Received: 19 November 2019 | Revised: 30 January 2020 | Accepted: 29 February 2020

DOI: 10.1002/cpe.5739

WILEY

SPECIAL ISSUE PAPER

Tesia: A trusted efficient service evaluation model in Internet of things based on improved aggregation signature

Fengyin Li[1] | Rui Ge[1] | Huiyu Zhou[2] | Yilei Wang[1] | Zhongxing Liu[1] | Xiaomei Yu[3]

[1]School of Information Science and Engineering, Qufu Normal University, Rizhao, China

[2]School of Informatics, University of Leicester, Leicester, UK

[3]School of Information Science and Engineering, Shandong Normal University, Jinan, China

Correspondence
Yilei Wang, School of Information Science and Engineering, Qufu Normal University, Rizhao, China.
Email: ylwangqfnu@163.com

Funding information
Foundation of National Natural Science Foundation of China, Grant/Award Number: 61771231, 6150028, 61672321, 61771289, 61832012; Natural Science Shandong Province, Grant/Award Number: ZR2016FM23, ZR2017MF010, ZR2017MF062; EU Horizon 2020 DOMINOES Project, Grant/Award Number: 771066

Summary

Service evaluation model is an essential ingredient in service-oriented Internet of things (IoT) architecture. Generally, traditional models allow each user to submit their comments with respect to IoT services individually. However, these kind of models are fragile to resist various attacks, like comment denial attacks, and Sybil attacks, which may decrease the comments submission rate. In this article, we propose a new aggregation digital signature scheme to resolve the problem of comments aggregation, which may aggregate different comments into one with high efficiency and security level. Based on the new aggregation digital signature scheme, we further put forward a new service evaluation model named Tesia allowing specific users to submit the comments as a group in IoT networks. More specifically, they aggregate comments and assign one user as a submitter to submit these comments. In addition, we introduce the synchronization token mechanism into the new service evaluation model, to assure that all users in the group may sign their comments one by one, and the last one who receives the token is assigned as the final submitter. Tesia has more acceptable robustness and can greatly improve the comments submission rate with rather lower submission delay time.

KEYWORDS

aggregation signature, Internet of things, service comment attacks, service evaluation, synchronization token

1 | INTRODUCTION

Service-oriented architecture is kind of software design, widely used in Internet of things (IoT), where facilities provide services.[1-5] The facilities form a platform/network, where individuals are able to communicate with local sensors through wireless communication devices.[6,7] In the service-oriented architecture, facilities may improve their services quality based on their service evaluation systems. For example, the smart home systems adapt the home appliance control quality according to the comments returned by the service evaluation system. Therefore, it's essential to convince the users to trust the service evaluation system, where service quality is evaluated. Otherwise, service-oriented architecture would be useless even it can be manipulated by unscrupulous vendors, who may arbitrarily revise their service quality through the architecture.*

In effect, users' comments are crucial to influence the service quality for IoT in the service evaluation systems.[15-18] For instance, compliments and complaints would elevate or lower the service quality in a trust service evaluation system, respectively. On the other hand, the IoT systems will improve their service quality according to the feedbacks from the users. It seems to be a benign system if there is no malicious comments or

*The emergence technology of blockchain and cloud computing may get around this arbitrary revision,[8-14] which will be addressed in the future works.

unscrupulous users. The third trusted authority is frequently maintains the trust service evaluation system. However, the IoT systems may collude with the third trusted authority and arbitrarily modify their unwanted comments. For example, they may delete and modify the complaints, which are corresponding comments deleting attacks and modifying attacks, respectively. These attacks can dramatically decrease the comments submission rate and delay submission time, which paralyze the whole system. Therefore, the designer for trust service evaluation system should consider how to resist such attacks such that other users trust the model works well.

The existing works indicate that it is hard to establish a perfect trust mechanism in service evaluation systems. One of the important reasons is that comments submitted by individuals could be easily manipulated by some IoT systems or the trusted authority. Second, malicious users have incentives to attack the systems to get some advantages. Therefore, it is essential to propose a new credible and efficient service evaluation model, which allows more users to cooperatively submit comments, while resist most service evaluation attacks from malicious IoT systems and malicious users.

A new service evaluation model named Tesia is proposed in this article. In the new model, users participate in trusted service evaluation in a cooperative manner. Without the need of a trusted third party, the service provider can maintain a credible and efficient service evaluation, which can effectively resist the comment rejection attack and comment modification attack. It improves the success rate (SR) of comment submission while ensuring security, and reduces the delay time (DT) of comment submission. Specifically, the main contributions of this article are as follows.

We propose a new aggregation digital signature scheme, to resolve the problem of comments aggregation, which may aggregate different comments into a whole with high efficiency and security level. Furthermore, under the security assumption of the difficulty of discrete logarithm problem, we have proved the security of the signature scheme in Existential unforgeability against chosen message attacks (EU-CMA) security model.

Tesia is proposed as a new trusted efficient service evaluation model, which allows multiple users to form a team and aggregate their comments as a whole by utilizing the new proposed aggregation digital signature scheme. Moreover, Tesia introduces synchronization token mechanism to guarantee sequential signature inside the team and finally assign a submitter.

We analyze the security of Tesia under current service comments attacks, such as linkability attacks, rejection attacks, modification attacks, and Sybil attacks. Tesia performs well to resist these attacks. Furthermore, we also evaluation the SR and DT by comparing with traditional models without trust mechanisms.

The rest of this article is organized as below. Section 2 presents the general trusted service evaluation, consisting of the network structure, and potential attacks. Section 3 proposes a new model called Tesia, which means a trusted efficient service evaluation model. The building blocks in Tesia include a new aggregation digital signature scheme and synchronization token mechanism. Then we present the workflow of Tesia based on these building blocks. Finally, security analysis with respect to various attacks. Section 4 compares the comments SR and DT between Tesia and traditional models without considering attacks. Section 5 summarizes this article and prospects the future works.

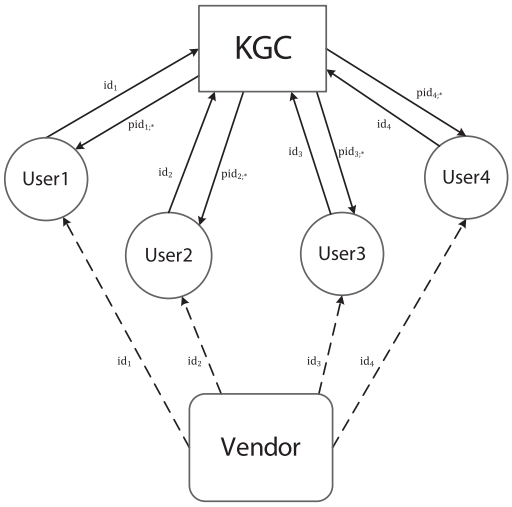
2 | RELATED WORK

Rajan and Hosamani[19] used additional monitors disposed in the untrusted vendor's site, which can ensure the accuracy of the evaluation results. The two-dimensional trust rating aggregation method is put forward by Wang and Li,[20] which enables a small group of trust vectors to indicate a large group of trust ratings. The trust management as an reasoning problem and a faith propagation algorithm are proposed by Aydey and Fekri[21] such that it can efficiently calculate the marginal probability distribution functions representing reputation values. Based on the calling logs' histories, Alamir and Navimipour[22] introduced a method to assure the credibility of users' comments, which can survey trust relationship among social network users. A dynamic trust computation model is proposed by Das and Islam[23] such that it can deal with the dynamically changed behaviors deriving from malicious vendors.

Wang and Li[20] proposed a new direct anonymous attestation mechanism to ensure the credibility of the enterprise comment received by the Alamir and Navimipour[23] proposed a noncooperative (NCP) system where each user in the NCP system submits their comments directly, and respectively, to the vendor. Based on the bilinear mapping convergence digital signature scheme and synchronization token technology, Xiao et al[7] designed bTSE model that was independently maintained by the vendor to receive comments. Zhu et al[6] proposed a SEER model for submitting comments in the form of an integrated chain using layered and aggregated digital signature and synchronization token technologies.

