Verifiable Credential Proof Generation and Verification Model for Decentralized SSI-Based Credit Scoring Data

Kang Woo CHO†(a), Byeong-Gyu JEONG†(b), Nonmembers, and Sang Uk SHIN†(c), Member

SUMMARY  The continuous development of the mobile computing environment has led to the emergence of fintech to enable convenient financial transactions in this environment. Previously proposed financial identity services mostly adopted centralized servers that are prone to single-point-of-failure problems and performance bottlenecks. Blockchain-based self-sovereign identity (SSI), which emerged to address this problem, is a technology that solves centralized problems and allows decentralized identification. However, the verifiable credential (VC), a unit of SSI data transactions, guarantees unlimited right to erasure for self-sovereignty. This does not suit the specificity of the financial transaction network, which requires the restriction of the right to erasure for credit evaluation. This paper proposes a model for VC generation and revocation verification for credit scoring data. The proposed model includes double zero knowledge - succinct non-interactive argument of knowledge (zk-SNARK) proof in the VC generation process between the holder and the issuer. In addition, cross-revocation verification takes place between the holder and the verifier. As a result, the proposed model builds a trust platform among the holder, issuer, and verifier while maintaining the decentralized SSI attributes and focusing on the VC life cycle. The model also improves the way in which credit evaluation data are processed as VCs by granting opt-in and the special right to erasure.

key words: blockchain, decentralized SSI, verifiable credential, credit scoring, fintech

1. Introduction

Since 2010, ongoing advancements in mobile computing have allowed various forms of financial transactions, such as business, banking, and shopping, to be conducted over online networks. These changes have directly involved all forms of financial transactions; furthermore, the management of public funds, etc., has increased the demand for related technologies. Research concerned with online financial transactions has aimed to eliminate the need for offline exchange and paperwork. This research has developed in the direction of providing a consistently satisfactory experience on the consumer side and an efficient and secure financial transaction environment on the business side. The rapid development of financial technology, referred to as fintech, has been discussed from various perspectives, including cost, convenience, and security, by stakeholders such as financial consumers, companies, and governments [1], [2].

In online finance, the first objective for secure financial transactions is trusted credit scoring of financial consumers. Thus, the scope of data security for secure online financial transactions transcends digital identity verification to include the collection of details of financial transactions such as loans, deposits, taxes, debts, and overdue records for the purpose of personal credit scoring. Credit evaluation data for credit scoring purposes are no longer inseparable from personal information. In other words, the strength of the security required to protect these data is the same as that required to protect personal information [3]–[5]. Therefore, the technology for identifying individuals in online financial transactions has improved. Various biometric authentication techniques, such as fingerprint imaging, iris scanning, and facial recognition, can achieve identity authentication efficiently and securely. However, existing identity storage techniques remain centralized. This continues to pose a security risk and has a negative impact on efficiency.

A number of studies have investigated decentralized self-sovereign identity (SSI) using blockchain technologies, and proposed solutions such as ShoCard [6], uPort based on Ethereum [7], and Sovrin based on Hyperledger Indy [8]. Decentralized identifiers (DIDs) and SSI technologies that employ a distributed ledger store authenticated identifiers in a decentralized registry and use a blockchain to ensure identity immutability and easy retrieval. In addition, by controlling the verifiable credential (VC) directly through the terminal, the consumer can exercise strong rights toward personal identity information [9], [10]. In the financial industry, SSI is currently in its infancy and is being studied as a means to prove the identity of financial consumers. The adoption of SSI by the financial industry to verify identity by using DIDs would also require the provision of consumer credit evaluation data corresponding to the personal information in the form of a VC. In this study, the following various requirements are analyzed with the aim of providing credit evaluation data as a VC [11], [12]. The relationship between these requirements is shown in Fig. 1.

1. Decentralization: The proposed credit scoring data collection model should consider the decentralization of the identity verification technique. Moreover, it must be securely shared in the form of a VC in the SSI environment.
2. Security: When collecting credit scoring data, resistance to various threat models, such as sniffing, leakage, masquerading, and alteration, is necessary.
3. Opt-In: Sharing of all credit scoring data must take place with the knowledge of the data subject. Data sharing should not occur without the direct consent of the data subject [13].
4. Limited revocation right: Credit scoring data contain information originating from individual financial consumers. Data subjects must be able to exercise reasonable sovereignty over their credit scoring data; however, they should not abuse this right, for example, by deleting data perceived to adversely affect their credit scoring.
5. Trustworthiness: The collected credit scoring data must be given undeniable credibility by the parties in the financial transaction. The credit bureau must be able to immediately perform data reliability verification of the transaction parties when necessary.
6. Legality: Procedures for transacting and sharing credit scoring data to assess the potential for input to the financial industry should be justified on the basis of a common trusted law. In this study, we analyze the legal validity of the proposed model on the basis of the European Union’s General Data Protection Regulation (GDPR) criteria.

The VC is created in the SSI model, and the roles of the holder and issuer are clearly distinguished, i.e., the issuer is a participant who notarizes the validity of the VC, whereas the holder acts as a real information subject who can exercise all rights to the VC. However, these common characteristics of VCs are not suitable for processing credit scoring data. Note that credit scoring data are not completely exclusive to financial consumers, they also include data generated by interactions with financial companies [14]. If the sovereignty of credit scoring data was entirely controlled by the financial consumer, a malicious consumer could delete unfavorable financial transactions. This would not be consistent with the honest exercise of self-sovereignty as stipulated by the GDPR. In addition, it is difficult to ensure equal notarization of issuers because only the zero-knowledge proof (ZKP) of consumers is required in the process of verifying the verifiable presentation (VP) between the consumer and the credit bureau. These problems are factors that hinder security, limited revocation rights, trustworthiness, and legality among the proposed requirements.

