Zero-Knowledge Authentication Protocol Based on Alternative Mode in RFID Systems

Hong Liu, *Student Member, IEEE*, and Huansheng Ning, *Member, IEEE*

***Abstract*—As radio frequency identification (RFID) applications become ubiquitous, security and privacy issues have been addressed with universal acceptances. This paper proposes a lightweight Zero-Knowledge Authentication Protocol (ZKAP) based on alternative mode to address such severe problems. In ZKAP, dual zero-knowledge proofs are randomly chosen to provide anonymity and mutual authentication without revealing any sensitive identifiers. Pseudo-random flags and access lists employed for quick search and check ensure high efficiency and scalability. Meanwhile, formal proof model based on reasonable mathematical assumptions is established to prove the adaptive completeness, soundness and zero-knowledgeness, and the attack models are adopted to analyze the resilience and resistance for malicious attacks. It indicates that ZKAP owns no obvious design defects theoretically and is robust enough to resist major attacks (e.g., forgery, replay, Man-in-the-Middle, and tracking). The protocol is attractive and appropriate for low-cost and resource-restricted RFID systems.**

***Index Terms*—Authentication, protocol, radio frequency identification (RFID), security, zero-knowledge proof.**

I. Introduction

RADIO FREQUENCY IDENTIFICATION (RFID) technology has been widely used in diverse applications. For certain applications, resource-restricted tags and readers are employed for the real-time identification and inventory management, which are vulnerable for various attacks due to inherent limitations. As a critical issue in ubiquitous wireless sensor networks, RFID has been triggered significant security/privacy concerns since potential threats are suffering from malicious attackers via an open air interface [1], [2]. In universal RFID applications, standard cryptography functions may not be available on low-cost tags. Hence, it is significant to investigate and design lightweight RFID authentication protocols.

Several researches have been presented to address the security/privacy problems, and unconventional cryptographic algorithms have been proposed for RFID systems [3], [4]. Thereinto, ultralightweight protocols base on bitwise logical operators to provide a minimum level of security [5], [6]. Lightweight protocols mainly execute Hash/Message Authentication Code (MAC) functions, Cyclic Redundancy Code (CRC), and Pseudo-Random Number Generator (PRNG) [7]–[9]. Middleweight protocols applying typical cryptographic primitives such as Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC) [10], [11], are considered to be highly secure and acceptable with ignoring tag cost and power consumption. Furthermore, existent schemes are mainly based on identity, in which readers and tags establish mutual trusts by recognizing each other's identifier. Such nonzero-knowledge proofs own innate tradeoff between trust and anonymity. The zero-knowledge proof is a subtle scheme that can realize efficient interaction to guarantee a prover convince validity of a verifier without releasing any sensitive information [12]–[16]. Goldreich *et al.* demonstrated the general notions in [17], and properties (i.e., completeness, soundness, and zero-knowledgeness) are introduced as significant evaluating indicators for cryptographic protocols. Thereinto: 1) completeness: if a prover reader is valid and properly follows the zero-knowledge proof protocol, the verifier tag will always accept the reader as a legal entity; 2) soundness: if a prover reader is invalid, the verifier tag will always reject the reader as an illegal entity; 3) zero-knowledgeness: any verifier does not learn anything except that a statement is true, and no sensitive information about the share commitments is revealed in the zero-knowledge proof. Meanwhile, most extant schemes apply single mode to offer homogenous approach with limitations against heterogeneous risks. Such single-mode schemes may be more easily cracked by persistent malicious monitoring, yet the dual-mode schemes in which the alternative mode is chosen randomly in each session are harder to be cracked due to the dynamic pattern. Hence, the alternative mode is necessary and critical to be introduced into the zero-knowledge authentication protocols. In the paper, the authors propose a lightweight Zero-Knowledge Authentication Protocol (ZKAP), which adopts alternative proof mode into the mutual authentication to guarantee security and reliability. In this paper, the main contributions are as follows.

- Devising the zero-knowledge proof algorithm with alternative mode to achieve anonymity. A major departure from the previous researches is that our scheme does not operate in a single mode, and alternative execution of dual zero-knowledge proofs is applied for RFID authentication. The zero-knowledge proofs based on hardness of mathematical problems enable a tag to verify a reader without exposing any secret identities during communications. Differing from pure zero-knowledge protocols, the scheme integrates with multiple-access control mechanisms, and executes one trial for a tag to judge a reader. Random partition and dynamic update are introduced into the proofs, which contribute to reduce the probability of misidentification.

Manuscript received March 09, 2011; revised April 28, 2011; accepted June 09, 2011. Date of publication June 20, 2011; date of current version November 02, 2011. This work is supported in part by the National Natural Science Foundation of China (NSFC) and the Civil Aviation Administration of China (CAAC) (61079019), and in part by the National High-Tech Research and Development Program of China (2008AA04A101). The associate editor coordinating the review of this paper and approving it for publication was Prof. Ralph Etienne-Cummings.

The authors are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: liuhongler@ee.buaa.edu.cn; ninghuansheng@buaa.edu.cn).

Digital Object Identifier 10.1109/JSEN.2011.2160052

- Realizing balance of security and performance. Except for necessary algebraic assumptions needed by the commitment algorithm, our scheme does not need any additional requirements to achieve high completeness and soundness. Meanwhile, our scheme is succinct and flexible without extra information exchanged. Moreover, the tag involves division, exaction and Boolean logic operations, instead of high-cost requirements (e.g., random number generator, hash function). Legal readers and tags realize distributed trusts, which reduces the database's computation load and increases flexibility.
- Achieving quick search and check. Pseudo-random flags and access lists are applied to search a specific entity for quick check. If a query arrives with the same flag within a certain time, the entities will refuse to reply. Meanwhile, pseudonyms are extracted as partial fields for verification. The dynamic partial pseudonyms are applied instead of exhaustive search in the storage, which reduces the time complexity of search operation effectively.

This paper is organized as follows. Related works on RFID security protocols are reviewed in Section II. Section III introduces the authentication phases of the proposed protocol. The detailed zero-knowledge proof model and protocol properties are presented in Section IV. In Section V, the attack model analysis is studied on the major attacks. Finally, Section VI draws the conclusion.

II. Related Works

In the section, the authors focus on the security-enhancing schemes based on the zero-knowledge proof.

Peng and Bao [12] proposed Publicly Verifiable Secret Sharing (PVSS), and the scheme supports immediate secret recovery without special requirements on the shared secrets. Discrete-logarithm-based function is employed to commit the secret, and Paillier encryption algorithm is used to encrypt the shares. Therefore, encrypted shares can be proved to recover an unique secret publicly. Specifically, order problem and logarithm-root assumption are defined for verification. Based on the assumption, the probability that a polynomial algorithm can calculate integers to satisfy $g^s x_1^N / x_2^{N_i} = 1 \pmod N$ without knowledge of factorization of N is negligible for $s \neq 0$, $x_1, x_2 \neq 1$. Public proofs are verified by comparing whether $g^w v_1^{N_i} (\prod_{j=0}^{t} C_j^{i^j})^c = a_1 \pmod N$, $g_i^w v_2^{N_i} c_i^c = a_2 \pmod{N_i^2}$, and $g^w v^N (\prod_{j=0}^{t} C_j^{i^j})^c = a \pmod N$ hold. Furthermore, the advanced scheme is improved by comparing whether $g^w v^N D_i^c = a \pmod N$. It is claimed that the protocol can achieve correctness, soundness, zero-knowledge, and efficiency simultaneously.

Lu *et al.* [13] proposed a pseudo trust based scheme, in which entity generates an unforgeable and verifiable pseudonym by hash function to conceal its real identity. The strengths of the design realize high security and scalability, low traffic and cryptography overheads. Specifically, the pseudo trust enables the trust management so that anonymity is protected during the authentication; the distributed trust is realized without a centralized trusted certificate authority; the Man-in-the-Middle attack is resisted by the assistant hash function so that pseudonyms and authentication paths are bound together. Diffie–Hellman key exchange is embedded into the authentication for generating a session key. Three-step verification is performed as follows. First, the verifier computes $u' = h_4(x, PI_R, T_R, g^a)$, and compares whether $u' = u$ holds. Second, the verifier checks whether PI_I equals $h_3(Seed_I, n_I)$. Finally, $y = c(\prod_{i=0}^{k} s_{j_i}^{e_{j_i}+u_i} \pmod{n_I})$ is challenged for zero-knowledge verification, then the verifier checks whether $y^2 = x(\prod_{i=0}^{k} v_{j_i}^{e_{j_i}+u_i} \pmod{n_I})$ holds. Hence, the interactive communication is regarded as anonymous and forward security.

Bhargav–Spantzel *et al.* [14] addressed the problem of identity misuse, and proposed multifactor identity verification based on aggregated zero-knowledge proofs (ZKPK). The scheme refers to verify the ownership of multiple strong identifiers without revealing them in public. A new cryptographic primitive is employed with aggregate signatures on commitments for the zero-knowledge proof. The co-gap Diffie–Hellman assumption is given for groups with bilinear maps, along with the bilinear aggregate signatures are aggregated into a whole signature $\sigma = \prod_{i=0}^{t} \sigma_i$. The verifier checks the validity of signatures by comparing whether $M = \prod_{i=0}^{t} M_i$ and $e(\sigma, g_2) = e(M, v)$ hold. Moreover, the commitments are semantically secure with refreshed secret and strong identifier enrolled, which leads that even if an attacker records all the values of the strong identifiers, it will not impersonate as a legal entity.

