CSE5095-010: Blockchain Technology

Lecture 16

Ghada Almashaqbeh
UConn - Fall 2020

Outline

- Anonymity and privacy in cryptocurrencies.
  - Background.
  - Centralized mixers.
  - Cryptography Review.
    - Commitments.
    - Zero knowledge proofs.
  - Decentralized mixers - Zerocoin.
  - Anonymous and private payments.
    - UTXO model - Zerocash.
    - Account model - Zether.

Anonymity and Privacy I

- Sensitive information in a cryptocurrency system:
  - Addresses of senders and recipients.
  - Transaction (currency) value.
  - Account balance (for these that use the account model).
  - Executed code (scripts or smart contracts).
  - Inputs and outputs of this executed code.
- Anonymity.
  - Hiding the addresses of senders and recipients.
- Privacy preserving:
  - Generally, it applies to the last four items in the above list.

Anonymity and Privacy II

- In some sources,
  - Hiding identities is also considered a privacy-preserving issue.
  - Hiding balances and transaction values are referred to as confidentiality.
    - E.g., confidential transactions; those with encrypted currency values.
- We will refer to these as:
  - ***Private payments.*** Currency transfer transactions that hide values and balances.
  - ***Secure (or privacy-preserving) function evaluation.*** Computing over private inputs, and possibly, producing private output.
  - ***Function privacy.*** Hiding the function (executed code) itself.
  - ***Anonymity.*** Hiding user identities.

Is Bitcoin Anonymous?

- Believed to be.
  - No real identities are required.
  - Users use random-looking keys as pseudonyms.
  - It is advised to generate a new key pair for each new transaction.

WikiLeaks

Bitcoin

Bitcoin is a secure and anonymous digital currency. Bitcoins cannot be easily tracked back to you, and are safer and faster alternative to other donation methods. You can send BTC to the following address:

1HB5XMLmzFVj8ALj6mfBsbifRoD4miY36v

Various sites offer a service to exchange other currency to/from Bitcoins. There are also services allowing trades of goods for Bitcoins. Bitcoins are not subject to central regulations and are still gaining value. To learn more about Bitcoins, visit the website (https://bitcoin.org) or read more on Wikipedia.

For a more private transaction, you can click on the refresh button above to generate a new address

Source: https://shop.wikileaks.org/donate

No it is not …

- The blockchain is public.
  - Transactions do not hide addresses of senders and recipients.
- Transactions linkability.
  - Track transaction flow to infer the real identities of the involved parties.
    - Cluster Bitcoin addresses into entities, link them to identities and/or Bitcoin addresses posted by their owners on forums, blogs, etc., [Reid et al. 2014]
    - Link this flow to users' IPs [Koshy et al. 2014].
      - Here, the use of anonymous communication protocols (e.g., Tor) could be useful. But anonymity is based on the security guarantees of such protocols (recall exit and entry points in Tor see the flow in the clear).

Is Bitcoin Private?

- Also NO.
  - Again, its blockchain is public.
  - Values of transactions are recorded in the clear.
  - Transaction scripts (locking and unlocking) are publicly known and logged in the clear as well.
  - Scripts operate on public inputs and produce public outputs.

How about Ethereum?

- For Ethereum, same as Bitcoin, it is more about functionality extension rather than privacy/anonymity.
- The account model requires different privacy/anonymity techniques than those used in the UTXO model.
- Having arbitrary smart contracts deployed by users raises the expectations.
  - Can these contracts operate on private inputs and produce private outputs?
    - Can existing techniques (e.g., MPC) be used here?
  - Can we preserve the privacy of the code itself? (i.e., hide the performed computation as well.)

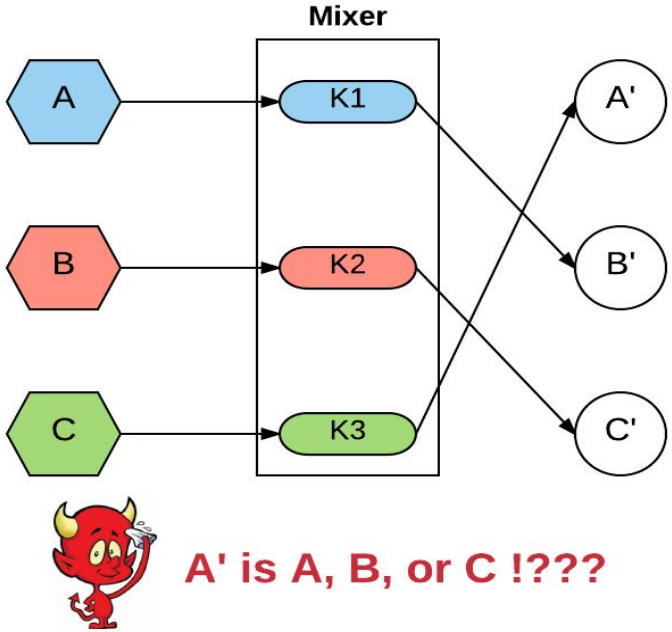
Does Anonymity/Privacy Matter?

