KACHINA: Private Smart Contracts

Thomas Kerber
papers@tkerber.org

Aggelos Kiayias
akiayias@ed.ac.uk

Markulf Kohlweiss
mkoohlwe@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

\[ \phi \]
Relation to Client/Server Model
ebay
facebook
Centralised privacy relies on trust
Blockchains and Smart Contracts
Blockchains and Smart Contracts

Alice: bid $5
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 amnally 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 flow of money between pseudonyms and amounts, are exposed on the blockchain.

Hawk presents a decentralized smart contract system that stores financial transactions in the clear on the blockchain, yet maintains transactional privacy from the public’s view. A smart contract program can write a private smart contract automatically without having to implement cryptography primitives. Malicious parties in the network cannot forge the contract.

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

The blockchain’s expressive power is further enhanced by the fact that blockchains naturally embody the notion of time, i.e., a clock that increments as blocks are mined. The existence of such a clock facilitates the expression for attaining finite security in cryptographic primitives.
Arbitrum: Scalable, private smart contracts

Harry Kalodner  
Princeton University

S. Matthew Weinberg  
Princeton University

Steven Goldfeder  
Princeton University

Xiaohui Chen  
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 previous systems such as Ethereum. Arbitrum, like Ethereum, allows parties to execute smart contracts by using code to specify the behavior of a virtual machine (VM) that implements the functionality. Arbitrum uses mechanisms to incentivize parties to agree off-chain on what a contract should do so that the Arbitrum miners need only verif...
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
$15
bid $15
There is Hope!
There is Hope!
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, Shashank Agrawal, Mahdi Zamani, and Dan Boneh

\(^2\)Stanford University, benedikt@cs.stanford.edu
\(^3\)Visa Research, shaagraw@visa.com
\(^4\)Visa Research, mzamani@visa.com
\(^5\)Stanford University, dabo@cs.stanford.edu

Abstract

Blockchain-based smart contract platforms like Ethereum have become quite popular in recent years. As a way to remove trust and add transparency to distributed applications, various types of important applications can be easily built on such platforms. While different approaches may be an easy way to add a meaningful level of privacy to such networks, there does not seem to be a fully-decentralized, confidential payment

In this paper, we propose Zether, a fully-decentralized, confidential peer-to-peer payment protocol based on Zether, a fully-decentralized, confidential contract platform that is compatible with Ethereum and other smart contract platforms. We describe techniques to transfer funds to/from accounts through encrypted channels. We also develop a mechanism to protect Zether payments from prying eyes. This helps to make several applications of our protocol.

We describe techniques to protect Zether payments from prying eyes. This helps to make several applications of our protocol.
zkay: Specifying and Enforcing Data Privacy in Smart Contracts

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

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

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

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

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

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

ABSTRACT
Privacy concerns of smart contracts are a major roadblock preventing their wider adoption. A promising approach to protect private data is hiding it with cryptographic primitives and then enforcing the 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 of a contract, with unclear privacy guarantees.

We present the zkay language, which addresses these problems, we present the zkay language, which directly type checks owners of private values. zkay provides two main capabilities: (i) zkay contracts are easily checked (ii) prevent unintended information leaks. zkay contracts are easy to follow by just enforcing zkay contracts, we automatically enforce zkay contracts, we automatically public blockchain in terms of privacy and a proof-of-concept implementation.

I INTRODUCTION
Smart contracts have gained significant popularity in recent years, such as Ethereum [53], that enable trusted execution with no intermediary, many real-world processes (e.g., trading [1] or insurance [45, 50]) are being ported to smart contracts.

When implementing applications in smart contracts, concern is data privacy. Smart contract transactions (called initiated by users) are processed by trusted third-party nodes (called miners), which requires the transactions to be made available to all that handle sensitive or medical-specific data.
Kachina

KACHINA – Foundations of Private Smart Contracts

Thomas Kerber
The University of Edinburgh, IOHK
papers@kerber.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 systems satisfying various 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 provides a unified security model for private smart contracts, KACHINA, which is based on the Universal Composition framework. The KACHINA protocol preserves smart contracts, which can potentially take advantage of cryptographic primitives.

A seemingly inherent limitation of the current computation paradigm is the fact that Protocol as a smart contract has to be completely known to the users; this can potentially expose smart contracts to a major security risk.
The Tools

We have two main tools at our disposal:

- The blockchain
- Users’ local machines
Back to State Machines

\[(\sigma, \rho_x)\]
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.