

Building trust of Blockchain-based Internet-of-Thing services using public key infrastructure

Wattana Viriyasitavat, Li Da Xu, Assadaporn Sapsomboon, Gaurav Dhiman & Danupol Hoonsopon

To cite this article: Wattana Viriyasitavat, Li Da Xu, Assadaporn Sapsomboon, Gaurav Dhiman & Danupol Hoonsopon (2022): Building trust of Blockchain-based Internet-of-Thing services using public key infrastructure, Enterprise Information Systems, DOI: [10.1080/17517575.2022.2037162](https://doi.org/10.1080/17517575.2022.2037162)

To link to this article: <https://doi.org/10.1080/17517575.2022.2037162>



Published online: 24 Feb 2022.



Submit your article to this journal [↗](#)



Article views: 124




View related articles [↗](#)



View Crossmark data [↗](#)



Building trust of Blockchain-based Internet-of-Thing services using public key infrastructure

Wattana Viriyasitavat ^a, Li Da Xu^b, Assadaporn Sapsomboon^a, Gaurav Dhiman^{c,d} and Danupol Hoonsoon^e

^aBusiness Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok, Thailand; ^bDepartment of Information Technology & Decision Sciences, Old Dominion University, Norfolk, VA, USA; ^cDepartment of Computer Science, Government Bikram College of Commerce, Patiala, India; ^dUniversity Centre for Research & Development, Department of Computer Science and Engineering, Chandigarh University, Gharuan, Mohali, Punjab; ^eBusiness Information Technology Division, Department of Marketing, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok, Thailand

ABSTRACT

The advancement of hardware, software, and Internet infrastructure leads to the increasing quantities of smart Internet of Things (IoT) devices. Meanwhile, security issues have increasingly brought to us the concerns due to the evolving IoT scope and mass communications. Trusting service vendors depends on their devices that generate information and provide executions. Blockchain becomes an attractive choice, as evidenced by its wide adoptions. However, trusting IoT-based services becomes an important issue since the implementation of Blockchain-based IoT (BloT) services is proprietary and independent. This paper introduces a generic architecture design that incorporates Public Key Infrastructure (PKI) to establish trust of BloT services. This can potentially solve the trust problem and based on our experiment it can be scaled well. We also demonstrate how specification languages can be useful to express requirements. It decouples users from Blockchain and thus they can specify qualities of BloT services without deep knowledge to work with Blockchain.

ARTICLE HISTORY

Received 21 August 2021
Accepted 30 January 2022



KEYWORDS

Blockchain technology (BCT); Blockchain-based Internet of Things (BloT); service; public key infrastructure (PKI); trust; specification language

1. Introduction

In recent years, numerous studies have been conducted on Internet of Things (IoT) and Blockchain integration. Since IoT is now getting broader, many new security issues arise, whereas the existing ones are getting more intense. One security vulnerability might have a high impact on many devices and a simple attack can be very devastating.

Despite of a lot of security concerns, the benefits of IoT outweigh the challenging security problems (Wu et al. 2018). The future of IoT is expected to revolutionise our society. Complex applications begin to rely on the collaborations of multiple IoT devices that can deliver more personalised local services to users. Interconnecting tons of IoT

CONTACT Wattana Viriyasitavat  hardgolf@gmail.com; wattana@cbs.chula.ac.th  Business Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok, Thailand, 10300

devices raises many security issues. IoT is not secure-by-design. The interoperation of IoT-based services leverages the risks of security attacks. Many attempts that apply traditional security mechanisms to IoT are facing difficulties and limitations to handle high-volume communications, decentralisation and scarcity of resources, which in many cases makes complex operations such as encryption impossible (Viriyasitavat, Anuphaptrirong, and Hoonsopon 2019; Wu et al. 2018). Other common security incidents found in IoT include privacy, insecure interface, unencrypted communications and physical security (Viriyasitavat, Anuphaptrirong, and Hoonsopon 2019). Therefore, these devices are more vulnerable to attacks than endpoint devices such as smartphones, tablets or computers (Khan and Salah 2018).

Even several doubts to the real benefits of Blockchain, this technology has been proven as evidenced by its wide applications in several domains (Perera et al. 2020). The integration of Blockchain with IoT has been studied intensively in recent years, mostly focusing on trust of IoT services (Huang et al. 2020). Several designs use Blockchain as a key enabler to provide a wide range of measures to strengthen security based on the following characteristics (Viriyasitavat et al. 2019a): (1) the chain of blocks offers immutability of historical data, and (2) every data item must be verified before being included in Blockchain (Zhang and Zhou 2020). The aspects in which Blockchain is considered as a potential solution, or at least part of a solution, for IoT security include Digital Identity and Access Management (IAM), lightweight secure communication, authentication and integrity of data and devices, privacy and secure software installation (Kshetri 2017; K. Biswas and Muthukkumarasamy 2017; Khan and Salah 2018).

Even though Blockchain philosophy is first created for securing information/value transfer, one simple misconfiguration may pose a serious security risk; especially, in BloT systems that deploy permissioned or private types of Blockchain. It has been a concern when a Blockchain is participated by a small number of participants or relying on selected parties as Blockchain validators. The security challenges imitate traditional centralised systems when the selection of validators is not diverse enough to promote trust to users of IoT services (Viriyasitavat and Hoonsopon 2018).

Problem statement: Trusting IoT-based services becomes an important issue in selecting or utilising them over the Internet. The implementation of BloT services is proprietary and independent where neither standards nor common guidelines exist for security assurance. Poor implementation and misconfiguration may result in security holes and can cause damage to users. Trust of BloT services relies on the implementation and configurations of Blockchain systems. Furthermore, different users may have different level of trust (Zhang, Kong, and Zhou 2018) even with the same BloT service. This is due to the criticality of executions. This dynamicity poses an obstacle to develop common security standards that apply to every BloT-based system.

To solve this, we develop a generic architecture design of Blockchain with the integration of Public Key Infrastructure (PKI) to establish trust of BloT services. PKI has been successfully used for the endorsement of universal identities as well as specification of public key encryption and digital signature algorithms. It can also be extended to cover and guarantee properties of BloT services. Our design can potentially solve the problem of

trust of IoT-based service vendors that provide services on top of Blockchain. To address the different trust level, we additionally demonstrate how specification language can be helpful to express requirements to specify the properties of BloT services. Our contributions are summarised as follows:

- (1) Challenges of trust of BloT systems are identified.
- (2) The current state and challenges of PKI application to the area of Blockchain and IoT are presented.
- (3) The architecture design with PKI integration to establish trust of BloT services is demonstrated.
- (4) Trust of a BloT service is based on BloT service properties and is evaluated by requirements expressed by formal specification languages.

2. Background and related works

2.1 Blockchain-based Internet of Things (Biot)

Blockchain immutability and auditability and its smart contract capability are a key driver for establishing trust of IoT services. With similar topology to IoT, Blockchain is considered a natural solution to a number of problems in IoT environments (Dorri, Kanhere, and Jurdak 2016). There has been an exponentially increasing number of proposals since 2018 that study the integration of Blockchain and IoT in different aspects including security and privacy, data management, new business models, authentication and access control, and system maintenance (Zhang and Chen 2020). BloT is starting to appear in many applications such as automobile (Huang et al. 2020), industries (Ding and Jiang 2018; Reinhardt, Dr Jorge, and Dr Denis 2020), healthcare (Aceto, Persico, and Antonio 2020) and other autonomous system (Girma et al. 2020).

In the area of trust of IoT services, many attempts have been conducted to tackle security and privacy. Due to IoT characteristics such as enormous scale, openness, dynamics and heterogeneity (Viriyasitavat, Anuphaptrirong, and Hoonsopon 2019), the attack landscape is far beyond than what have occurred in traditional systems. Blockchain is capable of securing IoT devices with identification, data integrity, authentication, transmission and access control (Demirkan, Demirkan, and Andrew 2020; Tsang et al. 2021). Some researches (Christidis and Devetsikiotis 2016; Viriyasitavat et al. 2019c) have identified the broad scope of Blockchain applications in business oriented IoT services. Major recent works in this area are listed as follows:

2.1.1. Privacy protection

Enigma (Zyskind, Nathan, and Pentland 2015) employed Blockchain as an external service to control the access over IoT networks. Dorri et al. (Dorri et al. 2017) provided a case study of using Blockchain in a smart home IoT devices to preserve privacy.

