B-IoT: Blockchain Driven Internet of Things with Credit-Based Consensus Mechanism

ICDCS 2019, Dallas, Texas

Junqin Huang, Linghe Kong, Guihai Chen, Long Cheng, Kaishun Wu and Xue Liu

Shanghai Jiao Tong University
Internet of Things Systems

Transportation  Healthcare  Industrial/Manufacturing Field
Internet of Things Systems

IoT smart objects are expected to reach 212 billion entities deployed globally by the end of 2020.
Open Issues in IoT Systems

• Single point of failure [1]
• Malicious attacks such as DDoS, Sybil attack [2], [3]
• Data disclosure & credibility [4]
• System scalability [5]

Combine Blockchain with IoT?

• Why Blockchain in IoT
  • non-manipulated source of data
  • break down monolithic data silos and enable trust across parties

• Related Work
  • A scalable access management system in IoT [IOTJ’18]
    • vulnerable to the single point failure and attacks
  • Consortium blockchain for secure energy trading in IIoT [TII’18]
    • data disclosure risk
  • A blockchain platform for clinical trial and precision medicine [ICDCS’17]
    • stuck in the concept stage
  • Integrating low power IoT devices to a blockchain-based infrastructure [EMSOFT’17]
    • bring too much overloads in IoT systems
Main Challenges

• The conflicts between high concurrency and low throughput

• The trade-off between efficiency and security

• The coexistence of transparency and privacy
Main Challenges

• The conflicts between high concurrency and low throughput
  • We explore a DAG-structured blockchain based solution
• The trade-off between efficiency and security
• The coexistence of transparency and privacy
Blockchains

• Distributed ledgers or databases that enable parties which do not fully trust each other to form and maintain consensus

Chain-structured blockchain
(bitcoin, Ethereum, Hyperledger, etc.)

Directed acyclic graph (DAG)-structured blockchain
(IOTA, Byteball, NANO, etc.)
Blockchains

- Distributed ledgers or databases that enable parties which do not fully trust each other to form and maintain consensus

DAG-structured blockchains have a higher throughput than chain-structured blockchains
B-IoT: System Overview

• **Node type:**
  • Light nodes
  • Full nodes

• **A case study of smart factory:**
  • Wireless sensors
  • Gateways
  • Manager
  • Tangle network
B-IoT: System Overview

• **Node type:**
  • Light nodes
  • Full nodes

• **A case study of smart factory:**
  • Wireless sensors
  • Gateways
  • Manager
  • Tangle network
B-IoT: System Overview

- **Node type:**
  - Light nodes
  - Full nodes

- **A case study of smart factory:**
  - Wireless sensors
  - Gateways
  - Manager
  - Tangle network
B-IoT: System Overview

• Node type:
  • Light nodes
  • Full nodes

• A case study of smart factory:
  • Wireless sensors
  • Gateways
  • Manager
  • Tangle network
Main Challenges

• The conflicts between high concurrency and low throughput
  • We explore a DAG-structured blockchain based solution

• The trade-off between efficiency and security

• The coexistence of transparency and privacy
Main Challenges

• The conflicts between high concurrency and low throughput
  • We explore a DAG-structured blockchain based solution

• The trade-off between efficiency and security
  • We design a moderate-cost credit-based PoW mechanism

• The coexistence of transparency and privacy
Tuning the difficulty of PoW algorithm

- Less than the target hash value, i.e. the length of prefix zero
- E.g. hash space is 0x00000000~0xffffffff

![Diagram showing hash space with different prefixes]
Credit-Based PoW Mechanism

Positive Component + Negative Component = Credit Value
Credit-Based PoW Mechanism

