Graduation Thesis Academic Year 2011

Group coding of RF tags

Faculty of Environment and Information Studies, Keio University
Yuki Sato

Abstract

RFID, radio frequency identification, is an automatic identification technology which identifies individual physical objects according to the unique IDs recorded by the RF tags attached on these objects. On the other hand, real-space objects often form a group. With RFID, we want to not only identify individual objects but also verify the integrity of the group of objects. In existing RFID systems, this "group verification" is done by looking up the list of grouped objects after the individual identification. This method requires some network connectivity to share the list of grouped objects and cannot be applied in the off-line environment. Against this problem, this thesis proposes the "group coding" of RF tags, which can check the integrity of the group in the off-line environment with writing the group related information into the memory of grouped RF tags. In addition, when some RF tags are missing from the group, the group coding can determine the information of RF tags missing from the group, the number of missing RF tags or the unique IDs of missing RF tags. Numerical simulation reveals that the determination of the number of missing RF tags can be done against 10 RF tags out of 20 RF tags within 0.5% error rate with writing 96-bit group related information into each RF tag's memory. It is also revealed that the determination of the unique IDs can be done against 64 missing RF tags out of 100 within 0.5% error rate with writing 840-bit group related information into each RF tag's memory. The validity of the theory and the numerical simulation is confirmed by the experimental evaluation using the implementation of the group coding on the commercial UHF-band RFID interrogator.

Key Words

1. RFID, 2. group of objects, 3. coding, 4. FEC

概要

RFID（radio frequency identification）は、個々の RF タグに割り当てられたユニークな ID を元に、電波を用いた読み取り機〜RF タグ間の通信により RF タグが貼付されたモノを個々に自動認識する技術である。これに対し、現実のモノは個々に存在するだけでなく、しばしばグループを形成する。RFID による自動認識においても、個々のモノの識別だけでなく、モノのグループの検証（グループからのモノの欠落の検知など）を行いたい。既存の RFID システムにおいてこのグループの検証は、ユニーク ID による個々のモノの識別の後にグループに属しているモノのリストを照会することによって行われている。この手法は、モノのリストを共有するためにネットワーク接続性が必要であり、オフライン環境ではグループの検証を行うことができない。これに対し、本論文では「RF タググループ符号化」を提案する。グループ符号化は、グループに関連するデータをグループに属す RF タグに書き込むことにより、オフライン環境であってもこのグループの検証を実現する。この場合、96bit のグループに関わる情報をグループに属する各タグにそれぞれ書き込むことにより、20 枚中 10 枚までの RF タグの欠落については 0.5%以下の誤り率で推定することができる。加えて、グループから RF タグが欠落していることを検知した場合、グループ符号化はグループから欠落した RF タグの枚数推定、または欠落した RF タグのユニーク ID の推定をすることができる。この場合、840bit の情報を各タグに書き込むことで、100 枚中 64 枚までの RF タグの欠落について、欠落した RF タグの 96bit のユニーク ID を 0.5%以下の誤り率で推定することができる。また、市販の UHF 帯 RFID リーダライタを用いた実験により、理論とシミュレーションの正当性を示す。

キーワード

1．RFID，2．モノのグループ，3．符号化，4．FEC

Contents

List of Figures

Chapter 1

Introduction

1.1 Background of the research

RFID, radio frequency identification, is an automatic identification technology which is composed of RF tags, interrogator and associated information system. Because an interrogator collects RF tags' information via RF communication, RFID can identify not only line-of-sight objects but also non-line-of-sight objects. With RF multiple access technique, RFID can identify many objects swiftly. We also can write data to each RF tag's memory. These features make RFID different from other existing automatic identification technologies such as bar-code.

RFID identifies a physical object using its unique ID recorded in the RF tag. On the other hand, in many business processes, not only identity of individual objects but also the integrity of a group of objects needs to be verified. Goods in a shipping container are typical practice of a group of objects. Automatic verification of the integrity can reduce the time and cost of such process. Figure 1.1 shows an example usage scenario for group verification system in an RFID portal. A group of RF tagged objects on a pallet passes through the portal. When all RF tags belonging to the group are verified to exist, the operator can let the pallet go. On the other hand, when it is found that some objects are missing from the group, the group verification system alerts the operator and executes another round of RF tag inventory. If the retrial inventory fails, the group verification system reports the missing of objects to the operator. The information about the missing objects (RF tags) will be useful for the operator to search missing objects or do some other operations.

EPCIS [1] defines events relating a group of objects as "aggregation event"

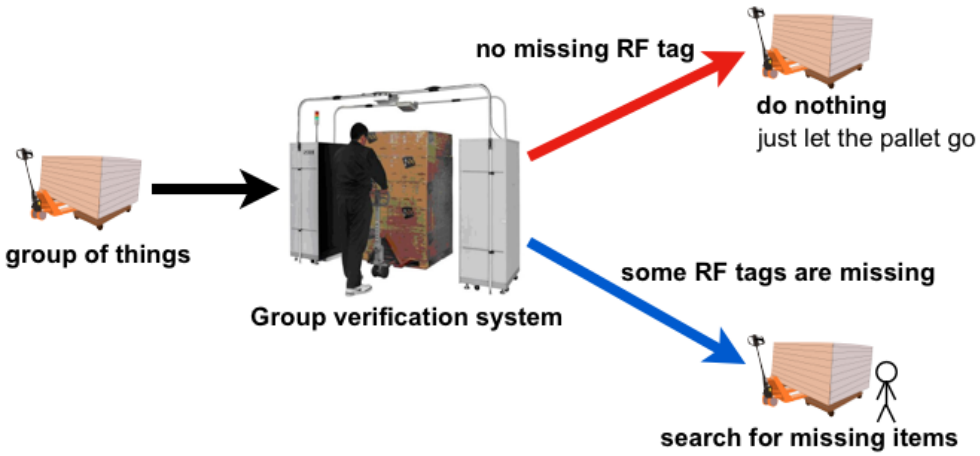

Figure 1.1: Usage scenario of group verification system

and "quantity event", but how to establish such group of objects is out-of-scope of the standard. In existing RFID systems, such group verification is done by looking up the list of grouped objects, such as the ASN (Advanced Shipment Notification), after individual identification. This type of verification usually requires a network connection to retrieve the list of grouped objects. However, verification of the integrity of a group is required not only in a network-reachable environment but also in an off-line environment. This motivated us to establish a method to verify the integrity of a group of objects and determine the information of missing objects without a network connection.

1.2 Nomenclature

Interrogator A device which identifies RF tags and collects their unique IDs. It also can read/write some information from/into RF tags' memory.

Inventory The detection of RF tags and collection of the unique IDs of these RF tags by an interrogator.

RF tag A small electronic device which responds to interrogators' commands (Figure1.2). Each RF tag records its own unique ID in its memory. In addition to

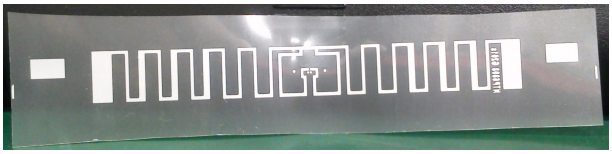

Figure 1.2: RF tag example

an unique ID, some of RF tags can record additional information in their memory. RF tags are classified into passive tags, semi-passive tags and active tags.

Unique ID An identifier uniquely assigned to each RF tagged objects, e.g. 96-bit EPC (Electronic Product Code).

User memory One kind of memory banks of RF tags where the user can write any information.

XOR The exclusive OR operation or its result. In this thesis, the XOR represents the bit-by-bit exclusive OR operation against two bit strings and is represented by "$\oplus$" in equations.

1.3 Structure of the thesis

This thesis is organized as follows. The chapter 2 introduces some related works on the verification of a group of objects and our approach to this problem. In the chapter 3 and 4, the theory and evaluation of our proposing "group coding" are introduced. The chapter 5 concludes this thesis.

Chapter 2

Prior works and our approach

This chapter introduces some prior works on the group verification and our approach to this problem.

2.1 Prior works

2.1.1 Yoking Proof / Coexistence Proof / Grouping Proof

Juels [2] introduced a yoking proof of RF tags, which confirms that some RF tags are read at the same time. The yoking proof is a research which focuses on the security issue of RFID. It prevents that the malicious reader makes a false proof, which means the reader says RF tags are read in spite of that the RF tags are actually not read. There are some researches [3, 4, 5, 6, 7] which improve the problem of the yoking proof or generalize the yoking proof to a group of RF tagged objects.

The yoking proof and the related researches use off-line or on-line verifier to make a proof. These verifiers and RF tags must previously share the secret key.

2.1.2 Statistical analysis of inventory results from multiple interrogators

Inoue [8] introduced a systematic scheme which detects failure to read of RF tags based on a statistical analysis of inventory results from multiple interrogators. The networked interrogators are located at the different place, and the inventory results

of these interrogators are gathered to a database. From the collected data, the proposed system can statistically detect inventory failures of RF tags. In other words, the system can judge whether the RF tags which are not read by one interrogator is actually missing at that time or not read because of just a inventory failure.

2.1.3 Integrity-check using total weight of grouped objects

Potdar [9] proposed an integrity-check method which uses total weight of grouped objects and inventory result of RF tags. The weight of grouped objects attached RF tags is measured simultaneously with the inventory of RF tags, then the measured weight is compared to the weight previously measured. With combining the inventory result of RF tags and the result of this comparison, the proposed system can check the integrity of the group, whether some RF tags are actually missing or there is a inventory failure.

2.2 Our approach

Generally, the verification of the integrity of a group of objects is required not only at the place where the group is established but also at the different place, e.g. a group of baggage is created by a consignor and verified by a consignee. All of the above-mentioned prior works require some network connectivity between these places to verify the integrity of a group. Even the original yoking proof, which uses off-line verifier, requires the network connectivity to share the secret key.

It is considered that there is similarity between this group verification problem and the packet-level forward error correction in the erasure communication channel [10, 11, 12, 13]. With treating the inventoried ID of an object as a packet, the missing of objects can be treated as the packet loss in an erasure communication channel. Therefore, the proposal is named “group coding” of RF tags. The group coding “encodes” RF tags to create a group and “decodes” them to verify the integrity of the group and determine the information about missing RF tags. The group coding adds the redundant information, which calculated in the group encoding procedure and used in the group decoding procedure, in the user memory of grouped RF tags, so the group decoding can be done without any network connection.

On the other hand, there are some points different between the group coding and packet-level coding. Generally, each encoded packet has a serial number as the header information, so decoder can sort the received packets before the

computation of decoding. However, the reading order of RF tags is principally random. In addition, the size of redundant information written into each RF tag is limited to the size of RF tag's memory. The group coding must consider these constraints.

2.3 Conclusion of this chapter

This chapter introduced three prior works on the group verification problem. All of them require some network connectivity and cannot be applied in the off-line environment. On the other hand, our proposing "group coding" of RF tags writes group-related information to grouped RF tags' user memory and eliminates external database or verifier. The group-related data is calculated from the information of grouped RF tags, and this redundant information is used to determine the information of missing RF tags at the verification of the integrity of a group.

