Zero-Knowledge Identification Scheme Using LDPC Codes

Haruka ITO†, Nonmember, Masanori HIROTOMO††, Youji FUKUTA††, Members, Masami MOHRI†††, and Yoshiaki SHIRAISHI††††, Senior Members

SUMMARY  Recently, IoT compatible products have been popular, and various kinds of things are IoT compliant products. In these devices, cryptosystems and authentication are not treated properly, and security measures for IoT devices are not sufficient. Requirements of authentication for IoT devices are power saving and one-to-many communication. In this paper, we propose a zero-knowledge identification scheme using LDPC codes. In the proposed scheme, the zero-knowledge identification scheme that relies on the binary syndrome decoding problem is improved and the computational cost of identification is reduced by using the sparse parity-check matrix of the LDPC codes. In addition, the security level, computational cost and safety of the proposed scheme are discussed in detail.

key words: zero-knowledge proof, identification, LDPC codes, syndrome decoding problem

1. Introduction

IoT (Internet of Things) is to embed the communication functions in devices and make automatic recognition and automatic control possible by connecting to the Internet. While various devices become IoT compliant products, encrypted communication and appropriate identification are still not used in many devices [1], and security measures for the IoT devices are not sufficient. The IoT devices connect to various devices. It also has the feature of power saving and small computing capacity. Currently, public key cryptosystems that rely on the discrete logarithm problem and the prime factorization problem applied to the identification techniques. However, since these are calculated over large finite fields, the computational cost is large and it is hard to apply these to IoT devices. Therefore, it is necessary to develop identification scheme with low computational cost for the IoT device.

Stern proposed an identification scheme that relies on the decoding problem of error correcting codes [2]. The calculation of the error correcting codes is calculation of vector and matrix. Therefore, there is a possibility that the computational cost can be reduced as compared with the above using the finite fields. In addition, Stern’s identification scheme is the zero-knowledge identification scheme. That is, regardless of whether or not the connection partner can be trusted in the situation of the IoT device, it can be safely authenticated.

Recently, LDPC codes have attracted attention as an error correcting codes with low computational cost of decoding algorithm and excellent decoding performance. In order to reduce the computational cost, we consider that LDPC codes are applied to Stern’s identification scheme. Meanwhile, McEliece cryptosystem [3] and Niederreiter cryptosystem [4] were proposed as public key cryptosystems that relies on the decryption of error correcting codes. McEliece cryptosystem relies on the nearest neighbor code problem and Niederreiter cryptosystem relies on the binary syndrome decoding problem. Cryptosystems in which these cryptosystem are improved by LDPC code [5], QC-LDPC code [6] and MDPC code [7] have been proposed. However, the computational cost is not small enough to be available for the IoT device. That is, they can not be simply applied to identification of the IoT device.

In this paper, we propose a zero-knowledge identification scheme using LDPC code in order to develop lightweight identification scheme for IoT devices. This scheme is a improvement of Stern’s identification scheme. Using the property of sparse parity-check matrix of the LDPC codes, the computational cost of identification scheme can be reduced. We examine undecodability of the proposed scheme, which is based on the syndrome decoding problem of LDPC codes, by iterative decoding algorithm. Furthermore, we evaluate the security level, computational cost of the proposed scheme in detail.

This paper is organized as follows. In Sect. 2, we describe the zero-knowledge identification scheme based on binary syndrome decoding problem and LDPC codes. In Sect. 3, we present the zero-knowledge identification scheme using LDPC codes. In Sect. 4, we present undecodability of the proposed scheme by iterative decoding algorithm. In Sect. 5, we show evaluation of identification scheme. In Sect. 6, we conclude this paper.

2. Preliminary

2.1 Zero-Knowledge Identification Scheme

Public key based identification schemes are very useful and
fundamental tools in many applications such as electronic funds transfer and online systems for preventing data access by invalid users. The identification schemes are typical applications of zero-knowledge (ZK) interactive proofs, and several practical ZK identification schemes have been proposed [8]. The ZK identification scheme satisfies three properties of the completeness, soundness and zero-knowledge. The security of the ZK identification scheme generally relies on hard problems on number theory, such as the discrete logarithm problem and the prime factorization problem. On the other hand, Stern has proposed a ZK identification scheme that relies on hard problems on coding theory [2]. That is the binary syndrome decoding (BSD) problem of error correcting codes. It is a 3-pass code-based identification scheme with a cheating probability of 2/3.

Recently, some researchers improved Stern’s scheme and proposed code-based ZK identification schemes. In [9], Gaborit et al. have improved Stern’s scheme by using double circulant matrix in order to reduce the size of secret and public keys. In [10], Véron has proposed an identification scheme based on the nearest neighbor code problem in order to reduce the communication cost. In [11], Cayrel et al. have proposed 5-pass identification scheme for which the success probability of a cheater is 1/2 by using q-ary syndrome decoding problem. It is lower communication cost than Stern’s scheme. In [12], Hu et al. have also proposed 3-pass identification scheme with cheater probability 2/3. It is superior to Cayrel et al.’s scheme. Furthermore, in [13], Aguilar et al. have proposed an identification scheme by applying double circulant matrix to Véron’s scheme in order to reduce the key size and the communication cost.

2.2 Binary Syndrome Decoding Problem

The security of a given ZK identification scheme relies on hardness of obtaining the secret key from the public key. Every public key identification scheme has to rely on a hard problem. In the case of coding theory, the main used is:

**Definition 1:** Binary syndrome decoding (BSD) problem

**Instance:** An \((n - k) \times n\) matrix \(H\) over \(\mathbb{F}_2\), a vector \(s \in \mathbb{F}_2^{n-k}\), and an integer \(t > 0\).

**Question:** Is there a vector \(x \in \mathbb{F}_2^n\) of weight \(\leq t\), such that \(Hx^T = s^T\)?

When a parity-check matrix \(H\) with error correction capability \(t\) and a syndrome \(s\) are given, it is a problem to find an error vector \(x\) that minimizes the weight. This problem is proven to be NP-complete [14]. However, if \(H\) has a specific structure, it can be an easy problem. For example, bounded distance decoding algorithms efficiently correct errors up to the distance \(t\) for algebraic codes such as Reed-Solomon codes. If such codes and \(t\) are used, it is easy to solve the problem.

