Blockchain-Based User Authentication with Anonymity for Internet of Things Applications

YongJoo Lee, Keon Myung Lee, Sang Ho Lee

Abstract: Security threats that target identities (IDs) have increased considerably in recent years. Various network attacks attempt to discover IDs that can be used in future attacks to obtain private information. In this paper, we propose a blockchain-based user authentication approach that can be used by various end-users for internet of things (IoT) applications. The proposed approach uses single-use authentication parameters and does not require any private information. It is based on the child-key mechanism of the hierarchical deterministic (HD) wallet. The HD wallet is accepted as a standard of Bitcoin and is in turn based on the elliptic curve digital signature algorithm. The authentication parameters of the proposed approach were created using the HD wallet mechanism. The transaction of the authentication key stored in the distributed ledgers of the blockchain could be shared by various IoT servers for subscription to different services, and the user account of the blockchain could be connected to the IoT servers for payments without a membership procedure. We used only hash values for an authentication request to protect against network attacks. The proposed approach could decrease the system load by using a lightweight feature with few parameters and simplify the approach without the need for additional procedures by IoT servers. We verified that the security requirements of the proposed approach were satisfied, by analyzing the transmitted parameters. Furthermore, we evaluated the security vulnerabilities from various threats and analyzed attack scenarios. Thus, we propose that the authentication servers verify an original EC domain parameter hash value and the hash reuse to protect against fake attacks by network sniffing and spoofing. We have also summarized the originality and the characteristics of the proposed research by comparing it with closely related studies and concluded with a future research guide.

Keywords: Blockchain, Privacy, Authentication, Peer to Peer, Identity, Elliptic curve cryptosystem

I. INTRODUCTION

The IoT is a growing paradigm with technical, social, and economic significance that comprises a wide ecosystem of interconnected services and devices such as sensors, consumer products and cars. Industry 4.0 and industrial internet of things (IoT) are frequently and rightfully associated with IoT focusing on digitizing industries. Industry 4.0 focuses on creating smart products, procedures, and processes, and the industrial IoT well manages the complexity and can be used to more efficiently manufacture goods.

II. ANONYMITY-PRESERVING USER AUTHENTICATION

This section describes the requirements of the proposed approach, a network configuration and a detailed description. The proposed approach consists of a registration, request of authentication, and authentication procedure.

2.1. The proposed User Authentication

In the blockchain-based decentralized network, many IoT servers can share distributed ledgers in a blockchain to keep...
The integrity of authentication data without a trusted third party. When a user needs to subscribe a service from a IoT server, the user can ask to be authenticated by using the parameters stored in the ledgers. Storing any private data in each server can be a big threat in the centralized network. Here, we aim at providing a light weight user authentication approach which provides anonymity and privacy. Anonymity in a user account of a blockchain will be connected to the proposed approach. The approach is based on a decentralized peer-to-peer (P2P) protocol where the parameters are accessed by various IoT servers without a trusted third party. Users can be connected to other IoT servers by the blockchain account for a payment. When the user wants to change the registered parameters, the user does not need to notify any IoT server of the change because a blockchain manages transactions in a chronological order by an internal time server. Figure 1 shows the simplified structure of the proposed approach.

2.1.1. Detailed Description

The degree of privacy is expressed through indistinguishable transactions on the blockchain. Although many blockchain systems take measures to protect privacy, it is not robust [7, 8]. Fixed IDs or keys can leave a clue for user private details in a specific service. A specification of hierarchical deterministic (HD) wallets was proposed in 2012 and accepted as Bitcoin standard BIP32, which generates wallet keys from a single master seed [9, 10]. The HD wallets based on Elliptic Curve Digital Signature Algorithm (ECDSA) which is based on the algebraic structure of elliptic curves over finite fields [11, 12]. Bitcoin uses elliptic curve signatures on curve secp256k1. We use the HD wallets to create a child key from a master seed and produce hash values from the child key. The proposed approach consists of three procedures: registration, request of authentication, and authentication. ECDSA uses Elliptic Curve (EC) domain parameters to generate a master key pair that strengthens ECDSA based approach. Figure 2 shows a brief procedure.