The following chapter introduces the theory of the group coding and its evaluation. Two group coding schemes are introduced in the chapter 3 and 4, respectively. The first one can determine the number of RF tags missing from the group. The second one can determine the unique IDs of RF tags missing from the group.

Chapter 3

Missing number determination by the matrix's rank

This chapter introduces a group coding scheme which determines the number of RF tags missing from a group. This group coding scheme uses the rank of a matrix to determine the number of missing RF tags.

3.1 Theory

In this chapter, the theory of the determination of the number of missing RF tags is introduced. The section 3.1.1 introduces the fundamental of the group coding, the detection of the missing of RF tags using a "group ID". Based on this section, the section 3.1.2 introduces the theory how the group coding determines the number of missing RF tags. The section 3.1.3 shows the theoretical analysis on the relationship between the design of the parameter of the group coding, a "group generation matrix", and the performance of the determination.

3.1.1 Detection of the missing of RF tags using a group ID

The group coding checks the integrity of a group with the check sum technique. The check sum of a group is calculated as the XOR of all hashes of RF tags' unique IDs belonging to the group, and the group coding system writes the check sum to these RF tags' memory. This check sum also has a role as an identifier of the group, it means it can be judged whether an RF tag belongs to the group or not by reading its memory, so it is called "group ID".

The procedure to create a group is as follows.

1. The interrogator collects the unique IDs of all RF tags belonging to the group which the operator want to create.

2. Group coding software calculates the hash of each unique ID with a hash function. The hash function is needed to avoid situation where unique IDs belonging to the group are sequential.

3. A group ID is calculated as the XOR of all hashes.

4. The calculated group ID is written to each RF tag's user memory, so each RF tag records both its own unique ID and the group ID.

By writing multiple group IDs to user memory, a single RF tag can belong to multiple groups.

The procedure to verify the integrity of a group is as follows.

1. The interrogator collects the unique IDs of all detected RF tags and the group IDs written in the user memory of these RF tags.

2. The collected unique IDs are sorted by group ID.

3. Group coding software calculates the hash of each unique ID which belongs to a group. The hash function must be the same to the one used in the group creation procedure.

4. The XOR of all hashes in the group is calculated. The result is referred to as the calculated group ID.

5. Group coding software compares the calculated group ID and the written group ID. If there is no RF tag missing from the group, the two group IDs will be the same. On the other hand, if these group IDs are different, there are some RF tags missing from the group.

These two procedures are also shown in Figure3.1.

This method, which checks the integrity of a group, may incur a "false positive" which means that the integrity of the group is verified in spite of missing RF tags. A false positive occurs when the XOR of the hashes of missing RF tags' unique IDs is coincidentally equal to zero. The probability of a false positive analytically is computed and verified it via numerical simulation.

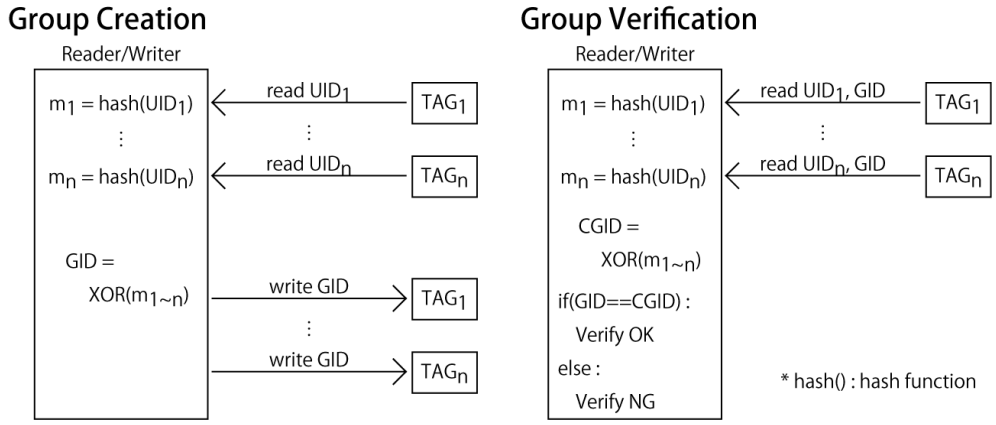

Figure 3.1: Overall procedure of the group coding which checks the integrity of a group

Naturally, when only one RF tag is missing, the calculated group ID will give a false positive only when the hash of the missing RF tag

$$m_j = 0. \tag{3.1}$$

Suppose the hash function generate uniformly distributed hashes, this probability P_1 is given by the following equation for L bits of an hash.

$$P_1 = \frac{1}{2^L} \tag{3.2}$$

When two RF tags, j and k, are missing, a false positive occurs when

$$m_j \oplus m_k = 0, \tag{3.3}$$

that is, when

$$m_j = m_k. \tag{3.4}$$

The probability of this condition, P_2, is

$$P_2 = \frac{1}{2^L}. \tag{3.5}$$

Likewise, the probability of false-positiveness against n missing RF tags, P_n, is generally given by the following equation.

$$P_n = \frac{1}{2^L} \tag{3.6}$$

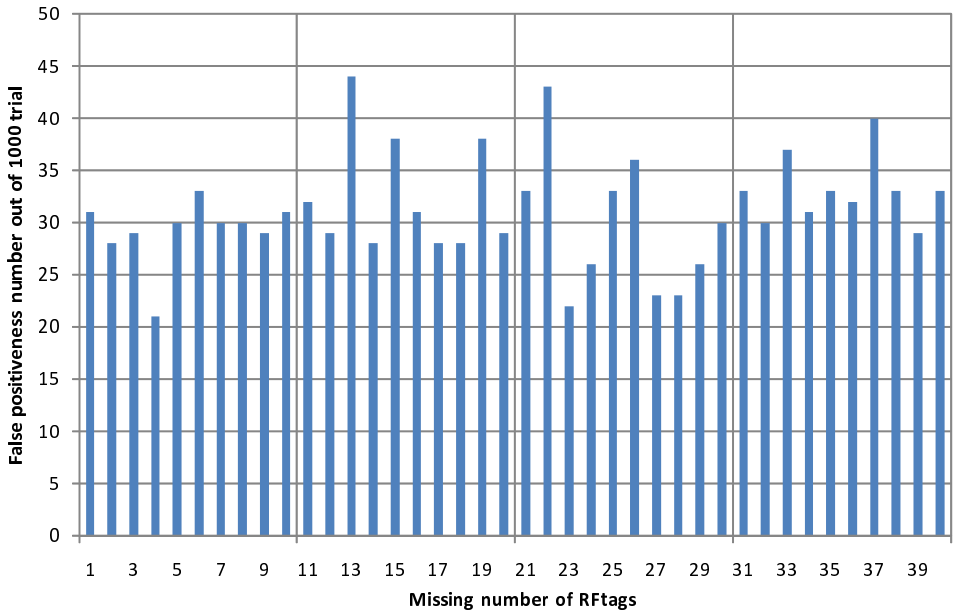

Figure 3.2: Probability of false-positiveness of fundamental group coding

For example, with the hash function which generates 16-bit hash, the probability of false positive becomes $\frac{1}{2^n} = \frac{1}{65536}$. This probability P_n becomes negligible with a sufficiently large L, the bit-length of hashes.

This probability of a false positive is verified with a numerical simulation. In this simulation, a unique ID of each RF tag which belongs to a group is 96-bit and generated randomly. The simulator removes a designated number of RF tags and computes the XOR of all hashes of removed RF tags' unique IDs. If the result of the computation is zero, it is recorded as a false positive. This simulation process is executed 1000 times for different numbers of missing RF tags (the number varying from 1 to 40). Because CRC-5, which generates 5 bits of hash, is used as the hash function for this simulation, the probability of a false positive is theoretically $\frac{1}{2^5} = \frac{1}{32}$. Therefore, it is expected that a false positive occurs about 31 times out of 1000 trials on average. The result of this simulation is shown in Figure3.2, and this result agrees well.

3.1.2 Determination of the number of missing RF tags using multiple group IDs

The determination of the number of RF tags missing from a group is done by multiple group IDs. The procedure to create a group is referred to as "group encoding". The procedure to verify the integrity of a group and determine the number of missing RF tags is referred to as "group decoding".

Group encoding

Group encoding starts by dividing the main-group into multiple sub-groups. This division to sub-groups can be expressed with a matrix equation. The division of a main-group of m RF tags into n sub-groups is expressed with the following equation. m_i is the hash of i'th RF tag in main-group, and g_i is the group ID of i'th sub-group.

$$\begin{Bmatrix} g_1 \\ \vdots \\ g_n \end{Bmatrix} = \begin{bmatrix} G \end{bmatrix} \begin{Bmatrix} m_1 \\ \vdots \\ m_m \end{Bmatrix} \tag{3.7}$$

In this equation, G is an m-column, n-row matrix composed of 0 and 1, that defines the structure of sub-groups. Because G is similar to the generation matrix in coding theory, this matrix is named the "group generation matrix".

For example, suppose that 5 RF tags are apportioned to 3 sub-groups. Sub-group 1 contains RF tags 1, 2, 3, and 4, sub-group 2 contains RF tags 4 and 5, and sub-group 3 contains RF tags 1, 2, and 5 as shown in Figure3.3. The Group ID of each sub-group is calculated as follows.

$$g_1 = m_1 \oplus m_2 \oplus m_3 \oplus m_4 \tag{3.8}$$
$$g_2 = m_4 \oplus m_5 \tag{3.9}$$
$$g_3 = m_1 \oplus m_2 \oplus m_5 \tag{3.10}$$

In this case, the group generation matrix G becomes

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}. \tag{3.11}$$

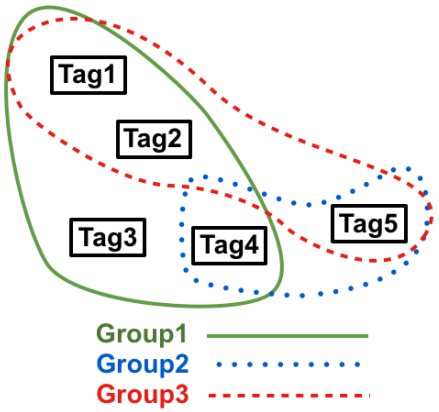

Figure 3.3: Example of division into sub-groups

Group decoding

When some RF tags are missing from the main-group, some sub-groups lose their integrity. Equation 3.7 can be rearranged to express these incomplete sub-groups as follows.

$$\left\{ \frac{g_c}{g_e} \right\} = \left[\frac{G_{cr} \mid G_{cm}}{G_{er} \mid G_{em}} \right] \left\{ \frac{m_r}{m_m} \right\}, \tag{3.12}$$

In this equation, g_c represents the array of group IDs of sub-groups which preserve their integrity, and g_e represents that of sub-groups which lose their integrity. Similarly, m_r represents the array of hashes of RF tags read by the interrogator, and m_m represents that of RF tags missing from the group. The group generation matrix G is separated to 4 partial matrices, G_{cr}, G_{er}, G_{cm} and G_{em}, in accordance with these 4 arrays, g_c, g_e, m_r and m_m.