2.3 Low-Density Parity-Check Codes

Low-density parity-check (LDPC) codes are linear block codes defined by sparse parity-check matrices [15]. Let \(C\) be a binary linear code of length \(n\) and dimension \(k\), and the parity-check matrix is denoted by

\[ H = [h_{ij}]_{1 \leq i \leq n, 1 \leq j \leq n} \]

Any codeword \(c = (c_1, c_2, \ldots, c_n) \in C\) satisfies

\[ Hc^T = 0 \] \hspace{1cm} (1)

The Hamming weight of \(c\) is denoted by \(W_H(c)\). This parity-check matrix is sparse so that the matrix \(H\) has fewer nonzero than zeros. If the weight of each row is \(d_r\) and the weight of each column is \(d_c\) in \(H\), the code \(C\) is called \((d_r, d_c)\)-regular LDPC code.

LDPC codes can be defined by bipartite graphs called Tanner graphs. The code \(C\) given by Eq. (1) can be represented by a bipartite graph \(G = (V \cup P, E)\) defined as follows. The vertex set consists of variable nodes \(V = \{v_1, v_2, \ldots, v_n\}\) and the parity-check nodes \(P = \{p_1, p_2, \ldots, p_{n-k}\}\). The set of \([v_i, p_j]\) is an edge if and only if \(h_{ij} = 1\). The set of edge sets is denoted by \(E\). LDPC codes can be decoded by iterative decoding algorithms which perform message passing over the Tanner graph. We describe hardness of the BSD problem using LDPC codes in Sect. 3.4 and Sect. 4.1.

3. ZK Identification Scheme Using LDPC Codes

In this section, we propose a ZK identification scheme using LDPC codes. This scheme is an improvement of Stern’s identification scheme [2]. In the proposed scheme, the sparse parity-check matrix of LDPC codes is applied to Stern’s identification scheme in order to reduce the computational cost of syndrome calculation.

3.1 Proposed Identification Scheme

We propose a ZK identification scheme that relies on the BSD problem using LDPC code. This scheme regards a three-move interaction between a prover and a verifier as one round. In the proposal scheme, the secret key and the public key are respectively generated and procedures are performed as follows.

**Secret Key**

- A vector \(x \in \mathbb{F}_2^n\) of weight \(w \leq t\)
- An \((n - k) \times n\) sparse parity-check matrix \(H_t\) of LDPC code over \(\mathbb{F}_2\) with the error correction capability \(t\)
- A random \((n - k) \times (n - k)\) nonsingular matrix \(S\) of row weight \(u\) and column weight \(u\) over \(\mathbb{F}_2\)
- A random \(n \times n\) permutation matrix \(P\) over \(\mathbb{F}_2\)

**Public Key**

- An \((n - k) \times n\) matrix \(H' = SH_tP\)

\(^1\)When all row weights and column weights are made constant like row weight \(u\) and column weight \(u\), the matrix \(S\) may not be regular. In that case, by increasing one of the row weights by 1, the matrix \(S\) is made regular.
A syndrome \(s (= xH^T) \in \mathbb{F}_2^k\)
- The weight \(w\) of vector \(x\)

**Step1. Initialization:** The prover \(P\) randomly chooses a vector \(y \in \mathbb{F}_2^k\) and a permutation \(\sigma\) on \([1, \ldots, n]\).

**Step2. Commitment:** The prover \(P\) sends the following commitments to the verifier \(V\):

\[
c_1 = h(\sigma, H^T y^T), \\
c_2 = h(\sigma(y)), \\
c_3 = h(\sigma(y + x)),
\]

where \(h()\) is a hash function and \(\sigma(y)\) is the image of \(y\) by the permutation \(\sigma\).

**Step3. Challenge:** The verifier \(V\) sends a random challenge \(b\) in \([0, 1, 2]\) to the prover \(P\).

**Step4. Response:** The prover \(P\) sends a response to the verifier as follows: If \(b = 0\), \(P\) reveals \((y, \sigma)\). If \(b = 1\), \(P\) reveals \((y + x, \sigma)\). If \(b = 2\), \(P\) reveals \((\sigma(y), \sigma(x))\).

**Step5. Verification:** The verifier \(V\) checks the response and commitments in accordance with \(b\). If \(b = 0\), \(V\) checks that \(c_1\) and \(c_2\), which were made in Step2, have honestly been computed as follows:

\[
c_1 = h(\sigma, H^T y^T), \\
c_2 = h(\sigma(y)).
\]

If \(b = 1\), \(V\) checks that \(c_1\) and \(c_3\) were correct as follows:

\[
c_1 = h(\sigma, H^T (y + x)^T + s^T), \\
c_3 = h(\sigma(y + x)).
\]

If \(b = 2\), \(V\) checks the weight property and \(c_2\) and \(c_3\) as follows:

\[
c_2 = h(\sigma(y)), \\
c_3 = h(\sigma(y) + \sigma(x)), \\
W_H(\sigma(x)) = w.
\]

The change from Stern’s scheme to the proposed scheme is that the three matrices \(H, S\) and \(P\) are generated as the secret key and the matrix \(H' = SH_P\) is used as the public key. Also, the syndrome calculation for the public key and Eqs. (2), (5) and (7) is executed by using Tanner graph. So, the computational cost of syndrome calculation in key generation, commitment, and verification is reduced.

Repeat Steps 1, 2, 3, 4 and 5 until the necessary security level is reached. For a given real number \(\varepsilon\), if the acceptance rate of verification is less than \(1 - \varepsilon\), identification fails; otherwise it is identification success. In this proposed scheme, the failure probability of verification at each iteration is \(2/3\). For instance to reach the weak and strong authentication probabilities of \(2^{-16}\) and \(2^{-32}\) of the norm ISO/IEC-9798-5, it needs 28 and 56 iterations respectively.

### 3.2 BSD Problem Using Sparse Parity-Check Matrix

The security of the proposed scheme relies on the BSD problem defined by sparse parity-check matrix, i.e., it is hard to find the secret key \(x\) from the public key \(H\) and \(s\). In this section, we explain how to apply the sparse parity-check matrix of LDPC codes to the BSD problem.

