Regular Article

Secure Information Flow for IoT Applications

Tri Minh Ngo, Vien Nguyen-Duy-Nhat

The University of Danang–University of Science and Technology, Danang, Vietnam

Correspondence: Tri Minh Ngo, tringominh@gmail.com

Communication: received 02 April 2019, revised 02 May 2019, accepted 06 May 2019

Online publication: 23 November 2019, Digital Object Identifier: 10.21553/rev-jec.233

The associate editor coordinating the review of this article and recommending it for publication was Dr. Nguyen Tan Hung.

Abstract– **This paper discusses how to ensure security, *i.e.*, confidentiality and integrity properties, for data in IoT applications. While confidentiality could be assessed via information flow analysis, integrity is ensured by error-correcting codes. In addition to errors, many communication channels also cause erasures, *i.e.*, the demodulator cannot decide which symbol the received waveform represents. The paper proposes a method that might correct both errors and erasures together. Our method is efficient in reducing memory storage as well as decoding complexity.**

Keywords– **Confidentiality, Integrity, Information Flow, Erasure, Separating Matrix, Covering Design.**

1 Introduction

It is estimated that Internet of Things (IoT) will generate billions of dollars in profit for industries over the next two decades. Many organizations have started to develop and implement their own IoT strategies. IoT enables devices would generate and transmit so many data such that security should be a top concern. IoT users require that communication technologies have to guarantee both efficiency and security. This paper discusses how to guarantee two main properties of security, *i.e.*, *confidentiality* and *integrity*, for IoT applications.

1.1 Confidentiality

Securing the data manipulated by information systems has been a challenge in the past few years. Several methods to limit the information disclosure has been proposed, such as *access control*, and *cryptography*. These are useful approaches, *i.e.*, they can prevent confidential information from being read or modified by *unauthorized* users. However, they still have a fundamental limitation, *i.e.*, they do not regulate the information propagation after it has been released. For example, access control prevents unauthorized file access, but is insufficient to control how the data is used afterwards. Similarly, cryptography provides a shield to exchange information privately across a non-secure channel, but no guarantee about the confidentiality of private data is given after it is decrypted. Thus, neither access control nor encryption provide a *complete* solution to protect confidentiality for information systems.

To ensure confidentiality for an information system, *i.e.*, IoT system, it is necessary to show that the system *as a whole* enforces a confidentiality policy, *i.e.*, by analyzing how information flows within the system. The analysis must show that information controlled by a confidentiality policy cannot flow to a place where that policy is violated. Thus, the confidentiality policy we wish to enforce is an *information flow policy*, and the method that enforces them is an *information flow analysis*.

Information flow analysis is a technique that has recently become an active research topic. In general, the approach of information flow security is based on the notion of *interference* [1]. Informally, *interference* exists inside a system when the private data affect public data, e.g., an attacker might guess private data from observing public data. *Non-interference*, *i.e.*, the absence of interference, is often used to prove that an information system is secured.

Non-interference is required for applications where the users need their private data strictly protected. However, many practical applications related to IoT might *intentionally* violate non-interference by leaking *minor* information. Such systems include password checkers, cryptographic operations etc. For instance, when an attacker tries to guess the password: even when the attacker makes a wrong guess, secret information has been leaked, *i.e.*, it reveals information about what the real password is not. Similarly, there is a flow of information from the plain-text to the cipher-text, since the cipher-text depends on the plain-text. These applications are rejected by the definition of non-interference.

Actually, the insecure property will happen only when it exceeds a specific threshold, or amount of interference. If the interference in the system is small enough, e.g., below a threshold given by specific security policy, the system is considered to be secure. The security analysis that requires to determine how much information flows from high level, *i.e.*, secret data, to low level, *i.e.*, public output, is known as *quantitative information flow*. It concerned with measure the leakage of information in order to decide if the leakage is tolerable.

Qualitative information flow analysis, *i.e.*, non-interference, aims to determine whether a program leaks private information or not. Thus, these absolute security properties always reject a program if it leaks any information. *Quantitative* information flow analysis offers a more general security policy, since it gives a method to tolerate a minor leakage, *i.e.*, by computing how much information has been leaked and comparing this with a threshold. By adjusting the threshold, the security policy can be applied for different applications, and in particular, if the threshold is 0, the quantitative policy is seen as a qualitative one. The idea of quantitative information flow analysis has been discussed in details in [2], one of our papers; readers can refer to it for more information.