In this study, we propose a VC generation model for credit scoring data based on zero knowledge - succinct non-interactive argument of knowledge (zk-SNARK) to solve the above-mentioned problems. Unlike the existing VC generation model that includes a simple proof, the proposed model generates credit scoring data including the issuer’s zk-SNARK proof at the time of VC generation. This provides the credit bureau with an opportunity to determine more reliably whether the issuer is notarized by using the Camenisch–Lysyanskaya (CL) signature technique for double signing, adopted by the Hyperledger Indy system, instead of the existing digital signature technique of VC proof [15], [16]. In addition, for VC revocation, a cross-VC revocation verification model including the issuer’s public key authentication is constructed. This is signed by the holder and the issuer as the joint data subject of the credit scoring data VC to prevent arbitrary deletion by a malicious holder. Furthermore, it provides a more reliable credit evaluation environment by sharing the VC that includes the double ZKP of the holder/issuer rather than the single ZKP verification of the holder.

The remainder of this paper is organized as follows. In Sect. 2, we analyze ZKP, SSI, and related work on credit scoring data collection schemes that use a decentralized system. In Sect. 3, we describe various assumptions and security threats to propose a new model. In Sect. 4, we propose a VC generation and revocation model including double ZKP verification and cross-revocation verification. In Sect. 5, we evaluate whether the proposed model satisfies the six above-mentioned requirements and compare it with related models. Finally, Sect. 6 concludes the paper.

2. Preliminaries

2.1 Zero-Knowledge Proof (ZKP)

Zero-knowledge proof was proposed by Goldwasser et al. in 1985 [17]. This technique is used to prove that a person owns information and does not require the information to be exposed. The information of which the possession is to be proved is referred to as a statement and the one-way secret evidence to confirm that the claim is true is referred to as a witness. The requirements for the operation of ZKP are as follows:

1. Completeness: If the statement is true, the honest verifier passes the verification.
2. Soundness: If the statement is false, any malicious
holder cannot pass the verification.

3. Zero knowledge: It does not expose any information other than the fact that the statement is true.

Non-interactive ZKP was proposed by Blum et al. in 1988 [18], and zk-SNARK, a popular variant of non-interactive ZKP, was developed by the ZCash working group in 2012. Unlike interactive ZKP, which requires the continuous online status of the verification parties for proofs, non-interactive ZKP provides an environment in which ZKP can be performed even if the holder is offline by exchanging only one proof. In addition, owing to the extremely small size of the proof, efficiency in terms of time cost can be achieved. Therefore, it satisfies succinct and non-interactive zero knowledge [19], [20]. An overview of zk-SNARK is as follows:

1. **KeyGen**(*x, y* → (*pk, vk*)): This step generates a key pair by a verifier. A key pair is generated by receiving an input consisting of the hash digest *x* and random sampling seed (RSS) *y* of witness *w*. Further, *pk* is generated as a proof key and *vk* is a verification key.

2. **Proof**(*pk, w, x* → *P*): This process generates a proof *P* by inputting the proof key, *w*, and *x* into the proof generation function which cannot be reversed.

3. **Verify**(*vk, x, P* → *T/F*): After receiving *P* from the holder, the verifier performs the verification of *P*. If the value is true, completeness is satisfied; if it is false, soundness is satisfied. In addition, zero-knowledge is satisfied by including *w* in *P* where inverse calculation is impossible.

The procedure for creating a challenge for zk-SNARK is as follows. We considered the cryptographically secure zk-SNARK (a detailed security analysis of the ZKP is beyond the scope of this paper). First, zero knowledge is achieved through homomorphic hiding (HH)[21]. HH in zk-SNARK satisfies the following three properties for the elements *x*, *y* and operations defined within the finite group *Zp*, defining the modular integer addition operation and the finite group *Zp* defining the modular decimal multiplication operation.

- When HH ciphertext *E*(*x*) exists for plaintext *x*, finding *x* through *E*(*x*) cannot be achieved within a reasonable polynomial time.
- Different function inputs result in different outputs.
- If *E*(*x*), *E*(*y*), and operation (·) are known for different plain texts *x* and *y*, *E*(*x* · *y*) can be computed.

In other words, the verifier transmits its challenge value to the holder. The holder performs verification by calculating *E*(*x* · *y*) from a given value. In the process of generating this challenge value, the HH-based blind evaluation of polynomials (BEoP) generation technique on finite field *Fp* is used. The procedure is as follows. First, the verifier transmits *Hidings* = *E*(*r0*), *E*(*r1*) · · · *E*(*rd*), which is the HH value of the private random number *r* to the holder. Next, the holder calculates the following *E*(*P(s)*) using *Hidings* and returns it to the verifier.

\[
P(s) = \sum_{i=0}^{d} a_i r^i
\]

\[
E(P(s)) = \sum_{i=0}^{d} E(a_i r^i) = \sum_{i=0}^{d} a_i E(r^i)
\]

The use of HH-BEoP enables the holder to calculate *E*(*P(s)*) without sharing the holder’s own private random number *r*. Moreover, the verifier can compute *E*(*P(s)*) without sharing the holder’s own polynomial *P(s)*.

