Session 1 - summary

Topic: Blockchain for Critical Infrastructure

Speakers:

- Talk1: Prof Salil Kanhere, UNSW, Australia, BC for CPS
- Talk 2: Dr Anh Dinh, Deakin University, Australia, BC and DB

- Session chair/summary: Dan Kim, University of Queensland, Australia

Talk 1: Blockchain for Cyber-Physical Systems

- Introduction
  - Security is a great challenge (e.g., Mirai botnet) to CPS
  - Establishing trust can be difficult
  - A lot of challenges facing CPS (e.g., heterogeneity in device resources, multiple attack surfaces)
- Salient Features of Blockchain can provide benefit to CPS and other areas.
  - e.g., tamper-proof storage of information
- Focusing on Supply Chain – a system of organizations, people, ..
  - A lot of concerns on Traceability (e.g., counterfeiting, needles in strawberries in Australia)
  - Current traceability systems (sliced, unreliability of data, …)

Talk 1 - summary

- Four proposed ideas

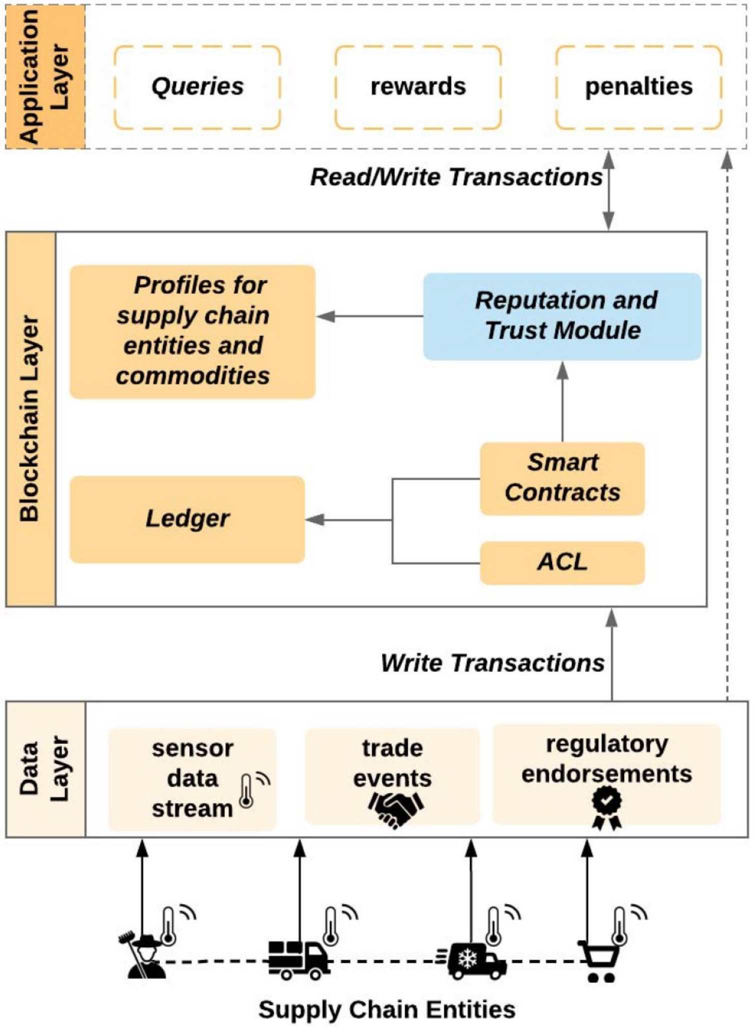
1. ProductChain
2. TrustChain
3. PrivChain
4. TradeChain

Talk 1: ProductChain [IEEE NCA’18]

- Challenges: integrity and traceability (in Food Supply Chain)
- A Holistic approach, consortium to manage a permissioned blockchain (BC)
- Transaction vocabulary,
- A Tiered Architecture
  - Data layer, storage layer, blockchain layer, application layer
- Access Control List collectively managed by consortium members; read and write access