- Just like traditional banking systems, we desire to hide our payment activity when needed/possible.
  - Blockchain records are public, anyone can access them at any time.
- Storing and processing sensitive data.
  - Blockchain-based applications for medical records, trading, auction, voting, etc.
- Without anonymity/privacy, one my forgo the advantages of employing a blockchain in such highly sensitive applications.
  - Front running in auctions, censorship in voting, etc.
- Sometimes in cryptocurrencies coins get tainted.
  - People reject coins that have some undesirable history.

Potential Solutions

- Mixing services (mainly in the context of Bitcoin).
  - Centralized.
  - Decentralized - Zerocoin.
- Anonymous/private cryptocurrencies.
  - UTXO model.
    - ZeroCash - an extension of Bitcoin.
  - Account model.
    - Zether - a token on top of Ethereum.

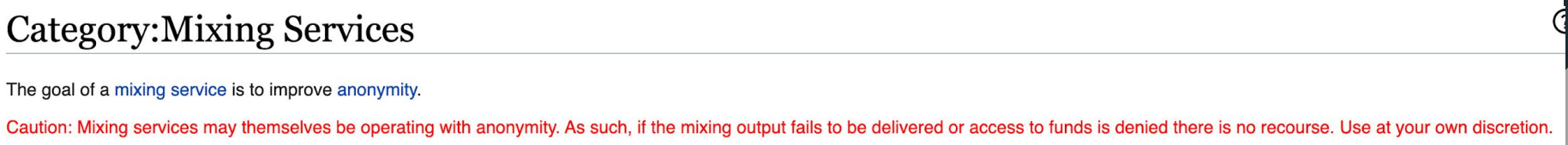
Mixers



- Break transaction linkability.
  - Participants send their coins to some entity, the mixer (or tumbler).
  - The mixer shuffles these coins and return them to them back.
    - Each party gets same value back but from a different owner (users use fresh addresses to receive these).

Centralized Mixers

From Bitcoin wiki (https://en.bitcoin.it/wiki/Category:Mixing_Services)


Category:Mixing Services

The goal of a mixing service is to improve anonymity.

Caution: Mixing services may themselves be operating with anonymity. As such, if the mixing output fails to be delivered or access to funds is denied there is no recourse. Use at your own discretion.

- Everything is controlled by a trusted party.
  - Parties sends their coins with a promise to return them back.
  - Huge trust risk, will the mixer ***return the coins back***?
    - Several theft incidents over the past years.
- The mixer has a full record of which coins were sent to who.
  - It still have the transaction linkability information.
  - Will it ***delete this record*** and not reveal them later?
- Do we trust the mixer to randomly shuffle coins?
  - May send coins in a non-random manner allowing deanonymization.

Mixcoin[Bonneau et al., 2014]

- Although anonymous cryptocurrencies were already out there, the goal is to have something efficiency and fully compatible with Bitcoin.
- Add accountability to expose theft.
  - A mixer issues a warranty to return the coins.
    - If it does not, the user exposes this warranty, destroying the reputation of the mixer.
  - The mixer creates an escrow address to which the funds will be sent.
  - Later, it shuffles the escrows and send each user an equal amount of funds back (to new fresh addresses).
- Calibrate incentives so that rational mixers will act honestly.
- Propose the use of a series of mixers to reduce the probability of local records risk.

Mixcoin[Bonneau et al., 2014]

- Still same security risks of a centralized mixer.
  - Theft.
    - Maybe it is worth it; destroy reputation but run away with a huge wealth.
  - Delays.
    - Users have to wait for long time to get coins back (to have a large anonymity set).
  - Local records exposure.
    - Mix networks (series of mixers) may not be always available.

- Detour -
Commitments
Zero Knowledge Proofs

Commitments

- A commitment scheme consists of three algorithms:
  - Setup: takes a security parameter $\boldsymbol{\lambda}$ as input and outputs public parameters **pp**.
  - Commit: takes inputs the public parameters **pp**, a message ***m***, and randomness ***r***, and outputs commitment ***c***.
- Opening a commitment ***c*** is simply revealing ***m*** and ***r***, and then verifying that **commit(*m; r*) = *c***.
- Acts like a digital envelope; commit to ***m*** (like a guarantee that ***m*** exists) but without revealing anything about ***m*** before the opening phase.

