UNIVERSITY of NICOSIA

Session 9

Blockchain Interoperability

BLOC 514: Emerging Topics in Blockchain and Digital Currency

Session Objectives

- Explain the need for cross-chain transactions
- Introduce a series of basic strategies for achieving blockchain interoperability
- Present a number of indicative use cases related to blockchain interoperability
- Present drawbacks and potential risks

The plethora of blockchains raises the need for supporting cross-chain transactions. In this session, a number of proposed approaches will be presented. Also, we will see that those approaches are characterized by some drawbacks and potential operational irregularities.

Agenda Slide

- Introduction
- Basic strategies of blockchain interoperability
- Use cases
- Pegged sidechains
- Two-way peg chains
- Notary schemes / relays / atomic swaps / hash-locking
- Possible failures
- Bridges
- Conclusions
- Resources

Introduction

- In the early years of blockchains, the idea having of one blockchain dominating the rest was present
    - Now, this idea seems unrealistic
    - Plethora of different blockchains across different domains/industries
- In such landscape: interoperability of different blockchains
    - How interoperability is defined?
    - Need to preserve the fundamental design principles of blockchains
- Definition of interoperability
    - Generic: the capability of a system to function with other systems
    - Software engineering: different systems can exchange information (data)
        - Common data formats, protocols, etc.
        - Example, Java-based programs run (almost) everywhere

Basic strategies of blockchain interoperability

Assume two blockchains, namely A and B

- Centralized / multisig notary schemes
  - A party enable an action to be executed on B when an event occurs in A
- Sidechains/relays
  - Sub-systems of A monitor and validate events that take place in B
- Hash-locking
  - Events that occur both in A and B are invoked by the same trigger (i.e., common cause)

Potential use cases of interoperability

- Portable assets
  - Ability to move coins between different blockchains
- Payment-versus-payment or payment-versus-delivery
  - Assume two users, U1 and U2, and their asset bundles S1 and S2, respectively. Both users have accounts in blockchains A and B, while S1 and S2 are stored in different blockchains.
  - U1 can transfer S1 to U2 if and only if U2 transfers S2 to U1 (i.e., both transfers should take place)
  - Equivalent to logical AND operation: 1 x 1 = 1 (unlike 0 x 1 = 0 x 1 = 0 x 0 = 0)
  - Also referred to as atomic swaps
- Cross-chain oracles
  - An action occurs on blockchain A given that an identity oracle on blockchain B provide a pre-determined proof about the address that is associated with the action
- Asset encumbrance
  - The locking/unlocking conditions applied over assets on blockchain A depend on actions that occur on blockchain

Variations and combinations of the above, however, the first two cases have attracted greater interest

Pegged sidechains: definitions

- Sidechain
  - A blockchain being able to validate data provided by other blockchains.
- Two-way peg
  - A mechanism that enables the bidirectional exchange of assets between sidechains according to a fixed (or deterministic) exchange rate.
- Pegged sidechain
  - A sidechain that supports the two-way peg mechanism.
- Reorganization
  - A situation that takes place when an accepted chain, C1, is surpassed by another chain, C2, due to more proof-of-work. As a result the blocks of C1 are eliminated from the consensus history.
- Simplified payment verification proof (SPV proof)
  - A proof that an event occurred. It is signed by a number of signers.

Pegged sidechains: desired properties

- Ability to "return back"
  - Assets should be able to return back to the initial blockchain
- No counterparty risk
  - No asset transfer by dishonest parties
- Atomic transfers
  - Transfers are fully executed, otherwise nothing takes place (i.e., no partial transfers)
- Firewalled sidechains
  - A failure in a sidechain should not affect other sidechains
- Local settlement of reorganization
  - Any blockchain reorganizations should affect other sidechains

Symmetric two-way peg

- Process when transferring assets from a parent chain to a sidechain
  - The assets are moved to a special output of the parent chain where they are locked.
  - The locked assets can be unlocked by an SPV proof of possession occurring on the sidechain.
  - The synchronization of parent chain and side chain in achieved through the utilization of two waiting periods, namely, confirmation and contest period
- Confirmation period
  - Functionality: Assets locked in parent chain before being transferred to sidechain
  - Purpose: Prevent DoS attacks during the next waiting period
- Contest period
  - Functionality: Assets just transferred in the sidechain can not be spent
  - Purpose: Avoid double-spending problems
- The waiting period can be regarded as security parameter
  - Trade-off between speed and security

Bridges

- Motivated by physical world
- In the physical world bridges connect separated territories (e.g., islands)

- Here, bridges connect separated blockchains
- What 'connection' means:
  - Transfer of assets
  - In general, transfer of information
- Examples:
  - In the physical world: Currency exchange as moving across different countries
  - Here: ETH from/to Mainnet to/from Arbitrum
- Additional benefits:
  - For end users: Utilization of various blockchains in the context of DApps
  - For developers: Collaboration across different platforms and teams (among other, innovation is fostered)