2.1.2. Securing devices and data

To protect faulty or forgery IoT devices and data, Huh et al. (Huh, Cho, and Kim 2017) adopted Solidity smart contracts with the to manage and control IoT devices authentication and registration. Ouaddah et al. (Ouaddah, Elkalam, and Ouahman 2016) analysed

challenges and current issues of access controls when the control has to be enforced by resource constrained IoT devices. The works in (Liu et al. 2017; Shafagh et al. 2017) utilised Blockchain to verify the integrity and access control of data in a distributed data storage system. Wu et al. (Wu et al. 2021) demonstrated the need of bidirectional authentication in IoT environments. They developed a framework to support this system which is applied to a case study of pallet pooling management (Wu et al. 2021). Other works (Axon and Goldsmith 2017; Fromknecht and Yakoubov 2014) applied PKI for device identities which, will be discussed in more details in the next section.

2.1.3. New business model

Blockchains and IoT have enabled several new business models (Demirkan, Demirkan, and Andrew 2020). Business architecture by (Zhang and Wen 2017) was designed for IoT commodities, where the assets are exchanged using coins as a medium according to pre-defined smart contracts. BPIIoT (Bahga and Madiseti 2016) defined an architecture of Blockchain smart contracts to simulate agreements for cloud-based manufacturing. Huckle et al. (Huckle et al. 2016) discussed the potential of Blockchain and IoT for shared economy.

2.1.4. Secure configuration

System configuration is another challenge to IoT devices. It involves a wide range of attacks if devices are not secured. Software embedded in IoT devices must be frequently updated to patch vulnerabilities. Boudguiga et al. (Boudguiga et al. 2017) used peer-to-peer (P2P) mechanism to deliver system updates to IoT devices. The updates are verified for their authenticity by using Blockchain. Samaniego and Ralph (Samaniego and Deters 2016) proposed Blockchain to define IoT security components and transmission to support interoperations of services.

Blockchain can benefit a wide range of security in IoT. IoT seems to be able to leverage the prevalence of Blockchain to support trustworthiness without central control. However, most articles rely on an assumption that Blockchain is inherently secure. There are many issues regarding Blockchain implementation and configuration. Trust of BloT services relies greatly on these grounds, such as types, number of nodes and implementations. The increasing scale of BloT services poses a major risk to business. Common standard or regulation to guide the implementations of secure Blockchain is lacking; this exposes serious vulnerabilities when a Blockchain is implemented inappropriately.

2.2 Blockchain challenges in IoT

Unfortunately, no governed entities are set up to police the validity and security of any Blockchain-based application. Several ICOs claiming strong Blockchain implementation appear to use centralised database on Cloud instead of Distributed Ledger (DLT). Several frauds have occurred counted for the loss over \$500 million in 2017 ('Hacks, Scams and Attacks: Blockchain's 2017 Disasters – CoinDesk' n.d.).

Although the futuristic Blockchain has promised various benefits, serious challenges in the context of BloT services exist. This section focuses on the challenges related to BloT configuration and implementation (Li Da and Viriyasitavat 2019).

2.2.1. Confirmation time

The time delay of transaction settlement, that is the time when a transaction is created until it is confirmed and included in a Blockchain, is caused by a consensus protocol, mostly used in public Blockchain. This is impractical in many BloT scenarios involving time-critical operations. Some popular consensus protocol such as Proof-of-Work (PoW) can guarantee this settlement with probability, which nowadays using six blocks deep suggested by Bitcoin (a transaction is confirmed when its block is appended with six subsequent blocks), which takes approximately 60 minutes. The confirmation is very important to ensure that information in a Blockchain is immutable. On the other hand, private and permissioned Blockchains do not suffer the delay since the consensus protocols in use such as Practical Byzantine Fault Tolerance (PBFT), voting or lottery-based scheme, are designed to deliver near real-time confirmation. This is possible because the protocols involve a smaller number of nodes to perform transaction validation.

2.2.2. Trust

Most systems behind BloT service vendors are independent, mostly exercise private or permissioned Blockchains to deliver services. Although some Blockchain benefits are retained, it introduces trust problem as they rely on a handful number of selected validators, authorised nodes that perform consensus operations. The systems are prone to collusion attacks when malevolent nodes collaborate to delicately validate transactions to defraud users. Trust is a major issue even when the collusion is absent. One notable example is Facebook Diem (formerly Libra). The validators in Diem Association are mostly from big companies in the United States. Trustworthiness is decent for users in United States but may not be the same from the viewpoint of others, like users in Russia or China. This situation usually encounters a trade-off of fast confirmation time and trust.

2.3 Blockchain-enabled trust of IoT services

The integration of Blockchain and IoT is very broad in terms of research topics and applications areas. This section is devoted to the domain of Blockchain-enabled trust of IoT services. IoT services have been increasing and some may offer similar tasks. Mainly, selecting the services depends mostly on trust that usually involves non-functional requirements, commonly known as Quality of Service (QoS). The examples of QoS are availability, reliability, response time and efficiency (Biswas and Giaffreda 2014; Viriyasitavat and Zhuming 2020). QoS-based trust is now facing the issue of scalability and trustworthiness of QoS information. The research in this area is quite limited. In this regard, Blockchain-based QoS scheme (Viriyasitavat et al. 2019b) was introduced to address trust, derived from trust of agents, trust of processes that monitor, collect and measure QoS value, and subsequently trust of QoS information stored in Blockchain. Scalability is lessened by applying layer architecture and sharding technique, where QoS measurement is collected from collectively trusted subnetworks. This scheme is implemented on permissioned Blockchain with smart contracts to control the registration of agents. Viriyasitavat et al. (Viriyasitavat, Xu, and Bi 2018) extended the work by encompassing common specification patterns found in Service-Based Application (SBAs), which are used to evaluate QoS of IoT services.

2.4 PKI and blockchain implementation

PKI technology relies on Certificate Authority (CA) to manage certificates assigned to requesting entities. Technically, an entity is bound to a unique public key. Information inside a certificate entails the specification of encryption technology, for example, RSA or ECC, and digital signature algorithm. The property specification directs the establishment of secure communication channel. In practice, CAs contain a lot of shortcomings. CAs are considered trust anchors, which are vulnerable to single point of failure due to centralisation. There are a few studies trying to exploit Blockchain advantages to solve traditional PKI shortcomings. SCPKI (Al-Bassam 2017) is a decentralised design PKI model based on web-of-trust to replace traditional PKI technology. It is built on Ethereum ecosystem with transparency that benefits fast detection of rogue certificates. We are motivated by this work has motivated to extend the capabilities to address more fine-grained attributes for Blockchain into certificate. Fromknecht et al. (Fromknecht and Yakoubov 2014) explored the problem of identity retention and introduced CertCoin with optimisation to be used in light nodes, such as smart phone. By addressing IoT characteristics (Viriyasitavat, Anuphaptrirong, and Hoonsopon 2019), Singa and Bertino (Singla and Bertino 2018) implemented Blockchain as an alternative to CA-based PKI to support IoT devices. However, this work neglects the benefits of existing PKI infrastructure and try to use Blockchain entirely to substitute current PKI settings.

From the literature above, most of research aims at using Blockchain as an alternative to existing PKI ecosystem. Other approaches assume that trust of IoT services is constructed from QoS. They establish trust of QoS information using Blockchain and smart contracts. To the best of our knowledge, Existing literature applies Blockchain benefits to gain advantage over traditional approach without the awareness of abilities and security of Blockchain itself.

3. Context

We adopt a design science practice according to the guidelines (Hevner et al. 2004). To position our paper, we have investigated related literature to identify challenges and research opportunities.

3.1. Design artefacts

The artefacts include (1) extended PKI certificate with fine-grained attributes to certify Blockchain implementation and configuration; (2) design architecture to support trust of BloT services by reducing or eliminating untrusted validators. (3) A specification language to specify requirements for Blockchain behind BloT services; (4) a data structure of information inside certificate, which will be presented in Section 5. Additionally, our discussion guides readers to extend the use of our scheme in a boarder or specific scope.

3.2. Problem definition

'In the context of BloT services, Blockchain implementation, mostly a permissioned type, behind those services are dynamically composed of proprietary devices with dependent system configurations, in which the selection of consensus protocol and validators

depend solely on each vendor. This faces an important challenge of trust of the Blockchain, which subsequently influences on the trust of the entire BloT services.' It is extremely difficult for service users to obtain knowledge or trusts of all validators in BloT services.