Robinson and Thomas[24] proposed a new method to mitigate service rejection comments. Aad et al[25] comprehensively expounded the impacts of service denial attacks on ad hoc networks and proposed new solutions based on the characteristics of the network. Distributed systems are vulnerable to sybil attacks, in which attackers control false identities or misuse pseudonyms to compromise the effectiveness of the systems. The theory that the sybil attacks can destroy the redundance in distributed memory systems is proposed by Douceur.[26] In the sensor networks, the theory that the Sybil attacks can impair routing efficiency is putted forward proposed by Karlof and Wagner.[27] Lots of defence mechanisms are put forwarded by Newsome et al.[28] Wei et al[29] discovered that the presence of a third trusted institution with respect to social network may ease various attacks such as the sybil attacks. However, they hold the idea that it's not acceptable since it will bring extra burdens on users.

FIGURE 1 The framework of the trusted service evaluation model



3 | GENERAL TRUSTED SERVICE EVALUATION MODEL

3.1 | Network structure

The network structure of a general trusted service evaluation model composes of diverse vendors that offer different or similar services to users. For simplicity, a single vendor offering a single service is considered in this article. More specifically, a wireless communication equipment with sufficient storage space is equipped by the vendor. The handhold communication equipment are equipped by each users. We also assume that the equipment's transmission scope is same for all users, but less than the vendor's transmission scope.

The framework of the trusted service evaluation model is shown in Figure 1. The key distribution center (KGC) is an infrastructure trusted by each user. Each user u_j ($j = 1, 2, \ldots, n$) has a private unique identity id_j. KGC verifies the unique id_j of each user u_j. Then KGC generates some pseudonyms $pid_{j;1}, pid_{j;2}, pid_{j;3}$ ($j = 1, 2, \ldots, n$) for each user u_j, each corresponds to a private key $psk_{j;*}$.

3.2 | Potential attacks on the network

There are some service comment attacks on the service evaluation models. Four major comment attacks are listed below. One of the main tasks is to resist on these attacks.

1. Comment linkability attacks: Comment linkability attacks are performed by malicious users who impersonate legitimate users. A malicious user prevents the KGC from tracking its unique identity so that the user cannot be linked to the comments submitted by it, thus breaking comment linkability.
2. Comment rejection attacks: When a user submits negative comments, the vendor will perform the comment rejection attacks. In this type attacks, the vendor quietly strikes out negative comments without replying to user submissions. Users will not be able to see these negative comments, and the incompleteness of comments will be misleading to the user.
3. Comment modification attacks: The vendor initiates a comment modification attack on the locally stored comment collection. The vendor inserts counterfeit comments or edits, strikes out negative comments in the comment collection. These attacks are designed to undermine the integrity of the comments and influence the users' right service choices.
4. Sybil attacks: The legitimate user are performed Sybil attacks, they leaves multiple false valid comments to the vendor within one time interval. Sybil attacks destroy the normal operation of the system through a large number of malicious bad comments or deliberate praises. Sybil attacks can be divided into two categories:
   - Sybil attack 1. The malicious users launch such sybil attack. A registered user leaves diverse comments to the same vendor within one time interval. The comments received by the vendor are fake and negative to it.
   - Sybil attack 2. The malicious vendors launch such sybil attack. A malicious vendor requires one colluded registered user to provide diverse valid comments to itself within one time interval, among which the comments are positive to the service vendor.

The above two sybil attacks undermine the validity of the trusted service evaluation model by generating false comment messages, which is unfair to users or vendors. The linkability of comments ensures that the comment is associated with the user's true identity. Therefore, the trusted

service evaluation model effectively resists sybil attacks by limiting users to generate a valid comment to the vendor at predefined intervals. If it is detected that any user produces two or more comments with different pseudonyms to the same vendor within one time interval, the real identity of the user will be disclosed. Limiting the number of comments in a time interval does not completely avoid the sybil attack. Any user can still generates incorrect comments at different time intervals using multiple different pseudonyms, and the model cannot immediately find the user who generated the malicious comment. The trusted service evaluation model can resist most Sybil attacks and ensure the true reliability of the user's comments on the vendors.

4 | TESIA: A NEW TRUSTED SERVICE EVALUATION MODEL

In this section we propose Tesia, a new trusted service evaluation model, comprising a new aggregation digital signature scheme and a synchronization token mechanism. The former is applied to the generation and aggregation process of the vendor's service comments and an the latter is utilized to sequentially sign the comments . Tesia is capable of overcoming four typical comment attacks defined in Section 3.2. In Tesia, there is no dedicated trusted third party to manage and maintain the comments, and the vendor can independently maintain the comments left by the users.

4.1 | A new aggregation digital signature scheme

This section proposes an aggregation digital signature scheme, which is described as follows:

1. Setup

   KGC chooses a prime q and the primitive root α, and chooses an encryption hash function $H_1:\{0,1\}^* \rightarrow Zq^*$ get the system parameters of digital signature scheme: params={q,α, H_1}.
2. Key generation
   - User u_i generates random integers X_i as u_i's private key, where X_i is $1<X_i<q-1$.
   - Calculate $Y_i=\alpha^{X_i} \bmod q$ as the public key of u_i.
   - User u_i's private key X_i is confidential and the public key Y_i is unclassified.
3. Signature
   - User u_i performs the following steps to sign comment content m_i, $\forall k_i \epsilon Zq^*$.
   - Calculate $K_i=\alpha^{k_i} \bmod q$, $S_i=k_i+X_i+H_1(m_i) \bmod q$.
   - Get signature $\sigma_i=(K_i, S_i)$.
4. Verification

   The verifier verifies the validity of the signature $\sigma_i=(K_i, S_i)$ on the comments m_i with Equation (1).

$$\alpha^{S_i} = K_i * Y_i * \alpha^{H_1(m_i)} \bmod q. \tag{1}$$

   In case Equation (1) holds, the signature σ_i is effective; otherwise, the signature σ_i fails.
5. Aggregation

   Multiple users' comments on the same vendor can be aggregated. Without loss of generalization, this article considers three users in comments submitting process. For the same vendor, u_1, u_2, and u_3 and their connections is shown in Figure 2. These users, respectively, obtained their comments m_1, m_2, and m_3, and the corresponding comment signatures are $\sigma_1=(K_1, S_1)$, $\sigma_2=(K_2, S_2)$, and $\sigma_3=(K_3, S_3)$. The signatures of the three users are aggregated as follows:

   σ_{agg}=(K, S)=($\prod^3_{i=1} K_i$, $\sum^3_{i=1} S_i$).

   Verify the validity of the aggregation signature with Equation (2):

$$\alpha^S = K \cdot \prod Y_i \cdot \alpha^{\sum H_1(m_i)}. \tag{2}$$

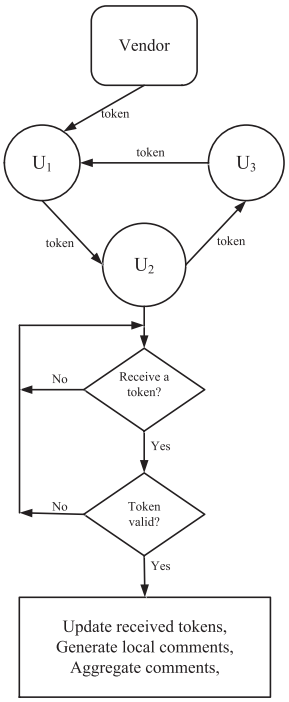
   In case Equation (2) holds, the aggregation signature is effective; otherwise, the aggregation signature fails.
6. Proof of correctness of the aggregation signature scheme

   Because $\alpha^{S_i} = \alpha^{k_i+X_i+H_1(m_i) \bmod q} = \alpha^{k_i} \cdot \alpha^{X_i} \cdot \alpha^{H_1(m_i)} \bmod q = K_i + Y_i + \alpha^{H_1(m_i)} \bmod q$.