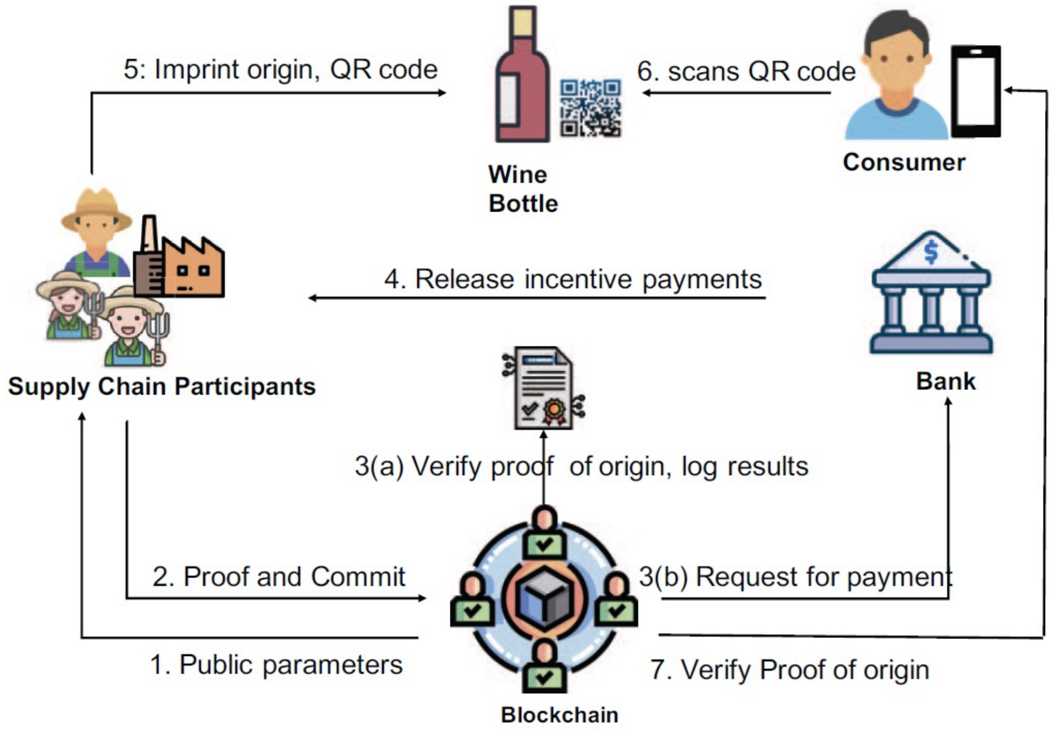
Talk 1: TrustChain [IEEE Blochain’19]

- Challenges: trust and reliability of the data
  - How do we trust data written into the blockchain?
  - Need for a trust management system with the some requirements
    - e.g., Multi-faceted assessment of trustworthiness of logged data in BC which incorporates inputs from IoT sensors, feedback provided by supply chain entities, physical audits, etc.
- Contributions
  - BC-based reputation & trust framework – commodity reputation (sensor data), participant reputation (buyer feedback) in blockchain layer [ICBC’22]
  - Smart contracts for automation of reputation calculation
  - Accountability mechanisms
  - Hyperledger fabric implementation
  - Minimal overheads in terms of throughput and latency



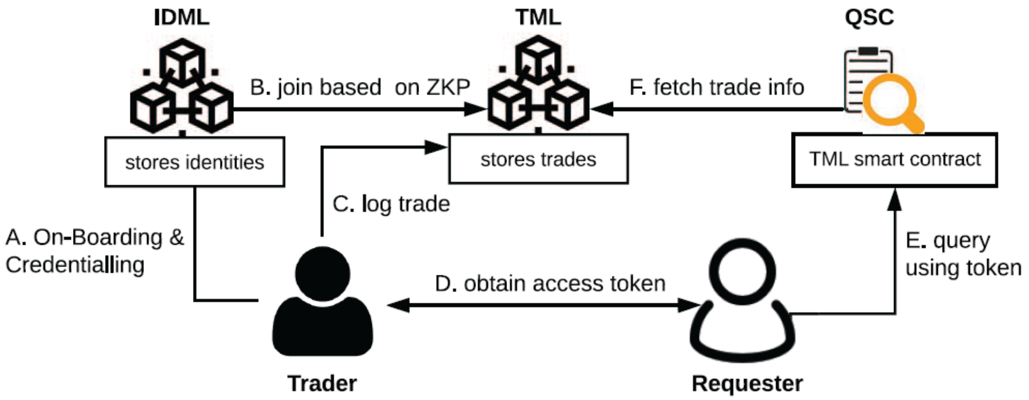
Talk 1: PrivChain [IEEE Blochain'22]

