Published in IET Communications
Received on 12th July 2010
Revised on 26th February 2011
doi: 10.1049/iet-com.2010.0625

In Special Issue on Distributed Intelligence and Data Fusion for Sensor Systems

ISSN 1751-8628

Scalable and distributed key array authentication protocol in radio frequency identification-based sensor systems

H. Ning[1] *H. Liu*[1] *J. Mao*[1] *Y. Zhang*[2]

[1]*School of Electronic and Information Engineering, Beihang University, Beijing, People's Republic of China*
[2]*Simula Research Laboratory and Department of Informatics, University of Oslo, Norway*
E-mail: yanzhang@ieee.org

Abstract: Radio frequency identification (RFID)-based sensor systems are emerging as a new generation of wireless sensor networks by inherently integrating identification, sensing, communications and computation capabilities. Security and privacy are critical issues in dealing with a large amount of sensed data. In the study, the authors propose a distributed key array authentication protocol (KAAP) that provides classified security protection. KAAP is synthetically analysed in three aspects: logic, security and performance. The logic analysis includes messages formalisation, initial assumptions and anticipant goals based on GNY Logic formal method to verify the design correctness of the protocol. The security analysis with respect to confidentiality, integrity, authentication, anonymity and availability is performed via the simulated attacks, which involves supposing the attacker's identity, simulating the attacker's authentication process and creating compromised conditions. Such analysis ensures that the protocol has an ability to resist both external attacks (spoofing, replay, tracking and Denial of Service) and internal forgery attacks. Additionally, the performance is evaluated and compared with other related protocols to show that KAAP can improve the reliability and efficiency of sensor systems with insignificantly increased complexity. The result indicates that the protocol is reliable and scalable in advanced RFID-based sensor systems.

1 Introduction

Radio frequency identification (RFID) is an automatic identification technology as well as a promising technology to be inherently integrated into traditional wireless sensor networks (WSN). This will add a novel identification dimension into WSN and become RFID-based sensor systems (called RFID systems). RFID systems use the electromagnetic or electrostatic coupling to identify objects or persons through wireless channels where neither visual nor physical contact is needed for communication. In RFID systems, readers as sensor nodes are deployed for distributed tag data sensing, gathering and extraction in open unattended environments. Several uncertain factors during the system operations, among others' security and privacy become key issues. Instead of collecting detailed information towards the ambient environment, RFID systems pay more attention to the tagged item's information. The sensed tag data are closely related to the individual privacy and commercial benefit owing to a wide variety of applications from logistics to asset management. Thereafter, RFID systems differ from the traditional sensor systems, which suffer from more insecure situations and may be subject to more attacks in the open channels.

In RFID systems, both the reader-database link and the reader-tag link need be well safeguarded, so that reliable security authentication should be provided for spatially distributed and large-scale sensor networks. Several schemes are proposed for enforcing distributed intelligence [1–3] to provide necessary safeguard. However, most approaches focus on the threats from the external illegal attackers, but ignore the attacks from the internal legal entities. Furthermore, there is a lack of classified security protection for an overall management, which means that all the legal readers that can access all entire identifiers of all legal tags. It is essential for authenticated entities to access the specified field areas of a tag identifier (TID).

We take supply chain management as an example. There are various interest groups (e.g. material supplier, manufacturer, carrier and retailer). Each group owns its authorised readers who are permitted to access the authorised tag data, whereas any irrelevant sensitive data of other groups is not disclosed in public. In the scheme, the external attacks refer to the attacks from an external illegal entity such as a business competitor or an adversary who want to eavesdrop, intercept and spoof TID maliciously, and the internal attacks refer to the attacks from an internal legal entity who impersonates as another legal entity to do authority-exceeding violation. For instance, a manufacturer's reader disgusts a carrier's reader to access a tag. Both external and internal attacks may lead to security threats and privacy disclosure during the whole life cycle of tags.

Several security schemes have been proposed for handling potential security problems in RFID systems. As studied in

the previous researches, three main orders of magnitude are categorised towards security protocols. The ultra-lightweight protocols mainly involve bitwise logical operators to achieve the forward link security [4–6]. The lightweight protocols mainly use Cyclic Redundancy Code (CRC) operator for low-cost system [7, 8]. Besides, some schemes adopt the equipped hash function and pseudorandom number generator (PRNG) to realise identity authentication and access control [9–11]. Even some protocols are based on serverless systems [12, 13]. However, the protocols may not be efficient or robust enough owing to various security vulnerabilities [14]. The middleweight protocols fulfil the high-security requirement by symmetric key cryptography, and they refer to the protocols demanding support on tags for conventional cryptographic algorithms [15, 16]. However, most existing protocols focus on the external attacks without immunity to internal forgery attacks. Hence, it is necessary to propose an advanced scheme to achieve improved robustness, reliability and security.

In the paper, we propose a distributed key array authentication protocol (KAAP) for RFID-based sensor systems. All legal readers and tags are divided into different groups, as well as an overall management is performed to realise classified security protection and resist both external and internal attacks. The main contributions of our work are as follows.

1. Distributed key array acting as a memory cell stores all the authentication keys assigned to each reader group and tag group. The distributed key architecture has significance in two aspects: one is realising the classified security protection for a certain TID; the other is preventing an authority-exceeding violation by an internal legal entity. Besides, a shared key is used to resist an unauthorised access by an external illegal entity. Both the external and internal attacks are considered, along with which a multilevel privacy framework is used to mitigate the privacy disclosure.
2. Pseudorandom identifiers are transmitted instead of the real identifiers in open channels. Sensitive tag data are hidden unless both the reader and tag pass the mutual authentication so that the forward untraceability is realised.
3. Access lists are adopted to search a certain reader or tag in the storage, which as index-pseudonyms can conserve memory and improve scalability efficiently. Meanwhile, an intelligent judgement is performed via the access list to refuse malicious repeated queries.

The remainder of the paper is organised as follows. In Section 2, the related protocols and GNY Logic formal method are introduced briefly. The authentication progress of the protocol is described in Section 3. Then the logic analysis by GNY Logic is given in Section 4. The security analysis with regard to the external and internal attacks is studied in Section 5. The performance is analysed in Section 6. Section 7 draws a conclusion.

2 Related works

In the section, related works are presented in two aspects. We firstly review the typical security protocols for RFID systems, and then discuss the selected formal analysis method on security protocol verification.

2.1 Related security protocols

With regard to the RFID security and private issues, there are several security-enhancing protocols proposed for RFID systems. According to the security efficiency and operation complexity, the protocols (including access control, authentication and encryption) can be classified into three categories: ultra-lightweight, lightweight, middleweight.

The ultra-lightweight protocols mainly use the bitwise logical operators, CRC operator or other simple functions to achieve the air interface security. Peris-Lopez's LMAP protocol [4] bases on index-pseudonyms and simple bitwise operations to realise tag anonymity and integrity. Meanwhile the scheme is extremely lightweight and claimed to be secure against many attacks. Chien's SASI [6] builds pubic sub-messages via bitwise operations. Exclusive-or (XOR) operation is the main functional component that is needed, and a pseudonym is preshared as the search index to determine a matched record in the database. The possible de-synchronisation attack can be resisted for the freshness and dynamic update mechanism applied for mutual verification. Hopper and Blum's HB protocol [5], related to the computational hardness of learning parity with noise (LPN) problem instead of relying on classical symmetric key cryptography, is secure against passive attacks. Then a series of modified variants are designed to resist active attacks [17]. HB and its family protocols are secure under concurrence and parallel executions, and are proved to be suitable for pervasive computing environments. Additionally, some protocols using CRC operator are brought forward based on the ultrahigh-frequency passive tags. Sun's Gen2$^+$ protocol [7] uses the built-in CRC checksum operator as a verification function to authenticate readers based on shared pseudonyms. Random string keypool is shared between each tag and the database, where keys are randomly drawn for mutual authentication so that TID is never revealed until the reader passes all the challenges.

The lightweight protocols mainly use non-reversible hash function and PRNG to realise access control, in which the hashed value metal D or secured identifier are transmitted to prevent direct exposure of TID. Weis's Random hash-lock protocol (RHLP) [9] based on pseudorandom function hashes a long random string so that the messages are unfixed. It is claimed that the scheme possesses the properties of TID anonymity, and it can resist replay and Denial-of-Service (DoS) attacks using a random number generated in each session. Similarly, Henrici and Muller's Hash-chain protocol (HCP) [10] uses two deferent hash functions, one to refresh the tag secrets, the other to make tag untraceable. The database endures a heavy burden with great space-time memory consumption since TIDs are refreshed instantly. Rhee's Hash-based ID variation protocol (HIDVP) [11] keeping track of its session number is designed to oppose eavesdropping and replay attacks by diversifying values, which is not suitable for the distributed database environment. Additionally, Tan's protocol [12] and Ahamed's YA-SRAP [13] provide comparable protection without the need for a persistent central database. Acceptable security and scalability are guaranteed simultaneously.

The middleweight protocols mainly use full-fledged public key cryptography der to fulfil the high-security requirement, and they refer to the protocols demanding support on tags for conventional cryptographic algorithms. Lee's protocol [15] based on elliptic curve cryptographic scalar multiplications and general modular arithmetic proves high security, which can support multiple cryptographic protocols. Specially, a redundant modular operation is designed to obtain an efficient modulo arithmetic and

doi: 10.1049/iet-com.2010.0625

reduce the computational workload. ART project team selected Advanced Encryption Standard (AES) [16] as a cryptographic primitive for symmetric authentication in which an AES-128 algorithm proved that it is feasible for current RFID technology without major additional costs.

To summarise previous researches, bitwise operators, hash functions and encryption algorithms are more practical. Compared with other protocols, our proposed KAAP is intermediate between lightweight and middleweight category, which bases on a new authentication mode to ensure security management. It differs from the conventional security scheme and defines the authentication progress with flexible algorithms to prevent both the external and internal attacks. Meanwhile, it realises the classified security protection to enhance the reliability and scalability. Considering the limitations of tags, the specific realisation for the protocol may adopt lightweight and fixable algorithms to reduce hardware requirements.