   So

$$K \cdot \prod Y_i \cdot \alpha^{\sum H_1(m_i)} = \alpha^{k_1+k_2+k_3} \cdot \alpha^{X_1+X_2+X_3} + \alpha^{H_1(m_1)+H_1(m_2)+H_1(m_3)}$$
$$= \alpha^{k_1+X_1+H_1(m_1)} \cdot \alpha^{k_2+X_2+H_1(m_2)} + \alpha^{k_3+X_3+H_1(m_3)}$$
$$= \alpha^{S_1} \cdot \alpha^{S_2} \cdot \alpha^{S_3}$$
$$= \alpha^S$$

FIGURE 2 Delivery and of tokens workflow



The correctness of the aggregation signature scheme is verified.

7. Proof of security of the aggregation signature scheme

Definition 1. EU-CMA security model.

Let (α, α^a) be an instance of the DL problem that the simulator receives, and suppose the adversary can break the given signature scheme. Then, the simulator can solve the hard DL problem with the ability of the adversary. Under the security assumption of the DL problem, the given signature scheme is secure. EU-CMA security model is described as below.

Setup. Let system parameters SP $= (G, q, \alpha)$ and H_1 be set as a random oracle controlled by the simulator B. Let ID_C be the challenge identity. The simulator B randomly chooses secret parameters and sets the secret key and public key for ID_c.

H-Query. The adversary makes hash queries in this phase. B prepares a hash list to record all hash queries and responses the hash queries from the adversary, where the hash list is empty at the beginning.

Signature Query. In this phase the adversary makes signature queries on message m_i. B prepares a signature list to record all signature queries and responses the signature queries from the adversary, where the signature list is empty at the beginning.

Signature Forgery. The adversary returns a forged signature $\sigma_{m_{i^*}}$ of message m_{i^*}. And B computes the solution to the DL problem instance. This phase completes the simulation and the solution.

Theorem 1. *Suppose the hash function H_1 is a random oracle. If the DL problem is hard, the proposed signature scheme is provably secure in the EU-CMA security model with at most q_H signature queries and reduction loss q_H, where q_H is the maximum number of hash queries to the random oracle.*

Proof Idea. Let (α, α^a) be an instance of the DL problem that the simulator receives. The proof goal is to solve the DL problem with the forged signature from the adversary.

Proof. Suppose there exists an adversary A who can break the proposed signature scheme in the EU-CMA security model with at most q_H signature queries. We construct a simulator B to solve the DL problem. Given as input a problem instance (α, α^a) over the cyclic group (G, q, α), B controls the random oracle, runs A, and works as follows. ∎

Setup. Let system parameters SP $= (G, q, \alpha)$ and H_1 be set as a random oracle controlled by the simulator B. Let ID_c be the challenge identity. The simulator B randomly chooses $x, y, k_c \in Z_q$ and sets the secret key for ID_c as $X_c = x^{-1}(y - a \cdot k_c)$. The public key is $Y_c = \alpha^{x^{-1} \cdot y} \cdot (\alpha^a)^{-x^{-1} \cdot k_c}$, which can be computed from the problem instance and the chosen parameters.

H-Query. The adversary makes hash queries in this phase. Before receiving queries from the adversary, B randomly chooses an integer, $i^* \in [1, q_H]$ where q_H denotes the maximum number of hash queries to the random oracle. Then, B prepares a hash list to record all hash queries and responses as follows, where the hash list is empty at the beginning.

For a hash query on m_i, if m_i is already in the hash list, B responds to this query following the hash list. Otherwise, let m_i be the ith new queried message. B randomly chooses $w_i \in Z_q$ and sets

$$H(m_i) \text{ as } \begin{cases} H_1(m_i) = x, & \text{if } i = i^* \\ H_1(m_i) = w_i, & \text{otherwise} \end{cases}$$

Then, B responds to this query with $H_1(m_i)$ and adds $(m_i, H_1(m_i))$ to the hash list.

Signature Query. In this phase the adversary makes signature queries on message m_i. B prepares a signature list to record all signature queries and responses as follows, where the signature list is empty at the beginning.

If m_i is the i^*th queried message in the hash list, abort. Otherwise, B computes signature σ_{m_i} as follows.

B randomly chooses $v_i, v_i' \in Z_q$ and sets the signature σ_{m_i} as $(\alpha^{v_i}, v_i + v_i' \cdot w_i)$,which is a valid signature of m_i. Then, B responds to this query with $\sigma_{m_i} = (\alpha^{v_i}, v_i + v_i' \cdot w_i)$ and adds $(m_i, \alpha^{v_i}, v_i + v_i' \cdot w_i)$ to the hash list.

B computes the signature of m_i as follows.

$$\begin{aligned} \sigma_{m_i} &= (K_c, S_c) \\ &= (\alpha^{k_c}, k_c + X_c \cdot H_1(m_i)) \\ &= (\alpha^{k_c}, k_c + x^{-1}(y - a \cdot k_c) \cdot H_1(m_i)) \\ &= (\alpha^{k_c}, k_c + x^{-1}(y - a \cdot k_c) \cdot x) \\ &= (\alpha^{k_c}, k_c + y - a \cdot k_c) \end{aligned}$$

which is a valid signature of m_i.

Signature Forgery. The adversary returns a forged signature $\sigma_{m_{i^*}}$ of message m_{i^*}. Since $H_1(m_i) = H_1(m_{i^*}) = x$, we have

$$\begin{aligned} \sigma_{m_i} &= (K_{m_{i^*}}, S_{m_{i^*}}) \\ &= (\alpha^{k_c}, k_c + X_c \cdot H_1(m_i)) \\ &= (\alpha^{k_c}, k_c + X_c \cdot H_1(m_{i^*})) \\ &= (\alpha^{k_c}, k_c + x^{-1}(y - a \cdot k_c) \cdot x) \\ &= (\alpha^{k_c}, k_c + y - a \cdot k_c) \end{aligned}$$

That is $S_{m_{i^*}} = k_c + y - a \cdot k_c$

B computes $a = k_c^{-1} \cdot (k_c + y - S_{m_{i^*}})$ as the solution to the DL problem instance. This completes the simulation and the solution.

Indistinguishable simulation. The correctness of the simulation has been explained above. The randomness of the simulation includes all random numbers in the key generation and the responses to hash queries. They are

$a \cdot k_c, x^{-1}(y - a \cdot k_c), w_1, \ldots, w_{i^*-1}, x, w_{i^*+1}, \ldots, w_{q_H}$.

According to the setting of the simulation, where a, x, y, w_i are all randomly chosen, it is easy to see that they are random and independent from the point of view of the adversary. Therefore, the simulation is indistinguishable from the real attack.

Probability of successful simulation and useful attack. If the simulator successfully guesses i^*, the queried signature on the message $m = m_i^*$ is simulatable and the forged signature is reducible because the message chosen for signature query must be different from m_i^*. Therefore, the probability of successful simulation and useful attack is $1/q_H$.

Advantage and time cost. Suppose the adversary breaks the scheme with (t, q_H, ε) after making q_H hash queries. The advantage of solving the DL problem is therefore ε/q_H. Let T_s denote the time cost of the simulation. We have $T_s = O(1)$. B will solve the DL problem with

$(t + T_s, q_H, \varepsilon/q_H)$

This completes the proof of the theorem.

4.2 | Synchronization token working mechanism in Tesia

Tesia borrows token technology to synchronize the submission process of comments sequentially. The delivery order of tokens is determined by the token distribution situation, and the submission of comments is implemented by following the steps below (ref. Figure 2).