Malek and Miri [15] proposed a zero-knowledge protocol for RFID forward link authentication. The asymmetric scheme based on error-correcting codes in which matrix operations are removed without compromising security. A reader verifies a tag by checking whether $d = H(y_0 + y_1)^T$, $t - \mu \leq w(y_0) \leq t + \mu$, $t - w(c) \leq w(y_1) \leq t + w(c)$, and $w(c) - \mu \leq w(y_0 + y_1) \leq w(c) + \mu$ hold. Based on the error-correcting codes and code-based cryptography, the tag can be authenticated by the reader without publicly sharing the tag's secret identifier.

Kizza [16] revised a Fiege–Fiat–Shamir zero-knowledge protocol for the Airborne Networks (ANs). The protocol aims to reduce the effect of ping-pong exchanges and speed up building trust between a prover and a verifier. In the scheme, the prover chooses two large Blum prime integers (p, q), and two security integers (k, t). Then, it creates a secret vector $s = \{s_1, s_2, \ldots, s_k\}$ with $GCD(s_i, n) = 1$, and computes the vector $v = \{v_1, v_2, \ldots, v_k | v_i \equiv s_i^2 \pmod n\}$ for further verification. The prover extracts a submatrix $D = \{a_{ij} = (0, 1)\}$ as a challenge, then obtains a vector $y = \{y_1, y_2, \ldots, y_f | y_i \equiv r\pi s_i^{a_{ij}} \pmod n\}$. The verifier checks whether each elements of y satisfies that $y_j^2 \equiv \pm x\pi v_i^{a_{ij}}$. The protocol has to repeat with different r and submatrices D until the verifier confirms that the prover indeed owns s_i that is the modular square roots of v_i. Additionally, the verifier and prover can build trust and confidence at $\{1 - ((1/2)^f)^k\}$ for each parallel burst or $\{1 - (((1/2)^f)^k)^t\}$ for t bursts of parallelism.

To summarize previous researches, most zero-knowledge proof protocols apply homogeneous mode for authentication. Nevertheless, the proposed ZKAP differs from the conventional schemes, and it defines the authentication progress with alternative mode to enhance capability for persistent attacks. Combining with the practical RFID applications, additional lightweight access control mechanisms including random

partition and mutual authentication are integrated into the zero-knowledge proof, and pseudorandom flags and access lists are applied for quick search and check to achieve high execution efficiency. It is demonstrated that ZKAP implements comprehensive approaches to realize high security and efficiency.

III. Protocol Description

The RFID system consists of readers R, tags T, and the database DB. The channels between R and DB are considered as secure, while the air interfaces between R and T are suffering from malicious attacks. Suppose that row vector $\{s_i\}_j$ and s_0 are owned by R, and T derives the row vector $\{p_i\}_j$ that has specific mapping relation with $\{s_i\}_j$, in which $i = 1, 2, \ldots, j$. Meanwhile, a secret K is preshared by DB and T, and access lists (L_R, L_T) are used to store flags (F_R, F_T) as indexes to check certain reader and tag. DB stores all the identifier information, communication data and functionalities, and plays an essential role in verifying the validity of tags or readers. The arithmetic relational expressions are satisfied that, $\forall i = 1, 2, \ldots, d$; $d \in \mathbb{Z}^*$; $d \leq j$

$$\prod_{i=1}^{d}(s_i - s_0)^i \cdot \sum_{i=1}^{d} b_i \equiv 1 \pmod{n} \quad (1)$$

$$(s_i - s_0)^{2i} p_i \equiv 1 \pmod{n}. \quad (2)$$

Thereinto, (1) indicates that the product of $\prod_{i=1}^{d}(s_i - s_0)^i$ and $\sum_{i=1}^{d} b_i$ has the same remainder with 1 upon division by n, and (2) means that $(s_i - s_0)^{2i} p_i$ and 1 have the same remainder upon division by n.

A. System Parameters

Table I shows the parameters in the protocol.

TABLE I
Parameters

Parameter	Description
R	The reader in the RFID system.
T	The tag in the RFID system.
DB	The database in the RFID system.
$\mathcal{A}$	The illegal attacker in the RFID system.
R_a, T_a	The reader/tag imitated by $\mathcal{A}$.
SID_R, SID_T	The secret identifier of R, T.
PID_R, PID_T	The pseudonym of R, T.
ID_{R_a}, ID_{T_a}	The imitative identifier of $\mathcal{A}$ (R_a, T_a).
F_R, F_T	The pseudo-random flags of R, T.
L_R, L_T	The access list storing flags of legal readers and tags.
r_R, r_T	The random number generated by R, T.
r_{R_a}, r_{T_a}	The random number generated by R_a, T_a.
s_0, p_0	The random integer generated by R, T.
$\{s_i\}_j$, $\{p_i\}_j$	The secret j dimensional row vector.
$\{b_i\}_j$	The nonzero j dimensional column vector.
K	The secret value that is safely pre-shared between tags and database, and is a secure value without being revealed to the third entity.
K_l, K_r	The left/right partial fields of K.
d	The random integer is computed by rounding operation on r_R, $d = [r_R]$ is used for random partitions, including obtaining the high order d bits of PID_R, PID_T for quick search and check, extracting the first d fields of $\{s\}_j$, $\{p\}_j$ for zero-knowledge proof, and dividing the secret K into left and right parts for calculation.
n	The Blum integer $n = pq$, in which p, q are large primes.
$g(.)$	The pseudo random function.
$x\|_d$	The first d fields of x.
$[.]$	The rounding operation.
$\oplus$	XOR bitwise logic operator.
$<<, >>$	Left/Right shift bitwise logic operator.
$\|\|$	Concatenate operator.
$\rightarrow$	Transition operator.
$\stackrel{?}{=}$	Comparison operator.

B. Authentication Phase

Fig. 1 shows the proposed ZKAP based on alternative mode. The protocol detail is described in the following.

1) Phase 1. $R \rightarrow T \rightarrow R$: The reader R generates a random number r_R to initiate a new session, and it computes A_R with r_R and n as its elements. Then, R sends $A_R||F_R||r_R$ to the tag T as a query. Upon receiving the query, T first checks the validity of R by checking whether there is a corresponding F_R in the access list L_R, in which F_R is used to mark the reader with a specific flag, time stamp, etc. If it is not valid, the protocol will terminate with an error code. Otherwise, T will perform the rounding operation on r_R to gain a round-off integer $d = [r_R]$. Then, T extracts the first d fields of its pseudonym PID_T to obtain $PID_T|_d$, and continues to divide the preshared secret K into $K_l||K_r$. The partitioning method is as that, mark the higher d bits as K_l and the other lower bits as K_r. During the random partitions, underflow should be considered, and zero is padded to the dummy bits. T computes g_T, and performs XOR, left shift operations to obtain A_T. Thereafter, T responds $A_T||F_T||PID_T|_d$ to R.

2) Phase 2. $R \rightarrow DB \rightarrow R$: When R receives the response, it first checks the validity of T by checking F_T in L_T. If there is a match, R will re-extract the first d fields of PID_T to obtain $PID_T|'_d$, and continues to check whether the received $PID_T|_d$ equals the computed $PID_T|'_d$. If the two values are consistent, T will pass the primary authentication by R. Otherwise, R will regard T as illegal and the protocol will terminate with an error code. Then, R performs the same rounding operation on r_R to gain d, extracts the first d fields of PID_R to obtain $PID_R|_d$, and continues to forward $A_T||A_R||r_R||PID_R|_d$ to DB for the further authentication. Upon receiving the messages, DB performs the same extraction on PID_R to obtain $PID_R|'_d$, and checks whether the computed $PID_R|'_d$ has the same first d bits as the received $PID_R|_d$. If there is a match, R will pass the further authentication by DB. Afterwards, DB computes g_{DB} according to the same calculation equations as g_T and performs right shift, XOR operations to obtain A_{DB}. DB proceeds with the further authentication on T by comparing whether A_{DB} equals A_R. If the two values are equal, DB will consider that the tag T is legal and send "OK" to R; otherwise, it responds with "Error" to terminate the protocol.