In Eq.3.12, of course, g_c, G_{cr} and m_r are known. G_{cm} can be treated as a zero matrix because the false-positiveness of the fundamental group coding can be practically ignored as explained in the previous section. Array g_e and partial matrix G_{er} are available because remaining RF tags belonging to these sub-groups records these information. Therefore, G_{em} and m_m are the only unknowns in this equation. The product of these two unknown terms, $G_{em}m_m$, can be calculated as follows.

$$G_{em}m_m = g_e \oplus G_{er}m_r. \tag{3.13}$$

Because $G_{em}m_m$ is an array of linear combinations of the mutually independent hashes of the missing RF tags' unique IDs, the number of missing RF tags can be

determined as the rank of $G_{em}m_m$.

Let us take a group of RF tags whose group generation matrix is given by Eq.3.11 as an example. When RF tags 2 and 4 are missing from this group, all sub-groups lose their integrity. This situation is shown in the following equation.

$$\left\{ \begin{array}{c} g_1 \\ g_2 \\ g_3 \end{array} \right\} = \left[\begin{array}{ccc|c} 1 & 1 & 0 & \\ 0 & 0 & 1 & G_{em} \\ 1 & 0 & 1 & \end{array} \right] \left\{ \begin{array}{c} m_1 \\ m_3 \\ m_5 \\ \hline m_m \end{array} \right\} \tag{3.14}$$

$G_{em}m_m$ is computed as follows.

$$G_{em}m_m = \left\{ \begin{array}{c} g_1 \oplus m_1 \oplus m_3 \\ g_2 \oplus m_5 \\ g_3 \oplus m_1 \oplus m_5 \end{array} \right\} = \left\{ \begin{array}{c} m_2 \oplus m_4 \\ m_4 \\ m_2 \end{array} \right\} \tag{3.15}$$

In this case, the rank of this array is 2, and it is equal to the number of missing RF tags.

3.1.3 The relationship between determination performance and group generation matrix

As mentioned above, the accurate determination of the group coding can be done only if the rank of the matrix $G_{em}m_m$ is equal to the number of missing RF tags. m_m is an array of mutually independent hashes of missing RF tags' unique IDs, so the rank of $G_{em}m_m$ is equal to that of G_{em}. There are two cases where the rank of this G_{em} matrix is different from the number of missing RF tags.

- **Case 1:** When G_{em} does not have enough rows, these rows can not be decomposed to each missing RF tag's information by a linear algebra. The rank of G_{em} becomes smaller than the number of missing RF tags and the determination of the group coding gives a smaller number than the number of actually missing RF tags. For example, suppose the main-group which contains 5 RF tags and whose group generation matrix and sub-groups' group IDs are represented by the following equation.

$$\left\{ \begin{array}{c} g_1 \\ g_2 \\ g_3 \end{array} \right\} = \left[\begin{array}{ccccc} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \end{array} \right] \left\{ \begin{array}{c} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{array} \right\} \tag{3.16}$$

When RF tags 2, 3, and 5 are missing from this group, this equation is changed as follows.

$$\begin{Bmatrix} g_1 \\ g_2 \\ g_3 \end{Bmatrix} = \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{array}\right] \left\{\begin{array}{c} m_1 \\ m_4 \\ \hline m_2 \\ m_3 \\ m_5 \end{array}\right\} \tag{3.17}$$

In this case, the rank of G_{em} is two although three RF tags are actually missing.

- **Case 2:** A sub-group cannot be detected if all RF tags belonging to this sub-group are missing. When all sub-groups which an RF tag belongs to are not detected, the group generation matrix loses the column of this missing RF tag. In this case, the rank of G_{em} becomes smaller than the number of missing RF tags and the determination fails because of the lack of the information about that missing RF tag. For example, suppose that a group contains 5 RF tags and whose group generation matrix and sub-groups' group IDs are represented by following equation.

$$\begin{Bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{Bmatrix} \tag{3.18}$$

When RF tags 1, 2, and 5 are missing from this group, this equation is changed as follows.

$$\left\{\begin{array}{c} g_3 \\ \hline g_2 \\ g_4 \end{array}\right\} = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{array}\right] \left\{\begin{array}{c} m_3 \\ m_4 \\ \hline m_2 \\ m_5 \end{array}\right\} \tag{3.19}$$

The sub-groups g_1 and g_5 cannot be detected because all RF tags belonging to these sub-groups, RF tags 1, 2 and 5, are missing. In this case, the rank of G_{em} is two although the number of missing RF tags is actually three.

The probability of these problems can be reduced by increasing the number of sub-groups. G_{em} is a partial matrix of the group generation matrix G, so the

probability of these problems is affected by this matrix. By increasing the rank of the group generation matrix, the probability of the first determination failure can be reduced, and this requirement can be acheived with a sparse group generation matrix, which contains many 0s. The second determination failure can be mitigated with a sufficient spread of the information of each RF tag to other RF tags, and this requirement can be achieved with a dense group generation matrix, which contains many 1s. However, the wide spreading of sub-group actually results in the loss of the rank of group generation matrix. Therefore, the group generation matrix should be designed to balance these two trade off requirements.

3.2 Evaluation

This section shows the evaluation of the missing number determination by the group coding introduced in the section 3.1. The evaluation is done by the numerical simulations and the experiments with the implementation of the group coding.

3.2.1 Numerical simulation on the determination performance

The performance of the above-mentioned group coding is evaluated with numerical simulations. The general procedure of the simulations is as follows.

1. The simulator prepares 20 virtual RF tags. Each RF tag has 96-bit EPC generated randomly as the unique ID.

2. The group of these RF tags is created with the group encoding procedure. A generation matrix for an LDPC [14], Low Density Parity Check code, is used as the group generation matrix. This matrix is randomly generated under different conditions for two variables, k and j. k signifies the number of RF tags belonging to each sub-group. j signifies the number of sub-groups which each RF tag belongs to. Group IDs are calculated from 16 bits of hashes generated by CRC-16.

3. Randomly selected n RF tags are removed from the group. n is a number between 1 and 19.

4. The number of missing RF tags is determined with the group decoding procedure.

Table 3.1: Conditions of the numerical simulation : Pattern I

Condition	**Value**
number of RF tags belonging to each sub-group (k)	4 (fixed)
number of sub-groups which each RF tag belongs to (j)	1-6

Table 3.2: Conditions of the numerical simulation : Pattern II

Condition	**Value**
number of RF tags belonging to each sub-group (k)	2-6
number of sub-groups which each RF tag belongs to (j)	4 (fixed)

5. This procedure is repeated 10000 times for every combination of conditions, then the error rate of determination is calculated, counting the frequency of the determined number differing from the actual number of missing RF tags.

The simulation was done with two patterns of the conditions of group generation matrix.

The first pattern is shown in Table 3.1. The number of RF tags belonging to each sub-group, k, is fixed at 4, and the number of sub-groups which each RF tag belongs to, j, varies between 1 and 6. From the above-mentioned analysis, it is expected that the accuracy of the determination improves with increasing j.

The result of this simulation is shown in Figure3.4. This chart shows that the accuracy of determination increases as j increases. This result agrees with the theory mentioned in the section 3.1.3 that the performance of general group coding becomes better with increasing the number of sub-groups. Figure 3.5 shows the maximum number of missing RF tags which can be determined within some tolerance. For example, with 4 sub-groups to each RF tag, all RF tags record 64-bit additional information in the user memory, the number of missing RF tags can be determined up to 4 within 0.5% error rate. On the other hand, with 6 sub-groups to each RF tag, all RF tags record 96-bit additional information, the number of missing RF tags can be determined up to 10 within the same error rate.

The second pattern of the conditions is shown in Table 3.2. In this pattern, k varies between 2 and 6, and j is fixed at 4.

The result of the simulation is shown in Figure3.6. The determination accuracy of general group coding improves by decreasing the number of RF tags in

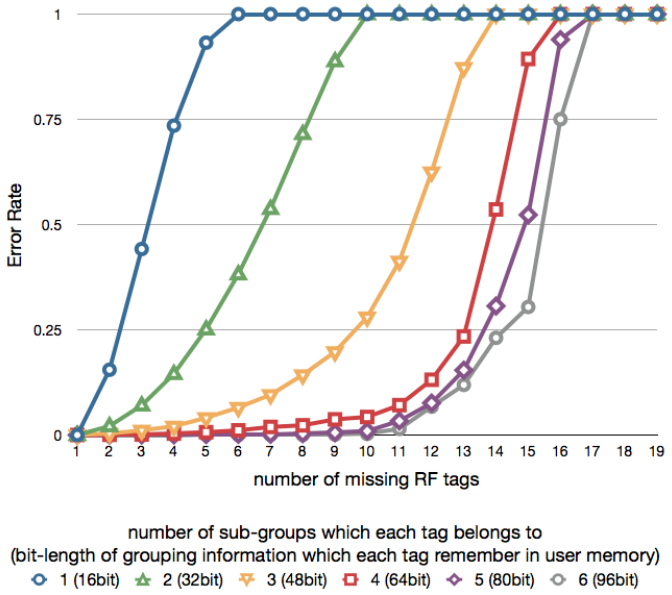

Figure 3.4: Error rate of the determination of the number of missing RF tags by general group coding : Pattern I

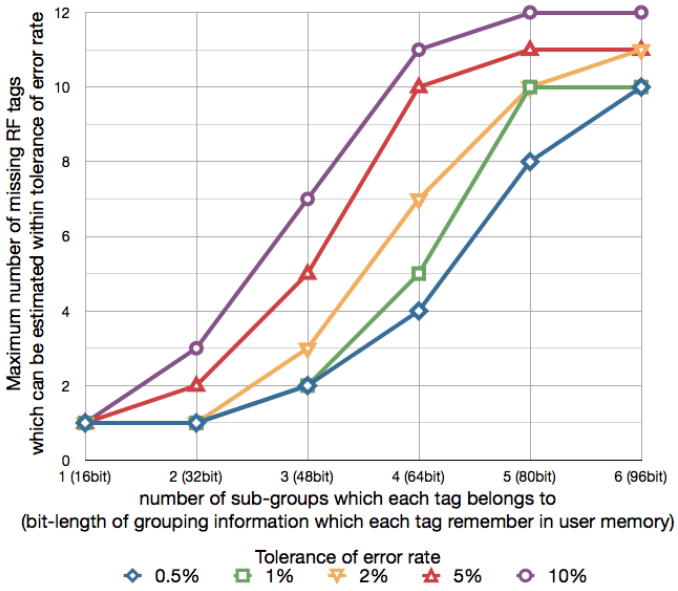

Figure 3.5: Maximum number of missing RF tags which can be determined within tolerance of error rate : Pattern I

each sub-group from 6 to 3. If there are 2 RF tags in each sub-group, however, the performance degrades. Figure 3.7 shows the subset of this result with the horizontal axis of k. From these charts, it is shown that the performance of general group coding becomes the highest at $k = 3$ or $k = 4$ under the conditions of this simulation. This result also agrees with the theory mentioned in the section 3.1.3 that a too-sparse or too-dense matrix is not suitable for the group generation matrix and there is an optimal point of the matrix's density which maximize the performance of general group coding.

3.2.2 Experimental evaluation

The afore-mentioned group coding is implemented using a commercial UHF-band RFID interrogator KU-U1601 (Figure3.8) provided by Panasonic System Networks Co., Ltd. This interrogator conforms to the EPCglobal UHF Class-1 Generation-2 air protocol [15]. In our implementation, the calculation of group coding is computed by the computer connected to the interrogator, and the program for the computation of group coding is written in Java.