1. Token delivery and confirmation mechanism

   Users can submit their comments only when they have a legal token. The user sends a token request message when a comment needs to be submitted. After receiving the request, the nearby user or the vendor who currently hold a token can transmit the token to the requesting user. Only the first valid token received is accepted by the requesting user, then the user replies to the ACK message. A successful forwarding token that uses an ACK reply is no longer forwarded, and each token is forwarded to only one user at a time.

The tokens may be lost due to user mobility or malicious discard. The token-pseudonym (TP) list maintained by the vendor. From the TP list, the vendor can perceive the comments submitted by each token, specifically find out which token is lost, and trigger new token distribution. If any comments associated with a token are not received within the predefined maximum duration, the vendor will perceive the token lost. If the token is lost, the vendor will distribute a new token in place of the lost token in order to preserve a constant number of active tokens to ensure the stability of the comment system. Each token forms an independent comments chain.

In order to prevent some users from comments submission failure due to lack of tokens, the system should allocate enough tokens to avoid token starvation. The number of comment chains is directly proportional to the vendor's comment modification ability. The more comment chains are, the lower the credibility of users comments. To make comments more credible, the vendor should keep the number of tokens as small as possible. However, there should be enough tokens to prevent token starvation, otherwise some of users will never get the token and cannot submit their comments.

2. TP list of synchronization token

The TP list reflects the dynamic delivery process of the token and assists the successful submission of the aggregated comments. The data structure of the TP list is {(token public key, token, user pseudonym, comment signature)}, that is, (pk_t, tok_i, $pid_{1;*}$, σ_j). In TP list, each token is linked to a given pseudonym, which pertains to a single user who most lately submitted a comment using this token. Whenever a vendor receives a new comment, it updates the TP list and periodically broadcasts it to all users within the vendor's transmission range. After publishing the token message, if the vendor delete any token from the TP list, this behavior will perceived by the public, because any modification to the list will result in inconsistency with before published message. The user with the token will forward it to the randomly selected adjacent user requesting the comment after using the token.

3. Example of token working mechanism

Without loss of generalization, this article explains the token structure and the token delivery process taking three users as an example. Three users u_1, u_2 *and* u_3 are considered, with u_1 neighboring u_2, and u_2 neighboring u_3. These three users, respectively, obtained the pseudonym $pid_{1;*}$, $pid_{2;*}$, $pid_{3;*}$ from KGC. Using the identifier tok to represent the initial token. The vendor produces a public/private key pair (pk_t, sk_t) for tok and releases the public key pk_t.

The vendor sends the initial token to u_1 which is the first user to submit a comment in the comment chain. Then the vendor signs the pseudonym $pid_{1;*}$ and timestamp T of u_1 with the token private key sk_t, gets $\text{tok}_1 = \text{Sign}_{skt}(pid_{1;*}||T_1)$ as the initial version of the token.

As the first user to submit a comment, u_1 must submit a comment using the pseudonym $pid_{1;*}$, which is disclosed to it by the vendor. After submitting a comment using tok_1 and $pid_{1;*}$, u_1 updates tok_1 to tok_2 and delivers tok_2 to u_2 as a response to the u_2's token request.

tok_2=(PF_1; $\text{Sign}_{psk1;*}(pid_{2;*}||T_2)$), where $\text{PF}_1 = (pid_{1;*}, T_1, \text{Sign}_{skt}(pid_{1;*}||T_1))$ is the token forwarding proof for u_1.

After successful comment submission, the token public key, current token version, and corresponding pseudonym of the user who generated the current token version(pk_t, tok, $pid_{1;*}$) is appended to the current TP list.

Assume that the first token received by u_2 is tok_2. u_2 does the following operations:

- Check the authenticity and validity of the token by verifying the validity of (pk_t, tok, $pid_{1;*}$) and $\text{Sign}_{psk1;*}(pid_{2;*}||T_2)$;
- Check whether the user pseudonym, who last successfully forwarded token, is $pid_{1;*}$ by viewing the TP list;
- Send an ACK response to u_1;
- Generate local comments and submit comments using tok_2 and $pid_{2;*}$.
- Update tok_2 to $\text{tok}_3 = (\text{PF}_1, \text{PF}_2, \text{Sign}_{psk2;*}(pid_{3;*}||T_3))$, where $\text{PF}_2 = (pid_{2;*}, T_2, \text{Sign}_{psk1;*}(pid_{2;*}||T_2))$ and send tok_2 to u_3.

After receiving the token tok_3, u_3 performs an operation similar to u_2 to submit the comment and update the token.

4.3 | The work flow of Tesia

The generation of the comment does not depend on tokens. The user can flexibly generate local comments. But when users want to submit comments, they must hold a valid token. In the process of submitting a comment, the user aggregates the received comments and its local comments, and submits the aggregated comments to the vendor.

4.3.1 | Users' network structure

Suppose the vendor receives *n* comments. Tesia defines four basic comment structures, and respectively, gives the vendor's corresponding comment modification capabilities on them.

1. Discrete structure. In this structure, the comments are discretely distributed points, which means each user can submit comments, respectively, and independently. This independency enables the vendor to control the n comments. So the vendor's modification capability is $O(n)$.

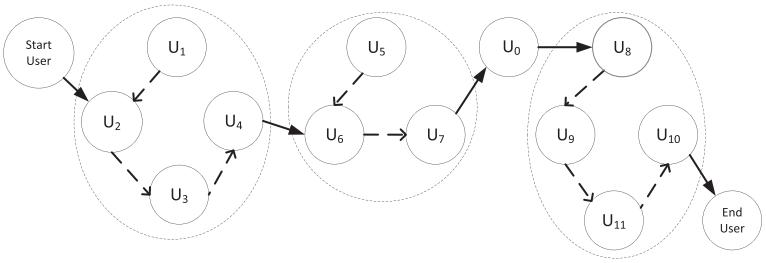


FIGURE 3 User's network structure model

2. Tree-like structure. In this structure, the vendor can delete any single comment corresponding to the leaf node. That is, the vendor's modification capability is $O(\log n)$.
3. Chain structure, the vendor's modification capability in this scenario is $O(1)$.
4. Ring structure, the vendor's modification capability in this scenario is zero.

Obviously, the modification ability strength of the above four structures follows the order of the Discrete structure > the Tree-like structure>the Chain structure>the Ring structure.

Therefore, with zero modification capability, the ring structure has the best security performance. That is, no modification or deletion of any comments can be performed in ring structure. To form a ring structure, users demand extensive cooperation efforts. That is, the first user that submitted a comment must know the pseudonyms of the subsequent users submitting the comment. This assumption is not easy to be realistic in a service-oriented social network. Therefore, a hybrid structure (ring and chain) is adopted to restrict the vendor's modification capability no more than $O(1)$ in the service-oriented social network, as shown in Figure 3.

4.3.2 | Generation of comments

Based on the comment content α_j and the comment value $cv_j \in [0, 1]$ for the vendor v, the user u_j, who uses the pseudonym $pid_{j;*}$, generates a local comment rev_j in the current time T_j (this time must be greater than all the timestamps embedded in the tok) as follows:

1. Generate a comment message: $m_j = (\alpha_j||cv_j||v||T_j)$.
2. Calculate the signature of the comment message:
   $\sigma_j = \text{Sign}_{pskj;*} = (m_j) = (\alpha_j||cv_j||v||T_j)$.
3. Generate local comments: $\text{rev}_j = \langle pid_{j;*}, \alpha_j, cv_j, v, T_j, \sigma_j\rangle$.
   Get the local comment rev_j of user u_j.