2.2 GNY logic formal method

Formal analysis methods are essential for detecting subtle design flaws of cryptographic protocol. In the paper, GNY Logic [18] is introduced to analyse the design correctness and verify whether there are obvious design defects in KAAP theoretically.

GNY Logic as a direct successor of the BAN logic [19], is suitable for protocol verification owing to its comparative simplicity and effectiveness, where the desired protocol goals are deduced by applying a set of axioms and inference rules to the assumptions and message exchanges of the protocols. Several RFID security protocols have been proved by GNY Logic [20, 21]. The reasoning progress is based on knowledge/belief use postulates and definitions to analyse whether the protocol goals can be derived from the initial assumptions and message exchanges. If such derivation exists, the protocol will be regarded as reasonable. In comparison with other logic methods, GNY Logic is more elaborate since it has several improved rules. The virtues and limitations of GNY Logic have been discussed in [22]. GNY Logic introduces the freshness rules and keeps rigorous reasoning to improve the logical analysing ability so that it is quite powerful in its ability to uncover even subtle protocol flaws.

3 Protocol description

3.1 Main scheme

In the RFID-based sensor system, the air interface between readers and tags is suffering insecure wireless communication environments. In order to resolve such security and privacy problems from both the external and internal attacks, a distributed KAAP is refined on the strength of the protocol in [23] that authors apply key array to RFID applications and perform informal security analysis. Classified security protection and overall management are realised based on the distributed key architecture.

Suppose that I readers and J tags in the system. R_i and T_j indicate the ith reader and the jth tag. All readers and tags are divided into M and N groups, respectively, in which R_i belongs to the mth reader group G_{R_m} and T_j belongs to the nth tag group G_{T_n}. Different reader groups own relative independent authorities for tag groups, which means two reader groups may access the diverse field areas of the same TID. Here, R_i has access permission for the partial or entire field areas of TID of T_j. Two types of keys are used for encryption: the shared key and the authentication key. An unique shared key k_{u} is given to legal readers and tags, and a distributed key array $K_{M\times N}$ is assigned to store the authentication keys in the database DB. $k_{m,n}$ represents the authentication key assigned to G_{R_m} and G_{T_n}, which also points to the specific field areas of TID. It can decide whether a reader can access a tag, and what tag data can be accessed by a reader. Thereinto, T_j only keeps authentication keys in the nth row of the array. Note that M is much smaller than J, and authenticated keys are stored in a distributed mode by tags.

In the scheme, the requirements of hardware and software are given as follows. Tags considered in the system are smart cards comprising an intelligent micro-processor unit (MPU), storage units and chip-operating system (COS). Assume that tags have the basic crypto-operational and storage capabilities to realise ciphertext transmitted in the air interface. Readers are static or mobile active devices intelligently distributed to cover the area where tags exit. Both the readers and the database are not power constrained, and besides the database is regarded as the credible entity. The communication channel between a reader and the back-end database is assumed to be secure, while the wireless communication channel between a reader and a tag is vulnerable. No other additional features are considered for the participants.

3.2 System parameters

Table 1 shows the parameters applied in the protocol. Subscripts i, j and a are used to describe a particular reader, tag, attacker and their respective variables.

3.3 Authentication process

We use R_i and T_j to describe the authentication phases according to the sequence of message exchanges. The process is illustrated in Fig. 1. There are five message exchanges among R_i, T_j and DB.

- *Phase 1:* Challenge messages: R_i generates a random number r_{R_i} followed by concatenates r_{R_i} and its pseudorandom identifier PID_{R_i} to form $r_{R_i} \| \mathrm{PID}_{R_i}$, and then R_i performs encryption on $r_{R_i} \| \mathrm{PID}_{R_i}$ with the shared key k_{u}. Afterwards, R_i sends the message $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_{\mathrm{u}}}$ to T_j as an initial query.
- *Phase 2:* Respond messages: On receiving the query, T_j decrypts the ciphertext $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_{\mathrm{u}}}$ by k_{u}. When T_j obtains r_{R_i} and PID_{R_i}, T_j verifies R_i by searching PID_{R_i} in the access list L_R which is prestored in the memory. If it is valid, T_j will search the corresponding authentication key $k_{m,n}$ along generating a random number r_{T_j}. Thereafter, T_j encrypts $r_{R_i} \| r_{T_j}$ by $k_{m,n}$ to obtain $\{r_{R_i} \| r_{T_j}\}_{k_{m,n}}$, and continues to encrypt its pseudorandom identifier PID_{T_j} and $\{r_{R_i} \| r_{T_j}\}_{k_{m,n}}$ by k_{u} to obtain $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_{\mathrm{u}}}$. The new ciphertext will be responded to R_i. Otherwise, T_j will stop the authentication process with an error code.
- *Phase 3:* Forward messages: When R_i receives the response, it extracts PID_{T_j} from $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_u}$ with k_{u}. Then, R_i forwards PID_{T_j} to the database DB for further authentication.

Table 1 Notations

Notation	Description
G_{R_m}, G_{T_n}	the mth reader group and the nth tag group in the RFID-based sensor system, ($m = 1, 2, \ldots, M$; $n = 1, 2, \ldots, N$)
R_i	the ith reader who belongs to G_{R_m}, ($i = 1, 2, \ldots, I$; $I \geq M$)
T_j	the jth tag who belongs to G_{T_n}, ($j = 1, 2, \ldots, J$; $J \geq N$)
$\hat{R}_i$, $\hat{T}_j$	the reader/tag imitated by another internal legal reader/tag belonging to different groups
R_a, T_a	the reader/tag imitated by an external illegal attacker
DB	the back-end database
ID_{R_i}, ID_{T_j}	the identifiers of R_i and T_j
PID_{R_i}, PID_{T_j}	the pseudorandom identifiers of R_i and T_j, which have special flags to mark R_i and T_j
ID_{R_a}, ID_{T_a}	the imitative identifiers of R_a and T_a
$\mathrm{ID}_{T_{jx}}$	the specific (partial/entire) field areas of ID_{T_j}, $x = a, b, \ldots$
L_R	the access list for tags to retrieve a certain reader
L_T	the access list for the database to retrieve a certain tag
r_{R_i}, r_{T_j}	the general formats of random numbers for R_i and T_j
$r_{R_{i\mathrm{gen}}}$, $r_{T_{j\mathrm{gen}}}$	the random numbers generated by R_i and T_j in one session
$r_{R_{i\mathrm{com}}}$, $r_{T_{j\mathrm{com}}}$	the random numbers computed by R_i and T_j in one session
r'_{R_i}, r'_{T_j}	the random numbers generated by R_i and T_j in the next session
r_{R_a}, r_{T_a}	the random numbers generated by R_a and T_a in one session
k_{u}	the shared key is pre-shared between legal readers and tags, and it is a secure value without being revealed to the third entity
$K_{M\times N}$	the key array which stores all the authentication keys in DB
$k_{m,n}$	the authentication key owned R_i and T_j, which is assigned to G_{R_m} and G_{T_n}
$\|$	concatenate operator
$\rightarrow$	transition operator
$\{.\}_k$	encryption with key k
$\stackrel{?}{=}$	comparison operator

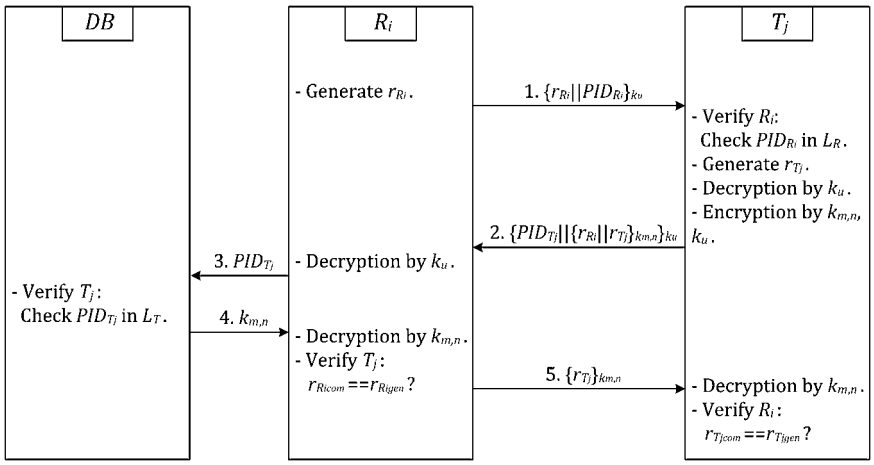

Fig. 1 *Distributed KAAP*

• *Phase 4:* Authentication by reader: While receiving PID_{T_j}, DB firstly retrieves PID_{T_j} to verify whether the message originates from a legal tag. If PID_{T_j} is acknowledged by DB, DB will deliver the corresponding authentication key $k_{m,n}$ to R_i. Then, R_i decrypts $\{r_{R_i} \| r_{T_j}\}_{k_{m,n}}$ to obtain r_{R_i} and r_{T_j} with $k_{m,n}$. R_i checks whether the current computed r_{R_i} equals the previous generated r_{R_i} in Phase 1. If the two values are identical, T_j will be successfully authenticated by R_i. Then R_i continues to encrypt r_{T_j} to obtain $\{r_{T_j}\}_{k_{m,n}}$, and forwards the ciphertext to T_j. Otherwise, T_j is considered as an imitative entity, and R_i will stop the authentication process with an error code.