In the process described above, a challenge value of mutual trust is created. Further, in the process of creating a challenge pair, which is a proof for zk-SNARK verification, a logic forcing the calculation is required to prevent malicious operation. This is implemented as the knowledge of coefficient (KC) test. In the KC test, the verifier creates an *α*-pair that cannot be reversed through the exponential function, which is an element of finite field *Fp* ∗. The holder receives this and selects a random element *δ* on *Fp* ∗ shared with the verifer to create a new *α*-pair. Here, a third party intervenes to verify the *α*-pair generated by the two entities. This third party is an extractor that is valid only within the KC test on an elliptic curve. The extractor extracts *δ* selected by the holder and securely delivers it to the verifier, providing public confidence of the zk-SNARK proof value.

### 2.2 SSI Based on Hyperledger Indy

SSI architecture based on the Hyperledger Indy project, such as Sovrin, supports ZKP operation in VP exchanges. In other words, combining the VCs the holder intends to prove, creating a VP, and performing ZKP for it, avoids direct exposure of the VC and claims that the holder possesses the VC. Unlike the existing SSI architecture using a general signature model, SSI using ZKP can guarantee security while exchanging lightweight proofs. In this section, we provide an overview of the VC structure including the zk-SNARK proof in the process of SSI architecture based on Hyperledger Indy [8].

| Table 1 | Symbols of SSI architecture |
|---------|----------------------------|
| **Symbol** | **Detail** |
| **Holder** | VC information subject |
| **Issuer** | VC issuance guarantor |
| **Verifier** | VC verification subject |
| **VCt** | x’s Verifiable Credential, Group of identity data |
| **VPt** | x’s Verifiable Presentation, Group of identity data for verification |
| **schema** | Information required when the VC is issued |
| **FKt** | x’s Public key for DID verification |
| **PKt** | x’s Public key for DID revocation |
| **vPrime** | VC issuance prime number |
| **vrPrime** | VC revocation prime number |
| **Link secret** | Information indicating the holder of the VC |
| **Accumulator** | VC revocation information registry |
| **Tails File** | VC revocation management list file |
to the analysis, it is assumed that in the Hyperledger Indy network environment, each entity (Holder/Issuer/Verifier) can write a DID document to the Indy node and share DIDs by using a transaction known as NYM. In addition, it is assumed that the holder has a secure mobile wallet for registering the DID private key and that the issuer has previously registered the VC schema and credential definition \((PK, PK_r)\) in the Indy node. Finally, it is assumed that each transaction is written without errors to four distributed ledgers (i.e., the Domain, Config, Pool, and Audit Ledgers) included in the blockchain Indy node.

When a request to create a VC is made by the holder, the issuer registers the revocation information of the VC in the blockchain and in the public registry owned by the issuer. Subsequently, the issuer modifies the accumulator containing the VC revocation information transaction as proof of VC revocation. The Tails File is a white list that records the valid VC’s ID and is stored in the publisher’s public repository. The holder and issuer can use this white list to inquire whether the VC is discarded [22].

Algorithm 1 represents the holder’s procedure for issuing the VC, and Algorithm 2 represents the issuer’s procedure. In Algorithm 1, the issuer issues the VC by creating a secret link for ZKP for ownership. To create the VC, the holder searches the schema and credential definition in the Indy node to obtain a public key \(PK, PK_r\) for VC creation and revocation. Further, random samples \(vPrime\) and \(vrPrime\) are generated by using a cryptographically secure pseudo-random number generator. The holder generates a Blinded Secret through an operation that takes Link secret, \(PK, PK_r, vPrime\), and \(vrPrime\) as inputs and sends it to the issuer. In Algorithm 2, the issuer writes the VC by entering the information of the holder who requested the issuance of the VC into the schema and then updates the Tails File for revocation management. Subsequently, the pre-VC revocation proof is created with Blinded Secret and Accumulator, \(PK, PK_r\) as inputs, and Indy node and Accumulator are updated. The issuer returns the pre-VC revocation proof to the holder. The holder can extract the VC revocation proof through the pre-VC revocation proof; thus, the holder can query the issuer’s Tails File for VC Proof. Therefore, the holder has the right to erase the VC directly by proving its validity or requesting a Tails File update.

2.3 Related Work

In 2019, Jain et al. proposed a credit scoring model that could be realized on a blockchain-based distributed network [23]. The model forms a blockchain network for credit evaluation and accepts as input details such as the SSN, driver’s license number, phone number, permanent address, and email address of users who participate in the network. The data are grouped and recorded as a single transaction, and the miner checks the validity of the transaction while calculating the temporary value of the block and adding it to the blockchain. Ten parameters are selected to calculate the credit score, and the FICO AI credit scoring estimation methodology is applied to each parameter. When individuals start taking loans, the method starts loading their credit history into the blockchain ledger. Based on the height of the block, the individual can accumulate credit to obtain large loans.

In 2020, Patel et al. proposed KiTRi, a blockchain-based credit scoring recommendation scheme for the financial industry [24]. KiTRi is a direct smart lending operation between prospective borrowers (PB) and prospective lenders (PL) to eliminate credit rating agencies, and provides a tuple that contains deep-learning-based credit recommendations based on the credit scoring information on a public blockchain. KiTRi stores the PB’s past transactions, current assets, and liabilities as sequence data on the public blockchain, and these data are credit-weighted by tuple deep-learning algorithms. In addition, the deep-learning algorithm creates a smart contract that recommends a PL with assets who could fulfill the required loan amount specified by the PB in the loan transaction.

In 2020, Zhu proposed a blockchain-based intelligent method, which reports the credit score and provides identity authentication [25]. This method, which provides a secure and open credit reporting system for credit data that has been established via a blockchain, can guarantee data transparency and immunity. Specifically, the identity is verified through biometric multidimensional authentication parameter management. It provides multidimensional credit data collection, distributed ledger-based storage, smart contracts with automatic credit score reporting, and blockchain-based credit white and black lists. Thus, the credit information is open to all financial consumers. This openness makes it easier to focus to consumers who engage in unusual financial transactions, and to implement the necessary risk control.