If we directly use the sparse parity-check matrix \(H_i\) for \(H\) in the BSD problem, the computational cost of finding a vector \(x\) satisfying \(H_i x^T = s^T\) is probably reduced by iterative decoding algorithms of LDPC codes. Thus, when the sparse parity-check matrix \(H_i\) is used as the public key of Stern’s identification scheme, it is easy to find the secret key \(x\) from the public key \(s\) and \(H_i\). On the other hand, when the parity-check matrix has short cycles, they suffer with the performance of iterative decoding algorithms and the exact error cannot be found by iterative decoding algorithms. Transforming the sparse parity-check matrix \(H_i\) into a matrix containing short cycles, it is hard to find the secret key from the public key by iterative decoding algorithm.

In the proposed scheme, the parity-check matrix is transformed as follows. An \((n-k) \times (n-k)\) nonsingular matrix \(S\) of row weight \(u\) and column weight \(u\) and an \(n \times n\) permutation matrix \(P\) are randomly generated respectively.

Let \(H' = SH_P\). The row weight and the column weight of \(H'\) are larger than that of \(H\). We use \(H'\) as the public key, and use \(H_i, S\) and \(P\) as the secret key. It implies that it is hard to obtain the secret key \(x\) satisfying \(H' x^T = s^T\) from the public key \(s\) and \(H'\). Furthermore, by multiplying \(H_i\) by \(S\) and \(P\), it makes impossible to find \(H_i\) from \(H'\).

**Example 1:** Consider a parity-check matrix of \((2,4)\)-regular LDPC code of length 10 and rate 1/2.

\[
H_i = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0
\end{bmatrix}.
\]

Randomly generate a nonsingular matrix \(S\) of size \(5 \times 5\) and a permutation matrix \(P\) of size \(10 \times 10\) as follows:

\[
S = \begin{bmatrix}
1 & 1 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1
\end{bmatrix},
\]

\[
P = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}.
\]

Calculating \(H' = SH_P\) yields the following matrix:
When the row weight of $S$ is $u$ and the row weight of $H_l$ is $d_r$, the row weight of $H'$ is expected to be $ud_r$. By increasing the weight of each row of $H'$, it is possible to create short cycles. For example, one in the $(2,1), (2,7), (4,1)$ and $(4,7)$ elements of $H'$ makes a cycle of length 4. $H'$ has a lot of short cycles. They decrease the performance of iterative decoding algorithms.

### 3.3 Syndrome Calculation Using Tanner Graph

The number of ones in the parity-check matrix $H_l$ of the LDPC codes is very small. Therefore, it is thought that $H'$ made by multiplying $H_l$ by an matrix $S$ of row weight $u$ also has the small number of 1s. Using this property, we calculate the syndrome using Tanner graph.

Let $J_i$ be a set of indices of variable nodes connecting the $i$th parity-check node $p_i$. The set is written by

$$J_i = \{ j \mid h_{ji} \neq 0, 1 \leq j \leq n\}$$

Then, the syndrome calculation is performed as follows:

$$s_i = \sum_{j \in J_i} x_j.$$

**Example 2**: Figure 1 shows a Tanner graph corresponding to $H'$ in Example 1. Consider the syndrome calculation $s = xH'\tau$ when the vector is

$$x = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0).$$

Assign $x$ to the variable nodes $v_j$ and pass it to all check nodes $p_i$ connected by $v_j$. Since $p_1$ is connected to $v_2, v_3, v_4, v_5, v_7, v_9$, $p_2$ is connected to $v_1, v_4, v_7, v_8, v_9, v_{10}$, and $p_3$ to $v_2, v_3, v_4, v_5, v_7, v_9$,

$$s_1 = x_2 + x_3 + x_4 + x_5 + x_7 + x_9 = 0,$$

$$s_2 = x_1 + x_4 + x_7 + x_8 + x_9 + x_{10} = 1,$$

$$s_3 = x_2 + x_3 + x_4 + x_5 + x_7 + x_9 = 0.$$

Similarly, the values of $s_4, \ldots, s_{n-k}$ are calculated respectively. Then, the syndrome $s = (s_1, s_2, \ldots, s_{n-k})$ is obtained as follows

$$s = (0, 1, 0, 0, 1).$$

Generally, in the syndromes calculation, $n(n-k)$ multiplications and $(n-1)(n-k)$ additions are required to obtain the product of an $(n-k) \times n$ matrix and an $n$-bit vector, the computational cost is $(2n-1)(n-k)$. On the other hand, in the syndrome calculation using the Tanner graph, only the position of nonzeros of the parity-check matrix (the nodes connected by the edge in the Tanner graph) is computed. Therefore, in the case of the parity-check matrix of the row weight $d_r$, only $(d_r - 1)(n-k)$ additions is required to the syndrome calculation.

### 3.4 Undecodability of the Proposed Scheme by Iterative Decoding Algorithms

In this section, we show that the BSD problem with $H' (= SH_lP)$ cannot be solved by iterative decoding algorithms. Solving the BSD problem is equivalent to the decoding problem that errors or codewords are found from received words. The computational cost of iterative decoding algorithms is small for LDPC codes. However, when the parity-check matrix contains small cycle, iterative decoding algorithms cannot find codewords unsuccessfully. Especially, cycles of length 4 (4-cycles) induce failures of iterative decoding algorithms. In the proposed scheme, multiplying $H_l$ by $S$ generates small cycles in $H'$. Additionally, the computational cost of syndrome calculation is small when nonzeros in $H'$ is few. Therefore, we show that $H'$ has 4-cycles when $H_l$ is multiplied by $S$ of row weight 2 and column weight 2 in the case of $u = 2$. We describe the following theorems and proofs with the assumption of $H' = SH_l$. Also, we assume that $H_l$ is a parity-check matrix of row weight $d_r$ and column weight $d_c$, because regular LDPC codes are often used.

Generally, the parity-check matrices are constructed without 4-cycles, because they reduce the performance of iterative decoding algorithms. First, we consider the case that $H_l$ has no 4-cycle.

**Theorem 1**: Let $H_l$ be a matrix of row weight $d_r \geq 3$ and column weight $d_c \geq 2$. If $S$ is a matrix of row weight 2 and column weight 2 and $H_l$ has no 4-cycle, $H' = SH_l$ has 4-cycle at each column.