3) Phase 3. $R \rightarrow T \rightarrow R \rightarrow DB \rightarrow R$: When R receives "OK," it extracts the first d fields of $\{s_i\}_j$ to obtain the subset $\{s_i\}_d = \{s_1, s_2, \ldots, s_d\}$, and computes the corresponding nonzero row vector $\{b_i\}_d$ by the appointed rule. Hereafter, the

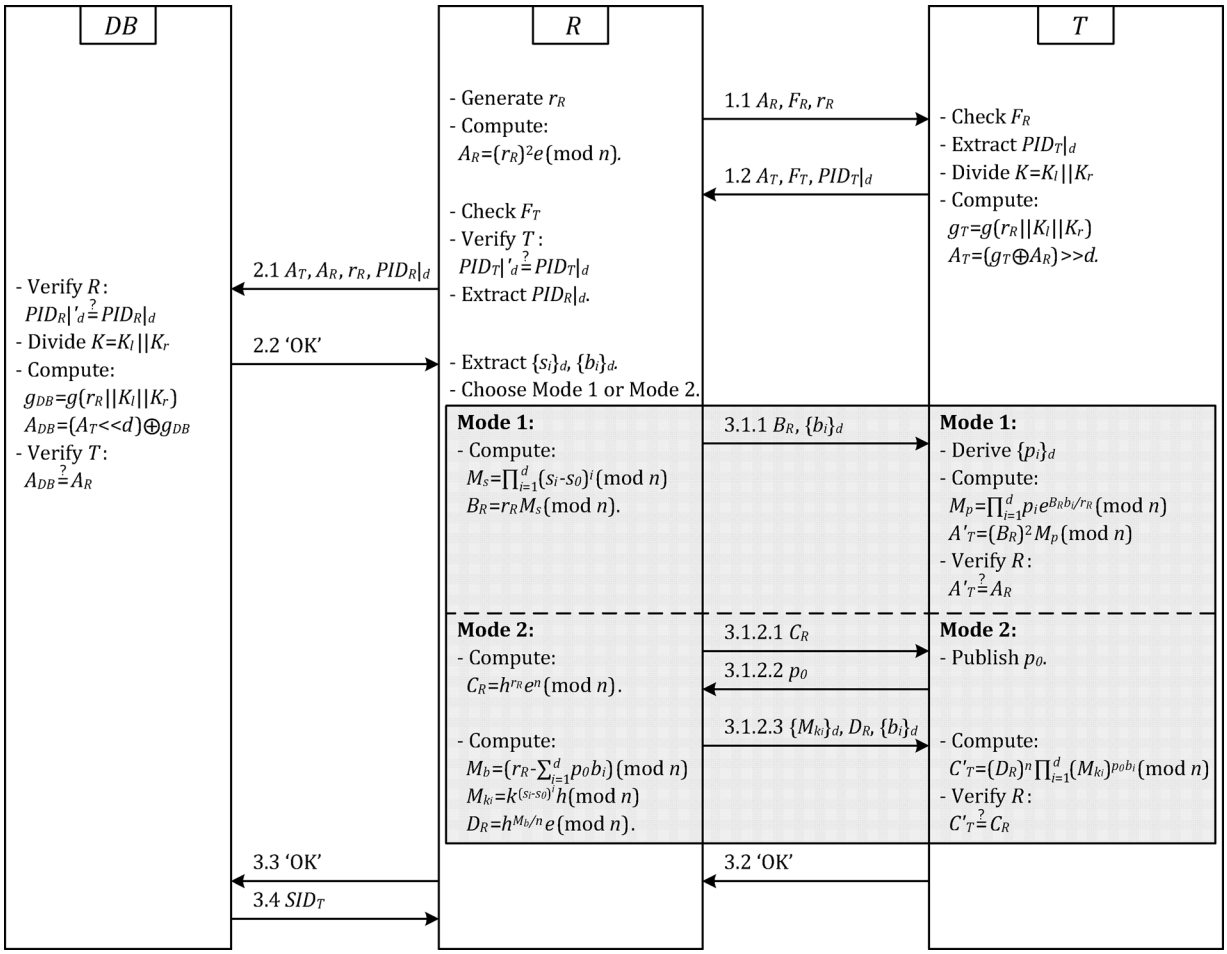

Fig. 1. The ZKAP based on alternative mode.

alternative zero-knowledge mode is randomly chosen by R, the detailed procedures of Modes 1 and 2 are as follows.

- Mode 1: R computes M_s and B_R, then transmits $B_R||\{b_i\}_d$ to T for the zero-knowledge proof. When T receives the messages, it first derives the corresponding subset $\{p_i\}_d = \{p_1, p_2, \ldots, p_d\}$ by the given mapping relation, and proceeds with the calculations to obtain M_p and A'_T. T checks whether the computed A'_T equals the former received A_R. If the two values are equal, T may believe that R is legal.
- Mode 2: R computes C_R, then transmits C_R to T. T publishes p_0 as the response to R. Upon receiving p_0, R continues to compute M_b, M_{k_i}, D_R sequentially, then transmits $\{M_{k_i}\}_d||D_R||\{b_i\}_d$ to T for zero-knowledge proof. When T receives the messages, it computes C'_T. T checks whether the computed C'_T equals the former received C_R. If the two values are equal, T may believe that R is legal.

Note that T judges the specific selected mode by the received data flows. If R passes the zero-knowledge authentication, T will send "OK" to R. Thereafter, R forwards "OK" to DB, and DB returns SID_T to R. Otherwise, it responds with "Error" to terminate the protocol.

In summary, ZKAP applies lightweight mechanisms to realize security, efficiency and reliability, including zero-knowledge proof, random partition, quick check, and mutual authentication. The main approaches are complementary as follows.

4) Zero-Knowledge Proof Mechanism: The improved alternative zero-knowledge proof scheme is adopted for safeguard, which is based on the intractability of Fiege–Fiat–Shamir algorithm [18], factoring large integers and discrete logarithm algorithm to realize nonreversibility without revealing any sensitive data. In the open air interface, $\{A_R, A_T, B_R, C_R, D_R, M_{k_i}\}$ can be safely published since an attacker may not resolve n from those messages if the logarithm is plenty robust. Differing from conventional zero-knowledge protocols which need several trials in one session, our scheme only executes one trial for a verifier to judge a prover. Random elements are introduced into the proof, which contribute to reduce the probability of misidentification.

5) Random Partition Mechanism: The random integer d is adopted for random partition. One aspect is dividing the preshared secret K into two partial fields $K_l||K_r$. The dynamic fields K_l and K_r are self-refreshed in each session, which can reduce additional update modules and workloads on the tag. The other aspect is extracting the subset of fields $\{s_i\}_d$ and $\{p_i\}_d$. The usage of random partitions is applied to zero-knowledge proof to enhance dynamic update. Note that high-cost random number generations are carried out by readers instead of tags, which effectively reduces the tag cost.

6) Quick Check Mechanism: Quick check mechanism includes two aspects. One refers to access lists (L_R, L_T) storing all the pseudorandom flags (F_R, F_T) which are used to mark certain reader and tag for quick search. R and T maintain their corresponding access list and associated rules, which enables the self-adapting addressing mode adopted. Meanwhile, flags may distinguish different sessions with time stamp. If a query arrives with unmatched flags, the entity will be rejected and DB will store the illegal flag in the blacklist. The other refers to pseudonyms (PID_T, PID_R) that are extracted as the first d fields ($PID_T|_d$, $PID_R|_d$) for verification. The dynamic partial pseudonyms are applied for quick search instead of exhaustive search 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.

7) Mutual Authentication Mechanism: The triple-step authentication procedure is performed to realize access control. R and T verify each other to ensure that the interactive entity is authorized, and DB verifies both R and T neutrally. If and only if both authentications succeed, DB will transmit SID_T to the reader via the secure channel.

Meanwhile, performance analysis is pivotal for protocol design, and several factors including tag storage, communication overhead and computation load should be considered. In ZKAP, the main tag storage includes the identifier SID_T, the pseudorandom identifier PID_T, the preshared secrets $\{F_T, L_R, K\}$, and other storage for Boolean logic operations. Lightweight mathematical calculation is employed to enhance high tag efficiency. Considering communication overhead, the total mutual authentication between reader and tag averagely completes in five rounds, which is a reasonable number of rounds. For the computational cost, the tag involves only simple bitwise operations (e.g., XOR, modulo, and bit rotation), which can be implemented with low cost and high efficiency in ubiquitous RFID systems. Furthermore, no additional hardware requirements are required, which may further reduce the computation load and increase flexibility.

IV. Protocol Model and Property Analysis

In this section, the authors focus on the zero-knowledge proofs of ZKAP, in which T as a prover attempts to pass the authentication by a column verifier DB, and also R as a prover attempt to pass the authentication by a verifier T. The authors also show how commitments A_R and C_R are used in the different modes to provide signature verification with hidden identifiers. Note that verifications according to F_R, F_T, $PID_R|_d$, $PID_T|_d$ are skipped in the zero-knowledge proof model, and the related data flows are also neglected. The details will be given in the attack models in Section V.

A. Zero-Knowledge Proof Model

In ZKAP, there is a set of linear combinations $(s_1 - s_0), (s_2 - s_0), \ldots, (s_j - s_0)$ such that elements of vector $\{b_i\}_j$ are calculated as $\prod_{i=1}^{d}(s_i - s_0)^i \cdot \sum_{i=1}^{d} b_i \equiv 1 \pmod n$. For $\{s_i\}_j$ and $\{p_i\}_j$, they satisfy the expression that $(s_i - s_0)^{2i} p_i \equiv 1 \pmod n$ for $i = 1, 2, \ldots, j$. Proof model is established based on above two mathematical assumptions, and parameters and symbols are defined as follows.

- Let $n = pq$ be the Blum integer in which p, q are large primes congruent to the type of $4a+3$ and $GCD(a, 3) = 1$. -1 is a quadratic nonresidue of n, and has a Jacobi symbol mod n of $+1$. Though n is published, its factorization is concealed. All the computations involving the integers are carried out modulo n.
- Let $L \subset \{0,1\}^*$ be a binary language, and G be a finite integral cyclic group of mod n in space $\mathbb{Z}_n^*$, in which the discrete logarithm assumption holds.
- Let k and h be generators of G, s_0 be a secret committed in M_s and M_{k_i}.
- $r_R \sim \mathrm{U}(\mathbb{Z}_n^*)$ refers to d that is uniformly distributed in $\mathbb{Z}_n^*$.
- L is a parameter satisfying that 2^{-L} is small enough and 2^L is smaller than p, q simultaneity.

