Kuwait J.Sci., Vol.49, No.(1), January.2022,pp(1-27)

A resilient micro-payment infrastructure: an approach based on blockchain technology

Soumaya Bel Hadj Youssef, Noureddine Boudriga*

School of Communication Engineering
University of Carthage, Tunisia
**Corresponding author: noure.boudriga2@gmail.com*

Abstract

Resilient micro-payment infrastructures are critical assets to digital economy as they help protecting transactions and promote micro shopping. In this paper, we present a micro-payment infrastructure based on blockchain technology that is capable of decreasing the complexity of transactions' verification, reducing losses, and protecting against various cyber attacks. This infrastructure is user trust-aware, in the sense that it builds a trust function capable of providing real time management of the user's trust levels based on historic activity and then adapts the level of verification and risk of user's misconduct. Moreover, three different trust models are developed to provide different estimations of the tokens' block size to be submitted to the blockchain network for verification and management of the user waiting time. The micro-payment infrastructure provides different security services such as authentication, double-spending and double-selling prevention, tokens forging prevention, transaction traceability, and resilience to cyber-attack. In addition, its reactivity is improved through the reduction of the verification delay and user waiting time.

Keywords: Blockchain technology; micro-payment systems; payment security; risk management; user' behaviour.

1. Introduction

While a macro-payment system allows the processing of larger amount transactions, micro payment allows for the payment of low amount transactions online. Micro-payment systems constitute an attractive solution as they can be used to pay low cost goods and services including online gaming, software purchasing, multimedia downloading, online advertising, and access to information. Using a micropayment system, the consumer (or the buyer) can benefit from several advantages including the speed of access, wide availability of goods and services, and easy accessibility. The literature is abundant in the description of micro-payment systems. It contains a large selection of micropayment systems relying on cryptographic mechanisms, blockchain technology, and/or trusted third parties to authenticate and validate transactions. However, most of these proposals do not deal with non-repudiation, prevention from double-spending, and anonymity (Ali *et al.*, 2017). Hence, one can agree that the micro payment systems need for more security, higher efficiency, better reactivity, and provable privacy.

In particular, the information contained in the transactions (related to sellers and buyers) should be protected since it can be intercepted and tampered in an open network in some of the available solutions. Trust in a payment system should also be a pre-condition for using this system.

Blockchain is a distributed ledger that stores transactions between parties efficiently and in a verifiable, secure, and permanent manner. It is a promising technology for micro-payment due to various reasons including embedded cryptography, persistency, auditability, trust, anonymity, and disintermediation. In particular, new information can be added to the blockchain ledger only when most of the network participants give their approval after receiving proof of trust. Moreover, this information can only be appended to previous data; and once entered, it cannot be altered or lost. This proves incorruptible historical records and provides immutability and transparency. In addition to the aforementioned distinctive characteristics of the technology, blockchains offer enhanced security due to the embedded cryptography it allows for the exchanged information.

In recent years, blockchain technology has received noticeable attention and has emerged in many applications (Syed *et al.*, 2019; Zile & Strazdina, 2018) including data management, data verification, financial applications, healthcare applications (Dai *et al.*, 2018), Internet of things (Dai *et al.*, 2019; Ensor *et al.*, 2018), and business (Konstantinidis *et al.*, 2018; Lahkani *et al.*, 2020; Weking *et al.*, 2019), among other applications. For a blockchain-based system, a transaction can encapsulate any type of data and be appended to the ledger after its validity is verified. The Miners propose blocks (or sets of transactions) to be added to the chain and are responsible for checking that each added transaction is valid, and that the current block will refer to the correct hash of the previous block (Makridakis & Christodoulou, 2019). Making micropayments viable via Blockchain technology will open up a new world of online commerce. However, despite the interesting characteristics of the blockchain technology, it does not provide a global and fast infrastructure that allows detecting attacks and preventing loss. Moreover, the validation delay (or user waiting time) of each transaction in the blockchain network is relatively long, which may be unacceptable when dealing with micro-transactions. Nevertheless, one can be convinced that integrating blockchain technology in a micropayment infrastructure can help making this infrastructure efficient and providing various attractive properties. A survey of the literature shows that several works have combined the blockchain technology and other techniques to build micro-payment infrastructures (Pass & abhi shelat, 2015; Rezaeibagha & Mu, 2018; Wan *et al.*, 2019; Decker & Wattenhofer, 2015; Poon & Dryja, 2016; Zhang *et al.*, 2018; Heilman *et al.*, 2016; Xu *et al.*, 2016; Lundqvist *et al.*, 2017; Radhakrishnan & Krishnamachari, 2018; Chen *et al.*, 2019; Ramachandran *et al.*, 2019; Strugar *et al.*, 2018). However, the provided systems did not provide resilience and suffer from exhaustive delays.

On the other hand, one can be persuaded that adapting the verification time to the level of participant trust can be at the basis of the resilience of a micro-payment infrastructure and user's satisfaction. For this, micro-payment can benefit from a large set of research works related to the management of trust in many areas such as wireless sensor networks (Ye *et al.*, 2017; Chen *et al.*, 2017; Duan *et al.*, 2013; ZHAO *et al.*, 2019; Feng *et al.*, 2015; Che *et al.*, 2015; Yu *et al.*, 2012; Sun *et al.*, 2019), ad hoc networks (Alnumay *et al.*, 2019), dynamic distributed network environment (Zhang *et al.*, 2016), mobile crowd-sensing (Zupancic & Zalik, 2019), risk management, and automation (Yang *et al.*, 2017; Wang *et al.*, 2018). To the best of our knowledge, no assessment of the risk of loss has been addressed in the proposed micro-payment systems using the blockchain network, since they only relied on the response of the blockchain network. They focused on reducing the transaction costs, reducing the load on the Bitcoin network, and achieving anonymity and some security properties. In addition, they

did not pay attention to the behaviour of the user, did not provide appropriate actions against malicious users, did not adapt the response time of the blockchain network to the trust level of the user, and did not bring resilience and robustness to the transaction processing.

In this paper, we propose a resilient micro-payment infrastructure integrating blockchain technology while providing user trust-awareness by adding realtime management of a user trust function. The infrastructure is able: (a) to detect misbehaving users and attacks (such as double spending), (b) to provide robustness as it is able to analyze the risk of loss due to the detected events, (c) to reduce the payment verification delay, control the block size in the blockchain, and diminish the risk of loss related to false micro-payment, and (d) to respond to attacks in a fast and effective manner. In fact, it is capable of reacting quickly to malicious users (such as those performing attacks against authentication or generating false tokens) and taking appropriate actions by identifying attackers and punishing misbehaving. Thus, it is a trust-aware, efficient, and robust micro-payment infrastructure.

Our contribution is three-fold. First, we specify the architecture of the infrastructure and discuss the interaction between its functions and the flow of communication messages it allows to handle transactions management. Along with the blockchain network, the system involves entities such as an auditor, who is in charge of aggregating users´ tokens into blocks, submitting the blocks to the blockchain network, managing users' trust, and providing some sort of insurance (by covering the loss due to misbehaving and attacks). Second, we introduce the notion of user's trust and build a function that adapts the block size to the user trust level and the willingness of the auditor to take a risk. Third, we develop three models along with functions to evaluate the size of the block of transactions to be submitted to the blockchain network for validation and the risk to be handled by the auditor. Finally, the resilience of the infrastructure is checked through the proof of various features included in the list of infrastructure requirements.

The remainder of this paper is organized as follows. Section 2 provides a literature review of micro-payment schemes based on the blockchain technology. It also discusses the trust mechanisms and presents the requirements for an efficient, resilient, and robust micro-payment infrastructure. In Section 3, we describe our micro-payment infrastructure and we detail the management functions of tokens. Section 4 describes how the auditor manages the user's trust and maintains the tokens' block size. Section 5 discusses techniques for the assessment of risk in the infrastructure. In Section 6, we show how the infrastructure is protecting efficiently the payment process against the most common security attacks and prove its resilience features. Section 7 conducts a numerical simulation and discusses its results. The last section concludes our paper and gives some perspectives.

2. State of the art

In this section, we give a survey of the literature related to micro-payment systems using the blockchain technology and the trust paradigm. Then, we present the requirements for an efficient, robust, and trust-aware micro-payment infrastructure.

2.1 Micro-payment systems based on the blockchain technology

Several micro-payment systems using the blockchain technology have been proposed during the last decades. In particular, authors in (Pass & abhi shelat, 2015) proposed three different probabilistic cryptocurrency-based micropayment systems designed for Bitcoin and following the lottery-based approach. The first scheme is a fully decentralized solution and the other two schemes rely on a trusted third party. The second solution enables performing transactions with fast validation times, but it requires the intervention of a partially-trusted third party for

every winning transaction. The third solution is optimistic and is based on an invisible third party, which is only invoked when users are dishonest. The three schemes successfully reduce transaction cost of micropayments, but they provide little privacy and suffer from high computation cost, inflexible payment, and possibly unfair exchange.

In (Rezaeibagha & Mu, 2018), the authors proposed a micro-payment scheme based on blockchain and cryptographic tools (e.g., hash function, chameleon hash function, and digital signature) with the aim to reduce transaction costs and improving transaction processing speed. They adopted the tool of chameleon hash function and incorporated it in their micro hash chain. Moreover, they defined an adversarial model and proved its security. In their scheme, the payer commits a total amount of payment to the payee and then the payer generates 'microcoins' from this amount and pays to the payee with the microcoins for each micropayment transaction. A verification of the commitment is performed by the miners. In (Wan *et al.*, 2019), two efficient, flexible and fair micropayment schemes were proposed: a basic MicroBTC, which integrates the hash chain technique into Bitcoin transactions; and an advanced MicroBTC, which is based on a non-interactive refund technique, an efficient hash chain verification technique, and reduced verification cost. The implementation of the two schemes was based on the source code of Bitcoin by adding a new transaction type and a new operator.

In (Decker & Wattenhofer, 2015), the authors presented a protocol for duplex micro-payment channels, which guarantees end-to-end security thanks to the use of hashed time-lock contracts and allows final and instant transfers hence enabling real-time scenarios. These two channels are established between payment service providers and enable near-infinite scalability for digital payments based on Bitcoin. The blockchain was only involved during the setup and the closure of the channel to reduce the reliance and the load on the blockchain. The work in (Poon & Dryja, 2016) was based on the Lightning network, which creates off-chain micropayment channels between users with contracts encumbered by time-locks and hash-lock outputs and without the need of a trusted third party for validation. The users can execute transactions off-chain and only the final state is submitted to the blockchain network. It that case, the lightning network enables reduction of the load on the Bitcoin network and fast and near-instant transactions. It is worth noting that these works provide a certain degree of anonymity. Moreover, scalability issues are only solved for Bitcoin and not for other blockchains.