3. Assumption and Threat Model

In this section, we present an overview of participants and
parameters as well as the related assumptions. Furthermore, several threat models are set up to evaluate the reliability and security of the proposed scheme. We consider an example scenario in which a loan transaction between a consumer and a company is executed. The loan history (credit evaluation data) is provided to a credit bureau.

The proposed model considers creating a link secret that includes both the holder’s zk-SNARK proof and the issuer’s zk-SNARK proof for VC creation as well as the VC revocation signatures on both sides. Unlike the existing SSI, the proposed model makes no distinction between the issuer and the verifier. In other words, each entity except the holder must be able to perform the roles of both the issuer and the verifier. In addition, the model is designed as an example of a secure transaction involving credit scoring data VCs. Therefore, three entities are assumed as participants: a financial consumer (C), a financial company (FC), and a credit bureau (CB). The two VCs used in the proposed model are classified as the previous credit score (CS) of C and credit evaluation data (CE) to calculate the new CS. A double zk-SNARK method is employed to issue the proof of creation by including each CL signature to prove that the CE are managed dependently by the C and FC. From the SSI point of view, the FC is both an issuer and a co-holder with C for the creation of VC. Therefore, the CB becomes the verifier of VC generation. At the same time, to prevent the CE from being dishonestly revoked by the co-holders (C or FC), the model generates a revocation proof including the public key of the CB. Therefore, C and CB are co-holders for the VC revocation proof, and FC is the verifier. As a result, the VC life cycle is classified into matters relating to generation and revocation: generation includes joint certification of C and FC and revocation includes joint certification of C and CB. Because C intervenes throughout the VC life cycle, the corresponding CE VC must be handled as C’s personal information. Further, it notarizes that VC was generated by interaction between C and FC. In addition, to revoke VC, cross-revocation verification of C and CB is required, which prevents malicious single revocation of the CE. Overall, this guarantees that C is the subject of the limited self-sovereignty of the CE VC.

Each entity’s DID document is managed by the Indy node, distributed ledgers that all entities can trust. Each entity can check whether the entity creating the current transaction is a legitimate participant by using the DID Auth process at any time. An FC with CE VC issuance rights registered the loan VC schema and credential definition with VP generation requirements in the blockchain storage in advance. Further, it is assumed that a CB with CE VC cross-revocation right has registered a credential definition that defines a revocation proof public key, random number, etc., to prove the revocation of CE generated by C and FC. Moreover, in the process of issuing VC, it is assumed that the zk-SNARK method of individual non-interactive ZKP described in Sect. 2.1, which is a joint proof, is adopted, and that each entity performs CL signature. The VP revocation verification key pair is generated by the CL signature key generation algorithm. Finally, it is assumed that the accumulator, an update factor for VC revocation verification, is recorded in the Indy node, and that the private keys of C and CB for the revocation verification are securely stored in each electronic wallet. The proposed model is outlined in Fig. 2.

In this study, we consider three threat models to evaluate the security of the proposed model.

- Threat 1: Sniffing may occur in the process of VP’s zk-SNARK [26]. This may occur because of the SSI data exchange method, which does not use hidden channels or encryption techniques. This poses a threat in that a malicious attacker could access the channel to sniff the verification process between the holder and the verifier.
- Threat 2: Man-In-The-Middle (MITM) attacks could occur. In the case of Algorithm 2 described in Sect. 2.2, VC generation information such as BS and CL signature transmitted from the holder to the verifier can be stolen by an attacker who could create malicious ZKP signatures. The verifier may verify the malicious ZKP signature of the attacking intermediary rather than the normal ZKP signature of the holder, and may issue a VC that includes the attacker’s ZKP signature. This is not a valid signature owing to the self-sovereign nature of SSI; however, it could cause confusion in the system for malicious purposes.
- Threat 3: Insider vulnerability. For example, an employee who manages the loan system of a financial company could execute a loan to themselves and at the same time directly approve it, delete disadvantageous loan history, or execute abnormal loans through the revocation right.

4. System Architecture

4.1 Occurrence of Financial Transactions

In the process of financial transactions, we focus on two VCs as examples of special-purpose VCs: C’s previous credit score CS as viewed by FC for financial transactions and credit evaluation data CE provided to CB to calculate C’s new credit score as evidence of financial transactions.
between C and FC. The financial transaction is assumed to be a loan application (L) from C to FC.

Initially, C requests a loan application transaction L from FC. The L includes the DID claiming who C is, and the amount of the loan that C wishes to apply for. The FC that has received the L performs two steps before approving the loan request. First, the DID Auth process is used to determine whether C indicated by the DID is the same person as C who actually requested L. FC analyzes the DID method and method-specific identifier (MSI) of the DID received from C and can query the Indy node for the correct DID document. FC, which has acquired the DID document, which is a type of public key, encrypts a unique random value that has not been disclosed to the outside. Then, it forwards this to C and requests it to be decrypted with the private key held in C’s electronic wallet. The return of an appropriate value during the DIDAuth process provides verification that C is the owner of the DID document. Second, FC verifies C’s credit score CS. This score, which is calculated by CB on the basis of previous financial transactions, is issued to C in the form of VC, and FC requests it in the form of VP. C combines their own CS VC with the VP required by the schema and the credential definition of FC recorded in the Indy node and creates a zk-SNARK proof by adding a non-interactive ZKP signature. In this process, the statement is composed of CS and the witness is composed of the wk generated by C. If C’s justification is proved through the above-mentioned procedure, FC approves a loan to C according to its policy. The financial transaction does not need to take place on the basis of SSI. This is similar to the process that is used to approve current loan applications initiated from mobile devices or online. The process is shown in Algorithm 3.