1.2 Integrity

Integrity means maintaining and assuring accuracy and completeness of data. However, during the wireless transmission in IoT applications, messages can be erroneous due to noisy channels. Error means the received symbol is different from the transmitted symbol. In order to protect data against errors, error correcting codes techniques are required. Error-correcting codes are applied in situations where retransmissions are relatively costly or impossible. Error-correcting codes ensure proper performance of IoT systems. They ensure the integrity of communication links in the presence of noise, distortion, and attenuation [3–6]. The use of a parity-bit as an error-detecting mechanism is one of the simplest and most well-known schemes used in digital communication. Data is portioned into blocks. To each block, an additional bit is appended to make the number of bits which are 1 in the block, including the appended bit, an even number. If a single bit-error occurs, within the block, the number of 1's becomes odd. Hence, this allows for detection of single errors [7, 8].

The most applications of error-correcting codes are in telecommunications. Many early applications of coding were developed for deep-space and satellite communication systems. For example, satellite photos were taken in space and sent back to earth. The channel for such transmission is space and the earth's atmosphere. These communication systems have limitations on their transmitted power. Solar activity and atmospheric conditions can introduce errors into weak signals coming from the spacecraft. Error-correcting codes are an excellent mean to guarantee reliable communications. With the applications of error-correcting codes, most of the pictures sent could be correctly recovered here on earth. As examples, a binary (32,6,16) Reed-Muller code was used during the Mariner and Viking mission to Mars around 1970 or a convolutional code was used on the Pioneer 10 and 11 missions to Jupiter and Saturn in 1972. The (24,12,8) Golay code was used in the Voyager 1 and Voyager 2 spacecrafts transmitting color pictures of Jupiter and Saturn in 1979 and 1980. When Voyager 2 went on to Uranus and Neptune, the code was switched to a concatenated Reed-Solomon code for its substantially more powerful error correcting capabilities.

The block and convolutional codes are also applied to the Global System for Mobile communications (GSM) which is the most popular digital cellular mobile communication system. Reed Solomon and Viterbi codes have been used for nearly 20 years for the delivery of digital satellite TV. Low-density parity-check codes (LDPC codes) are now used in many recent high-speed communication standards, such as Digital video broadcasting-S2 (DVB-S2), WiMAX, 10GBase-T Ethernet [9].

Most error correcting codes, in general, are designed to correct or detect errors. However, many channels cause erasures, *i.e.*, the demodulator cannot decide which symbol the received waveform represents, in addition to errors. In principle, decoding over such channels can be accomplished by deleting erased symbols and decoding the resulting vector with respect to the punctured code, *i.e.*, the code in which all erasures have been removed. For any given linear code and any given maximum number of correctable erasures, in [7], Abdel-Ghaffar and Weber introduced a parity-check matrix yielding parity-check equations that do not check any of the erased symbols and which are sufficient to characterize the punctured code. This allows for the separation of erasures from errors to facilitate decoding. However, these parity-check matrices have too many redundant rows. To reduce decoding complexity, parity-check matrices with small number of rows are preferred. This paper proposes a method that can build a matrix with a smaller number of rows.

Organization of the paper. The rest of this paper is organized as follows. Section 2 introduces the main ideas of error-correcting codes, errors and erasures. Section 3 presents methods to construct a parity-check matrix that can correct both errors and erasures. Section 4 discusses a general solution for the covering design, which is used in the proposal. Finally, Section 5 concludes the paper.

2 Codes, Errors and Erasures

2.1 Linear Block Codes

Let C be an $[n,k,d]$ linear block code. It means that C is a k-dimensional subspace of the n-dimensional vector space. The set of codewords of C can be defined as the null space of the row space of an $r \times n$ parity-check matrix $\mathbf{H} = (h_{i,j})$ of rank $n-k$. Since a vector $\mathbf{x}$ is a codeword of C iff $\mathbf{x}\mathbf{H}^T = \mathbf{0}$, where the superscript T denotes the transpose, we can derive r parity-check equations PCE, as follows,

$$\text{PCE}i\text{: } \textstyle\sum_{j=1}^{n} h_{i,j}x_j = 0 \text{ for } i = 1,2,...,r.$$

An equation PCE$i(\mathbf{x})$ is said to check $\mathbf{x}$ in position j iff $h_{i,j} \neq 0$.

2.2 Erasures

Sometimes, at the receiver, the demodulator cannot decide which symbol the received waveform represents.

In this case, we declare the received symbol as an erasure. When the received codeword contains erasures instead of errors, the iterative decoding can be used [8].