To provide anonymity, the authors in (Zhang *et al.*, 2018) proposed an anonymous off-blockchain scheme (AOM) designed for the face-to-face micro-payments in the real world aiming to enhance the anonymity of Bitcoin transaction system by merging multiple micropayments. The payer and the "honest-but-curious" intermediary generate puzzles based on the standard RSA encryption. By solving these puzzles, the payee can cash out the payment from the intermediary, and the latter can receive the payment from the payer and will randomly select the inputs of the merging transaction. AOM achieves strong unlinkability, unforgeability, resistance to some attacks, and ensures the correctness and fairness of transactions. Authors in (Heilman *et al.*, 2016) presented two main schemes for on-blockchain and off-blockchain (micropayment channel networks) bitcoin transactions by applying blind signatures (to achieve unlinkability) and smart contracts. Both schemes provide fair exchange security, forgery and double-spending prevention, resistance to DoS and Sybil attacks. Moreover, anonymity against malicious users and intermediary is achieved for the on-blockchain scheme. However, while anonymity against malicious users and honest-but-curious intermediary is achieved in the off-blockchain scheme, anonymity against a malicious intermediary is not achieved.

Others works have focused on the integration of micropayment with IoT. For instance, in (Xu *et al.*, 2016), the authors proposed a smart gas payment system designed for smart devices

in order to achieve automatic payment of the gas bills. This system contains an embedded bitcoin payment module containing a processor with a bitcoin wallet stored with it, an Elliptic Curves Cryptography (ECC) crypto chip, and a WI-FI module. Moreover, the authors designed a bitcoin payment protocol for transaction process. In (Lundqvist *et al.*, 2017), the authors presented a proof-of-concept allowing a smart cable to pay a smart socket for delivering electricity using Bitcoin Blockchain technology without any human interaction. Moreover, they presented a single-fee micro-payment protocol that aggregates multiple smaller payments incrementally into one larger transaction to alleviate the high transaction fees of cryptocurrencies like Bitcoin. It is worth noticing that this aggregation plays an important role in reducing the verification delay at the blockchain network.

On the other hand, authors in (Radhakrishnan & Krishnamachari, 2018) proposed a Streaming Data Payment Protocol (SDPP), which has three channels: a data channel for streaming data in real time (operated as a traditional client-server protocol), a payment channel for exchanging payments (implemented using a cryptocurrency protocol), and a records channel to store the transaction receipts (implemented using a distributed ledger technology (DLT)). In (Chen *et al.*, 2019), the authors presented PayFlow, a fine-granularity QoS micropayment system that allows end hosts in a software-defined network to make and pre-pay for guaranteed bandwidth reservations for their flows within the network for an arbitrary period of time. PayFlow uses a payment channel to make digital currency based micropayments and a record channel to store all relevant transaction records in an immutable ledger, which are similar to channels proposed in (Radhakrishnan & Krishnamachari, 2018). PayFlow introduces two additional channels (request, control). Furthermore, PayFlow has been implemented using OpenFlow, the IOTA cryptocurrency, and distributed ledger.

In (Ramachandran *et al.*, 2019), the authors introduced Micro-payments fOr Trusted vehIcular serVicEs (MOTIVE), a trusted and decentralized framework that allows vehicles to make peer-to-peer micropayments for data, compute and other services obtained from other vehicles or road-side infrastructure within radio range. MOTIVE incorporates a link prediction algorithm which allows the vehicles to calculate the contact duration based on the destination of the vehicles, speed, and the traffic conditions of the environment. Authors in (Strugar *et al.*, 2018) proposed a charging and billing mechanism for electric autonomous vehicles based on DLT. They used IOTA-based payment system and implemented a proof-of-concept with a Raspberry Pi and a temperature sensor. The proposed work ensures scalability, safety, efficiency, privacy and security, proof of delivery, and proof of payment.

2.2 Payee's trust management

It appears from the recent literature that trust management is an important security mechanism in dynamic network environments and that integrating trust is essential in the provision of users' behaviour differentiation. Several works have been conducted to develop effective methods of trust management. For instance, in (Ye *et al.*, 2017), an efficient dynamic trust evaluation model was proposed by dynamically adjusting the weights of direct and indirect trust and the parameters of the update mechanism for WSNs. The direct trust is calculated based on the Beta trust model and by taking communication trust, data trust, and energy trust into account with a punishment factor and regulating function. The indirect trust is invoked conditionally. In (Chen *et al.*, 2017), the authors proposed a trust evaluation model (behaviour trust, data trust, and historical trust) and data fusion mechanism based on trust. The model is used to construct the trust list and guide the process of data fusion. In (Duan *et al.*, 2013), authors proposed a distributed and fine-grained access control model based on the trust and risk degree.

Authors in (ZHAO *et al.*, 2019) proposed an exponential-based trust and reputation

evaluation system. An exponential distribution is applied to express the trust and reputation of WSN nodes. The trust of a node is used to look for reliable nodes to transmit data and weaken malicious attacks within the WSNs. In (Feng *et al.*, 2015), a credible Bayesian-based trust management scheme was proposed. The overall trust value is aggregated by both direct and indirect trust information. The direct trust is calculated by a modified Bayesian equation and updated by a sliding window. Authors in (Che *et al.*, 2015) proposed lightweight trust management, based on Bayesian and Entropy. The evaluated node's direct trust value was calculated by Bayesian and periodically updated according to the combination of effective history records and adaptive decay factor. In the three above works, the indirect trust computation is invoked conditionally according to the uncertainty of direct trust calculated via Entropy Theory.

Authors in (Alnumay *et al.*, 2019) discussed a quantitative trust model for an IoT-MANET. A Beta probabilistic distribution was used to combine different trust evidences and direct trust was calculated. The theory of ARMA/GARCH was used to combine the recommendation trust evidences and predict the resultant trust value of each node in multi-step ahead. Further, the authors designed a routing protocol to ensure the secure and reliable end-to-end delivery of packets. In (Zhang *et al.*, 2016), the authors presented a novel trust update mechanism based on time sliding-window for trust management system. In addition, a fast-fall and slow-rise updating pattern and a time-based forgetting factor were designed to control the trust decay rate and improve the evaluation accuracy. Authors in (Zupancic & Zalik, 2019) proposed a conceptual trust framework for mobile crowd-sensing systems, including a novel method that considers different user behaviours and is based on a comparison of users' trust attitudes by applying nonparametric statistic methods.

2.3 Requirements for a resilient micro-payment infrastructure

To be resilient, a micro-payment infrastructure should comply with several requirements. Among the important requirements, we mention the seven following requirements.

Tokens' aggregation: Checking every token separately is time wasting. Therefore, it is essential that the verification should be made after aggregating the tokens related to a single transaction into blocks. However, using large size blocks may lead to an unacceptable risk of money loss or user dissatisfaction in the case of low amount payments. Hence the aggregated block needs to have adaptive sizes.

Double-spending prevention: Double spending can be noticed when the same token is used by the same user in different payment transaction. Detecting and rejecting any type of double spending should be provided by robust micro-payment infrastructure. The client should use a token only once to pay items. This can be guaranteed when the infrastructure is able to track tokens and user identities.

Double-selling prevention: Double selling is a fraud performed by a merchant using the token submitted by a client in multiple requests of reimbursement or fake transactions. A resilient micro-payment infrastructure should be able to detect such unacceptable acts. For this, the payment infrastructure should be able to record the history of any token involved in any payment transaction. Information such as token owner, merchant receiving the token, and item bought using the token can be important to detect and prevent double selling.

Tokens forging attack prevention: Forging attacks can target tokens by launching modifications during their transfer to the merchant, the formation of blocks or the transmission of token blocks. To detect and prevent forging, the payment infrastructure should be able to identify modifications, recognize false tokens, and authenticate tokens. Each generated token should be verified and validated before payment process proceeds.

Authentication of payment transaction: Authenticating a transaction means that tokens occurring in the transaction, actors involved in the transaction, and the transaction content should be authenticated to prevent false transaction, client masquerading, and transaction replays. A resilient infrastructure should be able to authenticate tokens in all steps of micro-payment including token delivery to the client, token transfer, and token reimbursement. Authentication of a transaction assumes that the transmitter and the receiver it contains should be checked.

Payment tracing: Micro-payment resilient infrastructures should be able to track all transactions so that histories cannot be disconnected. This can be achieved through the blockchain network and by including the timestamp on every action. Token payment tracing can be achieved by including the identity of the actors and the product number in every token (or block of tokens). Moreover, time related issues such as generation of the token and arrival time of the token at the buyer, seller, auditor, and blockchain network can be added.

Actors' trust management: As resilience aims at reducing the risk of loss (of any form) and that trustful clients generate very low risk (or no risk), micro-payment infrastructure should provide tools for managing trust functions that are able to adapt the duration of validation to the trust level of the actors. For this, the infrastructure can include a trusted third actor to handle the risk.

The aforementioned survey shows that most of the solutions provided for micropayment have focused on reducing transaction costs and improving transaction speed (Rezaeibagha & Mu, 2018), achieving instant transfers (Decker & Wattenhofer, 2015), reducing the load on the Bitcoin network (Poon & Dryja, 2016), providing anonymity (Zhang *et al.*, 2018; Heilman *et al.*, 2016), and alleviating the high transaction fees (Lundqvist *et al.*, 2017). However, one can say that these solutions did not provide resilience. In particular, most of them did not integrate real time management of the user's trust, did not manage user's profiles, nor did they follow the historic activity of users (i.e., use of tokens). These micropayment systems did not deal with user trust and did not build trust functions serving as a basis for dealing with malicious behaviours as they only rely on the result of the blockchain network (transaction accepted or not). In addition, these solutions did not provide tools for the estimation of risk of loss due attackers or user misbehaviour and did not adapt the duration of transaction validation to the behaviour of the payees, while encouraging trustful client by reducing significantly their waiting time for approval. Finally, one can be convinced that the aforementioned works did provide tools for loss coverage and did not define actors capable of playing the role of insurer.

