Privacy Enhanced RFID Using Quasi-Dyadic Fix Domain Shrinking

Tomohiro Sekino*, Yang Cui†, Kazukuni Kobara‡ and Hideki Imai*†

*Department of Electrical, Electronic & Communication Engineering, Chuo University
1-13-27 Kasuga, Bunkyo-ku, Tokyo 112-8551, Japan
Email: tomohiro-sekino@imailab.jp
†Research Center for Information Security (RCIS),
National Institute of Advanced Industrial Science & Technology (AIST)
1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan
Email: {y-cui,k-kobara,h-imai}@aist.go.jp, WWW: http://www.rcis.aist.go.jp/

Abstract—Recently, Radio Frequency IDentification (RFID) systems are intensively studied and widely used in every-day applications, such as, retailing, supply chain management, and medical equipment management. Tags in RFID systems are highly efficient to be managed and tracked, but at the same time suffering from impersonation and privacy problems. Consequently, RFID systems are required to provide both efficient management, as well as authentication and privacy protection. In this paper, on the basis of fast and light-weight Niederreiter public-key cryptosystem, we propose an efficient RFID authentication protocol which satisfies the above requirements, and enjoys the following merits: 1) unlike most of the previous works that employ symmetric key cryptographic techniques, our proposal has a fast computation to find authenticated ID and needs no exhaustive search in database, which reduces the searching time significantly; 2) the memory size to store the key in RFID tags can be greatly reduced by our novel methods.

I. INTRODUCTION

Recently, Radio Frequency IDentification (RFID) systems are intensively studied and widely used in every-day applications, such as, retailing, supply chain management, and medical equipment management. The security requirement of RFID systems pursues authenticity between a reader (or a back-end server) and RFID tags, to protect the private information of valid tags and prevent authentication from impersonation of malicious tags. Since in usual RFID systems the ID information of tags is sent via radio waves, it is easy for adversary to eavesdrop all communications between a reader and a tag, or to track some certain tag in different places to obtain the traveling pattern. Thus, RFID systems require unlinkability of RFID tags. In a nutshell, even if the adversary knows a tag belongs to one of two distinct known IDs, he/she cannot distinguish it from the output of the tag with overwhelming probability larger than one half. Only the valid entity with authority such as the reader should identify the tag and its corresponding ID in database. Most of existing authentication solutions, such as, Randomized Hash Lock [1] and Hash Chain [2], are employing symmetric cryptographic techniques, which normally need to share a key between a reader and a tag beforehand. A crucial issue is that the computation complexity of the authentication process increases when the number of tags becomes large. This is because the reader must search the tag’s secret key from the database exhaustively. Thus, it is not suitable for RFID systems managing a large number of tags. Though a series of works since [3] has proposed some methods to reduce readers’ computation complexity, it is still not practical when the number of tags is large and may even introduce additional problems to degrade the security. More details refer to the survey paper by Ari Juels [4, Sec.III.C].

On the other hand, if public key cryptographic techniques could be used on low-power devices, a tag can make use of reader’s public key to encrypt its ID. Suppose that the underlying encryption is semantically secure, then unlinkability is naturally available since the ciphertext gives no useful information to adversary even if he knows one of two chosen messages (IDs). By adopting the above techniques, a reader can immediately recover the tag’s ID by one decryption operation only, without brute-force search or synchronization like [2]. However, most of the current public key cryptosystems such as RSA or Elliptic Curve Cryptosystem, need much computational cost, and they are too expensive to use on low-power devices. To solve this problem, in this paper we use Niederreiter PKC [6] whose encryption processing is fast and which needs only light-weight operation, such as bitwise exclusive-OR which could be performed on very simple circuits. In contrast, a modular exponentiation computation requires more computational time and much more complex circuits. The only drawback of Niederreiter PKC comes from its large public key size, which is typically around 2M bits. In this paper, we propose a authentication protocol which greatly reduces the size of the public key (stored in tags) by combining Fix Domain Shrinking (FDS) of Personalized Public-Key Cryptosystem (P2KC) [7] with Quasi-Dyadic (QD) codes [8]. P2KC is a method to make a public key individually used by certain ID, namely Personalized Public Key(PPK). The reader can obtain the tag’s ID by decrypting the ciphertext encrypted with PPK. We also propose Flexible Quasi-Dyadic codes that the Quasi-Dyadic structure is flexibly feasible, which we believe has independent interest in coding theory and cryptography.
II. PREPARATION

A. Model

We typically define users, tags, a reader, and a back-end server in a RFID system. Our purpose is to construct a secure authentication protocol implementable in low-power devices. We consider in the following model that the server manages a number of IDs and does not leak ID’s information. In initialization of the protocol, a user receives a tag in a secure way, and the reader communicates with the server via a secure channel.

