KOCAELI UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DOCTORAL THESIS

SECURITY AND PRIVACY OF RFID SYSTEMS

ATAKAN ARSLAN

KOCAELI 2019

SECURITY AND PRIVACY OF RFID SYSTEMS

A THESIS SUBMITTED TO
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
KOCAELI UNIVERSITY

BY

ATAKAN ARSLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Prof. Dr. Sarp ERTÜRK
Supervisor, Kocaeli University .....

Prof. Dr. Hasan OCAK
Jury member, Kocaeli University .....

Assist. Prof. Dr. Sultan ALDIRMAZ ÇOLAK
Jury member, Kocaeli University .....

Prof. Dr. Oğuzhan KÜLEKCİ
Jury member, İstanbul Technical University .....

Assist. Prof. Dr. Tolga ÖNEL
Jury member, National Defense University .....

Thesis Defense Date: 01.11.2019

ACKNOWLEDGMENTS

I would like to thank all the people who have helped and inspired me during my Ph.D. study. I especially want to express my sincere gratitude to my dissertation advisor Prof. Dr. Sarp ERTÜRK for his guidance, worthwhile endless support, and invaluable patience throughout my Ph.D. studies. Studying under his guidance is a privilege for me.

I would like really appreciate my thesis jury members for examining this thesis and letting my dissertation defense be a memorable moment. I am also grateful to Dr. Sultan Aldırmaz ÇOLAK for her valuable suggestions, contributions and helpful hints throughout the study of my thesis.

Additionally, I would like to thank all my colleagues, especially Dr. Ahmet Yasin Çitkaya, Soner Ay, Dr. Şenol İşci and Mehmet Emin Gönen for their support and strong friendship. I also want to thank Dr. Süleyman Kardaş for his helpful comments and good friendship. I express my heartfelt gratitude to my dear friend Dr. Muhammed Ali Bingöl for his endless support and great friendship over the years. I owe him a great deal of gratitude for always being there.

Last but not least my deepest gratitude goes to my beloved family for their unflagging love, patience, encouragement, and support throughout my life. Most of all, my wife deserves special acknowledgment for her great helpings, devotions, and sacrifices. She is always an intelligent and quite understanding woman to encourage and motivate me in my studies. I regard that I am a great fortunate man to have her as my wife.

I would like to dedicate this dissertation to my loving wife, my little curios son and my beautiful daughter.

November−2019 Atakan ARSLAN

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

Adv : Adversary
Adv^B : Blinded adversary
b_i : Legitimacy of the i^{th} tag (e.g. The i^{th} tag is legitimate when $b_i = 1$)
BS_j : Back-end Server
B : Blinder
DB : Database/back-end system
$distr$: Probability distribution
$H(.)$: Hash Function
ID_{T_n} : Unique identifier of n^{th} tag
K : Key
(K_P, K_S) : Public-private key pair for a reader R
$\mathcal{O}^A$: The oracle for the functionality of A
m : Message
PE : Polynomial evaluations
RNG_i : The output of RNG for i^{th} protocol instance
r_i : i^{th} random bit string
R : Reader
S : The whole memory of ψ_T
s_i : The corruption state of the RNG of a tag T for the i^{th} protocol instance
T : Tag
$tbl(.)$: Table function $tbl(\psi_{Ti}) = ID_i$
$\Phi_{t_0}^T$: The internal knowledge of the tag
θ_π : The pairs of protocol instances and RNG states (π_i, s_i)
α : Security parameter
ψ_{T_n} : Pseudonym of the n^{th} tag
π_i : i^{th} protocol transcript
$\in_R$: The random element choice operator
$\ll$: Left circular shift operator
$\gg$: Right circular shift operator
$\oplus$: XOR operator
$\|$: Concatenation operator

Abbreviations

AES : Advanced Encryption Standard
ANSI : The American National Standards Institute
C1G2 : Class-1 Generation-2
CBC : Cipher Block Chaining
CDH : Computational Diffie-Hellman
CPU : Central Processing Unit
CRC : Cyclic Redundancy Check
DDH : Decisional Diffie-Hellman

DES : Data Encryption Standard
DH : Diffie-Hellman
DoS : Denial of Service
DRNG : Deterministic Random Number Generator
EC : Elliptic Curve
ECC : Elliptic Curve Cryptography
ECDH : Elliptic Curve Diffie-Hellman
ECDLP : Elliptic Curve Discrete Logarithm Problem
ECDSA : Elliptic Curve Digital Signature Algorithm
ECHD : Elliptic Curve Diffie-Hellman
EPC : Electronic Product Code
FIBS : Federal Information
FPGA : Field Programmable Gate Array
GPU : Graphics Processing Unit
H : Hash
HF : High Frequency
HMAC : Hash Message Authentication Code
IEEE : The Institute of Electrical and Electronics Engineers
IoT : Internet of Things
ISM : Industrial, Scientific and Medical
ISO : The International Organization for Standardization
LFSR : Linear Feedback Shift Register
MAC : Message Authentication Code
MD5 : Message-Digest Algorithm 5
MiTM : Man-in-The-Middle
NFC : Near-Field Communication
NIST : The U.S. National Institute of Standards and Technology
PA : Point Addition
PC : Personal Computer
PIN : Personal Identification Number
PKC : Public-Key Cryptography
PM : Point Multiplication
PPT : Probabilistic Polynomial Time
PRNG : Pseudo-Random Number Generator
PUF : Physically Unclonable Functions
RAM : Random Access Memory
RF : Radio Frequency
RFID : Radio Frequency Identification
RNG : Random Number Generator
RSA : RivestShamirAdleman
RSÜ : Rastgele Sayi Üreteçlerinin
SHA : Secure Hash Algorithm
SSL : Secure Sockets Layer
TLS : Transport Layer Security
TRNG : Truly Random Number Generator
UHF : Ultra High Frequency
WISB : Wireless Identification and Sensing Platform
ZC : ZeitControl

SECURITY AND PRIVACY OF RFID SYSTEMS

ABSTRACT

Radio frequency identification (RFID) is a promising and widespread wireless communication technology for entity identification or authentication. However, RFID systems suffer from security and privacy issues. Recently, numerous privacy-friendly RFID authentication protocols have been proposed to mitigate these concerns. In this dissertation, we concentrated on the security and privacy of RFID authentication protocols.

We primarily extend Vaudenay's privacy model which combines the early models and presents a new mature model for formal security analysis. We define a novel adversary class called RANDOMEYE, which allows analyzing the security of random number generators (RNGs) in RFID protocols to enhance the model. We further successfully apply our extended model to existing RFID schemes in the literature.

Secondly, we extensively examine the state-of-the-art RFID authentication protocols based on elliptic curve cryptography (ECC) in terms of security and performance. Some of these works claim that their protocols provide all general security and privacy properties. We prove that they do not provide forward and/or backward privacy contrary to their claim by providing formal security analysis. Then, we propose a secure, privacy-preserving and efficient protocol. We also present a comprehensive security and performance analysis of our proposed protocol and compare it to the existing relevant schemes in detail. Furthermore, we implement our proposal in a real RFID system to demonstrate its practicability. To the best of our knowledge, our proposed scheme is the most efficient ECC based RFID authentication protocol realized in a real-world environment that satisfies all common security and privacy features including backward and forward privacy.

Keywords: Implementation, Privacy, Public-Key Cryptography, RFID Protocols, Security.

RFID SİSTEMLERİNİN GÜVENLİK VE MAHREMİYETİ

ÖZET

Radyo frekansı tanımlama (RFID), varlık tanımlaması veya doğrulaması için umut verici ve yaygın bir kablosuz iletişim teknolojisidir. Ancak, RFID sistemleri güvenlik ve gizlilik sorunlarından muzdariptir. Son zamanlarda, bu endişeleri azaltmak için giz-lilik dostu çok sayıda RFID kimlik doğrulama protokolü önerilmiştir. Bu tezde, temel olarak RFID kimlik doğrulama protokollerinin güvenliği ve gizliliğine odaklanıyoruz. Öncelikle, Vaudenay'ın önceki modelleri birleştiren ve formal güvenlik analizleri için yeni olgun bir model sunan mahremiyet modelini genişletiyoruz. Modeli geliştirmek için RFID protokollerinde rastgele sayı üreteçlerinin (RNG'ler) güvenliğini analiz etmeyi sağlayan RANDOMEYE adlı yeni bir rakip sınıfı tanımlıyoruz. Genişletilmiş modelimizi literatürdeki iki popüler RFID protokolüne başarıyla uyguluyoruz. İkincisi, güvenlik ve performans açısından eliptik eğri kriptografisine (ECC) dayanan en güncel RFID kimlik doğrulama protokollerini kapsamlı bir şekilde inceliyoruz. Bu çalışmalar-dan bazıları, protokollerinin tüm genel güvenlik ve gizlilik özelliklerini sağladığını iddia ediyor. Formal güvenlik analizleri ile ispat ediyoruz ki bu protokoller iddia ettiklerinin aksine ileri ve/veya geri mahremiyet sağlayamamaktadırlar. Ardından, güvenli, mahremiyet etkin ve verimli bir protokol öneriyoruz. Ayrıca önerilen protokolümüzün kapsamlı bir güvenlik ve performans analizini sunuyoruz ve onu mevcut protokollerle ayrıntılı olarak karşılaştırıyoruz. Dahası, önerimizi uygulanabilirliğini kanıtlamak için gerçek bir RFID sisteminde gerçekliyoruz. Bildiğimiz kadarıyla, önerdiğimiz protokol, geri ve ileri mahremiyet de dahil olmak üzere tüm ortak güvenlik ve gizlilik özelliklerini sağlayan gerçek dünya ortamında gerçekleştirilen en verimli ECC tabanlı RFID kimlik doğrulama protokolüdür.

Anahtar Kelimeler: Gerçekleştirme, Mahremiyet, Açık Anahtar Kriptografisi, RFID Protokolleri, Güvenlik.

INTRODUCTION

Radio Frequency IDentification (RFID) has become one of the most emerging wireless technologies used in order to identify and authenticate objects, animals and people in recent years. The popularity of RFID has been rising day by day with the expeditious development of the Internet of Things (IoT) paradigm. In fact, the first idea of IoT was originated from a network of objects connected by RFID, and IoT tells us that "anything that can be connected, will be connected". It is predicted that by 2020, the number of daily life things that will be connected to each other will reach about 50 billion [1]. This means that RFID will continue to have a high impact on our daily activities and behaviors, and penetrate in our everyday lives rapidly by providing easy, efficient, cheap, secure and private connections of "things" which also includes people [2]. Although RFID technology is used in numerous real-world applications such as payment systems, healthcare system, e-passports, e-voting, national e-ID management, smart homes, access control, manufacturing, asset management, supply chain, etc. [3–7], RFID is still regarded to be its infancy today [8]. It is also considered that near-field communication (NFC) technology in smart phones is a new up-to-the-minute opportunity for RFID technology and we are on the doorstep of a new RFID era [9, 10].