3.3. Design evaluation

Case study-based and simulation-based evaluations are conducted. We decide to use a case study to demonstrate our design application as well as performance evaluation in the simulation.

3.4. Research contributions

Please refer to [Section 1](#) for our contributions.

3.5. Rigour

The design is generic in the context of BloT services, where trust of Blockchain implementation and configuration is a key-enabler for a wide spread usage of the services.

3.6. Design alternatives

By conducting the search in [Section 2](#), trust of BloT services has been mainly evaluated by QoS, while other works attempt to replace traditional PKI with Blockchain. However, the problem of the trust of Blockchain has not been satisfactorily tackled. This is another important aspect to be investigated.

3.7. Intended audiences

Our attempt is to demonstrate architecture design at a technical level and explains how it can be employed to resolve the identified problems. This paper is intended for Blockchain and IoT academicians who are interested in trust of BloT services.

4. Trust of biot services

Our approach is to adopt PKI for trust establishment for BloT services. The design consists of three main elements: (1) BloT services, which are collaboratively built from multiple IoT devices or composed of BloT subservices, (2) PKI used to establish trust of BloT and (3) the specification language that allows users to specify requirements for acquiring BloT services (see [Figure 2](#)).

4.1 Three forms of biot service interoperation

The ecosystem of BloT services can be categorised into three forms shown in [Figure 1](#). This categorisation is based on the aspect of trust being applied to BloT services. (Note that devices in [Figure 1](#) are IoT devices or computable objects with high resources.)

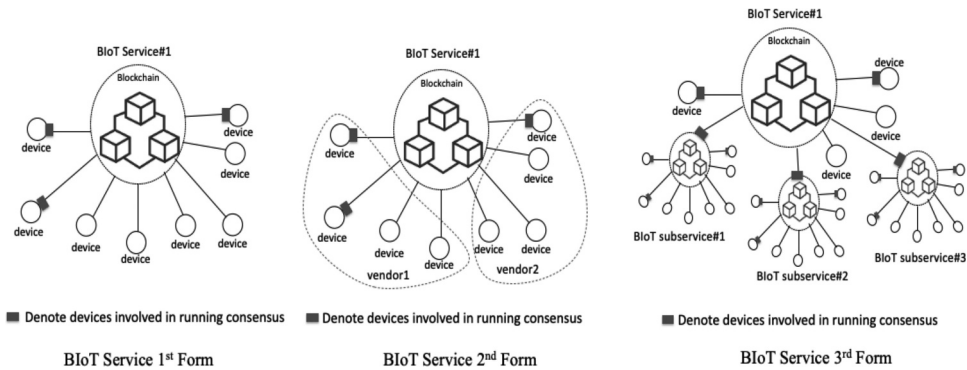


Figure 1. Three forms of Biot service ecosystem.

- (1) The first form consists of one private Blockchain controlled by one party, which constitutes BloT services. In terms of configuration, this mode is not different from centralised system, except the benefits of Blockchain processes and data structure are gained. Blockchain is mainly used for data validation where every data item generated by IoT devices is verified before included. In this mode, the vendor selects a set of validators from IoT devices or other computable objects like computers or smartphones for running consensus.
- (2) The second form appears when BloT services are shared by one or multiple vendors to form consortium. Permissioned Blockchain is mainly used, and validators usually involve service vendors and other agreed external parties.
- (3) The third form operates on the concept of nested architecture. A composite BloT service can be composed of multiple subservices. This mode creates more value-added services and promotes scalability and interoperation. Some subservices may serve as a validator for the upper layer.

It can be noticed that the implementation behind BloT services varies, ranging from simple to complex ones, which hierarchically involves several subservices.

4.2 Trust of Biot Services

In our focus, trust of BloT services is classified into two aspects.

4.2.1. Def 1

Trust of BloT services is classified into QoS and implementation. QoS information suggests quality of service provision, whereas the implementation implies security of BloT services.

The first trust aspect is derived from non-functional requirements using QoS information that is collected overtime. In this aspect, Blockchain is an attractive choice to provide trust by replacing a centralised trusted registry that originally manages QoS of services (Viriyasitavat et al. 2019b). The second trust aspect is based on the implementation and configuration of Blockchain behind BloT services. They influence trust from the perspective service users. This paper emphasises on the latter by looking at the implementation and configuration of Blockchain in BloT services. Specifically, we adopt PKI certificate to

endorse the information that contributes to trust of Blockchain. For instance, trust of Blockchain may include encryption algorithm, communication technology, and consensus protocols. Other attributes can additionally be embedded as elements inside the certificate.

Consensus protocols in public Blockchain such as Proof-of-Work (PoW) exhibits some restrictions. The big hindrance is the excessive delay for transaction confirmation, which is impractical in BloT environments that frequently involve time-critical operations. Most BloT services opt for permissioned Blockchain, for example, Practical Byzantine Fault Tolerant (PBFT) that offers near real-time confirmation. It consists of validators to verify transactions. Trust is a major problem when validators are improperly selected. However, fast confirmation time outweighs the problem of trust in BloT environments. The next section demonstrates that the incorporation of PKI can mitigate the problem.

In short, the implementation information of BloT service will be included as attributes in an extended Blockchain certificate covering validators, encryption algorithm, communication technology and other properties required in specific applications. PKI certificate has successfully been used and the concept for verifying domain names is identical to verify Blockchain validators and their properties. [Section 5](#) provides the details of the usage of this certificate.

5. Blockchain certificates

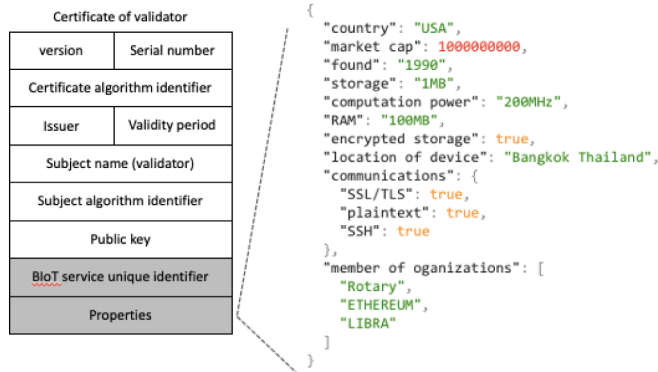
PKI certificate is an important document to establish trust. In open environments, various validator sets are found in different BloT services. It is extremely hard for users to obtain information of all validators, especially when the change of validators is frequent. Our approach applies PKI certificate and extends it to address this issue.

PKI certificate such as X.509 has long been used to endorse a public key of an entity associated with a domain name. It can also direct secure communications, for example, SSL/TLS, over the Internet. With the same concept, this certificate can be extended to address Blockchain implementation and configuration information behind BloT services. We customise X.509 elements to create two certificate types (see [Figure 2](#), the shaded fields indicate extension).

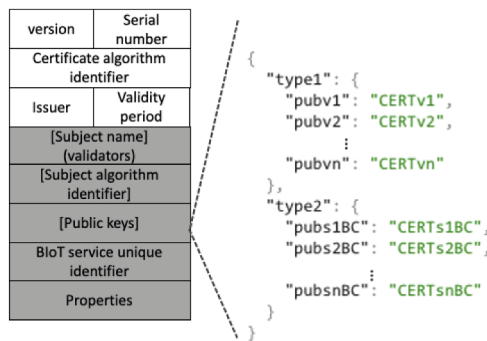
5.1. Def 2. certificate types

5.1.1. Type1

It (see [Figure 2A](#)) is a certificate of a validator. The objective is to endorse a validator and its properties required for participating in an intended BloT service. The properties field is flexible and extensible, suggesting that the JSON format can express additional and arbitrary key-value. However, in practice, standard templates or ontology are recommended for specific applications. The attribute descriptions are further elaborated in [Table 1](#).



(A) Type1: certificate of validator



Where,

$pubsnvn$

$CERTvn$

$pubsnBC$

$CERTsnBC$

is a public key of Blockchain validator n of subservice n ,

$CERTvn$ is a certificate of Blockchain validator n of subservice n .