Commitments - Properties

- Security properties:
  - Hiding: A commitment ***c*** does not reveal anything about ***m***.
  - Binding: A commitment ***c*** to message ***m*** cannot be opened to a different $m' \neq m$
- Homomorphic Commitments:
  - An additively homomorphic commitment scheme is a commitment scheme such that given m_1, m_2, r_1, r_2 we have:

    $$\textbf{commit}(m_1, r_1) + \textbf{commit}(m_2, r_2) = \textbf{commit}(m_1+m_2; r_1+r_2)$$

    (Note: it could be the case that multiplying two commitments produces a commitment to the addition of the two messages. The above is just a symbolic way to represent the property.)

Hash Commitments

- Relies on the security of a collision-resistant hash functions.
- Pick a salt ***s***, then compute a commitment to ***m*** as (***H*** is a collision resistant hash function):
  - $c = H(m||s)$
- Hiding: inverting a hash is hard.
- Binding: opening a hash to another $m' \neq m$ requires finding a collision.

Pedersen Commitments

- Work in a cyclic group $G = \langle g \rangle$ ($\mathbf{g}$ is the generator) of order p in which the Discrete Log Problem is hard.
  - Given a $\mathbf{g}^x$ it is hard to find x.
- Choose two generators for the group; $\mathbf{g}$, $\mathbf{h}$
  - No one knows the discrete log of $\mathbf{g}$ with respect to $\mathbf{h}$ and vice versa.
- Commit to a message m:
  - Select a random r from $\{0, \ldots, p-1\}$.
  - Compute $c = \mathbf{g}^m\mathbf{h}^r$
- Open a commitment:
  - Reveal m and r

Pedersen Commitments - Security

- Hiding: $\mathbf{h}^r$ is a random element in the group ***G*** and so is the commitment $\mathbf{g}^m\mathbf{h}^r$
  - A random commitment value does not reveal anything about ***m***
- Binding: reduced to DLP,
  - An attacker who can open a commitment to $m' \neq m$ can be used to construct an attacker to break DLP.

Pedersen Commitments - Additively Homomorphic

- Given $c_1 = g^{m1}h^{r1}$ and $c_2 = g^{m2}h^{r2}$
  - $\textbf{commit}(m_1+m_2, r_1+r_2) = c_1c_2 = g^{(m1+m2)}h^{(r1+r2)}$
- Very useful for tracking balances of accounts in a private way, or total of inputs for a transaction.
  - I own x coins, stored in a hidden way on the blockchain
    - as a commitment: $g^x g^{r1}$
  - Bob sent me y coins, also in a hidden way as a commitment: $g^y h^{r2}$
  - The miners can update my account without revealing anything.
    - Simply multiply the two commitments together.
    - More like performing arithmetic operations on hidden (secret) data.

Zero Knowledge Proofs

- Second set of slides.

Schnorr's Protocol

- A proof of a knowledge of a discrete logarithm
  - Such as the secret key corresponding to some public key in EC.
- Work in a cyclic group $G = \langle g \rangle$ (g is the generator) of order p in which the Discrete Log Problem is hard.
- To prove $y = g^x$, prove knowing x as follows:
  - First round: prover chooses r in $\{1, \ldots, p-1\}$, prover commits to r by sending $c = g^r$ to the verifier.
  - Second round: verifier replies with a challenge t (selected at random from $\{1, \ldots, p-1\}$).
  - Third round: prover sends $s = r + tx \pmod p$
- Verifier accepts if $g^s \equiv cy^t$
- Usually called a Sigma protocol.
  - Since it consists of three rounds or moves: commit/challenge/respond

Fiat-Shamir Heuristic

- Converts an interactive protocol into a non-interactive one.
  - The prover computes a proof, sends it to the verifier, and the verifier checks the proof
    - One round of communication.
- Works in the random oracle model.
  - Instead of waiting for a random challenge from the verifier, the prover computes it using a random oracle (usually using a hash function modeled as a random oracle).
- For example, in Schnorr's protocol:
  - Instead of having the verifier choose a random challenge ***t***, the prover computes ***t = H(g, y, c)***
  - Then, the verifier sends both ***c*** and ***t*** to the prover to check.