4.3.3 | Aggregation of comments

The aggregation of comment has two benefits:[30] effectively resisting the comment attacks and reducing the communication overhead. The user u_j locally generates the comment rev_j. Based on the local comment rev_j and the received aggregation comment REV_w, u_j generates an aggregation comment REV_j ($\text{REV}_j = \text{rev}_j$, if REV_w=null) using the aggregation algorithm. Then, u_j submits the aggregated comment REV_j to the vendor along with the local token. The vendor examines the validity of the REV_j and the local token, appends the aggregation comment REV_j to the TP list and broadcasts the updated TP list. The specific process of comments aggregation is described as follows:

Define three users u_1, u_2, and u_3, who respectively, obtained the pseudonym $pid_{1;*}, pid_{2;*}, pid_{3;*}$, the local comment generated by u_1: $\text{rev}_1 = \langle pid_{1;*}, \alpha_1, cv_1, v, T_1, \sigma_1\rangle$, user u_1 has received the initial token tok_1 from the vendor.

The process of passing tokens between these three users is described in Figure 2. Since u_1 is the first user, $\text{REV}_1 = \text{rev}_1$, u_1 passes REV_1 and token tok_2 to u_2. After a series of security verification, user u_2 aggregates its local comment rev_2 with REV_1 into REV_2, as follows:

$\text{rev}_2 = \langle pid_{2;*}, \alpha_2, cv_2, v, T_2, \sigma_2\rangle$

$\text{REV}_2 = \langle [pid_{1;*}, pid_{2;*}], [\langle \alpha_1, cv_1\rangle, \langle \alpha_2, cv_2\rangle], [T_1, T_2], v, \text{agg}(\sigma_1, \sigma_2)\rangle$

The same as before, u_2 passes REV_2 and token tok_3 to u_3. After a series of security verification, u_3 aggregates its local comment rev_3 and REV_2 into REV_3, as follows:

$\text{rev}_3 = \langle pid_{3;*}, \alpha_3, cv_3, v, T_3, \sigma_3\rangle$

$\text{REV}_3 = \langle [pid_{1;*}, pid_{2;*}, pid_{3;*}], [\langle \alpha_1, cv_1\rangle, \langle \alpha_2, cv_2\rangle, \langle \alpha_3, cv_3\rangle], [T_1, T_2, T_3],$

$v, \text{agg}(\sigma_1, \sigma_2, , \sigma_3)\rangle$

In the collaborative submission of comments, the next user who receives REV_i also follows the comment aggregation operations above to aggregate the comments, and then complete the collaborative submission of the comments.

4.3.4 | Submission of comments

The user u_j submits the aggregation comment REV_j and the token tok_{j+1} to the vendor. After receiving the aggregation comments, the vendor updates and broadcasts the TP list. After a given time interval, user u_j checks which pseudonym is associated with the current token in the TP list.

1. If the current token is related to $pid_{j;*}$, it means that u_j has successfully submitted REV_j. Then u_j forward the token to the nearby user, and the user who received the token can submit his local comments.
2. If the current tok is still related to $pid_{w;*}$, it means that the submission of u_j fails. Then u_j will take to cooperative submission by transmitting token and REV_j to users requesting tokens nearby.
3. If the current tok is related to other pseudonyms $pid_{x;*}$, it means that u_w has send the token to multiple users including u_x and u_j, and u_x has submitted the comment successfully. That is the submission of u_j fails. At this point, u_j will attempt to request a new token to complete the submission of the local comment REV_j.

4.3.5 | Verification of comments

The recipient of the comment (the vendor or next user) needs to verify that rev_j indeed generated by user u_j at time T_j, not forged by the vendor or any other illegal user. The correctness of the signature σ_j can be assured by the signature scheme in Section 3.1.

The specific verification method is described as follows:

If the comment is a separate submission, the recipient of the comment verifies the signature $\sigma_j = (K_j, S_j)$ of the user u_j by the following equation.

$$\alpha^{S_i} = K_i + Y_i + \alpha^{H_1(m_i)} \bmod q$$

In case the equation holds, the signature σ_j is effective; otherwise, the signature fails.

If the comment is an aggregation comment, the recipient of the comment verifies the signature $\sigma_j = (K_j, S_j)$ of the user u_j by the following equation.

$$\alpha^{S} = K \cdot Y_i \cdot \alpha^{\sum H_1(m_i)}$$

If the equation holds, the signature j of the comment is valid; otherwise, the signature is invalid.

Note that, u_j cannot forge a comment on u_w because it cannot get the key for the pseudonym $pid_{w;*}$, and u_j cannot replace the received comment with any other comments because there is a timestamp to prevent the comment from being replayed.

In addition, the token records the forwarding history, so u_j or replaced cannot forward a token without submitting REV_w or rev_j. When the vendor later receives a comment submitted by another user, the comment missing can be detected.

4.4 | Security analysis

4.4.1 | Comment linkability attacks

In the comment linkability attacks, if users are allowed to submit unlinked comments to vendors, malicious users may abuse the identity of legitimate users to produce false comments, which can compromise system performance.

The u_j's pseudonym $pid_{j;*}$ and the comment message m_j are associated with the unforgeable comment signature σ_j, and they constitute a triplet $< pid_{j;*}, m_j, j >$. Any user who does not have a private key associated with the pseudonym $pid_{j;*}$ cannot forge this triple. The comment rev_j is valid if and only if the vendor assures that the triplet is valid.

In addition, when KGC generates the pseudonym $pid_{j;*}$ and the corresponding private key $psk_{j;*}$ for the user u_j, KGC assures the validity of the unique ID of u_j and always keeps an association between $pid_{j;*}$ and id_j. Therefore, if a user submits a malicious comment with no linkability, KGC can link the malicious comment to the unique identity of its generating user. Thereby, Tesia can effectively resist the comment linkability attacks.

4.4.2 | Comment rejection attacks

In the comment rejection attacks, the vendor rejects all real but unwelcome comments. During the submission of the comments, user u_j tries to directly submit his comment rev_j to the vendor using tok_j multiple times. If all the trials fail due to the comment rejection attack or communication failure, u_j will pass the local tok_j and rev_j to a nearby user, say u_k, for cooperation submission. The user u_k needs to submit u_j's comment rev_j and its

own comment rev_k to the same vendor. The user u_k firstly assures the validity of the received tok_j and rev_j. Then he aggregates rev_j and rev_k into the aggregation comment REV_k. At last, he submits the aggregation comment REV_k *to* the vendor together with the tok_j for cooperation submission.

The vendor either accepts REV_k (which contains the previously rejected comment rev_j) or refuses it (containing the new rev_k). As comments are aggregated, the initiation loss will increase (reject a large number of useful comments) for launching the comment rejection attack. If the vendor eventually accepts the entire aggregated comment, it in fact accepts all the comments rejected before. Therefore, the submission of cooperation and the aggregation of comments can effectively counter comment rejection attacks.

4.4.3 | Comment modification attacks

In the comment modification attacks, by inserting false positives, modifying or deleting existing undesired true comments, the vendor can control its locally stored comments. In this article, digital signature technology is used to guarantees the comment content integrity. The comments are linked in the comment chain, and the number of comment chains depends on the number of tokens. Users can submit comments directly or cooperatively. The cooperation submission allows indirect submission of comment in the event of failure of direct submission. Since comments are submitted indirectly, the different users submitted comments will be aggregated to form a comments chain.

As shown in Figure 3, users can submit comments directly or indirectly. In Figure 3, the blank nodes indicate users who submitted comments individually. The nodes (eg, $U_1U_2U_3U_4$) inside the dotted circle denote these nodes submit the comments corporately, and all the users who submitted the comment form a group according to the cooperation relationship. The dotted arrow represents the delivery process of the token when the group submits the comment in cooperation, and the solid arrow represents the delivery process of the token when the comment is submitted directly.

This article indexes the users in each cluster. The smallest indexed user in the group is called the starting user, and the largest indexed user is called the end user. Smaller indexes show that the user gets the token earlier and submits its comments earlier. The larger indexed user can aggregate the former failed comments with its own comments and submits the aggregation comments to the vendor. Outside these clusters, the arrowed line indicates the direction in which the token is forwarded.

The following theorems show that the comment modification attack can be resisted.