4.2 Generation of CE VC

Usually, when a financial transaction involving a loan for C takes place, C and FC automatically proceed to issue a transaction certificate for the transaction. The reason for enforcing this mandatory step is to prevent C or FC from deliberately omitting proof of the financial transaction to avoid adversely affecting their financial transaction history. C performs the process of creating CE VC to prove the history of L with FC. First, PKC, PKrC included in the credential definition of FC and PKrCB, vPrimeC included in the credential definition of CB are acquired. In addition, random numbers vPrimec and vPrimeC are generated to ensure the uniqueness of Link Secret and VC generation and to verify the revocation. The creation of the Link Secret proceeds with the CL signature to prove C’s strong ZKP VC ownership. The proposed model creates a double ZKP between C and FC by using the CL signature as follows [27]:

- **KeyGen**: Choose two random prime numbers (p, q) and compute n = pq. Then, random k + 2 quadratic residues: (R1, ..., Rk, S, Z) are selected. This forms a key pair with the previously selected (p, q), i.e., a key pair with (p, q) as the private key and (n, R1, ..., Rk, S, Z) as the public key is created.
- **SigGen**: After selecting random k values m1, ..., mk, we compute the Euler function \( \phi(n) = \phi(p)\phi(q) = (p - 1)(q - 1) \) for two prime numbers (p, q). Next, another sufficiently large prime number e and a large integer v are selected. Then, the inverse of \( e \mod n \) is calculated. This satisfies \( e \cdot e^{-1} \equiv 1 \mod n \). Finally, after calculating Eq. (3), the final CL signature (A, e, v) is output.

\[
A \equiv (A^e)^{e^{-1}} \equiv \left( \frac{z}{S^v \cdot \prod R_i} \right)^{e^{-1}} \pmod{n} \tag{3}
\]

As for CE VC in the proposed model, FC is the issuer and at the same time acts as C and the co-holder; hence, the two entities each select two different prime numbers and then perform the above KeyGen and SigGen processes to generate their own CL signatures CLc and CLFC. The LinkSecret, which represents the holder’s proper zero-knowledge ownership, is a zk-SNARK proof created on the basis of CLc, CLFC. C creates a BlindedSecret including LinkSecret, PKC, PKrC, vPrimeC, vPrimeC, and PKrCB and then sends it to FC. The FC receiving this transmission can generate a pre-VC revocation proof that inputs the BlindedSecret, Accumulator, and the generation and revocation public keys in the same manner as the existing VC issuance method. The difference with respect to the existing method is that it generates not only the PKrC used in the BlindedSecret but also the PKrCB of the CB as input for generating the revocation proof. Therefore, the FC must acquire the public key of CB’s revocation proof (PKrCB) on the Indy node blockchain during the VC issuance process. In this process, the proposed model generates a zk-SNARK proof including C’s and FC’s notarized CL signatures, and includes the public key of the CB’s revocation proof in the verification. The issued CE VC must be provided in the form of a VP according to the schema that CB has disclosed.
to the Indy node. This process is shown in Algorithm 4.

4.3 Double Verification of CE VC

CBs that receive CE VC in the form of a VP must be able to verify it. Verification of the CL signature is as follows [27], and zk-SNARK verification is accomplished with an existing method [15].

- **VeriSig**: The verifier who has received the CL public key \( m_1, \ldots, m_k \) and CL signature \((A, e, v)\) substitutes the values into the verification equation (Eq. (4)) to verify whether \( e \) and \( v \) exist within the predefined range \( z \) and whether \( e \) is a prime number.

\[
A^e \equiv \frac{z}{S^e \prod_i R_i^{m_i}} \pmod{n} \quad (4)
\]

The CB, which has verified the validity of the two signatures \( CL_C \) and \( CL_FC \) using double VeriSig, can trust that CE is the credit evaluation data generated as a result of the interaction between \( C \) and \( FC \). In other words, CE could be considered to be a form of VC that can be shared without restriction on the SSI network and issued as reliable and secure credit evaluation data through the notarization of \( C \) and \( FC \). In addition, each of the CL signatures achieves zero knowledge via CE VP including non-interactive ZKP. Simultaneously, CB is in possession of reliable CE and can use it for C’s credit scoring. The VC verification process is presented in Algorithm 5. The procedure followed by the system from VC generation to verification is shown in Fig. 3.

4.4 Cross-Revocation Verification

Figure 3 presents a procedure for the revocation of CE VC. In general, when an honest financial transaction occurs, the proof of execution for that financial transaction is not revoked in any case. Therefore, the aspect of VC revocation in the proposed model aims to provide resistance to abnormal VC revocation attempts rather than to offer a description of normal VC revocation.