- Challenges: Traceability vs privacy
- Contributions:
  - Zero-knowledge Proof (ZKP) based privacy preservation
  - Automated verification using smart contract
  - Implemented the framework on Hyperledger fabric
- Supply chain participants can provide ZKP proofs and get reciprocated by the committed incentive amounts for utilizing their resources.
  - participants share proofs of their valid data pertaining to products.
  - The verification of such proofs is then automated by a blockchain smart contract.
- The blockchain can verify these proofs, initiate an off-chain payment mechanism and log the results in an immutable way.



Talk 1: TradeChain [IEEE TrustCom21]

- Challenge: Identity privacy
  - Permissioned blockchain -> Identities
- Contributions
  - ensuring privacy through keeping the identities private.
  - Integrated framework for two separate ledgers: a) a public permissioned blockchain for maintaining identities and b) the permissioned blockchain for recording trade flows
  - uses Zero Knowledge Proofs (ZKPs) on traders' private credentials to prove multiple identities on trade ledger
  - allows data owners to define dynamic access rules for verifying traceability information from the trade ledger using access tokens and Ciphertext Policy Attribute-Based Encryption (CP-ABE)
- Three key components
  - Identity Management Ledger (IDML) – a public permissioned blockchain for managing decentralised identifiers (DIDs), based on Sovereign Identity Design
  - Trade Management Ledger (TML) – a permissioned blockchain for recording supply chain events
  - Query Smart Contract (QSC)



Talk 2 - summary

1. TAP: Transparent and Privacy-Preserving Data Services [USENIX Security 2023 summer]

2. GlassDB: An Efficient Verifiable Ledger Database System Through Transparency [CoRR, July 2022]

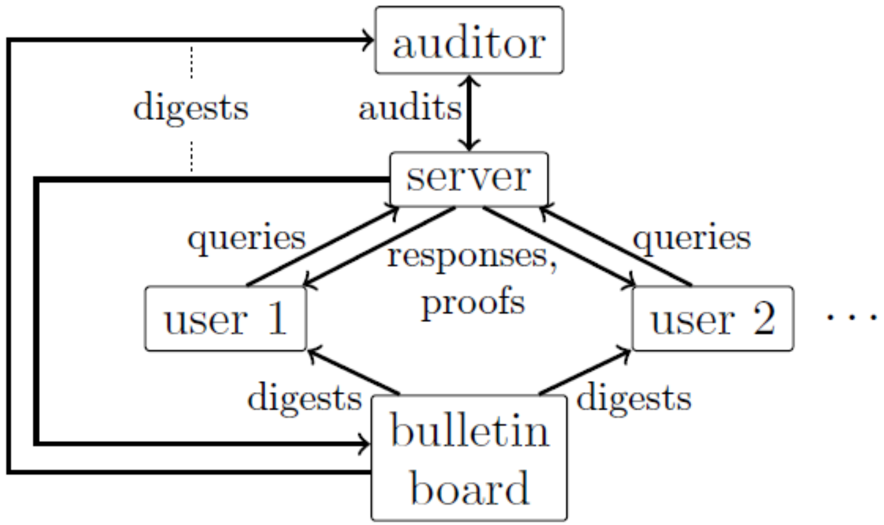
Talk 2: Blockchain and database – a math made in the Cloud