Theorem 2. *If the vendor wants to insert comments into the comment chain, it must compromise the start user.*

Proof. Assume that the vendor has cracked the starting user u_s and gained all the signed private keys. Let u_j correspond to the end user u_v. The token forwarding proof list is $(PF_s \dots, PF_j, PF_v)$. If inserting a fake comment with the pseudonym u_m, then the token must be changed to $(PF_s \dots, PF_j, PF_m, PF_v)$, where

$PF_j = (pid_{j;*}, Sign_{pskj;*}(pid_{m;*}||T_j))$

$PF_m = (pid_{m;*}, Sign_{pskm;*}(pid_{v;*}||T_m))$

It is easy to verify the invalidity of the modified token because the end user with the maximum index outputs an unforgeable aggregation signature, which can detect forged comments, thus effectively failed the comment modification attack. ∎

Even if the vendor has successfully inserted a fake comment, the vendor must insert a false token forwarding certificate into the token, which means that the vendor must compromise the users who have used the token. Contradicts with the assumptions assumed in the security model of this article, so the success of comment insertion is not valid. This accomplishes the security certification.

Theorem 3. *If the vendor wants to delete the subchain of comments from the comment chain, it must compromise the current user and span all users later than the current user.*

Proof. A group of aggregation comments is considered as a whole since the vendor cannot separate them, and the changes can only be retained or deleted altogether. Vendors can only delete comments from given users and must delete all subsequent comments. The comment chain will be destroyed after deleting all subsequent comments. The invalidity of the comment chain can be detected unless subsequent comments are removed. ∎

If the vendor destroys the initiating user (gets the signature key of all subsequent users), the vendor will be able to strike out any number of consecutive comments from the initial user, even the whole comment cluster. The break of comment chain can be detected. Under the current security model assumptions, the probability of a vendor compromising all users keys is negligible.

4.4.4 | Sybil attack

In a particular single-vendor and service-oriented social network, KGC verifies the identity of each user u_j and generates a series of pseudonyms for it. The series of pseudonyms of the user u_j uniquely corresponds to the same identity id_j. We limit that different pseudonyms can only submit one

valid comment at the same time interval. If the same pseudonym submits multiple comments within the same time interval, KGC will track to and punish the user according to the unique identity id_j which corresponds to the pseudonym. This can effectively resist Sybil attacks.

Although this article cannot avoid malicious users who use different pseudonyms to launch Sybil attacks at different time intervals, it can still resist Sybil attacks to a large extent and improve the security of Tesia.

5 | PERFORMANCE EVALUATION

In this section, we analyze the performance and security of Tesia by comparing Tesia with another three traditional evaluation models, that is, the NCP model, bTSE model, and SEER model.

Before the simulation operation, we firstly define the following two indexes to measure performance of different models.

1. Comment SR
   SR denotes the rate of the number of successfully submitted comments to the total number of comments generated in the network.
2. Comment submission SD
   SD denotes the average DT between the comment generating time and the comment receiving time.

5.1 | Simulation setup

Based on the new aggregation digital signature scheme and the cooperation submission policy, Tesia in this article can effectively resist the comment linkability attacks and the comment modification attacks. Furthermore, the new model can effectively mitigate comment rejection attacks.

The comment linkability attacks and comment modification attacks have no effect in the submission process of comments. In our emulation, we just experiment the impact of comment rejection attack on the model performance. The vendor executes comment rejection attack by refusing all negative comments while accepting all positive comments.

The number of tokens in the analyze is set between [1,10], and each comment value is ranged in [0,1]. If the comment value is less than 0.5, it is considered negative, otherwise it is positive. When multiple comments are aggregated and submitted together, if their average value is not less than 0.5, the vendor accepts all comments, or rejects them all. In this emulation, 100 authenticated legitimate users are selected to submit comments to the same vendor. For a given token number between [1,10], 50 comment submissions are performed, and calculate the average SR and the average SD, respectively. The changing curves of SR and SD value with token numbers are analyzed, respectively, in different situations based on whether there is a comment rejection attack or not. In Tesia, the average delayed submission time for each comment in the submitted aggregated comment is taken as SD. In both models, SD only considers the case where the final submission is successful.

5.2 | Simulation results

5.2.1 | Under no comment rejection attack

In the case of no comment rejection attack, the analyzing results of the Tesia proposed in this article are analyzed in this section. In a NCP system, since each user submits their comments separately, their SR and SD have no relationship with the number of tokens in the system, that is, their SR and SD are fixed values.

In Tesia, the submission process of comments is limited by the token possession. Moreover, when a user is unable to submit a comment directly, the collaborative comment submission process will be triggered. We then simulate how the number of tokens impacts Tesia performance. Intuitively, when the number of tokens rises, users have more opportunities to submit comments, which improves model performance. In the bTSE model, the submission process of reviews is limited by token possession. Intuitively, when the number of tokens increases, users have more opportunities to submit evaluations, and the overall performance of the bTSE model will be correspondingly improved.

This instinct is verified by the results in Figures 4 and 5. As shown in Figure 4, the SR of Tesia first rises as the number of tokens increases, and gradually stabilizes at a fixed value as the number of tokens continues to increase. Due to the instability of the evaluation and submission mechanism of the bTSE model, the SR of the bTSE model first rises with the increase of the number of tokens, and the SR fluctuates with the continuous increase of the number of tokens. Although the SEER model also uses a synchronous token mechanism to ensure consistent and orderly evaluation transfer, the SR of the SEER model fluctuates greatly with the increase in the number of tokens, and the stability is poor. After the number of tokens reached 5, it caused confusion in the evaluation submission, and the SR value decreased significantly.