**Proof**: Multiplying $H_l$ by $S$ of row weight 2, each row of $H'$ is obtained by XORing two rows of $H_l$. Assume the $k$th row $H'_k$ of $H'$ is obtained by XORing the $i$th row $h_i$ and the $j$th row $h_j$ of $H_l$. Since $H_l$ has no 4-cycle, nonzeros of $h_i$ cannot collide with more than two nonzeros of $h_j$. Thus, $H'_k$ contains more than $d_r - 1$ nonzeros of $h_i$. Furthermore, multiplying $H_l$ by $S$ of column weight 2, $h_i$ is also added to the other row $H'_j$. Similarly, $H'_j$ contains more than $d_c - 1$ nonzeros of $h_j$. Therefore, there are more than $d_r - 1$ nonzeros of $h_i$ at the same columns for two rows of $H'$, so
that this is 4-cycle.

Next, we prove that \( \mathbf{H}' \) contains 4-cycles at each column. It can be proved by showing that \( \mathbf{H}' \) contains more than two nonzero at each column. Multiply \( \mathbf{H}_l \) by \( \mathbf{S} \) of row weight 2, each column of \( \mathbf{H}' \) is obtained by XORing two columns of \( \mathbf{H}_l \). Then, each column of \( \mathbf{H}' \) contains \( 2d_r - 2a \) nonzeros, where \( a \) is the number of collision of XORing two rows of \( \mathbf{H}_l \). Hence, the weight of each row of \( \mathbf{H}' \) is even, so that the weight is not one. In addition, the weight of column of \( \mathbf{H}' \) is 0 if \( \mathbf{S} \) has the same vectors in two columns, but such the matrix \( \mathbf{S} \) is singular. Therefore, \( \mathbf{H}' \) contains more than two nonzeros at each column. Consequently, \( \mathbf{H}' \) has 4-cycle at each column. □

Next, we consider the case that \( \mathbf{H}_l \) has 4-cycles.

**Theorem 2:** Let \( \mathbf{H}_l \) be a matrix of row weight \( d_r \geq 3 \) and column weight \( d_c \geq 2 \). If \( \mathbf{S} \) is a matrix of row weight 2 and column weight 2, \( \mathbf{H}' = \mathbf{S}\mathbf{H}_l \) has 4-cycle at each column.

**Proof:** Similar to the proof of Theorem 1, we consider that the \( k \)-th row \( \mathbf{h}'_k \) of \( \mathbf{H}' \) is obtained by XORing the \( m \)-th row \( \mathbf{h}_m \) and the \( m \)-th row \( \mathbf{h}_m \) of \( \mathbf{H}_l \) when \( \mathbf{H}_l \) is multiplied by \( \mathbf{S} \) of row weight 2. We consider the case that \( \mathbf{H}_l \) contains 4-cycles, because we have proved that \( \mathbf{H}' \) has 4-cycles in the case of \( \mathbf{H}_l \) of no 4-cycles in Theorem 1. Here, we classify the Case I that nonzeros of \( \mathbf{H}_l \) collide with at least \( d_r - 2 \) nonzeros of \( \mathbf{H}_l \) and the other case.

Case I) The row \( \mathbf{h}'_k \) of \( \mathbf{H}' \) contains more than two nonzeros of the row \( \mathbf{h}_k \) and two nonzeros of the row \( \mathbf{h}_j \) of \( \mathbf{H}_l \). Similar to the proof of Theorem 1, it is proved that \( \mathbf{H}' \) has 4-cycles at each column.

Case II) In this case, nonzeros of \( \mathbf{h}_k \) of \( \mathbf{H}_l \) collide with \( d_r - 1 \) nonzero of \( \mathbf{h}_j \), if \( \mathbf{h}_j \) of \( \mathbf{H}_l \). Assume that there are a nonzero of \( \mathbf{h}_k \) at the \( r \)-th column and a nonzero of \( \mathbf{h}_j \) at the \( s \)-th column of \( \mathbf{H}_l \) when the \( b \)-th row \( \mathbf{h}_b \) of \( \mathbf{H}' \) is obtained by XORing the \( m \)-th row \( \mathbf{h}_m \) and the \( m \)-th row \( \mathbf{h}_m \) of \( \mathbf{H}_l \). If there are nonzeros at the \( r \)-th column and the \( s \)-th column for a certain row \( \mathbf{h}_k \), \( \mathbf{H}' \) has a 4-cycle. In the case that the \((x, r) \) row of \( \mathbf{H}_l \) is a nonzero, the \((l, r) \) row of \( \mathbf{H}' \) is a nonzero if the \((l, x) \) row of \( \mathbf{S} \) is a nonzero, where the \((l, x') \) row of \( \mathbf{S} \) is zero if the \((x', r) \) row of \( \mathbf{H}_l \). Similarly, in the case that the \((y, s) \) row of \( \mathbf{H}_l \) is a nonzero, the \((l, s) \) row of \( \mathbf{H}' \) is a nonzero if the \((l, y) \) row of \( \mathbf{S} \) is a nonzero, where the \((l, y') \) row of \( \mathbf{S} \) is zero if the \((y', s) \) row of \( \mathbf{H}_l \). Therefore, \( \mathbf{H}' \) has a 4-cycles at the \( r \)-th and \( s \)-th columns.

From Cases I and II, we confirm that \( \mathbf{H}' \) has 4-cycles at each columns under the condition of \( \mathbf{H} \) with 4-cycles. □

The matrices of row weight 2 and column weight 2 are singular, so we add a nonzero to \( \mathbf{S} \) in order to obtain nonsingular matrices of row weight almost 2 and column weight almost 2 in the proposed scheme. However, we assume that the weight of each row and column of \( \mathbf{S} \) is 2 in the above theorems and proofs. We can prove the theorems by the similar way to the above proofs, but we omit the detailed proofs because the discussion will be confused.