Here, we summarize the iterative decoding procedure using an example of the (7,4,3) binary Hamming code with the following parity-check matrix,

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

Since a vector $\mathbf{x}{=}(x_1 x_2 x_3 x_4 x_5 x_6 x_7)$ is a codeword iff $\mathbf{x}\mathbf{H}^T = 0$. Hence, every codeword has to satisfy three parity-check equations as follows,

$$x_1 + x_3 + x_4 + x_5 = 0, \quad (1)$$

$$x_2 + x_4 + x_5 + x_6 = 0, \quad (2)$$

$$x_3 + x_5 + x_6 + x_7 = 0. \quad (3)$$

Assume that the received vector is $**010*0$, where the erased symbol is denoted by $*$. Equation (1) checks on x_1, x_3, x_4 and x_5. If exactly one of these four symbols is erased, it can be retrieved from this equation. Thus, $x_1 = 1$ since $x_3 = 0$, $x_4 = 1$, and $x_5 = 0$. Similarly, we can derive that $x_2 = 1$, and $x_6 = 0$ from Equation (2) and (3). Therefore, the iterative decoding decided that the transmitted codeword is 1101000.

Iterative decoding is successful iff erasures do not fill the positions of a *nonempty stopping* set. A stopping set is a set of positions in which there is no parity-check equation that checks exactly one symbol in these positions. The performance of iterative decoding techniques for correcting erasures depends on the sizes of the stopping sets associated with the parity-check matrix representing the code. The parity-check matrix with redundant rows could benefit the decoding performance, *i.e.*, reducing the size of stopping sets, while increasing the decoding complexity. More information on stopping set can be found in [2, 10, 11].

2.3 Separation of Errors from Erasures

In this part, we discuss how to handle errors together with erasures. In this case, we can apply an algorithm using trials in which erasures are replaced by 0 or 1; and the resulting vector is decoded by a decoder which is capable of correcting errors. For binary code, two trials are sufficient [8, 12].

For example, if C is a binary (n,k)-code with a Hamming distance $d = 2t_\epsilon + t_? + 1$, then C can correct t_ϵ errors and $t_?$ erasures. In the presence of no erasures, C is able to correct up to $t_\epsilon + \lfloor t_?/2 \rfloor$ errors. Let $\mathbf{r}$ be a received vector having at most t_ϵ errors and at most $t_?$ erasures. Suppose the decoder constructs two vectors $\mathbf{r}_0$ and $\mathbf{r}_1$, where $\mathbf{r}_i$ is obtained by filling all erasure positions in $\mathbf{r}$ with the symbols i, $i = 0, 1$. Since C is binary, in either $\mathbf{r}_0$ or $\mathbf{r}_1$, at least half the erasure locations have the right symbols. Hence, either $\mathbf{r}_0$ or $\mathbf{r}_1$ has a distance at most $t_\epsilon + \lfloor t_?/2 \rfloor$ from the transmitted codeword. Thus, any standard error correction technique can be applied. If the correction decodes both $\mathbf{r}_0$ and $\mathbf{r}_1$ to codewords, and these codewords are the same, then this is the transmitted codeword. If they are different, then there is one, and only one, vector requiring at most t_ϵ changes in non-erasure positions to become the right codeword. More information on this algorithm can be found in [6].

$$\mathbf{H} = \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}\right]$$

Figure 1. A parity check matrix for the code C.

Abdel-Ghaffar and Weber proposed another way of decoding over such channels [7]. First, all erasures are deleted from the received message. Errors in the resulting codeword will be corrected based on the punctured code, *i.e.*, codewords consist of symbols in positions which are not erased. After all errors have been corrected, the erasures will be recovered by the iterative decoding.

The decoder can compute a parity-check matrix for the punctured code after receiving the codeword. However, this leads to time delay which is unacceptable specially in IoT applications. To reduce time delay, we can store parity-check matrices of all punctured codes corresponding to all erasure patterns. The drawback of this solution is the requirement of huge memory storage at the decoder.

Abdel-Ghaffar and Weber proposed using a *separating matrix* with redundant rows, providing enough parity-check equations which do not check any of the erased symbols and are sufficient to form a parity-check matrix for the punctured code obtained by deleting all erasures [7]. Having these parity-check equations not checking any of the erased symbols lead to the concept of *separation* of errors from erasures.

The basic concept of this decoding technique can be illustrated by an example as follows. We consider an [8,4,4] binary extended Hamming code with the following parity-check matrix, given in Figure 1.

A normal parity-check matrix just has only four rows as the first four rows in this separating matrix. Allowing redundant rows simplifies the decoding of erasures in addition to errors. Assume that we get a codeword $\mathbf{r} = 0*011000$ with one erasure in the second position. Applying the decoding technique mentioned above, firstly we delete the erasure and the resulting vector is $\mathbf{r}' = 0011000$. This vector $\mathbf{r}'$ can be considered as a codeword of the (7,4,3) punctured code. In $\mathbf{H}$, the first, the second and the sixth row have zeros in the second position. It means that three corresponding parity-check equations do not check the erased symbol. Based on these three rows, we can form a parity-check matrix $\mathbf{H}'$ for the punctured code, as given in Figure 2.