• *Phase 5:* Authentication by tag: While receiving $\{r_{T_j}\}_{k_{m,n}}$, T_j extracts r_{T_j} by decryption. T_j checks whether the current computed r_{T_j} equals the previous generated r_{T_j} in Phase 2. If the two values are identical, R_i will be successfully authenticated by T_j. Otherwise, R_i will be considered as an imitative entity, and T_j will stop the authentication with an error code. Thus, the entire authentication is accomplished. T_j will send the authorised field areas of identifier $\mathrm{ID}_{T_{jx}}$ to R_i. The authentication phases (P) can be described as follows:

P1 $(R_i \rightarrow T_j)$: $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_{\mathrm{u}}}$;

P2 $(T_j \rightarrow R_i)$: $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_{\mathrm{u}}}$;

P3 $(R_i \rightarrow \mathrm{DB})$: PID_{T_j};

P4 $(\mathrm{DB} \rightarrow R_i)$: $k_{m,n}$, $r_{R_{i\mathrm{com}}} \stackrel{?}{=} r_{R_{i\mathrm{gen}}}$;

P5 $(R_i \rightarrow T_j)$: $\{r_{T_j}\}_{k_{m,n}}$, $r_{T_{j\mathrm{com}}} \stackrel{?}{=} r_{T_{j\mathrm{gen}}}$.

doi: 10.1049/iet-com.2010.0625

The protocol based on a distributed key array adopts a challenge-response mutual authentication mechanism and a random access control mechanism to enhance security. The main approaches include:

1. Two types of secret keys are adopted in the authentication. The shared key k_u is used to protect the PID_{R_i} and PID_{T_j} against external illegal entities. The distributed key array $K_{M\times N}$, where authentication key $k_{m,n}$ is allocated to G_{T_n} and G_{R_m}. Thereinto, $k_{m,n}$ has significance in two aspects. One is realising the classified security protection of a TID. Readers are assigned into different authorities, which assures specified field areas of TID can and only can be accessed by authorised readers; the other is preventing an authority-exceeding violation by a legal entity in one group forging another group's entity to violate privacy or skim-sensitive data.
2. Pseudorandom identifiers (PID_{R_i}, PID_{T_j}) are transmitted instead of the real identifiers (ID_{R_i}, ID_{T_j}). Moreover, the random numbers (r_{R_i}, r_{T_j}) are generated, respectively, which are independent variables to assure dynamic refresh in each session.
3. Access lists (L_R, L_T) are used to retrieve a certain reader or a tag in the storage. For instance, the pseudorandom identifier PID_{R_i} is sent as the search request for verifying authenticity, and T_j checks L_R for matching entry that has the same bits as PID_{R_i}. The access lists as index-pseudonyms effectively eliminate the retrieve workload and enable more scalable for dynamic systems.

Note that fixable cryptographic algorithms may be used to guard the authentication reliability and security. The distributed key array $K_{M\times N}$ and access lists need to be maintained and updated periodically. Additionally, reserved cells are allocated in the key array for group extension; the sizes of the authentication keys may vary considerably according to different group requirements; key update refers to the cryptographic technique in the traditional networks; flags (e.g. time stamp) can be added to deal with the synchronisation of the key update.

During the authentication progress, two main functional mechanisms make the protocol more intelligent.

1. *Access list control:* L_R and L_T are used for preliminary check besides quick search. T_j and DB maintain a list of pseudorandom identifiers and their associated rules, which enables the self-adapting addressing mode to adopt. The corresponding retrieve rules can be retained and stored by previous learning.
2. *Random access control:* T_j and R_i extract and store the last received random numbers or pseudorandom identifiers as temp lists. If a query arrives with the same data frames within a certain time, there will be no response to the repeated query. The intelligent judgement fosters recognition capability even during malicious replaying or jamming attacks.

In summary, KAAP is an advanced security-preserving authentication scheme, based on a distributed key array, in which TID is never exposed in plain form. It is significant for ranking various authorities to realise classified security protection.

3.4 Case study

The subsection presents a case study of supply chain management to illustrate how the proposed KAAP is implemented in a potential application scenario.

Suppose that I readers $\{R_1, R_2, \ldots, R_I\}$ and J tags $\{T_1, T_2, \ldots, T_J\}$ in the whole supply chain system. Readers are divided into four main interest groups, including material supplier, manufacturer, carrier and retailer. Tags fall into three groups: general, confidential and classified groups. R_i belongs to the third reader group (i.e. R_i is a carrier's reader) and T_j belongs to the second tag group (i.e. T_j is a confidential tag). A shared key k_u is given to all legal readers and tags. A distributed key array $K_{4\times 3}$ is assigned to store all the authentication keys in DB, in which an authentication key $k_{3,2}$ represents the authentication key assigned to the carrier and confidential tag, which also points to the field areas $ID_{T_{jc}}$ for T_j. Different reader groups own relative independent authorities for tag groups, which means: (i) even if a manufacturer's reader cannot access T_j, R_i may own the access authority; (ii) a manufacturer's reader and R_i may read different tag data of the same tag T_j. The specific access authority and access contents depend on the authentication key $k_{2,3}$ for R_i and T_j. Fig. 2 shows the distributed key array architecture of the supply chain system.

The proposed KAAP is penetrated in each link of the supply chain management. In a scenario, R_i as a carrier's reader wants to access T_j which is a confidential tag. R_i and T_j perform a four-round verification in an authentication session.

1. R_i sends the encrypted query to T_j, thereinto k_u is used to resist an external illegal reader owned by another commercial rival. Then, T_j verify R_i by searching PID_{R_i} in the access list L_R so that it makes preliminary confirmation whether it is responding to an external adversary or to a legal reader within the system.
2. DB judges whether the receiving pseudorandom identifier PID_{T_j} is owned by a legal tag within the system. If a commercial rival forges a tag to deceive the carrier, DB will recognise the external illegal tag by the absense of matching entry in the access list L_T.
3. R_i performs decryption by k_u to obtain the computed random number r_{R_i}, and it compares the computed $r_{R_{i\text{com}}}$ with the former generated $r_{R_{i\text{gen}}}$ to verify whether T_j belongs to the desired confidential group. If a general or classified tag disguises the confidential tag, R_i will find the authority-exceeding violation by an internal legal reader.
4. T_j performs the similar operations to verify whether R_i is a carrier's reader as its declaration. If a material supplier or manufacturer's reader disguises the carrier's reader, T_j will find the authority-exceeding violation by an internal legal tag.

In the case, privacy protection is enhanced by the distributed key array in two aspects. On one hand, all the readers are assigned into different groups (i.e. material supplier, manufacturer, carrier and retailer), which own discrepant authorities assuring specified field areas of a TID only can be accessed by certain readers. The specified field

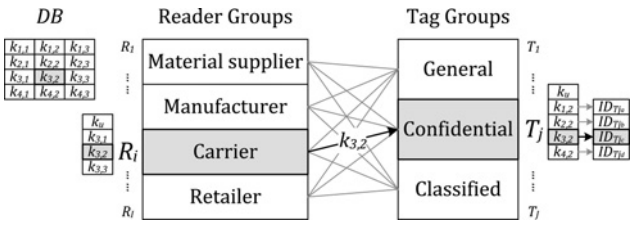

Fig. 2 *Distributed key array architecture of the supply chain system*

areas are decided by detailed types of tag groups (i.e. general, confidential or classified groups). On the other hand, an authority-exceeding violation by an internal legal non-carrier's reader is prevented by the authentication keys, along with an unauthorised access by an external illegal entity is resisted by the shared key. Thus, the classified privacy is protected.

If and only if all above authentication phases are passed, T_j will send the authorised field areas $\mathrm{ID}_{T_{j_c}}$ to R_i. Other unrelated or unauthorised field areas $\{\mathrm{ID}_{T_{j_a}}, \mathrm{ID}_{T_{j_b}}, \mathrm{ID}_{T_{j_d}}\}$ are not exposed. Additionally, a more detailed key array can be minutely stipulated within each group for further classified security protection.

4 Formal analysis of authentication protocol with GNY logic

Generally, authentication protocols have been designed and proved using informal methods. However, the absence of formal analysis may lead to concealed security vulnerabilities undetected. According to the proposed distributed KAAP, basic security verification has been described in the intuitive way. However, design flaws and security errors may be ignored by the informal method. In the section, GNY Logic [18] is applied to analyse the design correctness of protocols. It provides rigorous and thorough means of evaluating the protocol so that even subtle defects can be uncovered. With the formal method, a protocol can be proved to be reasonable by achieving the protocol goals using logical postulates.

The GNY Logic-based formal analysis includes two aspects, the logic analysis and the logic verification. The analysis involves the following steps: (i) formalisation of the protocol messages; (ii) declaration of initial assumptions; (iii) declaration of anticipant goals; (iv) verification by logical rules and formulas.

4.1 Logic analysis

4.1.1 Formalisation of messages: Formalisation of the protocol messages refers to specifying the protocol in the language of GNY Logic by expressing each message exchange as a logical formula. Table 2 shows notations to facilitate the formal descriptions.

According to the authentication phases, the formalised messages (M) delivered between R_i, T_j and DB can be obtained; a formalised version of protocol is as follows:

M1 $(R_i \rightarrow T_j)$: $T_j \triangleleft * \{\mathrm{PID}_{R_i}\}_{k_\mathrm{u}}$, $T_j \triangleleft * \{r_{R_i}\}_{k_\mathrm{u}}$;

M2 $(T_j \rightarrow R_i)$: $R_i \triangleleft * \{\mathrm{PID}_{T_j}\}_{k_\mathrm{u}}$, $R_i \triangleleft * \{r_{R_i}\}_{k_\mathrm{u},k_{m,n}}$,

$R_i \triangleleft * \{r_{T_j}\}_{k_\mathrm{u},k_{m,n}}$;

M3 $(R_i \rightarrow \mathrm{DB})$: $\mathrm{DB} \triangleleft * \mathrm{PID}_{T_j}$;

M4 $(\mathrm{DB} \rightarrow R_i)$: $R_i \triangleleft * k_{m,n}$;

M5 $(R_i \rightarrow T_j)$: $T_j \triangleleft * \{r_{T_j}\}_{k_{m,n}}$.