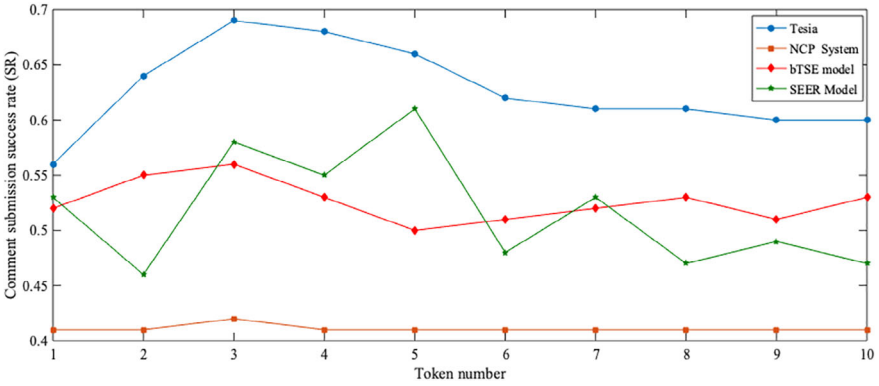


FIGURE 4 Tesia vs other Model SR comparison without comment rejection attack

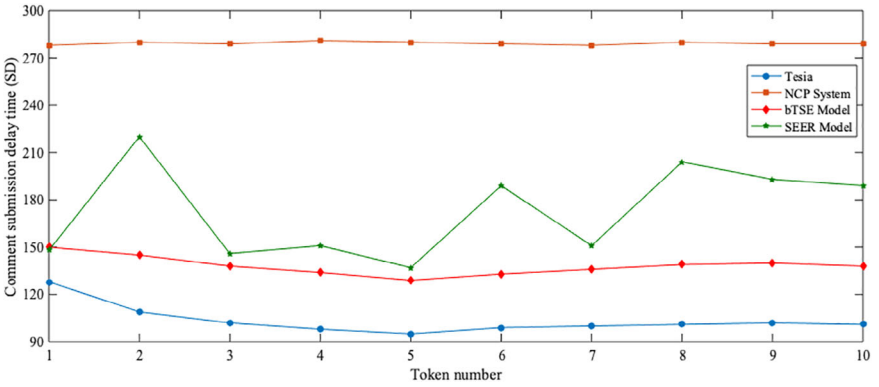


FIGURE 5 Tesia vs other Model SD comparison without comment rejection attack

As shown in Figure 5, the SD of Tesia decreases slightly and at last stabilizes at a fixed value as the number of tokens increases. The SD of the bTSE model decreases with the increase of the number of tokens, and the SD time is the lowest when the number of tokens is 5. As the number of tokens continues to increase, SD slowly increases and gradually stabilizes. There is no obvious correlation between the changes in the SD of the SEER model and the fluctuations with the changes in the number of tokens. An arguable phenomenon can be observed from the analyzing results: when the number of tokens surpasses a certain value, SR and SD will stabilize. The reasons are as bellow: the initial user can readily gain the token and submit a comment when there are more tokens in the network. After a user's comment is submitted to a vendor or forwarded to another user, the user no longer participates the comment submission process. The participating users' networks become sparse, and as the network density decreases, the chances of these current users to receive tokens are also increased. When all token requirements are met, the SR gradually stabilizes at affixed value. As shown in Figure 5, under no comment rejection attack, the SD of Tesia is much lower than the NCP system, which is approximately doubled. Because the comment cooperation submission in Tesia can effectively reduce SD. Figures 4 and 5 show that the Tesia has better overall performance than bTSE model without the comment rejection attack, in which the SR value is increased by 15% to 20%, and the SD value is reduced by about 30%. The Tesia is more stable than the SEER model because the Tesia cooperates with users to submit aggregated comments in a special network structure, and the SEER model submits and evaluates distributedly in the form of integrated chains, which does not guarantee the performance and stability of the entire model.

5.2.2 | Simulation results under comment rejection attack

In the case of the comment rejection attack, Figures 6 and 7 indicates the performance contrast of Tesia and the NCP system. It can be observed from the experimental data that the SR of the NCP system drops by about 15% when there is a comment rejection attack. This is because the NCP system is not equipped with any security mechanism to resist the comment rejection attack, and it will definitely cause a decrease in SR under the refusal of comments. Since users in the NCP system submitted their comments directly and only calculated the DT of the successfully submitted comment, so the SD of the NCP system did not show any significant changes. Tesia's SR is higher than that of NCP system, which is approximately twice as large as the NCP system. Under the comment rejection attack, the SR of the bTSE model decreased by about 20%, and the SD value increased by 20% to 25%. Because the security mechanism of the bTSE model is not enough to resist the comment rejection attack, the loss of comment during the comment submission process will cause the SR to decrease. The SR of the SEER model decreased by about 10% compared with the nonevaluation

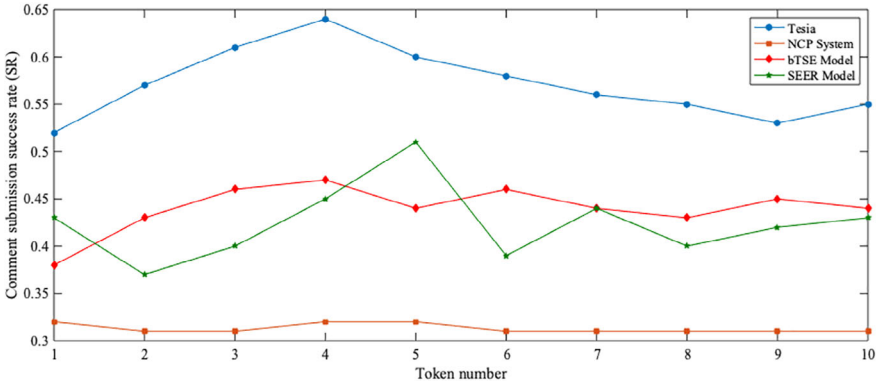


FIGURE 6 Submission Rate with comment rejection attack

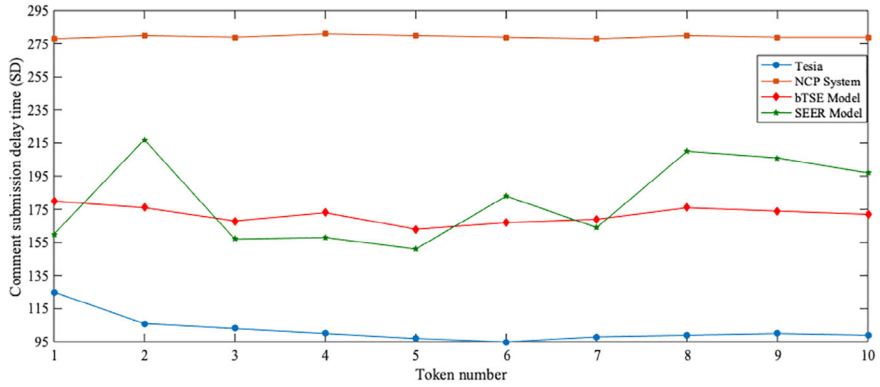


FIGURE 7 Submission delay time with comment rejection attack

TABLE 1 Comparison of the comment attack resistance of each model

	Comment linkability attack	Comment rejection attack	Comment modification attack	Sybil
NCP	Y	N	N	N
bTSE	Y	Y	Y	Y
SEER	Y	Y	Y	N
Tesia	Y	Y	Y	Y

rejection attack. The SR still fluctuated greatly with the increase of the number of tokens. The SD increased irregularly with the increase of the number of tokens.

Comparing the Figures 4 and 6, it can be seen that the SR of Tesia has dropped only about 10% due to the comment rejection attack. These results thanks to the cooperation submission and comment aggregation mechanism has been performed. While in NCP system, without any security policy, the SR has dropped about 30% due to the comment rejection attack. Comparing the Figures 5 and 7, only considering successful comment submission, SD of Tesia is almost unchanged. Tesia's SR is higher than that of NCP system, which is approximately twice as large as the NCP system. Tesia has a higher SR than the bTSE model, which is about 33% better than the bTSE model. Tesia's SD is approximately 55% lower than the bTSE model. At the same time, Tesia has a more stable and higher SR than the SEER model, which is about 41% higher than the bTSE model. Tesia's SD is reduced by about 60% compared with the bTSE model. The above emulation results show that Tesia can maintain good performance under the comment rejection attack.

5.3 | Cost analysis

In this section, the security and efficiency of Tesia and NCP model, bTSE model, and SEER model are compared and analyzed. The attack analysis compares their ability to resist the linkability attack, the rejection attack, the modification attack, and the sybil attack proposed in Section 3.2. The analysis results are as shown in Table 1.

	NCP	bTSE	SEER	Tesia
Average comment submission delay time/ms	278	133	140	95

TABLE 2 Comparison of the optimal comment submission delay time of each model

	S-one token	S-one comment	S-n token	S-n comment
bTSE	2|G|	2|G|	2|G|	2n|G|
SEER	2|G|	2|G|	4|G|	(3n+1)|G|
Tesia	$2Z_p^* + H$	$2Z_p^* + H$	$2Z_p^* + nH$	$2nZ_p^* + nH$

TABLE 3 Comparison of computation of each model

Table 1 shows that the NCP only uses pseudonym technology, and it can only resist the comment linkability attacks. Simultaneously, SEER cannot resist Sybil attacks. Tesia and bTSE can both resist the four comment attacks mentioned above.

The efficiency analysis compares the performance of the four trusted service evaluation models from the aspects of communication overhead.

The communication overhead can be obtained from the average comment submission SD value in the simulation experiment. The SD value can represent the communication overhead of the entire model. The optimal SD values of these four models without evaluation attacks are as shown in Table 2.

From Table 2, we can see that the optimal SD cost of Tesia is significantly lower than the other three models in the case of no comment attack.