Security and privacy concerns arise since a tag communicates with a reader over an insecure wireless channel. Tag impersonating, tracking (forward and backward), eavesdropping, replay, man-in-the-middle (MiTM) and denial of service (DoS) attacks can be performed by an attacker using the messages transmitted in the air [11]. Implementing heavy cryptographic algorithms to overcome these issues is a challenging task due to the limited capabilities of low-cost RFID tags [6, 10, 12–14]. For protocol designers, such constraints enforce a trade-off between security and practicality. Furthermore, over the past few years, numerous authentication protocols have been proposed so as to mitigate security and privacy concerns for RFID systems [15]. Most of the new protocols claimed that they were impregnable against every type of attack, providing different RFID system features such as scalable identification, tag ownership transfer, mutual authentication, robustness against noisy environments, reader corruption resiliency, etc. Unfortunately, many of them failed to satisfy the claimed security and privacy properties [15–18].

Public key (asymmetric) cryptography (PKC) can bring elegant solutions to security and privacy problems stated above. Especially nowadays, elliptic curve cryptography (ECC) is preferred in various RFID authentication schemes in order to reduce the key sizes, memory storage, and computation cost. Many protocol designers think that using ECC in their

designs efficiently and achieve security and privacy properties (see Section 2.2). Although some researchers have doubts that PKC might not be affordable for constraint tags, the feasibility of using ECC in the tags is shown in [19–23]. Moreover, both privacy and scalability in RFID systems are more easily accomplished by using PKC rather than symmetric cryptographic blocks [23, 24].

On the other hand, privacy models have been presented to systematically analyze the security and privacy of proposed authentication protocols. Such an evaluation is theoretically accomplished based on the privacy models to examine the security, anonymity and untraceability properties before using an RFID protocol in real-life systems. Recently, several models have been proposed to formalize security and privacy in the context of RFID systems [25–33]. A privacy model should be detailed, attentive and flexible not to overlook the realities of practical RFID systems. Although it has been considered that Vaudenay's model [27] is one of the most evolved and well-defined privacy models, some papers have been published to ameliorate his model [29,32–34]. These results, to the best of our knowledge, have claimed that their improvements fulfill the missing parts of the model but the privacy model has still fractures. In our opinion, the design of a new, appropriate, complete, and flexible security and privacy model considering the various abilities of an adversary is an essential need. Most importantly, we have noticed that Vaudenay's model has not taken the misuse of random number generators into consideration and this is a new and different adversary ability especially for real-word scenarios introduced in this thesis.

Designers generally build the security and privacy of their protocols on the utilization of a random number generator (RNG) which is one of the most common primitive cryptographic functions. Even though designers regard RNGs as secure, their improper deployment might cause serious weaknesses in a protocol scheme. More importantly, many proposed RNGs that are asserted secure today, might be broken or become weaker in the near future. In the literature, presented RNG attacks [35–38] show that protocol designers should put care into the deployment of RNGs in order not to encounter security and privacy issues in their protocols.

The feasibility of using ECC in practice is important for real-life RFID applications. Recent works show the implementations of their protocols in different environments such as Wireless Identification and Sensing Platform (WISP), Field Programmable Gate Array (FPGA) [39, 40]. There are also some RFID tags that are presented for implementations in Java cards, BasicCard, Mifare Card, and NFC cards. Especially HF RFID tags including NFC (Near Field Communications) tags have been densely preferred for IoT security applications [41]. In particular, the BasicCard environment [42] offers good opportunities for RFID systems as a powerful development tool in simulation and implementation.

In this dissertation, we show that RNGs could be the weakest point in RFID authentication

protocols and misusing them can cause severe security and privacy issues. From this point of view, we first revisit and extend Vaudenay's privacy model [27] by introducing the notion of RNGs based on their improper usage. To do so, we formalize a new privacy level called RANDOMEYE privacy that is integrated into Vaudenay's model. We also claim that Vaudenay's model is not sufficient for some real-world scenarios. For instance consider the following case that are not covered by Vaudenay's model: An adversary obtains some random numbers in a scheme and predicts the outputs of the RNG or the RNG loses its randomness because of some reasons such as aging, environmental effects, etc. (see Section 2.3 for further some explanations and existing attacks about RNGs). Motivated by this need, we introduce a novel adversary class what we called RANDOMEYE and define a new random oracle $\mathcal{O}^{RNG}$. We further apply our enhanced model to two existing RFID schemes and analyze their security with respect to RANDOMEYE adversary class. First, we address the scheme by Song and Mitchell [43], and then the scheme by Akgün et al. [44]. We show that these schemes are vulnerable to RNG attacks and are not RANDOMEYE private according to our extended model. Namely, the adversary can obtain the secrets of the RFID tags by benefiting from the improper usage of RNGs. We point out that RNGs might be the bottleneck of many RFID schemes. We highlight that using RNGs to mitigate security and privacy concerns can be Achilles' heel of an RFID authentication protocol.

We also show that the existing works [39, 40, 45, 46] do not provide forward and/or backward privacy, contrary to their claim. We reveal the vulnerabilities of these schemes under Vaudenay's [27] formal privacy model by utilizing privacy games. We propose a new secure and privacy-friendly ECC based authentication protocol by improving [40]. We elaborately analyze our proposed scheme in terms of security and performance and our analysis indicates that our scheme achieves all well-known security and privacy properties. Moreover, we give the implementation results of our proposed protocol in a real-world RFID system in order to show the practicability and feasibility of our proposal. We present detailed security and performance comparisons between our protocol and the related existing schemes. To the best of our knowledge, in the RFID literature, we claim that only our proposal is the up-to-the-minute implemented and tested protocol that can efficiently satisfy all essential security and privacy requirements for an RFID system

Organization

This dissertation is organized in 5 chapters. In Chapter 1, we consider on RFID systems, RFID authentication protocols and cryptographic primitives explaining the important definitions and notions. In Chapter 2, we review the literature on RFID privacy models, ECC based RFID authentication protocols, and RNGs used in the RFID protocols. In Chapter 3, we propose and describe our new privacy model. We improve the Vaudenay's privacy model defining a novel adversary class called RANDOMEYE. Then, we provide security

and privacy analysis of two existing RFID protocols under the enhanced privacy model as case studies. In Chapter 4, we consider on the recent ECC based RFID authentication schemes and prove that these schemes do not provide forward and/or backward privacy as they claimed. Secondly, we present our proposed protocol and its security and privacy analysis in detail. Then, we introduce our simulation and implementation environment, and show how to simulate and implement our proposal in the environment. Finally, we demonstrate our results and give a comprehensive comparison. In Chapter 5, we concluded the dissertation.

1. BACKGROUND INFORMATION

This chapter presents some background information on several notions, definitions and terminology addressed in this dissertation. We primarily introduce RFID systems. Then, we focus on RFID authentications protocols and explain the crucial terms, definitions, and notions about the protocols. Thirdly, we give some cryptographic primitives mentioned throughout the dissertation for the sake of readability.

1.1. RFID Systems

RFID is a wireless communication technology using radio waves to exchange data between parties. RFID is also one of the most likely technologies to promote the Internet of Things (IoT) paradigm and is proliferated in many real-life applications such as access control, supply chain, hospital care system, automatic toll collection, payment systems, e-passport, vicinity/proximity cards, etc.

A simple RFID system consists of a tag (transponder), a reader (interrogator) and a back-end server. A tag basically has a microchip which stores data and an antenna used to transmit and receive messages through electromagnetic waves. Generally, it is considered that a back-end server is separated from an RFID reader and it acts as a mediator between the tags and the server for the communication.

A back-end server keeps all the information (secret keys, data, etc.) about tags. Furthermore, RFID tags can be categorized into active, passive and semi-passive tags. Passive tags do not have their own power source and energize their integrated circuit (IC) by using the waves transmitted by the reader.

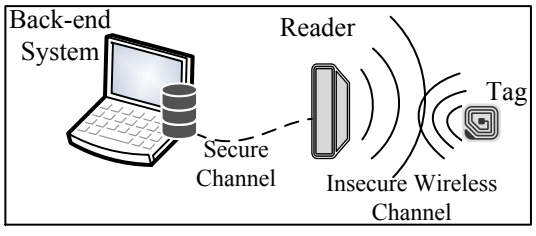

Figure 1.1. Architecture of a simple RFID system

Moreover, tags can also divided into four groups with respect to their operating frequency that usually depends on the availability of frequency bands and regulations: Low frequency (LF, 125-134.2 kHz and 140-148.5 kHz), high frequency (HF, ISM band at 13.56 MHz), ultra high frequency (UHF, 860-960 MHz) and microwave ($>$2.45 GHz) [10]. Passive low-

cost RFID tags of smaller sizes are highly preferred in many applications and this desire introduces some computation, energy and size restrictions on the tag. The production price of the tags is usually around $0.05 - $0.10 and the cost pressure is quite dominant on hardware capabilities [12]. Note that the reader and the back-end system (server or database) are regarded as only one entity throughout this dissertation. (see in Figure 1.1).

Security and privacy concerns in RFID systems result from exchanging sensitive information (i.e. credit card data, personal healthcare data) of tags with a reader in an insecure wireless channel. An adversary might be able to catch and change the messages transmitted in the air. The adversary can cause security and privacy issues with performing various attacks such as tag impersonating, reader spoofing man-in-the-middle (MiTM), tracking, replay, denial of service (DoS) attacks, etc. Therefore, many authentication protocols have been designed for mitigating security and privacy problems in RFID systems [15].

In the RFID literature, all protocol designers claim that their own schemes are secure and privacy-friendly while providing some other requirements such as mutual authentication and scalability. However, it is shown in the literature that most of them are not resistant to every type of attacks and do not efficiently accomplish a least one of security and privacy properties such as forward privacy, backward privacy, impersonation resistance, desynchronization resistance, etc [15, 47–52]. Also, some RFID privacy models are presented to methodically and formally analyze authentication schemes in terms of security and privacy. Although Vaudenay's model [27] is still successful and acceptable, in the course of time, several works have been proposed to improve and extend his model [29, 33, 34, 47].

1.2. RFID Authentication Protocols

In this section, we will briefly consider on RFID authentication protocols giving some definitions and explaining important notions.

Definition 1 (Identification). Identification is a process of declaring an identity without providing the attesting evidence in a protocol.

Definition 2 (Authentication). Authentication is a process that one party identify the identity of a second party using the attesting evidence in a protocol whilst the second party is active at the time the evidence supported.

Entity authentication methods might be consists of three main categories with respect to the attesting evidence [53]. The first one is "something known". In this category, the prover presents a piece of knowledge to the verifier such as password, personal identification number (PIN), etc. In the second one "something possessed", the prover provides a physical accessory such as smart card, password generator, etc. In the third category which is "something inherent", the prover assures a human inherited characteristics such as fingerprints, handwritten signatures, retinal patterns, etc. These authentication factors are

generally called in classical textbooks as ”what you know”, ”what you have”, and ”what you are”.

Definition 3 (RFID Protocol/Scheme). An RFID protocol (or scheme) is a set of procedures (or rules) providing data exchange between an RFID reader (*Verifier*) and an RFID tag (*Prover*) for a specific purpose.