Positive Component + Negative Component = Credit Value

\[ Cr_i^P = \sum_{k=1}^{n_i} \frac{w_k}{\Delta T} \]
Credit-Based PoW Mechanism

\[ Cr_i^P = \sum_{k=1}^{n_i} \frac{w_k}{\Delta T} \]
\[ Cr_i^N = -\sum_{k=1}^{m_i} \alpha(B) \cdot \frac{\Delta T}{t - t_k} \]
Credit-Based PoW Mechanism

\[ C_{r_i}^P = \sum_{k=1}^{n_i} \frac{w_k}{\Delta T} \]

\[ C_{r_i}^N = -\sum_{k=1}^{m_i} \alpha(B) \cdot \frac{\Delta T}{t - t_k} \]
Malicious Behaviours

• Double-spending
• Lazy-tips

$$\alpha(\mathcal{B}) = \begin{cases} 
\alpha_l & \text{if } \mathcal{B} \text{ is lazy tips behaviour;} \\
\alpha_d & \text{if } \mathcal{B} \text{ is double-spending behaviour,}
\end{cases}$$
Malicious Behaviours

• Double-spending
• Lazy-tips

\[ \alpha(\mathcal{B}) = \begin{cases} 
\alpha_l & \text{if } \mathcal{B} \text{ is lazy tips behaviour;} \\
\alpha_d & \text{if } \mathcal{B} \text{ is double-spending behaviour,} 
\end{cases} \]
Malicious Behaviours

• Double-spending
• Lazy-tips

\[ \alpha(B) = \begin{cases} 
\alpha_l & \text{if } B \text{ is lazy tips behaviour;} \\
\alpha_d & \text{if } B \text{ is double-spending behaviour,}
\end{cases} \]
Malicious Behaviours

• Double-spending
• Lazy-tips

\[ \alpha(B) = \begin{cases} \alpha_l & \text{if } B \text{ is lazy tips behaviour;} \\ \alpha_d & \text{if } B \text{ is double-spending behaviour,} \end{cases} \]
Credit-Based PoW Mechanism

Positive Component

$C r_i^P = \sum_{k=1}^{n_i} \frac{w_k}{\Delta T}$

Negative Component

$C r_i^N = - \sum_{k=1}^{m_i} \alpha(B) \cdot \frac{\Delta T}{t - t_k}$

Credit Value

$C r_i = \lambda_1 C r_i^P + \lambda_2 C r_i^N$
Main Challenges

• The conflicts between high concurrency and low throughput
  • We explore a DAG-structured blockchain based solution

• The trade-off between efficiency and security
  • We design a moderate-cost credit-based PoW mechanism

• The coexistence of transparency and privacy
Main Challenges

• The conflicts between high concurrency and low throughput
  • We explore a DAG-structured blockchain based solution

• The trade-off between efficiency and security
  • We design a moderate-cost credit-based PoW mechanism

• The coexistence of transparency and privacy
  • We propose an efficient data authority management method
Data Authority Management Method

\[ M_1 = \text{Enc}_{PK_D}(\text{sign}_{SK_u}(SK_S, TS_1, \text{nonce}_u)) \]
\[ M_2 = \text{Enc}_{SK_1}(\text{sign}_{SK_b}(\text{nonce}_b, TS_2, \text{nonce}_a)) \]
\[ M_3 = \text{Enc}_{SK_2}(\text{sign}_{SK_u} (\text{nonce}_b, TS_3)) \]
\[ \text{Dec}_{SK_D}(M_1) \]
\[ \text{Dec}_{SK_1}(M_2) \]
\[ \text{Dec}_{SK_2}(M_3) \]
\[ \text{Dec}_{SK_b}(M_3) \]

Generate symmetric secret key \( SK_S \)

Distribute the symmetric secret key without central trust server

IoT Device

Manager

\[ (PK_D, SK_D) \]

\[ (PK_M, SK_M) \]
Data Authority Management Method

\[ M_1 = Enc_{pk_D} \{ sign_{sk_u} (SK_s, TS_1, nonce_u) \} \]

\[ M_2 = Enc_{sk_a} \{ sign_{sk_b} (nonce_b, TS_2, nonce_a) \} \]

\[ M_3 = Enc_{sk_b} \{ sign_{sk_u} (nonce_b, TS_3) \} \]

\[ \text{nonce}_b \leftarrow Dec_{sk_b} (M_3) \]