$pubsnBC$ is a public key of Blockchain implementation of subservice n ,

$CERTsnBC$ is certificate of Blockchain implementation of subservice n ,

(B) Type2: certificate of Blockchain implementation and configuration

Figure 2. Examples of certificate of validators and certificate of Blockchain implementation and configuration. (A) Type1: certificate of validator Where, $pubsnvn$ is a public key of Blockchain validator n of subservice n , $CERTvn$ is a certificate of Blockchain validator n of subservice n . $pubsnBC$ is a public key of Blockchain implementation of subservice n , $CERTsnBC$ is certificate of Blockchain implementation of subservice n , (B) Type2: certificate of Blockchain implementation and configuration

5.1.2. Type2

It is a certificate of Blockchain implementation and configuration (see Figure 2B), which is the assertion of properties of a BloT service. This certificate mainly includes, if any, the certificates of its subservices and all validators that are responsible for running consensus. Therefore, a list of public keys inside this certificate is classified into two types. If the participant is a device or computational objects, a public key from associated certificate of a validator is required. But, if the participant is a BloT subservice, the certificate of

Table 1. Type1: certificate of validator.

Attributes	Descriptions
Version	Version number
Serial number	Unique identifier of certificate
Certificate algorithm identifier	Algorithm of CA used for certificate encoding and digital signature such as RSA and ECC
Issuer	Entity that issues certificate
Validity period	Start and end datetime
Subject name	The name of a validator that certificate endorses
Subject algorithm identifier	Algorithm used by a validator including encryption and digital signature such as RSA and ECC
Public key	Public key associated with a validator
Properties	Denoting attributes of a validator (see Figure 2A)

Table 2. Type2: certificate of Blockchain implementation.

Attributes	Descriptions
Consensus	A selected consensus protocol used in a BloT service such as Voting, PBFT, etc.
[Subject names]	A list of validators involved in a BloT service
[Subject algorithm identifiers]	A list of algorithms used by and associated with each validator including encryption and digital signature such as RSA and ECC. This defines communication channel among validators during running consensus protocol.
[Public keys]	Public key associated with each validator. In the case of a nested BloT service, the public key will be a certificate of Blockchain implementation and configuration of each BloT subservice.
BloT service unique identifier	An ID of a BloT service that the validators in the list participate. This will be used as part of nested BloT services.
Properties	Denote the attributes of BloT services. These properties are encoded using specification language.

*The descriptions of attributes (Version, Serial number, Certificate algorithm identifier, Issuer, and Validity period) are similar to Certificate of Validator in Table 1.

Blockchain implementation and configuration is needed. [Table 2](#) provides the details of the attributes. (Note that example in [Figure 2B](#) includes basic attributes such as subject name, subject algorithm identifier, and BloT service unique identifier.) Later, the process of certificate validation is illustrated with examples in [Section 6](#) and [7](#).

Trust of a BloT service depends heavily on the trust of the validators and Blockchain implementation and configuration. These certificates will be issued to validators and BloT services based on issuer (CA) policies, which will serve as an initial trust for BloT services. Users of BloT services and a BloT service that consists of BloT subservices can obtain initial trust from the certificates. Thus, interoperation among BloT services and users will be leveraged.

6. Usage architecture

Our strategy is to aggregate and expand existing PKI to establish trust of BloT services. We believe that incorporating PKI to certify properties of BloT services will be sufficient for establishing trust, and hence encourage large scale interoperations. Our architecture is demonstrated in a way that imitates the current practice of PKI with extended certificate to support Blockchain implementation and configuration.

[Figure 3](#) shows an overview of our usage architecture (see [Figure 3](#)), where the core element is to incorporate Blockchain with IoT services and PKI.

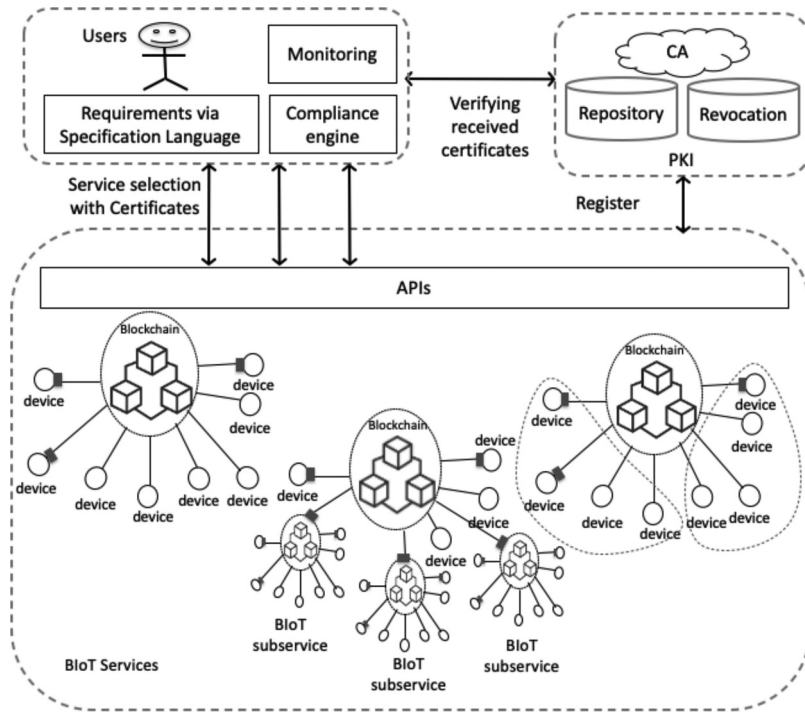


Figure 3. Overview architecture of PKI and usage of Biot services.

6.1. Pre-initialisation

During the formation of a BloT service, each validator is required to supply its certificate that endorses its own information entailed with serial number (BloT unique identifier in type1 certificate). These certificates will be shared and gathered to constitute a type2 certificate of such BloT service. For example, according to the three forms of BloT services mentioned in Section 4.1 and Figure 1, the certificates of each form are demonstrated as follows.

The first and the second forms, a certificate type2 ($CERT_{BC}$) of the BloT service include certificate of its validators.

$$[CERT_{v1}, \dots, CERT_{vn}], \text{ where}$$

$CERT_{vn}$ denotes type1 certificate of validator n

The third form, a certificate type2 ($CERT_{BC}$) of BloT service include certificate of sub-service#1 to subservice#n

$$[CERT_{s1,BC}], \dots, [CERT_{sn,BC}]$$

and certificates of its validators

$$[CERT_{v1}, \dots, CERT_{vn}], \text{ where}$$

- 1) $CERT_{sn,BC}$ denotes type2 certificate of BloT subservice n , and
- 2) $CERT_{vn}$ is type1 certificate of validator n .

It is possible to have certificate combination from some BloT subservices to be part of an upper BloT service. Specifically, $CERT_{BC}$ can include subservice certificate.

6.2. Initialisation

A BloT service to be used by a user is evaluated based on user's requirements. One example of the requirements can state that a BloT service shall use permissioned Blockchain with PBFT, or another requirement can specifically state that the validators in running PBFT must be from USA and China. The compliance of these requirements reflects trust of that BloT service, and trust is subjective in the viewpoint of different users. Specification language serves as a formal tool to encapsulate the requirements (Viriyasitavat, Da Xu, and Martin 2012), which becomes a fundamental ground for automatic compliance checking (Viriyasitavat, Da Xu, and Viriyasitavat 2014).

6.3. Propagation

Each BloT service supplies a set of certificates that endorse the properties according to user's requirements. Multiple certificates are presented when the BloT service contains one or more BloT subservices.

6.4. Verification

The certificates will be tested for validity using similar processes defined in PKI. Elements to be checked are, for instance, expiry date, revocation, etc.

6.5. Compliance checking

The engine checks the compliance between user's requirements and BloT service properties based on certificates. Positive result suggests a green light to use that service.

During service usage, user continually monitor to ensure that the compliance is still satisfactory. This is necessary because properties of a BloT service can be changed, and user's requirements can be updated. If such changes happen, the compliance needs to be re-evaluated. This dynamicity creates a scalability challenge, and this topic will be discussed later in [Section 8](#).

7. An application example and evaluation

This section provides an example to demonstrate the incorporation of PKI to establish trust of BloT services. We also report the evaluation of our approach.