If R and T are honest, the challenged messages can be shared and reconstructed. Otherwise, there is a dispute about validity of the challenged messages, and the protocol will terminate without revealing any sensitive identifiers SID_R and SID_T. The detailed procedure is as follows.

1) Preliminary Authentication Phase: R randomly generates an integer r_R from $\mathbb{Z}_n^*$, and computes $A_R = (r_R)^2 e \pmod n$. If R passes the quick flag check, T will compute g_T, A_T as that

$$g_T = g(r_R || K_l || K_r) \tag{3}$$

$$A_T = (g_T \oplus A_R) \gg d. \tag{4}$$

Then, T sends A_T to R. If T passes the quick flag and pseudonym checks, R will send $A_T || A_R || r_R || PID_R|_d$ to DB. Similarly, if R passes the quick pseudonym check, DB computes g_{DB} and A_{DB} as that

$$g_{DB} = g(r_R || K_l || K_r) \tag{5}$$

$$A_{DB} = (A_T \ll d) \oplus g_{DB}. \tag{6}$$

DB checks T by comparing whether A_{DB} equals A_R. If it holds, DB will send "OK" to R. If R and T are valid, g_{DB} should equal g_T, and DB will obtain A_{DB} as that

$$\begin{aligned} A_{DB} &= (A_T \ll d) \oplus g_{DB} \\ &= (((g_T \oplus A_R) \gg d) \ll d) \oplus g_T \\ &= A_R. \end{aligned} \tag{7}$$

2) Zero-Knowledge Proof Phase: R maps SID_{R_i} into $\mathbb{Z}_n^*$, and divides SID_R into fields $\{SID_{R_i}\}_j$. Its output set $\{s_i\}_j = f(\{SID_{R_i}\}_j) \rightarrow \mathbb{Z}_n^*$ is signatures for SID_{R_i}. Then, R extracts the subset $\{s_i\}_d$ of $\{s_i\}_j$. Suppose that $x_i = (s_i - s_0) \pmod n$ for $i = 1, 2, \ldots, d$, R solves the equation $\prod_{i=1}^{d}(x_i)^i \cdot \sum_{i=1}^{d} b_i = 1 \pmod n$ to obtain the corresponding nonzero row vector $\{b_i\}_d = (b_1, b_2, \ldots, b_d)^T$. Hereafter, dual zero-knowledge proof modes are randomly chosen.

In Mode 1, R aggregates the signatures $\{s_i\}_d$ into a polynomial M_s, then computes B_R as that

$$M_s = \prod_{i=1}^{d}(s_i - s_0)^i \pmod n \tag{8}$$

$$B_R = r_R M_s \pmod n. \tag{9}$$

Thereafter, R sends $B_R||\{b_i\}_d$ for the zero-knowledge proof, then T derives $\{p_i\}_d$, and computes M_p and A'_T. T verifies R by comparing whether A'_T equals A_R

$$M_p = \prod_{i=1}^{d} p_i e^{B_R b_i / r_R} \pmod{n} \tag{10}$$

$$A'_T = (B_R)^2 M_p \pmod{n}. \tag{11}$$

In Mode 2, R computes C_R as that

$$C_R = h^{r_R} e^n \pmod{n}. \tag{12}$$

R transmits C_R to T as a shared secret. Then, T publishes p_0 as a reply. Upon receiving p_0, R continues to compute M_b, M_{k_i} and D_R as that

$$M_b = (r_R - \sum_{i=1}^{d} p_0 b_i) \pmod{n} \tag{13}$$

$$M_{k_i} = k^{(s_i - s_0)^i} h \pmod{n} \tag{14}$$

$$D_R = h^{M_b/n} e \pmod{n}. \tag{15}$$

Thereafter, R sends the commitments $\{M_{k_i}\}_d||D_R||\{b_i\}_d$ to T for the zero-knowledge proof. When T receives the messages, it computes C'_T. T verifies R by comparing whether C'_T equals C_R. If the verification succeeds, T will accept R. Otherwise, it will reject R and regard C_R as an invalid share

$$C'_T = (D_R)^n \prod_{i=1}^{d} (M_{k_i})^{p_0 b_i} \pmod{n}. \tag{16}$$

T can publicly verify the knowledge of discrete logarithms, and it accepts R only if the transmitted proofs are successfully verified. The proofs in Modes 1 and 2 guarantee that the same shared values are committed, and T can verify the validity of R without revealing any sensitive identifier to solve the dispute. Until now, message "OK" will be forwarded to DB from T, then DB will transmit SID_T to R via secure channel.

B. Protocol Property Analysis

The authors illustrate the protocol based on above zero-knowledge proof model, and follows the method in references [12], [13] to analyze ZKAP's protocol properties.

Theorem 1. (Completeness): If R is valid, the probability that it passes the authentication is almost 1. Precisely, if a legal reader R properly follows the authentication phases, T will always accept R as valid.

Proof: Suppose that R is valid, and properly follows the protocol.

In Mode 1, T operates public verification according to (1) and (2)

$$\begin{aligned} A'_T &= (B_R)^2 M_p \pmod{n} \\ &= (r_R M_s)^2 \prod_{i=1}^{d} p_i e^{B_R b_i / r_R} \\ &= (r_R)^2 e^{\prod_{i=1}^{d} (s_i - s_0)^i \cdot \sum_{i=1}^{d} b_i} \prod_{i=1}^{d} (s_i - s_0)^{2i} p_i \\ &= (r_R)^2 e \pmod{n} \\ &= A_R. \end{aligned} \tag{17}$$

In Mode 2, T operates public verification according to (1)

$$\begin{aligned} C'_T &= (D_R)^n \prod_{i=1}^{d} (M_{k_i})^{p_0 b_i} \pmod{n} \\ &= (h^{M_b/n} e)^n \prod_{i=1}^{d} (k^{(s_i - s_0)^i} h)^{p_0 b_i} \\ &= h^{r_R} e^n h^{-\sum_{i=1}^{d} p_0 b_i} k^{\sum_{i=1}^{d} (s_i - s_0)^i p_0 b_i} h^{\sum_{i=1}^{d} p_0 b_i} \\ &= h^{r_R} e^n \pmod{n} \\ &= C_R. \end{aligned} \tag{18}$$

Thus, T always accepts R since the value A'_T and C'_T are still consistent with A_R and C_R, only if R has the proper shared secrets.

Theorem 2. (Soundness): If $\hat{R}$ is invalid, the probability that it passes the authentication will be negligible. Precisely, if an illegal reader $\hat{R}$ interferes with the communication by impersonating as R and convincing T that it is R, T will always reject $\hat{R}$ as invalid entity.

Proof: Suppose that $\hat{R}$ is invalid, and ZKAP bases on the hardness of finding a set that $(s_1 - s_0), (s_2 - s_0), \ldots, (s_d - s_0)$ such that elements of nonzero vector $\{b_i\}_d$ and $\{p_i\}_d$ are calculated as $\prod_{i=1}^{d} (s_i - s_0)^i \cdot \sum_{i=1}^{d} b_i \equiv 1 \pmod{n}$ and $(s_i - s_0)^{2i} p_i \equiv 1 \pmod{n}$. $\hat{R}$ sends $\hat{B}_R = \hat{r}_R \prod_{i=1}^{\hat{d}} (\hat{s}_i - \hat{s}_0)^i \pmod{n}$ (or $\hat{M}_{k_i} = k^{(\hat{s}_i - \hat{s}_0)^i} h \pmod{n}$, $\hat{D}_R = h^{(\hat{r}_R - \sum_{i=1}^{\hat{d}} p_0 \hat{b}_i)/n} e \pmod{n}$) and $\{\hat{b}_i\}_{\hat{d}}$ to T.

1) The success probability $P(m)$ of correctly guessing the chosen authentication mode is that $P(m) = 1/2$.
2) The success probability $P(d)$ of correctly guessing the random integer $\hat{d}$ is decided by all possible values of d which vary from 1 to j, it turns out that $P(d) = 2^{-j}$.
3) The success probability $P(s_0)$ of correctly guessing the constant $\hat{s}_0$ is decided by its bit length l, it turns out that $P(s_0) = 2^{-l}$.
4) The success probabilities $P(s_i)$ and $P(b_i)$ of correctly guessing the vectors $\{\hat{s}_i\}_{\hat{d}}$, $\{\hat{b}_i\}_{\hat{d}}$ are decided by their dimension size $\hat{d}$, it turns out that $P(s_i) = P(b_i) = 2^{-\hat{d}}$.

Hence, R has negligible probability to publish accredited $\{\hat{s}_i\}_{\hat{d}}$ and $\{\hat{b}_i\}_{\hat{d}}$ to meet $\prod_{i=1}^{\hat{d}} (\hat{s}_i - \hat{s}_0)^i \cdot \sum_{i=1}^{\hat{d}} \hat{b}_i = 1 \pmod{n}$, and T has negligible probability to extract proper $\{p_i\}_d$ such that $(\hat{s}_i - \hat{s}_0)^{2i} p_i = 1 \pmod{n}$ holds. T will reject $\hat{R}$ as a large probability event. It indicates that the scheme satisfies soundness and owns negligible soundness error even for an all-powerful personator.