CB reserves the right to collect C’s financial transaction history for C’s credit scoring. CB has the authority to collect CE, but C decides whether to provide it. In other words, the entity that provides its CE VC to participate in
the financial network is $C$, and if $C$ does not submit it, they are considered not to participate in the financial network. Therefore, the decision to share $CE$ can be regarded as an explicit agreement indicating that $C$ intends to participate in the network. In other words, owing to $CB$’s legitimate request for collection, the sharing of $CE$ on the SSI financial network serves to express an Opt-In to participate in the network. However, if $C$ was permitted to self-revoke the $CE$ VC, the reliability of the Opt-In would be lost. Therefore, a separate revocation verification operation is required to grant limited revocation rights to $CE$ VC. In the existing SSI scheme, the procedure for VC revocation is divided into two stages: request and response. Each of these stages is as follows [9]:

- **Revocation Request**: The issuer receives a VC revocation request from the holder. This request includes $Tails$ $File$, $w$, and $Accumulator$. $Accumulator$ is calculated by multiplying each factor of the issuer $Tails$ $File$. Thus, $Accumulator$ guarantees the integrity of $Tails$ $File$ and is used to query Indy node to determine whether the requested VC has actually been revoked.
- **Revocation Response**: The issuer who receives the VC revocation request from the holder obtains $PK_r_c$ and the factors included in the VC to be revoked from the blockchain storage. Further, the asymmetric key operation by the holder with the private key of $PK_r_c$ verifies that the sovereign of the VC is the holder. Upon completion of the verification, the VC is revoked by searching for and deleting the VC recorded in $Tails$ $File$. As $Tails$ $File$ is a type of white list of valid VCs, VCs deleted from $Tails$ $File$ are no longer valid in the blockchain. After the VC is revoked, the $Accumulator$ is recalculated using the factors of the updated $Tails$ $File$, and it responds to the holder. As a result, when VC revocation is completed, the issuer updates the revocation registry recorded in the Indy node, and the holder can verify the revocation by comparing the $Accumulator$ received with the revocation registry of the Indy node.

The existing method of revocation as described above consists of only two of the three SSI participants. As in the threat model assumed in Sect. 3, it is vulnerable to insider threats if $C$ and $FC$ are the same, such as loan approvers allocating loans to themselves. In other words, it is not desirable in that it is possible to delete loans executed by oneself in an environment where revocation verification is performed only by interaction between $C$ and $FC$. Therefore, the proposed model includes the revocation public key of $CB$ in the revocation verification process, and it is designed to participate in the VC revocation process as a notary. The $CE$ VC revocation procedure in the proposed model is as follows. The procedure of cross-revocation verification is shown in Fig. 4.

- **Revocation Request**: The $FC$ receives a $CE$ VC revocation request from $C$. This revocation request includes $Tails$ $File$, $w$, and $Accumulator$. $Accumulator$ is cal-
culated by multiplying each factor of the issuer Tails File. Thus, Accumulator guarantees the integrity of Tails File and is used to query Indy node as to whether the requested VC has actually been revoked.

- **Revocation Response:** The FC who receives the VC revocation request from C obtains PKrC, and the factors included in the VC to be revoked from the blockchain storage. Further, PKrCB, the revocation notary of CB, is obtained through the credential definition of CB previously recorded in the Indy node. To revoke the VC, PKrC and PKrCB are verified with each of C’s and CB’s private keys. In other words, for VC revocation requested by C, the revocation of C and CB is required. In the case of normal CE VC, the public key verification may be rejected during this process because CB, the revoked notary, would not agree to this. In the case of normal CE VC, the public key verification can be rejected during this process because CB with the revocation notarization authority would not agree. FC can no longer proceed with the CE VC revocation procedure when public key verification is rejected by CB, and responds to C with a revocation reject.

5. **Comparison and Analysis**

In this section, we analyze the proposed model by comparing it with existing models. Using the proposed model, we analyze scenarios in which the requirements of various special-purpose VCs are complied with that could not be achieved by the existing VC structures. In addition, related models and the proposed model are compared and analyzed from the perspective of the six requirements stated in Sect. 1.

5.1 **Analysis of the Proposed Model**

The proposed model contains two major modifications compared with the existing SSI network. The first is the use of double zk-SNARK proof. In a normal network, a non-interactive ZKP is used, which does not require each entity to be online. In addition, the role of the issuer, who may not be involved in the creation of the VC, was adjusted such that the issuer is adopted as a new participant in VC verification. In the existing SSI model, the method to prove the fact of VC issuance involved a query using the issuer’s DIDs. However, this is comparable to asking a trusted third party for VC. If such a fact was to be leaked, it could give rise to an MITM attack, as suggested in Sect. 3. Thus, third-party vulnerability could cause confusion on the network when a malicious attacker intercepts the issuance proof. In the proposed scheme, the issuer participates with the notary concept of VC rather than a trusted third party. In the process of proving VC, the issuer has the same effect as the holder, which solves the problem of ownership conflict and vulnerabilities such as leakage and sniffing by introducing a CL signature that enables the verification of zero-knowledge ownership.

Second, the proposed model involves the verifier in the VC revocation process. Unlike before, the verifier acts as a holder with a real right to revoke VC. As suggested in Sect. 3, if the holder and issuer are the same, the system is exposed to serious insider vulnerabilities. To prevent this, the VC revocation is conducted using a multilateral transaction involving the holder/issuer verifier rather than only the holder/issuer. In the proposed model, we realized this by adopting a verification method that utilizes the verifier’s VC revocation public key recorded in the blockchain. In addition to the financial industry, which is the example considered in this study, the proposed model has sufficient flexibility to enable it to be applied to any SSI system that can maliciously revoke disadvantageous VCs.

As a result, in the VC generation process, the holder and issuer are co-holders, and in the VC revocation process, the holder and verifier are co-holders. This is a flexible grafting of different entities that have separate roles on the SSI network in the VC life cycle, and it was designed to allow all participating entities to trust this process without establishing a separate trust network. In addition, the notarization process restricted the right to revoke by distributing signature keys to other entities for special-purpose VCs, which should prevent the holder from exercising the right to revoke indiscriminately. Instead, a key holder can intervene in all other processes, thus achieving limited self-sovereign behavior and requirements for explicit consent. The double zk-SNARK verification and cross-revocation verification of the proposed model are illustrated in Fig. 5. Table 2 summarizes the solution of the proposed model to the threat model assumed in Sect. 3.
Combining the above-mentioned concepts, the proposed model satisfies the requirements of the special type of VC proposed in Sect. 1 as follows.