7.1 Application example

To understand the application of the approach, the first example deals with Tsunami sensors that are part of a disaster detection service (BloT service), which run on a permissioned Blockchain. The sensors are deployed across several locations, and we can have multiple sensors at the same location. This example assumes the sensors belong

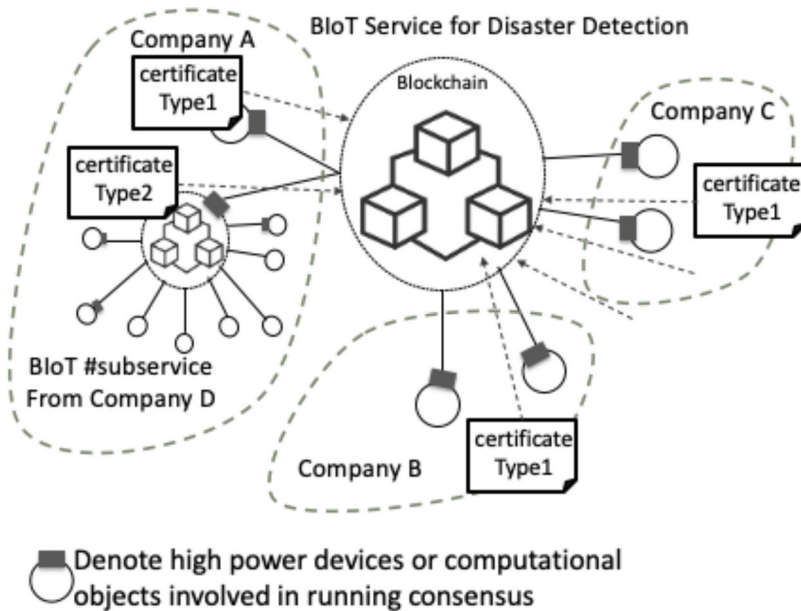


Figure 4. The structure of Biot service interoperation in this example.

to four interoperating companies (A, B, C and D), where Company D is a subservice used by Company A (see Figure 4). This setup represents the 2nd and 3rd of BloT service from in Figure 1.

The information of Company A-D endorsed by certificates (type1 and type2) will be shown as follows. It is important to note that in this example we use country as a sole validator’s property and demonstrate the fields inside the certificates as just enough to show how our approach works. More properties can be included by adding more field inside the certificates.

Company A (certificate type1)

```
Subject Name: CompanyA
Algorithm Identifier: RSA or ECC
Public key: 0xaaa .....
Properties: {
    Country: "USA",
    Communication: "SSL/TLS"
}
```

Company A (certificate type2 from subservice CompanyD)

```
Subject Name: CompanyD
Algorithm Identifier: RSA or ECC
Public key: {
    Certificate of validators: {
        pubv1: "certv1",
        pubv2: "certv2"
    }
}
BloT Unique Identifier: 12593949
Properties: {
    Consensus: PBFT
}
```

Company B (certificate type1 two computational objects shown in Figure 4 process the same values)

```

Subject Name: CompanyB
Algorithm Identifier: RSA
Public key: 0xbbb .....
Properties: {
  Country: "CN",
  Communication: "SSL/TLS"
}

```

Company C (certificate type1 two computational objects shown in Figure 4 process the same values)

```

Subject Name: CompanyC
Algorithm Identifier: RSA or ECC
Public key: 0xccc .....
Properties: {
  Country: "JP",
  Communication: "SSL/TLS"
}

```

Disaster detection service (certificate type2 for the BloT service)

```

Subject Name: Disaster Detection System
Algorithm Identifier: RSA or ECC
Public key: {
  Type1: {
    pubv1: "CERT from companyA",
    pubv2: "CERT from companyB",
    pubv3: "CERT from companyC"
  }
  Type2: {
    pubsBC: "CERT from subservice companyD"
  }
}
BloT Unique Identifier: 172345678
Properties: {
  Consensus: PBFT,
  tx verification: A AND (B OR/AND C)
}

```

Type2 certificate from BloT subservice from company D is included in the main certificate of BloT service of disaster detection, where information inside the certificate of company D encapsulates a lower layer information regarding its Blockchain implementation and configuration.

We develop a smart contract (*txVerification*) for an additional requirement of disaster detection service. This requirement states that in order to assert a transaction (*tx*), company A must be part of every *tx* verification and at least two companies are also part of the verification. Function *vote()* at line 6 Listing1 demonstrates the implementation of this requirement. This smart contract is embedded inside Blockchain of the BloT service. After verified by CA, this property is included inside the certificate and is signed with the following key-value. The value part is presented using Propositional logic as an example mechanism to encapsulate this requirement.

transaction verification : A AND (B OR/AND C)

Listing1 illustrates the mechanism to implement the requirement using a contract.

Listing1 A Smart Contract for tx verification with Company A and one other

```

1: contract txVerification {
    //consensus with selected validators
    //public keys of company A-D as validators
2:   address[] val_list = [0xaa . . . . ., 0xbb . . . . ., 0xcc . . . . ., 0xdd . . . . .];
    //issuing log when tx is verified
3:   event write (byte32 log);
    // check if a validator is in the list
4:   function chk(address _addr) public return (bool) {
4.1:     if (_addr in val_list) then return true;
4.2:     else return false;
4.3:   }
    // keep track of voted validators of a tx
5:   address[][] voted_addr = [][]
    // 1st-stage voting for tx verification
6:   function vote(address _addr, byte32 _tx) public
      return (bool) {
6.1:     require(chk(_addr));
    //create or insert into voted_addr
6.2:     if voted_addr [_tx].contains(_addr) {
6.3:       voted_addr[_tx] << _addr;
6.4:       if voted_addr[_tx] > 1 &&
          voted_addr[_tx].contains("0xaa . . . . . ") {
          //process tx by sending to PDFT
6.5:         send(_tx);
6.6:         emit write("_tx is verified");
6.7:       }
6.8:     } else return false;
6.9:   }
7: } // end contract

```

In total, the disaster detection service contains three type1 certificates (from company A to C) and two type2 certificates. One is from subservice from company D and another one is for itself. In the viewpoint of users, the decision of using this BloT service depends on trust. In this scenario, we focus on the properties of validators and Blockchain implementation and configuration to initialise trust. Meanwhile, trust may be subject to other factors like QoS during runtime. Please refer to the work in (Viriyasitavat et al. 2019b) that adopted Blockchain to establish trust of QoS information as a source for trust evaluation during runtime.

Regarding the above setup, this paragraph demonstrates how trust is established from the requirements from service users. The proven method is to encapsulate these requirements using requirement specification languages. The main benefit of the languages is that they support automatic compliance checking. In what follows, we borrow the notations from the specification language in (Viriyasitavat, Da Xu, and Martin 2012) to express the requirements. In this example, we suppose that user X states four requirements to use the disaster detection service.

REQ1: In terms of security, all validators of the BloT service must operate on SSL/TLS using RSA or ECC. This requirement can be expressed as

$$A_v(\text{Communication} : \text{SSL/TLS}) \wedge A_v(\text{Algorithmidentifier} : \text{RSA} \vee \text{ECC})$$

which is equivalent to

$$A_v(\text{Communication} : \text{SSL/TLS}) \wedge A_v(\text{Algorithmidentifier} : \text{RSA} \vee \text{ECC})$$

$$A_v(\text{Communication} : \text{SSL/TLS}) \wedge$$

$$(A_v(\text{Algorithmidentifier} : \text{RSA}) \vee A_v(\text{Algorithmidentifier} : \text{ECC}))$$

A_v indicates all validators (v) with property SSL/TLS and (\wedge) all validators must use or (\vee) ECC for encryption.

REQ2: Every BloT service and its subservices must use PBFT as a consensus protocol.

$$A_s(\text{Consensus} : \text{PBFT})$$

A_s indicates that all subservices must operate using PBFT as a consensus protocol

REQ3: At least one company from Russia must be a validator in running PBFT indicates that some validators (v) with property (Russia).

$$E_v(\text{Country} : \text{RU})$$

REQ4: Company from USA must verify every tx in this Blockchain. This is important because users tend to trust validators from their own country.

$$A_{tx}(\text{txverification} : \text{USA})$$

The main purpose of using formal specification language is that it facilitates compliance checking between requirements and properties of BloT service. Listing2 demonstrates a simple algorithm used in this scenario.