The procedure of the experiment is as follows.

1. 20 physical RF tags whose unique IDs (96-bit) are generated randomly is placed in front of the antenna of the interrogator.

2. The group of these physical RF tags is created by the group encoding procedure. The detailed conditions of the group encoding are the same as those of the numerical simulation shown in TABLE 3.1. This process includes writing data to physical RF tags' user memory.

3. N RF tags are physically removed from the group randomly. N is the maximum number of missing RF tags which can be determined within a specified tolerance of error, and it is given by Figure3.5. For example, if the tolerance of error is 0.5% and j is 6, the value of N is 10, so 10 RF tags are removed from the group. The case of 0.5% and 10% determination error are examined.

4. The number of missing RF tags is determined by the group decoding procedure.

5. This procedure is repeated 10 times for each j, and the average from the results is calculated.

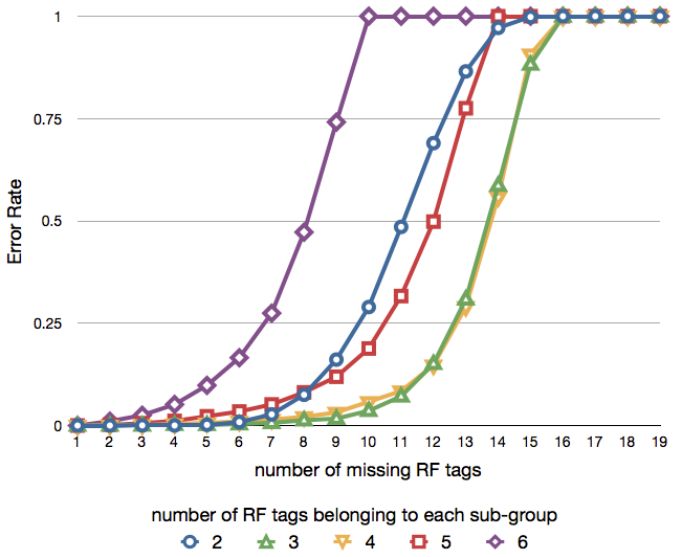

Figure 3.6: Error rate of the determination of the number of missing RF tags by general group coding : Pattern II

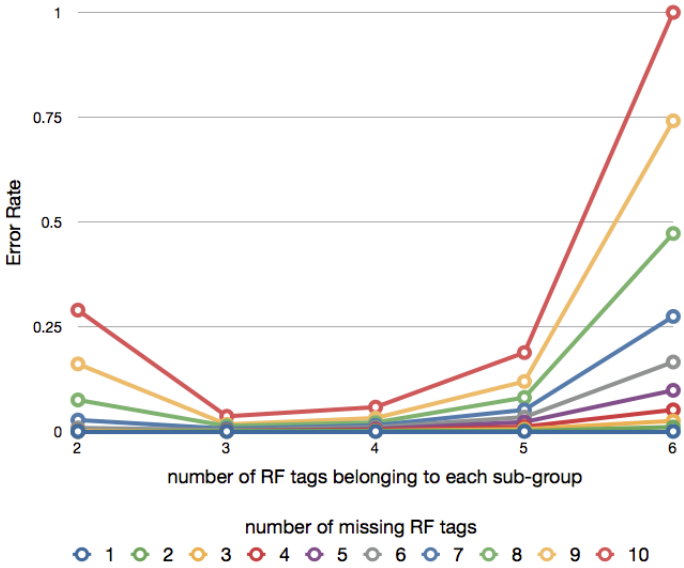

Figure 3.7: Error rate of the determination of the number of missing RF tags by general group coding : Pattern II (horizontal axis of k)

Figure 3.8: KU-U1601 UHF-band RFID interrogator

When the tolerance of error is 0.5%, it is expected that the result of this experiment will agree exactly with the numerical simulation shown in Figure3.5 because 0.5% error rate is small against 10-time trials. On the other hand, when the tolerance of error is 10%, it is expected that the result of the experiment is equal to or less than the number of missing RF tags. Figure 3.9 shows the result for the error tolerance of 0.5%. Figure 3.10 shows the result for the error tolerance of 10%. When the error tolerance is 0.5%, the result of the experiment matches that of the numerical simulation. When the error tolerance is 10%, the result of the experiment is a little different where the number of sub-groups which each RF tag belongs to is 2 and 3. Both of them affirm the expectation and prove that the group coding works on real RFID system.

3.3 Conclusion of this chapter

This chapter introduced the group coding which determines the number of RF tags missing from the group. The determination accuracy of 10 RF tags' missing out of 20 is 99% with writing 96-bit group related data to each RF tag's memory. The performance of the group coding, the accuracy of the missing number determination, can be adjusted by controlling the size of the group related data. The performance increases in proportion to the number of sub-groups. When the high performance is not required, the memory consumption for each RF tag can be reduced. In addition, even if the memory consumption of RF tags is the same, the coding strength of general group coding can be affected by the structure of group codes. The parameter of group coding should be determined with considering these characteristics. By the experimental evaluation using the implementation of

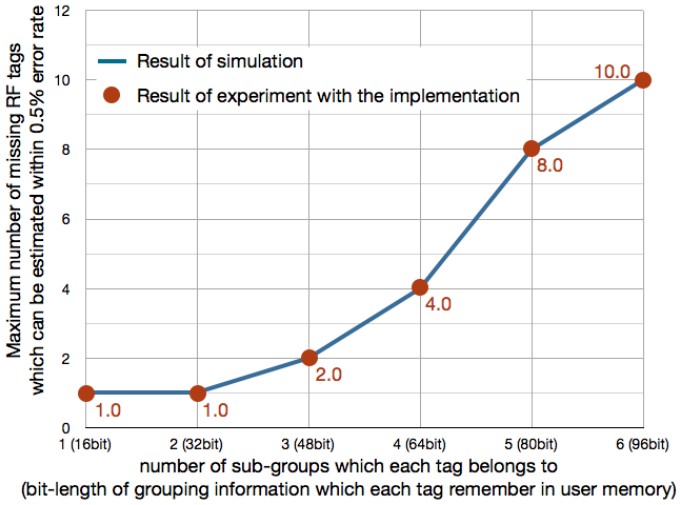

Figure 3.9: Comparison between the results of simulation and experiment (0.5% error rate)

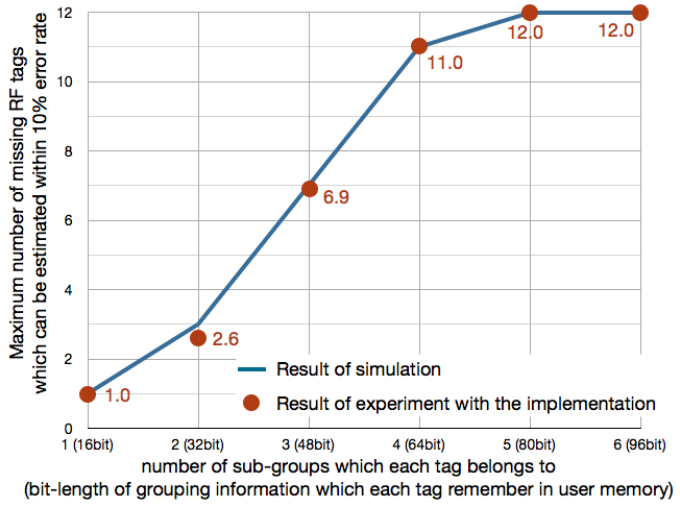

Figure 3.10: Comparison between the results of simulation and experiment (10% error rate)

the group coding, it is confirmed that the missing number detection works well on the commercial RFID interrogator.

Chapter 4

Missing ID determination by an iterative decoding

This chapter introduces an extended scheme of the group coding introduced in the chapter 3. The extended scheme of the group coding can determine the unique IDs of RF tags missing from the group. This scheme is referred to as the extended group coding.

4.1 Theory

The original group coding introduced in the chapter 3, which determines the number of missing RF tags, is simple and easily-computed, but cannot determine the missing unique IDs. The principal reason, which impedes us to determine the missing IDs, is the unavailability of the topology (edges) of Tanner graph, which relates the unique IDs and sub-groups. In the extended group coding scheme, the graph edge information is added to RF tags' memory in addition to group IDs and its unique ID. This section describes the theory of this extended group coding, how the extended group coding determines the unique IDs of missing RF tags.

4.1.1 Group encoding and decoding in the extended group coding

Just like the original group coding in the chapter 3, the procedures to create a group and to determine the missing unique IDs are referred to as the group encoding and decoding, respectively.

Group encoding

Similar to the original group coding, the extended group coding use a group generation matrix to calculate group IDs. The entire group is logically split into mutually overlapped sub-groups. Group IDs are calculated as the XOR of unique IDs which we want to retrieve, so the bit-length of one group ID is the same as the bit-length of unique ID which we want to retrieve. It is presumed that the length of unique IDs are consistent in a group.

In addition to the group ID, the extended group coding assigns "short ID" to all RF tags belonging to a group. Each RF tag in a sub-group records the group ID of this sub-group and the short IDs of other RF tags belonging to this sub-group. Note that the short ID of its own is not required to be stored in its memory. Since these short IDs are used to re-establish the Tanner graph of the group, short IDs are required to be unique only within the group. Accordingly, the bit-length of a short ID can be shorter than that of the unique ID which we want to retrieve. For example, 8-bit short IDs are generally enough to distinguish less than 255 RF tags. Since these short IDs do not have to be recalculated in the group decoding procedure, any data, for example a portion of factory programmed Tag ID [16], can be used as the short IDs. This combination of a group ID and short IDs is a set of the information required of each RF tag to record per sub-group. Each RF tag must record as many sets of information as the number of sub-groups which it belongs.

For example, suppose a group containing 5 RF tags. There are 3 sub-groups to represent the whole group. Sub-group 1 contains RF tag 1, 2, 3 and 4, sub-group 2 contains RF tag 3, 4 and 5, and sub-group 3 contains RF tag 1, 2 and 5. g_x represents the x'th sub-group's group ID and s_x represents the short ID of x'th RF tag in this example. In this group, RF tag 1 records the information combination of sub-group 1, g_1, s_2, s_3 and s_4, and that of sub-group 3, g_3, s_2 and s_5, and RF tag 1 does not record its own short ID which is s_1. Likewise, other RF tags also records the information of sub-groups which they belong to. All the information recorded in these RF tags in this example is shown in Figure4.1.