Definition 4 (Mutual Authentication Protocol). A protocol is said to be a mutual authentication protocol if both parties (e.g. RFID readers and tags) involved in the protocol mutually authenticate each other.

RFID mutual authentication protocols are desired in many RFID applications instead of one-sided authentication schemes where only one party authenticates the other one [54].

Most of RFID protocols are usually expected to satisfy the five basic cryptographic goals: confidentiality, integrity, availability, authenticity, non-repudiation. Generally, these goals are also called security services. Confidentiality is a service that states preventing the content of information against unauthorized parties. Secrecy is a term synonymous with confidentiality and privacy [53]. Integrity is a service that states preventing unauthorized modification (insertion, deletion or substitution) of the data exchanged between parties. Availability is a service that states the ability to access the information among authorized parties when needed. Authenticity is a service that provides ensuring the only authorized parties communicate between each other. This service may be divided into sub-services entity authenticity and data origin authenticity that implicitly implies data integrity. Finally, non-repudiation is a service used to prevent denying previous actions of authorized parties [53].

1.2.1. Classification of the protocols

RFID authentication protocols are commonly classified into four classes based on the complexity of the cryptographic operations utilized in RFID tags: heavyweight (fully fledged) protocols, simple weight protocols (simple protocols), lightweight protocols, and ultra-lightweight protocols [50, 55–57].

Heavyweight protocols support symmetric and public-key algorithms and simple weight ones use hash functions and RNGs in the tag side. Unlike wireless protocols that require conventional cryptographic operations [24, 58–61] such as symmetric and public-key algorithms, restricted systems (in terms of computational power, storage, bandwidth, etc.) require lightweight or ultra-lightweight authentication protocols. Low-cost RFID systems are one of the prominent real-life applications of these protocols due to the capabilities and the price range of RFID tags.

Lower cost and smaller size demands for RFID tags enforce them to be some resource lim-

itations such as reduced number of logic gates, lower energy consumption, and low computational complexity. Lightweight and ultra-lightweight protocols need to be designed by taking into account the constraints of low-cost RFID tags. Hence, low-cost tags introduce many challenges in terms of security and privacy; numerous researchers have proposed protocols in order to obviate security and privacy concerns [15].

Extremely restricted RFID tags require ultra-lightweight protocols that only support bitwise operations (such as XOR, AND, OR, rotation, permutation, etc.) and are compliant to EPC Class-1 Generation-2 specification. Some of the famous ultra-lightweight protocols are SASI [55], LMAP [62], M2AP [63], EMAP [64] and Gossomar [65]. On the other hand, lightweight protocols use the same bitwise operations, as well as RNGs and Cyclic Redundancy Check (CRC) but no cryptographic hash functions. Several well-known protocols are presented in [66–68]. However, the restrictions mentioned above greatly limit aptitudes of RFID tags and cause security and privacy vulnerabilities. Avoine et al. [48] have evaluated and compared well-known lightweight protocols and indicated the security and privacy weaknesses. Zeeshan has also quite recently addressed the security and privacy issues in low-cost RFID systems in his Ph.D. thesis in [10]. However, public-key cryptography approaches in RFID authentication protocols present better solutions to security and privacy problems as well as they fulfill both constant-time identification and strong privacy [48]. Using PKC on low-cost RFID tags has been becoming more feasible day after day [69–71].

1.2.2. Threats on the protocols

We present some possible threats in RFID protocols to demonstrate the security and privacy issues of RFID systems clearly [72]. An adversary mainly aims to attack to destroy the security services and obtain the critical assets of the entities involved in an RFID authentication protocol. There are basically two types of adversary. One of them is passive adversary who is solely able to eavesdrop the communication of the insecure channel. The second one is active adversary who is able to manipulate the messages on the channel. The essential attacks [50, 61, 72–75] against an RFID system are listed and introduced below.

In a replay attack, an adversary firstly eavesdrops the messages between valid a reader and a tag during previous protocol executions. Then, she reuses the intercepted and collected past protocol messages within the next protocol sessions to successfully authenticate herself as a legitimate party or revealing the secrets of any parties.

In a man-in-the-middle (MiTM) attack, an adversary intercepts and modifies the messages between a tag and reader to deceive at least one of the valid party as a legitimate party.

In a tag impersonation attack, an adversary impersonates a valid tag to the valid reader without having the internal secrets of the tag and the reader authenticates her as a legitimate

tag.

In a reader spoofing attack, an adversary deceives a valid reader acting as an authorized tag and the reader authenticates her.

In a de-synchronization attack, an adversary aims to break the synchronization between a tag and a reader by modifying or obstruct the update messages between each other. Hence, the internal values of the parties are not sufficient and the authentication mechanism is collapsed between legitimate parties.

In a denial of service (DoS) attack, an adversary targets the availability between a reader and a tag engaging the reader, the tag or both parties. This attack is closely related to de-synchronization attack because if the availability is not provided for further protocol sessions, it means that de-synchronization attack is performed. The de-synchronization attack is said to be a kind of DoS attacks [74, 76].

In a cloning attack, an adversary aims to obtain the secrets of a tag or a reader to form an invalid party that is able to pass the authentication against a legitimate party.

1.2.3. Security and privacy requirements of the protocols

In the previous section, we have talked about the possible attacks on RFID authentication protocols [72]. In fact, being resistant to the each mentioned attacks is one of the essential security requirements of an RFID system such as replay attack resistance, MiTM attack resistance, tag impersonation attack resistance, reader spoofing attack resistance, de-synchronization attack resistance, DoS attack resistance, cloning attack resistance.

In addition, for a secure and private RFID system, authentication scheme should fulfill the following crucial requirements (security and privacy goals) or properties: mutual authentication, confidentiality, integrity, availability, location privacy, forward privacy, backward privacy. In particular, the primary target of an adversary is violating the security and privacy goals of an authentication scheme by exploiting its vulnerabilities. In this respect, satisfying a security or privacy requirement might sometimes corresponding resistance of one or more forenamed attacks.

Mutual authentication is said to be provided if both legitimate tag and reader successfully authenticate each other. Tag authentication and reader authentication are accomplished if an adversary cannot authenticate herself to the reader or tag, respectively.

Confidentiality is said to be provided if no adversary knows the content of the messages transmitted in an RFID protocol execution and she never reveals the secrets of a legitimate entity. In other words, this property is also satisfied if no adversary can compute any function of the secret data from the transactions of a protocol [77].

Integrity is said to be provided if no adversary alters (insertion, deletion or substitution) of the messages exchanged between two legitimate entities in an RFID protocol execution.

Availability is said to be provided if the successive communication between two parties should be always accomplished. Availability is also a service that states the ability to access the information among authorized parties when needed.

Tag anonymity is said to be provided if an adversary is not able to identify a specific tag while eavesdropping and manipulating the messages transmitted between the tag and a valid reader.

Location privacy is said to be provided if tag anonymity is achieved and the adversary is not able to trace the tag anymore. Therefore, location privacy is more comprehensive notion because this property requires both tag anonymity untraceability [78].

When considering the traceability of a tag in its past and future protocol transactions, the notion of forward and backward privacy (or untraceability) is appearing [78,79]. In general, untraceability means indistinguishability of two different tags in an RFID scheme.

In the RFID literature, the notion of untraceability is categorized into two types: backward untraceability and forward untraceability [79–81]. Sometimes they are also called forward secrecy/privacy and backward secrecy/privacy, respectively. Notably, these terms express the opposite meaning of their word meaning. For instance, backward privacy means keeping the privacy of an RFID scheme, even if the tags in the scheme had been corrupted in the past. Actually, "backward" and "forward" terms are originated from the certain time that an adversary can obtain the internal privileges of an RFID tag (i.e. tampering or having ownership transfer) [79]. She is also able to record both a set of backward and forward protocol interactions so that she can destroy the tag privacy.

Let $prb_s^{Adv}\left(t, \Phi_{t_0}^T\right) \rightarrow y$ be a function that outputs the probability of an adversary Adv to successfully trace a tag T at time t knowing $\Phi_{t_0}^T$, where $\Phi_{t_0}^T$ denotes the whole internal knowledge (e.g. secret keys and parameters) of T at time t_0 (i.e. Adv can obtain $\Phi_{t_0}^T$ by corrupting T at time t_0) and $0 \leq y \leq 1$.

Definition 5 (Backward Untraceability / Forward Privacy). An RFID scheme satisfies backward untraceability property, if $prb_s^A\left(t, \Phi_{t_0}^T\right)$ is negligible, where $t < t_0$.

Definition 6 (Forward Untraceability / Backward Privacy). An RFID scheme satisfies forward untraceability property, if $prb_s^A\left(t, \Phi_{t_0}^T\right)$ is negligible, where $t > t_0$.

It has been considered that both backward and forward privacy are the essential security requirements for an RFID authentication scheme. Lim and Kwon [79] introduce forward untraceability property and argue that in general, providing this property for an RFID

scheme is harder than accomplishing backward untraceability. They concentrated on the importance of forward untraceability and state that it is at least as crucial as backward untraceability for RFID authentication schemes.

1.2.4. Operational requirements of the protocols

For a real-life application of an RFID system, efficiency requirements of an RFID authentication protocol must be achieved while providing the security and privacy goals. The notion of efficiency closely relates to computational cost and communication cost. Briefly, communication overhead and the number of protocol rounds determine the communication cost, and the operations used in the protocol determine the computational cost. At this point, scalability which requires efficient tags searching in the database is a vital requirement of RFID systems to directly affect the computational workload [72].

Scalability is said to be provided if the computational workload does not increase while the number of the tags are rising during the authentication process [82–84]. In particular, PKC based RFID protocols can easily achieve constant time identification [48, 70].

1.3. Cryptographic Primitives

In this section, we introduce some preliminaries on main cryptographic topics utilized in this dissertation such as private-key and public-key cryptosystems, digital signatures, hash functions and random number generators. We also give the definitions of some cryptographic notions and terms.

Definition 7 (Cryptosystem). A cryptosystem is a quintet (P,C,K,E,D) satisfying the following conditions:

– P is the plaintext space that includes a finite set of probable plaintexts; C is the ciphertext space that includes a finite set of probable ciphertexts, and K is the key space that includes a finite set of probable keys;

– $\forall k \in K$, $E_k \in E$ is an encryption algorithm $E_k : P \times K \rightarrow C$, and corresponding $D_k \in D$ is a decryption algorithm $D_k : C \times K \rightarrow P$ such that $D_k(E_k(m)) = m$ for each plaintext $m \in P$.

– $\forall k_e \in K$, $\exists k_d \in K$ such that $\forall m \in P$, $D_{k_d}(E_{k_e}(m)) = m$.

A symmetric (or private-key) cryptosystem is defined by a cryptosystem satisfying following properties (i) $k_e = k_d$, or (ii) it is "easy to derive "k_d from k_e. On the other hand, an asymmetric (or public-key) cryptosystem (PKC) is defined by a cryptosystem satisfying that it is "hard" to derive k_d from k_e. Generally speaking, determining the private key k_d from the corresponding public key k_e is computationally infeasible.