While the solution we build in the next sections complies with the above resilience requirements, it also provides tools for risk analysis, loss coverage, and block size adaptation. That is the main reason for integrating the auditor entity, which manages and assess the risk of loss, controls the behaviour of payees, reacts against malicious users, takes appropriate actions against them (if needed), and plays the role of an insurer. Our approach allows adapting the blockchain's response time to the trust level of the user. This can be at the basis of the resilience of a micro-payment infrastructure.

3. Trust-aware micro-payment infrastructure using blockchain

In this section, we describe the architecture of our trust-aware resilient micro-payment infrastructure, which we refer to as μPIB. For this, we describe its entities, the functions they implement and the messages they exchange during transaction management. This architecture is depicted in Figure 1.

3.1 μPIB entities

The major entities of μPIB are the users, the merchants, the auditor, the blockchain network, and the bank.

The user (buyer or payee) is a customer who makes low cost purchases from the vendors using tokens. The user is responsible for the collection of tokens from the bank and the submission of tokens to pay the items he bought from the vendors.

The vendor's main role is to sell its goods (products or services) to the users. He is responsible for collecting the tokens related to a purchase, forming a payment transaction including the received tokens, the identity of the user, his identity, and useful information related to the purchase (nature and price of the good, for example), submitting the transaction to the auditor for token verification and payment, and delivering the goods.

The auditor is a trusted third-party that manages the tokens queues of users and token payment. Each queue corresponds to one user and is built up using the tokens received from vendors for the same user. The auditor is responsible for building blocksz of tokens (for the same user) and submitting them to the blockchain for verification. A token block can contain all the tokens related to a given transaction, a part of it, or the tokens related to more than one transaction, depending on the size defined based on rules considering the users' behaviour. In addition, the auditor manages the user's trust value and proceeds to the payment of the vendors for the received tokens. He also serves as an insurer for loss.

The Bank is responsible for generating and delivering tokens to the buyers. It provides two main other actions: a) paying the auditor for the validated tokens; and b) publishing the generation of tokens with the blockchain network. Moreover, the bank can implement a policy of token delivery to the users according to the user's trust and the information received about the payment.

The blockchain network represents a peer-to-peer network of independent nodes (e.g., servers) communicating together. Their main role consists in verifying the validity of the tokens received from the auditor. To perform validation, the blockchain network uses the information provided by the bank when generating tokens.

3.2 µPIB functions and entity interactions

The major μPIB functions include token generation, micro-payment transaction generation and processing, block forming, trust management, block size management, and token payment. As shown in Figure 1, these functions collaborate through messages exchange.

Token generation: A user requesting a set of tokens can acquire them from a bank using a bank account or presenting credentials for authentication (to support tracing, for example). On receiving the user request, the bank generates the set of tokens and delivers them to the user (Message 1 as depicted in Figure 1). For the sake of protection, each token has a unique identifier s. It is represented by a data structure containing signed information involving a unique serial number, the token value, the issuer bank B, the timestamp $t_{B,s}$ of the bank (or generation time), and the bank signature Sig_B that guarantees the correctness of the token. In the sequel, the tokens are assumed to have a same value v. Formally, the data structure of token $tok_{B,s}$ is given by:

$$tok_{B,s} =< Cert_B, s, v, t_{B,s}, Sig_B\left(s, v, t_{B,s}\right) > \tag{1}$$

where $Cert_B$ is the digital certificate of B. On receiving the tokens, the user verifies the token (authenticity and timestamp) and the certificate validity; and the bank publishes the tokens in the blockchain network. The verification of the token authenticity is performed thanks to the digital signature.

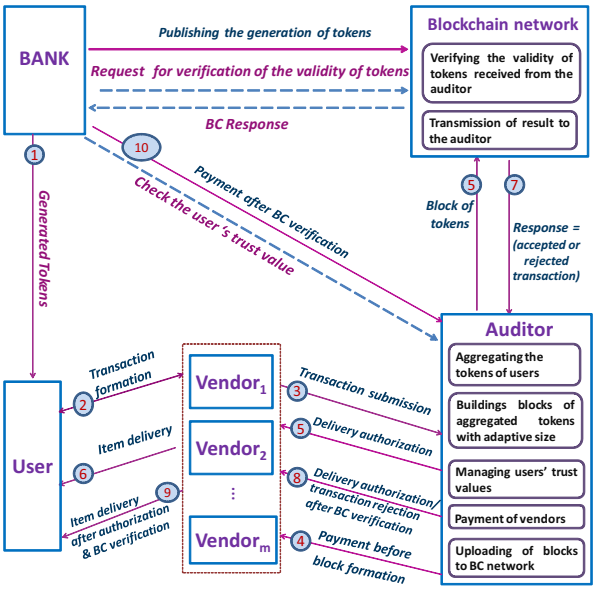

Fig. 1. Micro-payment infrastructure

Micro-payment transaction management: When a user u wants to buy a service or an item from a vendor m, he starts building a transaction with the vendor (Message 2 as shown in Figure 1). The data structure representing the transaction contains information such as the user identifier, the merchant identifier, information about the item to buy, the tokens involved, and the time of transaction construction. For this, the digital certificates of the user and the vendor can be used. In that case, the data structure is given by:

$$Tr_{u,m,t} =< Cert_u, Cert_m, t, TOK, INF, Sig_m\left(Sig_u(t, TOK, INF)\right) > \quad (2)$$

where TOK is a set of tokens covering the price of the item to buy, INF is the information related to the item, $Cert_u$ is the certificate of user u, $Cert_m$ is the certificate of vendor m, t is the time of transaction creation, Sig_u (respectively, Sig_m) is the signature of u (respectively, m) over INF, TOK, t. The transaction is presented to the auditor using Message 3, as depicted in Figure 1, after verification, if needed. In particular, the vendor verifies the tokens' timestamp and the user certificate. In addition, the vendor may check the user trust value (as built by the auditor). Finally, after receiving a response from the auditor (Message 5 as shown in Figure 1) authorizing it to deliver, the vendor delivers the sold item immediately to the user (Message 6 as shown in Figure 1).

Block management: When the auditor receives a transaction from a vendor, he accomplishes many actions: i) he verifies the received message using certificates and signatures; ii) he extracts the tokens from TOK and inserts them into the file corresponding to the user; iii) if the file contains n tokens, he forms a block of n tokens and sends it to the blockchain network for verification (Message 5 as depicted in Figure 1); otherwise, he informs the vendor to start the item delivery (Message 5); and iv) on receiving the response from the blockchain network (Message 7 as shown in Figure 1), he sends a message to the vendor authorizing the delivery or rejecting the transaction (Message 8 as depicted in Figure 1). Both messages are signed by the auditor using his certificate $Cert_{Au}$. In case of delivery authorization, the vendor delivers the sold item immediately to the user (Message 9 as shown in Figure 1). The current size n of the block is maintained using the trust function. It is discussed in the following section.

It is worth noticing that the block can be in two states: block under building or block under verification. While the transaction is successfully executed in first case, the transaction can be rejected in the second case, while the user is waiting for verification.

Block verification: Upon receiving a block of tokens from the auditor, the blockchain network starts the verification and the validation process. It verifies the authenticity and the timestamp of each token in the block, and checks the validation of the auditor certificate, using the information published by the bank about the tokens in the received block along with the content of the different blocks previously sent by the auditor. The blockchain network rejects a token in the three following cases: i) the token is altered (i.e., the content of the token does not fit with the bank signature on the token); ii) the token is a duplicate (i.e., the occurred in a previous block); and iii) the token does not occur in the bank publication.

When the verification is completed, the blockchain sends a signed message (message 6 as depicted in Figure 1) to the auditor containing the following data structure:

$$< (tok_{B,s_1}, d_{s_1}), ..., (tok_{B,s_n}, d_{s_n}) >$$

where n is the size of the block of tokens under processing, s_i is the identifier of the ith token in the block, B is the bank that has generated the tokens, d_{s_i} is the validation decision related to token tok_{B,s_i}. The decision d_{s_i} is equal to 11 if the token is valid. Elsewhere (i.e., if it is invalid), it is set to 01,10, and 00. d_{s_i} is equal to: 01 if the token is altered; 10 if the token is a duplicate; and 00 if the token does not occur in the bank publication. Then, the blockchain network sends the result of verification to the auditor which consists in a block containing the states of tokens.

Token reimbursement and transaction payment: Two situations can occur depending on the state of the block. If a transaction tr is received by the auditor and a block cannot be submitted to the blockchain network yet, the merchant proceeds with the item delivery and the auditor will redeem all the tokens in the transaction. In that case, the merchant is assumed to receive $size(tr) \times (v - \rho)$, where $size(tr)$ is the number of tokens in the transaction, v is the token value, and ρ is a value compensating auditor risk (i.e., withholding of payment); the cost of transaction can be considered as equal to $size(tr) \times \rho$. On the other hand, if the transaction is followed by a block formation, the vendor has to wait for the validation result of the transaction and cannot deliver any item to the user. If the block is valid, the transaction tr is accepted and the merchant receives $size(tr) \times (v - \rho)$. If one token in the block is invalid, the transaction is rejected and no item is received by the buyer.

In the last case, let $< (tok_{B,s_1}, d_{s_1}), ..., (tok_{B,s_n}, d_{s_n}) >$ be the answer of the blockchain network telling that l tokens are invalid and $n - l$ tokens are valid. Among the invalid tokens let us assume that l_0 tokens occur in tr. Then, the auditor will lose an amount equal to $(l - l_0) \times (v - \rho)$, because he already paid them before block formation (Message 4 as shown in Figure 1). The definition of the trust function takes into account the risk of loss caused by the user. It is addressed in the sequel.

Finally, let us mention that before proceeding to the generation of tokens (Message 1 as depicted in Figure 1) or the payment of the tokens (Message 10 as shown in Figure 1) to be presented by the auditor, the bank may need to check the user trust value, get knowledge of the user behaviour, and request information about the history of a token. To do so, messages can be sent by the bank to the auditor and the blockchain network.

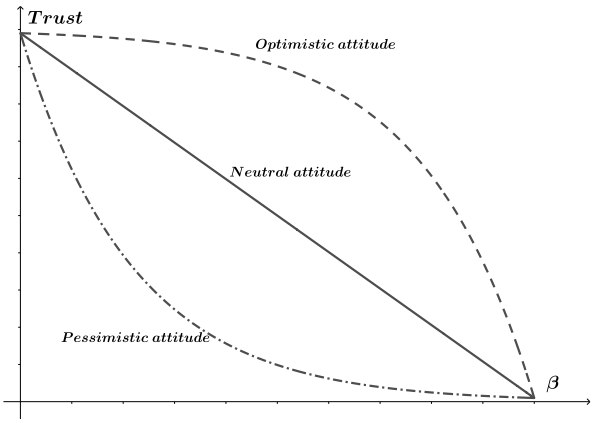

Fig. 2. Trust models forms

4. Auditor trust management in µP IB

In this section, we show how the auditor manages the trust values associated with a user and how he maintains the block size based on the user behaviour and the auditor profile. For this, three auditor profiles will be considered. They are the risk neutral auditor, the risk averse auditor, and the risk lover auditor.