1. Decentralization: Hyperledger Indy SSI architecture operates on the basis of the Indy node, an integrated public blockchain node composed of four distributed ledgers. In the Indy node, the DID document is changed from time to time, and it is composed of a Merkle–Patricia tree structure to immediately reflect this. In addition, it follows the plenum technique as the consensus algorithm of the decentralized blockchain [22].

2. Security: The process of sharing credit scoring data is performed through non-interactive ZKP. Owing to prior agreement of the ZKP, the proposed model can deliberately undermine the reliability of data for third parties that do not form part of the verification party. Therefore, even if leakage was to occur, security is guaranteed because an attacker cannot trust the information in a probability distribution.

3. Opt-In: The only participant involved in the entire VC life cycle is the holder. In other words, although the issuer and verifier are involved in the generation and revocation of VC, this cannot occur without the consent of the holder. Consequently, it is technically impossible to create a VC of the holder of which the holder is unaware.

4. Limited revocation right: The holder is a subject who can exercise the right to revoke a VC. However, in the proposed model, malicious revocation rights are prevented by adding revoked notaries [28]. The model is designed to partially limit self-sovereignty; however, this limit does not affect honest behavior and is only intended to prevent dishonest behavior.

5. Trustworthiness: As the VC generation process is notarized by the issuer and the revocation process is notarized by the verifier, the proposed model ensures trustworthiness. In addition, as all notaries include CL signatures, the integrity of the VC can be verified using ZKP.

6. Legality: The GDPR recognizes that, in the free data marketplace, trusted de-identification data are shared. The proposed model corresponds to a type of irreversible generalized de-identification data based on the difficulty of factorization in CL signatures. Therefore, the VC sharing model in this study satisfies the non-identifying data criteria of the GDPR.

### 5.2 Comparative Analysis of Related Work

Jain et al. proposed a model that incorporates blockchain technology into credit scoring. All digital identity information is stored on the blockchain network, and identity verification is performed with the aid of mining. This has the effect of achieving decentralization and security with the blockchain in credit evaluation, and it resembles the traditional blockchain technique that lends credibility to identity verification owing to miners. However, the identity information recorded in the public blockchain can be viewed without the Opt-In of the subject, and this affects the self-sovereign operation. Moreover, it is difficult to achieve the legal purpose of the GDPR, which must be shared as de-identification data.

Patel et al. proposed a deep-learning-based algorithm that imposes a credit weighting to allow transparent and reliable credit scoring using the block height. In addition, as the recommendation algorithm is executed with the consent of the PB, it guarantees the Opt-In. However, owing to the nature of the blockchain, the stored information cannot be deleted directly, which makes it difficult to guarantee the right to erasure by the GDPR.

Zhu achieved decentralization and a high level of security by proposing a smart contract method based on a distributed ledger in the credit reporting system. In addition, dynamic identity security authentication and intelligent risk control were realized. However, it is difficult to guarantee the Opt-In and right to erasure because the credit report is shared as a collective white list and black list. In addition, the user’s credit information is not de-identified and is shared as a white list, which does not comply with the GDPR standard.

Table 3 presents the results of the comparative analysis of the proposed model and related models.

---

| Decentralize | Jain et al. [23] | Patel et al. [24] | Zhu [25] | Proposed Model |
|-------------|-----------------|-----------------|---------|----------------|
| Security    | Ordinary blockchain network | Ordinary blockchain network | User’s biometric and behavioral features | zk-SNARK proof (prior agreement) |
| Opt-In      | X | Prospective borrowers Opt-In | X | Holder-based VC generation scheme |
| Limited revocation right | X | X | X | Public key based cross revocation proof |
| Trustworthiness | Mining transactions | Deep-learning algorithm | Multidimensional credit data | Double CL signature based zk-SNARK proof |
| Legality | X | X | X | Self-Sovereignty of SSI architecture |

---

Table 3 Comparison between the proposed model and related models
6. Conclusion

In this paper, we proposed a method for generating and transacting VC data that requires verification of the ownership of multiple holders in the SSI scheme. We discussed the requirements and threat models for special-purpose VCs created by multiple stakeholders. In addition, we proposed a special VC generation and revocation model that supports double zk-SNARK verification and cross-revocation verification based on the requirements and threat model. As an example, we assumed a VC that recognizes ownership attributable to multi-holders by incorporating loan history data into credit scoring data. This was designed to prevent indiscriminate deletion by malicious holders and allowed verification of the VC holder, who acts as a co-holder and issuer from the viewpoint of a verifier.

In the proposed model, we contribute to the application of data that require various forms of joint ownership verification, such as joint land ownership, distributed equity sharing, and ring signature data, to the SSI architecture. Moreover, as our aim was to develop a model in which the role of each entity flexibly transitions according to the attributes of the VC to be verified without distinguishing between the issuer and the verifier, our work contributes to the construction of a complex and massive SSI network. As a result, the proposed model can be expected to continuously revitalize academic discussions such that the SSI network architecture, which remains in its infancy, can be considered for mainframe governance.