In this respect, we should explain hardness (commonly known as hard problems) term in a cryptologic manner. This term is closely related to the ability of an adversary (generally refers to an algorithm) attacking a cryptosystem. Turing machines might be used to describe all types of algorithms. Let n be an input string, $|\, n \,|$ be the length of the string, and p denotes some polynomials, a polynomial time Turing machine is one that halts within $p(|\, n \,|)$ steps for any n string. The problem solved by a deterministic polynomial-time Turing machine is polynomial time problem and its complexity class is P.

Instead, considering the average case complexity is far better appropriate to evaluate cryptologic algorithms or problems. At this point, it is good to introduce the probabilistic Turing machines. In a nutshell, a probabilistic Turing machine makes uniformly at random choices for each step during its execution among possible ones. In other words, a probabilistic polynomial time (PPT) Turing machine (M) is a Turing machine that always halts in x^c steps by taking an input x with a random bit string r, where c is a nonnegative integer [85, 86].

Definition 8 (Hard Problem). Let *Adv* be an algorithm. A problem is said to be "hard" if it cannot be solved by a PPT *Adv* according to the input size.

1.3.1. Advanced encryption standard algorithm

In 2001, Vincent Rijmen and Joan Daemen's symmetric block cipher algorithm (Rijndael algorithm) was announced as a new Advanced Encryption Standard cipher (shortly called AES) instead of DES by The National Institute of Standards and Technology of the United States (NIST) after a worldwide competition [53, 87].

AES is a kind of block cipher algorithm that encrypts and decrypts the 128 bits length of blocks using different key size 128, 192 or 256 bits. Generally, the key size utilized in AES name the algorithm as AES-128, AES-192, or AES-256. The interface block diagram (input/output) of AES algorithm is illustrated in Figure 1.2. Although increasing the key sizes strengthen the algorithm (means higher security level), AES requires more round iterations for the encryption.

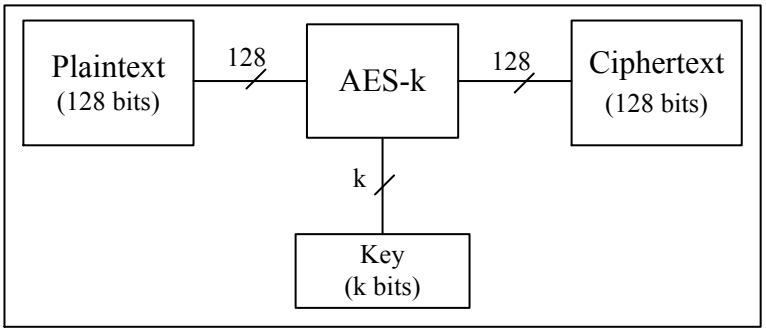

Figure 1.2. Interface block diagram of AES algorithm

Today, the AES is still greatly preferred in many application areas and it has no significant discovered security vulnerability [77, 88]. The US National Security Agency (NSA) suggests different key sizes of the AES algorithm depends on the security level of the information in order to protect. NSA says that all key sizes of AES (specified in FIPS 197 [87]) are sufficient from the lower security level up to the SECRET level but either the 192 or 256 key sizes of AES is required for TOP SECRET information [89]. Finally, one of the most important reasons that AES being a very popular symmetric key algorithm is higher efficiency in both hardware and software implementations besides security strength [90–92].

1.3.2. Elliptic curve cryptography cryptosystems

Elliptic curve cryptography is public-key cryptography based on elliptic curves over Galois or finite fields [93]. More than 30 years ago, the use of EC in cryptography is firstly discussed by Koblitz [94] and Miller [95], independently. Today, more than billions of wireless communication systems prefer ECC based solutions to fulfill the security requirements in different sectors such as financial services, health care, government services, etc., because they need efficient and secure asymmetric cryptosystem for confidentiality, integrity, authentication, privacy, non-repudiation (i.e. signature), etc. The advantages of ECC for wireless security is briefly overviewed in [96, 97].

As follows, initially the theory of ECC is briefly summarized, then security and benefits of ECC are discussed.

Theory of ECC: An elliptic curve (E) used for cryptographical purposes can be generally defined over a prime finite field $\mathbb{F}_p$ (or Galois field) includes a group of points (x,y) that satisfies $y^2 \equiv x^3 + ax + b(mod p)$ equation, where $(a,b) \in E$, $\triangle = 4a^3 + 27b^2 \implies \triangle \not\equiv 0(mod p)$ and p is a large prime number. The EC cyclic group is formally defined as $E(\mathbb{F}_p) = \{(x,y) : x,y \in E\,(a,b)\} \cup \{\mathscr{O}\}$, where $\mathscr{O}$ denotes point at infinity and satisfies the following group properties. Let $\forall P,R,S \in E\,(\mathbb{F}_p)$ and $P = (x_0,y_0)$, $R = (x_1,y_1)$, $S = (x_2,y_2)$,

- Existence of identity: $R + \mathscr{O} = \mathscr{O} + R$
- Existence of inverse: $R + (-R) = \mathscr{O}$, where $-R = (x_1, -y_1)$. Also, $\mathscr{O} = -\mathscr{O}$
- Closure: $R + S = (x_3,y_3) \in E\,(\mathbb{F}_p)$
- Associativity: $P + (R + S) = (P + R) + S$
- Point doubling: $R \neq (-R) \implies R + R = 2R = (x_4,y_4) \in E\,(\mathbb{F}_p)$

The EC point addition is shown as $R + S = (x_3,y_3)$. Also, the EC point doubling is shown as $R + R = 2R = (x_4,y_4) \in E\,(\mathbb{F}_p)$, where $R \neq (-R)$. Lastly, the EC point multiplication is defined as $Q = P + P + P + ... + P = nP$, where $Q,P \in E\,(\mathbb{F}_p)$ and $n \in \mathbb{Z}$. This operation corresponds to adding P by itself n times.

Order of an EC group refers to the number of points (elements) in that group and can be easily computed by Schoof's algorithm. Actually, ECC uses cyclic subgroups formed by EC with having cyclically repeated points. This type of groups has a base point (generator). Note that Schoof's algorithm cannot be used for finding the order of the subgroup. Let G be a cyclic subgroup of $E(\mathbb{F}_p)$ with order k and generator P, then $nP = \mathcal{O}$. Furthermore, a cofactor of G is $h = N/k$, where N is the order of $E(\mathbb{F}_p)$ and $h \in \mathbb{Z}$ because of Lagrange's theorem. In fact, cryptographers want a high order of an EC subgroup so before they find a generator, they first choose a large enough order then try to reach a suitable generator.

Domain Parameters: ECC domain parameters are all the elements defining an elliptic curve such as base point, prime order, cofactor of the base point, etc.

It is generally assumed for an RFID system that both tag and reader in the setup phase agree on the curve to securely and efficiently execute a protocol. We prefer one of the secure standard curves, namely the ECC Brainpool [98] to use in our protocol design. Below, we give the domain parameters (hexadecimal representation) of brainpoolP160r1 standard elliptic curve as an example, where satisfying the Equation 1.1. Briefly, p is the prime that specifies the EC field over $E(\mathbb{F}_p)$. a and b are the coefficients of the Equation 1.1. $G = (x, y)$ is the base point (generator point of E), where x and y denote the coordinates, respectively. q is the prime order of the group generated by the base point G and the co-factor of G is h, i.e. $\#E(\mathbb{F}_p)/q$.

$$E : y^2 \equiv x^3 + ax + b \quad (mod\, p) \tag{1.1}$$

p = E95E4A5F737059DC60DFC7AD95B3D8139515620F

a = 340E7BE2A280EB74E2BE61BADA745D97E8F7C300

b = 1E589A8595423412134FAA2DBDEC95C8D8675E58

x = BED5AF16EA3F6A4F62938C4631EB5AF7BDBCDBC3

y = 1667CB477A1A8EC338F94741669C976316DA6321

q = E95E4A5F737059DC60DF5991D45029409E60FC09

h = 1

Security and Benefits of ECC: Using an EC scheme offers several advantages: smaller key sizes with respect to the other known PKC algorithms (at the same security level, see Table 1.1 [99]), higher speed, reduced memory storage as well as consumed power and bandwidth efficiency. Thus, these benefits make ECC desirable in many high-security usage areas of asymmetric cryptographic schemes such as key agreement, encryption and digital signatures [96, 100–104].

The security of ECC-based schemes depends on the hardness of the EC discrete logarithm problem (ECDLP). Although many cryptoanalysts consider their attention on ECC based schemes.

Table 1.1. Key size (bits) comparisons for equivalent security levels [99]

Minimum Strength (bits)	Symmetric Algorithm	RSA and DL Group (bits)	ECC (bits)
80	Two-key 3DES	1024	160
112	Three-key 3DES	2048	224
128	AES	3072	256
192	AES	7680	384
256	AES	15360	521

Definition 9 (Elliptic curve discrete logarithm problem (ECDLP)). Given $P,Q \in E\left(F_q\right)$ and $Q = kP$ where $k \in [1, q-1]$ and q is the point order. Then, it is hard to compute k by an algorithm in polynomial-time.

The DH problem is also synonymously named as DH assumption and the assumption is sometimes called the Computational DH (CDH) assumption to emphasis the difference of Decisional DH assumption (DDH) [86].

Definition 10 (Computational Elliptic Curve Diffie-Hellman Problem). Given P,R, $S \in E\left(\mathbb{F}_q\right)$, $S = sP$ and $R = rP$ where $s,r \in [1, q-1]$ and q is the order of the base point P. Then, it is hard to compute srP by an algorithm in polynomial-time.

At this point, we would like to define DDH for the reader to figure out the nuance between CDH and DDH.

Definition 11 (Decisional Elliptic Curve Diffie-Hellman Problem). Given $P,R\,,S \in E\left(\mathbb{F}_q\right)$, $S = sP$ and $R = rP$ where $s,r \in [1, q-1]$ and q is the order of the base point P. Then, it is hard to distinguish srP from a random EC point zP in polynomial-time even if the points $S = sP$ and $R = rP$ is given, where $z \in_R [1, q-1]$.

Evidently, these problems satisfy $DL \Leftarrow DH \Leftarrow DDH$. For specific groups, DHP is sometimes called Computational Diffie-Hellman assumptions because DHP is assumed as a hard problem. Note that the security of any scheme depends on under the assumption evaluated. For instance, one scheme is secure under DL assumption while it might be insecure under the computational DH assumption. Moreover, public keys in ECDH schemes might be either static or ephemeral (ECDHE).

1.3.3. Elliptic curve diffie-hellman

ECDH (a variant of the DH scheme) is a secure key agreement scheme whereby two or more entities can agree on a secret key by using ECC over an insecure channel. Both

X		**Y**
$[a,b,p,G,n,h]$		$[a,b,p,G,n,h]$
Choose a private key:		Choose a private key :
$k_x \in_R [1,n-1]$		$k_y \in_R [1,n-1]$
$k_X = k_x G$		$k_Y = k_y G$
	$\xrightarrow{k_X}$	
	$\xleftarrow{k_Y}$	
$k_{XY} = k_x k_Y$		$k_{YX} = k_y k_X$
	$k_{XY} = k_{YX} = k_x k_y G$	

Figure 1.3. Elliptic curve Diffie-Hellman key exchange scheme

entities already have pre-shared public keys of each other. They use their own private keys to recover the shared key, but an adversary cannot calculate the shared key from the public information. This scheme is briefly depicted in Figure 1.3.