Lemma 1: If $\hat{R}$ extracts B_R (i.e., n, $\{s_i\}_d$, $\{b_i\}_d$), D_R and M_{k_i} such that $\prod_{i=1}^{d} (s_i - s_0)^i \cdot \sum_{i=1}^{d} b_i \neq 1 \pmod{n}$ or $(s_i - s_0)^{2i} p_i \neq 1 \pmod{n}$ for $i = 1, 2, \ldots, d$. The probability that $\hat{R}$ can pass the verification is negligible.

Proof: If $\hat{R}$ extracts n, $\{s_i\}_d$ and $\{b_i\}_d$, and passes the verification by T with a non-negligible probability based on equation $\prod_{i=1}^{d} (x_i)^i \cdot \sum_{i=1}^{d} b_i = 1 \pmod{n}$ and $(x_i)^{2i} p_i = 1 \pmod{n}$ having no solution for $x_i = s_i - s_0, i = 1, 2, \ldots, d$, there will be a contradiction. Since $\hat{R}$ passes the verification, it has to successfully prove the validity of $\{s_1, s_2, \ldots, s_d\}$ and $\{d_1, d_2, \ldots, d_d\}$ with a non-negligible probability. Thereinto,

$\{s_i\}_d$ and $\{b_i\}_d$ are based on reasonable mathematical assumptions and discrete logarithm algorithms. Therefore, the committed integer n and subsets ($\{s_i\}_d$, $\{b_i\}_d$) do not satisfy the given assumptions (1) and (2), and the probability that $\hat{R}$ passes the authentication is negligible.

Lemma 2.1: In Mode 1, if R generates two different integers r_R and $\hat{r}_R$, then transmits two different commitments $B_R||\{b_i\}_d$ and $\hat{B}_R||\{b_i\}_d$ to T for verification respectively. T will not obtain the same A'_T and $\hat{A}'_T$. Specifically, $A'_T = (B_R)^2 M_p \pmod n$, $\hat{A}'_T = (\hat{B}_R)^2 \hat{M}_p \pmod n$. Note that r_R and $\hat{r}_R$ are not a pair of opposite numbers.

Proof: Suppose that two different integers r_R and $\hat{r}_R$ are generated to obtain different $A_R = (r_R)^2 \pmod n$ and $\hat{A}_R = (\hat{r}_R)^2 \pmod n$. Passing the preliminary authentication phase, R continues to compute and transmit different commitments $B_R||\{b_i\}_d$ and $\hat{B}_R||\{b_i\}_d$ to T. According to the computation, the result will be that $A'_T = \hat{A}'_T$, and we have that

$$\begin{aligned}(B_R)^2 M_p &= (\hat{B}_R)^2 \hat{M}_p \\ \Rightarrow \left(\frac{B_R}{\hat{B}_R}\right)^2 &= \frac{\hat{M}_p}{M_p}. \end{aligned} \tag{19}$$

The left side of (19) yields that

$$\text{Left} = \left(\frac{B_R}{\hat{B}_R}\right)^2 = \left(\frac{r_R}{\hat{r}_R}\right)^2 \pmod n. \tag{20}$$

The right side of (19) yields that

$$\begin{aligned}\text{Right} = \frac{\hat{M}_p}{M_p} &= \prod_{i=1}^{d} e^{(\hat{B}_R/\hat{r}_R - B_R/r_R)b_i} \pmod n \\ &= e^{0\cdot\sum_{i=1}^{d} b_i} \pmod n \\ &= 1 \pmod n. \end{aligned} \tag{21}$$

Due to r_R and $\hat{r}_R$ are not a pair of opposite numbers, the authors have that $(r_R/\hat{r}_R)^2 = 1$, i.e., $r_R = \hat{r}_R$. The result will lead a contradiction with the initial assumption that $r_R \neq \hat{r}_R$. Hence, T will not obtain the same A'_T and $\hat{A}'_T$ when R challenges different r_R and $\hat{r}_R$.

Lemma 2.2: In Mode 2, if T publishes two different challenges p_0 and $\hat{p}_0$ to reply the same commitment C_R, R will respond two different $\{M_{k_i}\}_d||D_k||\{b_i\}_d$ and $\{\hat{M}_{k_i}\}_d||\hat{D}_k||\{\hat{b}_i\}_d$ for verification, respectively, and R can compute an integer $\hat{s}_i$ such that $\{\hat{M}_{k_i}\}_d = k^{(\hat{s}_i - s_0)^i} h$.

Proof: Suppose that the two different challenges p_0 and $\hat{p}_0$ are replied for the same commitment C_R, and the corresponding responses are $\{M_{k_i}\}_d||D_k||\{b_i\}_d$, $\{\hat{M}_{k_i}\}_d||\hat{D}_k||\{\hat{b}_i\}_d$. T obtains that

$$\begin{aligned}(D_R)^n \prod_{i=1}^{d} (M_{k_i})^{p_0 b_i} &= (\hat{D}_R)^n \prod_{i=1}^{d} (\hat{M}_{k_i})^{\hat{p}_0 b_i} \\ \Rightarrow \left(\frac{D_R}{\hat{D}_R}\right)^n &= \prod_{i=1}^{d} \frac{(\hat{M}_{k_i})^{\hat{p}_0 b_i}}{(M_{k_i})^{p_0 b_i}}. \end{aligned} \tag{22}$$

The left side of (22) yields that

$$\begin{aligned}\text{Left} &= h^{M_b - \hat{M}_b} \pmod n \\ &= h^{\sum_{i=1}^{d} (\hat{p}_0 - p_0) b_i} \pmod n. \end{aligned} \tag{23}$$

The right side of (22) yields that

$$\text{Right} = \prod_{i=1}^{d} (k^{(s_i - s_0)^i} h)^{(\hat{p}_0 - p_0) b_i} \pmod n. \tag{24}$$

According to the Euclidean algorithm, R computes α and β satisfying that $\beta(\hat{p}_0 - p_0) = \alpha n + GCD(n, \hat{p}_0 - p_0) = \alpha n + 1$. We have that

$$\begin{aligned} h^{\sum_{i=1}^{d} (\hat{p}_0 - p_0) b_i} &= \prod_{i=1}^{d} (k^{(s_i - s_0)^i} h)^{(\hat{p}_0 - p_0) b_i} \pmod n \\ \Rightarrow \prod_{i=1}^{d} h^{\beta(\hat{p}_0 - p_0) b_i} &= \prod_{i=1}^{d} (k^{(s_i - s_0)^i} h)^{(\alpha n + 1) b_i} \pmod n \\ \Rightarrow \prod_{i=1}^{d} (k^{-\alpha n (s_i - s_0)^i} h^{\beta(\hat{p}_0 - p_0) - \alpha n})^{b_i} & \\ = \prod_{i=1}^{d} (M_{k_i})^{b_i} \pmod n. \end{aligned} \tag{25}$$

Thus, R computes the integer $\hat{s}_i = s_0 - (\alpha n)^{1/i}(s_i - s_0)$ in polynomial time such that $\{\hat{M}_{k_i}\}_d = k^{(\hat{s}_i - s_0)^i} h$ for $i = 1, 2, \ldots, d$.

Theorem 3. (Zero-Knowledgeness): ZKAP is strict honest-verifier zero knowledge.

Proof: A zero-knowledge proof protocol has zero-knowledge property, meaning that there is a simulator for the proof. Simulator is a procedure in which pseudo elements are generated instead of genuine elements indistinguishably. For perfect zero-knowledge proof, the distributions published by the simulator and the protocol are distributed exactly the same. In ZKAP, R and T do not reveal any knowledge of the secret identifiers (SID_R, SID_T) to achieve zero-knowledge verification. Any reader without revealing any knowledge of s_i, s_0 and r_R can simulate the proof transcript $\{A_R, A_T, B_R, C_R, D_R, M_{k_i}\}$ to satisfy the verification.

In Mode 1, ZKAP is based on the Feige–Fiat–Shamir scheme in which R does not reveal any sensitive information to T during the whole communication even if an attacker tries to operate interception. In Mode 2, T randomly chooses p_0 from $\{0, 1, \ldots, 2^L\}$, and computes $C'_T = (D_R)^n \prod_{i=1}^{d} (M_{k_i})^{p_0 b_i} \pmod n$. In the proof transcript, we have the following:

1) r_R, M_b, M_{k_i}, D_R are uniformly distributed in $\mathbb{Z}_n^*$;
2) p_0 is uniformly distributed in $\{0, 1, \ldots, 2^L\}$;
3) $C_R = h^{r_R} e^n \pmod n$ holds;
4) $C'_T = (D_R)^n \prod_{i=1}^{d} (M_{k_i})^{p_0 b_i} \pmod n = C_R$ holds.

Thus, the simulated transcript and the proof transcript own the same distribution. All messages transmitted among R, T and DB are indistinguishable with the uniform distributions with random numbers. ZKAP allows any entity to verify the signature of a commitment without knowing details. Moreover, the commitments are semantically secure embedded random numbers, which ensures that even if an attacker learns the exchanged values, it will also not forge itself as a legal entity.