Listing2 Basic compliance checking algorithm

```

1: Function compliance ( $\Phi$ );
2: Begin
3:   If pre( $\Phi = = A_n$  { //for all
4:     For  $i$  in  $n$  {
5:       If ( $\Phi$  not satisfies  $i$ ) { return false; }
6:     } //end for
7:     return true;} //end if
8:   If pre( $\Phi = = E_n$  { //for some
9:     For  $i$  in  $n$  {
10:      If ( $\Phi$  satisfies  $i$ ) { return true; }
11:    } //end for
12:    return false;} //end if
13:    $\Phi_1 =$  compliance ( $\Phi$ . LeftNode
14:    $\Phi_2 =$  compliance ( $\Phi$  . RightNode
15:   Case  $\Phi$  .Element {
16:     ~:If( $\Phi_1 = =$  true) {return false; }
17:     Else { return true; }
18:     ^:If( $\Phi_1 = =$  true and  $\Phi_2 = =$  true) {return true; }
    
```

```

19:     Else { return false;} //end if
20:   √:if( $\Phi_1 == \text{true}$  or  $\Phi_2 == \text{true}$ ) {return true; }
21:     Else { return false; } //end if
22:   →:if( $\Phi_1 == \text{false}$ ) { return true; }
23:     Else {return  $\Phi_2$ ; } //end if
24:   ↔:if(( $\Phi_1 == \text{false}$  and  $\Phi_1 == \text{false}$ ) or ( $\Phi_1 == \text{true}$  and  $\Phi_1 == \text{true}$ )) { return true; }
25:     Else {return false;} //end if
26:   } //end case
27: END

```

Suppose that the supplied certificates are valid, the result of these requirements are as following:

REQ1	REQ2	REQ3	REQ4
True	True	False	True

REQ1 is satisfied because all validators contain SSL/TLS property, and they are all using RSA or ECC, except company B that uses only RSA. This property satisfies REQ1. The BloT service of disaster detection also satisfies REQ2 because all subservices from company D and BloT service itself are running PBFT as the consensus protocol. REQ3 is evaluated as false as no validators are from Russia (RU). Finally, REQ4 is satisfied because a validator with property USA is part of every *tx* verification. This is evident by the smart contract in Listing1 and endorsed by certificate type2 of the disaster detection service.

The compliance checking of REQ4 is special with two involved formulae. One is from REQ4, $A_{tx}(tx\text{verification} : \text{USA})$ and another one is inside type2 certificate indicating in the properties part, *tx verification*: A AND (B OR/AND C). The algorithm translates the properties of A AND (B OR/AND C) into Exclusive Disjunctive Normal Form (EDNF) as follows:

$$(A \text{ AND } B) \text{ XOR } (A \text{ AND } C) \text{ XOR } (A \text{ AND } B \text{ AND } C)$$

This means that every possible occurrence of *tx* verification falls within three patterns, according to the following clauses (1) A AND B, (2) A AND C and (3) A AND B AND C. As such, Line 5 and 10 of the algorithm in Listing1 need extension with the following term inside if statement.

```

5: If ( $\Phi$  not satisfies i) {return false;}
   can be replaced by
5.1 For j in possible occurrence of tx verification {
5.2 If ( $\Phi$  not satisfies j) {return false;}
5.3: }

```

and

```

10: If ( $\Phi$  satisfies i) {return true;}
    can be replaced as
10.1 For j in possible occurrence of tx verification {
10.2 If ( $\Phi$  satisfies j) {return true;}
10.3: }

```

Therefore, every possible occurrence of *tx* verification inside this BloT service will involve company A (with property from the United States) as part of verification.

7.2 Analysis of the algorithm

The complexity depends on the number of requirements (R) to be verified. In this example, there are four requirements. Function compliance in Listing2 contains recursive calls to the number of operators (e.g. \sim , \wedge , \vee , \rightarrow , and, \leftrightarrow) where the complexity is calculated from the number of variables (T). The loop at Line 5.1–5.3 and 10.1–10.3 run according to the number of possible occurrence (C). As a result, the complexity will be $O(|R| \times 2^{(|T||C|)})$. In many cases, such as in REQ#1-#3, where the occurrence of each txs verification is unimportant, the complexity will be $O(|R| \times 2^{|T|})$.

8. Experiments

This section reports the performance of the proposed approach. Two types of scenarios are evaluated: (1) the normal cases when tx verification is unimportant (REQ#1-#3) where the complexity has been qualitatively analysed as $O(|R| \times 2^{|T|})$ and (2) the special scenario (REQ#4) where the compliance checking requires the enumeration through every possible validators combination in each round of txs verification, resulting in the qualitative analysis of $O(|R| \times 2^{(|T||C|)})$. Figure 5 reports the relationship between the numbers of variables (T) and time complexity. The result reveals that the complexity is exponentially proportional to the number of T. In the second scenario, Figure 6 shows that the complexity increases faster than in the first scenario.

All programme scripts are developed in RUBY programming language on a 64-bit Windows 10, Intel® Core™ i5-2435 M CPU @ 2.40 GHz, 4 GB RAM. We design the experiment to evaluate time performance with the number of variables (T) in a formula ranging from 1 to 20, and the number of occurrences (C) following the equation $3f + 1$, where f is the maximum number of faulty validators where PBFT can tolerate. The result in Figure 5 shows the performance of the first scenario, while Figure 6 shows the performance when the number of the validators are 4, 7, 10, 13 and 16, respectively, with the number of |T| from 1 to 5. The term |R| is negligible in this experiment as its effect is insignificant to the time complexity.

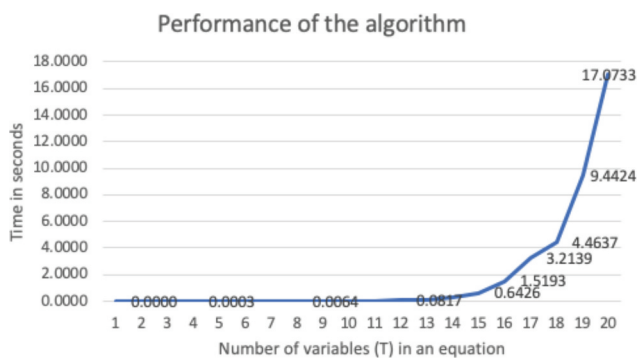


Figure 5. The time complexity of the first scenario.