1.3.4. Elliptic curve digital signature algorithm

ECDSA is a variant of the Digital Signature Algorithm (DSA) that uses ECC. ECDSA is used for authentication, non-repudiation, and integrity. Therefore, the source of the message is authenticated, the entity that transmitted the message cannot deny it and the integrity of the message is ensured over an insecure channel. In this scheme, basically, the signer signs the message by using its own secret key and the verifier verifies the signature with a public key of the signer by using ECC. The security of ECDSA is based on ECDLP. Moreover, ECDSA is more effective than other known schemes such as RSA and DSA. It is accepted by ANSI, IEEE, and NIST. Lastly, the description of ECDSA is briefly depicted in Figure 1.4.

Singing	Verifying
$[a,b,p,G,n,h]$	$[a,b,p,G,n,h]$
private-public key pairs: $[k_{prv}, k_{pub} = k_{prv}G]$	public key : $[k_{pub}]$
message: m	message and signature: (m,R,s)
Choose a random number: $r \in_R [1,n-1]$	$v_1 = s^{-1}m \ (mod\, n)$
$R = rG = (x,y)$	$v_2 = s^{-1}x \ (mod\, n)$
$s = r^{-1}(m + k_{prv}x) \ (mod\, n)$	$V = v_1 G + v_2 k_{pub}$
signature:(R,s)	checking: $V \stackrel{?}{=} R$
Proof of correctness:	
$V = v_1 G + v_2 k_{pub} = s^{-1}mG + s^{-1}xk_{pub} = s^{-1}(mG + xk_{prv}G) = r^{-1}G = R$	

Figure 1.4. Elliptic curve digital signature algorithm scheme

1.3.5. Cryptographic hash function

Informally, a cryptographic hash function can be defined as a deterministic algorithm that maps an arbitrary size of input at fixed size output. These functions are used in many cryptographic schemes to provide the integrity of data. It is formally defined as below [86].

Definition 12 (Hash Function). A hash function $H : x \mapsto y$, where $x \in \{0,1\}^*$ and $y \in \{0,1\}^n$, mapping a bit string with an arbitrary length to a fixed length (n) one, where $n \geq 0$. For a given string x, $H(x)$ should be computable in polynomial time. Moreover, a function H is also called a cryptographic hash function, if H satisfies the following well-known requirements:

– Pre-image resistance (one-wayness): Given a hash value y, it is hard to find any bit string x such that $H(x) = y$. By providing this property, the functions are resistant to pre-image attacks.

– Second pre-image resistance (weak collision resistance) : Given a bit string x , it is hard to find any bit string $x' \neq x$ such that $H\left(x'\right) = H(x)$. By providing this property, the functions are resistant to second pre-image attacks.

– Collision resistance (strong collision resistance): It is hard to find two different bit strings $\left(x, x'\right)$ such that $H(x) = H\left(x'\right)$.

If H is resistant against collision attacks, it always provides second pre-image resistance, otherwise, but opposite implication might not be valid. This assumption is theoretically true, however, it is recommended that cryptographic hash functions need to satisfy all three requirements in practical applications. In practice, there are several known cryptographic hash functions with different digest sizes from 128 bits to 512 bits e.g. SHA family MD5, BLAKE, etc.

1.3.6. Random number generators

Generating random numbers is essential for cryptographic operations. Producing and using random numbers is always critical for a cryptographic algorithm. In particular, generating truly random numbers is problematic and as far as we know that there is not any mechanism to prove the true randomness. The generated true random numbers are only the numbers that can pass the known randomness tests.

There are two types of random number generator: pseudo-random number generator (PRNG) and truly random number generator (TRNG). TRNG is an algorithm that generates random numbers from a natural source of randomness. PRNG, also known as deterministic random number generator (DRNG), is an algorithm for generating random numbers with a provided initial value called a seed. The output of the PRNG is called a pseudo-random bit sequence. The output of a PRNG is much longer than the length of the seed. In addition to this, the output of a PRNG seems to be random because it has to be statistically indistinguishable from random values and it is assumed to be unpredictable when its seed is not known.

Two general conditions are required from the security perceptive for a pseudorandom ran-

dom generator: (i) the output of a PRNG should be statistically indistinguishable from truly random sequences, (ii) the next output of the sequence should be unpredictable to an adversary with limited computational resources. Theoretically, the next output can be predictable with a negligible probability such as 2^{-80}. In fact, the minimum security requirement is that the length of the random seed has to be sufficiently large (s-bit) to be infeasible for the adversary to search over a 2^s sized space (s is called the security parameter). In other words, the complexity of that attack is 2^s.

It is impossible to prove that the output of an RNG is random but there are various statistical tests that measure the quality of an RNG. This is accomplished by taking sample output sequences and apply the tests. The tests are probabilistic so they determine whether the samples look like a truly random sequence or not. If the generator fails, the output is regarded to be non-random. On the other hand, if an RNG passes all the tests, it is not rejected as being non-random. The five basic tests are (i) frequency test (mono bit test), (ii) serial test (two-bit test), (iii) poker test, (iv) runs test, (v) auto-correlation test [53]. Detailed information about tests, generators, algorithms, and definitions are presented in [53]. Moreover, some institutes, research centers, government agencies or organizations have specified some criteria to control the randomness of RNGs. For instance, the German Federal Office for Information Security has established several procedures for quality assessment of RNGs [105].

Computational Capabilities

Hashcat is the well-known fastest password recovery cracker [106] and different versions are available for Linux, OSX, and Windows. It also comes in two variants: CPU-based (Hashcat password recovery tool) or GPU-based (oclHashcat, accelerated tool). oclHashcat is a GPU-based multi-hash cracker using a brute-force attack (implemented as a mask attack), combinator attack, dictionary attack, hybrid attack, mask attack, and rule-based attack.

Table 1.2. Performance of oclHashcat

Hash Type	PC1	PC2	PC3	PC4
MD5	8581 Mh/s	2753 Mh/s	135232 Mh/s	92672 Mh/s
SHA1	3037 Mh/s	655 Mh/s	42408 Mh/s	31552 Mh/s
SHA256	1122 Mh/s	355 Mh/s	16904 Mh/s	12288 Mh/s
SHA512	414 Mh/s	104 Mh/s	5240 Mh/s	4552 Mh/s

The performance of oclHashcat in different operating systems (PC1, PC2,PC3 and PC4) for MD5, SHA1, SHA256, SHA512 is depicted in Table 1.2 [106], where PC1: Windows 7, 32 bit Catalyst 14.9 1x AMD hd7970 1000MHz core clock oclHashcat v1.35, PC2: Windows 7, 64 bit ForceWare 347.52 1x NVidia gtx580 stock core clock oclHashcat v1.35, PC3: Ubuntu 14.04, 64 bit ForceWare 346.29 8x NVidia Titan Xstock core clockoclHashcat

v1.36 and PC4: Ubuntu 14.04, 64 bit Catalyst 14.9 8x AMD R9 290X stock core clock oclHashcat v1.35.

It is seen that PC3 can do 135232 Mh/s against MD5, which approximately accounts to 135 million tries per millisecond. Hence, if the same computer is used for exhaustive search, less than 32 ms will be required to find the result matching the output of 32-bit PRNG.

2. RELATED WORK

In this chapter, we give an overview of the existing RFID privacy models, ECC based RFID authentication protocols, and RNGs used in RFID protocols.

2.1. RFID Privacy Models

Privacy models are proposed as a base for analyzing the security and privacy of authentication protocols in a methodological manner. For this purpose, the privacy models formally define some properties such as RFID schemes, security and privacy prerequisites of the schemes and abilities of an adversary. In this context, Avoine et al. has published a framework to formalize privacy in RFID protocols in 2005 [107]. Avoine also extended the previous model in his thesis [25]. Then, Juels and Weis modified Avoine's model by adding a side channel information attribute [26]. Furthermore, different model definitions were provided in [108, 109]. Although there were several other attempts to design useful, proper and complete privacy model to represent and analyze RFID systems, the models did not consider all, or miss some important adversary properties (corruption, using side channel information, etc.) and they did not appropriately model an RFID scheme in terms of authentication, identification, protocol execution, etc. However, in 2007, Vaudenay has proposed a well-designed and relatively complete privacy model that has been quite popular among many protocol designers [27]. In time, several researchers have improved Vaudenay's model [29, 32–34] for which more detail is provided below.

In 2010, Avoine et al. [29] introduced the notion of time and formalized it by modifying Vaudenay's model with a new privacy class called TIMEFUL privacy. They show that an adversary can trace an RFID tag by only following the time that a reader has taken to authenticate the tag. According to their model, an adversary can call timer oracle to learn the spent time for its overall computations during authentication and can distinguish the tag. They stated that if an RFID protocol is TIMEFUL-private, an adversary cannot obtain anything about the tag identity using time information.

In 2011, Akgün et al. [32] defined the notion of forward untraceability by extending Vaudenay's model. In their model, they emphasized the relay of valuable information on each communication round of the protocol and they claim that Vaudenay's model does not represent real-world settings because an adversary can miss some communication rounds due to some reason such as low signal to noise ratio. They applied their revised model to analyze some existing RFID protocols and showed that the schemes are not resistant to forward untraceability and server impersonation as claimed.

In 2014, Kardaş et al. [33] improved Vaudenay's model by claiming that an adversary has the capability to corrupt a tag at most k times. Hence, they introduced k-strong privacy that is an extension of the privacy classes of Vaudenay's model and is positioned between strong privacy and destructive privacy.

Hermans et al. [34] modified Vaudenay's model by introducing insider privacy notion based on the insider attack that is first discussed for RFID schemes by Deursen et al. [59]. They analyzed some existing RFID protocols to show the applicability of their model. Moreover, they propose a new RFID authentication protocol that provides wide-forward-insider privacy.

2.2. ECC Based RFID Authentication Protocols

This section introduces previous works in ECC based RFID authentication protocols and outlines the contributions of this thesis. To solve the various security and privacy problems in RFID systems, countless RFID protocols have been published. A recent comprehensive survey of related work about these protocols is provided in Avoine's RFID Security and Privacy Lounge [15]. Among all researchers that used public-key cryptography (PKC), nearly all preferred ECC-based protocols because of their ability to provide stronger security with smaller key sizes, as well as lighter and efficient computations.

In 2005, Wolkerstorfer [69] asserts that ECC implementation in RFID tags was suitable. In 2006, Tulys and Batina [70] firstly propose an ECC-based RFID identification scheme using the Schnorr identification protocol [110] by referring to the conclusion of Wolkerstorfer's work. They claim that their scheme is secure against cloning attacks. But, the implementation of this protocol is caused by security and privacy vulnerabilities. In the interactive phase, an adversary can obtain the information to calculate the ID-verifier and she can track the tag. The protocol has also a scalability problem. In the authentication phase, the verifier has to search the many public keys for each tag. Moreover, this protocol does not provide mutual authentication and anonymity [111].