In ZKAP, a certain modulus n is secret such that the responses do not reveal any information about the shared commitments. The preshared mathematical relationships are confidential, and

the parameters $(\{s_i\}_d, \{b_i\}_d, \{p_i\}_d, s_0)$ are unpredictable since d are randomly generated for division and extraction. d subtly leads that dimension size of vectors is dynamic in each session so as to ensure untraceability and forward security which means that even if an attacker obtains secret values in the current session, it still cannot compromise the values in past sessions. Besides, the mapping relation between $\{SID_{R_i}\}_j$ and $\{s_i\}_d$ are not publicly disclosed. Thus, ZKAP can employ such zero-knowledge proofs to guarantee completeness, soundness, and zero-knowledgeness.

V. Attack Model Analysis

In Section V, the attack model is established to analyze the security and privacy. Suppose that the communication between R and DB is secure, while the open air interface between R and T is assumed to be insecure. Major attacks including forgery, replay, Man-in-the-Middle, and tracking are analyzed.

A. Forgery Attack

Forgery attack refers that an attacker can masquerade as a legal reader or tag in order to access the system resources, which consists of two scenarios: reader forging attack and tag forging attack.

1) Reader Forging Attack: During the reader forging attack, an attacker $\mathcal{A}$ imitates as a reader R_a in order to access T, and the exchanged messages are as follows:

- In one session:
  $\mathcal{A}(R_a) \rightarrow T$: $A_{R_a}||F_{R_a}||r_{R_a}$.
  T: T cannot recognize $\mathcal{A}(R_a)$ for no matching F_{R_*} in L_R.
  $T \nRightarrow \mathcal{A}(R_a)$: The protocol fails.
- In bad conditions:
  T: $d_a = [r_{R_a}]$, $g_T = g(r_{R_a}||K_l||K_r)$, $A_T = (g_T \oplus A_{R_a}) \gg d_a$.
  $T \rightarrow \mathcal{A}(R_a)$: $A_T||F_T||PID_T|_{d_a}$.
  $\mathcal{A}(R_a) \rightarrow DB$: $A_T||A_{R_a}||r_{R_a}||ID_{R_a}|_{d_a}$.
  DB: $PID_{R_*}|'_{d_a} \neq ID_{R_a}|_{d_a}$.
  $DB \nRightarrow \mathcal{A}(R_a)$: The protocol fails.
- In worse conditions:
  $DB \rightarrow \mathcal{A}(R_a)$: "OK."
  — Mode 1: $\mathcal{A}(R_a)$: $B_{R_a} = f(r_{R_a}, s_{i_a}, s_0, d_a)$.
  $\mathcal{A}(R_a) \rightarrow T$: $B_{R_a}||\{b_{i_a}\}_{d_a}$.
  T: $A'_T = f(B_{R_a}, p_i, b_{i_a}, d_a) \neq A_{R_a}$.
  $T \nRightarrow \mathcal{A}(R_a)$: The protocol fails.
  — Mode 2: $\mathcal{A}(R_a) \rightarrow T$: $C_{R_a} = f(r_{R_a}, h, d_a)$.
  $T \rightarrow \mathcal{A}(R_a)$: p_0.
  $\mathcal{A}(R_a)$: $D_{R_a} = f(r_{R_a}, h, p_0, b_{i_a}, d_a)$.
  $\mathcal{A}(R_a) \rightarrow T$: $\{M_{k_{i_a}}\}_{d_a}||D_{R_a}||\{b_{i_a}\}_{d_a}$.
  T: $C'_T = f(D_{R_a}, k, s_{i_a}, s_0) \neq C_{R_a}$.
  $T \nRightarrow \mathcal{A}(R_a)$: The protocol fails.

In one session, $\mathcal{A}$ disguises as a reader R_a, and sends a query $A_{R_a}||F_{R_a}||r_{R_a}$ to T. Upon receiving the query, T first verifies the validity of R_a by checking whether there is a corresponding F_{R_*} in L_R, and it will find that there is no matching flag, then the protocol fails.

In bad conditions, T ignores the mistake and performs extraction, division and calculation to obtain $PID_T|_{d_a}$, g_T and A_T. Then, T responds $A_T||F_T||PID_T|_{d_a}$ to R_a. R_a continues to transmit $A_T||A_{R_a}||r_{R_a}||ID_{R_a}|_{d_a}$ to DB ignoring the primary verification on T. Thereafter, DB verifies whether a corresponding $PID_{R_*}|'_{d_a}$ equals $ID_{R_a}|_{d_a}$. The result will be that $PID_{R_*}|'_{d_a} \neq ID_{R_a}|_{d_a}$. DB recognizes that R_a is illegal and the protocol fails.

In worse conditions, $\mathcal{A}$ passes the authentication by DB, and DB replies "OK" to continue the protocol.

— Mode 1: R_a computes B_{R_a} that consists of $\{r_{R_a}, s_{i_a}, s_0, d_a\}$ as variable parameters, then sends $B_{R_a}||\{b_{i_a}\}_{d_a}$ to T for authentication. T computes A'_T related with $\{B_{R_a}, p_i, b_{i_a}, d_a\}$, and compares whether the computed A'_T equals the former received A_{R_a}. The result will be that $A'_T \neq A_{R_a}$ since vector $\{s_{i_a}\}_j$ is forged by R_a. Even though $\prod_{i=1}^{d_a}(s_{i_a} - s_0)^i \cdot \sum_{i=1}^{d_a} b_i = 1 \pmod n$ holds, the probability that $(s_{i_a} - s_0)^{2i} p_i = 1 \pmod n$ is also negligible.

— Mode 2: R_a computes C_R, and sends C_R to T. T replies p_0 to R. Upon receiving p_0, R continues to compute D_R that consists of $\{r_{R_a}, h, p_0, b_{i_a}, d_a\}$ as variable parameters, then transmits $\{M_{k_{i_a}}\}_{d_a}||D_R||\{b_{i_a}\}_{d_a}$ to T for authentication. T computes C'_T related with $\{D_{R_a}, k, s_{i_a}, s_0\}$, T checks whether the computed C'_T equals the former received C_R. The result will be that $C'_T \neq C_{R_a}$ since the probability that $\prod_{i=1}^{d_a}(s_{i_a} - s_0)^i \cdot \sum_{i=1}^{d_a} b_{i_a} = 1 \pmod n$ is negligible.

Hence, ZKAP can recognize the imitated reader and realize self-protection against the reader forging attack.

2) Tag Forging Attack: During the tag forging attack, an intruder $\mathcal{A}$ imitates as a tag T_a in order to cheat R with false data, and the exchanged messages are as follows.

- In one session:
  $R \rightarrow \mathcal{A}(T_a)$: $A_R||F_R||r_R$.
  $\mathcal{A}(T_a)$: $d = [r_R]$, $g_{T_a} = g(r_R)$, $A_{T_a} = (g_{T_a} \oplus A_R) \gg d$.
  $\mathcal{A}(T_a) \rightarrow R$: $A_{T_a}||F_{T_a}||ID_{T_a}|_d$.
  R: R cannot recognize $\mathcal{A}(T_a)$ for no matching flag F_{T_*} in L_T.
  $R \nRightarrow DB$: The protocol fails.
- In bad conditions:
  R: Ignore the unmatched F_{T_a}, $PID_{T_*}|'_d \neq ID_{T_a}|_d$.
  $R \nRightarrow DB$: The protocol fails.
- In worse conditions:
  $R \rightarrow DB$: $A_{T_a}||A_R||r_R||PID_R|_d$.
  DB: $PID_{R_*}|'_d = PID_R|_d$, $g_{DB} = g(r_R||K_l||K_r) \neq g_{T_a}$, $A_{DB} = (A_T \ll d) \oplus g_{DB} \neq A_R$.
  $DB \nRightarrow R$: The protocol fails.

In one session, $\mathcal{A}$ disguises as a tag T_a, and receives a query $A_R||F_R||r_R$ from R. Upon receiving the query, T_a performs extraction, division, and calculation to obtain $ID_{T_a}|_d$, g_{T_a} and A_{T_a}. Here, $g_{T_a} = g(r_R)$ without $K_l||K_r$ as its element. T_a sends $A_{T_a}||F_{T_a}||ID_{T_a}|_d$ to R. Then, R verifies the validity of T_a by checking F_{T_a} in L_T, and it will find that there is no matching flag F_{T_*}, then the protocol fails.

In bad conditions, R ignores the mistake and verifies T_a by checking whether a corresponding $PID_{T_*}|'_d$ equals $ID_{T_a}|_d$. The result will be that $PID_{T_*}|'_d \neq ID_{T_a}|_d$ since the imitated ID_{T_a} is invalid. R recognizes that T_a is illegal and the protocol fails.

In worse conditions, $\mathcal{A}$ passes the primary authentication by R, and R transmits $A_{T_a}||A_R||r_R||PID_R|_d$ to DB. After DB verifies R, it calculates $g_{DB} = g(r_R||K_l||K_r)$ and $A_{DB} = (A_T \ll d) \oplus g_{DB}$, and compares whether A_{DB} equals A_R. The result will be that $g_{DB} \neq g_{T_a}$ and $A_{DB} \neq A_R$ since the secret K has never exposed before. Hence, ZKAP can recognize the imitated tag and realize self-protection against tag forging attack.

B. Replay Attack

Replay attack refers to that an attacker impersonates a legal entity to involve into the communications so as to access, modify, and even delete the messages. The protocol employs dynamic values embedded with a random integer d to resolve the replay attack.