Using $\mathbf{H}'$, $\mathbf{r}'$ is decoded into 0011010. Putting back the erasure, we get $0*011010$. The third row of $\mathbf{H}$, which checks the erased symbol, can be used to recover the erasure. Thus, the decoded codeword corresponding to $\mathbf{r}$ is 01011010.

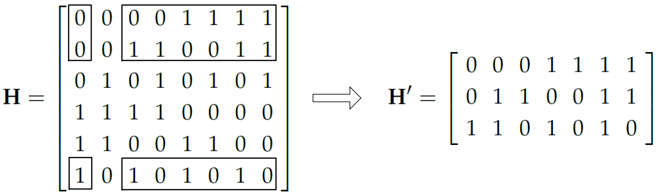


Figure 2. A parity check matrix for the punctured code.

A normal parity-check matrix cannot be used for decoding of both errors and erasures together. Decoding is feasible when we pay the price of storing a parity-check matrix with more rows than a normal one. In order to reduce the memory storage as well as the decoding complexity, a parity-check matrix with small number of rows is preferred.

Given any linear code and any given maximum number of correctable erasures, Abdel-Ghaffar and Weber introduced separating matrices yielding parity-check equations that do not check any of the erased symbols and which are sufficient to characterize all punctured codes corresponding to this maximum number of erasures [7]. This allows for the separation of erasures from errors to facilitate decoding. However, their proposal yields separating matrices which typically have *too many* redundant rows. The following part of this paper discusses an improved method to construct such separating matrices, applying covering design, with a *smaller number of rows*.

3 How to Build an L-separating Matrix

3.1 Set Separation

Let $\mathbf{H} = (h_{i,j})$ of rank $n-k$ be an $(r \times n)$ parity-check matrix of C, $r \geq n-k$. Let S be a subset of $\{1,2,\ldots,n\}$ and T be a subset of $\{1,2,\ldots,r\}$, define $\mathbf{H}_S^T = (h_{i,j})$ with $i \in T$ and $j \in S$, be a $|T| \times |S|$ submatrix of $\mathbf{H}$. For the code C with the length n, define $C_{\overline{S}} = \{c_{\overline{S}} : c \in C\}$ be the punctured code consisting of all codewords of C in which the symbols in positions indexed by S, $S = \{1,2,...,n\} \setminus \overline{S}$ are deleted. Clearly, $C_{\overline{S}}$ is a linear code over $GF(q)$ of length $n' = |\overline{S}|$, dimension $k' \leq k$, and Hamming distance $d' \leq d - |\overline{S}|$. Let $\widetilde{S} = \{i : 1 \leq i \leq r, h_{i,j} = 0, \forall j \in S\}$, define $\mathbf{H}(S) = \mathbf{H}^{\widetilde{S}}_{\overline{S}}$.

Definition 1 **([7]).** *A parity-check matrix $\mathbf{H}$ of an $[n,k,d]$ linear code C separates a set S of size $|S| \leq d-1$ iff $\mathbf{H}(S)$ has rank $n-k-|S|$.*

Theorem 1 ([7]). *A parity-check matrix $\mathbf{H}$ of an $[n,k,d]$ linear code C separates a set S of size $|S| \leq d-1$ iff $\mathbf{H}(S)$ has rank $n-k-|S|$.*

Definition 2 **([7]).** *If $\mathbf{H}$ separates all sets S of size l for a fixed $l \leq \min\{d, n-k\}$, it is l-separating.*

If $\mathbf{H}$ is an l-separating parity-check matrix of the code C, based on $\mathbf{H}$, we can construct a parity-check matrix for any code punctured up to a fixed number l of symbols. $\mathbf{H}$ has two features:

- $\mathbf{H}$ can separate erasures from errors, since $\mathbf{H}$ has enough parity-check equations that do not check any erased symbols, and are sufficient to characterize the punctured code. It means that the punctured code, which is formed by deleting erased symbols, can be corrected errors by a sub-matrix of $\mathbf{H}$.
- In case $l \leq \min\{d, n-k\} - 1$, $\mathbf{H}$ has no stopping set of size l or less. For any pattern of l or fewer erasures, not only are there enough parity-check equations that do not check any of the erased symbols characterize the punctured code, but also there is a parity-check equation that checks exactly one of the erased symbols. It means that after all errors have been corrected, erasures can be recovered by the iterative decoding procedure.