1) Motivation: Our proposal must satisfy the following conditions.
   • Correctness: A valid tag succeed in authentication.
   • Soundness: An invalid tag fail in authentication with overwhelming probability.
   • Security: Secure information of tag for authentication can not be leaked, and even in some cases authentication itself can not be linked to meaningful targets.

2) Attack model: We assume that an adversary can passively eavesdrop all messages between tag and the reader, as well as impersonate with obtained information.

B. Niederreiter Public Key Cryptosystem [6]

The security of Niederreiter PKC relies on Syndrome Decoding Problem.

• Syndrome Decoding Problem (SDP): Given $s \in GF(q)^{n-k}$ and $H \in GF(q)^{(n-k)\times n}$, an integer $0 < t < n$, find $e \in \{0, 1\}^n$ and its Hamming weight $W_H(e)=t$, s.t. $eH^T=s$.

Syndrome Decoding Problem is an NP-hard problem, if $H$ is random matrix [10], and $t$ is small. Algorithms of Niederreiter PKC are shown as follows:

1) KeyGen(): input: security parameter; output: public key $pk= (K, t)$ and secret key $sk= (S, P, \phi)$
   • $H$: Parity-check matrix of a t-error-correcting(n,k) binary linear code. And $\phi$ is error correcting algorithm corresponding to $H$.
   • $S$: $(n-k) \times (n-k)$ random binary non-singular matrix.
   • $P$: $n \times n$ random binary non-singular matrix.

Public key: $K=SHP$, $t$.

2) Encryption(): input: $pk=(K,t)$, $msg$; output: $c$. The message is a binary vector of length $n$ with Hamming weight $t$, $c$ is a binary vector of length $n-k$,

$$c = msg \cdot K^T$$

3) Decryption(): input: $sk = (S, P, \phi)$, $c$; output: $msg$.
   First, it computes $S^{-1}c^T$ as follows:

$$S^{-1} \cdot c^T = H \cdot P \cdot msg^T$$

Second, it applies the decoding algorithm $\phi$ to $S^{-1}c^T$. And it calculates $P \cdot msg^T$.

$$P \cdot msg^T = \phi(S^{-1} \cdot c^T)$$

Third, $msg$ of $c$ is given by

$$msg = [P^{-1} \cdot P \cdot msg^T]^T$$

C. Niederreiter Personalized Public Key Cryptosystem [7] (P²KC)

In general, the public key cryptosystems cannot identify encryptor from the ciphertext. This section explains why P²KC that can identify the encryptor from the ciphertext. P²KC generates encryption key $ppk$ from the public key of Niederreiter PKC with $n$ dimension vector $pv$ (personalized vector). The sender encrypts plaintext with using $ppk$. The decrypter can identify $pv(ID)$ from the plaintext, and identify encrypter. The construction of $ppk$ takes $(n_1+2)$ subset from $n$ columns of public key $K$ of the Niederreiter PKC.

• $Sub_0$: set coordinates of $pv$ be 0.
• $Sub_1$: set coordinates of $pv$ be 1. And a set is exclusive-ORed to make $c_2$.
• $Sub_2$: set coordinates of $pv$ be $(n_1+2) \geq i \geq 2$. And set of coordinates to be exclusive-ORed to make the $(i-1)th$ column of $K_1$.