4.1 Modeling the auditor profiles in µP IB

As the main functions of the auditor aim at aggregating the tokens, managing the blocks that have to be transmitted to the blockchain network, and analyzing the risk of loss incurred by a block of transactions, it is worth assuming that trust should vary with risk and that the block size should get smaller if the user attempts more invalid transactions. However, the reaction of the auditor to user behaviour can be stronger, softer to reduce the block size, or neutral as he can reduce the size linearly.

User trust is dynamic and depends on the risk of loss, as it is not fixed and it changes over time with the actions/purchases performed by the user. For instance, trust decreases when the user does not behave correctly. It can be computed/modified after direct observations made by the auditor on the user after the submission of transactions.

For a new user, the auditor selects an initial value of the block size W_0 depending on the information delivered by the bank, the profile of the user, and the experience of auditor. Then, it computes the initial trust value assigned to the user. The user's trust value will be recomputed after reception of each result related to the submission of a block to the blockchain network. Now, let us denote by W the size of a block, and assume that $W = \alpha + \beta$, where α and β denote the number of valid and invalid tokens, respectively. The trust value at the construction of the (i)th block depends on the size of the $(i-1)$th block and the number of invalid tokens β_i involved in the result received from the blockchain network.

Three types of functions can represent the user's trust models and hence the profile of the auditor. In the optimistic model, the auditor decreases the user's trust slowly with respect to the increase of β (i.e., the auditor behaves well toward the dishonest users). In the pessimistic model, the user's trust decreases rapidly with β (i.e., the auditor barely tolerates dishonest users). On the other hand, the neutral model shows a linear growth/decrease of the trust value. As depicted in Figure 2, the three models have the same start and end points. The optimistic trust model is represented by a decreasing convex function, while the pessimistic trust model is represented by a decreasing concave function.

Table 1. Notations

Notation	Description
W	Tokens Block Size
α	Number of valid tokens
β	Number of invalid tokens
e	$0, 1, 2$
$T_0(.)$	Neutral trust function
$T_1(.)$	Optimistic trust function
$T_2(.)$	Pessimistic trust function
u	User
i	Number of block
ρ	withholding on token payment
v	Token value

In what follows, we present and detail the three trust functions $T_e(.)$ related to the aforementioned user profile types. The next table 1 specifies the notations used in the computation.

The functions describe the auditor perception of the trust value assigned to a user after receiving the response from the blockchain connected to a block of W_{i-1} transactions related to the user.

For the neutral profile, the trust value of user u computed by the auditor, according to the beta distribution, can be expressed as $E(beta(\alpha+1, \beta+1)) = \frac{\alpha+1}{\alpha+\beta+2}$, where $E(.)$ is the expectation. Hence, the neutral profile is represented by a linear trust function $T_0(.)$ as follows:

$$T_0(\beta_i) = \frac{(W_{i-1} - \beta_i) + 1}{W_{i-1} + 2} \tag{3}$$

The trust function for the optimistic profile $T_1(.)$ can be expressed by an exponential trust function:

$$T_1(\beta_i) = 1 - \gamma_2 \times \exp(-\delta_2 \times (W_{i-1} - \beta_i)) \tag{4}$$

where $\gamma_2 = \frac{W_{i-1}+1}{W_{i-1}+2}$ and $\delta_2 = \frac{\log(1+W_{i-1})}{W_{i-1}}$.

The trust function for the pessimistic profile $T_2(.)$ can be represented by an exponential trust function:

$$T_2(\beta_i) = \gamma_1 \times \exp(\delta_1 \times (W_{i-1} - \beta_i)) \tag{5}$$

where $\gamma_1 = \frac{1}{W_{i-1}+2}$ and $\delta_1 = \frac{\log(1+W_{i-1}}{W_{i-1}})$.

It is worth noting that we have:

- $T_0(\beta = 0) = T_1(\beta = 0) = T_2(\beta = 0) = \frac{W_0+1}{W_0+2}$. This value of trust corresponds to the initial and higher value of trust. In fact, at the beginning, we assign a higher value of trust to the user and we assume that all the tokens are valid.

- $T_0(\beta = W) = T_1(\beta = W) = T_2(\beta = W) = \frac{1}{W+2}$. This value of trust corresponds to the lower value of trust. It is computed when all the tokens are invalid.

Moreover, one can say that $T_2(.)$ is decreasing since its first order derivative is negative and that $T_2(.)$ is concave since its second order derivative is positive on $[0, W]$. In addition, $T_1(.)$ is

decreasing because its first order derivative is negative and $T_1(.)$ is convex because its second order derivative is negative for $\beta \in [0, W]$. Finally, one can easily show that, for a given W, we have the following double inequality:

$$\forall \beta : T_2(\beta) \leq T_0(\beta) \leq T_1(\beta) \tag{6}$$

The finding of the above equation conforms to the results shown in Figure 2. Moreover, the convex trust function $T_1(.)$ characterizes optimistic attitude of the auditor and the concave trust function $T_2(.)$ characterizes pessimistic attitude. A linear trust function $T_0(.)$ characterizes neutral attitude of the auditor. As shown in Figure 2, the convex trust function is never below the linear trust function and the concave one is never above it.

4.2 Block size management in μPIB

In this section, we show how μPIB is trust aware. We first detail how the auditor maintains the size of the blocks of tokens it sends to the blockchain network using the auditor trust profiles. The auditor modifies the tokens block size W according to the new value of the auditor profile function.

The main idea behind our trust mechanism is to punish the dishonest users by reducing the block size (i.e., increasing the waiting time), while encouraging the honest users. To do so, the auditor applies a four-step process as follows:

- Step 1. For a given user u, the auditor selects an initial value of the block size $W_{0,u}$, a withholding of payment ρ and a trust model $T_e(.)$. This can be done based on the will of the auditor, the profile of the user, the history of payments made by the user, and the information delivered by the bank, if any.
- Step 2. At the ith construction of a block for user u, let $W_{i-1,u}$ be the size of the tokens block decided at the construction of the (i)th block. The auditor keeps receiving the tokens of the user sent by the merchants until the $(W_{i-1,u})$th token. Then, it sends the formed block to the blockchain network for verification, and informs the merchant that the payment is stopped until receiving a response from the blockchain network.
- Step 3. On receiving the verification result of the ith block, the auditor pays $W_{i-1,u} \times v \times (1-\rho)$ to the vendor, where v is the value of a token and ρ is the withholding on token payment, provided that the (i)th block is declared valid by the blockchain network.
- Step 4. If the (i)th block is invalid, the auditor deduces the number β_i of invalid tokens in the (i)th block (of length $W_{i-1,u}$) and computes a new trust value $T_e(\gamma_i)$, where $\gamma_i = \sum_{j=1}^{i} \beta_j$ is the sum of the number of invalid tokens in the previous blocks, taking into consideration the history of payment related to user u. Then, the auditor computes the solution x_i of the following equation:

$$\frac{T_e(\gamma_i) - T_e(\gamma_{i-1})}{T_e(\gamma_{i-1})} = \frac{x_i - W_{i-1,u}}{W_{i-1,u}} \tag{7}$$

and sets the new block size $W_{i,u}$ as the highest integer that is lower than x_i, meaning that $W_{i,u} = \lfloor x_i \rfloor$, where $\lfloor - \rfloor$ is the flour function.

Since $T_e(\gamma_i) = T_e(\gamma_{i-1} + \beta_i)$ and that $T_e(.)$ is decreasing, one can deduce that $W_{i,u} \leq W_{i-1,u}$. More generally, we have the following result.

Proposition 1. *Let $W_{0,u}$ be the initial value of the block size, for user u, and $T_e(.)$ is the trust model applied by the auditor to user u. Then, the following statements hold*

1. *The sequence $(T_e(\gamma_n))_{n\geq 0}$ is a decreasing series. It decreases towards 0 if there is enough strictly positive γ_n. In that case, let n_e be the least integer n such that $T_e(\gamma_n)$=0. Then,* $W_{n_e,u} = 0$.

2. *Let n_0, n_1, and n_2 be the least integers such that $T_0(n_0) = T_1(n_1) = T_2(n_2) = 0$. Then,* $n_2 \leq n_0 \leq n_1$.

Proof. The first statement can be deduced recursively from the fact that, if $x, y > 0$, $x < W_{0,u}$, and $y < W_{0,u} - x$; then, for $e = 0, 1, 2$ we have:

$$T_e(x) > T_e(x + y) \tag{8}$$

It is worth to mention that the first term of above inequality (8) gives the trust value related to a block of tokens of size W containing x invalid tokens. The second term gives the trust related to a block of tokens containing $W - x$ tokens and having y invalid tokens, or the trust related to a block of tokens containing W tokens and having $x + y$ invalid tokens.

We prove now that $W_{n,u} = 0$ for large values of n. For this, let us first show that there is $A_e > 0$ such that:

$$\forall x \geq 0, y \geq 1 : T_e(x) - T_e(x + y) > A_e.$$

The following inequality can be set up:

$$\forall x \geq 0, y \geq 1 : T_e(x) - T_e(x + y) > T_e(x) - T_e(x + 1) \tag{9}$$

It is sufficient to show that it is true for $y = 1$. For this, let us consider the function $f(x) = T_e(x) - T_e(x + 1)$. It is easy to show that the derivative of f is negative and that $f(x) \geq f(W_{0,u})$ for all $x \geq 0$. Thus, we can choose $A_e = f(W_{0,u})$.

We can deduce that the sequence $T_e(n)$ is decreasing for $n \geq 1$. Since it is positive, it should converge towards a number $\alpha \geq 0$. α should be null, otherwise let n_∞be the first integer such that $T_e(n_\infty) = \alpha$. Thus, we have:

$0 = T_e(n_e) - T_e(n_e + 1) \geq A_e > 0$,

which leads to a contradiction.