Let \( I \) be the set of users; \( F \) is the set of child key; \( \mathbb{F} \) is the prime finite field, which contains \( p \) elements, where each element is a set of integers \( \{0, 1, 2, \ldots, p - 1\} \); \( EC \) is the domain parameter; \( T \) over \( \mathbb{F} \) is a sextuple, \( T = (p, a, b, G, n, h) \), \( T(G) \) is an elliptic curve base point, which is a generator of the elliptic curve with large prime order \( T(n) \), and \( n \) is the integer order of \( T(G) \). We use \( \cdot \) to denote the elliptic curve point multiplication by a scalar. We assume that all parameters are appropriately generated and verified. A user selects a cryptographically secure random integer, \( d_i^A \) in the interval \( [1, n - 1] \). The corresponding public key is \( Q^A = (d_i^A \cdot G) \). The elliptic curve master key pair \( \text{ECMPK}_i \) is

\[
\text{ECMPK}_i = (d_i^A)(Q^A)
\] (1)

The user generates a verifying key, \( \text{VK} \),

\[
\text{EK} = \text{hash}(G, n) \quad \text{(2)}
\]

\[
\text{VK} = \text{hash}(d_i^A, \text{EK}) \quad \text{(3)}
\]

The user registers transaction, \( \text{Tran}_i \),

\[
\text{Tran}_i \leftarrow (d_i^A, \text{VK}) \quad \text{(4)}
\]

We do not use an encryption algorithm for \( \text{Tran}_i \) because it consists of a random integer and a hash value. The registration is a blockchain transaction that is broadcast to every node. A new block is added only after a successful verification. The transaction address, which is the hash value of the public key, is returned as a transaction ID. \( \text{Tran}_i \) will be shared by various IoT servers which want to authenticate the user.

Figure 1. Simplified procedures

Figure 2. Proposed procedures
- Request of authentication by a user
When the user wants to get authenticated by an IoT server for a service subscription, hash values from a master key pair are generated. The user generates a pseudorandom sequence of a child key under a given ECMSK using only knowledge and calculates two hash values ($HV_i$ and $HD$),

$$HV_i = \text{hash}(j, Q^{\lambda})$$  
\hspace{1cm} (5)  
$$d_i = (d_i + HV_i)(mod\ p)$$ \hspace{1.5cm} (6) 
$$HD_i = \text{hash}(d_i, EK)$$ \hspace{1cm} (7)

The user transmits three hash values, ($HV$, $HD$, and $EK$) to the IoT server. The first hash value, $HV$, will be transmitted to enable the server to create a child key under the same master seed. The second, $HD$ is for comparison with the received value and the calculated value. The last, $EK$ will be used to verify the original EC domain parameters. The master key, $Q^{\lambda}$ is an important source to create a child key. We transmit the hashed value of $j$ and $Q^{\lambda}$ to protect against network attacks. The result of the hash function from a bit change is totally different; hence, $j$ cannot be reused even after a single usage.

- Authentication by IoT Server
When the user asks an authentication to a IoT server with the hash values, the server gets $HV$ of the user from the ledger of a blockchain and generates NVK and nc from $HV$ and $d_i$. Then the server calculates NHD from nc,

$$NVK \leftarrow \text{hash}(d_i, EK)$$ \hspace{1cm} (8)  
$$nc = (d_i + HV_i)(mod\ p)$$ \hspace{1cm} (9) 
$$NHD \leftarrow \text{hash}(nc, EK)$$ \hspace{1cm} (10)

The server compares NVK and VK to verify the original parameter set. If they are the same, the server can confirm that the received hash values were generated from the same EC domain parameters. Next, the server compares NHD with $HD$ to authenticate the user. If they are the same, the server can confirm that the new child key was generated from the same master seed which the user created for the first time. Then the IoT server can authenticate the user, who requested the authentication, as the owner of the account. If authenticated, the user registers the hash value $HV_i$ to the blockchain to prevent a hash reuse attack. The hash value is publicly known as a one-way function which is cryptographically secure. The result of the hash function from a bit change is totally different; hence, $j$ cannot be reused even after a single usage.