Let $|Sub_1|$ denote the cardinality of $Sub_1$, $Sub = \{Sub_0,|Sub_1|,...,|Sub_{n_1+1}|\}$, $n = \sum_{i=0}^{n_1} |Sub_i|$. $Sub$ should fulfill the following:

$$msg^* \cdot Sub^t + |Sub_1| \leq t$$

where $msg^*$ is a vector of length $n_1$, and $Sub^t = \{|Sub_0|,|Sub_1|,...,|Sub_{n_1}|\}$. The output of $ppk(K_1,c_2,t,Sub)$ becomes $(n-k) \times n_1$ binary matrix $K_1$, dimension vector $c_2$, and $Sub$ is $(n_1+2)$ sequence. Encryption is

$$c = (msg^*:K_1^T) \oplus c_2.$$
D. Quasi-Dyadic (QD) Goppa code [8]

Quasi-Dyadic Goppa codes are composed of the parity-check matrix of Quasi-Dyadic matrix by using the structure of Goppa codes.

[Definition 1]: Given ring $R$, vector $h = (h_0,..,h_{n-1}) \in R^n$. Dyadic matrix $\psi(h) \in R^{n \times n}$ is a symmetric matrix given $\psi_{ij} = h_{i \oplus j}$. The vector $h$ is called signature of the code. For example, when given signature $h = (h_0,h_1,h_2,h_3)$, dyadic matrix $M$ is

$$M = \begin{bmatrix}
h_0 & h_1 & h_2 & h_3 \\
h_1 & h_0 & h_3 & h_2 \\
h_2 & h_3 & h_0 & h_1 \\
h_3 & h_2 & h_1 & h_0
\end{bmatrix}.$$ (3)

For $k > 0$ and any $2^k \times 2^k$ dyadic matrix has a symmetric matrix every $2^{k-s} \times 2^{k-s}(s = 1,..,k)$ block. So, the dyadic matrix consists of signature $h$.

The size of the public key can be reduced by using this technique.

[Definition 2]: A quasi-dyadic matrix is a block matrix component that blocks are dyadic submatrices.

[Definition 3]: Given two disjoint sequences $z = (z_0,..,z_{t-1}) \in GF(q)$ and $L = (L_0,..,L_{n-1}) \in GF(q)^n$ of distinct element. The Cauchy matrix $C(z,L)$ is $C_{ij} = 1/(z_i - L_j)$, where $0 \leq i \leq t - 1, 0 \leq j \leq n - 1$. The Goppa code generated by monique polynomial becomes a parity-check matrix with the structure of the Cauchy matrix.

1) Flexible Quasi-Dyadic Construction: Let $p = 2^s, q = p^d = 2^m, q/2 \geq N \gg n$, and the code length of $t$ error corrected is $n = lt$($l > d$). Other methods[8] construct $N \times N$ full dyadic matrix $H$ and then remove the row of $H$ to get $t \times N$ quasi-dyadic matrix $H'$. Further, $t \times n$ quasi-dyadic matrix $H$ is composed by shortening, rearranging, and permuting $H'$.

We propose to use construction method that generates the parity-check matrix of $t \times n$ directly (called Flexible Quasi-Dyadic or FQD) [9]. Due to this, we can use smaller $m$ than the original quasi dyadic construction method. That is, our quasi-dyadic construction method can be smaller than the original quasi dyadic construction method. In addition, operation of shorting, rearranging, and permuting can be cut by choice of $\Delta$ and $\delta$. $\delta$ defines inner structure of each full dyadic matrix and $\Delta$ defines the relationship among each full dyadic matrices.

FQD construction is as follows. It generates one small $u \times u$ dyadic matrix using $\delta_i$ for $0 \leq i < \log_2 u$. FQD generates the other $u \times u$ dyadic matrix by shifting them using both $\Delta_i$ and $\Delta_i$ for $0 \leq i < \lceil n/u \rceil$ and $1 \leq i_1 < \lfloor t/u \rfloor$. $z_i$ and $L_j$ are given as follows:

$$z_{i_0} = \bigoplus_{b=0}^{\log_2 u-1} i_0[b] \cdot \delta_b \quad for \ 0 \leq i_0 < u$$ (4)

$$L_{j_1,u+j_2} = z_{j_0} \oplus \Delta_{j_1} \quad for \ 0 \leq j_1 < \lfloor n/u \rfloor$$ (5)

$\Delta$ is a symmetric matrix given $z_i = \bigoplus_{b=0}^{m} i[b] \cdot \delta_b$. $\Delta_i$ and $\Delta_i$ must be chosen at random while making all the $z_i$ for $0 \leq i < t$ and $L_i$ for $0 \leq j < n$ distinct. It is composed as follows:

- $\delta_0$ for $0 \leq b < \log_2 u$ are linearly independent.
- $\forall r \in \{0,1\}^{\log_2 u}$,
  $$\Delta_{i_1}, \Delta_{i_1}, (\Delta_{i_1} + \Delta_{i_1}), (\Delta_{i_1} + \Delta_{i_1}) \notin \bigoplus_{b=0}^{\log_2 u-1} r[b] \cdot \delta_b$$

where $r[b]$ denotes the $(b+1)$-th bit of $r$ in the binary form.