Using equation (7), we deduce that $W_{n_e,u} = 0$ or $W_{n_e,u} \neq 0$. If the second case is feasible, then the auditor is able to form an (n)th block containing $W_{n_e,u}$ tokens including one false token from the user and send it to the blockchain network. The response he receives allows him to deduce that $\beta_{n_e+1,i} = 1$ and that $0 \leq T_e(\gamma_{n_e+1,i}) < T_e(\gamma_{n_e,i})$, which is a contradiction.

The second statement comes from the inequality $T_2(\beta) \leq T_0(\beta) \leq T_1(\beta)$ showing that with pessimistic model, the trust reaches 0 before the linear trust does and that the optimistic model reaches 0 after the linear trust does.

5. Risk assessment in µPIB

The risk can be defined as the possibility that the auditor loses money due to the increase of the number of invalid tokens in the different blocks. After the reception of (n)th block of $W_{n-1,u}$ tokens, the auditor automatically computes a representation of the risk. The latter depends on the number of valid tokens (α_n), the number of invalid tokens (β_n), the withholding on each token payment (ρ), and the token value (v). Let $Rsk_{n,u}$ be the risk value after the verification of the (n)th block validation. It can be represented by the difference between the amount of payment made to the merchant and the amount received from the bank for the valid tokens in the (n)th block.

To compute $Rsk_{n,u}$, we first define some useful parameters. Let $tr_{n,1}, ..., tr_{n,p}$ be the transactions submitted before the block is completed and let $\mid tr_i \mid$ be the number of tokens in transaction tr_i. Then, on receiving the validation result of the (n)th block, two cases can happen. If the block is valid, then all the tokens are valid and the auditor pays an amount equal to $W_{n-1,u}v(1-\rho)$ and receives an amount equal to $W_{n-1,u}v$. In that case, the risk $Rsk_{n,u}$ is given by:

$$Rsk_{n,u} = W_{n-1,u}v(1-\rho) - W_{n-1,u}v = -W_{n-1,u}v\rho \tag{10}$$

This shows that the value of risk is negative meaning that the auditor is gaining some amount which is equal to $W_{n-1,u}v\rho$. In fact, the more is the value of the token v or the withholding on each token payment ρ or the size of the $(n-1)$th block the more is the profit of the auditor. It is worth noting that in this case the size of the (n)th block is the same as $(n-1)$th block (i.e., $W_{n,u} = W_{n-1,u}$).

If the (n)th block is invalid then let $\beta_{n,u,1}, ..., \beta_{n,u,p-1}$ be the number of invalid tokens in $tr_1, ..., tr_p$, respectively. In that case, the auditor rejects transaction tr_p. He pays to the merchant an amount equal to $\sum_{i\leq p-1} \mid tr_i \mid v(1-\rho)$ and receives from the bank an amount equal to $\sum_{i\leq p-1}(\mid tr_i \mid -\beta_{n,u,i})v$. Therefore, the instant risk $Rsk_{n,u}$ is given by:

$$Rsk_{n,u} = \sum_{i\leq p-1} \mid tr_{n,i} \mid v(1-\rho) - \sum_{i\leq p-1} (\mid tr_i \mid -\beta_{n,u,i})v \tag{11}$$

Therefore, the risk $Rsk_{n,u}$ can be expressed as follows:

$$Rsk_{n,u} = v(\sum_{i\leq p-1} \beta_{n,u,i} - \rho \mid tr_i \mid) \tag{12}$$

In addition, the long time risk (or whole loss) related to user u at instant n is equal to $Rsk_{n,u} = \sum_{i\leq p-1} Rsk_{i,u}$. The following result computes the above expression for particular types of transactions and user behaviour.

Proposition 2. *Using the above notations, let us suppose that all transactions have the same number t of tokens, the transactions submitted by user u has at most one invalid token, and the π probability that a transaction contains one invalid token. Then, the average risk of $Rsk_{n,u}$ related to user u is given by:*

$$\overline{Rsk}_{n,u} = v(-t\rho + \pi)(\left\lceil \frac{W_{n,u}}{t} \right\rceil - 1) \tag{13}$$

where $\lceil - \rceil$ is the ceiling function. In particular, the average risk is null, provided that $t\rho = \pi$.

Proof. This result comes from the following facts. First, the average number of invalid tokens in the first $p-1$ submitted transactions is equal to $\pi(p-1)$. Second, the auditor receives from the bank an amount equal to $((p-1)t - \pi(p-1))v$. Third, the auditor pays to the merchant an amount equal to $t(p-1)v(1-\rho)$. Finally, p is equal to $\left\lceil \frac{W_{n,u}}{t} \right\rceil$.

Now let us discuss how the instant risk varies with respect to the variations of $W_{0,u}$, ρ, and probability π.

Using Equation 13, we can deduce that the risk instant average risk increases when $W_{0,u}$ increases, since the latter induces the increase of $W_{n,u}$, provided that other are fixed. Moreover, when $t\rho \geq 1$, then the auditor cannot lose. On the opposite, the average instant risk increases when π increases, since $(-t\rho + \pi)$ decreases and starts being positive. Finally, when the size of the transactions gets larger the risk gets lower and the auditor starts gaining when the $t\rho$ becomes higher than 1.

Using Equation 12, we can deduce that v amplifies the risk $Rsk_{n,u}$. In addition, the risk increases if the number of invalid tokens gets larger, since $\beta_{n,u,i} - \rho \mid tr_i \mid$ gets larger in that case.

6. Proof of infrastructure resilience

In this section, we first show how the infrastructure is protecting efficiently the payment process against the most common security attacks targeting blockchain networks. Then, we discuss the resilience features of the infrastructure based on the requirements we presented in Subsection 2.3.

6.1 Potential threats on blockchain

In this subsection, we discuss how μPIB detects and prevents the most common attacks that can target blockchain networks. Four attacks can be highlighted.

The 51% attack: This attack may occur when a single miner (or mining pool), which has exceptionally more computational resources (i.e., more than 50%) than the rest of the network, dominates the verification and approval of transactions and controls the content of a blockchain. The attacker (i.e., dominant miner) can perform double spending (i.e., spend same coins many times), reject transactions, insert fraudulent transactions, reverse transactions, prevent new transactions from gaining confirmations, or even steal asset from others. This attack can be considered as a major threat for blockchain networks because it has the power to destroy the stability of the whole networks.

Since the access to the blockchain in μPIB is only performed by the auditor and the banks, attacks performed by 51% of users cannot succeed, since the auditor is able to stop the submission of transactions for validation, for a particular user, when block size reaches zero. With a maximum of $W_{0,u}$ attempts performed by user u, the transaction for u will be stopped. This means that even when all the users attempt attacks, the payment process is not affected. This feature comes with a cost: the auditor will have to cover the loss.

Distributed Denial-of-Service (DDoS): In DDoS, multiple attackers launch the attack simultaneously unlike DoS attack, where a single attacker performes the attack. The DDoS attack is used to make resources unavailable to network participants by flooding with large traffic in a distributed way (e.g., submitting invalid transactions, and transmitting a large number of invalid blocks to the blockchain network). If multiple users attempt DDOS and create many transactions, the merchants will be able in the near future to block selling and submitting the transactions of any user involved in the DDOS. In fact, a transaction for any user u attempting the DDOS are stopped at most $W_{0,u}$ transaction invalidations for u.

Sybil attack: In this attack, an attacker (i.e., called Sybil node) can create multiple virtual identities to take control over the whole blockchain network and that cause severe impact in public blockchain (i.e., permissionless) by creating a large number of fake user accounts. These fake nodes can corrupt the network in order to validate unauthorized transactions and alter valid transactions. Moreover, they can disconnect the genuine nodes from the blockchain network and can act like genuine nodes. This may launch several other attacks such as DoS and DDoS.

False identities can be detected by the merchants and the auditor based on digital certification. Indeed, if the certificate verification is performed after every received message and the PKI infrastructure is robust, then fake accounts cannot be undetected.

Eclipse attack: In this attack, the adversary controls a sufficient number of IP addresses in order to monopolize all of the victim's incoming and outgoing connections, and thus isolates the victim(s) from the rest of the network. Then, the adversary can force the victim to waste its resources (e.g., computing power) on obsolete views of the blockchain.

Eclipse attacks can be performed with our system, as the attacks can take place on the communication network connecting the merchant and the auditor system, but not on μPIB. Robust links between the merchants and the auditor, between the bank and the blockchain network, and between the auditor and the blockchain network can prevent such attacks.

6.2 µPIB resilience and robustness

Six features related to resilience are taken care of in μPIB. We discuss these features in the following.

Double-spending prevention: Double spending can be attempted by the user. However, this attack is detected by the nodes of the blockchain network during the verification process. Indeed, the nodes verify whether a token has been spent or not. In addition, in our protocol, a unique identifier allowing to identify each token, certificates of the payer (or user) and the payee (or vendor) are added to each token included in a transaction. Moreover, a transaction containing a token used in a previous transaction will be rejected by the blockchain and this will lead the auditor to reduce the size of a future block generated for the user providing the fake token. This helps the auditor to detect double spending and take appropriate decisions according to the behaviour of the user. Moreover, before proceeding with payment, the bank verifies each received token to check its validity by requesting the blockchain network.

Double-selling prevention: Double selling can be launched by a vendor that has sold a product to a user. For this, the vendor resubmits the transaction previously provided by the user. After verification, the transaction is inserted in a block by the auditor and the block is sent to the blockchain network for validation. The information contained in the different tokens occurring in the transaction allows the blockchain network to invalidate the transaction. In that case, the blockchain is able to discover that the transaction has already been submitted and identifies the malicious vendor. This operation can also be performed by the auditor, when it receives the invalidation result sent by the blockchain, provided that it kept in memory the history of submitted transactions.

Tokens forging prevention: Token forgery can be detected by the vendor, if it has been attempted before transaction construction. This is achieved thanks to the certificates of the bank and the user by simply verifying the related signatures. If the forgery is done after transaction formation, then the auditor can detect it by simply verifying the attached signatures. Finally, the forgery can be detected by the blockchain network, if the block it receives contains a modified token. This can be achieved thanks to the information provided by the bank issuing the tokens, since the modified token will not be found among those published by the bank.

Enforcement of traceability of payment: This function is guaranteed by the payment protocol through the blockchain technology and by the auditors since they keep track of handled tokens, transactions and blocks. It includes token payment tracing and product payment tracing. For token payment tracing, the protocol requires that each actor, after reception of each token, should verify the authenticity of the tokens received and their owners, the timestamp included in the tokens; then it includes its certificate, and its timestamp. When a block of aggregated tokens is uploaded into the blockchain network, the tokens and the transactions in the block are checked and stored. For product payment tracing, the auditor is able to retrieve all the information related to the purchases made by a user based on the transactions it receives from the vendors and the related results provided by the blockchain.