4.1.2 Initial assumptions: The subsection specifies the initial possessions and abilities of each participant. The following initiative assumptions (IA) can be obtained:

- For T_j:

IA1.1: $T_j \ni r_{T_j}$;
IA1.2: $T_j \ni \mathrm{PID}_{T_j}$, $T_j| \equiv \sharp \mathrm{PID}_{R_i}$;
IA1.3: $T_j \ni k_\mathrm{u}$, $T_j| \equiv \sharp k_\mathrm{u}$, $T_j| \equiv T_j \overset{k_\mathrm{u},\, k_{m,n}}{\longleftrightarrow} R_i$;
IA1.4: $T_j \ni k_{m,n}$, $T_j| \equiv \sharp k_{m,n}$, $T_j| \equiv T_j \overset{k_{m,n}}{\longleftrightarrow} \mathrm{DB}$.

These expressions indicate that: T_j possesses r_{T_j}, PID_{T_j}, k_u and $k_{m,n}$; T_j believes that k_u and $k_{m,n}$ are fresh, and T_j is entitled to believe that PID_{R_i} is fresh; T_j believes k_u and $k_{m,n}$ are suitable secrets for T_j and R_i; T_j believes $k_{m,n}$ is a suitable secret for T_j and DB.

- For R_i:

IA2.1: $R_i \ni r_{R_i}$;
IA2.2: $R_i \ni \mathrm{PID}_{R_i}$, $R_i| \equiv \sharp \mathrm{PID}_{T_j}$;
IA2.3: $R_i \ni k_\mathrm{u}$, $R_i| \equiv \sharp k_\mathrm{u}$, $R_i| \equiv R_i \overset{k_u,\, k_{m,n}}{\longleftrightarrow} T_j$.

These expressions indicate that: R_i possesses r_{R_i}, PID_{R_i} and k_u; R_i believes that k_u is fresh, and R_i is entitled to believe that PID_{T_j} is fresh; R_i believes that k_u and $k_{m,n}$ are suitable secrets for T_j and R_i.

- For DB:

IA3: $\mathrm{DB} \ni k_{m,n}$, $\mathrm{DB}| \equiv \#k_{m,n}$, $\mathrm{DB}| \equiv \mathrm{DB} \overset{k_{m,n}}{\longleftrightarrow} T_j$.

These expressions indicate that: DB possesses $k_{m,n}$; DB believes that $k_{m,n}$ is fresh; DB believes that $k_{m,n}$ is a suitable secret for DB and T_j.

Table 2 Symbol notations

Notation	Description
$P \triangleleft X$	P receives a message containing X, P can read and repeat X
$P \triangleleft {}^* X$	P receives X, X is a not-originated-here formula
$P \ni X$	P possesses, or is capable of possessing X
$P\| \sim X$	P once conveyed X
$P\| \equiv \sharp X$	P believes, or is entitled to believe that X is fresh
$P\| \equiv \phi X$	P believes, or is entitled to believe that X is recognisable
$P\| \equiv C$	P believes, or would be entitled to believe, that statement C holds
$P\| \Rightarrow C$	P is an authority on statement C, and has jurisdiction over C
$P\| \equiv P \overset{K}{\longleftrightarrow} Q$	P believes, or is entitled to believe, that K is a suitable secret for P and Q
$\{X\}_K$	symmetric encryption
$\{X\}_{K^{-1}}$	symmetric decryption
(X, Y)	concatenation

doi: 10.1049/iet-com.2010.0625

In summary, each principal possesses its random number, the pseudorandom identifier, or the secret shared/authentication key(s). Each principal possesses secret values. Principals believe or are entitled to believe that the secret values are fresh, along with all the principals believe or are entitled to believe that all the keys are suitable secrets.

4.1.3 Anticipant goals: The objectives of the protocol are the belief and freshness of data among R_i, T_j and DB. It guarantees that the messages are from trustable entities and were not used in former sessions. The following anticipant goals (G) can be obtained

G1: $T_j| \equiv R_i| \sim r_{R_i}$,

T_j believes that R_i conveyed r_{R_i}.

G2: $T_j| \equiv R_i| \sim \text{PID}_{R_i}$,

T_j believes that R_i conveyed PID_{R_i}.

G3: $R_i| \equiv T_j| \sim r_{R_i}$,

R_i believes that T_j conveyed r_{R_i}.

G4: $R_i| \equiv T_j| \sim \text{PID}_{T_j}$,

R_i believes that T_j conveyed PID_{T_j}.

G5: $\text{DB}| \equiv T_j| \sim \text{PID}_{T_j}$,

DB believes that T_j conveyed PID_{T_j}.

G6: $R_i| \equiv \text{DB} \overset{k_{m,n}}{\longleftrightarrow} T_j$,

R_i believes the authentication key $k_{m,n}$ between DB and T_j.

G7: $T_j| \equiv \sharp\{\text{PID}_{R_i}\}_{k_u}$,

T_j believes that $\{\text{PID}_{R_i}\}_{k_u}$ is fresh.

G8: $R_i| \equiv \sharp\{\text{PID}_{T_j}\}_{k_u}$,

R_i believes that $\{\text{PID}_{T_j}\}_{k_u}$ is fresh.

The first to the sixth goal indicate belief requirements. Messages are sent from legal participants instead of malicious attackers. The seventh and the eighth goal indicate freshness requirements. The received messages were not used by malicious attackers in the previous sessions.

4.2 Logic verification

The logic verification is based on the initial assumptions, protocol messages and related rules provided by GNY Logic [18].

- For G1 and G2: From M1, T_j is informed not-originated-here messages $\{\text{PID}_{R_i}\}_{k_u}$ and $\{r_{R_i}\}_{k_u}$. T_j has not received or sent them in the previous sessions, we have

$$T_j \triangleleft *\{r_{R_i}\}_{k_u}, \quad T_j \triangleleft *\{\text{PID}_{R_i}\}_{k_u} \tag{1}$$

Applying the Being-Told Rule T1: $(P \triangleleft (*X))/(P \triangleleft X)$ yields

$$T_j \triangleleft \{r_{R_i}\}_{k_u}, \quad T_j \triangleleft \{\text{PID}_{R_i}\}_{k_u} \tag{2}$$

Thus, T_j receives $\{r_{R_i}\}_{k_u}$ and $\{\text{PID}_{R_i}\}_{k_u}$. R_i can derive the truth and acknowledge the received messages without the not-originated-here asterisk.

Applying IA1.3 and the Being-Told Rule T3: $(P \triangleleft \{X\}_K, P \ni K)/(P \triangleleft X)$ yields

$$T_j \triangleleft r_{R_i}, \quad T_j \triangleleft \text{PID}_{R_i} \tag{3}$$

Thus, T_j is considered to have been informed the decrypted contents of r_{R_i} and PID_{R_i} with the shared key k_u.

Applying the Possession Rule P1: $(P \triangleleft X)/(P \ni X)$ yields

$$T_j \ni r_{R_i}, T_j \ni \text{PID}_{R_i} \tag{4}$$

Thus, T_j is capable of possessing anything that it has been informed, and it possesses r_{R_i} and PID_{R_i}.

Applying the Possession Rule P4: $(P \ni X)/(P \ni H(X))$ yields

$$T_j \ni H(r_{R_i}), T_j \ni H(\text{PID}_{R_i}) \tag{5}$$

Thus, T_j is capable of possessing the one-way computationally feasible functions $H(r_{R_i})$ and $H(\text{PID}_{R_i})$.

Applying the Reconcilability Rule R6: $(P \ni H(X))/(P| \equiv \phi(X))$ yields

$$T_j| \equiv \phi(r_{R_i}), T_j| \equiv \phi(\text{PID}_{R_i}) \tag{6}$$

Thus, T_j is entitled to believe that r_{R_i} and PID_{R_i} are recognisable.

Applying IA1.3: $T_j| \equiv \sharp k_u$ and the Freshness Rule F1

$$\frac{P| \equiv \sharp(X)}{P| \equiv \sharp(X, Y), P| \equiv \sharp(F(X))}$$

yields

$$T_j| \equiv \sharp(r_{R_i}, k_u), T_j| \equiv \sharp(\text{PID}_{R_i}, k_u) \tag{7}$$

Thus, T_j is entitled to believe that (r_{R_i}, k_u) and (PID_{R_i}, k_u) are fresh.

Applying the Message Interpretation Rule I1

$$\frac{P \triangleleft *\{X\}_K, P \ni K, P| \equiv P \overset{K}{\longleftrightarrow} Q, P| \equiv \phi(X), P| \equiv \sharp(X, K)}{P| \equiv Q| \sim X, P| \equiv Q| \sim \{X\}_K, P| \equiv Q \ni K}$$

yields

$$T_j| \equiv R_i| \sim r_{R_i}, \quad T_j| \equiv R_i| \sim \text{PID}_{R_i} \tag{8}$$

As a consequence, T_j believes that R_i once conveyed r_{R_i} and PID_{R_i}.