Later, Batina et al. [22] propose a new scheme by applying Okamoto's identification protocol [112] to improve security and privacy. They also aim to discuss the feasibility of ECC based RFID identification protocols and present the implementation of Okamoto's protocol as an example. However, Batina et al.'s protocol does not solve the security, privacy and efficiency issues [113]. The adversary still can obtain the ID-verifier and track the tag. In addition, the forward privacy is not provided in Batina et al.'s scheme similar to the situations in [70].

Lee et al. [111] show the weaknesses of Schnorr's and Okamoto's identification problems and propose a new RFID authentication protocol named EC-RAC using ECC to mitigate the security and privacy flaws mentioned above. But it is shown that this protocol has

security and privacy issues, and is vulnerable to tracking attack, MiTM attack, algebraic attacks, etc. [23, 114–117]. Similarly, the protocol provides only one-way authentication.

In 2009, Lee et al. [118] revise the EC-RAC protocol [111] and propose six different RFID authentication protocols by expanding the EC-RAC protocol. They state that their protocols are secure against common attacks, but each protocol provides different security properties. In 2010, Lee et al. [116] address the existing vulnerabilities of EC-RAC protocols and present a new efficient searching scheme for the RFID reader so as to query for a specific tag while protecting the tag's privacy.

In 2011, Zhang [119] et al. present an ECC-based randomized key RFID authentication protocol to improve EC-RAC and Schnorr protocols to defeat their weaknesses. The proposal focuses on finding a way to solve the tracking attack effectively. However, this scheme is defenseless to active-tracking attack. Furthermore, updating tag information increases the computation complexity of the back-end server and causes scalability problems in this scheme. It also lacks mutual authentication [120]. Lv et al. [117], in 2012, show the weaknesses of EC-RAC protocols and propose three ECC-based RFID protocols which are the revision of EC-RAC protocols to overcome tracking attack. Later, An et al. [121] demonstrate that Lv et al.'s protocols are not secure against MiTM attack.

In 2014, Liao and Hsiao [73] present an ECC-based RFID authentication scheme to satisfy the essential requirements of RFID systems including mutual authentication, anonymity, forward privacy, confidentiality, and scalability. But, it is shown that this scheme is inadequate in terms of computational cost and memory storage [120, 122–124]. Zhao [125] proposes a new protocol and shows that Liao and Hsiao's protocol suffers from the key compromise attack in which the adversary can obtain the private key stored in the tag. It is shown that Liao and Hsiao's protocol does not achieve any security and privacy properties in [126]. Chien [120] also proves that Liao and Hsiao's protocol is vulnerable to active tracking attack. Zhao's scheme does not provide tag anonymity, location privacy, data integrity, backward and forward privacy [39, 40, 127].

Later, Chou [113], in 2014, designs a new and efficient RFID mutual authentication protocol based on ECC. Unfortunately, this scheme is defenseless against tag impersonation, cloning, and tracking attacks and it also does not satisfy tag anonymity, forward privacy and mutual authentication [40, 128, 129]. Zhang and Qi [129] point out that Chou's scheme does not provide tag information, backward and forward privacy. They also propose an enhanced new RFID scheme based on Chou's protocol to overcome the vulnerabilities of his scheme. But, it is shown that Zhang and Qi's scheme does not provide location privacy, backward and forward privacy [39, 40]. In the same year, He et al. [123] propose a new ECC RFID scheme that integrated with an ID verifier transfer. However, Jin et al. [130] state that He et al.'s scheme is not resistant to various attacks such as tag impersonation,

server spoofing, replay, DoS, etc. On the other hand, in 2015, He and Zeadally [49] present a detailed survey of ECC based RFID authentication protocols up to that date.

In 2016, Farash et al. [124] demonstrate the security and privacy vulnerabilities of [73, 113, 125, 129]'s schemes. In fact, it is shown that none of them provide forward privacy and provable security. Farash et al. also compare their performance and propose an efficient RFID authentication scheme to improve the security and privacy of previous protocols. However, their protocol does not fulfill tag anonymity and location privacy [39]. Jin et al. [131] present an RFID mutual authentication scheme based on ECC to enhance patient privacy while achieving security requirements and overcoming various existing attacks. But, it is shown that their scheme does not provide data integrity and is vulnerable to key compromise problem [45, 127].

In 2017, Chien [120] shows the attacks on [73, 119]'s schemes and proposes a new ECC-based RFID mutual authentication to defeat the security weaknesses. In the same year, Benssalah et al. [39] propose a secure RFID authentication scheme (we call BDD17) based on elliptic curve message recovery (ECMR) signature to provide significant security features and better performance compared to famous authentication protocols based on ECC in the RFID literature. They analyze their design using a formal security analysis with a random oracle model and claim that their protocol is provably secure. Besides, they implement ECMR in FPGA and validate its practical feasibility. However, it is shown in this thesis that BDD17 scheme does not provide forward and backward privacy.

In 2017, Ibrahim and Dalkılıç [40] propose an authentication scheme (we call ID17) for RFID tags based on both symmetric and asymmetric cryptographic algorithms such as ECC and advanced encryption standard (AES). They claimed that their protocol design is secure, private and provides mutual authentication only in two steps. Moreover, they implement their protocol in the wireless identification and sensing platform (WISP5) and present the performance results. However, it is shown again in this thesis that their proposal does not provide forward and backward privacy.

In 2018, Alexander et al. [127] present a survey of the most promising ECC based RFID authentication protocols proposing a different methodology to evaluate recent RFID schemes. They develop a ranking method to compare several RFID protocols in terms of performance and security properties. However, in their evaluation, all ranking points in each category are equal. In other words, different ranks in different categories are weighted the same value. For instance, if a scheme is vulnerable to an attack (i.e. impersonation attack), it loses only a point and is classified into appropriate rank order. We do not agree with their evaluation because we think that firstly security and performance of a scheme should be evaluated separately, and secondly ranking the security properties or performance of a scheme is not the proper approach to compare RFID protocols because it is hard to grade a

certain security property among the others. Besides, Alexander et al. claim that Dinarvand and Barati's [45] scheme (we call DB17) provides all security and privacy requirements. But, we show that their scheme grade security does not provide backward privacy.

Very recently, in 2018, Liu et al. [46] propose a novel ECC based RFID authentication protocol (we call LZKZ18) establishing a key negotiation mechanism. They claim that their protocol design has higher security and privacy. However, it is shown in this thesis that their scheme does not achieve forward and backward privacy.

2.3. RNGs Used In RFID Protocols

The use of RNGs has become the key function in most private and secure light- weight RFID protocols for low cost RFID tags. Low cost RFID tags have approximately 5K-10K gates and only 0.4K-4K gates can be dedicated to security operations [132]. Furthermore, designers are also restricted with the time that is required by a tag while generating a random number because RFID readers should be able to read a bunch of tags in a certain amount of time. Many publications have been presented to design and use RNGs in low cost RFID tags. Melia-Segui et al. have presented a lightweight PRNG design for low-cost passive RFID tags, called J3Gen in 2013 [133]. J3Gen is based on an LFSR (Linear Feedback Shift Register) configured with multiple feedback polynomials that are changed during the generation of sequences by a physical source. They have demonstrated that their most efficient J3Gen design, that has a 32-bit LFSR output with 16-bit feedback polynomials, requires around 1.2K logic gate equivalence (GE). Peinado et al. [35] analyzed J3Gen and they claimed that there are two possible cryptanalytic attacks on J3Gen. In March 2015, Garcia-Alfaro et al. [134] showed that Peinado et al.'s assumptions are incorrect and their attack against J3Gen is not valid. At this point, although Garcia-Alfaro et al. fend off the attack on J3Gen, the literature is still waiting for objections to J3Gen is PRNG.

Peris et al. proposed a PRNG, named LAMED, for low cost RFID tags compliant with the EPC C1G2 standard in 2009 [132]. They claimed that LAMED successfully passes several randomness tests. LAMED requires roughly 1.6K gates and 1.9 ms to generate a 32-bit random number.

Melia-Segui et al. [36] presented a practical attack on a weak PRNG proposed by Che et al. [135] designed for EPC Gen2 tags. Che et al. proposed an LFSR based PRNG with the combination of thermal noise signal modulation. Melia-Segui et al. obtained the feedback polynomial function of the LFSR that they could predict its generated sequences. They showed that an adversary can reach the PRNG configuration with a confidence of 42% by only eavesdropping 128 bits of PRNG data.

In 2008, Garcia et al. [37] have shown that the PRNG used in the MIFARE Classic chip has vulnerabilities.

In 2014, Armknecht et al. [12] have pointed out that ensuring a sufficient level of entropy for RNGs is still a difficult task. They said that different experts from industry who provided the information, all agree stated that generating more than 128 true random bits per authentication on an RFID tag in the price range of $0.05-$0.10 seems currently improbable.

The EPC C1G2 (Class-1 Gen-2) RFID standard was proposed and adopted by EPCglobal in 2004. In 2006, it was published as an amendment to the ISO 18000-6 standard for low-cost lightweight UHF RFID tags. The new version of standards has been recently ratified in 2013 with some optional cryptographic properties [66, 136]. According to the recent standard, a tag generates 16-bit pseudo-random numbers (RN16) using the RNG. The RNG shall meet three randomness criteria: probability of a single RN16, probability of simultaneously identical sequences and probability of predicting an RN16. Although these requirements may be more stringent, a brute-force attack can be applied to reveal the random numbers because lightweight low-cost RFID tags are able to use 32-bit output of PRNG which is a weakness. If an adversary eavesdrops the messages between the reader and the RFID tag, then a brute-force attack or a time-memory trade-off attack can be used to reveal the secrets of a victim tag.

RNGs are implemented by electronic circuits and their randomness quality can be affected by various factors such as seed entropy, aging, environmental effects (such as temperature, humidity, pressure, vibration, electromagnetic field, chemicals, etc.). As a result, biased RNGs cause irretrievable weaknesses.

Bayon et al. [38] demonstrated a practical attack ring oscillator (RO) based TRNG by injecting an EM signal and they also mention previous work about another practical assault to RO based TRNGs by injecting a sine wave signal onto the power pad of the device. Both attacks showed that it is possible to dynamically control the bias of the TRNG output.

In [53], the authors claimed that randomness and size of key generation help to eliminate the advantages of adversaries. Then, they gave an example using Data Encryption Standart (DES) encryption algorithm has 2^{56} key space size. In this case, when a secret key is selected by using a TRNG, an adversary has to try on average 2^{55} possible searches to find the correct key. On the other hand, if the encryption key was selected by using a 16-bit random secret and expanding it into with a 56-bit key by using well-known functions the adversary would need to try on average only 2^{15} possible keys to find the correct one.