Adapted from https://ethereum.org/en/bridges/

Bridges

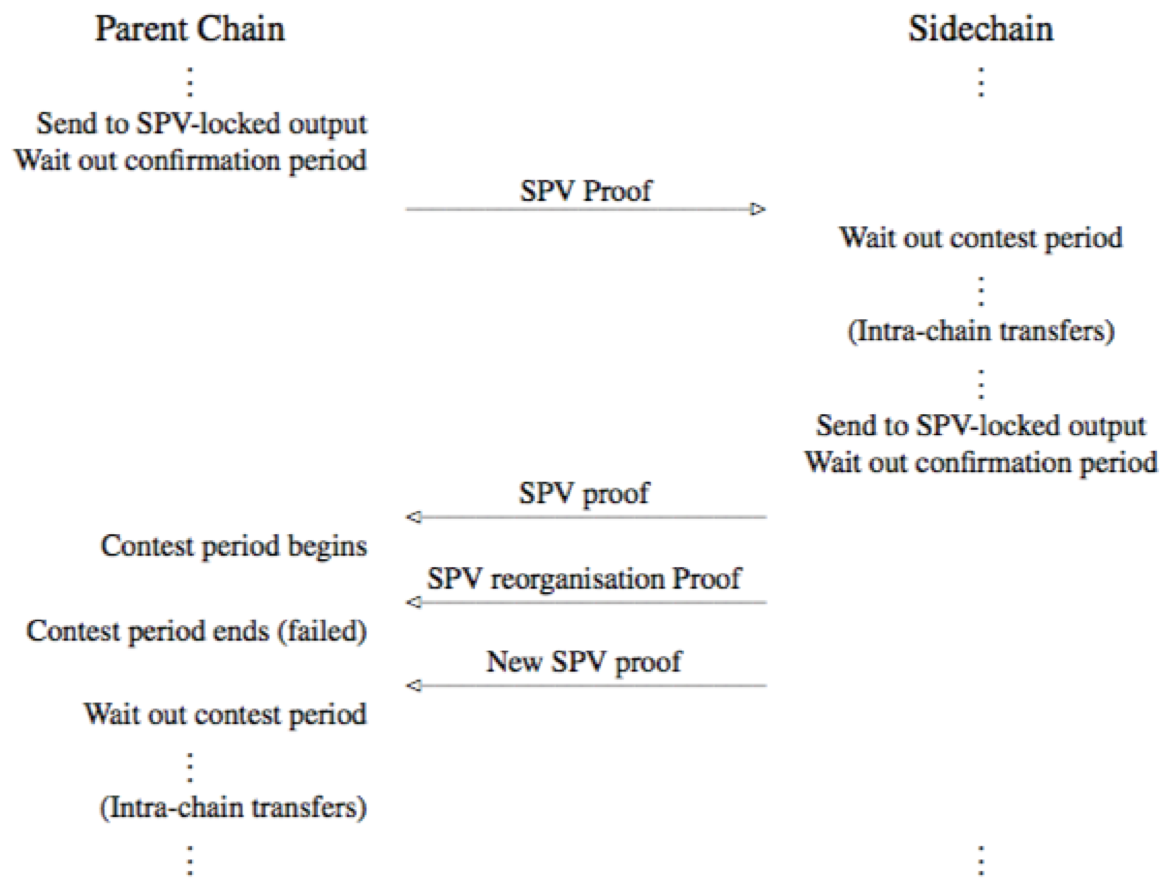
- Two main types of bridges

- Centralized:
  - Operationally: dependence of 3rd party machinery
  - Assumptions regarding
    - Funds custody
    - Security mechanisms
  - Users do not directly, exclusively control their funds

- Decentralized:
  - Algorithmic operation (also involving smart contracts)
  - On-chain trust, i.e., through the respective blockchain protocol
  - Users do not directly, exclusively control their funds

Adapted from https://ethereum.org/en/bridges/

Symmetric two-way peg



Drawbacks

- Complexity at different levels
  - Network level: Need to synchronize the transfers between independent blockchains
  - Asset level: Arbitrarily multiple assets
- Fraudulent transfers: Through the manipulation of the contest period during asset transfers
  - Solution 1: Increase the contest period
  - Solution 2: Contest period as a function of the blockchains' hashpowers
- Centralization of mining
  - Unlike strong miners, small-scale miners can not work for every blockchain
- Soft –forks: The isolation of the sidechains is relaxed
  - Stricter soft-forking rules may be established
  - Example: both blockchains may require the examination of each other's chain

Notary schemes

- Notary schemes constitute the easiest way for implementing cross-chain operations
    - Notary schemes utilize notary mechanisms that rely on a trusted entity (or more)
- Trusted entities can
    - Claim to blockchain A that an event occurred on blockchain B, or
    - Claim to blockchain A that a specific claim regarding blockchain B is accurate
- Trusted entities exhibit two broad operational modes
    - Active: Monitor the occurrence of events, automatic event-triggered actions
    - Reactive: Actions are invoked when the entities are explicitly asked to do so
- Example: The Interledger project (https://interledger.org)
    - Basic idea: An open protocol enabling the transfer of protocols across different blockchains similarly to the Internet routing systems.
    - Invention of the Interledger protocol by Ripple; development by Interledger W3C Community Group (https://www.w3.org/community/interledger/)