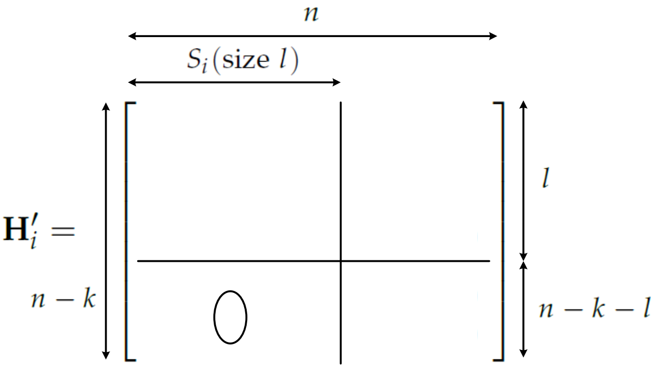


Figure 3. Independent-row separation.

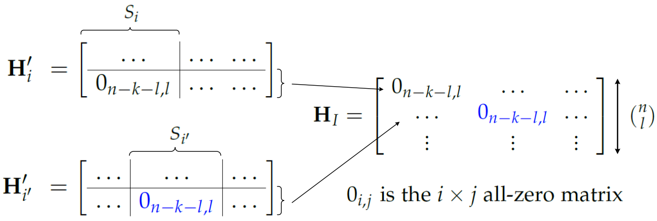


Figure 4. An l-separating matrix.

3.2 Separating Matrix

Let $\mathbf{H}'$ be a full rank parity-check matrix, $S_i \subseteq \{1,2,...,n\}$, in which $i = 1,2,\ldots,\binom{n}{l}$, be distinct subsets of $\{1,2,...,n\}$ of size l, For each i, it is trivial that $\mathbf{H}'_{S_i}$ has rank l ($l \leq \min\{d, n-k\} - 1$). By elementary row operations on $\mathbf{H}'$, we can obtain an $(n-k) \times n$ matrix $\mathbf{H}'_i$, for each $i = 1,2,...,\binom{n}{l}$, of rank $n-k$, such that its last $n-k-l$ rows have zeros in positions indexed by S_i, as given in Figure 3.

Let $\mathbf{H}_I$ be a matrix which rows is the union of sets of the last $n-k-l$ rows in $\mathbf{H}'_i$, for $i = 1,2,...,\binom{n}{l}$. $\mathbf{H}_I$ is an l-separating matrix of the code C, and it has at most $\binom{n}{l}(n-k-l)$ rows [7], as illustrated in Figure 4.

3.3 A More Efficient Separating Matrix

In this section, we propose a method that can construct an l-separating matrix with a smaller number of rows. This method implements the idea of *covering design* [13, 14]. Basically, given $1 \leq t \leq u \leq v$, a (v,u,t)

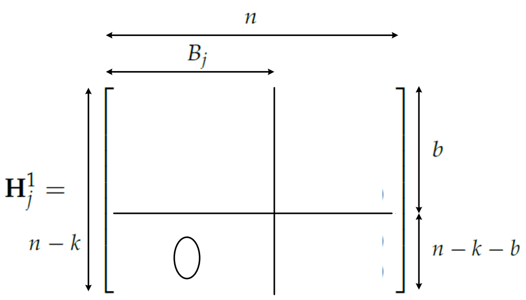


Figure 5. Row separation - Step 1.

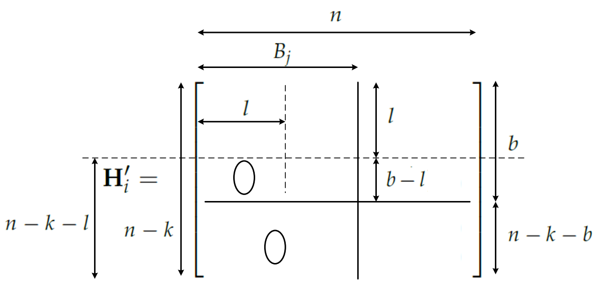


Figure 6. Row separation - Step 2.

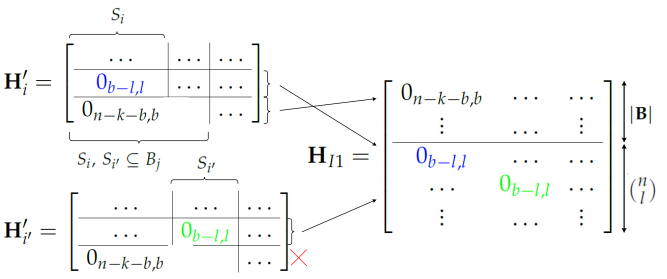


Figure 7. A more efficient l-separating matrix.

covering design is a collection of u-element subsets of $V = \{1,2,...,v\}$, called blocks, such that each t-element subset of V is contained in at least one block, e.g., $\{1,2\}$ is contained in $\{1,2,3\}$.