Group decoding

The group decoding in the extended group coding starts with the integrity-check of each sub-group by its group ID. The information of each incomplete sub-group, which is composed of the linear combination of missing RF tags' unique IDs and short IDs of each missing RF tag, is collected. The linear combination of

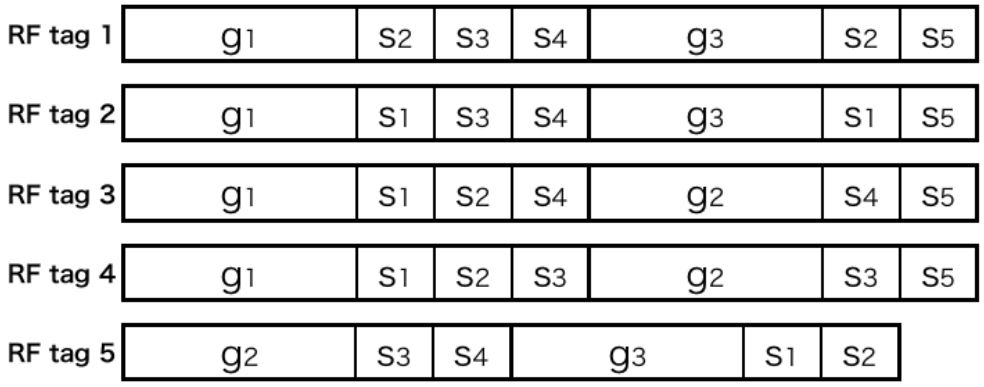

Figure 4.1: Example of the information recorded in RF tags belonging to a group of 5 objects

the unique IDs of RF tags missing from each sub-group can be calculated as the XOR of the sub-group's group ID and the unique IDs of the all identified RF tags belonging to the sub-group just like the $G_{em}m_m$ in Eq.3.12. The short IDs of RF tags missing from each sub-group are recorded by all the identified RF tags belonging to the sub-group.

To determine the unique IDs of missing RF tags, the group coding system must retrieve the matrix G_{em} in Eq.3.12, which is the partial matrix of the group generation matrix and shows the structure of combination of missing unique IDs. G_{em} can be retrieved from the information of short IDs of RF tags missing from each sub-group. For example, suppose that RF tag 2 and 4 are missing from the group shown in Figure4.1. From the information collected from identified RF tag 1, 3 and 5, the following information about missing RF tags is retrieved.

- Two RF tags whose short IDs are s_2 and s_4 are missing from the group.
- The missing RF tag whose short ID is s_2 belongs to sub-group 1 and 3.
- The missing RF tag whose short ID is s_4 belongs to sub-group 1 and 2.

Figure 4.2 shows a graph expression of these information. This graph is a bipartite graph composed of a set of nodes representing missing RF tags and a set of nodes representing incomplete sub-groups. Each "missing RF tag node" is connected, by a single edge, to "incomplete sub-group nodes" which it belongs to. The meaning of this graph structure is just equivalent to that of G_{em}, the relationship between

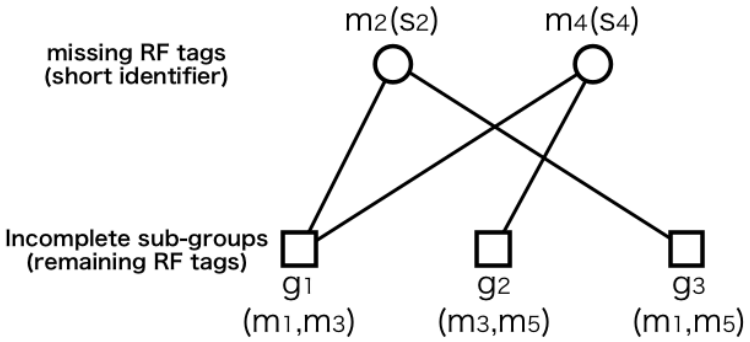

Figure 4.2: Graph expression of retrieved relationship between missing RF tags and sub-groups

missing RF tags and incomplete sub-groups, so this information can be expressed by the following equation.

$$\begin{Bmatrix} g_1 \oplus m_1 \oplus m_3 \\ g_2 \oplus m_3 \oplus m_5 \\ g_3 \oplus m_1 \oplus m_5 \end{Bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} m_2 \\ m_4 \end{Bmatrix} \tag{4.1}$$

The left side of this equation is the $G_{em}m_m$, the array of linear combinations of missing RF tags' unique IDs, and the unknown terms in this equation are only m_2 and m_4, the unique IDs of missing RF tag 2 and 4. In this case, these terms can be calculated as $g_3 \oplus m_1 \oplus m_5$ and $g_2 \oplus m_3 \oplus m_5$, respectively.

4.1.2 Iterative decoding

The group generation matrix describes the Tanner graph of the sub-group and unique IDs, and the whole procedure of the group decoding can be alternatively interpreted as a message passing decoder. The group decoding also starts with the integrity-check of each sub-group by its group ID. Let us take a case where RF tag 2 and 4 are missing from the group shown in Figure4.1 as an example. Figure 4.3 shows the iterative decoding method. The initial Tanner graph can be established as shown in (1) in Fig.4.3 where the missing IDs are represented by filled circles. Thin and thick lines indicate un-identified and identified edges, respectively. Collected variables (unique IDs) are passed to factors (group IDs) as in (2) to recover missing unique IDs (m_2 and m_4) as shown in (3). In process (3), the destination of factor passing are determined by short IDs. The second variable

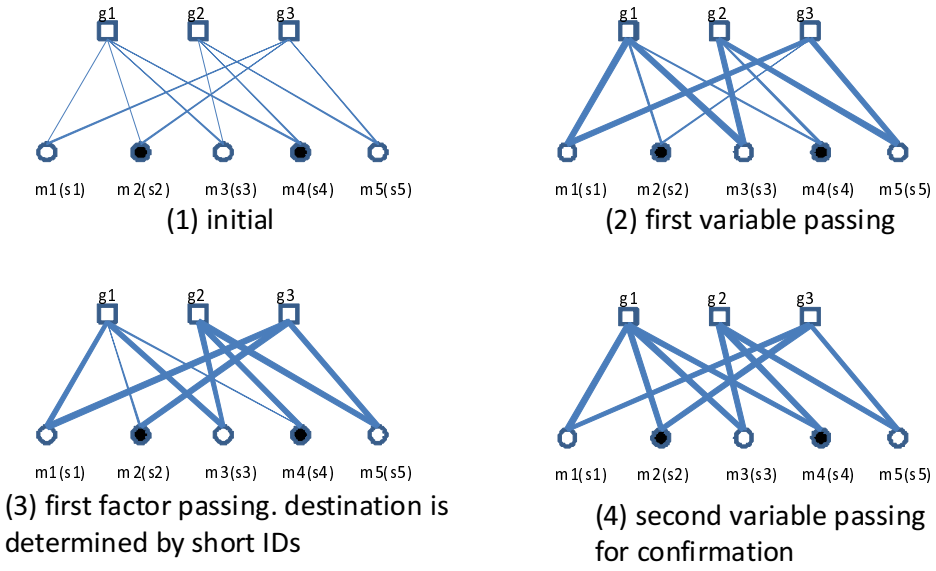

Figure 4.3: Example of the iterative decoding procedure

passing shown in (4) verifies that the recovered unique IDs satisfy the parity check of g_1.

4.2 Evaluation

This section shows the evaluation of the missing ID determination by the extended group coding introduced in the section 4.1. The evaluation is done by the numerical simulations and the experiments with the implementation of the group coding.

4.2.1 Numerical simulation on the determination performance

The missing object identification performance of the proposed method is evaluated with a numerical simulation. The procedure of the simulation is as follows.

1. The simulator prepares 100 virtual RF tags. Each RF tag has 96-bit EPC generated randomly as the unique ID.

2. The simulator applies the group encoding procedure in the extended scheme introduced in the previous section to these 100 RF tags. The group generation matrix is generated like a generation matrix of LDPC [14]. The number

Table 4.1: The conditions of group generation matrix in the numerical simulation

Condition	**V**alue
Number of RF tags which each sub-group contains	4
Number of sub-groups which each RF tag belongs to	2-7 (variable j)

of RF tags which each sub-group contains is fixed at 4, and the number of sub-groups which each RF tag belongs to is given by a variable number j which varies from 2 to 7, as shown in Table 4.1. The 8-bit serial number generated by the group coding system is used as a short ID of each RF tag. Since a set of the information needed to be recorded to an RF tag for a sub-group is 120-bit (96-bit group ID and three short IDs of the other RF tags in the subgroup), the bit-length of group's information recorded by each RF tag is $120j$-bit. It is expected that the performance of the determination increases by increasing the number of sub-groups which each RF tag belongs to.

3. The simulator removes n RF tags randomly from the group. n varies from 1 to 99.

4. The simulator applies the group decoding procedure in the extended scheme to $(100 - n)$ identified RF tags to determine the unique IDs of missing RF tags. If there is any difference between the result of the group decoding and the unique IDs of RF tags actually missing, it is counted as an error.

5. The simulator repeats the above procedure 10000 times for each j and n, and calculates error rate, which is the number of the errors occurred in 10000 trials, for each pattern.

Figure 4.4 shows the result of the simulation. From this chart, it is clear that the performance of the determination increases by increasing the number of sub-groups which each RF tag belongs to. Figure 4.5 shows the maximum number of RF tags which can be determined by the group coding within the several specified error rates, and this number also increases when the number of sub-groups which each RF tag belongs to is increased. For example, when each RF tag belongs to 4 sub-groups and records 480-bit additional information, the group coding system can determine up to 46 missing RF tags' unique IDs within 0.5% error rate. On the other hand, with 7 sub-groups for each RF tag, which means each RF tag records 840-bit information, the unique IDs of 64 missing RF tags can be determined

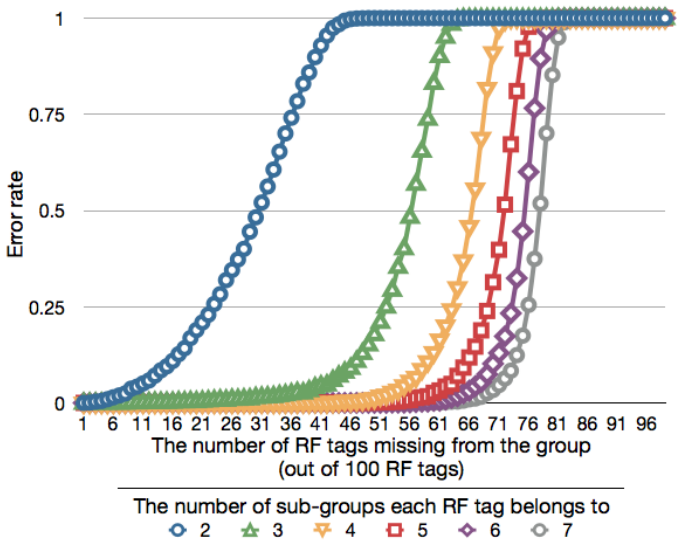

Figure 4.4: Error rate of the determination of missing RF tags' unique IDs by the group decoding

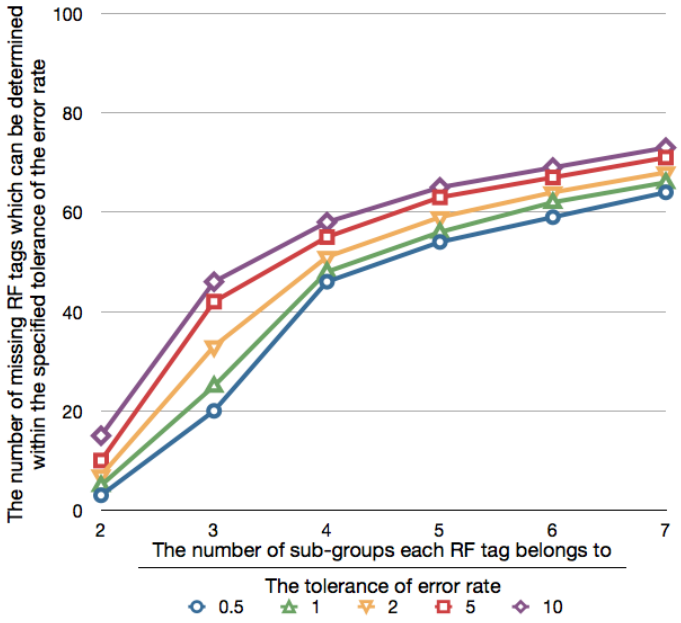

Figure 4.5: The maximum number of RF tags which can be determined within the specified error rate

within the same error rate. These results agree to the afore-mentioned expectation that the performance of the determination increases when increasing the amount of group-related information.