If blockchain is private type and located in a secure place, registering the hash value is not essential. Figure 3 (a) shows a sequence diagram among a user, an IoT server and a blockchain and figure 3 (b) shows a flow chart in the IoT server which fulfills an authentication mechanism by receiving parameters from the user and retrieving parameters from the blockchain [17,18].

2.2. Use-case Scenario for Smart Factory
We describe a use-case scenario to show how the proposed approach can be adopted in smart factory as an IoT application. We define user groups and a network configuration for the end-users in smart factory.

2.2.1. Definition of role-based user group
We define user groups according to their roles to develop a use-case for industrial IoT applications as shown in Table 1. Authentication methods are assigned to each user group.

![Sequence diagram and flow chart](image-url)

Figure 3. Sequence diagram and flow chart

Identity/password (ID), authentication token (Token), media access control (MAC) address-based scheme, certificate-based scheme (Cbs), and cryptography-based scheme (Cry) are described as an authentication method. One-way and Mutual authentications are described as an authentication direction as shown in Table 1. We propose a cryptography-based approach for one-way user authentication which is commonly used for high-security-level applications in IoT applications.
Table 1: Role-based user group definitions in Smart Factory

<table>
<thead>
<tr>
<th>User-group</th>
<th>Role</th>
<th>Type</th>
<th>Method</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managers</td>
<td>Monitoring</td>
<td>One way</td>
<td>ID, Cbs</td>
<td>Security, efficiency</td>
</tr>
<tr>
<td>System Command</td>
<td></td>
<td>One way</td>
<td>ID, Cbs, Cry</td>
<td>Security</td>
</tr>
<tr>
<td>Security Control</td>
<td></td>
<td>One way</td>
<td>ID, Cbs, Cry</td>
<td>Security</td>
</tr>
<tr>
<td>Applications</td>
<td>Event, command</td>
<td>One way</td>
<td>Token, MAC</td>
<td>Efficiency</td>
</tr>
<tr>
<td>End_users</td>
<td>Customer, designer, provider, manufacturer</td>
<td>One way</td>
<td>ID, Cbs, Cry</td>
<td>Security, privacy, anonymity</td>
</tr>
</tbody>
</table>

2.2.2. Decentralized network configuration for Smart Factory

Using the defined role-based user groups from Table 1, we consider machine-to-machine (M2M) devices, network, and application domains with end-users in the blockchain-based decentralized environment. The Internet of service in the application domain is connected with an open API, which is provided by the IoT Platform. The end users in the application domain can be connected to the cloud or IoT platform. The precise architecture depends on the development. IBM uses a cloud infrastructure for blockchain services to track high-value items as they move across supply chains. The service capabilities of the IBM Watson IoT platform enable users to add IoT data to private distributed ledgers that can be included in shared transactions [6]. We assume an interoperable private blockchain. Devices with sensors and actuators are connected with M2M communication. The IoT platform, cloud, and blockchain are located in the network domain. Figure 4 shows the network configuration with existing implementable technologies.

III. RESULTS AND DISCUSSION

We evaluated the security requirements and vulnerabilities. We also analyzed possible attack scenarios including network and fake attacks. We compared the proposed approach with closely related studies to show originality and differences in this section.

3.1. Security Evaluation

- Vulnerability from the parameter exposure: We analyzed vulnerabilities from registering to the blockchain. \( d^A \) is chosen by \( n \) and lies between 1 and \( n - 1 \). Attackers need \( Q^A \) to create the authentication parameters. The attackers need elliptic curve base point \( T(G) \) to create \( Q^A \). If an attacker can easily guess \( T(G) \), it can be risky. But the probability to guess the EC domain parameter \( T(G) \) without any knowledge is \( P = \frac{1}{2^n} \), where \( n \) is the bit size of the EC domain parameters \( T(G) \). The common bit size of \( n \) is from 163 to 512, and \( T(G) \) is bigger than \( n \). The value is extremely small. If we view it from a different attack point, the attackers can generate a new EC domain parameter set which satisfy the original \( d^A \), because \( d^A \) is chosen between 1 and \( n - 1 \). Hence, we register \( VK \) to verify the original EC domain parameters from the hash value of \( T(n, G) \),

\[
\text{hash}(d^A, \text{hash}(G, n)) \neq \text{hash}(d^A, \text{hash}(G', n'))
\]

