-of-con
and imp

1 INTRODUCTION

Smart contracts have gained significant popularity in recent years.
In a nutshell, they are programs deployed on top of blockchains,
such as Ethereum [53], that enable trusted execution without a
trusted third party. To benefit from trusted execution with no in-
termediary, many real-world processes (e.g., trading [1] or insur-
ance [45, 50]) are being ported to smart contracts.
When implementing applications in smart contracts
concern is data privacy: Smart contract transactio
calls initiated by users) are processed by
(called miners), which requires the
to be made available to all
that handle sensitive
of medic


zkay
C
type-check & transform
SOL
$\overline{C}$
solc
BIN
ZOK
$\phi, \ldots$
ZoKrates
SOL
verify$_\phi, \ldots$
deploy
PY
transactions
transform
JS
web3
execute
EVM

Kachina



KACHINA – Foundations of Private Smart Contracts

Thomas Kerber
The University of Edinburgh, IOHK
paperz@tkerber.org

Aggelos Kiayias
The University of Edinburgh, IOHK
akiayias@ed.ac.uk

Markulf Kohlweiss
The University of Edinburgh, IOHK
mkohlwei@ed.ac.uk

Abstract—**Smart contracts present a uniform approach for deploying distributed computation and have become a popular means to develop security critical applications. A major barrier to adoption for many applications is the public nature of existing systems such as Ethereum. Several applications have proposed definitions of privacy and requiring various trust assumptions have been proposed; however, none achieved the universality and uniformity that Ethereum achieved for non-private contracts: One unified method to construct most contracts.**

We provide a unified security model for private smart contracts. h is based on the Universal Composition

pose a novel core protocol, KACHINA, reserving smart contracts, which

demonstrate the KACHINA using it to

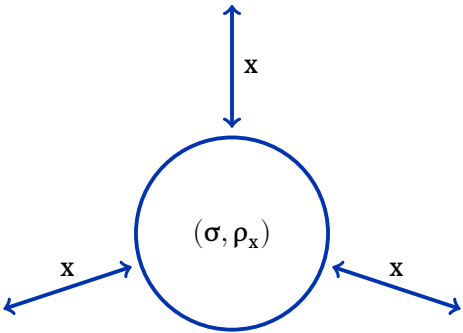
A seemingly inherent limitation of the decentr putation paradigm is the fact that proto as a smart contract has to be co naturally, is a major drawb can potentially tak cryptographi

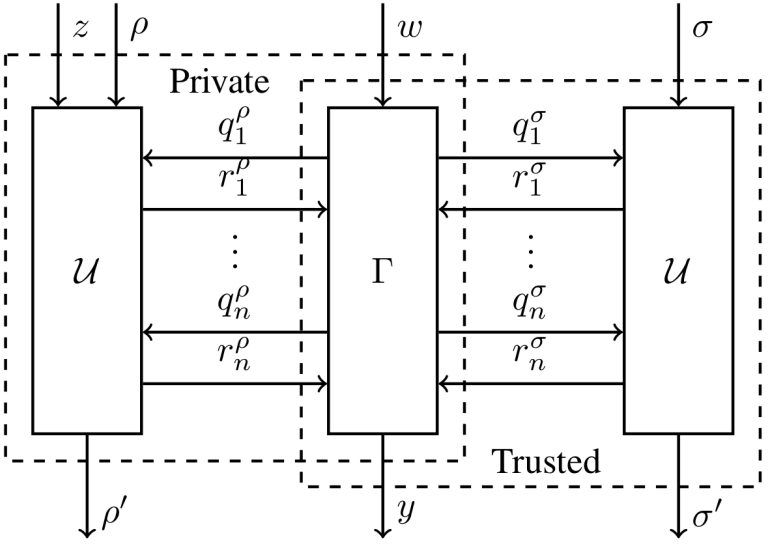
The Tools

We have two main tools at our disposal:

- The blockchain
- Users' local machines

Back to State Machines




z
ρ
w
σ
Private
q_1^ρ
r_1^ρ
q_1^σ
r_1^σ
$\mathcal{U}$
Γ
$\mathcal{U}$
q_n^ρ
r_n^ρ
q_n^σ
r_n^σ
Trusted
ρ′
y
σ′


ZEROCASH


PRIVATE
PAYMENTS


ZEROCASH

≈


PRIVATE
PAYMENTS

- Modular design of contracts is possible again
- Including complex features such as gas costs
- Without compromising on privacy
- New privacy techniques can be implemented *as contracts*

Thank you!

Please see `https://drwx.org/2020-07-03-copyright.txt` for copyrights and attribution of used images.