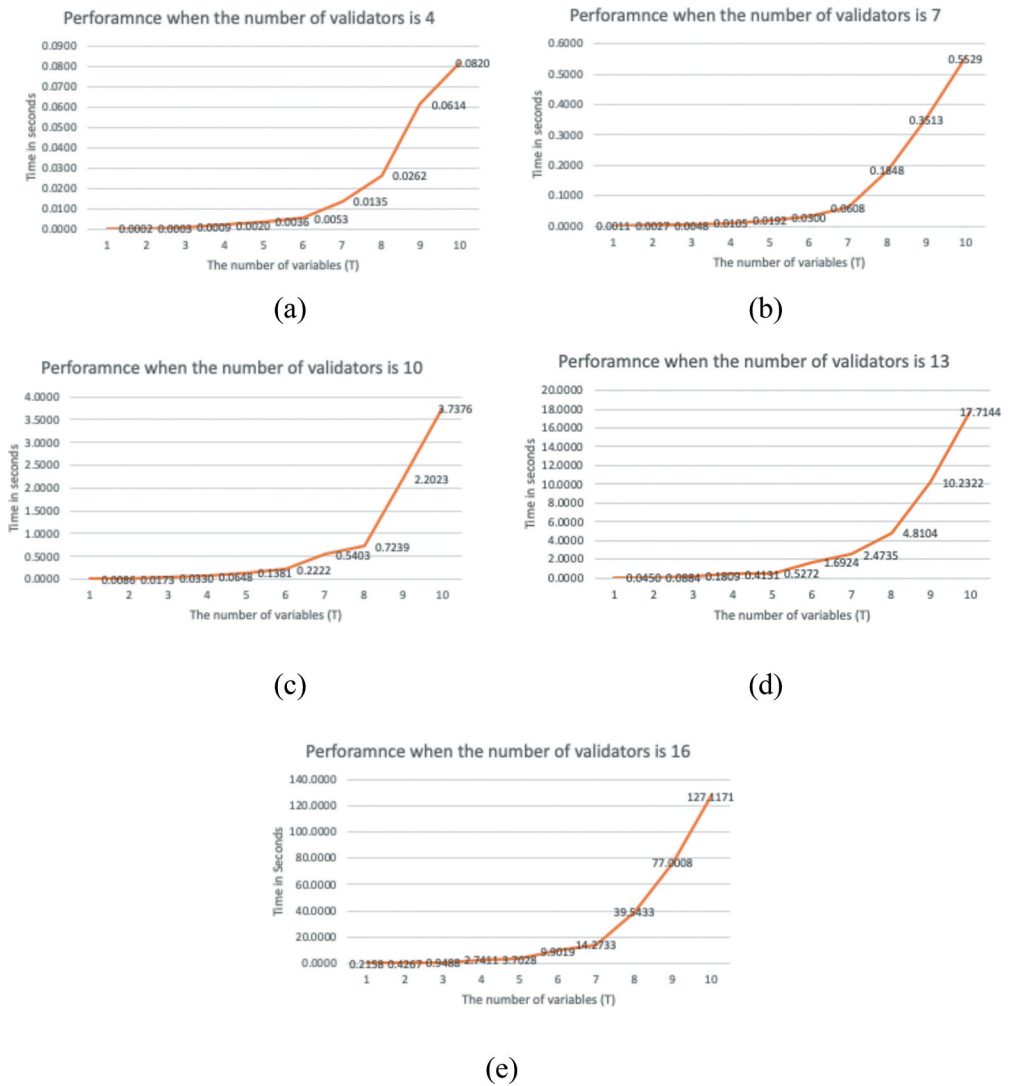


Figure 6. The time complexity of the second scenario where the number of validators is 4, 7, 10, 13 and 16.

9. Discussion and conclusion

This section concludes our work and identify major challenges the capability of our approach to respond to the problems. Trust of BloT services largely depends on Blockchain implementation and configuration. BloT services are dynamically composed of proprietary devices with various configurations. Most of them employ permission Blockchain and an underlying system. Our approach employs PKI to establish trust of this type of service, which can reduce risk and influence the prevalent use of BloT services.

In the users' point of view, trust is subjected to their requirements that are initially issued as constraints before deciding to use a service. Using the same service from different users may be subject to different requirements. In our approach, a specification language is employed to mathematically capture the requirements and certificates are used to endorse properties of validators and BloT services. The main reason of using PKI is the robustness. PKI has been widely and successfully used for verification of identities and deployed into various fields that require certification of information. Certifying BloT services and validators is relatively an extended idea from original PKI. However, some challenges of our approach are discussed below.

9.1. Dynamic change of validators and properties

In open environments, the frequent changes of validators, BloT services and their properties imply the invalidity of previous certificates. This requires intensive monitoring and executions to issue new certificates to cope with the changes. The enhancement of traditional PKI is necessary, especially to cope with the wide use of BloT in the near future.

9.2. Scalability

One major issue is that certifying properties of validators and BloT services requires intensive computation and monitoring. This hinders scalability to our approach. However, the increase of computation power will resolve the issue.

9.3. QoS

While our approach is developed for initial trust, QoS is another important factor for trust during runtime. QoS scheme also requires intensive monitoring for service performance. The works presented in (Viriyasitavat et al. 2019b, 2018) incorporate Blockchain and smart contracts to address this issue. We believe that these works are complement towards the full-scale trust of BloT services.

9.4. Compliance checking

Specification language is used to encapsulate user's requirements and the properties inside certificate are expressed by Propositional logic. Since they are both logic-based, a new class of compliance checking is necessary. In our future work, an efficient algorithm for this checking will be studied.

Finally, Blockchain has been considered one of the technological breakthroughs and its distributed topology is similar to IoT environments. In the near future, the wide use of BloT services is very much anticipated, where trust will become a major issue to the prevalent use of the services. This paper introduces a generic architecture design incorporating PKI to establish trust of BloT services. Our work will potentially solve the trust problem when selecting service vendors that provide services on top of Blockchain.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Wattana Viriyasitavat  <http://orcid.org/0000-0001-7247-4596>

References

- Aceto, G., V. Persico, and P. Antonio. 2020. "Industry 4.0 And Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0." *Journal of Industrial Information Integration* 18 (June): 100129. doi:10.1016/J.JII.2020.100129.
- Al-Bassam, M. 2017. "SCPki: A Smart Contract-Based PKI and Identity System." In *BCC 2017 - Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, Co-Located with ASIA CCS 2017*, 35–40. Association for Computing Machinery, Abu Dhabi, United Arab Emirates. doi:10.1145/3055518.3055530.
- Axon, L., and M. Goldsmith. 2017. "PB-PKI: A Privacy-Aware Blockchain-Based PKI." In *ICETE 2017 - Proceedings of the 14th International Joint Conference on e-Business and Telecommunications*, 4: 311–318. Madrid, Spain: SciTePress. doi:10.5220/0006419203110318.
- Bahga, A., and V. K. Madiseti. 2016. "Blockchain Platform for Industrial Internet of Things." *Journal of Software Engineering and Applications* 09 (10): 533–546. doi:10.4236/jsea.2016.910036.
- Biswas, A. R., and R. Giaffreda. 2014. "IoT and Cloud Convergence: Opportunities and Challenges." In *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, 375–376. IEEE Computer Society, Seoul, Korea. doi:10.1109/WF-IoT.2014.6803194.
- Biswas, K., and V. Muthukkumarasamy. 2017. "Securing Smart Cities Using Blockchain Technology." In *Proceedings - 18th IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, 1392–1393. Yanuca Island, Cuvu, Fiji: Institute of Electrical and Electronics Engineers . doi:10.1109/HPCC-SmartCity-DSS.2016.0198.
- Boudguiga, A., N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey. 2017. "Towards Better Availability and Accountability for IoT Updates by Means of a Blockchain." In *Proceedings - 2nd IEEE European Symposium on Security and Privacy Workshops, EuroS and PW 2017*, 50–58. Paris, France: Institute of Electrical and Electronics Engineers. doi:10.1109/EuroSPW.2017.50.
- Christidis, K., and M. Devetsikiotis. 2016. "Blockchains and Smart Contracts for the Internet of Things." *IEEE Access*. Institute of Electrical and Electronics Engineers, vol. 4, pp. 2292–2303, May 2016. doi:10.1109/ACCESS.2016.2566339.
- Demirkan, S., I. Demirkan, and M. Andrew. 2020. "Blockchain Technology in the Future of Business Cyber Security and Accounting." *Journal of Management Analytics* 7 (2): 189–208. doi:10.1080/23270012.2020.1731721. Taylor and Francis Ltd.
- Ding, K., and P. Jiang. 2018. "RFID-Based Production Data Analysis in an IoT-Enabled Smart Job-Shop." *IEEE/CAA Journal of Automatica Sinica* 5 (1): 128–138. doi:10.1109/JAS.2017.7510418.
- Dorri, A., S. S. Kanhere, and R. Jurdak. 2016. "Blockchain in Internet of Things: Challenges and Solutions, Issued August 18, 2016." <http://arxiv.org/abs/1608.05187>
- Dorri, A., S. S. Kanhere, R. Jurdak, and P. Gauravaram. 2017. "Blockchain for IoT Security and Privacy: The Case Study of a Smart Home." In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 618–623. Kona, HI, USA: IEEE. doi:10.1109/PERCOMW.2017.7917634.
- Fromknecht, C., and S. Yakoubov. 2014. "CertCoin: A NameCoin Based Decentralized Authentication System 6.857 Class Project."