2) Key Generation of Public Key Cryptosystem with Quasi Dyadic: The generation of the public key randomizes with keeping structure of Quasi-Dyadic. $t \times n$ parity-check matrix $H$ is $[B_0[1][B_{n-1}]$, and $B$ is a $t \times t$ full dyadic matrix. The method of randomizing is as follows:

step1 given $p \in GF(q^n)$, multiply nonzero scale factor $\sigma_0,..,\sigma_{t-1} \in GF(p)$ to each block $B$ of $H$.

step2 compute the trace matrix $H'$ in order to convert $GF(q^n)$ to $GF(q)$.

step3 multiply random dyadic matrix to $H'$, and output public key $K$.

$$\begin{bmatrix} A & B \\ B & A \end{bmatrix} \times \begin{bmatrix} C & D \\ D & C \end{bmatrix} = \begin{bmatrix} AC + BD & AD + BC \\ AD + BC & AC + BD \end{bmatrix}$$ (7)

Given $(n,k,t)$, since the public key $K$ has construction of quasi-dyadic and could be done by Gaussian elimination, we can express the size of public key in $(n-k) \times k/t$ matrix. We show security evaluation and the size of public key based on [12] in table 1.

<table>
<thead>
<tr>
<th>security (log)</th>
<th>m</th>
<th>p</th>
<th>t</th>
<th>n</th>
<th>k</th>
<th>t</th>
<th>public key size (bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>86</td>
<td>11</td>
<td>4</td>
<td>24</td>
<td>2016</td>
<td>1684</td>
<td>32</td>
<td>18304</td>
</tr>
<tr>
<td>103</td>
<td>11</td>
<td>4</td>
<td>25</td>
<td>1984</td>
<td>1280</td>
<td>64</td>
<td>14080</td>
</tr>
<tr>
<td>91</td>
<td>11</td>
<td>2</td>
<td>16</td>
<td>1920</td>
<td>864</td>
<td>96</td>
<td>9504</td>
</tr>
<tr>
<td>80</td>
<td>11</td>
<td>2</td>
<td>16</td>
<td>1920</td>
<td>688</td>
<td>112</td>
<td>7568</td>
</tr>
</tbody>
</table>
public key $K$ with the structure of FQD, and makes FDS adjust to $K$.

step1 generate the public key $M = [K_{(n-k) \times k}]$ with the structure of FQD by II-D2.

step2 the setting of FDS is made $P_{sub0} = P_{sub1} = \{1, \ldots, n-n_1\}$ and $P_{sub1} = \{n-n_1, \ldots, n\}$. Where $n_1$ is multiple of $t$.

step3 output $n$ dimension vector $pv$, $(n-k) \times n_1$ matrix $K_1$, and $n-k$ dimension vector $pv_2$ by II-C.

step4 output $K_1$ by $K_1 = [K_1^t|_{n-k}]$.

step5 since $K_1^t$ is a matrix that consists of the dyadic matrix of each the block of $t \times t$, the matrix $K_1$ who left signature of each block is output.

$K_1$ and $c_2$ is stored in a tag.

B. Challenge Response Authentication with FQD-FDS

We use the challenge response style authentication to make the RFID system. This protocol shows as follows: The user receives the tag that has $K_1$ and $c_2$.

step1 The tag receives random $chal$ from the reader.

step2 The tag generates random $msg^* \in \mathcal{G}^{q-1}$ of hamming weight $W_H(\cdot) \leq 1$ where $W_H(\cdot) \leq 1$ denote the Hamming weight of $msg^*$.

step3 The tag computes $c_1 = \text{Enc}(msg^*, K_1^T)$

, and output pseudo-ID PID as following $PID = c_1 \oplus c_2$.

step4 The tag output $aux$ where $aux$ is

$aux = h(chan||msg^*||c_2)$. (10)

Where $h(\cdot)$ is hash function.

step5 The tag sends $aux$ and $PID$ to the server.

step6 The server decrypts $PID$ to $msg$, and identifies $pv$ from $msg$.

step7 The server sequences $msg^*$, $c_2$, and $chal$ and puts those into the hash function $h(\cdot)$. The server verifies whether it becomes the same as $aux$.