- Vulnerability of the authentication parameters: the user transmitted three hashed values by a P2P network in the approach. A hash value is one-way and the original of the hash value cannot be extracted. Although, attackers successfully capture the authentication parameters, they cannot calculate the original keys,

\[
d^A \times TK \Rightarrow \text{hash}(d^A, EK) \equiv (j, Q^A) \times T(G) \Rightarrow \text{hash}(j, Q^A)\]

\[
d^A \times EK \Rightarrow \text{hash}(d^A, EK) \equiv (j, Q^A) \Rightarrow \text{hash}(Q^A, EK)
\]

Currently, secure hash algorithm (SHA)-256 is accepted by many blockchains including Bitcoin. Collision was detected in SHA-1, but SHA-256 included in SHA-2 is publicly known as secure.
3.2. Analysis of Attack Scenarios

We analyzed possible attacks during the user authentication performance. We supposed attacks of network sniffing, fake user, and malicious user.

Case 1: Network Attack: Data sniffing/spoofing

The sniffing attack is the interception of data by capturing the data traffic using a sniffer, which is an application to capture network packets. We suppose that the attackers attempt to obtain the hashed values. Figure 5(a) shows the data sniffing attack scenario.

Figure 5. Network attack scenario

The parameters transmitted to the IoT server are only hashed values. Although, attackers successfully obtain the hashed values by network attacks, they cannot extract the original parameters.

Case 2: Fake attack by the same hash values

An attacker can pretend to be a fake user by sending the same data captured from network attacks instead of trying to extract the original parameters. Suppose that the attackers get the hash values from the network attacks and attempt to be a fake user by sending the same data in the next transaction. Figure 5(b) shows the fake attack scenario. We do not reuse a child knowledge, j, and the hash value from the same j is thrown out in the proposed approach. A user saves the hash value \(HV^j\) generated from a random integer after a single use. The other party checks the reuse of the hash value before accepting the authentication. Hence, even if a fake user sends the same data, the authentication cannot be accepted.

Case 3: Fake attack by a new parameter set

Attackers can try to get authenticated by the original \(d^A\) and a new parameter set. Suppose that the attackers generate new \(EK^I\) from \(G\) and \(d^A\), new \(HV^I\) from \(Q^A\), and new \(HD^I\) from \(E\) and \(d^I\). This attack can be easily protected by comparing \(VK^I\) with \(VK\) which is generated by the original \(n\) and \(G\) of the EC domain parameter because the original \(n\) and \(G\) are included in the hash value, \(EK\). The EC domain parameter, \(n\) and \(G\) cannot be extracted by the attackers. Figure 6 shows the attack procedure by a new parameter set.

Figure 6. Analysis of the fake attack scenario by a new parameter set

3.3. Comparable ECC and RSA Security Level

The public key of the ECC-based scheme represents a point on a particular Elliptic Curve (EC). It contains an identification byte, a 32-byte X coordinate, and a 32-byte Y coordinate defined in secp256k1. Figure 7 (a) shows a point on a simplified EC used by Bitcoin, \(y^2 = x^3 + 7\), over a field of contiguous numbers. The public key size can be reduced by 50% without changing any fundamentals by dropping the Y coordinate. Only two points along the curve share any particular X coordinate, therefore a 32-byte Y coordinate can be replaced with a single bit to indicate whether the point is on the top side or the bottom side [8]. Figure 7 (b) shows comparable key sizes for ideal symmetric ECC-based and RSA-based schemes. Smaller key sizes are required to provide the equivalent security level [6, 7].