In multiple sessions, an attacker $\mathcal{A}$ can impersonate a tag T_a to cheat the reader R. Suppose that $\mathcal{A}$ has recorded all the exchanged messages in one session. In another session, the attacker $\mathcal{A}$ pretends to be a tag T_a in order to intercept and obtain reader's query. Besides, $\mathcal{A}$ disguises as a reader R_a, and forwards the query to the legal tag T. Thus, the attacker may cheat both sides as if he were a legal entity, and the exchanged messages are as follows.

- In a former session:
  $\mathcal{A}$ has intercepted all the previous messages.
- In another session:
  $R \rightarrow \mathcal{A}(T_a) \nrightarrow T$: $A'_R||F'_R||r'_R$.
  $\mathcal{A}(T_a) \rightarrow R$: $A_T||F_T||PID_T|_d$,
  R: R cannot recognize $\mathcal{A}(T_a)$ for incorrect F_T.
  $R \nRightarrow DB$: The protocol fails.
- In bad conditions:
  R: Ignore the incorrect F_T, $PID_{T_*}|'_{d'} \neq PID_T|_d$.
  $R \nRightarrow DB$: The protocol fails.
- In worse conditions:
  $R \rightarrow DB$: $A_T||A'_R||r'_R||PID_R|_{d'}$.
  DB: $PID_{R_*}|'_{d'} \neq PID_R|d$, $g_{DB} = g(r'_R||K'_l||K'_r) \neq g_T$, $A_{DB} = (A_T \ll d') \oplus g_{DB} \neq A'_R$.
  $DB \nRightarrow R$: The protocol fails.

In a former session, $\mathcal{A}$ has intercepted all previous messages including A_R, F_R, r_R, A_T, F_T, $PID_T|_d$, B_R (or C_R, p_0, $\{M_{k_i}\}_d$, D_R), and $\{b_i\}_d$. In another session, $\mathcal{A}$ disguises as T_a to intercept the refreshed query $A'_R||F'_R||r'_R$ from R to T. When T_a receives the query, it will responds the former learnt messages $A_T||F_T||PID_T|_d$ to R, where $A_T = (g_T \oplus A_R) \gg d$. Upon receiving the response, R checks flag F_{T_*}, and it will find that the received flag F_T with a false time stamp is incorrect, then the protocol fails.

In bad conditions, R ignores the mistake and verifies T_a by checking whether the received $PID_T|_d$ equals the computed $PID_{T_*}|'_{d'}$. The result will be that $PID_{T_*}|'_{d'} \neq PID_T|_d$ since the refreshed numbers. R recognizes that T_a is illegal and the protocol fails.

In worse conditions, $\mathcal{A}$ passes the primary authentication by R, and R transmits $A_T||A'_R||r'_R||PID_R|_{d'}$ to DB. DB continues to check whether the received $PID_R|_d$ equals the computed $PID_{R_*}|'_{d'}$. The result will be that $PID_{R_*}|'_{d'} \neq PID_R|_d$. Furthermore, g_{DB} does not correspond with the former generated $g_T = g(r_R||K_l||K_r)$. Therefore, $A_{DB} = (A_T \ll d') \oplus g_{DB}$ does not equal the refreshed A'_R. Thus, the protocol will fail with an error code. Without regard to the correctness or incorrectness of the selected proof mode, the protocol still can resist the replay attack by the dynamic update mechanism.

C. Man-in-the-Middle Attack

Man-in-the-Middle (MITM) attack refers to an active eavesdropping in which an attacker $\mathcal{A}$ can make independent connections with a pair of legal reader and tag, can modify and relay messages between the reader R and the tag T without either entity recognizing that the communication channel has been compromised.

During the MITM attack, an attacker $\mathcal{A}$ imitates as a reader R_a and a tag T_a simultaneously. In order to cheat T, $\mathcal{A}$ has to directly forward the intercepted query $A_R||F_R||r_R$ to T, or forges a modified query $A_{R_a}||F_{R_a}||r_{R_a}$ to send it to T. In one case, $\mathcal{A}$'s behaviors are similar to the relay attack, in which R directly connects with $\mathcal{A}(T_a)$. In the other case, the query is modified by $\mathcal{A}(R_a)$, and the exchanged messages are as follows:

- In one session:
  $R \rightarrow \mathcal{A}(T_a) \nrightarrow T$: $A_R||F_R||r_R$.
  $\mathcal{A}(R_a) \rightarrow T$: $A_{R_a}||F_R||r_{R_a}$.
  $T \rightarrow \mathcal{A}(R_a) \nrightarrow R$: $A_T||F_T||PID_T|_d$.
  $\mathcal{A}(T_a) \rightarrow R$: $A_{T_a}||F_T||PID_T|_{d_a}$.
  R: $PID_{T_*}|'_d \neq PID_T|_{d_a}$.
  $R \nRightarrow DB$: The protocol fails.
- In bad conditions:
  $R \rightarrow DB$: $A_{T_a}||A_R||r_R||PID_R|_d$.
  DB: $PID_{R_*}|'_d = PID_R|_d$. $g_{DB} = g(r_R||K_l||K_r) \neq g_{T_a} A_{DB} = (A_{T_a} \ll d) \oplus g_{DB} \neq A_R$.
  $DB \nRightarrow R$: The protocol fails.
- In worse conditions:
  $DB \rightarrow R$: "OK."
  — Mode 1: R: $B_R = f(r_R, s_i, s_0, d)$.
  $R \rightarrow \mathcal{A}(T_a) \nrightarrow T$: $B_R||\{b_i\}_d$.
  $\mathcal{A}(R_a)$: $B_{R_a} = f(r_{R_a}, s_{i_a}, s_0, d)$.
  $\mathcal{A}(R_a) \rightarrow T$: $B_{R_a}||\{b_{i_a}\}_d$.
  T: $A'_T = f(B_{R_a}, p_i, b_{i_a}, d) \neq A_{R_a}$.
  $T \nRightarrow \mathcal{A}(R_a)$: The protocol fails.
  — Mode 2: $R \rightarrow \mathcal{A}(T_a) \nrightarrow T$: $C_R = f(h, d)$.
  $\mathcal{A}(R_a) \rightarrow T$: $C_{R_a} = f(h_a, d)$.
  $T \rightarrow \mathcal{A}(R_a) \nrightarrow R$: p_0.
  $\mathcal{A}(T_a) \rightarrow R$: p_{0_a}.
  R: $D_R = f(r_R, h, p_{0_a}, b_i, d)$.
  $R \rightarrow \mathcal{A}(T_a) \nrightarrow T$: $\{M_{k_i}\}_d||D_R||\{b_i\}_d$.
  $\mathcal{A}(R_a)$: $D_{R_a} = f(r_{R_a}, h_a, p_{0_a}, b_{i_a}, d)$.
  $\mathcal{A}(R_a) \rightarrow T$: $\{M_{k_{i_a}}\}_d||D_{R_a}||\{b_{i_a}\}_d$.
  T: $C'_T = f(D_{R_a}, k, s_{i_a}, s_0) \neq C_{R_a}$.
  $T \nRightarrow \mathcal{A}(R_a)$: The protocol fails.

In one session, $\mathcal{A}$ intercepts $A_R||F_R||r_R$, and sends $A_{R_a}||F_R||r_{R_a}$ to T as an initiator. After receiving the modified query, T responds $A_T||F_T||PID_T|_d$ to $\mathcal{A}$. Then, $\mathcal{A}$ continues to deliver the modified $A_{T_a}||F_T||PID_T|_{d_a}$ to R as an responder. R checks whether the received $PID_T|_{d_a}$ equals the computed $PID_{T_*}|'_d$. The result will be that $PID_{T_*}|'_d \neq PID_T|_{d_a}$ since the modified numbers (d, d_a). R recognizes that T_a is illegal and the protocol fails.

In bad conditions, $\mathcal{A}$ may relay $PID_T|_d$ to R without modifying into $PID_T|_{d_a}$, and R transmits $A_{T_a}||A_R||r_R||PID_R|_d$ to DB to continue the protocol. DB computes g_{DB} and A_{DB} for

tag authentication. The result will be that $g_{DB} \neq g_{T_a}$ since the modified numbers (r_R, r_{R_a}). Subsequently, $A_{DB} = (A_{T_a} \ll d) \oplus g_{DB}$ does not equal the received A_R. DB recognizes that T_a is illegal and the protocol fails.

In worse conditions, $\mathcal{A}$ may relay r_R to R without modifying into r_{R_a}, and DB replies "OK" to continue the protocol.

— Mode 1: R computes B_R, and sends $B_R||\{b_i\}_d$ to T. $\mathcal{A}$ intercepts the message and delivers the modified $B_{R_a}||\{b_{i_a}\}_d$ to T. Then, T computes A'_T, and compares whether the computed A'_T equals the former received A_{R_a}. The result will be that $A'_T \neq A_{R_a}$ since the probability of $(s_{i_a} - s_0)^{2i} p_i = 1 \pmod n$ is negligible.