For our specific situation, consider an (n,b,l) covering design. Let $\mathbf{B} = \{B_i\}$ be a set of b-element subsets, $1 \le l \le b \le \min\{d, n-k\} - 1$, such that every l-element subset S_i is contained in at least one member of $\mathbf{B}$. Assign to each S_i, $i = 1,2...,\binom{n}{l}$, an element B_j of $\mathbf{B}$ such that S_i is contained in B_j. $\mathbf{H}'_{B_j}$ has rank b. For any B_j, by elementary row operations on $\mathbf{H}'$, we can obtain an $(n-k) \times n$-matrix of rank $n-k$ such that its last $n-k-b$ rows have zeros in positions indexed by B_j. After arranging columns, we obtain a matrix $\mathbf{H}_j^1$ with the following format (Step 1), given in Figure 5.

Consider the set S_i assigned to B_j, by further elementary row operations, $\mathbf{H}_j^1$ can be changed into a matrix such that rows $l+1, l+2, ..., b$ have zeros in positions indexed by S_i, and rows $b+1, b+2, ..., n-k$ have zeros in positions indexed by B_j. After column arrangement, we obtain a matrix with the following format (Step 2), given in Figure 6.

Following this method, if S_i and $S_{i'}$ belong to the same B_j, the last $n-k-b$ rows in $\mathbf{H}'_i$ and $\mathbf{H}'_{i'}$ are the same. Figure 7 shows that the matrix which rows is the union of the last $n-k-l$ rows in $H'_j, j = 1,2,...,|\mathbf{B}|$, and the rows $l+1, l+2, ..., b$ of $H'_i, i = 1,2,\ldots,\binom{n}{l}$, is an l-separating parity-check matrix of C. Let $B(n,b,l)$ denote the *minimum* size of $\mathbf{B}$, *i.e.*, $B(n,b,l) = \min|\mathbf{B}|$. This matrix has at most $(n-k-b)B(n,b,l) + \binom{n}{l}(b-l)$ rows. It is obvious to see that the upper bound on number of rows in this part is strictly smaller than in the previous one. In case $b = l$, two approaches are the same. For a given l, we can choose an appropriate b to achieve the best result.

4 Covering Design

Consider a (v,u,t) covering design, where $1 \le t \le u \le v$.

Example 1.

Given that $v = 8, u = 3, t = 2$. There are $\binom{8}{2} = 28$ subsets of 2-elements, and $\binom{8}{3} = 56$ subsets of 3-elements of $V = \{1,2,...,8\}$. However, we only need at most 21 subsets of 3-elements, *i.e.*, $\{1,2,3\}$, $\{1,2,4\}$, $\{1,2,5\}$, $\{1,2,6\}$, $\{1,2,7\}$, $\{1,2,8\}$, $\{1,3,4\}$, $\{1,3,5\}$, $\{1,3,6\}$, $\{1,3,7\}$, $\{1,3,8\}$, $\{1,4,5\}$, $\{1,4,6\}$, $\{1,4,7\}$, $\{1,4,8\}$, $\{1,5,6\}$, $\{1,5,7\}$, $\{1,5,8\}$, $\{1,6,7\}$, $\{1,6,8\}$, $\{1,7,8\}\}$, to form all 28 subsets of 2-elements. For example, based on the subset $\{1,2,3\}$, we can form $\{1,2\}$, $\{2,3\}$, $\{1,3\}$. Using 21 subsets of size 3 mentioned above, we can construct all 28 subsets of size 2.

The covering design problem has been investigated since many years ago. However, until now, there is no general *optimal* solution for all triples (v,u,t). In this section, we propose a covering design valid for all triples (v,u,t). This design is not optimal but it can give a *general solution* for the problem.

4.1 Approach 1

Firstly, we show that with at most $\binom{v-(u-t)}{t}$ subsets of size u, we can form all $\binom{v}{t}$ subsets of size t.

1) Take the first $u-t$ elements, *i.e.*, $\{1,2,...,u-t\}$, out of $V = \{1,2,...,v\}$.
2) The rest of the set is $\{u-t+1, u-t+2, ..., v-1, v\}$. Based on these elements, form all subsets of size t. The number of subsets is $\binom{v-(u-t)}{t}$.
3) Put the first $u-t$ elements into each subset of size t to have subsets of size u. With these $\binom{v-(u-t)}{t}$ subsets of size u, it is easy to see that we can form all $\binom{v-(u-t)}{t}$ subsets of size t.

4.2 Approach 2

By modifying *Approach 1*, we show that some u-element subsets can be merged to reduce $|\mathbf{B}|$.

1) Take the first $u-t$ elements, *i.e.*, $\{1,2,...,u-t\}$, out of $V = \{1,2,...,v\}$.
2) The rest of the set is $\{u-t+1, u-t+2, ..., v-1, v\}$. Based on these elements, form all subsets of size t and arrange them into columns based on the following rules:
   - Elements in each subset are arranged in ascending order, e.g., $\{1,2,3\}$.