In the computation overhead analysis of the model, we do not calculate the comment content and the size of the public string, because their size is negligible compared with the signature. Since each user in the NCP model submits their evaluation independently, there is no credibility of the synchronization token mechanism and digital signature to ensure the evaluation, so we just carry on the analysis on the other three models. The computation cost of the model is reflected by the following four indicators: the size of signature on one token(S-one token), the size of signature on one comment (S-one comment), the size of n-aggregated signatures on tokens (S-n token), the size of k-aggregated signatures on comments (S-n comment). Where |G| represents a bilinear mapping; Z_p^* represents a multiplication operation on Z_p^*; H represents a hash operation. The computational overhead of a |G| bilinear mapping is about 10 times that of a multiplication on Z_p^*, and the computational overhead of the H hash operation is negligible.

Based on Table 3, the computation cost of Tesia based on the multiplication operations is significantly lower than the other two models using bilinear mapping. The computation cost of SEER will linearly increase with the number of comments and lead to low efficiency of the model. Therefore, Tesia's computation overhead is better than the other two models.

6 | CONCLUSIONS

This article proposes a new aggregation digital signature scheme, and then designs a trusted effective service evaluation model named Tesia. In Tesia, users submit comments in a distributed manner in the form of a chain, which improves the comment SR and reduces the comment DT. By ensuring the integrity of the comments, the ability for IoT systems to arbitrarily modify comments is reduced, and the security of the user's comments is guaranteed. Security analysis shows that the trusted efficient service evaluation model can effectively resist the current mainstream service comment attacks without relying on any trusted third party. The experimental results show that in the case of comment rejection, Tesia is significantly better than the NCP comment system in terms of the SR and DT of comments. However, the mechanism in this article just focuses on the centralized architecture, and how to apply this mechanism into distributed architecture is our key points in the future work.

ACKNOWLEDGMENTS

This study was funded by Foundation of National Natural Science Foundation of China (grant number:61771231, 6150028, 61672321, 61771289, 61832012, 61373027), Natural Science Shandong Province (grant number: ZR2016FM23, ZR2017MF010, ZR2017MF062), Key Research and Development Program of Shandong Province (NO. 2019GGX101025), EU Horizon 2020 DOMINOES Project (Grant Number: 771066).

ORCID

Yilei Wang https://orcid.org/0000-0003-4054-0549

REFERENCES

1. Liang X, Lin X, Shen XS. Enabling trustworthy service evaluation in service-oriented mobile social networks. *IEEE Trans Parallel Distrib Syst*. 2013;25(2):310-320.

2. Liang X, Li X, Lu R, Lin X, Shen X. Seer: a secure and efficient service review system for service-oriented mobile social networks. In: O'Conner L, ed. *2012 IEEE 32nd International Conference on Distributed Computing Systems*. Macau, China: IEEE; 2012:647-656.
3. Liang X, Xu L, Luan TH, Lu R, Shen X. Morality-driven data forwarding with privacy preservation in mobile social networks. *IEEE Trans Veh Technol*. 2012;61(7):3209-3222.
4. Wang H, Zheng Z, Wu L, Ping L. New directly revocable attribute-based encryption scheme and its application in cloud storage environment. *Clust Comput*. 2017;20(3):2385-2392.
5. Wang H, He D, Shen J, Zheng Z, Yang X, Au MH. Fuzzy matching and direct revocation: a new CP-ABE scheme from multilinear maps. *Soft Comput*. 2018;22(7):2267-2274.
6. Zhu H, Tan Y-a, Zhu L, Wang X, Zhang Q, Li Y. An identity-based anti-quantum privacy-preserving blind authentication in wireless sensor networks. *Sensors*. 2018;18(5):1663.
7. Xiao X, Zheng X, Zhang Y. A multidomain survivable virtual network mapping algorithm. *Secur Commun Netw*. 2017;2017:5258010.
8. Lin Q, Yan H, Huang Z, Chen W, Shen J, Tang Y. An ID-based linearly homomorphic signature scheme and its application in blockchain. *IEEE Access*. 2018;6:20632-20640.
9. Tian H, Chen Z, Chang C-C, et al. Public audit for operation behavior logs with error locating in cloud storage. *Soft Comput*. 2019;23(11):3779-3792.
10. Zheng X, Liu H. A scalable coevolutionary multi-objective particle swarm optimizer. *Int J Comput Intell Syst*. 2010;3(5):590-600.
11. Wang H, He D, Shen J, Zheng Z, Zhao C, Zhao M. Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing. *Soft Comput*. 2017;21(24):7325-7335.
12. Wang Y, Bracciali A, Li T, Li F, Cui X, Zhao M. Randomness invalidates criminal smart contracts. *Inf Sci*. 2019;447:291-301.
13. Yilei W, Zhao M, Hu Y, Gao Y, Cui X. Secure computation protocols under asymmetric scenarios in enterprise information system. *Enterp Inf Syst*. 2019;1-21. https://doi.org/10.1080/17517575.2019.1597387.
14. Lifeng Z, Wang Y, Li F, Hu Y, Au MH. A game-theoretic method based on Q-learning to invalidate criminal smart contracts. *Inf Sci*. 2019;498:144-153.
15. Luan TH, Cai LX, Chen J, Shen X, Bai F. Vtube: towards the media rich city life with autonomous vehicular content distribution. In: Prasant M, ed. *2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. Utah, USA: IEEE; 2011:359-367.
16. Yu Z, Wu Z, Trappe W. Adaptive location-oriented content delivery in delay-sensitive pervasive applications. *IEEE Trans Mob Comput*. 2011;10(3):362-376.
17. Li T, Gao C, Jiang L, Pedrycz W, Shen J. Publicly verifiable privacy-preserving aggregation and its application in IoT. *J Netw Comput Appl*. 2019;126:39-44.
18. Riad K, Hamza R, Yan H. Sensitive and energetic IoT access control for managing cloud electronic health records. *IEEE Access*. 2019;7:86384-86393.
19. Rajan H, Hosamani M. Tisa: toward trustworthy services in a service-oriented architecture. *IEEE Trans Serv Comput*. 2008;1(4):201-213.
20. Yan W, Lei L. Two-dimensional trust rating aggregations in service-oriented applications. *IEEE Trans Serv Comput*. 2011;4(4):257-271.
21. Ayday E, Fekri F. Iterative trust and reputation management using belief propagation. *IEEE Trans Depend Secure Comput*. 2011;9(3):375-386.
22. Alamir P, Navimipour NJ. Trust evaluation between users of social networks using the quality of service requirements and call log histories. *Kybernetes*. 2016;45(10):1505-1523.
23. Das A, Islam MM. SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems. *IEEE Trans Depend Secure Comput*. 2012;9(2):261-274.
24. RRR R, Thomas C. Evaluation of mitigation methods for distributed denial of service attacks. In: Xie W, ed. *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. Singapore: IEEE; 2012:713-718.
25. Aad I, Hubaux JP, Knightly EW. Impact of denial of service attacks on ad hoc networks. *IEEE/ACM Trans Netw*. 2008;16(4):791-802.
26. Douceur JR. The sybil attack. In: KFRA D, ed. *International Workshop on Peer-to-Peer Systems*. Cambridge, MA: Springer; 2002:251-260.
27. Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Netw*. 2003;1(2):293-315.
28. Newsome J, Shi E, Song D, Perrig A. The sybil attack in sensor networks: analysis & defenses. In: Gastpar M, ed. *Third International Symposium on Information Processing in Sensor Networks*. Berkeley, CA: IEEE; 2004:259-268.
29. Wei W, Xu F, Tan CC, Li Q. Sybildefender: defend against sybil attacks in large social networks. In: Randall Berry JM, ed. *The 31st Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2012)*. Orlando, FL: IEEE; 2012:1951-1959.
30. Wu L, Xu Z, He D, Wang X. New certificateless aggregate signature scheme for healthcare multimedia social network on cloud environment. *Secur Commun Netw*. 2018;2018:1-13.

How to cite this article: Li F, Ge R, Zhou H, Wang Y, Liu Z, Yu X. Tesia: A trusted efficient service evaluation model in Internet of things based on improved aggregation signature. *Concurrency Computat Pract Exper*. 2020;e5739. https://doi.org/10.1002/cpe.5739