4.2.2 Experimental evaluation

This extended group coding is also implemented on a commercial RFID interrogator KU-U1601 (Figure3.8). With this implementation, the missing object identification performance is evaluated. The procedure of the experiment is as follows.

1. 20 physical RF tags whose unique IDs are generated randomly are prepared. These RF tags are positioned in front of the antennas connected to the interrogator.

2. The group encoding procedure in the extended scheme is applied to these RF tags. $j = 3$ case and $j = 7$ case are examined, and other conditions of group generation matrix are the same as those of the afore-mentioned numerical simulation. In this experiment, 16-bit hashes of RF tags' unique IDs are used to calculate group IDs, which means that each 16-bit hash is treated as the unique ID of each RF tag, so a set of the information about one sub-group has 40-bit length (16-bit group ID and three 8-bit short IDs). Therefore, each RF tag records 120-bit and 280-bit group-related information in $j = 3$ case and $j = 7$ case, respectively.

3. n RF tags are removed from the group and apply the group decoding procedure in the extended scheme to identified $(20 - n)$ RF tags. Then, the result of the group decoding and the hashes of physically missing RF tags' unique IDs are compared. Three cases, $n = 4$, $n = 8$ and $n = 12$, are examined.

4. The above procedure is repeated 10 times for each j and n to calculate the actual error rate of the determination for each pattern.

The performance of the proposed scheme against 20 RF tags is revealed by an additional numerical simulation and the result is shown in Figure4.6. The result of the numerical simulation and that of the experiment are compared.

The results of the experiment of $j = 3$ case and $j = 7$ case are shown in Figure4.7 and Figure4.8, respectively. In these charts, the line shows the result of the numerical simulation, and filled circles represent the result of this experiment.

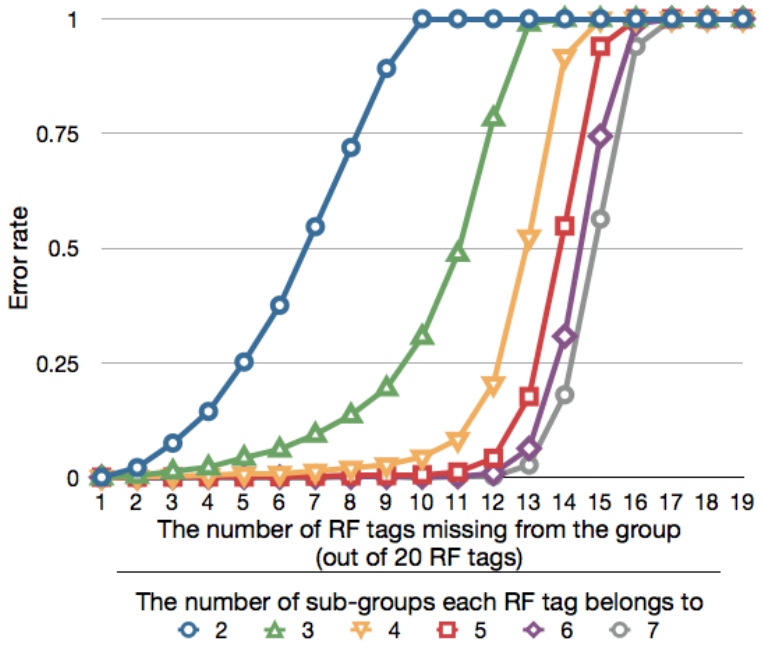

Figure 4.6: Error rate of the determination of missing RF tags' unique IDs by the group decoding (against 20 RF tags)

Because the number of trials for each pattern is small, there is a slight difference between the result of the numerical simulation and that of the experiment in $j = 3$ case, and there is no discernible difference between them. In $j = 7$ case, there is no difference between the result of the numerical simulation and that of the experiment. In this experiment, the time duration of the computation of the group coding is also measured. It is revealed from the measurement that the computation of the group encoding and decoding procedure takes about 0.1 second and 0.2 second, respectively. These duration are negligible in the whole process including the access to the RF tags. Therefore, it is confirmed from these results that the proposed scheme works well on the real RFID system.

4.3 Conclusion of this chapter

The extended group coding writes the graph topology of each RF tag and sub-groups into RF tags' memory in addition to group ID. By doing this, the extended group coding can determine the unique IDs of RF tags missing from the group.

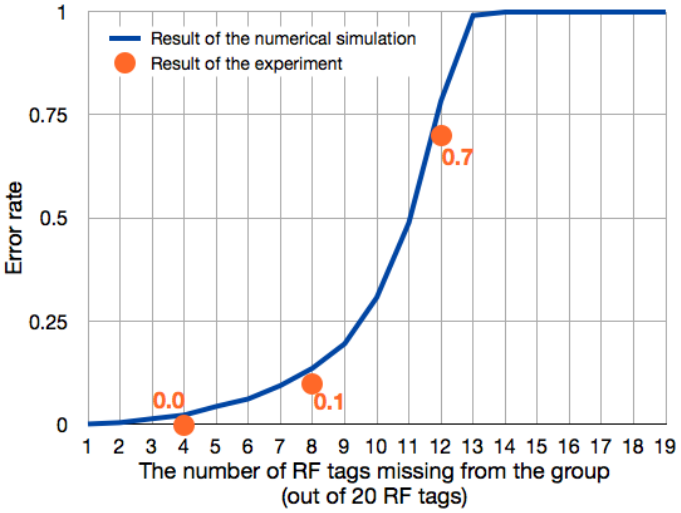

Figure 4.7: The result of the experiment ($j = 3$)

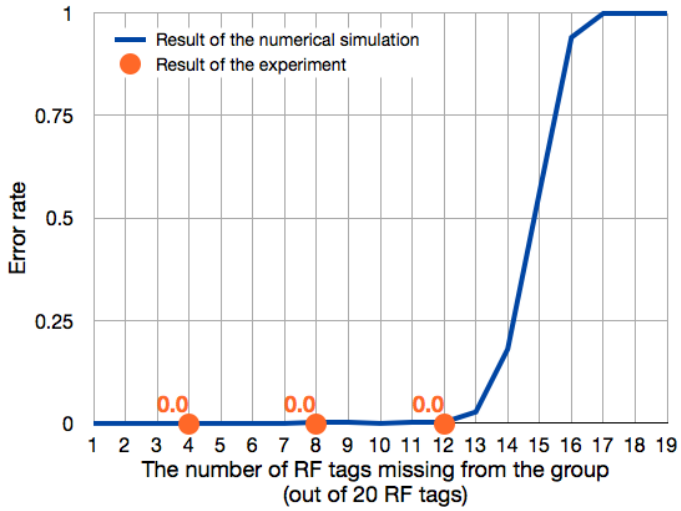

Figure 4.8: The result of the experiment ($j = 7$)

The group decoding can be done by an iterative decoding method by using short IDs. Numerical simulation reveals that the proposed scheme can identify up to 64 missing RF tags' 96-bit unique IDs out of 100 RF tags by writing 840-bit group-related information to each RF tags. Experimental evaluation is also performed with a group of 20 RF tags. Measured accuracy of the determination of 12 missing RF tags are 30% and 100% with 120-bit and 280-bit RF tag memory consumption, respectively. This accuracy agrees well with the numerical simulation.

Chapter 5

Conclusion

This thesis introduces the group coding of RF tags. The group coding can check the integrity of a group in the off-line environment by writing group-related data into grouped RF tags' user memory. In addition, when some RF tags are missing from the group, the group coding can determine the information of missing RF tags. The group coding scheme introduced in the chapter 3 can determine the number of missing RF tags. The extended group coding scheme introduced in the chapter 4 can determine the unique IDs of missing RF tags.

The performance of the group coding is revealed by the numerical simulations. The first group coding scheme can determine 10 RF tags missing from the group of 20 RF tags within 0.5% error rate with writing 96-bit group-related information to each RF tags. The second group coding scheme can determine 64 missing RF tags' 96-bit group IDs from the group of 100 RF tags within 0.5% error rate with writing 840-bit group-related information to each RF tag. The accuracy of the determination increases with increasing the size of group-related information recorded by each RF tag. It is also revealed that the accuracy of the determination is affected by the selection of the parameters of the group coding, which are represented by a group generation matrix. The validity of these numerical simulations is confirmed by the experimental evaluation with the implementation of the group coding using a commercial UHF-band RFID interrogator, and it is confirmed that the group coding works well on the real RFID system.

Appendix

A Overview of the group coding implementation

This section introduces the overview of the implementation of group coding, which is used for the numerical simulations and the experimental evaluations. The programs of the implementation are written in Java. In the following subsections, some principal classes in this implementation are introduced. The simulator and the experiment's system is made with using following classes classified into 4 packages.

A.1 'groupingprogram' package

'groupingproof' package contains classes which compute the procedure of the group coding. Some classes related to the computation are also contained. FigureA.1 shows the class diagram of this package.

'GroupingProgram' interface

'GroupingProgram' interface must be implemented by all the classes that computes the procedure of the group coding. The procedure of group encoding and decoding are written in 'makeGroup' and 'verifyGroup' method, respectively.

'FundamentalGroupCoding' class

'FundamentalGroupCoding' class implements 'GroupingProgram' interface. The group coding procedure of this class detects the missing of RF tags from the group (Section 3.1.1). The hash function is specified as the argument of the constructor.