- For G3: From M4, R_i is informed a not-originated-here message $k_{m,n}$.

$$R_i \triangleleft *k_{m,n} \tag{9}$$

Applying T1: $(P \triangleleft (*X))/(P \triangleleft X)$, R_i receives $k_{m,n}$. Thus, R_i

can read and repeated $k_{m,n}$.

$$R_i \triangleleft k_{m,n} \tag{10}$$

Applying P1: $(P \triangleleft X)/(P \ni X)$, R_i possesses $k_{m,n}$.

$$R_i \ni k_{m,n} \tag{11}$$

From M2, R_i is informed a not-originated-here message $\{r_{T_j}\}_{k_u,k_{m,n}}$.

$$R_i \triangleleft *\{r_T\}_{k_u,k_{m,n}} \tag{12}$$

Applying T1: $(P \triangleleft (*X))/(P \triangleleft X)$, R_i receives $\{r_{T_j}\}_{k_u,k_{m,n}}$. Thus, R_i can read and repeat the ciphertext.

$$R_i \triangleleft \{r_{T_j}\}_{k_u,k_{m,n}} \tag{13}$$

According to IA2.3: $R_i \ni k_u$ and T3: $(P \triangleleft \{X\}_K, P \ni K)/(P \triangleleft X)$, if R_i is informed $\{r_{T_j}\}_{k_u,k_{m,n}}$, along with R_i possesses k_u and $k_{m,n}$, R_i will be believed to have been informed the decrypted content of r_{T_j}.

$$R_i \triangleleft r_{T_j} \tag{14}$$

Applying P1: $(P \triangleleft X)/(P \ni X)$, R_i is capable of possessing anything that it is informed, and so R_i possesses the random number r_{T_j}.

$$R_i \ni r_{T_j} \tag{15}$$

Applying P4: $(P \ni X)/(P \ni H(X))$, if R_i possesses r_{T_j}, R_i will be capable of possessing a one-way computationally feasible function $H(r_{T_j})$.

$$R_i \ni H(r_{T_j}) \tag{16}$$

Applying R6: $(P \ni H(X))/(P| \equiv \phi(X))$, if R_i possesses $H(r_{T_j})$, R_i will be entitled to believe that r_{T_j} is recognisable.

$$R_i| \equiv \phi(r_{T_j}) \tag{17}$$

Applying IA2.3: $R_i| \equiv \sharp k_u$ and F1

$$\frac{P| \equiv \sharp(X)}{P| \equiv \sharp(X, Y), P| \equiv \sharp(F(X))}$$

if R_i believes that k_u is fresh, R_i will be entitled to believe that any message of which k_u is a component is fresh. Thus, R_i is entitled to believe that r_{T_j}, k_u and $k_{m,n}$ are fresh.

$$R_i| \equiv \sharp(r_{T_j}, k_u, k_{m,n}) \tag{18}$$

As a consequence,

1. $R_i \triangleleft *\{r_{T_j}\}_{k_u,k_{m,n}}$: R_i receives a message consisting of encrypted with k_u and $k_{m,n}$, and the message is signed with a not-originated-here mark;
2. $R_i \ni k_u$, $R_i \ni k_{m,n}$: R_i possesses k_u and $k_{m,n}$;
3. $R_i| \equiv R_i \overset{k_u,\ k_{m,n}}{\longleftrightarrow} T_j$: R_i believes that k_u and $k_{m,n}$ are suitable secrets for itself and T_j;
4. $R_i| \equiv \phi(r_{T_j})$: R_i believes that r_{T_j} is recognisable;
5. $R_i| \equiv \sharp(r_{T_j}, k_u, k_{m,n})$: R_i believes that r_{T_j}, k_u and $k_{m,n}$ are fresh.

According to I1

$$\frac{P \triangleleft *\{X\}_K, P \ni K, P| \equiv P \overset{K}{\longleftrightarrow} Q, P| \equiv \phi(X), P| \equiv \sharp(X, K)}{P| \equiv Q| \sim X, P| \equiv Q| \sim \{X\}_K, P| \equiv Q \ni K}$$

R_i is entitled to believe that T_j once conveyed r_{T_j}.

$$R_i| \equiv T_j| \sim r_{T_j} \tag{19}$$

- For G4: We can deduce that

$R_i \triangleleft *\{\text{PID}_{T_j}\}_{k_u}$	//by M2
$R_i \triangleleft \{\text{PID}_{T_j}\}_{k_u}$	//by T1
$R_i \triangleleft \text{PID}_{T_j}$	//by IA2.3, T3
$R_i \ni \text{PID}_{T_j}$	//by P1
$R_i \ni H(\text{PID}_{T_j})$	//by P4
$R_i| \equiv \phi(\text{PID}_{T_j})$	//by R6
$R_i| \equiv \sharp(\text{PID}_{T_j}, k_u)$	//by IA2.3, F1
$R_i| \equiv T_j| \sim \text{PID}_{T_j}$	//by I1

As a consequence,

1. $R_i \triangleleft *\{\text{PID}_{T_j}\}_{k_u}$: R_i receives a formula consisting of PID_{T_j} encrypted with keys k_u, and the message is signed with a not-originated-here mark;
2. $R_i \ni k_u$: R_i possesses k_u;
3. $R_i| \equiv R_i \overset{k_u}{\longleftrightarrow} T_j$: R_i believes that k_u is a suitable secret for himself and T_j;
4. $R_i| \equiv \phi(\text{PID}_{T_j})$: R_i believes that PID_{T_j} is recognisable;
5. $R_i| \equiv \sharp(\text{PID}_{T_j}, k_u)$: R_i believes that PID_{T_j} and k_u are fresh.

According to I1, R_i is entitled to believe that T_j once conveyed PID_{T_j}.

- For G5: We can deduce that

$\text{DB} \triangleleft *\text{PID}_{T_j}$	//by M3
$\text{DB} \triangleleft \text{PID}_{T_j}$	//by T1
$\text{DB} \ni \text{PID}_{T_j}$	//by P1
$\text{DB} \ni H(\text{PID}_{T_j})$	//by P4
$\text{DB}| \equiv \phi(\text{PID}_{T_j})$	//by R6
$\text{DB}| \equiv \sharp(\text{PID}_{T_j}, k_{m,n})$	//by IA3, F1
$\text{DB}| \equiv T_j| \sim \text{PID}_{T_j}$	//by I1

As a consequence,

1. $\text{DB} \triangleleft *\text{PID}_{T_j}$: DB receives PID_{T_j}, and the message is signed with a not-originated-here mark;
2. $\text{DB} \ni k_{m,n}$: DB possesses $k_{m,n}$;

doi: 10.1049/iet-com.2010.0625

3. DB| $\equiv$ DB $\overset{k_{m,n}}{\longleftrightarrow} T_j$: DB believes that $k_{m,n}$ is a suitable secret for itself and T_j;
4. DB| $\equiv \phi(\mathrm{PID}_{T_j})$: DB believes that PID_{T_j} is recognisable;
5. DB| $\equiv \sharp(\mathrm{PID}_{T_j}, k_{m,n})$: DB believes that PID_{T_j} and $k_{m,n}$ are fresh.

According to I1, DB is entitled to believe that T_j once conveyed PID_{T_j}.

- For G6: From the secure communication channel between R_i and DB, we have

$$R_i| \equiv \mathrm{DB}| \Rightarrow (\mathrm{DB}| \equiv *) \tag{20}$$

From IA3: DB| $\equiv$ DB $\overset{k_{m,n}}{\longleftrightarrow} T_j$, we have

$$R_i| \equiv \mathrm{DB}| \Rightarrow (\mathrm{DB} \overset{k_{m,n}}{\longleftrightarrow} T_j) \tag{21}$$

Thus, R_i believes that DB is honest and competent, and DB believes that the authentication key $k_{m,n}$ owned by DB and T_j is honest.
Applying the Jurisdiction Rule J3

$$\frac{P| \equiv Q| \Rightarrow (Q| \equiv *), P| \equiv Q| \equiv (Q| \equiv C)}{P| \equiv Q| \equiv C}$$

yields

$$R_i| \equiv \mathrm{DB}| \equiv (\mathrm{DB} \overset{k_{m,n}}{\longleftrightarrow} T_j) \tag{22}$$

Thus, R_i believes that *DB* believes that the authentication key $k_{m,n}$ owned by DB and T_j is honest.
Applying the Jurisdiction Rule J1

$$\frac{P| \equiv Q| \Rightarrow C, P| \equiv Q| \equiv C}{P| \equiv C}$$

yields

$$R_i| \equiv \mathrm{DB} \overset{k_{m,n}}{\longleftrightarrow} T_j \tag{23}$$

As a consequence, R_i is entitled to believe that the authentication key $k_{m,n}$ between DB and T_j is creditable.

- For G7 and G8: From IA1.2 and IA1.3: $T_j| \equiv \sharp \mathrm{PID}_{R_i}$, $T_j \ni k_u$.From IA2.2 and IA2.3: $R_i| \equiv \sharp \mathrm{PID}_{T_j}$, $R_i \ni k_u$.
Applying the Freshness Rule F2

$$\frac{P| \equiv \sharp(X), P \ni K}{P| \equiv \sharp\{X\}_K, P| \equiv \sharp\{X\}_{K^{-1}}}$$

if T_j believes that PID_{R_i} is fresh, along with T_j possesses k_{u}, T_j will be entitled to believe that the ciphertext $\{\mathrm{PID}_{R_i}\}_{k_{\mathrm{u}}}$ is fresh. Similarly, R_i will be entitled to believe that the ciphertext $\{\mathrm{PID}_{T_j}\}_{k_{\mathrm{u}}}$ is fresh, we have

$$T_j| \equiv \sharp\{\mathrm{PID}_{R_i}\}_{k_{\mathrm{u}}}, R_i| \equiv \sharp\{\mathrm{PID}_{T_j}\}_{k_{\mathrm{u}}} \tag{24}$$

As a consequence, T_j is entitled to believe that $\{\mathrm{PID}_{R_i}\}_{k_{\mathrm{u}}}$ is fresh, and R_i is entitled to believe that $\{\mathrm{PID}_{T_j}\}_{k_{\mathrm{u}}}$ is fresh.

The formal verification using GNY Logic is demonstrated above. In KAAP, the protocol goals can be derived from the initial assumptions and authentication phases. Readers and tags can build beliefs in the mutual authentication; furthermore, messages are sent between credible participants in communication sessions. The protocol is proved to be correct and can ensure the non-existence of obvious design defects.

5 Security analysis

In RFID systems, the backward link between readers and the database which refers to the traditional network security domain is assumed in secure. However, the forward link is confronting more serious situations. In the paper, physical destructions such as removing a tag physically from a tagged item are not considered for there are no technical solutions to discriminate between intentional or unintentional behaviours and few countermeasures to address them. We consider the non-physical attacks that can be grouped into three categories labelled as Mimic, Gather and DoS. Mimic (e.g. spoofing, cloning) owns the purpose of impersonating identity to unauthorised access. Gather (e.g. skimming, eavesdropping, data tampering) aims at acquiring sensitive data. DoS refers to traditional DoS attack, unauthenticated killing and jamming to interdict the communication link. These attacks that have severe impacts on the design of a robust authentication protocol may be launched by a strong intruder in either passive or active mode.