- Subsets are arranged into columns. Subsets are in one column iff their first $t-1$ elements are the same (except the *special column* mentioned below). Hence, subsets in one column are different from each other only in the last element. The subset with the smaller last element will be listed above.
- *Special column*: In case $t \geq 2$, we arrange all subsets containing both element $v-1, v$ in a column and name it special column. It is easy to see that there are $\binom{v-(u-t)-2}{t-2}$ subsets in this column.

3) Put the first $u-t$ elements into each subset of size t to have subsets of size u.
4) If the number of subsets in the *longest column* is greater or equal to three and the *special column exists*, we can merge the last two subsets, which contain either $v-1$ or v, in each column (*except* the special column) into one, *i.e.*, the merged set, by this rule
   - Take the union of two subsets, *i.e.*, the size of the union subset is $u+1$.
   - Eliminate the first element of the union subset, *i.e.*, its size is now u.

We can merge the last two subsets in each column (except the special column) because: (a) The first $u-1$ elements in the last two subsets are also in another subset in the columns. Thus, any subset of size t formed by using these $u-1$ elements can be form by any other subset in the column; (b) Any subset of size t containing $\{1, v-1\}$ or $\{1, v\}$ can be formed by subsets in the *special column*.

Example 2.

Given that $v = 9, u = 5, t = 4$. Following the first three steps of Approach 2, we get the following result, as illustrated in Figure 8.

First, take $\{1\}$ out of the set. Following Step 2, form all subsets of size 4, and arrange them into columns. Put $\{1\}$ back into each subset of size 4 to have subsets of size 5. We denote subsets in boxes are subsets which can be merged.

The merged step (Step 4): For example, consider the first column, two subsets in the box are $\{1,2,3,4,8\}$ and $\{1,2,3,4,9\}$. First, take the union of the two, *i.e.*, $\{1,2,3,4,8,9\}$, and then eliminate the first element; thus, this results in $\{2,3,4,8,9\}$.

Therefore, Step 4 of Approach 2 gives the following result. It is easy to see that any subset of size 4 can be formed by subsets in Figure 9.

The number of reduced subsets is equal to the number of subsets that contain the elements $v-1$ or v. Thus, the number of reduced subsets is $\binom{v-(u-t)-2}{t-1}$. Therefore, with at most $\binom{v-(u-t)}{t} - \binom{v-(u-t)-2}{t-1}$ subsets of size u, we can form all $\binom{v}{t}$ subsets of size t.

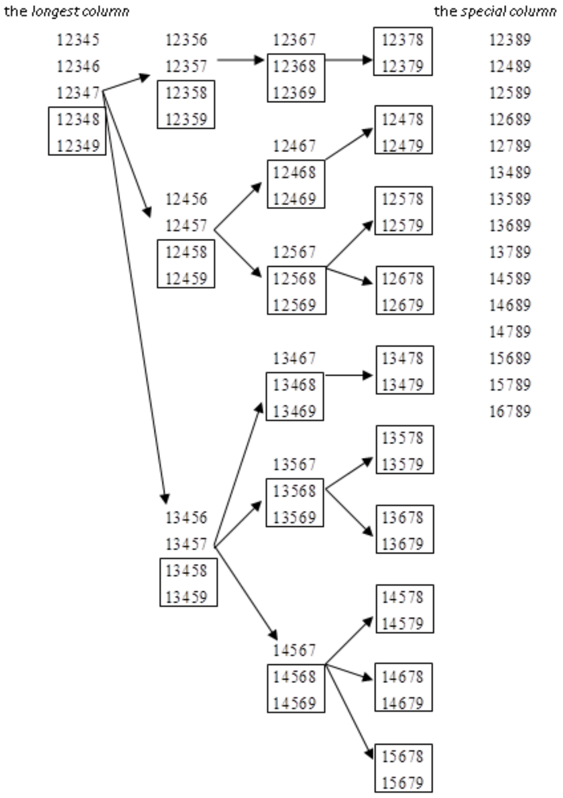


Figure 8. The first three steps of Approach 2.