In [137], the authors presented a detailed survey paper about random number generators. They compared different types of PRNGs and TRNGs. They also criticized about real randomness, theoretic and practical RNG approaches. They stated that most researchers chose the minimum-action strategy: design a TRNG, obtain at least one random number

sequence that passes a chosen set of randomness tests and publish. However, this does not mean that the corresponding TRNGs have a really good randomness quality because small variations in hardware can weaken them. Hence, a theoretical design cannot proceed towards a product without a detailed investigation of hardware and without extensive randomness proof. Furthermore, Barak, Shaltiel and Tomer [138] proposed an extractor functions to make RNGs robust against aging, temperature changes, etc. Moreover, they presented a couple of weak RNGs caused by hardware imperfections.

3. TOWARDS A MORE MATURE RFID PRIVACY MODEL

In this chapter, the proposed modified version of the well-known Vaudenay's privacy model [27] is introduced before the analysis of privacy aspects of RFID schemes. In the context of our model, the adversary abilities which includes the proposed RANDOMEYE adversary class are presented. Finally, we also apply our new model to two popular existing RFID schemes to provide their security and privacy analysis.

An RFID system is basically composed of three entities: a tag T, a reader R and a back-end system/database DB. A tag T is interrogated by a reader R and the reader identifies/authenticates T by using a unique identifier of the tag ID (in this article it is sometimes denoted as ID_T to improve the readability). DB stores all identifiers and secret keys of valid tags. R communicates with both T and DB and provides a link between them. DB might be considered as a part of the reader. Moreover, T has restricted memory and computational capacities and can communicate with R for a limited distance. We assume that R is much more talented than the tag which is the common case [29]. An adversary Adv can corrupt a tag and use its internal secrets against the system but she cannot corrupt R. We also assume that the communications between R and DB is protected by a secure channel such as Secure Sockets Layer (SSL)/Transport Layer Security (TLS).

3.1. Definitions of RFID Scheme

An RFID system is defined by the following procedures.

– SETUPREADER$(1^\alpha) \longrightarrow (K_P, K_S)$ is a setup algorithm that generates a public-private key pair (K_P, K_S) for the reader R where α is the security parameter, and then initializes an empty database DB to store all identifiers and secret keys of all tags. Although K_s is kept secretly in the DB with the security parameter α; K_p is publicly released.

– SETUPTAG$(K_p, ID) \longrightarrow (K, S)$ is a probabilistic algorithm which returns a tag secret K and the initial state S of a tag T with the input identifier ID. When T is legitimate, the pair (ID, K) is to be stored in the database DB.

– IDENT$\longrightarrow$ *Output* is an interaction protocol between a tag T and the reader R to complete the protocol transcripts. At the end of the protocol, if T is legitimate, R accepts the tag (R identifies T) and outputs its identifier *Output*=ID, otherwise (i.e. if it is not valid) R refuses T and outputs $\perp$.

3.2. Definitions of the Oracles

An adversary *Adv* against an RFID scheme acts as an honest reader and/or an honest tag to attack the system. We assume that there is only one legitimate reader R in the RFID system and both valid readers and tags of the system have no prior information about the entity that is interacting with themselves. We also suppose that each experiment always starts with executing the algorithm SETUPREADER thus, K_p, K_s and 1^α are already generated. We consider that K_p and 1^α are already available to *Adv* but K_s is kept secret because R cannot be corrupted. Furthermore, we assume that there are no tags in the system at the beginning of each experiment and *Adv* is allowed to call $\mathcal{O}^{CreateTag}$ oracle to add new tags to the system.

According to Vaudenay's model [27], a tag is considered as either a free tag or a drawn tag. Drawn tags are the set of tags that *Adv* has visual contact and communicates *Adv* cannot interact with initially free tags. When *Adv* calls the $\mathcal{O}^{CreateTag}$ oracle, she generates a new tag whose status is free. The following oracles are used by the adversary *Adv* to interact with the RFID system. First of all, *Adv* setups a new tag of identifier ID.

– $\mathcal{O}^{CreateTag}(ID, b)$: It creates a free tag T with a unique identifier ID using SETUPTAG. T is legitimate when $b = 1$, otherwise $b = 0$ and T is not valid. It also inserts (ID, K) into DB. b is implicitly 1 when neglected.

Then, the adversary may change the status of the tag from free to drawn by calling the following oracle.

– $\mathcal{O}^{DrawTag}(distr, n) \rightarrow (\psi_{T_1}, b_1, \ldots, \psi_{T_n}, b_n)$: It randomly selects n free tags among all existing ones with distribution probability of the given $distr$. The oracle assigns a new pseudonym, ψ_{T_i} for each tag and changes their status to drawn. Hence, the oracle returns an array of fresh pseudonyms $(\psi_{T_1}, \psi_{T_2}, \ldots, \ldots, \psi_{T_n})$ of the tags (ψ_{T_n} is the pseudonym of the n^{th} tag). The pseudonyms are always changed from session to session so that the adversary may interact to drawn tags for only one single session. The relations (ψ_{T_i}, ID_i) are stored in a hidden table tbl such that $tbl(\psi_{Ti}) = ID_i$. This oracle also returns a bit array $(b_1, b_2 \ldots, \ldots, b_n)$ where b_i of the i^{th} tag shows whether it is legitimate or not. Furthermore, the oracle may return $\perp$ if the querying tags are already drawn or there are no existing tags.

When the tag is drawn, the adversary is only able to interact to the tag with pseudonym ψ_T. ψ_T is defined as a temporary identifier of a tag and used for pointing to the tag anonymously. In this case the following oracles can be called.

– $\mathcal{O}^{Free}(\psi_T)$: This oracle changes the state of tag T that is represented by the pseudonym ψ_T from drawn to free. Afterwards, *Adv* is no longer able to interact with T. The secret

key of the tag with the pseudonym ψ_T is denoted as $key[\psi_T]$. The adversary can corrupt the drawn tags by using the following oracle and obtain the internal values of the tag including its secret key.

- $\mathcal{O}^{Corrupt}(\psi_T) \rightarrow S$: S is the whole memory of ψ_T. Adv obtains the $key[\psi_T]$. Eventually, the tag T with the pseudonym ψ_T is destroyed and Adv cannot interact to T anymore.
- $\mathcal{O}^{Launch}() \rightarrow \pi$: This makes the reader R start a new IDENT protocol with transcript π.
- $\mathcal{O}^{SendReader}(m, \pi) \rightarrow m'$: This sends the message m to the reader R in the protocol transcript π with outputs the response m'.
- $\mathcal{O}^{SendTag}(m, \pi) \rightarrow m'$: This sends the message m to T and outputs the response m'. Also, Adv asks for the reader's result of the protocol transcript π. The adversary can use the corresponding oracle to change the state of the tag so she can start to interact with the tag change, the state to drawn or she can free the tag (after which she communicate). anymore.
- $\mathcal{O}^{Execute}(\psi_T) \rightarrow (\pi, transcript)$: This executes a complete protocol between the reader and the tag with pseudonym ψ_T. It returns the transcript of the protocol instance that is the list of all successive messages of the protocol.
- $\mathcal{O}^{Result}(\pi) \rightarrow x$: This returns $x = 1$ when π completes successfully after the IDENT returns $Output \neq \perp$ which means that the tag T is identified. Otherwise, if T is not identified and $Output = \perp$, this oracle returns $x = 0$.

Finally, we introduce a new oracle called RNG oracle, $\mathcal{O}^{RNG}$ as follows. The adversary Adv is allowed to obtain the results of the RNG bit string used in the protocol by a tag T by querying the following oracle. For simple explanation, π_i denotes the i^{th} protocol instance, s_i is the corruption state of the RNG of a tag T for the i^{th} protocol instance. If $s_i = 0$, Adv does not corrupt T but if $s_i = 1$, she corrupts T and captures the $key[\psi_T]$ for the protocol instance π_i. The array of (π_i, s_i) values is denoted by $\theta_\pi := \{(s_1, \pi_1), (s_2, \pi_2), \ldots, (s_n, \pi_n)\}$ and θ_π defines the sufficient number of n tuples where each tuple includes the protocol transcript and tag corruption information.

- $\mathcal{O}^{RNG}(\theta_\pi, \psi_T) \rightarrow (RNG_1, RNG_2 \ldots, RNG_i, \ldots, RNG_n)$: This outputs the set of the RNG bit string used on the tag T with the unique identifier ID_T for each protocol instance π_i and the state s_i. The oracle returns $\perp$ for any protocol instance π_i, when the RNG used in this instance cannot be obtained.

Adv performs her attack by running an experiment or playing a game and obeying the corresponding rules. Firstly, she constructs an RFID system and uses the oracles and gets a result. She wins or looses depending on the corresponding rules.

3.3. Definition of the Adversary Classes

We define different adversary classes for playing security games. The definition includes Vaudenay's model [27] and our own novel adversary class.

Definition 13 (Adversary Classes). An adversary *Adv* against an RFID system who has an arbitrary number of accesses to the above oracles except the $\mathcal{O}^{RNG}$ oracle is regarded to be in one of the following classes.

– STRONG *Adv* uses all oracles without any restrictions.

– DESTRUCTIVE *Adv* cannot use an oracle against a tag after using $\mathcal{O}^{Corrupt}$ oracle (i.e. the tag has been killed).

– FORWARD *Adv* can only use $\mathcal{O}^{Corrupt}$ oracle after her first call to this oracle.

– WEAK *Adv* uses all oracles except $\mathcal{O}^{Corrupt}$ oracle

– NARROW *Adv* has no access to $\mathcal{O}^{Result}$ oracle.

– RANDOMEYE *Adv* can access the RNG oracle $\mathcal{O}^{RNG}$, and extracts the random number(s) used in a tag. This is a novel class introduced in this thesis.

3.4. Security in Formal Analysis

Some security properties of an RFID system such as completeness and soundness are visited below.

Definition 14 (Completeness). An RFID system is complete if the reader R of the system returns the tag identifier ID at the end of the protocol (IDENT) for a legitimate tag T with very high probability.

Definition 15 (Strong Completeness). An RFID system is complete if the reader R of the system returns the tag identifier ID at the end of the protocol (IDENT) for a legitimate tag T with very high probability although the RFID scheme has been already attacked.

According to Vaudenay's model, security is a vital property and should be withheld against every attack by the strongest adversary. But it is obvious that the security of a scheme is violated when tag impersonation occurred if the adversary uses $\mathcal{O}^{Corrupt}$ oracle. Hence, the model permits an adversary to use all oracles except the $\mathcal{O}^{Corrupt}$ oracle.

Definition 16 (Soundness). An RFID system is said sound if an adversary *Adv* impersonates a legitimate tag T with negligible probability [29].

3.5. Privacy in Formal Analysis

Vaudenay defines a privacy notion that is the deducing ability of an adversary to obtain the *ID* relations of a tag from its protocol instances. He explains *anonymity* and *untraceability* properties under the privacy notion in that one is about unveiling the *ID* of tags and the other is about indistinguishability of any two tags, respectively [27].