— Mode 2: R computes and sends C_R for authentication. $\mathcal{A}$ intercepts the message and delivers the modified C_{R_a} to T. Thereafter, T replies p_0 to $\mathcal{A}$, then p_{0_a} is transmitted to R . Upon receiving p_{0_a}, R computes D_R and replies $\{M_{k_i}\}_d||D_R||\{b_i\}_d$ as response. $\mathcal{A}$ continues to transmit $\{M_{k_{i_a}}\}_d||D_{R_a}||\{b_{i_a}\}_d$ to T. T computes C'_T, and checks whether the computed C'_T equals the former received C_{R_a}. The result will be that $C'_T \neq C_{R_a}$ since the probability of $\prod_{i=1}^{d}(s_{i_a} - s_0)^i \cdot \sum_{i=1}^{d} b_{i_a} = 1 \pmod n$ is negligible.

Hence, ZKAP can resist the MITM attack, and the modified exchange messages between R and T via $\mathcal{A}$ cannot pass the multistep verifications.

D. Tracking Attack

Tracking attack is a typical passive attack where the attacker traces tags by malicious readers. Multiple malicious readers in fixed locations transmit the same query to challenge a certain tag. If the tag's response remains invariant in all transmissions, the reader may track tagged items passing by, and may estimate a detail of correlating privacy.

Under the tracking attack, the exchanged messages are as follows.

- In a series of sessions:
  $\mathcal{A}(R_{aj}) \rightarrow T$: $A_{R_{am}}||F_{R_{am}}||r_{R_{am}}$.
  T: T cannot recognize $\mathcal{A}(R_{am})$ for no matching flag F_{R_*} in L_R.
  $T \nRightarrow \mathcal{A}(R_{am})$: The protocol fails.
- In bad conditions:
  $T \rightarrow \mathcal{A}(R_{am})$: $A_{T_m}||F_{T_m}||PID_T|_{d_m}$.
  $\mathcal{A}(R_{aj})$: $\mathcal{A}$ cannot recognize T for $\{r_{R_{am}}\}$ and $\{d_m\}$ are random.
  $\mathcal{A}(R_{am}) \nRightarrow DB$: The protocol fails.
- In worse conditions:
  $\mathcal{A}(R_{am}) \rightarrow DB$: $A_{T_m}||A_{R_{am}}||r_{R_{am}}||ID_{R_a}|_{d_m}$.
  DB: $PID_{R_*}|'_{d_m} \neq ID_{R_a}|_{d_m}$.
  $DB \nRightarrow \mathcal{A}(R_{am})$: The protocol fails.

In a series of sessions, $\mathcal{A}$ disguises as several malicious readers $R_{am}(m = 1, 2, \ldots, M)$ to capture messages from T, then $\mathcal{A}$ continuously challenges T with a serious of queries to monitor traffic flows of T. $\mathcal{A}$ tries to analyze the consistent response of T to obtain its location information. The protocol will terminate since T cannot recognize $\mathcal{A}$ for no matching flag F_{R_*} in L_R.

In bad conditions, T may respond R_{am} by mistake, and the authentication will continue. In one site, T responds with $A_{T_1}||F_{T_1}||PID_T|_{d_1}$. In another site, T responds with $A_{T_2}||F_{T_2}||PID_T|_{d_2}$, and so forth. Any two responses are independent since $d_1, d_2, \ldots, d_M$ are calculated by the randomly generated numbers $r_{R_{a1}}, r_{R_{a2}}, \ldots, r_{T_{aM}}$. $\mathcal{A}$ cannot confirm which tag the response belongs to since the tags' responses will be refreshed in each session.

In worse conditions, $A_{T_m}||A_{R_{am}}||r_{R_{am}}||ID_{R_a}|_{d_m}$ is delivered from $\mathcal{A}$ to DB. DB verifies whether a corresponding $PID_{R_*}|'_{d_m}$ equals $ID_{R_a}|_{d_m}$. The result will be that $PID_{R_*}|'_{d_m} \neq ID_{R_a}|_{d_a}$. DB recognizes that R_{am} is illegal and the protocol fails. Thus, ZKAP ensures tag privacy protection against the tracking attack.

VI. Conclusion

In this paper, the authors present an authentication protocol based on alternative zero-knowledge mode. The alternative proof scheme integrates with multiple access control mechanisms (i.e., random partition, quick check, and mutual authentication), which is an advantage over general schemes. Reliable trust is built in anonymous environment without revealing any sensitive identifier. Completeness, soundness and zero-knowledgeness are achieved with reasonable algebraic assumptions. Diverse attacks can be resisted to enhance security/privacy protections. Moreover, the lightweight protocol achieves the security properties based on simple algebraic and logical operations without requiring expensive cryptographic algorithms, and it can be efficiently implemented in low-cost and resource-restricted RFID systems.

References

[1] Y. Zuo, "Survivable RFID systems: Issues, challenges, and techniques," *IEEE Trans. Syst. Man Cybern. C Appl. Rev.*, vol. 40, pp. 406–418, 2010.

[2] S. Cheng, K. Tom, L. Thomas, and M. Pecht, "A wireless sensor system for prognostics and health management," *IEEE Sensors J.*, vol. 10, pp. 856–862, 2010.

[3] M. Chen, S. Gonzalez, Q. Zhang, and V. C. M. Leung, "Code-centric RFID system based on software agent intelligence," *IEEE Intell. Syst.*, vol. 25, pp. 12–19, 2010.

[4] J. Chen, Q. Yu, Y. Zhang, H. Chen, and Y. Sun, "Feedback based clock synchronization in wireless sensor networks: A control theoretic approach," *IEEE Trans. Veh. Technol.*, vol. 59, pp. 2963–2973, 2010.

[5] H. Y. Chien and C. W. Huang, "A lightweight authentication protocol for low-cost RFID," *J. Sign. Process. Syst.*, vol. 59, pp. 95–102, 2010.

[6] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and A. Ribagorda, "LMAP: A real lightweight mutual authentication protocol low-cost RFID tags," in *Proc. 2nd Workshop on RFID Security*, 2006, pp. 137–148.

[7] H. M. Sun and W. C. Ting, "A Gen2-based RFID authentication protocol for security and privacy," *IEEE Trans. Mob. Comput.*, vol. 8, pp. 1052–1062, 2009.

[8] L. Kulseng, Z. Yu, Y. Wei, and Y. Guan, "Lightweight mutual authentication and ownership transfer for RFID systems," in *Proc. IEEE INFOCOM*, 2010, pp. 1–5.

[9] D. R. Lin, C. I. Wang, and D. J. Guan, "Efficient vehicle ownership identification scheme based on triple-trapdoor Chameleon Hash function," *J. Network Comput. Appl.*, vol. 34, pp. 12–19, 2011.

[10] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID aystems using the AES algorithm," in *Proc. Cryptogr. Hardw. Embed. Syst., CHES'04*, 2004, vol. 3156, LNCS, pp. 357–370.

[11] S. I. Ahamed, F. Rahman, and E. Hoque, "ERAP: ECC based RFID authentication protocol," in *Proc. 12th IEEE Int. Workshop on Future Trends of Distrib. Comput. Syst. (FTDCS'08)*, 2008, pp. 219–225.

[12] K. Peng and F. Bao, "Efficient publicly verifiable secret sharing with correctness, soundness and ZK privacy," *Inf. Secur. Appl.*, vol. 5932, pp. 118–132, 2009.

[13] L. Lu, J. Han, Y. Liu, L. Hu, J. Huai, L. M. Ni, and J. Ma, "Pseudo trust: Zero-knowledge authentication in anonymous P2Ps," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, pp. 1325–1337, 2008.

[14] A. Bhargav-Spantzel, A. C. Squicciarini, R. Xue, and E. Bertino, "Multifactor identity verification using aggregated proof of knowledge," *IEEE Trans. Syst. Man Cybern. C Appl. Rev.*, vol. 40, pp. 372–383, 2010.
[15] B. Malek and A. Miri, "Forward-link authentication for RFIDs," in *Proc. 2010 IEEE 21st Int. Symp. Personal Indoor and Mobile Radio Commun. (PIMRC)*, 2010, pp. 2644–2649.
[16] J. M. Kizza, "Fiege-Fiat-Shamir ZKP scheme revisited," *Int. J. Comput. ICT Res.*, vol. 4, pp. 9–19, 2010.
[17] O. Goldreich, S. Micali, and A. Wigderson, "Proofs that yield nothing but their validity or all language in np have zero-knowledge proof systems," *J. Assoc. Comput. Mach.*, vol. 38, pp. 691–729, 1991.
[18] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances Cryptology: Crypto 86*. : Springer, 1987, vol. 263, LNCS, pp. 186–194.

Hong Liu (S'10) received the B.E. degree from the Department of Electrical and Electronics Engineering, Nanyang Institute of Technology, Nanyang, China. Currently, she is working towards the Ph.D. degree at the School of Electronic and Information Engineering, Beihang University, Beijing, China.

Her research interests are security authentication protocol in RFID air interface and wireless sensor networks, and security architecture for the Internet of Things.

Huansheng Ning (M'10) was born in 1975 in Anhui Province, China. He received the B.S. degree from Anhui University, Hefei, in 1996 and the Ph.D. degree from Beihang University, Beijing, China, in 2001.

Currently, he is an Associate Professor at the School of Electronic and Information Engineering, Beihang University. He has presided over several research projects at the Natural Science Foundation of China (NSFC), National High Technology Research, and the Development Program of China (863 Project), etc. He has published more than 30 papers in journals, international conferences and workshops, and four books on RFID and Internet of Things. His current research focuses on RFID, Internet of Things and its application in aviation security.