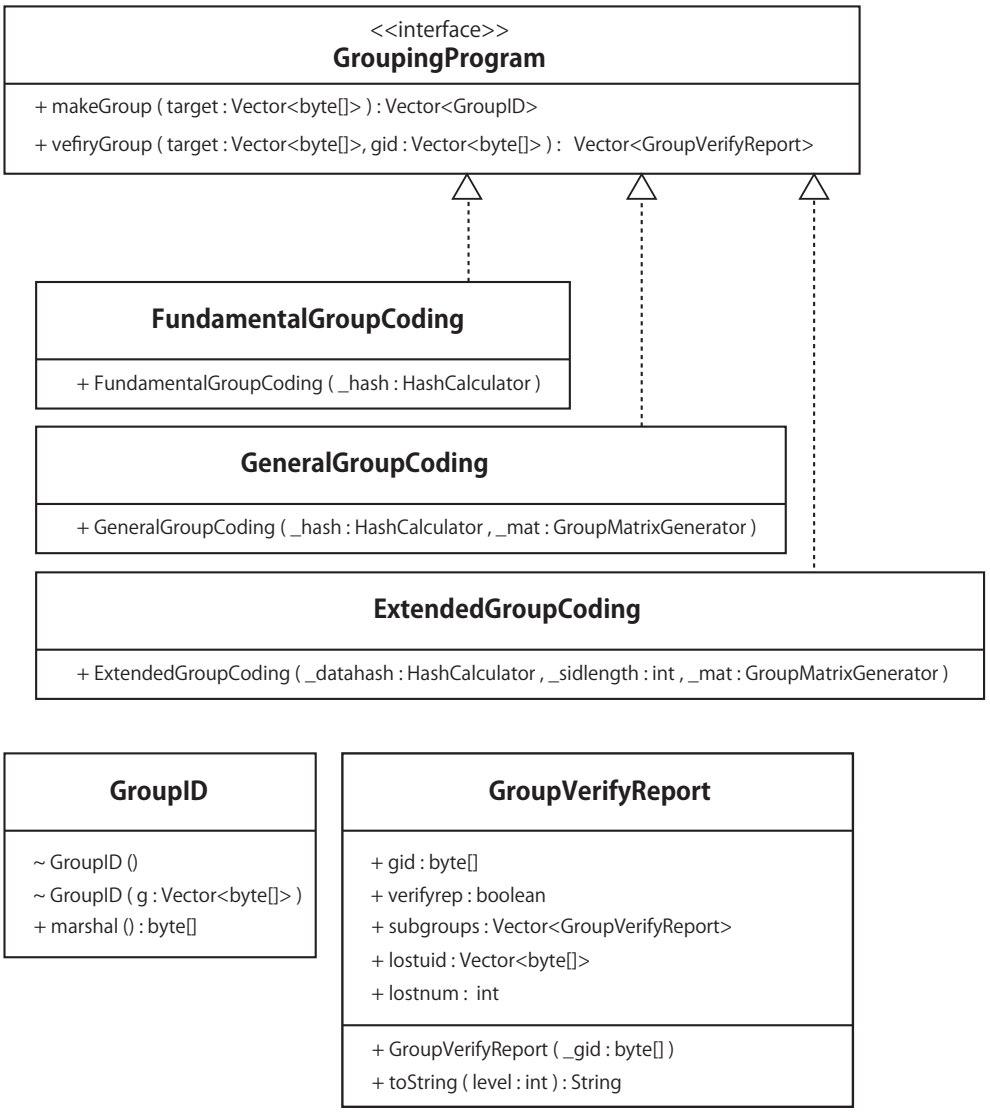

Figure A.1: Class diagram of 'groupingprogram' package

‘GeneralGroupCoding’ class

‘GeneralGroupCoding’ class implements ‘GroupingProgram’ interface. The group coding procedure of this class determines the number of RF tags missing from the group (Section 3.1.2). The hash function and the group generation matrix are specified as the first and second argument of the constructor, respectively.

‘ExtendedGroupCoding’ class

‘ExtendedGroupCoding’ class implements ‘GroupingProgram’ interface. The group coding procedure of this class determines the unique IDs of RF tags missing from the group (Section 4.1). The hash function, the byte length of short IDs and the group generation matrix are specified as the first, second and third argument of the constructor, respectively.

‘GroupID’ class

‘GroupID’ class represents a set of sub-group IDs for one RF tag. ‘marshal’ method returns an array of bytes which is written to the RF tag’s memory.

‘GroupVerifyReport’ class

‘GroupVerifyReport’ class represents the result of the group encoding, which is the return value of ‘verifyGroup’ method of the ‘GroupingProgram’ interface and its implementations. The variable ‘verifyrep’, ‘lostnum’ and ‘lostuid’ represent the result of the integrity-check, the number of missing RF tags and the unique IDs of missing RF tags, respectively. ‘toString’ method return a string containing these information.

A.2 ‘groupingprogram.hashcalculator’ package

‘groupingprogram.hashcalculator’ package contains classes which calculates a hash of given bit string (an array of bytes). FigureA.2 shows the class diagram of this package.

‘HashCalculator’ interface

‘HashCalculator’ interface must be implemented by all the classes that computes a hash of RF tag’s unique ID. ‘calcHash’ method calculates a hash of an array of

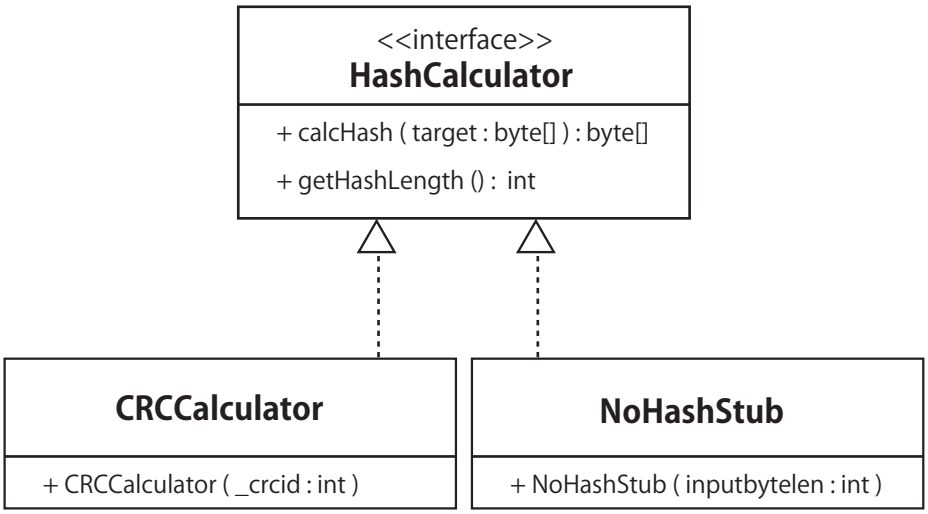

Figure A.2: Class diagram of 'groupingprogram.hashcalculator' package

bytes given as the argument and returns the result of the calculation as another array of bytes. 'getHashLength' method returns the length of an array of bytes which 'calcHash' method returns.

'CRCCalculator' class

'CRCCalculator' class implements 'HashCalculator' interface. This class calculates a CRC, cyclic redundancy code, of given array of bytes as its hash. The identification number of a generator polynomial of CRC (shown in Table A.1) is specified as the argument of the constructor.

'NoHashStub' class

'NoHashStub' class implements 'HashCalculator' interface. 'calcHash' method of this class does not calculate a hash but returns given array of bytes without any computation. When the above-mentioned 'ExtendedGroupCoding' class is used to determine the unique IDs of missing RF tags, the first argument of its constructor must be an instance of this class.

Table A.1: CRC generator polynomials available at 'CRCCalculator' class [17]

ID	Name	Generator polynomial
0	CRC-16	$x^{16} + x^{15} + x^2 + 1$
1	SDLC (CCITT)	$x^{16} + x^{12} + x^5 + 1$
2	CRC-16 Reverse	$x^{16} + x^{14} + x + 1$
3	SDLC Reverse	$x^{16} + x^{11} + x^4 + 1$
4	LRCC-16	$x^{16} + 1$
5	CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$
6	LRCC-8	$x^8 + 1$
7	ETHERNET	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11}$ $+x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

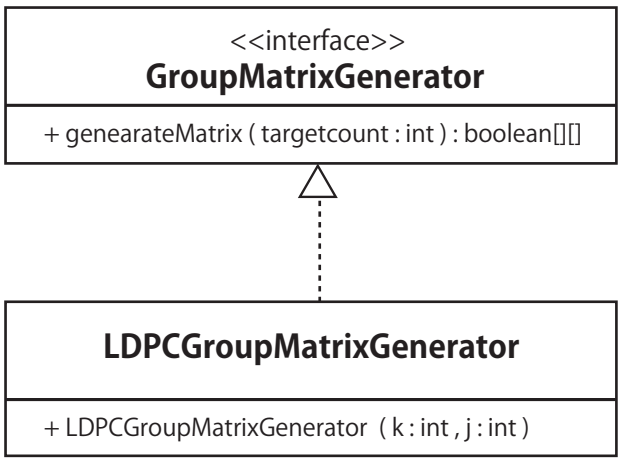

Figure A.3: Class diagram of 'groupingprogram.matrixgenerator' package

A.3 'groupingprogram.matrixgenerator' package

'groupingprogram.matrixgenerator' package contains classes which generate group generation matrixes. The classes in this package is used by 'GeneralGroupCoding' class and 'ExtendedGroupCoding' class. FigureA.3 shows the class diagram of this package.

'GroupMatrixGenerator' interface

'GroupMatrixGenerator' interface must be implemented by all the classes that generate group generation matrixes. 'generateMatrix' method generates a group

generation matrix against the number of RF tags, which is given as the first argument, and return it.

'LDPCGroupMatrixGenerator' class

'LDPCGroupMatrixGenerator' class implements 'GroupMatrixGenerator' interface. 'generateMatrix' method of this class generate matrixes like LDPC's generation matrix [14]. The number of RF tags belonging to each sub-group and the number of sub-groups which each RF tag belongs to are specified as the first and second argument of the constructor, respectively.

A.4 'rwprogram' package

'rwprogram' package contains classes which controls a physical or virtual RFID interrogator. Some related classes and enumeration types are also contained. FigureA.4 shows the class diagram of this package.

'RFIDReaderWriter' interface

'RFIDReaderWriter' interface must be implemented by all the classes that provide methods to control an RFID interrogator. 'inventory' method returns the list of unique IDs of read RF tags. 'read' method reads data from the specified memory area of specified RF tags and returns it. 'write' method writes specified data into the specified memory area of specified RF tags. 'getBlockLength' method returns the byte length of one memory block, which is a minimum unit of RF tag's memory.

'SimulationReaderWriter' class

'SimulationReaderWriter' class implements 'RFIDReaderWriter' interface. This class prepares virtual RF tags which have randomly generated 96-bit unique IDs and user memory. 'inventory', 'read' and 'write' methods access to these virtual RF tags. The number of virtual RF tags is specified as the argument of the constructor. The numerical simulations of the group coding are done using this class.