We confirm that undecodability of \( \mathbf{H}' = \mathbf{S}\mathbf{H}_l \) by iterative decoding algorithms. Figure 2 shows decoding fail-

![Fig. 2 Decoding failure probabilities of iterative decoding algorithm](image-url)
We evaluate the work factor under the case of uncorrected parity-check matrix. In this section, we evaluate the work factor of Eq. (12) depends on the error correction capability of the parity-check matrix. In this section, we evaluate the work factor under the case that the matrix is sparse and have a few ones, we can represent these parity-check matrices using the feature that the matrix is sparse also does not work with short cycles. Therefore, we can not attack the proposed scheme using the iterative decoding algorithm using the feature that the matrix is sparse also does not work with short cycles. Therefore, we can

\[
P_t = \frac{n}{2} \left( (n-k) (n+k) + (l_1 + l_2) \sum \left( k_1 \right) + \sum \left( k_2 \right) - k_1 \right) + \min(1, q_1) (p_1) (p_2) \sum \left( l_1 \right) + \min(1, q_2) (p_1) \sum \left( l_2 \right) + 2(t - p_1 - p_2 - q_1 - q_2)(p_1 + p_2) \left( l_1 \right) \left( l_2 \right) \cdot 2^{l_1 + l_2},
\]

where \( p_1, p_2, q_1, q_2, l_1, \) and \( l_2 \) are parameters of BCD and \( k = k_1 + k_2 \). Then, the security level is evaluated by the following work factor:

\[
WF = \frac{N}{P_t}. \tag{12}
\]

Using the Eq. (12), we investigate the parameters at the specific security level of the proposed scheme.

Parameters to be determined by the proposed scheme are the code length \( n \), the dimension \( k \), and the error correction capability \( t \) of parity-check matrices \( H \). Although the proposed scheme can be realized by not only regular LDPC codes but also irregular LDPC codes, it is necessary to include short cycles in \( H' \) in order to make it hard to solve the BSD problem. Since it depends on the number of nonzero elements in \( H' \), we assume that \( H' \) is a parity-check matrix of \((d_s, d_r)\)-regular LDPC codes. As shown in Sect. 3.4, it is possible to make a lot of 4-cycles in \( H' \) under the case of \( u = 2 \). The work factor of Eq. (12) depends on the error correction capability of the parity-check matrix. In this section, we evaluate the work factor under the case that \( H' \) is the parity-check matrix of (3, 6)-regular LDPC codes constructed by Gallager’s construction [15], which is famous LDPC code construction and its typical minimum distance is \( d = 0.02274n \).

Table 1 shows the parameters of the proposed scheme for each security level. The parity-check matrix of the proposed scheme has no algebraic structure and the iterative decoding algorithm using the feature that the matrix is sparse also does not work with short cycles. Therefore, we can

| Proposed scheme | LDPC codes with Gallager’s construction | Security level | \((n, k, t)\) | \((d_s, d_r)\) | \(u\) |
|-----------------|------------------------------------------|---------------|-------------|-------------|--------|
| Stern’s scheme  | Goppa codes                              | 50bit         | \((2820, 1410, 31)\) binary code | (3, 6) | 2 |
|                 |                                          | 80bit         | \((5802, 2901, 65)\) binary code | (3, 6) | 2 |
| Aguilar et al.’s scheme | Asymptotically good codes given by Gibert-Varshamov bound | 50bit         | \((1024, 524, 50)\) binary code | — | — |
|                 |                                          | 80bit         | \((2048, 1751, 27)\) binary code | — | — |
| Hu et al.’s scheme | Asymptotically good \(q\)-ary codes given by Gibert-Varshamov bound | 81bit         | \((698, 349, 70)\) binary code | — | — |

Table 2 Key size of identification schemes

| Proposed scheme | Security level | Key size [bit] | Key size | Public key |
|-----------------|---------------|---------------|----------|------------|
|                 | 50bit         | 228,420       | 272,138  |
| Stern’s scheme  | 80bit         | 469,962       | 559,901  |
|                 | 80bit         | 1,047         | 1,404    |
| Aguilar et al.’s scheme | 81bit         | 2,048        | 608,561  |
|                 | \(q = 3\)    | 80bit         | 792      | 78,812    |
|                 | \(q = 4\)    | 80bit         | 656      | 54,128    |
|                 | \(q = 5\)    | 80bit         | 876      | 64,394    |

For Stern’s identification scheme, the secret key is a vector \( x \in \mathbb{F}_2^n \), so the secret key size is \( n \) bits. The public key is an \((n-k) \times n\) matrix \( H \), a vector \( x \in \mathbb{F}_2^{n-k} \) and an integer \( u \). So, the public key size is \( n(n-k) + n-k + 8 \) bits, where the integer is stored within a memory of 1byte.

For the proposed scheme, the secret key is a vector \( x \in \mathbb{F}_2^n \), an \((n-k) \times n\) matrix \( H \) of row weight \( d_s \), an \((n-k) \times (n-k)\) matrix \( S \) of row weight \( n \) and an \( n \times n \) matrix \( P \) of row weight 1. Since \( H, S \) and \( P \) have a few ones, we can represent these matrices by the position of ones. So, the secret key size is \( n + 16d_s(n-k) + 16u(n-k) + 16n \) bits, where each position is stored within a memory of 2bytes. Also, the public key is an \((n-k) \times n\) matrix \( H' \) of row weight \( ud_s \), a vector \( x \in \mathbb{F}_2^{n-k} \) and an integer \( u \). The public key size is \( 16ud_s(n-k) + n-k + 8 \).

Table 2 shows the key size of Stern’s scheme and the proposed scheme for each security level. The public key size of the proposed scheme is smaller than that of Stern’s scheme. For 50bit security, the public key size of the proposed scheme is half of that of Stern’s scheme. Although the secret key size of the proposed scheme is larger than that of Stern’s scheme, total key size of the proposed scheme is smaller than that of Stern’s scheme. Additionally, the key size of Aguilar et al.’s scheme and Hu et al.’s scheme is shown in Table 2. The estimate of key size is referred to appendix. In Aguilar et al.’s scheme, the matrix for public key is constructed by double circulant matrix to reduce the
public key size. In Hu et al.’s scheme, the parity-check matrix for q-ary codes is represented by the systematic form to reduce the public key size. Therefore, the key size of these schemes is smaller than that of our scheme.

4.3 Computational Cost

In this section, we compare the computational cost of Stern’s identification scheme and the proposed scheme. The procedure in which the calculation process occurs in the identification scheme is the generation of the public key, the generation of c1, c3 in the commitment, and the verification in the case of b = 0, 1, 2. In addition, hash function h(), P and σ() does not significantly change the size of input data.