User's trust management: Ensuring the trustworthiness of our infrastructure is done through the provision of three different trust models that compute the trust of the user (i.e., neutral, optimistic, and pessimistic models). In fact, the micro-payment tokens are aggregated by a trusted third party and then uploaded to the blockchain network after insertion in a block. The

latter will determine the validity of each token included in the block and then send the result of verification to the auditor. Using the verification result, the auditor is able to maintain the user's trust value and updates the estimation of risk of loss to be dealt with for the users submitting the transactions occurring in the block. Therefore, the user trust is revisited after every transactions submission.

Overhead reduction: In terms of communication and cost, the overhead is reduced for the following three main reasons. First, the number of messages transmitted towards the blockchain is reduced. This is more perceptible for high numbers of users. In fact, only the auditor can communicate with the blockchain by transmitting aggregated transactions (into blocks) for validation. Second, as long as the user has not reached his assigned size of the related blocks to be transmitted to the blockchain network, there is no waiting time for the user since his transactions with the vendors are promptly performed. Third, μPIB imposes a fee on the vendor. In fact, the latter pays a percent ρ to the auditor for every token he received. However, this overhead compensates the vendor's risk of loss if the auditor was not involved.

In terms of processing, the overhead is reduced at the vendor, auditor, and blockchain sites. At the blockchain network level, the miners will receive less transaction so that less computation will be required, since they only receive blocks coming from the auditor. At the vendor level, the tokens received from the user are aggregated into transactions of tokens and only formed transactions are transmitted to the auditor. Finally, the main periods of time consumed by the auditor are of three types: (a) the period spent in the formation of blocks of transactions; (b) the time spent during the communication when transmitting the blocks for verification; and (c) the time consumed during the processing of payment and loss coverage. One can agree, that the three types of periods tend to be reduced because the number of verifications is reduced (made on the basis of transactions) and the number of transmissions are reduced.

7. µPIB Simulation

In this section, we evaluate the performance of μPIB. For this, we have conducted a simulation using the Matlab environment and we have presented a numerical proof of double spending prevention by assessing the following metrics:

a) The Mean Tokens Block Size $\overline{W}$ per user over time.
b) The Average Mean Tokens Block Size $\overline{W}$ of all the users over time.
c) The Average risk evolution over time.

Moreover, we are interested to show the scalability of our solution by varying the number of users and also the number of vendors.

In the sequel, we first introduce our simulation model and then discuss the results of our numerical simulation.

7.1 Simulation models

We consider the following four models:

Purchase model: we assume that a user has a large number of tokens generated by the bank. He can buy products using micro-payments from the multiple vendors. We assume that the time is slotted so that in every time slot, a user can buy one product with a fixed percentage p (with values equal to 0.1, 0.2, and 0.3). We also assume that n users and m vendors are involved in the simulation and that the products are assumed to have the same cost. Values proposed for n, and m are equal to 1, 50, or 100 users, and 1, 2, or 10 vendors, respectively. The cost can be equal to 1, 5, or 25 tokens.

User profile: we assume that a user can use the same token twice. Therefore, the invalidity of a transaction is caused by a double spending attack. A token paid to a vendor is assumed to

Table 2. Parameters values

Parameter	Value
Initial value of W (W_0)	100
Product cost (p)	5
Frequency of payment	0.4
Number of users	1
Number of vendors	1
Rate of double spending (q)	0.2
Auditor's withholding of payment (ρ)	20

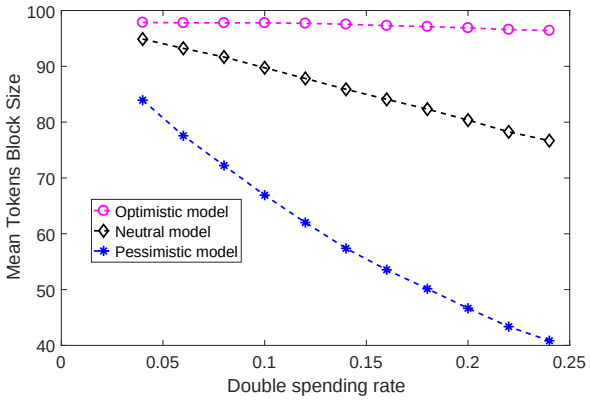

Fig. 3. Mean Tokens Block Size w.r.t Generation rate of double spending tokens

be q percent double paid (with value equal to 1%, 5%, or 10%).

Modeling blocks size: the starting value of Tokens Block Size W_0 is fixed to a value equal to 5, 10, or 15.

Auditor profile: the three profiles of the auditor are considered in the simulation. An auditor can be risk neutral auditor (i.e., applying the trust function for the neutral profile), the risk averse auditor (i.e., applying the trust function for the pessimistic profile), and the risk lover auditor (i.e., applying the trust function for the optimistic profile).

The parameters used in our simulation are summarized in the following table 2.

7.2 Simulation results

In what follows, we describe the results of our simulation.

In the first simulation, as indicated in Figure 3, we evaluated the Mean Tokens Block Size $\bar{W}$ with respect to the generation rate of double spending tokens q for the three proposed trust models, by considering one user and one vendor. The initial value of the Tokens Block Size W_0 is set to 100. The frequency of buying products is set to 0.4. The product cost is equals to 5. We note that the Mean tokens block size $\bar{W}$ decreases with the increase of the double spending rate, for the three trust models. Moreover, with the increase of the double spending rate, the optimistic model decreases slowly, however, the pessimistic model decreases rapidly. But, the decrease of the neutral model is moderate. We observe that $\bar{W}$ of the optimistic trust model is always upper than $\bar{W}$ of the neutral trust model. The latter is upper than $\bar{W}$ of the pessimistic model. One can say that, if the auditor wants to tolerate the dishonest users, the optimistic model is used and, if the auditor does not want to tolerate them, the pessimistic model can be used.

In the second simulation, as depicted in Figure 4, we simulated the Mean Tokens Block

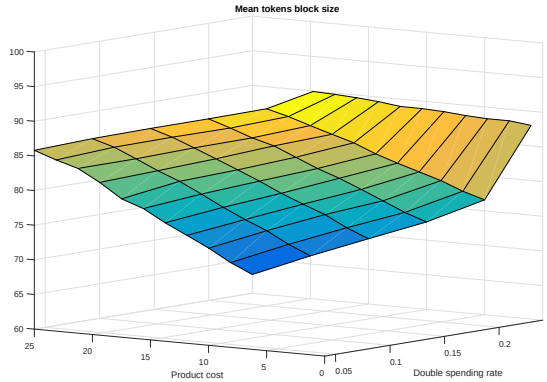

Fig. 4. Mean Tokens Block Size w.r.t Generation rate of double spending tokens

Size $\overline{W}$ with respect to the generation rate of double spending tokens q and the product cost (i.e., which is varied from 1 to 25) by applying the neutral trust model. W_0 is set to 100. The frequency of buying products is set to 0.4. We observe that the Mean Tokens Block Size $\overline{W}$ decreases with the increase of the double spending rate. Moreover, we note that $\overline{W}$ decreases with the increase of the product cost. In fact, the more we increase the double spending rate and the product cost the lower is the Mean Tokens Block Size $\overline{W}$. It is worth noting that the highest value of the Mean Tokens Block Size $\overline{W}$is obtained for the lowest value of the product cost (i.e., 1 token) and the lowest value of the double spending rate. Moreover, we note that the lowest value of $\overline{W}$ is achieved for the highest values of both the product cost and the double spending rate. For honest users, the product cost can be increased. However, for dishonest users, it is not useful to increase the product cost. It is better to decrease the product cost for the dishonest users.

In the third simulation, as indicated in Figure 5, we simulated the Mean Tokens Block Size $\overline{W}$ with respect to the initial value of W (W_0) for the three trust models. The product cost is equals to 5. The double spending rate is 0.2. We observe that the increase of the value of W_0 contributes to the increase of the value of $\overline{W}$. In addition, we note that the values of $\overline{W}$ of the optimistic model are higher than the values of $\overline{W}$ of the neutral model. The values of $\overline{W}$ of the pessimistic model are lower than the values of $\overline{W}$ of the neutral model. To conclude, we can say that the decision making of the auditor (i.e., more or less strict) depends on type of the trust model that will be applied (i.e., optimistic or pessimistic or neutral). For instance, if the auditor wants to behave well toward the dishonest users (i.e., he tolerates them) the optimistic or neutral model will be chosen. However, if the auditor wants to be strict with dishonest users it will select the pessimistic model.

In the fourth simulation, as shown in Figure 6, we evaluated the Mean tokens block size with respect to W_0 (i.e., which is varied from 20 to 100) and the product cost (i.e., which is varied from 1 to 25), by applying the pessimistic model. The double spending rate is 0.2. We note that the Mean tokens block size increases with the increase of W_0 and decreases with the increase of the product cost. Hence, we can say that the more we increase the product cost and the more we reduce W_0 the less is the Mean Tokens Block Size $\overline{W}$. Moreover, it is interesting to note that the highest value of the Mean Tokens Block Size is obtained for the highest value of W_0 and the lowest value of the product cost (i.e., 1 token). To resume, we can say that if the auditor wants to be tolerant with dishonest users, it will increase the initial value of W (W_0). However, if the auditor wants to be intolerant with dishonest users, it will decrease the initial value of W (W_0).