Relays

- Relays can be regarded as a direct way for achieving interoperability
  - The exploitation of trusted parties (intermediaries) is eliminated
  - How: the validation of the required events is performed by the blockchains themselves
- Assume two blockchains A and B
  - Hypothesize that the notion of "block headers" is applicable in A (similarly to Bitcoin, etc)
  - Suppose that B aims to find out whether:
    - An event has occurred in A, or
    - A certain value is contained in the state of A
  - A contract is created on B that takes as input the appropriate headers of A
    - First, the headers are verified according to A's consensus algorithm
    - Then, the desired info (events in A, A's states, etc) is verified
- Example: BTC Relay (http://btcrelay.org)
  - Basic idea: Enables the verification of Bitcoin transactions through Ethereum smart contracts.
  - Application: Use of Ethereum-based DApps via Bitcoin payments

Relays for atomic swaps

- Basic idea: Exchange assets in blockchain A for assets in blockchain B
- Currently, technical challenges due to possibility of attacks based on race conditions
  - Race conditions: When two (or more) processes access shared resources trying to commit changes concurrently
- Use case: Assume that User 1 wishes to exchange 10 ETH for 1 BTC
  - User 1 puts the ETH amount into a contract
  - Contract: "I will transfer 10 ETH to the party that is able to prove the transfer of 1 BTC to address X"
  - Suppose that User 2 transfers 1 BTC to address X
  - User 1 may attempt to transfer 1 BTC to the same address
    - If User's 1 BTC arrives first, User 2 is left with nothing!
- Solution: Involved blockchains should support Ethereum-like capabilities
  - Example: Maker DAO (https://github.com/makerdao/btc-market)

Hash-locking

- Hash-locking does not require blockchains to share much information about their state
- In notary schemes: hash-locking eliminates the demand for trust among notaries
- Assume two blockchains, A and B:
    - Step 1: Secret S is created on blockchain A and the hash of it, H=Hash(S), is sent to B.
    - Step 2: Both blockchains A and B lock their assets in the context of a smart contract.
        - Blockchain A locks first the asset, while B does the same after verifying A's lock
        - On blockchain A: the asset is sent to B if secret S is provided within 2 x TIME (TIME is a system parameter, e.g., in seconds). Otherwise, the asset is returned to A.
        - On blockchain B: the asset is sent to A if S is provided with TIME. Otherwise, the asset is returned to B.
- Note that:
    - The fact that S is revealed by A within TIME (in order to claim B's asset), enables B to become aware of S (thus, B can claim A's asset)

Interoperability: possible failures

- Interoperability-related proposal are based on the assumption that the involved blockchains operate normally
- However, a series of irregularities/failures may occur in one (or both) blockchains
- Examples:
  - 51% attacks that can cause the reversion of the transactions
  - 51% attacks that can generate invalid chains
  - Soft forks that can change the functionality
  - Hard forks where all (or the majority of) nodes migrate to a new blockchain
- Cross-chain application should include a failure handler for addressing such failures
  - This constitutes an open (and challenging) research area
  - A possible direction of is the design and development based on cross-chain programming languages

Indicative projects

- **Polkadot**
    - Aim: Enable cross-chain transfers applicable to various data/assets
    - Use of Nominated Proof-of-Stake
    - Governance token: DOT
    - White paper

- **Cosmos**
    - Aim: Interconnection of numerous blockchains, thus, creating an ecosystem of blockchains
    - Use of Proof-of-Stake
    - Governance token: ATOM
    - White paper

Conclusions

Conclusions

- The presence of various blockchains is a reality
  - Not a single blockchain is expected to rule out the rest
  - This fact enables the creation of numerous use cases
  - Applications that support cross-chain transaction constitute a natural result of this reality
- Blockchain interoperability: a number of approaches have been proposed
  - Each approach depends on the characteristics of the constituent blockchains
  - Complexity issues
- Future direction
  - The research on this area is expected to be active in the short-term future since a number of technical challenges remain open
  - A possible solution for tackling such challenges is the development of a programmable layer of communication that can be incorporated between the chains and the end applications
    - A step towards the development of standards

Bibliography

References

- Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra,Jorge Timón, and Pieter Wuille. **"Enabling Blockchain Innovations with Pegged Sidechains"**
  - Section 1 – 4
  - PDF

Additional Bibliography (optional)

- Rafael Belchior et al. **"A Survey on Blockchain Interoperability: Past, Present, and Future Trends"**
  - PDF
- Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. **"Bitcoin and cryptocurrency technologies"**, Princeton University Press
  - Section 10.5 – 10.6
  - Website
- Johnny Dilley, Andrew Poelstra, Jonathan Wilkins, Marta Piekarska, Ben Gorlick, and Mark Friedenbach. **"Strong Federations: An Interoperable Blockchain Solution to Centralized Third-Party Risks"**
  - PDF

UNIVERSITY of NICOSIA

Instructor’s Email:
iosif.e@unic.ac.cy