5.1 Attack model

Several attacks have been briefly given in [23]. Here, we do further research on the external attacks (spoofing, replay, tracking and DoS) and the internal forgery attacks to analyse the security. KAAP is analysed in the following the mode to evaluate whether it satisfies the security requirements: (i) suppose the attacker's identity; (ii) simulate how the attack is performed by an attacker by steps; (iii) create compromised conditions and deduce the security.

The protocol implements a cryptographic algorithm to establish credence. It is assumed that the shared key and the authentication keys can merely be owned by special legal entities. Furthermore, the backward link between a reader and the database is secure, and an attacker cannot replicate a reader or a tag, which is a reasonable assumption since it is always possible to resistant tamper by hardware.

5.1.1 Spoofing attack: Spoofing attack is a risk with access control systems. It is technically feasible that an attacker forges a legal reader to get the access authority, obtains the contents of tags and damages the normal communication. It is also a type of malicious exploiture of open channels whereby unauthorised access commands. For instance, the attacker pretends as a reader or tag which tries to obtain valid responses to cheat the legal entities.

Under the spoofing attack, an attacker A performs the following actions:

- In one session:

$\mathcal{A}(R_a) \rightarrow T_j$: $r_{R_a} \parallel \mathrm{ID}_{R_a}$.

T_j: Verify R_a.

$T_j \not\Rightarrow$: Authentication will fail.

- In the worse conditions:

$T_j \rightarrow \mathcal{A}(R_a)$: $\{\mathrm{PID}_{T_j} \| \{r_{R_a} \| r_{T_j}\}_{k_{*,j}}\}_{k_u}$.

- In the next session:

$R_i \rightarrow \mathcal{A}(T_a) \nrightarrow T_j$: $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$;

$\mathcal{A}(T_a) \rightarrow R_i$: $\{\mathrm{PID}_{T_j} \| \{r_{R_a} \| r_{T_j}\}_{k_{*,j}}\}_{k_u}$;

$R_i \rightarrow$ DB: PID_{T_j};

DB $\rightarrow R_i$: $k_{*,j}$;

$R_i \cdot r_{R_a} \neq r_{R_i}$;

$R_i \not\Rightarrow$: Authentication will fail.

1. In one session, $\mathcal{A}$ disguises as a reader R_a, and sends plaintext $r_{R_a} \| \mathrm{ID}_{R_a}$ to T_j. T_j receives the imitated message, it may neglect the malformed data and obtain r_{R_a} and ID_{R_a} directly. T_j verifies ID_{R_a} whether it is valid by searching its data fields in an access list L_R. The result will be no suitable ID_{R^*} exiting, and the authentication will be end of failure with an error code.
2. In the worse conditions, T_j may respond $\{\mathrm{PID}_{T_j} \| \{r_{R_a} \| r_{T_j}\}_{k_{*,j}}\}_{k_u}$ by mistake, in which $k_{*,j}$ is randomly selected, the authentication will continue.
3. In the next session, $\mathcal{A}$ disguises as a tag T_a to intercept $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$ sent to T_j. T_a responds R_i with $\{\mathrm{PID}_{T_j} \| \{r_{R_a} \| r_{T_j}\}_{k_{*,j}}\}_{k_u}$. On receiving the response, R_i will decrypt the ciphertext and forward PID_{T_j} to DB for further verification. DB verifies that PID_{T_j} is valid and sends $k_{*,j}$ to R_i. R_i decrypts $\{r_{R_a} \| r_{T_a}\}_{k_{*,j}}$ by $k_{*,j}$ to obtain r_{R_a} and r_{T_a}. Then R_i will find that r_{R_a} differs from r_{R_i} since the probability that r_{R_a} equals r_{R_i} is negligible.

Similarly, if $\mathcal{A}$ disguises as a tag T_a, it will also not pass the authentication by DB since there is no suitable PID_{T^*} in the access list L_T. In KAAP, access lists are available for preliminary verifications via quick searches. Meanwhile, random numbers are valid operators for one time. Hence, the protocol can resist the spoofing attack and an unauthentic third party cannot be granted to access the TID.

5.1.2 Replay attack: Replay attack means that an attacker impersonates a legal entity to involve into the communication to insert, modify, and even delete the messages. For instance, an attacker pretends as a legal tag and intercepts the messages from a legal reader, and then it pretends as a legal reader to repeat the query. Thus, the communication stream in the air interface can be controlled by the attacker.

Under the replay attack, an attacker $\mathcal{A}$ performs the following actions:

- In one session:

$\mathcal{A}$ has learnt: $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$, $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_u}$, $\{r_{T_j}\}_{k_{m,n}}$.

- In the next session:

$R_i \rightarrow \mathcal{A}(T_a) \nrightarrow T_j$: $\{r'_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$;

$\mathcal{A}(T_a) \rightarrow R_i$: $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_u}$;

$R_i \rightarrow$ DB: PID_{T_j};

DB $\rightarrow R_i$: $k_{m,n}$;

R_i: $r_{R_j} \neq r'_{R_j}$;

$R_i \not\Rightarrow$: Authentication will fail.

- In the worse conditions:

DB $\rightarrow \mathcal{A}(R_a) \nrightarrow R_i$: $k_{m,n}$;

$\mathcal{A}(R_a) \rightarrow T_j$: $\{r_{T_j}\}_{k_{m,n}}$;

T_j: $r_{T_j} \neq r'_{T_j}$;

$T_j \not\Rightarrow$: Authentication will fail.

1. In one session, $\mathcal{A}$ has learnt all the messages exchanged between R_i and T_j.
2. In the next session, $\mathcal{A}$ disguises as a tag T_a to intercept the refreshed query $\{r'_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$. Then, T_a responds the former intercepted $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_u}$ to R_i. On receiving the message, R_i does decryption and transmits PID_{T_j} to DB. DB verifies that PID_{T_j} is valid and sends $k_{m,n}$ to R_i. R_i performs decryption on $\{r'_{R_i} \| \mathrm{PID}_{R_i}\}_{k_{m,n}}$ to gain r_{R_i}. Then R_i verifies that r_{R_i} differs from r'_{R_i}. Since r'_{R_i} and r_{R_i} are generated, respectively, in different sessions. The probability that r'_{R_i} equals r_{R_i} is negligible. The authentication will be end of failure with an error code.
3. In the worse conditions, $\mathcal{A}$ continues to disguise as a reader R_a to intercept $k_{m,n}$. R_a forwards the former intercepted $\{r_{T_j}\}_{k_{m,n}}$ to T_j, then T_j regains r_{T_j} for verification. The result is that r_{T_j} differs from the generated r'_{T_j} in the new session for r_{T_j} has been used in the former session. The authentication will be end of failure with an error code.

In KAAP, $\mathcal{A}$ may not pass the verifications with a dynamic update mechanism. Hence, the protocol can resist the replay attack even if the messages exposed during the resending process.

5.1.3 Tracking attack: Tracking attack is a passive privacy attack in which the attacker traces tags through malicious readers. Multiple malicious readers in fixed locations transmit the same query to a tag. If the tag's response remains invariant in all transmissions, the reader may track RFID-tagged items passing by, and may estimate a detail of correlating privacies such as the locations, interests, system capabilities etc.

Under the tracking attack, an attacker $\mathcal{A}$ performs the following actions

$\mathcal{A}(R_{ai}) \rightarrow T_j$:

$r_{R_{a1}} \| \mathrm{ID}_{R_{ai}}, r_{R_{a2}} \| \mathrm{ID}_{R_{ai}}, \ldots, r_{R_{aM}} \| \mathrm{ID}_{R_{ai}}$.

T_j: T_j searches $\mathrm{ID}_{R_{ai}}$ in the access list L_R, it turns out that there is no matching entry.

doi: 10.1049/iet-com.2010.0625

$T_j \nRightarrow$: Authentication will fail.

- In the worse conditions:

$T_j \rightarrow \mathcal{A}(R_{ai})$:

$\{\mathrm{PID}_{T_j} \| \{r_{R_{a1}} \| r_{T_{j1}}\}_{k_{*,n}}\}_{k_u}$;

$\{\mathrm{PID}_{T_j} \| \{r_{R_{a2}} \| r_{T_{j2}}\}_{k_{*,n}}\}_{k_u}$;

...

$\{\mathrm{PID}_{T_j} \| \{r_{R_{aM}} \| r_{T_{jM}}\}_{k_{*,n}}\}_{k_u}$;

$\mathcal{A}(R_{ai}) \nRightarrow$: $(r_{T_{j1}}, r_{T_{j2}}, \ldots, r_{T_{jM}})$ are random.

1. A disguises as different readers $R_{ai}(i = 1, 2, \ldots, M)$ to capture messages from T_j, then $\mathcal{A}$ continuously queries T_j with a fixed format value which yields a consistent response so as to monitor traffic flows.
2. A tries to analyse the consistent response of T_j to obtain its location information. While the authentication will fail for no suitable ID_{R^*} in the legal tag's storage.
3. In the worse conditions, T_j may respond R_{ai} by mistake, and the authentication will continue. In one site, T_j responds with $\{\mathrm{PID}_{T_j} \| \{r_{R_{a1}} \| r_{T_{j1}}\}_{k_{*,n}}\}_{k_u}$. In another site, T_j responds with $\{\mathrm{PID}_{T_j} \| \{r_{R_{a2}} \| r_{T_{j2}}\}_{k_{*,n}}\}_{k_u}$, and so forth. Any two responses are independent since $(r_{T_{j1}}, r_{T_{j2}}, \ldots, r_{T_{jM}})$ are randomly generated by the legal tag T_j. Thus, the attacker is incapable of tracking the special tag according to the random numbers.

In KAAP, $\mathcal{A}$ cannot confirm which tag the response belongs to since the tags' responses will be refreshed in each session. Therefore the attacker is impossible to analyse the traffic so that the location privacy is guaranteed. Hence, the protocol can counter the tracking attack with variable random numbers.