The row weight of H′ is d′, and the row weight of the matrix S is u. To calculate H′ = SH′, each row of H′ is obtained by XORing the rows of H′ u times. So, the cost of generating the public key H′ = SH′P is

ud′(n − k).

As shown in Sect. 3.3, the cost of syndrome calculation is generally

(2n − 1)(n − k).

On the other hand, the row weight of H′ is at most ud′, so the cost of the syndrome calculation of the proposed scheme is

(ud′ − 1)(n − k).

The estimated cost of syndrome calculation is led to the cost of each calculation in Table 3.

In Stern’s identification scheme, the cost for generating the public key is

(2n − 1)(n − k),

and the cost of the commitment is

(2n − 1)(n − k) + n. (14)

The verification for b = 0, 1, 2 is executed with probability 1/3 at each round, so the cost of verification is

\[ \frac{2}{3} \{ (2n - 1)(n - k) + n \} \]. (15)

Therefore, the cost of whole steps of Stern’s identification scheme for r iterations is given by

\[ (2n - 1)(n - k) + \frac{5}{3} r \{ (2n - 1)(n - k) + n \} \]. (16)

In the proposed scheme, the cost of generating the public key is

(2ud′ − 1)(n − k),

the cost of commitment is

(ud′ − 1)(n − k) + n, (18)

and the cost of verification is

\[ \frac{2}{3} \{ (ud′ - 1)(n - k) + n \} \]. (19)

Therefore, the cost of whole steps of the proposed scheme for r iterations is given by

\[ (2ud′ - 1)(n - k) + \frac{5}{3} r \{ (ud′ - 1)(n - k) + n \} \]. (20)

We apply the parameters in Table 1 to Eqs. (16) and (20) and evaluate the computational cost of Stern’s scheme and the proposed scheme. In the proposed and Stern’s schemes, they need 28 iterations to reach failure probability of identification to 2−16. Hence, we evaluate the computational cost in case of r = 28. Table 4 shows the computational cost of Stern’s scheme and the proposed scheme. Additionally, the computational cost of Aguilar et al.’s and Hu et al.’s schemes is shown in Table 4. Note that Aguilar et al.’s scheme needs 18 iterations to reach failure probability of identification to 2−16. The estimate of these computational cost is referred to appendix. We confirm from Table 4 that the cost of the proposed scheme is smaller than that of Stern’s and Aguilar et al.’s schemes. Hu et al.’s scheme performs the operations over \( \mathbb{F}_q \) since it is designed by q-ary codes. In Table 4, mul. and add. imply the number of multiplication and addition in \( \mathbb{F}_q \) respectively. Although the cost of multiplication and addition in \( \mathbb{F}_q \) depends on implementation and calculation algorithm, the cost of addition in \( \mathbb{F}_q \) is larger \( \lceil \log_2 q \rceil \) times than that in \( \mathbb{F}_2 \) and the cost of multiplication in \( \mathbb{F}_q \) is larger \( \lceil \log_2 q \rceil^2 \) times than that in \( \mathbb{F}_2 \). Therefore, the cost of our scheme is smaller than that of Hu et al.’s scheme.

| Calculation | Cost | Calculation | Cost |
|-------------|------|-------------|------|
| Generation of public key | s(= xH′) | H′ (= SH′P) | ud′(n − k) |
| Commitment | c1 H′ | (2n − 1)(n − k) | H′g′ | (ud′ − 1)(n − k) |
| Verification | b = 0 H′ | (2n − 1)(n − k) | H′g′ | (ud′ − 1)(n − k) |
| | b = 1 H(y + x) | (2n − 1)(n − k) | H′(y + x)′ | (ud′ − 1)(n − k) |
| | b = 2 σ(y) | (2n − 1)(n − k) | σ(y) + σ(x) | (ud′ − 1)(n − k) |
| | l udr | Wl(σ(x)) = w | n | Wl(σ(x)) = w | n |
Table 4  Computational cost of identification schemes

| Proposed scheme | Security level | $r$ | Cost |
|-----------------|---------------|-----|------|
| 50bit           | 28            | 887,830 |      |
| 80bit           | 28            | 1,826,663 |    |
| Stern’s scheme  | 50bit         | 48,834,620 |
| 80bit           | 28            | 58,068,488 |
| Aguilar et al.’s scheme | q = 3 | 80bit | 822,096 mul. + 812,064 add. |
|                 | q = 4         | 80bit | 565,691 mul. + 557,381 add. |
|                 | q = 5         | 80bit | 449,291 mul. + 441,893 add. |

5. Conclusion

In this paper, we have proposed a ZK identification scheme using LDPC codes. In the proposed scheme, the computational cost of identification is reduced by applying the sparse parity-check matrix of LDPC codes to the BSD problem. If the sparse parity-check matrix is directly applied to the BSD problem, the computational cost of solving the problem is reduced by the iterative decoding algorithm. Thus, there is a possibility to obtain the secret key from the public key. Theorems 1 and 2 show that it is possible to create more 4-cycles within $H'$ by multiplying the matrix $S$ of the row weight 2 and the column weight 2 by the parity-check matrix $H_i$ of the LDPC code and making it $H' = SH_iP$. By using $H'$ as the binary syndrome decoding problem, it is hard to solve the problem by the iterative decoding algorithm. In coding theory, the goal is to construct a code with high error correction capability and easy to decode, but the identification scheme is realized by daring to convert the LDPC code with the high error correction capability to the structure hard to decode in the proposed scheme.

We have discussed the security level, computational cost and undecodability of the proposed scheme in detail. Although the length and dimension of codes for our scheme are larger than those of Stern’s scheme at the same security level, the computational cost of our scheme are greatly reduced. Nevertheless, the size of secret and public keys of our scheme is larger than that of Aguilar et al.’s and Hu et al.’s schemes. It leads that the transmission data and the communication cost increase. To future works, we will improve our scheme to reduce the key size and the communication cost.