- Observation (system model)
  - Settings:
    - Some data involved multiple users
    - Computation on the data
    - Outsourced to untrusted servers
  - Examples: blockchains, key management
- Solutions:
  - The blockchain way:
    - A consensus ensures that bad thing do not happen (given some assumption)
  - The certificate transparency way:
    - Servers made accountable via auditing: delete bad things after the fact

Talk2: TAP: Transparent and Privacy-Preserving Data Services [Security 2023]

Entities:

- Users.
  - send data to the server and issue queries on the aggregate data through a client.
  - Each user monitors the data structure by verifying that her data is properly stored by the server and verifies that query results are computed correctly.
- Server.
  - stores the data provided by the users in a database, and maintains an ADS on top of the data.
  - computes responses to user queries, and generates proofs for the responses using the ADS.
- Auditor.
  - validates the server's ADS.
- Bulletin board.
  - The server periodically publishes the digest of its ADS to an immutable bulletin board, e.g., a public blockchain.
  - Users and auditors download the latest digests during monitoring, auditing, and query verification.



TAP's system model with an untrusted server

Talk2: TAP (cont.)

- Challenge: transparency
  - the service's processing of the data is verifiable by users and trusted auditors.
- Goal: build a multi-user system that provides data privacy, integrity, and transparency for a large number of operations, while achieving practical performance.
- Proposed ideas: a novel tree data structure (authenticated data structure) that supports efficient result verification, and relies on independent audits that use zero-knowledge range proofs.
  - TAP combines a chronological prefix tree with sorted sum trees whose roots are stored in the prefix tree leaves.
  - TAP supports a broad range of verifiable operations (e.g., sum/average/count, min/max, quantiles and sample standard deviations.
- Applications: Smart Grids (dynamic pricing), congesting pricing (e.g., based on the number of cars in CBD), advertising.

Talk2: TAP (cont.)

- Application of transparency model: Dynamic pricing
- Retailer's cost is lowest if the total demand spread out over the day, retailer wants consumer to shift loads to low-demand period e.g., smart meters: fine-grained tracing
- Goals:
  - Transparency: retailer cannot exaggerates beyond a bound
  - Privacy: it does not reveal data to curious consumers
- Approach
  - Building blocks: commitments, ZK range proofs
  - Baseline:
    - Retailer computes C for all data and sums (C – additive HE)
    - Retailer computes range proofs
  - Merkle tree based solution
    - Retailer builds Merkle tree on commitments
    - Sends inclusion proofs to consumer
    - Consumer verifies proofs
    - Auditor checks all range proofs

Talk 2: GlassDB - Practical Verifiable Ledger Database Through Transparency

- Ledger DB
  - maintains a history of operations
  - Integrity: server cannot tamper with the result
  - Append-only: server cannot change the history of operations (i.e., the database server cannot fork the history log without being detected)
- Existing systems' limitations: the lack of transaction support and the inferior efficiency
- Verifiable ledger DB
  - protects the integrity of user data and query execution on untrusted database providers.
  - An example - blockchain protects the integrity of the log against Byzantine attackers, by running a distributed consensus protocol among the participants.

Talk 2 – GlassDB (cont.)

- Three challenges
  1. the lack of a unified framework for comparing verifiable ledger databases
  2. the lack of database abstraction, that is, transactions
  3. how to achieve high performance while retaining security

- Proposed approach

  1. Establishing the design space consisting of 3D: abstraction, threat model, & performance.

  2&3. Designing and implementing GlassDB:

  - supports distributed transactions and has efficient proof sizes
  - achieves high throughput by building on top of a novel data structure: a two-level Merkle-like tree

Some improvements (my thought)

- Threat model
  - Assumed that attackers cannot mount denial of service attacks? If this does not hold?
  - What are fault and security threats to Verified Ledger DB?
- Performance metrics:
  - It used two metrics: user’s verification cost and database throughput?
- More analysis on failure recovery
  - One node crash was used.
  - Multiple nodes failures? Recovery time is longer, …
  - Under varying workload models?