However, the proposed model is based on double zk-SNARK verifications. This presents a shortcoming in that the ZKP verification time linearly increases compared to the existing VC, which is a factor that degrades the performance of the entire process. To a certain extent, this problem can be overcome by using a technique in which the verifier batch verifies the double zk-SNARK during special-purpose VC verification. This requires a verifier to be able to verify both ZKPs in batches, and the resulting increase in the computational intensity and resource consumption must be within an acceptable range for a general lightweight device. In this regard, it may consider the application of a lightweight zk-SNARK technique such as [29], [30]. In addition, in the current situation in which Hyperledger Indy-based implementation cases are scarce, we plan to implement double zk-SNARK based special-purpose VCs within a reasonable amount of computation for lightweight devices such as smart devices.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. 2019R1I1A3A01060652), and a part of the project titled ‘Future fisheries food research center,’ funded by the Ministry of Oceans and Fisheries, Korea.

References

[1] D. Roman and G. Stefano, “Towards a reference architecture for trusted data marketplaces: The credit scoring perspective,” Proc. 2nd International Conf. on Open and Big Data, Vienna, The Austria, pp.95–101, Aug. 2016. DOI:10.1109/OBD.2016.21
[2] K. Gai, M. Qiu, and X. Sun, “A survey on FinTech,” Journal of Network and Computer Applications, vol.103, no.1, pp.262–273, 2018. DOI:10.1016/j.jnca.2017.10.011
[3] S. Wachter, “Normative challenges of identification in the Internet of Things: Privacy, profiling, discrimination, and the GDPR,” Computer Law & Security Review, vol.34, no.3, pp.436–449, 2018. DOI:10.1016/j.clsr.2018.02.002
[4] C. Tankard, “What the GDPR means for businesses,” Network Security, vol.2016, no.6, pp.5–8, 2016. DOI:10.1016/S1353-4858(16)30056-3
[5] G. Kondova and J. Erbguth, “Self-sovereign identity on public blockchains and the GDPR,” Proc. 35th Annual ACM Symposium on Applied Computing, New York, The USA, pp.342–345, March 2020. DOI:10.1145/3341105.3374066
[6] Y. Liu, D. He, M.S. Obaidat, N. Kumar, M.K. Khan, and K.-K.R. Choo, “Blockchain-based identity management systems: A review,” Journal of Network and Computer Applications, vol.166, no.102731, pp.1–11, 2020. DOI:10.1016/j.jnca.2020.102731
[7] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena, “Uport: A platform for self-sovereign identity draft version,” Blockchain Lab, https://whitepaper.uport.me/uPort_whitepaper_DRAG20170221.pdf, accessed Jan. 28 2021.
[8] P. Windley and D. Reed, “Sovrin™: A protocol and token for self-sovereign identity and decentralized trust,” Sovrin Foundation, https://sovrin.org/wp-content/uploads/Sovrin-Protocol-and-Token-White-Paper.pdf, accessed Dec. 27 2020.
[9] D. Reed, M. Sporny, and M. Sabadello, “Decentralized identifiers (DIDs) v1.0,” W3C Working Draft, https://www.w3.org/TR/did-core/, accessed Jan. 11 2021.
[10] M. Sporny, G. Noble, D. Longley, D.C. Burnett, and B. Zundel, “Verifiable credentials data model 1.0,” W3C Editor’s Draft, https://w3c.github.io/vc-data-model/#claims, accessed Jan. 11 2021.
[11] C. Kim, “The laws of identity,” Identity Blog, https://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf, accessed Nov. 9 2020.
[12] A. Mühle, A. Grüner, T. Gayvoronskaya, and C. Meinl, “A survey on essential components of a self-sovereign identity,” Computer Science Review, vol.30, pp.80–86, 2018. DOI:10.1016/j.cosrev.2018.10.002
[13] Q. Stokkink and J. Pouwelse, “Deployment of a Blockchain-Based Self-Sovereign Identity,” Proc. 2018 IEEE International 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), IEEE, Halifax, NS, The Canada, no.18735154, pp.1336–1342, July 2018. DOI:10.1109/Cybermatics_2018.2018.00230
[14] M.A. Azad, S. Bag, and F. Hao, “PrivBox: Verifiable decentralized reputation system for online marketplaces,” Future Generation Computer Systems, vol.89, pp.44–57, 2018. DOI:10.1016/j.future.2018.05.069
[15] D. Schröder, “How to Aggregate the CL Signature Scheme,” Proc. Computer Security-ESORICS 2011, Leuven, The Belgium, pp.298–314, 2011. DOI:10.1007/978-3-642-23822-2_17
[16] A. Lysyanskaya, R.L. Rivest, A. Sahai, and S. Wolf, “Pseudonym Systems,” Proc. International Workshop on Selected Areas in Cryptography, Ontario, The Canada, Aug. 1999. DOI:10.1007/3-540-46513-8_14
[17] S. Goldwasser, S. Micali, and C. Rackoff, “The Knowledge Complexity of Interactive Proof Systems,” SIAM Journal on computing, vol.18, no.1, pp.186–208, 1989. DOI:10.1137/0218012
Kang Woo Cho received his B.E. degree in Dept. of Information Communication Engineering from Pukyong National University, Republic of Korea in 2020. He is currently a master course student in Dept. of Information Security from Pukyong National University. His research interests are related with blockchain, identity authentication, and privacy security.

Byeong-Gyu Jeong received his B.E. degree in Dept. of IT Convergence and Engineering from Pukyong University, Republic of Korea in 2018. He is currently a master course student of Pukyong National University. His research interests are related with blockchain, digital forensics, and information security.

Sang Uk Shin received his M.S. and Ph.D. degrees from Pukyong National University, Busan, Korea in 1997 and 2000, respectively. He worked as a senior researcher in Electronics and Tele-communications Research Institute, Daejeon Korea from 2000 to 2003. He is currently a professor in Department of IT Convergence and Application Engineering, Pukyong National University. His research interests include digital forensics, e-Discovery, cryptographic protocol, mobile and wireless network security and multimedia content security.