References

[1] M.B. Barcena and C. Wueest, “Insecurity in Internet of Things,” Symantec Security Response, https://www.a51.nl/sites/default/files/pdf/insecurity-in-the-internet-of-things.pdf, March 2015.
[2] J. Stern, “A new paradigm for public key identification,” IEEE Trans. Inf. Theory, vol.42, no.6, pp.1757–1768, Nov. 1996.
[3] R.J. McEliece, “A public-key cryptosystem based on algebraic coding theory,” DSN Progress Report 42, pp.114–116, 1978.
[4] H. Niederreiter, “Knapsack-type cryptosystem based on algebraic coding theory,” Problems of Control and Information Theory, vol.15, no.2, pp.157–166, 1986.
[5] C. Monico, J. Rosenthal, and A. Shokrollahi, “Using low density parity check codes in the McEliece cryptosystem,” Proc. 2000 IEEE Int. Symp. Inf. Theory (ISIT 2000), Sorrento, Italy, p.215, 2000.
[6] M. Baldi, M. Bodrato, and F. Chiaraluce, “A new analysis of the McEliece cryptosystem based on QC-LDPC codes,” Security and Cryptography for Networks (SCN 2008), R. Ostrovsky, R.D. Prisco, and I. Visconti Eds., Lecture Notes in Computer Science, vol.5229, pp.246–262, Springer, Berlin, Heidelberg, 2008.
[7] R. Misoczki, J.-P. Tillich, N. Sendrier, and P.S.L.M. Barreto, “MDPC-McEliece: New McEliece variants from moderate density parity-check codes,” Proc. 2013 IEEE Int. Symp. Inf. Theory (ISIT 2013), Istanbul, Turkey, pp.2069–2073, 2013.
[8] A. Fiat and A. Shamir, “How to prove yourself Practical solutions to identification and signature problems,” Crypto 86, Lecture Notes in Computer Science, vol.263, pp.181–187, 1986.
[9] P. Gaborit and M. Girault, “Lightweight code-based identification and signature,” Proc. 2007 IEEE Int. Symp. Inf. Theory (ISIT 2007), Nice, France, pp.191–195, 2007.
[10] P. Véron, “Improved identification schemes based on error-correcting codes,” Applicable Algebra in Engineering, Communication and Computing, vol.8, no.1, pp.57–69, Jan. 1997.
[11] P.-L. Cayrel, P. Véron, and S.M.E.Y. Alaoui, “A zero-knowledge identification scheme based on the q-ary syndrome decoding problem,” Proc. 17th Int. Workshop on Selected Areas in Cryptography (SAC 2010), A. Biryukov, G. Gong, D. Stinson, Eds., Lecture Notes in Computer Science, vol.6544, pp.171–186, Springer, Berlin, Heidelberg, 2011.
[12] R. Hu, K. Morozov, and T. Takagi, “On zero-knowledge identification based on q-ary syndrome decoding,” Proc. 8th Asia Joint Conference on Information Security (AsiaJCIS 2013), Seoul, South Korea, pp.12–18, July 2013.
[13] C. Aguilar, P. Gaborit, and J. Schrek, “A new zero-knowledge code based identification scheme with reduced communication,” Proc. 2011 IEEE Information Theory Workshop (ITW 2011), Paraty, Brazil, pp.648–652, Oct. 2011.
[14] E. Berlekamp, R. McEliece, and H. van Tilborg, “On the inherent intractability of certain coding problems,” IEEE Trans. Inf. Theory, vol.24, no.3, pp.384–386, May 1978.
[15] R.G. Gallager, Low Density Parity Check Codes, Cambridge, MA: MIT Press, 1963.
[16] J. Stern, “A method for finding codewords of small weight,” Coding Theory and Applications 1988, G. Cohen and J. Wolfmann Eds., Lecture Notes in Computer Science, vol.388, pp.106–113, New York: Springer-Verlag, 1989.
[17] F. Chabaud, “On the security of some cryptosystems based on error-correcting codes,” Advances in Cryptology - EUROCRYPT’94, Lecture Notes in Computer Science, vol.950, pp.131–139, Springer, Berlin, Heidelberg, 1995.
[18] A. Canteaut and N. Sendrier, “Cryptanalysis of the original McEliece cryptosystem,” Advances in Cryptology - ASIACRYPT’98, Lecture Notes in Computer Science, vol.1514, pp.187–199, Berlin, Springer, Heidelberg, 1998.
[19] A. Canteaut and F. Chabaud, “A new algorithm for finding minimum-weight words in a linear code: application to McEliece’s cryptosystem and to narrow-sense BCH codes of length 511,” IEEE Trans. Inf. Theory, vol.44, no.1, pp.367–378, Jan. 1998.
[20] D.J. Bernstein, T. Lange, and C. Peters, “Attacking and defending the McEliece cryptosystem,” Post-Quantum Cryptography (PQCrypto 2008), J. Buchmann and J. Ding Eds., Lecture Notes in Computer Science, vol.5299, pp.31–46, Springer, Berlin, Heidelberg, 2008.
[21] D.J. Bernstein, T. Lange, and C. Peters, “Smaller decoding exponents: Ball-collision decoding,” Advances in Cryptology (CRYPTO 2011), P. Rogaway Ed., Lecture Notes in Computer Science, vol.6841, pp.743–760, Springer, Berlin, Heidelberg, 2011.
[22] M. Repka and P. Zajac, “Overview of the McEliece cryptosystem and its security,” Journal of Slovak Academy of Sciences, vol.60, no.1, pp.57–83, Sept. 2014.
### Appendix A: Aguilar et al.’s Identification Scheme

The notation and procedure of Aguilar et al.’s identification scheme using \((n, k, t)\) codes are referred to [13].

The secret key is a vector \(e \in \mathbb{F}_q^n\) of weight \(w\) and a random vector \(m \in \mathbb{F}_q^n\), so the secret key size is \(n + k\) bits. The public key is a generator matrix \(G\) for an \((n, k, t)\) code, a vector \(x(= mG + e) \in \mathbb{F}_q^n\), and the weight \(w\) of vector \(e\). In Aguilar et al.’s scheme, the \(k \times n\) matrix \(G\) is constructed by a double-circulant matrix, so it is represented by \(n\) bits. Thus, the public key size is \(2n + 8\) bits.