C. The Cost of Encryption

The encryption process of our proposal only compute extraction from the signature and matrix calculation. The computation time takes around $25\mu s$, where they use Intel(R) Core(TM)2 Duo CPU running at 1.06GHz with 2.0GByte RAM.

D. The Size of Encryption Key

In proposal, tag needs to store $n-k$ dimension binary vector $c_2$ and $(n-k) \times (n_2 - (n-k))/t$ matrix $K_1$.

E. The Number of IDs

The possible number of IDs is up to $n_2$=ID choices from $c_2 = n_2 \cdot K_1^T$, which is $\binom{n_2}{t}$ s.t. $n_2 = n-n_1$, $t_2 = t - t_1$.

IV. Security Evaluation

We model the adversary to be a passive attack that spoofs as server and to steal ID from tag and an active attack that pretend to be a valid tag. In this model, information that the adversary supposed to know is $chal$, $PID$, $aux$, and $K_1$ of public key between tag and the reader only. The security parameter of system is $\lambda$, which is commonly required to be larger than 80, i.e. $2^{80}$ security level.

Passive attack: Adversary sends $chal$ to the tag. The tag generates $msg^*$, computes $PID$ and $aux = h(chan||msg^*||c_2)$, and then sends $PID$ and $aux$ to the adversary. The adversary guesses $pv$ from $(chal, PID, aux)$. Since the adversary of this model knows the public encryption key, the adversary can employ a Chosen-Plaintext Attack(CPA) [5]. The adversary must invert the Niederreiter PKC to output $pv$. When given success probability that this adversary succeeds with probability $Pr[A_{EP}^{\mathcal{H}}(1^{\lambda})]$ and negligible $\epsilon_{ow}$, it is necessary to become

$Pr[A_{EP}^{\mathcal{H}}(1^{\lambda})] \leq \epsilon_{ow}$ (11)

In this model, if the adversary can break one way hash function of $aux$, one output $msg^*$ and $c_2$. As a result, the spoofing and the leakage of ID happens. When given success probability that this adversary break one way function $Pr[A_{EP}^{\mathcal{H}}(1^{\lambda})]$ and negligible $\epsilon_{ow}$, it is necessary to become

$Pr[A_{EP}^{\mathcal{H}}(1^{\lambda})] \leq \epsilon_h$ (12)

Therefore, if 11 and 12 is fulfilled, it is secure against the passive attack.

Active attack: The adversary receives $chal$ and makes $PID$ and $aux = h(chan||\cdot||\cdot)$, then sends $PID$ and $aux$ to the server. The server authenticates from $PID$ and $aux$.

The adversary builds $PID$ and $aux$ from $c_2$ and $msg^*$ chosen by the adversary. However, if it is not $c_2$ made from registered $pv$, it fails to be authenticated. Since $c_2$ is $n-k$ dimension vector, probability that this adversary succeeds is,

$Pr[A_{EP}^{\mathcal{H}}(1^{\lambda})] = \frac{1}{2n-k}$ (13)

where $\frac{1}{2n-k}$ is negligible.

Note that it needs to use a semantically secure variant (easy to obtain) of Niederreiter PKC if a stronger notion “unlinkability” of RFID is required. The detailed proofs refer to the full version of this paper.
TABLE II
SAMPLE PARAMETER OF FQD-FDS AUTHENTICATION PROTOCOL