Published By: Blue Eyes Intelligence Engineering & Sciences Publication

Retrieval Number: C10680183C19/19©BEIESP
3.4. Comparison and discussion

- Evaluation of the approach requirements: The proposed approach is designed for being used by multiple servers and connected with an account for a payment. The approach has a light-weight feature with few hash parameters. Table 2 shows the evaluation for each requirement.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymity</td>
<td>A user does not need to reveal an identity or private information.</td>
</tr>
<tr>
<td>Privacy</td>
<td>The authentication parameters are hash values and do not include user information. Therefore, anonymity in a blockchain account is connected to IoT servers.</td>
</tr>
<tr>
<td>Light-weight</td>
<td>We use three parameters for an authentication request. An authentication procedure is one-step without an additional membership procedure by IoT servers for service subscription.</td>
</tr>
<tr>
<td>Multiple-server</td>
<td>Transactions stored in distributed ledgers can be shared by various IoT servers for the same user account.</td>
</tr>
</tbody>
</table>

- Comparison of related studies: We listed the closely related studies here to verify the necessity and originality of the proposed research in table 3. In the proposed approach, the user can subscribe many services by various IoT servers without subscription with private data, through one-time parameter registration to blockchain. The users do not depend on the security of a specific system or a trusted third party because the users choose and register the authentication data by themselves. Authentication parameters do not include any private information to provide anonymity.

<table>
<thead>
<tr>
<th>Difference</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aref et al. [13]</td>
<td>Application domain security provide authentication between entities: sensitive data is recommended to be stored in the secure storage of the M2M system, security depends on the trusted and centralized M2M system.</td>
</tr>
<tr>
<td></td>
<td>Not based on decentralized control.</td>
</tr>
<tr>
<td></td>
<td>No user control in security and storage. Need a trusted third party.</td>
</tr>
<tr>
<td>Sergio et al. [14]</td>
<td>Capability-based security approach to manage access control in the IoT. A capability is a communicable token of authority. It was developed for a centralized IoT environment.</td>
</tr>
<tr>
<td></td>
<td>Need a trusted third party and centralized storage.</td>
</tr>
<tr>
<td></td>
<td>No anonymity and privacy mechanisms.</td>
</tr>
<tr>
<td>Jason et al. [15]</td>
<td>Role-based access control uses smart contract for challenge and response protocol. The user role is issued by the trusted organization, and other users verify the role through the role-issuing organization for every service as a trusted third party.</td>
</tr>
<tr>
<td></td>
<td>No privacy and anonymity mechanisms. System loads to verify the roles every time. Not light-weight</td>
</tr>
<tr>
<td>Yanqi et al. [16]</td>
<td>Blockchain-based fair payment. It uses the broker-less publication and subscribes a protocol by the blockchain to exchange private data.</td>
</tr>
<tr>
<td></td>
<td>No mechanisms for privacy and against ID attacks.</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

We proposed a user authentication approach that can be useful in the convergence of blockchain and IoT applications. We focused on providing anonymity and privacy to protect the private information of users. We created single-use parameters by using the HD wallet child-key mechanism based on ECDSA. The authentication parameters registered in the distributed
ledgers of the blockchain could be shared by various IoT servers for service provisions without additional membership procedures. The user account of the blockchain was connected to the IoT servers for the payments of service subscriptions. A user who registered the authentication parameters to the blockchain did not have to notify the IoT servers after changing the parameters, because the transactions in the blockchain were updated by timestamps. The proposed approach decreased the system load by using a lightweight feature with few parameters for an authentication request. We accepted the blockchain technology to enable entities to act in a decentralized environment without a trusted third party and connect a user account to an IoT service for payments. Then, we evaluated the procedure security and analyzed the attacks. We confirmed that the proposed approach was secure from the analyzed attacks. Although the proposed protocol offers several advantages, it has room for improvement. In the future, we intend to develop a detailed access model with a policy structure. A mutual authentication approach between two entities for cooperation should also be developed to the required security level with decentralized control.

ACKNOWLEDGMENT

This work was supported by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea, Republic of Korea (Grant no.: NRF-2017M3C4A7069432).

REFERENCES