- Girma, A., N. Bahadori, T. G. Mrinmoy Sarkar, M. R. Tadewos, B. M. Nabil Mahmoud, A. Karimoddini, and A. Homaifar. 2020, July. "IoT-Enabled Autonomous System Collaboration for Disaster-Area Management." *IEEE/CAA Journal of Automatica Sinica*. 1–14. doi:10.1109/jas.2020.1003291.
- "Hacks, Scams and Attacks: Blockchain's 2017 Disasters - CoinDesk." n.d. Accessed 8 June 2020. <https://www.coindesk.com/hacks-scams-attacks-blockchains-biggest-2017-disasters>
- Hevner, A. R., S. T. March, J. Park, and S. Ram. 2004. "Design Science in Information Systems Research." *MIS Quarterly: Management Information Systems* 28 (1): 75–105. doi:10.2307/25148625.
- Huang, X., Y. Dongdong, Y. Rong, and L. Shu. 2020. "Securing Parked Vehicle Assisted Fog Computing with Blockchain and Optimal Smart Contract Design." *IEEE/CAA Journal of Automatica Sinica* 7 (2): 426–441. doi:10.1109/JAS.2020.1003039.
- Huckle, S., R. Bhattacharya, M. White, and N. Beloff. 2016. "Internet of Things, Blockchain and Shared Economy Applications." *Procedia Computer Science* 58:461–466. doi:10.1016/j.procs.2016.09.074. Elsevier B.V.
- Huh, S., S. Cho, and S. Kim. 2017. "Managing IoT Devices Using Blockchain Platform." In *International Conference on Advanced Communication Technology, ICACT*, 464–467. Institute of Electrical and Electronics Engineers, Pyeongchang-gun, Gangwon-do, South Korea. doi:10.23919/ICACT.2017.7890132.
- Khan, M. A., and K. Salah. 2018. "IoT Security: Review, Blockchain Solutions, and Open Challenges." *Future Generation Computer Systems* 82 (May): 395–411. doi:10.1016/j.future.2017.11.022.
- Kshetri, N. 2017. "Can Blockchain Strengthen the Internet of Things?." *IT Professional* 19 (4): 68–72. doi:10.1109/MITP.2017.3051335.
- Li Da, X., and W. Viriyasitavat. 2019. "Application of Blockchain in Collaborative Internet-of-Things Services." *IEEE Transactions on Computational Social Systems* 1–11. doi:10.1109/TCSS.2019.2913165.
- Liu, B., X. L. Yu, S. Chen, X. Xiwei, and L. Zhu. 2017. "Blockchain Based Data Integrity Service Framework for IoT Data." In *Proceedings - 2017 IEEE 24th International Conference on Web Services, ICWS 2017*, 468–475. Chicago, IL, USA: Institute of Electrical and Electronics Engineers. doi:10.1109/ICWS.2017.54.
- Ouaddah, A., A. A. Elkalam, and A. A. Ouahman. 2016. "FairAccess: A New Blockchain-Based Access Control Framework for the Internet of Things." *Security and Communication Networks* 9 (18): 5943–5964. doi:10.1002/sec.1748.
- Perera, S., M. N. N. Samudaya Nanayakkara, S. S. Rodrigo, and R. Weinand. 2020. "Blockchain Technology: Is It Hype or Real in the Construction Industry?." *Journal of Industrial Information Integration* 17:100125. doi:10.1016/j.jii.2020.100125. Elsevier B.V.
- Reinhardt, I. C., C. O. Dr Jorge, and T. R. Dr Denis. 2020. "Current Perspectives on the Development of Industry 4.0 In the Pharmaceutical Sector." *Journal of Industrial Information Integration* 18 (June): 100131. doi:10.1016/J.JII.2020.100131.
- Samaniego, M., and R. Deters. 2016. "Using Blockchain to Push Software-Defined IoT Components onto Edge Hosts." In *ACM International Conference Proceeding Series*, 1–9. New York, New York, USA: Association for Computing Machinery. doi:10.1145/3010089.3016027.
- Shafagh, H., L. Burkhalter, A. Hithnawi, and S. Duquennoy. 2017. "Towards Blockchain-Based Auditable Storage and Sharing of IoT Data." In *CCSW 2017 - Proceedings of the 2017 Cloud Computing Security Workshop, Co-Located with CCS 2017*, 45–50. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3140649.3140656.
- Singla, A., and E. Bertino. 2018. "Blockchain-Based PKI Solutions for IoT." In *Proceedings - 4th IEEE International Conference on Collaboration and Internet Computing, CIC 2018*, 9–15. Institute of Electrical and Electronics Engineers, Philadelphia, PA, USA. doi:10.1109/CIC.2018.00-45.
- Tsang, Y. P., C. H. Wu, W. H. Ip, and W. L. Shiau. 2021. "Exploring the Intellectual Cores of the Blockchain–Internet of Things (Biot)." *Journal of Enterprise Information Management* 34 (5): 1287–1317. doi:10.1108/JEIM-10-2020-0395/FULL/PDF.
- Viriyasitavat, W., T. Anuphaptrirong, and D. Hoonsopon. 2019, May. "When Blockchain Meets Internet of Things: Characteristics, Challenges, and Business Opportunities." *Journal of Industrial Information Integration* 15: 21–28. doi:10.1016/J.JII.2019.05.002.

- Viriyasitavat, W., L. Da Xu, and A. Martin. 2012. "SWSpec: The Requirements Specification Language in Service Workflow Environments." *IEEE Transactions on Industrial Informatics* 8 (3): 631–638. doi:10.1109/TII.2011.2182519.
- Viriyasitavat, W., L. Da Xu, and W. Viriyasitavat. 2014. "Compliance Checking for Requirement-Oriented Service Workflow Interoperations." *IEEE Transactions on Industrial Informatics* 10 (2): 1469–1477. doi:10.1109/TII.2014.2301132.
- Viriyasitavat, W., L. Da Xu, B. Zhuming, and D. Hoonsopon. 2019a. "Blockchain Technology for Applications in Internet of Things - Mapping from System Design Perspective." *IEEE Internet of Things Journal* 6 (5): 8155–8168. doi:10.1109/JIOT.2019.2925825.
- Viriyasitavat, W., L. Da Xu, B. Zhuming, D. Hoonsopon, and N. Charoenruk. 2019b. "Managing QoS of Internet-of-Things Services Using Blockchain." *IEEE Transactions on Computational Social Systems* 6 (6): 1357–1368. doi:10.1109/TCSS.2019.2919667.
- Viriyasitavat, W., L. Da Xu, B. Zhuming, and V. Pungpapong. 2019c. "Blockchain and Internet of Things for Modern Business Process in Digital Economy - the State of the Art." *IEEE Transactions on Computational Social Systems* 6 (6): 1420–1432. doi:10.1109/TCSS.2019.2919325.
- Viriyasitavat, W., and D. Hoonsopon. 2018, July. "Blockchain Characteristics and Consensus in Modern Business Processes." *Journal of Industrial Information Integration*. doi:10.1016/J.JII.2018.07.004.
- Viriyasitavat, W., D. X. Li, B. Zhuming, and A. Sapsomboon. 2018, May. "Blockchain-Based Business Process Management (BPM) Framework for Service Composition in Industry 4.0." *Journal of Intelligent Manufacturing*. 1–12. doi:10.1007/s10845-018-1422-y.
- Viriyasitavat, W., and B. Zhuming. 2020. "Service Selection and Workflow Composition in Modern Business Processes." *Journal of Industrial Information Integration* 17 (March): 100126. doi:10.1016/j.jii.2020.100126.
- Wu, C. H., Y. P. Tsang, C. K. M. Lee, and W. K. Ching. 2021. "A Blockchain-IoT Platform for the Smart Pallet Pooling Management." *Sensors* 2021 21 (18): 6310. doi:10.3390/S21186310.
- Wu, N., L. Zhiwu, K. Barkaoui, L. Xiaoou, T. Murata, and M. Zhou. 2018. "IoT-Based Smart and Complex Systems: A Guest Editorial Report." *IEEE/CAA Journal of Automatica Sinica* 5 (1): 69–73. doi:10.1109/JAS.2017.7510748.
- Zhang, C., and Y. Chen. 2020. "A Review of Research Relevant to the Emerging Industry Trends: Industry 4.0, IoT, Blockchain, and Business Analytics." *Journal of Industrial Integration and Management*. World Scientific Publishing Co. Pte Ltd. 05 (1): 165–180. doi:10.1142/S2424862219500192.
- Zhang, P., Y. Kong, and M. Zhou. 2018. "A Domain Partition-Based Trust Model for Unreliable Clouds." *IEEE Transactions on Information Forensics and Security* 13 (9): 2167–2178. doi:10.1109/TIFS.2018.2812166.
- Zhang, Y., and J. Wen. 2017. "The IoT Electric Business Model: Using Blockchain Technology for the Internet of Things." *Peer-to-Peer Networking and Applications* 10 (4): 983–994. doi:10.1007/s12083-016-0456-1.
- Zhang, P., and M. Zhou. 2020. "Security and Trust in Blockchains: Architecture, Key Technologies, and Open Issues." *IEEE Transactions on Computational Social Systems* 7 (3): 790–801. doi:10.1109/TCSS.2020.2990103.
- Zyskind, G., O. Nathan, and A. Pentland. 2015, June. "Enigma: Decentralized Computation Platform with Guaranteed Privacy." *New Solutions for Cybersecurity*. <http://arxiv.org/abs/1506.03471>