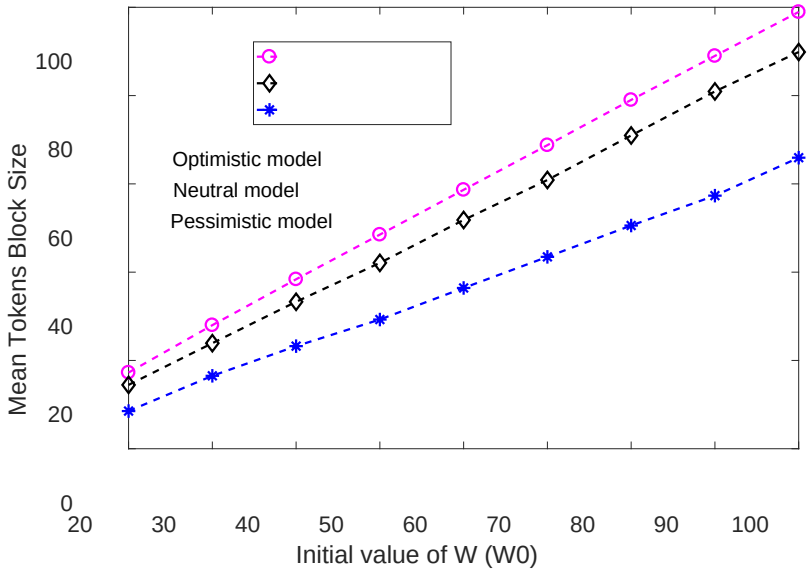

Fig. 5. Mean Tokens Block Size w.r.t Initial value of W (W_0)

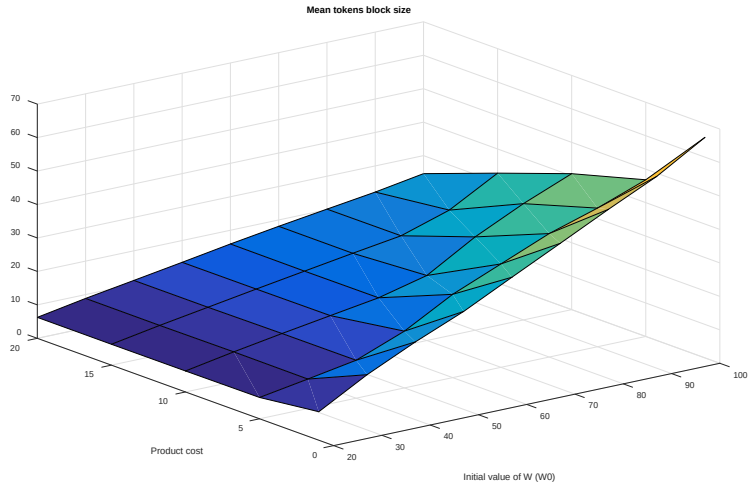

Fig. 6. Mean Tokens Block Size w.r.t Initial value of W (W_0)

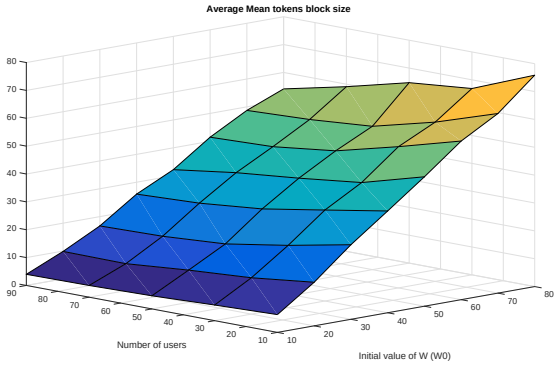

Fig. 7. Average Mean Tokens Block Size w.r.t Initial value of W (W_0)

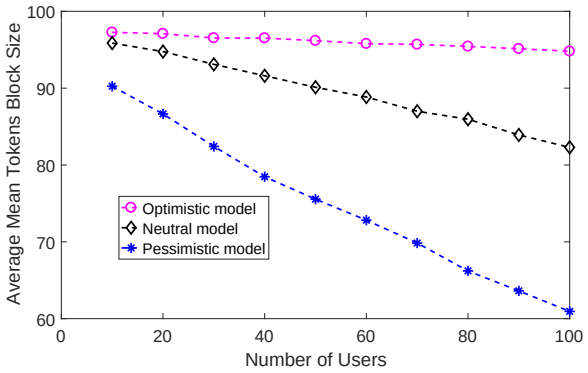

Fig. 8. Average Mean Tokens Block Size w.r.t Number of Users

In the fifth simulation, as shown in Figure 7, we simulated the Average Mean Tokens Block Size of all the users with respect to the initial value of W_0 (i.e., which is varied from 20 to 80) and the number of users (i.e., which is varied from 10 to 90), for the pessimistic model. The product cost is set to 5. We note that the Average Mean Tokens Block Size increases with the increase of W_0. Moreover, we observe that the less is the number of users the more is the Average Mean Tokens Block Size. In fact, when we increase the number of users the chance to have dishonest users will increase so that the Average Mean Tokens Block Size decreases. To resume, we can say that the highest value of the Average Mean Tokens Block Size is obtained for the lowest number of users and the highest value of W_0.

In the sixth simulation, as shown in Figure 8, we simulated the Average Mean Tokens Block Size of all the users with respect to the number of users. The initial value of W (W_0) is set to 100. The product cost is equal to 5. We observe that the more is the number of users the less is the Average Mean Tokens Block Size for the three trust models. In fact, the more we increase the number of users the more is the chance to have dishonest users. In addition, we observe that the average of $\overline{W}$ decreases slowly for the optimistic trust model and decreases rapidly for the pessimistic trust model. However, this decrease is moderate for the neutral model. Moreover, we note that the optimistic trust model shows higher values of Average Mean Tokens Block Size than the two other trust models. To conclude, we can say that the pessimistic model is suitable of the dishonest users.

In the seventh simulation, as indicated in Figure 9, we evaluated the Mean Tokens Block Size $\overline{W}$ with respect to the generation rate of double spending tokens q for the three proposed trust models in the case of two vendors (Product cost equals to 10 tokens and Product cost equals

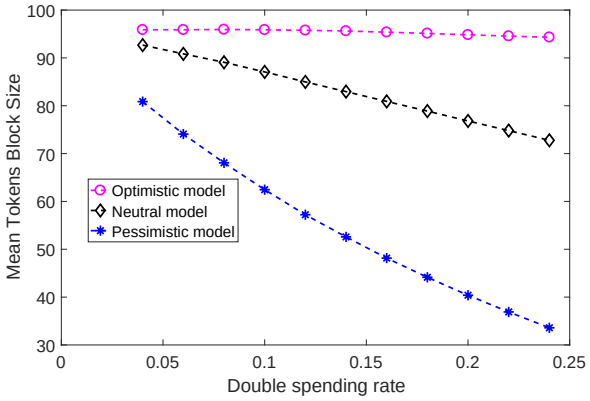

Fig. 9. Mean tokens block size w.r.t Generation rate of double spending in the case of two vendors

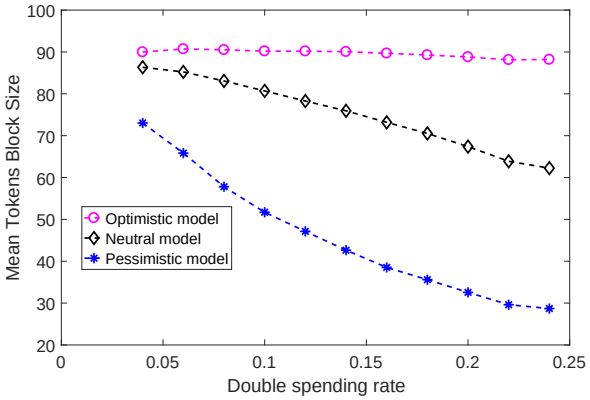

Fig. 10. Mean tokens block size w.r.t Generation rate of double spending in the case of ten vendors

to 20 tokens). W_0 is set to 100. The frequency of buying products is set to 0.4. We observe that $\overline{W}$ decreases with the increase of the double spending rate. In addition, we note that $\overline{W}$ in the optimistic trust model is always upper than $\overline{W}$ in the two other trust models. Moreover, we observe that, for a fixed value of double spending rate, the decrease of the values of $\overline{W}$ for the pessimistic model is high compared to the two other models. In addition, we note that the values of the Mean Tokens Block Size are lower compared to the values obtained in the case of one vendor is used (as shown in Figure 3). This can be explained by the increase of the product cost.

In the eighth simulation, we evaluated the Mean Tokens Block Size $\overline{W}$ with respect to the generation rate of double spending tokens q for the three proposed trust models in the case of ten vendors and having different product costs, as indicated in Figure 10. W_0 is set to 100. We observe that $\overline{W}$ decreases with the increase of the double spending rate as shown in Figure 9. In addition, the pessimistic trust model shows the lowest values of $\overline{W}$ compared to the two other trust models. In addition, we observe that the values of $\overline{W}$ are lower than the values obtained in Figure 9. To resume, we can say that the more we increase the number of vendors the less are the values of the Mean Tokens Block Size because of the variation of the value of product cost and the increase of the likelihood of the user's misbehaviour.

In the ninth simulation, as depicted in Figure 11, we simulated the average risk evolution with respect to the generation rate of double spending tokens q for the three proposed trust

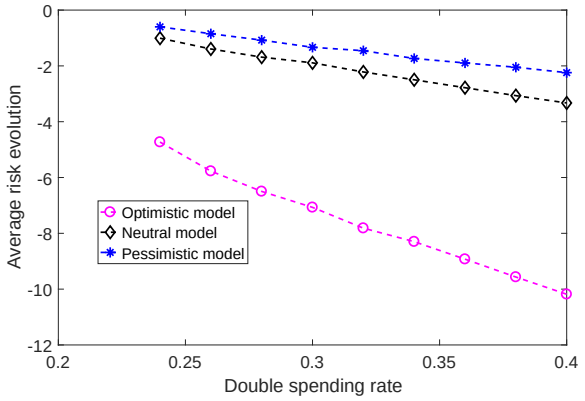

Fig. 11. Average risk evolution w.r.t Generation rate of double spending tokens

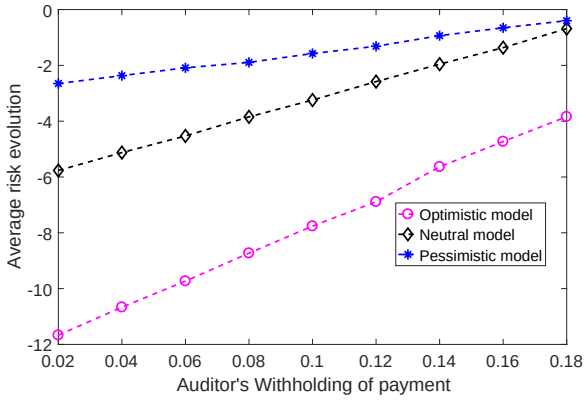

Fig. 12. Average risk evolution w.r.t Auditor's withholding of payment

models. W_0 is set to 50. The frequency of buying products is set to 0.4. The auditor's withholding of payment is set to 20. As shown in Figure 11, we observe that the average risk evolution decreases with the increase of the generation rate of double spending tokens for the three trust models. Moreover, we note that the risk values are negative and the pessimistic trust model shows better results (i.e., less loss) than the optimistic and neutral trust models. This is explained by the fact that when the auditor applies the pessimistic trust model the risk is low. To conclude, we can say that, for higher values of double spending rate, it is better to use the pessimistic trust model.