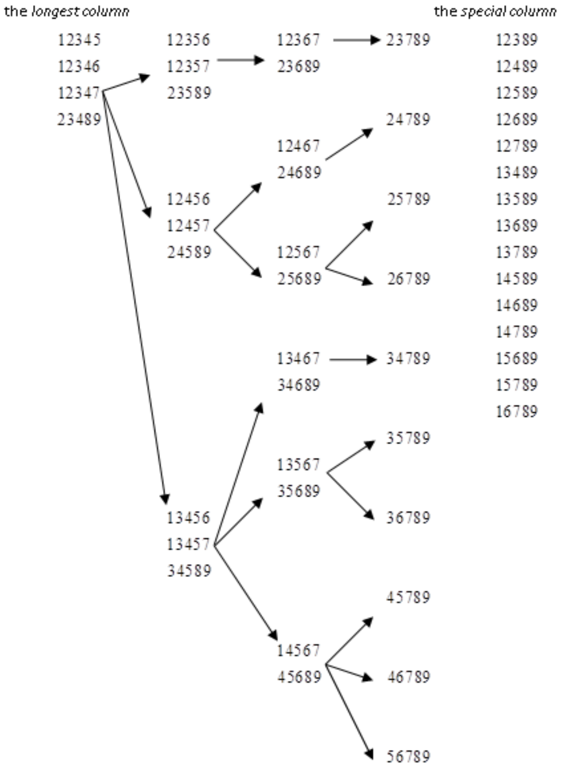


Figure 9. The last step of Approach 2.

5 Conclusion

This paper discusses how to ensure confidentiality and integrity for data in IoT systems. The paper focuses more on integrity which can be ensured via the implementation of error-correcting codes. Separating parity-check matrices are useful for decoding over channels causing both errors and erasures. We propose a way to build a separating parity-check matrix with a smaller set of rows. This method not only reduces both decoding complexity and memory storage. Besides, we also present a covering design. This design is not optimal but it gives a general solution for all triple (v, u, t).

Acknowledgments

The authors are supported by The University of Danang – University of Science and Technology through the grant T2019-02-13 and T2019-02-14.

References

[1] J. A. Goguen and J. Meseguer, "Security policies and security models," in *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 1982, pp. 11–11.

[2] T. M. Ngo and M. Huisman, "Complexity and information flow analysis for multi-threaded programs," *The European Physical Journal Special Topics*, vol. 226, no. 10, pp. 2375–2392, 2017.

[3] R. M. Roth, *Introduction to Coding Theory*. Cambridge University Press, 2006.

[4] S. Lin and D. J. Costello, *Error control coding*. Pearson Education India, 2004.

[5] F. J. Mac Williams and N. J. A. Sloane, *The theory of error-correcting codes*. North-Holland Publishing Company, 1977, vol. 16.

[6] S. A. Vanstone and P. C. Van Oorschot, *An introduction to error correcting codes with applications*. Norwell, MA: Kluwer, 1989, vol. 71.

[7] K. A. Abdel-Ghaffar and J. H. Weber, "Separating erasures from errors for decoding," in *Proceedings of the International Symposium on Information Theory*. IEEE, 2008, pp. 215–219.

[8] J. H. Weber, "Lecture notes: Error-correcting codes," Delft University of Technology, 2007.

[9] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information theory*, vol. 48, no. 6, pp. 1570–1579, 2002.

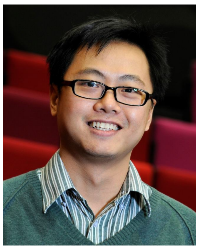
[10] M. Schwartz and A. Vardy, "On the stopping distance and the stopping redundancy of codes," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 922–932, 2006.

[11] J. H. Weber and K. A. S. Abdel-Ghaffar, "Results on parity-check matrices with optimal stopping and/or dead-end set enumerators," *IEEE Transactions on Information Theory*, vol. 54, no. 3, pp. 1368–1374, 2008.

[12] H. D. Hollmann and L. M. Tolhuizen, "On parity-check collections for iterative erasure decoding that correct all correctable erasure patterns of a given size," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 823–828, 2007.

[13] J. H. Dinitz and D. R. Stinson, *Contemporary design theory: A collection of Surveys*. John Wiley & Sons, 1992, vol. 26.

[14] [Online]. Available: http://www.ccrwest.org/cover.html



Tri Minh Ngo was born in Danang, Vietnam, in 1982. He received a B.E. degree from The University of Danang-University of Science and Technology, Danang, Vietnam, in 2005, and M.E. and Ph.D. degrees from Delft University of Technology and The University of Twente, The Netherlands, in 2009 and 2014, respectively. He is now a Lecturer at the Faculty of Electronic and Telecommunication Engineering, The University of Danang-University of Science and Technology, Danang, Vietnam.

Vien Nguyen-Duy-Nhat received the B.Eng. degree in Electronic Engineering from The University of Danang, University of Science and Technology, (DUT), Vietnam, in July 1997. In September 1997, he joined the Department of Electronics and Telecommunications Engineering, Danang University of Technology. In 2003, he received the M.Eng. degree in Electrical Engineering from Ho Chi Minh City University of Technology (HCMUT), Vietnam. And his Ph.D. degree from The University of Danang in 2017. His current research interests include next-generation wireless communications, wireless system design and optimization, Internet of Things.