<table>
<thead>
<tr>
<th>security(log)</th>
<th>m</th>
<th>p</th>
<th>l</th>
<th>n</th>
<th>k</th>
<th>n1</th>
<th>k1</th>
<th>t</th>
<th>t1</th>
<th>IDsl(log)</th>
<th>c2</th>
<th>K1</th>
<th>stored key size(bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>82</td>
<td>11</td>
<td>15</td>
<td>70</td>
<td>2016</td>
<td>1664</td>
<td>864</td>
<td>32</td>
<td>30</td>
<td>19</td>
<td>352</td>
<td>5632</td>
<td>5984</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>11</td>
<td>12</td>
<td>57</td>
<td>1984</td>
<td>1280</td>
<td>1088</td>
<td>64</td>
<td>60</td>
<td>34</td>
<td>704</td>
<td>4224</td>
<td>4928</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>11</td>
<td>9</td>
<td>44</td>
<td>1920</td>
<td>864</td>
<td>1440</td>
<td>96</td>
<td>80</td>
<td>97</td>
<td>1056</td>
<td>4224</td>
<td>5280</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>11</td>
<td>8</td>
<td>39</td>
<td>1920</td>
<td>688</td>
<td>1568</td>
<td>112</td>
<td>100</td>
<td>1232</td>
<td>5696</td>
<td>4928</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A. Method of Attack

Since we assume a cryptographic hash function used, even if the adversary gets $aux$, it does not give helpful information. Here, the adversary may get $PID$ and $K_1$. The best known attack that may be employed in this case is Information Set Decoding (ISD), which is used to find low weight vector.

Information-Set-Decoding: Information set decoding (ISD) is defined as the attack finding $e \in W_H(t)$ by searching a set of $n-k$ columns of $K$ containing the $t$ positions. Recently, an improvement to Stern's algorithm is proposed by Finiasz and Sendrier [12], which is one of the most efficient decoding techniques in cryptanalysis.

The best work factor against ISD is given as follows [12]

$$2l\min\{l, 2^{n-k}\} \cdot \left(\frac{k}{p}\right)^{1/2} \lambda^{-\frac{n-k-l}{l-p}} \cdot \sqrt{\frac{k+l}{p}}$$

where integer $p, l, k = 1 - e^{-1} \approx 0.63$.

Given two $PID$ ($PID', PID''$) output by same tag. The adversary XORs $PID'$ and $PID''$ and output $\Delta c_1$.

$$PID' \oplus PID'' = (c'_1 \oplus c_2') \oplus (c''_1 \oplus c_2')$$

And since $c_1 = msg \cdot K_1$

$$\Delta c_1 = (msg' \oplus msg'') \cdot K_1$$

If it is $t_1 < n_1$, The probability that Hamming weight of $\Delta msg''$ is $2t_1$ rises. The adversary outputs $\Delta msg''$ from $K_1$, $2t_1$, and $\Delta$. With ISD. This attacks is shown as following.

Let $l, p$ be integer, since $\frac{n_1}{2t_1} < 2^l$

$$2l\min\{l, 2^{n-k}\} \cdot \left(\frac{k}{p}\right)^{1/2} \lambda^{-\frac{n-k-l}{l-p}} \cdot \sqrt{\frac{k+l}{p}}$$

Structural Attack: Structural attack is the method that makes the structure of the code clear so as to find the private key from the public key. In this case, if there are 280 patterns of $\ell$ nonzero scale factors $\delta_0, \ldots, \delta_{\ell-1} \in GF(p)$ and $mt \times mt$ matrix which consists of $t \times t$ random dyadic matrix blocks to make $H$ random, the structure of the code is hidden securely.

We design the parameter that is secure against all above attacks.

VI. CONSIDERATION

We see that when $m$ becomes small, security increases and the size of the encryption key become small from the table. Since $n-k$ becomes small, the probability that all $t$ rows corresponding to the error in $n-k$ rows falls. The size of the encryption key stored in tag becomes around 4,928 bits. Considering a typical key size of Fix Domain Shrinking is 130,000bit[11], it is possible to reduce the key 1/26 times shorter in our proposal. Note that 1). Niederreiter PKC has a very fast and light-weight computation for encryption, i.e. bit-wise exclusive-OR only; 2). can recover the ID with one decryption on the server side. It is very exciting to reduce the stored key shorter!

VI. SUMMARY

In this paper, we propose an efficient authentication protocol for RFID system. Concretely, we show that the key needed in tags can be reduced to 4,928 bits from several hundred kilobits, by using our novel techniques. The protocol performs fast and needs only light-weight computation, without any exhaustive search nor synchronization in database, thus considered to be possibly implemented on low-power devices, such as RFID tags.

ACKNOWLEDGMENTS

We would like to thank Dr. T. Kitagawa for useful discussion.

REFERENCES