In the last simulation, as shown in Figure 12, we simulated the average risk evolution with respect to the auditor's withholding of payment for the three proposed trust models. W_0 is set to 50 and the rate of double spending tokens is set to 0.2. The frequency of buying products is set to 0.4. We note that the average risk increases with the increase of the auditor's withholding of payment. Moreover, we observe that the values of the average risk are negative and the pessimistic trust model shows better results compared to the two other trust models. Therefore, we can say that for lower values of the auditor's withholding of payment (ρ), it is better to use the pessimistic trust model. In addition, the optimistic model shows highest risk because the values of average risk are very high in absolute terms. To resume, the pessimistic model allows the auditor to obtain low loss because the average risk of the auditor is low compared to the neutral model and the optimistic model.

8. Conclusion

In this paper, we firstly presented our trust-aware, efficient, and robust micro-payment infrastructure based on the blockchain technology. Secondly, we presented our three trust models that compute the trust values of the user. Thirdly, we present the decision making of the auditor that consists in the computation of the future value of the tokens block size and the related risk to be dealt with. Then, we detailed the validation of our proposed infrastructure. Finally, we evaluated the performance of our proposed user's trust models, by assessing the mean tokens block size per user over time, the average mean tokens block size of all the users over time, and the average risk evolution.

Our approach can be extended to different directions. First, many auditors can be allowed assuming that they interact to correlate their trust functions. Second, more parameters can be integrated when evaluating the risk of loss by looking to more sophisticated attacks. Finally, the waiting delay of a customer can be more shortened by involving more interactions between the auditor and the blockchain system.

References

Ali, S. T., Clarke, D. & McCorry, P. (2017). 'The nuts and bolts of micropayments: a survey'. *CoRR* **abs/1710.02964**. arXiv:1710.02964v1 [cs.CR].

Alnumay, W., Ghosh, U. & Chatterjee, P. (2019). 'A trust-based predictive model for mobile ad hoc network in internet of things'. *sensors* **19(6)**. doi:10.3390/s19061467.

Che, S., Feng, R., Liang, X. & Wang, X. (2015). 'A lightweight trust management based on bayesian and entropy for wireless sensor networks'. *Security and communication networks* **8(2)**, 168–175.

Chen, D., Zhang, Z., Krishnan, A. & Krishnamachari, B. (2019). Payflow: Micropayments for bandwidth reservations in software defined networks. *in* 'IEEE INFOCOM 2019 - IEEE Conf. on Computer Communications Workshops (INFOCOM WKSHPS)'. Paris, France. pp. 26–31.

Chen, Z., Tian, L. & Lin, C. (2017). 'Trust model of wireless sensor networks and its application in data fusion'. *sensors* **17(4)**.

Dai, H.-N., Zheng, Z. & Zhang, Y. (2019). 'Blockchain for internet of things: A survey'. *IEEE Internet of Things J.* **6(5)**, 8076 – 8094.

Dai, H., Young, H. P., Durant, T. J. S., Gong, G., Kang, M., Krumholz, H. M., Schulz, W. L. & Jiang, L. (2018). 'Trialchain: A blockchain-based platform to validate data integrity in large, biomedical research studies'. *CoRR* **abs/1807.03662**. arXiv:1807.03662v1 [cs.DC].

Decker, C. & Wattenhofer, R. (2015). A fast and scalable payment network with bitcoin duplex micropayment channels. *in* 'Proc. of the Int. Symp. on Stabilization, Safety, and Security of Distributed Systems (SSS)'. Edmonton, Canada.

Duan, J., Gao, D., Foh, C. H. & Leung, V. C. M. (2013). Trust and risk assessment approach for access control in wireless sensor networks. *in* 'Proc. of the 2013 IEEE 78th Vehicular Technology Conference (VTC Fall)'.

Ensor, A., Schefer-Wenzl, S. & Miladinovic, I. (2018). Blockchains for iot payments: a survey. *in* 'Proc. of the 2018 IEEE Globecom Workshops (GC Wkshps)'. Abu Dhabi, United Arab Emirates.

Feng, R., Han, X., Liu, Q. & Yu, N. (2015). 'A credible bayesian-based trust management scheme for wireless sensor networks'. *Int. J. of Distributed Sensor Networks* **2015**.

Heilman, E., Baldimtsi, F. & Goldberg, S. (2016). Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transacations. *in* S. B. Heidelberg, ed., 'the 20th Int. conf. on financial cryptography and data security, Proc. of the FC 2016 Int. Workshops, BITCOIN, VOTING, and WAHC'. Christ Church, Barbados. pp. 43–60. https://eprint.iacr.org/2016/056.pdf.

Konstantinidis, I., Siaminos, G., Timplalexis, C., Zervas, P., Peristeras, V. & Decker, S. (2018). Blockchain for business applications: A systematic literature review. *in* 'Proc. of the 21st Int. Conf. on Business Information Systems'. Berlin, Germany. pp. 384–399.

Lahkani, M. J., Wang, S., nski, M. U., & Egorova, M. (2020). 'Sustainable b2b e-commerce and blockchain-based supply chain finance'. *sustainability* **12**.

Lundqvist, T., de Blanche, A. & Andersson, H. R. H. (2017). Thing-to-thing electricity micro payments using blockchain technology. *in* 'Proc. of the 2017 Global Internet of Things Summit (GIoTS)'. Geneva, Switzerland. doi: 10.1109/GIOTS.2017.8016254.

Makridakis, S. & Christodoulou, K. (2019). 'Blockchain: Current challenges and future prospects/applications'. *Future Internet* **11(258)**. MDPI.

Pass, R. & abhi shelat (2015). Micropayments for decentralized currencies. *in* 'CCS '15 Proc. of the 22nd ACM SIGSAC Conf. on Computer and Communications Security'. Denver, Colorado, USA. pp. 207–218.

Poon, J. & Dryja, T. (2016). The bitcoin lightning network: Scalable off-chain instant payments. Technical report. https://lightning. network/lightning-network-paper. pdf.

Radhakrishnan, R. & Krishnamachari, B. (2018). Streaming data payment protocol (sdpp) for the internet of things. *in* '2018 IEEE Int. Conf. on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)'. Halifax, NS, Canada. pp. 1679–1684.

Ramachandran, G. S., Ji, X., Navaney, P., Zheng, L., Martinez, M. & Krishnamachari, B. (2019). 'Motive: micropayments for trusted vehicular services'. *CoRR* **abs/1904.01630**. arXiv:1904.01630v1 [cs.DC].

Rezaeibagha, F. & Mu, Y. (2018). 'Efficient micropayment of cryptocurrency from blockchains'. *The Computer J.* **62(4)**, 507–517.

Strugar, D., Hussain, R., Mazzara, M., Rivera, V., Lee, J. & Mustafin, R. (2018). On m2m micropayments : A case study of electric autonomous vehicles. *in* 'Proc. of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)'. Halifax, NS, Canada, Canada. pp. 1697–1700.

Sun, G., Zhang, Z., Zheng, B. & Li, Y. (2019). 'Multi-sensor data fusion algorithm based on trust degree and improved genetics'. *sensors* **19(9)**.

Syed, T. A., Alzahrani, A., Jan, S., Siddiqui, M. S., Nadeem, A. & Alghamdi, T. (2019). 'A comparative analysis of blockchain architecture and its applications: Problems and recommendations'. *IEEE Access* **7**, 176838–176869. doi: 10.1109/ACCESS.2019.2957660.

Wan, Z.-G., Deng, R. H., Lee, D. & Li, Y. (2019). 'Microbtc: Efficient, flexible and fair micropayment for bitcoin using hash chains'. *J. OF COMPUTER SCIENCE AND TECHNOLOGY* **34(2)**, 403–415.

Wang, C., Zhang, C. & Yang, X. J. (2018). 'Automation reliability and trust: A bayesian inference approach'. *Proc. of the Human Factors and Ergonomics Society Annual Meeting* **62(1)**, 202–206.

Weking, J., Mandalenakis, M., Hein, A., Hermes, S., Bohm, M. & Krcmar, H. (2019). 'The impact of blockchain technology on business models - a taxonomy and archetypal patterns'. *Electronic Markets* **30**, 285–305.

Xu, A., Li, M., Huang, X., Xue, N., Zhang, J. & Sheng, Q. (2016). 'A blockchain based micro payment system for smart devices'. *Signature* **256(4936)**, 115.

Yang, X. J., Unhelkar, V. V., Li, K. & Shah, J. A. (2017). Evaluating effects of user experience and system transparency on trust in automation. *in* 'Proc. of the 2017 ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI '17)'. Vienna, Austria. pp. 408–416.

Ye, Z., Wen, T., Liu, Z., Song, X. & Fu, C. (2017). 'An efficient dynamic trust evaluation model for wireless sensor networks'. *sensors* **2017**.

Yu, Y., Li, K., Zhou, W. & Li, P. (2012). 'Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures'. *J. of Network and Computer Applications* **35(3)**, 867–880.

Zhang, D., Le, J., Mu, N. & Liao, X. (2018). 'An anonymous off-blockchain micropayments scheme for cryptocurrencies in the real world'. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* pp. 1–11. DOI: 10.1109/TSMC.2018.2884289.

Zhang, J., Sun, Q., Zhou, A. & Li, J. (2016). A novel trust update mechanism based on sliding window for trust management system. *in* 'Proc. of Int. Conf. on Computational Science and Its Applications (ICCSA 2016)'. Beijing, China. pp. 521–528.

ZHAO, J., HUANG, J. & XIONG, N. (2019). 'An effective exponential-based trust and reputation evaluation system in wireless sensor networks'. *Special Section on Artificial Intelligence and Cognitive Computing for Communication and sensors* **7**, 33859 – 33869.

Zile, K. & Strazdina, R. (2018). 'Blockchain use cases and their feasibility'. *Applied Computer Systems* **23(1)**, 12–20.

Zupancic, E. & Zalik, B. (2019). 'Data trustworthiness evaluation in mobile crowdsensing systems with users' trust dispositions' consideration'. *sensors* **19(6)**.

Submitted: 18/09/2020
Revised: 01/01/2021
Accepted: 06/03/2021
DOI: 10.48129/kjs.v49i1.10578