Table A-1 shows the estimated cost of each calculation in Aguilar et al.’s scheme. For the calculation of \(mG\), \(nk\) multiplications and \(n(k - 1)\) additions are required to obtain the product of \(k \times n\) matrix and \(k\)-bit vector. Thus, the cost of generating the public key \(x = e + mG\) is given by

\[
2nk. \tag{A-1}
\]

The cost of the commitment is

\[
2nk. \tag{A-2}
\]

The response for \(b = 0, 1\) is executed with probability 1/2 at each round, so the cost of response is

\[
\frac{1}{2}k. \tag{A-3}
\]

The verification for \(b = 0, 1\) is executed with probability 1/2 at each round, so the cost of verification is

\[
k + \frac{1}{2}n. \tag{A-4}
\]

Therefore, the cost of whole steps of Aguilar et al.’s scheme for \(r\) iterations is given by

\[
4nk + \frac{1}{2}r(k + 2nk + n). \tag{A-5}
\]

### Appendix B: Hu et al.’s Identification Scheme

The notation and procedure of Hu et al.’s scheme using \((n, k, t)\) \(q\)-ary codes are referred to [12].

The secret key is a vector \(e \in \mathbb{F}_q^n\) of weight \(w\), so the secret key size is \(nN\) bits where \(N = \lceil \log_2 q \rceil\). The public key is a parity-check matrix \(H\) for \((n, k, t)\) \(q\)-ary codes and the weight \(w\) of vector \(e\). In Hu et al.’s scheme, the parity-check matrix \(H\) takes the systematic form \([I \ H']\) where \(I\) is an identity matrix. Thus, the public key size is \((n - k)(k + 1)N + 8\) bits.

Table A-1 shows the estimated cost of each calculation in Hu et al.’s scheme. Since \(H\) takes the systematic form, the product of \(H\) and \(e = (e_1, e_p)\) is calculated as follows:

\[
y = He^T = [I \ H'](e_t, e_p)^T = e_t + He_p^T
\]

Thus, the cost of generating the public key is

\[
k(n - k) \text{ mul.} + (k - 1)(n - k) \text{ add.} \tag{A-6}
\]

The transformation \(\Pi_{\gamma, \Sigma}\) is defined as follows:

\[
\Pi_{\gamma, \Sigma} : \mathbb{F}_q^n \to \mathbb{F}_q^n
\]

\[
v \mapsto (\gamma_{\Sigma(1)}v_{\Sigma(1)}, \ldots, \gamma_{\Sigma(n)}v_{\Sigma(n)}),
\]

where \(\Sigma\) is a permutation on \([1, \ldots, n]\) and \(\gamma = (\gamma_1, \ldots, \gamma_n) \in \mathbb{F}_q^n\). The cost of this transformation is \(n\) mul, so the cost of the commitment is given by

\[
k(n - k) + 2n \text{ mul.} + (k - 1)(n - k) + n \text{ add.} \tag{A-7}
\]

The response for \(b = 0, 1, 2\) is executed with probability 1/3 at each round, so the cost of response is

\[
\frac{1}{3}n \text{ mul.} \tag{A-8}
\]

The verification for \(b = 0, 1, 2\) is executed with probability 1/3 at each round, so the cost of verification is

\[
\frac{2}{3}(k(n - k) + n) \text{ mul.} + \frac{1}{3}(2k(n - k) + k) \text{ add.} \tag{A-9}
\]

Therefore, the cost of whole steps of Hu et al.’s scheme for \(r\) iterations is given by

\[
2k(n - k) + 2n + r\left(\frac{2}{3}k(n - k) + n\right) \text{ mul.}
\]

\[
+ 2(k - 1)(n - k) + n + \frac{1}{3}r(2k(n - k) + k) \text{ add.} \tag{A-10}
\]
Haruka Ito received the B.E. degree from Saga University, Japan in 2017. Since 2017, she has been a master’s student in Graduate School of Science and Engineering, Saga University. Her current research interest is in information security.

Masanori Hirotomo received the B.E., M.E. and D.E. degrees from the University of Tokushima, Japan, in 2000, 2002, and 2006 respectively. From 2005 to 2006 he was a Research Associate at the Department of Intelligent Systems and Information Science, Faculty of Engineering, the University of Tokushima, Japan. From 2006 to 2008 he was a Researcher at the Hyogo Institute of Information Education Foundation, Japan. From 2008 to 2011 he was an Assistant Professor at the Graduate School of Engineering, Kobe University, Japan. From 2011 to 2013 he was an Assistant Professor at the Computer and Network Center, Saga University, Japan. Since 2013, he has been an Associate Professor at the Graduate School of Science and Engineering, Saga University, Japan. His research interests are in coding theory and information security. He is a member of the IEEE.

Youji Fukuta received his M.E. degree from the University of Tokushima in 2002. He joined Aichi University of Education in 2005 and has been engaged in research on information security. In 2007, he received his Ph.D. degree in Engineering from the University of Tokushima. Since 2016, he has been a lecturer at the Department of Informatics, Kindai University, Japan. He is a member of IEEE and IPSJ.

Masami Mohri received B.E. and M.E. degrees from Ehime University, Japan, in 1993 and 1995 respectively. She received Ph.D degree in Engineering from the University of Tokushima, Japan in 2002. From 1995 to 1998 she was an assistant professor at the Department of Management and Information Science, Kagawa Junior College, Japan. From 1998 to 2002 she was a research associate of the Department of Information Science and Intelligent Systems, the University of Tokushima, Japan. From 2003 to 2007 she was a lecturer of the same department. From 2007 to 2017, she was an associate professor at the Information and Multimedia Center, Gifu University, Japan. Since 2017, she has been an associate professor at the Department of Electrical, Electronic and Computer Engineering in the same university. Her research interests are in coding theory, information security and cryptography. She is a member of IEEE.

Yoshiaki Shiraishi received B.E. and M.E. degrees from Ehime University, Japan, and Ph.D degree from the University of Tokushima, Japan, in 1995, 1997, and 2000, respectively. From 2002 to 2006 he was a lecturer at the Department of Informatics, Kindai University, Japan. From 2006 to 2013 he was an associate professor at the Department of Computer Science and Engineering, Nagoya Institute of Technology, Japan. Since 2013, he has been an associate professor at the Department of Electrical and Electronic Engineering, Kobe University, Japan. His current research interests include information security, cryptography, computer network, and knowledge sharing and creation support. He received the SCIS 20th Anniversary Award and the SCIS Paper Award from ISEC group of IEICE in 2003 and 2006, respectively. He received the SIG-ITS Excellent Paper Award from SIG-ITS of IPSJ in 2015. He is a member of IEEE, ACM, and a senior member of IPSJ.