- Zerocoin -
A Decentralized Mixer

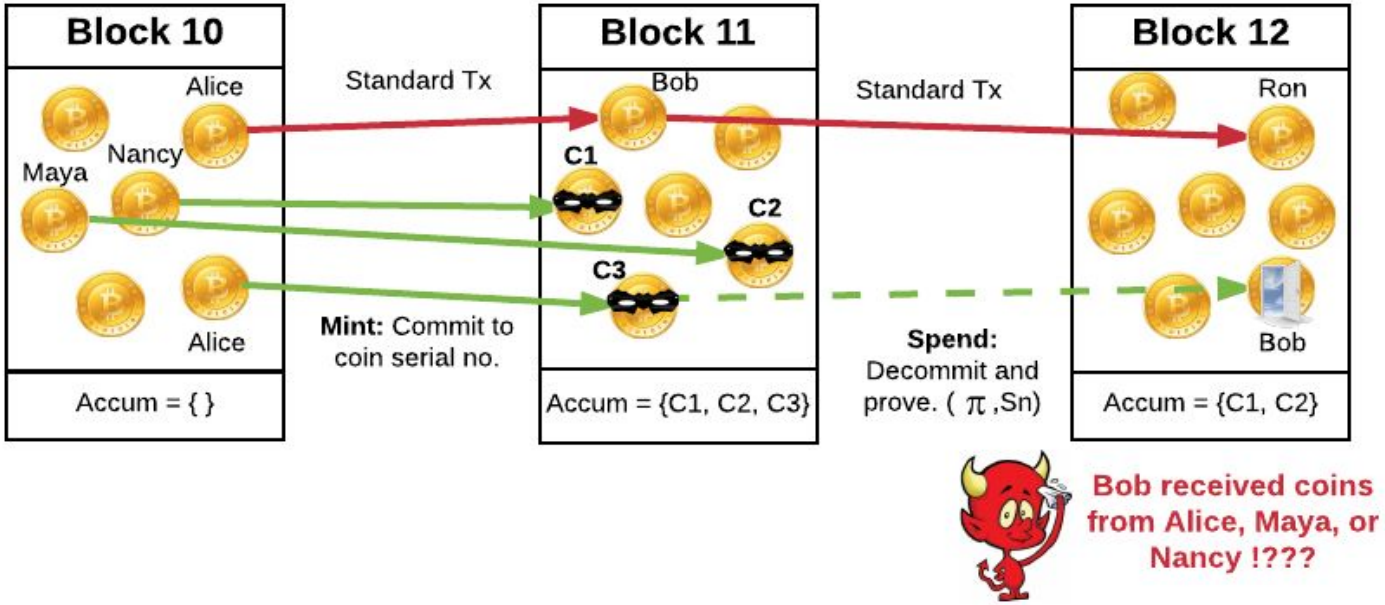
Zerocoin [Miers et al., 2013]

- An extension to Bitcoin to support anonymity.
- Turn a bitcoin into a zercoin.
  - This will create a pool of zercoins or an anonymity set.
- Then a zercoin can be sent to a new address.
  - Go back to bitcoin.
- Thus, it creates a decentralized mixer without any trusted entity.
  - Users have full control of their coins.
- The larger the pool, aka anonymity set, the greater the anonymity level.

Zerocoin

- General idea:
  - Creates an anonymity set, a pool of hidden coins in the form of commitments.
    - Clients **mint** anonymous coins.
    - The coin is simply a random serial number.
  - Spending a coin from that pool does not reveal the owner.
    - Reveal the serial number, and
      - Needed to prevent double spending.
    - provide a ZKP that a coin with a specific serial number belongs to the pool.
- Why is it anonymous?
  - ZKP does not provide any info beyond there exists a zercoin with the revealed serial number on the blockchain.

Zerocoin Pictorially



Zerocoin [Miers et al., 2013]

- Protocol:
  - Use an efficient RSA one way accumulator.
    - Accumulate all Zerocoins C1, …, CN into a short value called an accumulator A.
  - To spend a coin C, prove the knowledge of a witness such that C \in A and that C opens to serial number SN.
  - Pour: Spend a zerocoin.
- Disadvantages:
  - Only hides the originator, but not the destination or the amount,
  - computationally heavy.
    - A proof has a size of almost 25 Kb

- Zerocash -
Private/Anonymous Payments
(UTXO Model)

- Zether -
Private/Anonymous Payments
(Account Model)

References

- [Reid et al. 2014] Reid, Fergal, and Martin Harrigan. "An analysis of anonymity in the bitcoin system." In Security and privacy in social networks, 2013.
- [Koshy et al. 2014] Koshy, Philip, Diana Koshy, and Patrick McDaniel. "An analysis of anonymity in bitcoin using p2p network traffic." In Financial Cryptography, 2014.
- [Bonneau et al., 2014] Bonneau, Joseph, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. "Mixcoin: Anonymity for Bitcoin with accountable mixes." In Financial Cryptography, 2014.
- [Miers et al., 2013] Miers, Ian, Christina Garman, Matthew Green, and Aviel D. Rubin. "Zerocoin: Anonymous distributed e-cash from bitcoin." In IEEE S&P, 2013.
- [Ben Sasson et al., 2014] Sasson, Eli Ben, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. "Zerocash: Decentralized anonymous payments from bitcoin." In IEEE S&P, 2014.