5.1.4 DOS attack: DoS attack may be caused by flooding an area with RF energy or jamming of channels, thus incapacitating the communication. The attacker disguises as a legal reader to send a huge number of queries with false addresses. Numerous tags respond simultaneously and wait for a long time until connections are disconnected. As resources allocated for the requests cannot be released, the system will be exhausted and the communication will not be established. The purpose of the DoS attack is not obtaining the sensitive TID, but rather trying to ensure that a legal reader cannot access the tag normally.

In KAAP, dual-intellect approaches are adopted to provide protection against the DoS attack.

1. Access lists (L_R, L_T) are used for preliminary check besides quick search. For instance, if an attacker $\mathcal{A}$ triggers a tag consecutively through imitating a legal reader, the tag will discern the illegal reader by no matching pseudorandom identifier in the access list L_R. Similarly, L_T arrests the malicious blocking on the database efficiently.
2. Random/pseudorandom numbers $(r_{R_i}, r_{T_j}, \mathrm{PID}_{R_i}, \mathrm{PID}_{T_j})$ are incorporated into the access control. The legal readers and tags can extract and store the last received random numbers and pseudorandom identifiers as temp lists. If a query arrives with the same query within a certain time, the entities will refuse to reply. The attacker $\mathcal{A}$ cannot involve into the sessions to disturb the normal communication by the intelligent recognition.

5.1.5 Internal forgery attack: Internal forgery attack means an internal legal reader (and/or tag) in one group forges another legal reader (and/or tag) in another group and oversteps its access authority to deceive others' private information.

Under the internal tag forgery attack

$R_i \rightarrow \hat{T}_j(T_j)$: $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$;

$\hat{T}_j(T_j) \rightarrow R_i$: $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{\hat{T}_j}\}_{k_{m,\hat{n}}}\}_{k_u}$;

$R_i \rightarrow \mathrm{DB}$: PID_{T_j};

$\mathrm{DB} \rightarrow R_i$: $k_{m,n}$;

$R_i \nRightarrow$: Authentication will fail.

1. The internal legal tag $\hat{T}_j$ in group $G_{T_{\hat{n}}}$ disguises as another legal tag T_j in group G_{T_n}, along with $\mathrm{PID}_{\hat{T}_j}$ disguises as PID_{T_j} to communicate with R_i. Owing to owning k_u and $k_{m,\hat{n}}$, but without $k_{m,n}$, $\hat{T}_j$ has to use its authentication key $k_{m,\hat{n}}$ for encryption and sends $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,\hat{n}}}\}_{k_u}$ to R_i.
2. When receiving the response, R_i decrypts the ciphertext with k_u to obtain PID_{T_j} and $\{r_{R_i} \| r_{T_j}\}_{k_{m,\hat{n}}}$, and then R_i forwards PID_{T_j} to DB.
3. DB retrieves PID_{T_j} to acknowledge that the tag is valid, and DB deliver $k_{m,n}$ to R_i. The result is that R_i cannot decrypt $\{r_{R_i} \| r_{T_j}\}_{k_{m,\hat{n}}}$ by $k_{m,n}$. Thus, R_i stops the authentication on $\hat{T}_j$ and responds with error code.

Under the internal reader forgery attack

$\hat{R}_i(R_i) \rightarrow T_j$: $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$;

$T_j \rightarrow \hat{R}_i(R_i)$: $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_u}$;

$\hat{R}_i(R_i) \rightarrow \mathrm{DB}$: PID_{T_j};

$\mathrm{DB} \rightarrow \hat{R}_i(R_i)$: $k_{\hat{m},n}$;

$\hat{R}_i(R_i) \nRightarrow$: Authentication will fail.

1. The internal reader $\hat{R}_i$ in group $G_{R_{\hat{m}}}$ disguises as another legal reader R_i in group G_{R_m}, along with $\mathrm{PID}_{\hat{R}_i}$ disguises as PID_{R_i} to send query $\{r_{R_i} \| \mathrm{PID}_{R_i}\}_{k_u}$ to T_j.
2. T_j gains PID_{R_i} by k_u, and sends $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_u}$ to $\hat{R}_i$. Then $\hat{R}_i$ executes decryption using the share key k_u to obtain PID_{T_j}, and it forwards PID_{T_j} to DB.
3. DB retrieves PID_{T_j} to acknowledge that the tag is valid, and DB delivers $k_{\hat{m},n}$ to $\hat{R}_i$. Then $\hat{R}_i$ cannot decrypt $\{r_{R_i} \| r_{T_j}\}_{k_{m,n}}$ with $k_{\hat{m},n}$. Thus, the authentication will be aborted. Unless the attackers broke the tags or readers via physical methods, they cannot pass the authentication to gain the tag information. Hence, the protocol is secure against the forgery attacks from internal legal entities.

In the protocol scheme, the forward security can be ensured by the distributed key array authentication, and security compromise will not reveal the data previously transmitted. The access list is an effective primary treatment for supplemental protection. In particular, the shared key k_u is

mainly used to resist the external attacks, along with the authentication key $k_{m,n}$ is specially used to defend the internal attacks. For instance, a 128-bit symmetric encryption algorithm such as AES is used. The probability that the attacker cracks a 128-bit symmetric key is $1/2^{128}$, which is negligible. Furthermore, even with a correct symmetric key, an attacker additionally needs to guess the random number r_{R_i} or r_{T_j} introduced into encryption and decryption, the probability is negligible. According to KAAP, an attacker cannot launch privacy attacks for the any sort of secret keys are not revealed. Even attacker decodes the ciphertext $\{\mathrm{PID}_{T_j} \| \{r_{R_i} \| r_{T_j}\}_{k_{m,n}}\}_{k_u}$, it cannot obtain tag's identifier replaced by the pseudorandom identifier PID_{T_j}, Furthermore, even if the attacker continues to decode the ciphertext $\{r_{R_i} \| r_{T_j}\}_{k_{m,n}}$, it cannot predict the random numbers generated dynamically and randomly in all phases. Table 3 shows the comparison of anti-attack abilities with related protocols. Hence, the protocol can resist both the external and internal attacks effectively.

5.2 Security requirement

The proposed KAAP ensures a secure mutual authentication with lightweight encryption, and can satisfy such security requirements, including confidentiality, integrity, authentication, anonymity and availability.

1. *Confidentiality:* Confidentiality requires that all of the messages are securely transmitted during all sessions. It is necessary to authenticate the participants when the TID is transmitted in the open channel, or to transmit the encrypted data so that only the authenticated reader can read the TID. In both forward and backward links, the real identifiers are substituted by the pseudorandom identifiers, besides the reader's identifier is hidden in the ciphertext form. The intruders cannot gather any information from the intercepted messages owing to the fulfilled encryption algorithm and the random control mechanism. Moreover, each legal reader with respective permission is authorised by tag groups, and overstepping authority is forbidden by the specific authentication key.
2. *Integrity:* A memory block of the tag is rewritable, and so forgery and data modifications may be possible. The access lists and secret keys could implement periodic updates. Thus, the exchanged data are protected by both the shared key and authentication keys to fight against arbitrary deletion, creation and replication by either illegal readers or unauthorised legal readers.
3. *Authentication:* The scheme provides mutual authentication between tags and readers by checking whether the computed value equals to the previously generated value. Four-round authentication is executed to block an unauthorised access, including the double searches in the access lists and the reader-tag mutual verifications.
4. *Anonymity:* The protocol offers anonymity using pseudorandom identifiers instead of exposing the real identifiers so that the attacker cannot identify the entire TID. Additionally, messages transmitted in the reader–tag link are random wraps because of random numbers employed to enforce dynamic update and variety. Even if the attacker intercepts and decodes the messages, it may only obtain the irregular pseudorandom identifiers but not the desired TID.
5. *Availability:* Denial of Access/Service that threatens availability is an inherent problem in RFID systems. There is less denial of authorised access to communication in which the access list and random control approaches are recommended for intelligent recognition. Even though some violent attacks (e.g. power interruption, message hijacking) occur, the authentication may provide recovery function based on ending authentication with an error code.

Table 3 Anti-attack abilities comparisons between related protocols

	LMAP [4]	RHLP [9]	HCP [10]	HIDVP [11]	KAAP
spoofing	Y	N	N	Y	Y
replay	Y	N	N	Y	Y
tracking	N	Y	Y	Y	Y
DoS	Y	N	N	N	Y
internal forgery	N	N	N	N	Y

Note: Y, the protocol has ability to resist the attacks; N, the protocol does not have ability to resist the attacks

6 Performance analysis

In RFID systems, the performance is another important metric besides the security issue, such that the optimisation and balance between security and performance are necessary for RFID systems [24]. In order to provide a comprehensive evaluation, KAAP is investigated from three aspects: storage requirement, communication overhead and computation load.

6.1 Storage requirement

In KAAP, T_j stores the TID ID_{T_j}, pseudorandom identifier PID_{T_j}, access list L_R, secret keys $(k_u, \{k_{1,n}, k_{2,n}, \ldots, k_{M,n}\})$, and revisable values are stored in rewritable memory. A 64-bit length is assumed for TID delivery according to ISO/IEC related standard. Access list stores all readers' pseudorandom identifiers. We are aware that the size of key array $K_{M\times N}$ increases in multiples by the numbers of reader groups and tag groups in the sensor system. For T_j, it only holds the authentication keys in the jth row of the key array, whose amount depends on the reader groups' number. Therein, the reader group number M is negligibly small as against the tag number J. Hence, such distributed key architecture can deal with the storage requirement for a signal tag and alleviate the maintenance effectively. Additionally, the memory consumption on cryptographic algorithm is another concern. AES as a cryptographic primitive for symmetric authentication may be recommended in KAAP scheme, and an efficient implementation of AES encryption functionality [16] needs about 3.4 K logic gates which is comparable with the simplest hash function (1.7 K), and even smaller than some common hash functions (e.g. MD5, SHA-1, SHA-256) which require between 16.0 and 23.0 K additional gates [25]. The required hardware complexity is acceptable [26, 27] and the implementation of KAAP will be suitable for medium-cost tags so as to apply lightweight encryption algorithms.