Distribute the symmetric secret key without central trust server
Data Authority Management Method

\( (PK_D, SK_D) \)

IoT Device

\[ M_1 = Enc_{PK_D} \{ sign_{SK_u}(SK_S, TS, nonce_u) \} \]

\[ Dec_{SK_D}(M_1) \]

\( (PK_M, SK_M) \)

Manager

Generate symmetric secret key \( SK_S \)

\[ M_2 = Enc_{SK_D}(sign_{SK_D}(nonce_b, TS, nonce_u)) \]

\[ Dec_{SK_M}(M_2) \]

\[ M_3 = Enc_{SK_M}(sign_{SK_D}(nonce_b, TS)) \]

\( nonce_b \leftarrow Dec_{SK_M}(M_3) \)

Distribute the symmetric secret key without central trust server
Data Authority Management Method

\[(PK_D, SK_D)\]  
\[(PK_M, SK_M)\]

\[\text{IoT Device}\]

\[\text{Manager}\]

\[M_1 = Enc_{PK_D} \{\text{sign}_{SK_M}(SK_S, TS_1, nonce_a)\}\]

\[\text{Dec}_{SK_D}(M_1)\]

\[M_2 = Enc_{SK_S}(\text{sign}_{SK_P}(nonce_b, TS_2, nonce_a))\]

\[\text{Dec}_{SK_S}(M_2)\]

\[M_3 = Enc_{SK_B}(\text{sign}_{SK_B}(nonce_b, TS_3))\]

\[\text{Dec}_{SK_B}(M_3)\]

\[\text{nonce}_b \leftarrow \text{Dec}_{SK_B}(M_3)\]

Distribute the symmetric secret key without central trust server

Generate symmetric secret key \(SK_S\)
Data Authority Management Method

Distribute the symmetric secret key without the central trust server
Implementation

• Full nodes: manager & gateway
  • commercial computer
  • implemented based on IRI
  • SHA-256 & AES encryption

• Light nodes: IoT devices
  • Raspberry Pi Model 3B
  • implemented based on PyOTA
  • Extended with local PoW
  • AES encryption
Performance in Credit-Based PoW

When one malicious attack happens

When two malicious attacks happen
Performance in Credit-Based PoW

When one malicious attack happens

When two malicious attacks happen
Performance in Credit-Based PoW

It will take longer time to recover normal transaction rate if the node conducts malicious attacks twice or more
Performance in Credit-Based PoW

• Four control experiments:
  • PoW
  • Cr-PoW w/o malicious attacks
  • Cr-PoW with a malicious attack
  • Cr-PoW with two malicious attacks

![Average time per transaction (seconds)](chart)
Performance in Credit-Based PoW

- Four control experiments:
  - PoW
  - Cr-PoW w/o malicious attacks
  - Cr-PoW with a malicious attack
  - Cr-PoW with two malicious attacks
Performance in Credit-Based PoW

- Four control experiments:
  - PoW
  - Cr-PoW w/o malicious attacks
  - Cr-PoW with a malicious attack
  - Cr-PoW with two malicious attacks

Credit-based PoW can speed up transactions for honest nodes, also can defend malicious attacks efficiently
Efficiency of Data Authority Management
Efficiency of Data Authority Management

![Graph showing running time against message length (log₂ bytes)]
Efficiency of Data Authority Management

The data authority management method has tiny impact on the whole transaction process.
Conclusion & Thank you!

• A general DAG-structured blockchain-based IoT system to address aforementioned challenges for IoT
• The credit-based PoW mechanism helps to make the blockchain more suitable for IoT systems
• The data authority management method can protect data privacy without affecting the system performance
• Future directions:
  • sensor data quality control
  • storage limitations