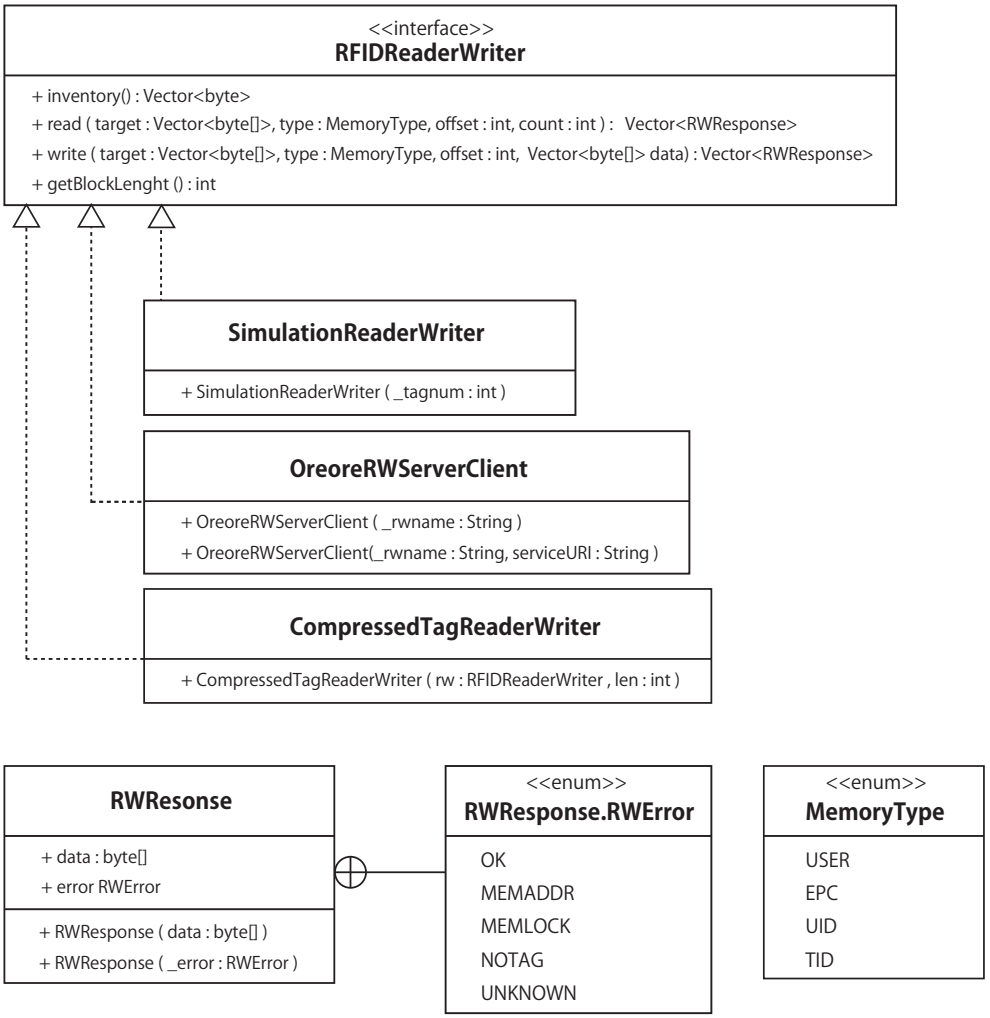

Figure A.4: Class diagram of ‘rwprogram’ package

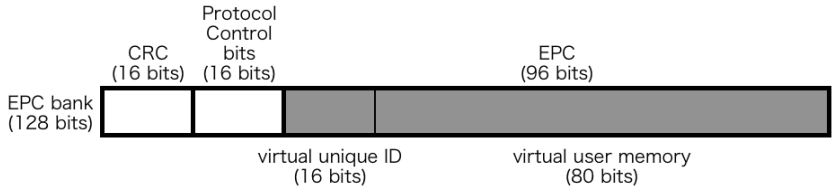

Figure A.5: Virtual user memory allocated in EPC

'OreoreRWServerClient' class

'OreoreRWServerClinet' class implements 'RFIDReaderWriter' interface. This class is used to control the "KU-U1601" interrogator, which is used for the experiments. Detail of the implementation to control KU-U1601 is described in the appendix B.

'CompressedTagReaderWriter' class

'CompressedTagReaderWriter' class implements 'RFIDReaderWriter' interface. This class is a virtual interrogator which works on another class implementing 'RFIDReaderWriter' interface, which is specified as the first argument of the constructor. Using this class, group-related data is written into each physical RF tag as the part of unique ID of the RF tag as shown in FigureA.5. This class is used for fast demonstration of the group coding because the information required to the group coding can be collected without reading of user memory of RF tags. The byte-length of virtual unique ID is specified as the second argument of the constructor.

'RWResponse' class

'RWResponse' class represents the result of an access to RF tags, which is the return value of 'read' and 'write' method of 'RFIDReaderWriter' interface. The variable 'error', whose type is 'RWResponse.RWError' enumeration type, is used to indicate the result of the access. The variable 'data' is used by only 'read' method and contains an array of bytes read from the memory of an RF tag when 'error' is 'RWError.OK'.

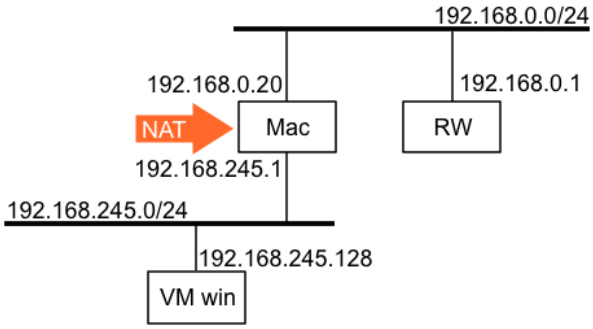

Figure B.1: Network structure to control KU-U1601

'MemoryType' enumeration type

'MemoryType' is an enumeration type which represents the kinds of memory bank of RF tags. This type is used by 'read' and 'write' method of 'RFIDReader-Writer' interface to specify which memory bank is read/written.

B Implementation of KU-U1601 controller

The KU-U1601 interrogator, which is provided by Panasonic System Networks Co., Ltd, communicates with a PC on IP, Internet Protocol. However, the communication between the interrogator and the PC is done on the closed protocol. Panasonic provides an API library to control the interrogator, but this library is made for Windows and difficult to call directly from the group coding program written in Java.

To avoid this problem and execute the group coding program on my MacBook Pro, the system which remotely controls the interrogator is implemented. The overview of the implemented system is as follows. The group coding program introduced in the appendix A is executed on the MacBook Pro whose operating system is Mac OS X (referred to as "Mac"). There is a virtual machine whose operating system is Windows XP in the Mac. The supervisor on the Mac has a role of NAT server and makes a private network for this virtual machine. The interrogator's IP address is statically set to '192.168.0.1/24'. Accordingly, the IP address of the Mac is set to '192.168.0.20/24'. This network structure is also shown in FigureB.1 With this network, the group coding software can order the Windows on the virtual machine to access to the interrogator through the inner network, and the Windows can access to the interrogator through NAT.

Detail of the implementation (Windows side)

The system on the Windows is implemented as a web service. When this web service is called, the system accesses to the interrogator and returns the result of the access. The whole system is implemented using C# and ASP.NET. IIS, internet information service, is used as the web server.

Detail of the implementation (Java side)

Because the system on the Windows, which accesses to the interrogator, is a web service, the Java program which orders the Windows to access to the interrogator is a client of this web service. This program is written with the assistance of Apache Axis2. The Axis2 makes the fundamental program to access to the web service from its WSDL, and 'OreoreRWServerClient' class is implemented to adapt this program to 'RFIDReaderWriter' interface.

Bibliography

[1] EPC Information Services (EPCIS) Version 1.0.1 Specification. Errata Approved by TSC on September 21, 2007.

[2] A. Juels. "Yoking-proofs" for RFID tags. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 138–143, 2004.

[3] J. Saito and K. Sakurai. Grouping proof for RFID tags. In *19th Conference on Advanced Information Networking and Applications (AINA)*, volume 2, pages 621–624. IEEE, 2005.

[4] L. Bolotnyy and G. Robins. Generalized Yoking-Proofs for a Group of RFID Tags. In *Proc. International Conference on Mobile and Ubiquitous Systems (Mobiquitous)*. Citeseer, 2006.

[5] S. Piramuthu. On existence proofs for multiple RFID tags. In *ACS/IEEE International Conference on Pervasive Services*, pages 317–320. IEEE, 2006.

[6] C.C. Lin, Y.C. Lai, J. Tygar, C.K. Yang, and C.L. Chiang. Coexistence Proof using Chain of Timestamps for Multiple RFID Tags. *Advances in Web and Network Technologies, and Information Management*, pages 634–643, 2007.

[7] Y. Lien, X. Leng, K. Mayes, and J.H. Chiu. Reading order independent grouping proof for RFID tags. In *IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 128–136. IEEE, 2008.

[8] S. Inoue, D. Hagiwara, and H. Yasuura. Systematic error detection for RFID reliability. In *The First International Conference on Availability, Reliability and Security (ARES)*, page 7. IEEE, 2006.

[9] V. Potdar, P. Hayati, and E. Chang. Improving RFID read rate reliability by a systematic error detection approach. In *1st Annual RFID Eurasia*, pages 1–5. IEEE, 2007.

[10] J.W. Byers, M. Luby, and M. Mitzenmacher. A digital fountain approach to asynchronous reliable multicast. *IEEE Journal on Selected Areas in Communications*, 20(8):1528–1540, 2002.

[11] M. Luby. LT codes. In *43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 271–280. IEEE, 2003.

[12] D.J.C. MacKay. Fountain codes. In *IEE Proceedings Communications*, volume 152, pages 1062–1068. IET, 2005.

[13] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6):2551–2567, 2006.

[14] R. Gallager. Low-Density Parity-Check Codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.

[15] EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz Version 1.2.0.

[16] EPC Tag Data Standard Version 1.5. August 18, 2010.

[17] S.G. Gaitonde and T.V. Ramabadran. A Tutorial on CRC Computations. *IEEE Micro*, 8(4):62–75, 1988.

謝辞

本論文執筆にあたり御助言を頂きました、慶應義塾大学環境情報学部教授 村井純博士、同教授 中村修博士、同准教授 楠本博之博士、同専任講師 Rodney D. Van Meter III 博士、同准教授 植原啓介博士、同教授 武田圭史博士に感謝いたします。

また、研究について日頃から御指導頂きました、慶應義塾大学環境情報学部准教授 三次仁博士、政策メディア研究科特任講師 羽田久一博士、同特任助教 中根雅文氏、慶應義塾大学インフォメーションテクノロジーセンター本部助教 鈴木茂哉氏、神奈川工科大学情報学部情報工学科准教授 稲葉達也博士に感謝いたします。特に三次仁博士には、学部二年次に研究室に所属して以来本研究について直接御指導頂きました。この場を借りて深く御礼申し上げます。

Auto-ID Lab にてお世話になりました、慶應義塾大学 SFC 研究所 松本伸史氏、株式会社日放電子 白石雅彦氏に感謝いたします。また、研究生活を共にした、神谷尚保氏、佐藤泰介氏、佐藤龍氏、金仙麗氏、田村哲郎氏、江村桂吾氏、山田真弘氏、広石達也氏、富田千智氏、斉藤俊氏、能島良和氏、鈴木詩織氏、杉本健一氏、山口修平氏、米村茂氏、宮崎圭太氏、横石雄大氏、Doan Hoai Nam 氏、吉田守氏、五十嵐祐貴氏、三樹良亮氏、内山貴博氏、小薗宏樹氏、清水真有氏に感謝いたします。特に arashi と mayu、卒論執筆にかまけて研究指導がおざなりになってしまっていたかもしれません。ごめんなさい。今後ともよろしくお願いいたします。また、徳田・村井・楠本・中村・高汐・バンミーター・植原・三次・中澤・武田合同研究プロジェクトの皆様に感謝いたします。

最後に、父 佐藤俊寛、母 佐藤千恵子、姉 佐藤真理子に心から感謝いたします。

以上をもって本論文の謝辞とさせていただきます。