6.2 Communication overhead

Communication overhead is the average number of transmitted bit stream for each phase or for a full run of the protocol. In KAAP, the number of possible transmitting frames and expected receiving frames depends on message exchanges in authentication phases. The communication overhead refers to the sum of signalling loads and

doi: 10.1049/iet-com.2010.0625

Table 4 Performance comparison between related protocols

		LMAP [4]	RHLP [9]	HCP [10]	HIDVP [11]	KAAP
Stor.	T	$6L$	$1.5L$	$3L$	$3L$	$3L$
Comm.	$T \rightarrow$ R	$3L$	L	L	$3L$	L
	R $\rightarrow T$	$3L$	L	–	$2L$	L
	phases	4	5	4	5	5
Comp.	T	19B	R + H	2H	3H	R + 2E
	DB + R	21B	nH	nH $\times$ i	3H + R	R + 2E

Note: L, length of identifier/access list; i, length of hash-chain; m, number of readers; n, number of tags; B, bitwise operation; H, hash operation; R, RNG operation; E, encryption; –, no require
Ignoring the length of keys and random numbers

cryptographic processing loads during each authentication session. Suppose the identifiers of readers and tags have the same length L, the total length of message deliveries between a reader and a tag are $2L$ which is smaller than protocols [4, 11]. Moreover, only two messages are exchanged in the challenge-respond stage and another three in the mutual authentication stage. The total authentication progress completed via five phases is acceptable in real sensor system. It improves the system security and remains low in terms of complexity, and also the protocol can easily satisfy communication data rate restrictions.

6.3 Computation load

During the entire round, each reader and each tag performs two encryption–decryption operations and two random number generation (RNG) operations, respectively. The access lists L_R and L_T realise to retrieve the entities' identifiers without requiring exhaustive searches in the storage, which reduce the time complexity of search operation with the least complexity of $O(1)$ instead of $O(n)$ in batch mode. Meanwhile, L_T also makes the database not depend on the number of tags greatly. Unlike the protocols based on hash function or conventional encryption algorithm, KAAP has not given detailed encryption algorithms but recommend some lightweight algorithms such as [15, 16]. Meanwhile, the messages forwarded to the database have been decrypted by readers, which may further reduce the database's computation load and increase flexibility. Based on the existing technology, smart cards (e.g. MIFARE Plus, MIFARE DESFire) [28] comprise MPU, storage units, and COS. They can support the cryptographic algorithms such as AES, RSA and DES efficiently. There is no requirement of additional functional improvement except that a power-saving module should be considered to deal with multi-round encryption and decryption operations. However, we should recognise that the computation load of the protocol is higher than other protocols that only perform bitwise operations.

Above all, Table 4 shows the performance comparisons with the related protocols. KAAP has the similar storage requirement as protocols [10, 11], and it is much less than LMAP [4]. There are no exhaustive searches in KAAP, unlike protocols [9, 10] require at least J searches in the storage. The communication overhead of KAAP is lightweight comparing with schemes [4, 11]. Encryption operations are applied in KAAP which may increase the computation load theoretically, and so lightweight algorithms have been recommended. Note that all the components are assumed L bits sized, and the length of keys and random numbers are ignored for the sake of simplicity. KAAP has moderate complexity in storage requirement and communication overhead, and it owns acceptable computation load. Hence, KAAP can be used in the applications requiring high security yet compromise in tag cost.

7 Conclusions and future works

In the paper, a novel distributed KAAP is proposed for classified security protection in RFID-based sensor systems. The protocol adopts challenge-response mutual authentication mechanism and random access control mechanism to enhance security and privacy protection. Particularly, access lists are used as index-pseudonyms for retrieving identifiers in the storage, and pseudorandom identifiers are applied to guarantee tag anonymity. As a formal analysis, the protocol is verified by GNY Logic to provide that there is non-existence of obvious design flaws and security errors. Furthermore, the protocol satisfies more security requirements and is sufficiently robust to withstand several external and internal attacks. According to the performance analysis, the protocol turns out with acceptable complexity and moderate operations can satisfy high reliability. Therefore KAAP is suitable for the high-security and large-scale RFID applications such as finance and military fields.

In the future, several topics should be taken into consideration, such as adaptive lightweight cryptographic algorithms should be designed for the proposed authentication scheme. Moreover, the investigation of the RFID authentication protocols with anti-collision mechanism should also be paid more attention to.

8 Acknowledgments

This work was supported by the National High-Tech Research and Development Program of China (Grant No. 2008AA04A101).

9 References

1 Chen, Y., Rapajic, P.: 'Ultra-wideband cognitive interrogator network: adaptive illumination with active sensors for target localisation', *IET Commun.*, 2010, **4**, (5), pp. 573–584
2 Jo, M., Youn, H.Y.: 'Intelligent recognition of RFID tag position', *IET Electron. Lett.*, 2008, **44**, (4), pp. 308–309
3 Chen, M., Gonzalez, S., Zhang, Q., Leung, V.C.M.: 'Code-centric RFID system based on software agent intelligence', *IEEE Intell. Syst.*, 2010, **25**, (2), pp. 12–19
4 Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: 'LMAP: a real lightweight mutual authentication protocol low-cost RFID tags'. Proc. Second Workshop on RFID Security, July 2006
5 Hopper, N.J., Blum, M.: 'Secure human identification protocols'. Proc. Seventh Int. Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, Gold Coast, Australia, December 2001, pp. 52–66

6 Chien, H.Y.: 'SASI: a new ultralightweight RFID authentication protocol providing strong authentication and strong integrity', *IEEE Trans. Dependable Secur. Comput.*, 2007, **4**, (4), pp. 337–340
7 Sun, H.M., Ting, W.C.: 'A Gen2-based RFID authentication protocol for security and privacy', *IEEE Trans. Mob. Comput.*, 2009, **8**, (8), pp. 1052–1062
8 Ren, X., Xu, X., Tang, H.: 'A new mutual authentication scheme for low-cost RFID'. Proc. IET Conf. Wireless, Mobile and Sensor Networks (CCWMSN'07), Shanghai, China, December 2007, pp. 170–173
9 Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: 'Security and privacy aspects of low-cost radio frequency identification systems', *Secur. Pervasive Comput.*, 2004, **2802**, pp. 201–212
10 Henrici, D., Muller, P.: 'Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers'. Proc. Second IEEE Annual Conf. Pervasive Computing and Communications Workshops, Florida, USA, March 2004, pp. 149–153
11 Rhee, K., Kwak, J., Kim, S., Won, D.: 'Challenge-response based RFID authentication protocol for distributed database environment', *Secur. Pervasive Comput.*, 2005, **3450**, pp. 70–84
12 Tan, C.C., Sheng, B., Li, Q.: 'Secure and serverless RFID authentication and search protocols', *IEEE Trans. Wirel. Commun.*, 2008, **7**, (4), pp. 1400–1407
13 Ahamed, S.I., Hoque, M.D.E., Rahman, F., Kawsar, F., Nakajima, T.: 'YA-SRAP: yet another serverless RFID authentication protocol'. Proc. IET Fourth Int. Conf. Intelligent Environments, Seattle, USA, July 2008, pp. 1–8
14 Piramuthu, S.: 'Lightweight cryptographic authentication in passive RFID-tagged systems', *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, 2008, **38**, (3), pp. 360–376
15 Lee, Y.K., Sakiyama, K., Batina, L., Verbauwhede, I.: 'Elliptic-curve-based security processor for RFID', *IEEE Trans. Comput.*, 2008, **57**, (11), pp. 1514–1527
16 Feldhofer, M., Wolkerstorfer, J., Rijmen, V.: 'AES implementation on a grain of sand', *IEE Proc. Inf. Secur.*, 2005, **152**, pp. 13–20
17 Bringer, J., Chabanne, H.: 'Trusted-HB: a low-cost version of HB secure against man-in-the-middle attacks', *IEEE Trans. Inf. Theory*, 2008, **54**, (9), pp. 4339–4342
18 Gong, L., Needham, R., Yahalom, R.: 'Reasoning about belief in cryptographic protocols'. Proc. IEEE Computer Society Symp. Research in Security and Privacy, California, USA, May 1990, pp. 234–248
19 Burrows, M., Abadi, M., Needham, R.: 'A logic of authentication', *ACM Trans. Comput. Syst.*, 1990, **8**, pp. 18–36
20 Zuo, Y.: 'Secure and private search protocols for RFID systems', *Inf. Syst. Front.*, 2009, **12**, (5), pp. 507–519
21 Godor, G., Imre, S.: 'Security analysis of the simple lightweight authentication protocol'. Proc. 2010 Ninth Int. Conf. Networks (ICN), French Alps, France, April 2010, pp. 231–236
22 Mathuria, A., Safavi-Naini, R., Nickolas, P.: 'Some remarks on the logic of Gong, Needham and Yahalom'. Proc. Int. Computer Symp., Taiwan, China, December 1994, pp. 303–308
23 Ding, Z., Guo, L., Wang, Y.: 'An authentication protocol based on key array for RFID', *Electron. Inf. Technol.*, 2009, **31**, (3), pp. 722–726
24 Olteanu, A., Xiao, Y., Zhang, Y.: 'Optimization between AES security and performance for IEEE 802.15.3 WPAN', *IEEE Trans. Wirel. Commun.*, 2009, **8**, (12), pp. 6030–6037
25 http://www.heliontech.com/core.htm, accessed May 2010
26 Huang, Y., Yuan, C., Chen, M., Lin, W., Teng, H.: 'Hardware implementation of RFID mutual authentication protocol', *IEEE Trans. Ind. Electron.*, 2010, **57**, (5), pp. 1573–1582
27 Che, W., Chen, W., Meng, D., *et al.*: 'Power management unit for battery assisted passive RFID tag', *IET Electron. Lett.*, 2010, **46**, (8), pp. 589–590
28 http://www.nxp.com, accessed December 20101

doi: 10.1049/iet-com.2010.0625