In the RFID literature, there are two types of untraceability notions: "forward untraceability" and "backward untraceability". If an RFID system provides the forward untraceability feature, an adversary *Adv* who compromises a legitimate tag at a time t, cannot trace the future interactions of the tag, $t' > t$. If an RFID system provides the backward untraceability feature, *Adv* cannot trace past interactions of the tag, $t' < t$. The backward untraceability property is also referred to as forward privacy or forward secrecy and this notion is more important than forward untraceability for real-life scenarios.

Vaudenay also considers the privacy of the RFID system based on the adversary classes in Definition 13. In his model, he presents a blinded adversary called blinder B.

Definition 17 (Blinder, trivial adversary). A blinder B for an adversary *Adv* is a polynomial time algorithm that observes the same messages as *Adv* and simulates LAUNCH, SENDTAG, SENDREADER, and RESULT oracles without having access to the secret keys nor the database of the system. The adversary *Adv* uses all outputs of the oracles. A blinded adversary Adv^B is an adversary who never uses LAUNCH, SENDREADER, SENDTAG, and RESULT oracles. An adversary *Adv* is said to be *trivial* if there exists a blinded adversary Adv^B such that $\mid Prob[Adv\ wins] - Prob[Adv^B\ wins] \mid$ is negligible.

If the success probability of the simulator and the blind adversary is nearly the same, this means that the blind adversary has attack ability at least as high as the simulator of the system (except using the secret keys). Hence, the authentication and identification of a tag can be considered private. Vaudenay says that an adversary accomplishes his attack (plays a security game) into two phases. In the first phase, she queries the allowed oracles and collects the outputs. In the second phase, she analyses the obtained results without using any oracle. Between the two phases, she also has access to the hidden table tbl of the $\mathcal{O}^{DrawTag}$ oracle. If she outputs true from her analysis, then she wins the game.

Definition 18 (Privacy). An RFID system is P-private if all the adversaries who belong to class P are trivial following Definition 17 [27].

The following well-known links (see Figure 3.1) between Vaudenay's privacy classes which are rather obvious by definition.

Vaudenay also defines an untraceability property related to the notion of privacy. He says that only a STRONG adversary can break the forward untraceability of an RFID scheme

since she can call other oracles after corrupting the tag. Vaudenay also shows in his thesis that the ultimate privacy level for RFID systems can be ensured by using PKC [27].

STRONG	$\Rightarrow$	DESTRUCTIVE	$\Rightarrow$	FORWARD	$\Rightarrow$	WEAK
$\Downarrow$		$\Downarrow$		$\Downarrow$		$\Downarrow$
NARROW-STRONG	$\Rightarrow$	NARROW-DESTRUCTIVE	$\Rightarrow$	NARROW-FORWARD	$\Rightarrow$	NARROW-WEAK

Figure 3.1. The relationship between privacy classes

3.6. The Proposed RANDOMEYE Adversary Class

Now we are ready to explain our RANDOMEYE adversary class and its relationship to the other adversary classes. The RANDOMEYE adversary class formalizes the weakness and/or misuse of random number generators for real-life RFID systems. Tangibly, an adversary *Adv* that can query the $\mathcal{O}^{RNG}$ oracle, might learn the random numbers used in the authentication protocol. If *Adv* cannot infer the *ID* of the tag by using this information, we consider that the protocol is RANDOMEYE private. Hence Vaudenay's original adversary classes are not complete and the relationship between them has changed with the newly introduced class. Therefore, we give the new link (depicted in Figure 3.2) for the STRONG class for clear comprehensibility:

RANDOMEYE-STRONG	$\Longrightarrow$	STRONG
$\Downarrow$		$\Downarrow$
NARROW-RANDOMEYE-STRONG	$\Longrightarrow$	NARROW-STRONG

Figure 3.2. The relationship of RANDOMEYE with STRONG

3.7. Case Studies

In this section, we consider two popular existing RFID schemes to apply our new model and provide analysis. We first briefly introduce Song and Mitchell's and Akgün et al.'s schemes. Then, we explain how an adversary attacks and break the schemes step by step. Our analysis further shows that the schemes do not provide security and privacy properties with respect to the presented weakness. Hence, according to our improved model, the protocols are not RANDOMEYE private.

3.7.1. First study example: Song and Mitchell's protocol

Firstly, we investigate the scheme designed by Song and Mitchell (SM) [139] to provide private and secure authentication between low cost RFID tags. Their protocol is depicted below.

In this protocol, the reader generates a nonce r_1 and sends it to the tag to start the protocol. The tag receives the nonce and generates a random bit string, r_2 as a temporary secret for the protocol instance. The tag computes $M_1 = r_1 \oplus tid_i$ and $M_2 = f_{tid_i}(r_1 \oplus r_2)$. Then, the tag sends M_1 and M_2 to the reader. The reader evaluates and searches its database by using

M_1, M_2, and r_1. If the reader does not find any match, it will stop the session. In case of a successful match, the reader authenticates the tag and updates the tag information which is $(u_i)_{old}$ and $(tid_i)_{old}$. Then it computes $M_3 = u_i \oplus (r_2 \gg {}^{l}/_{2})$ and sends M_3 message to the tag. The tag computes u_i using M_3 and checks that $h(u_i) = t_i$. If a match is obtained, the tag authenticates the reader and updates its u_i and t_i values. Otherwise, the tag does not update the current values. This process is shown in Figure 3.3.

We prove below that a RANDOMEYE adversary can trace a tag in this protocol without corrupting it.

Theorem 3.7.1 The SM protocol does not ensure the RANDOMEYE-WEAK privacy.

Proof 3.7.1 An adversary *Adv* can perform the following attack.

1. *Adv* creates two legitimate tags by using $\mathcal{O}^{CreateTag}(tid_1, 1)$ and $\mathcal{O}^{CreateTag}(tid_2, 1)$ oracles. Then, *Adv* draws two tags from the system by calling $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and obtains two pseudonyms T_1 and T_2. At this point, *Adv* does not know tid_1 and tid_2 that are the identifiers of the T_1 and T_2 tags respectively.

2. *Adv* calls $\mathcal{O}^{Execute}(T_1)$ and gets $\theta_\pi = (0, \pi_1)$ for T_1.

3. Then, *Adv* requests $\mathcal{O}^{RNG}[\theta_\pi, \boldsymbol{T}_1]$ and obtains $(RNG_1, 1)$ for T_1. For this protocol RNG_1 is equal to the random bit strings r_2 generated by the tag, T_1. $\mathcal{O}^{RNG}$ oracle performs the following procedures:

3.1 It generates all possible random strings for r_2 with respect to the seed of the RNG used in the tag. Lets call the list $R = \left[r_2^1, r_2^2, ..., r_2^j, ...r_2^{|K|}\right]$ where $|K|$ is the entropy of the seed.

3.2 It has the list of all the possible $X = \left[tid_1^1, tid_1^2, ..., tid_1^j, ...tid_1^{|K|}\right]$ values by computing $X = M_1 \oplus R$ because M_1 is obtained within the protocol instance.

3.3 Then, it does the exhaustive search to check for the M_2 messages with computing $f_X(r_1 \oplus R)$. Finding $M_2 = f_{M_1 \oplus r_2^j}\left(r_1 \oplus r_2^j\right)$, *Adv* obtains r_2 that is equal to r_2^j.

4. *Adv* obtains the tid_1 for tag T_1 computing $M_1 \oplus r_2$ and updates the internal values of the tag according to the protocol procedure. Therefore, *Adv* has the $tid_{1(new)}$ value of T_1.

5. *Adv* performs step 2, step 3 and step 4 for the T_2 tag. *Adv* updates the internal values of the tag and gets the $tid_{2(new)}$ value of T_2.

6. *Adv* frees both tags with request $\mathcal{O}^{Free}(\boldsymbol{T}_1)$ and $\mathcal{O}^{Free}(T_2)$, then she reaffects only one of them using $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$. She obtains a new T_3.

7. *Adv* performs step 2, step 3 and step 4 for the T_3 tag and obtains tid_3.

8. Then *Adv* compares tid_3 with $tid_{1(new)}$ and $tid_{2(new)}$.

9. If $tid_3 = tid_{1(new)}$, *Adv* claims that $T_3 = T_1$ else she claims that $T_3 = T_2$.

The success probability of this adversary is equal to 1. Therefore, it is clear that Song and Mitchell's Protocol is not RANDOMEYE-WEAK private.

Reader $[(u_i, tid_i)_{new}, (u_i, tid_i)_{old}, D_i]$		***Tag*** T_i $[tid_i]$
$r_1 \in_R \{0,1\}^l$	$\xrightarrow{r_1}$	$r_2 \in_R \{0,1\}^l$ $M_1 = tid_i \oplus r_2$ $M_2 = f_{tid_i}(r_1 \oplus r_2)$
	$\xleftarrow{M_1, M_2}$	
Search for a value tid_i for which $r_2 \leftarrow M_1 \oplus tid_i$ and $M_2 = f_{tid_i}(r_1 \oplus r_2)$ $M_3 = u_i \oplus (r_2 \gg l/2)$		
	$\xrightarrow{M_3}$	
$u_{i(old)} \leftarrow u_i$ $tid_{i(old)} \leftarrow tid_i$ $u_{i(new)} \leftarrow (u_i \ll l/4) \oplus (t_i \ll l/4) \oplus r_1 \oplus r_2$ $tid_{i(new)} \leftarrow h(u_{i(new)})$		$u_i \leftarrow M_3 \oplus (r_2 \gg l/2)$ $h(u_i) \stackrel{?}{=} tid_i$ $tid_i \leftarrow h((u_i \ll l/4) \oplus (t_i \ll l/4) \oplus r_1 \oplus r_2)$

Figure 3.3. Song and Mitchell's protocol [139]

3.7.2. Second study example: Akgün et al.'s protocol

Akgün et al. [44] introduced a new authentication protocol and claimed that it is the first protocol that provides destructive privacy according to Vaudenay's model with constant identification time. This scheme is a simple challenge/response protocol enhanced with Physically Unclonable Functions (PUFs) in order to achieve a higher level of privacy. This scheme is shown in Figure 3.4.

This scheme has two phases. In the first phase, the system initializes itself. In this initialization phase, a shared secret S is randomly generated for the back-end server. Two random values, a and b are generated for each tag. Then each tag performs its own PUF $P(.)$ to calculate $c = S \oplus P(a) \oplus P(b)$. The back-end server stores all values $[ID_i, a_i, b_i, DATA_i]$ for each tag where $DATA_i$ contains the information about a tag T_i.

In the second phase called the authentication phase, the reader generates a random number r_1 and broadcasts it to the tag.

Secondly, a tag T_i which receives the signal of the reader generates another random number r_2. The tag also computes $M_1 = H(r_1, r_2, a_i)$, $M_1 = H(r_2, r_1, 1) \oplus ID_i$ and $h = H(r_2, 1, 2)$. Then, it uses PUF to calculate $k = P_i(a_i) \oplus r_2$ and deletes the r_2 and $P_i(a_i)$ values from

the volatile memory. The tag updates k by computing $k = k \oplus P_i(b_i) \oplus c_i$ and then $P_i(b_i)$ is deleted from the memory too. The tag transmits M_1, M_2 and k back to the reader.

Thirdly, the reader generates a new random number r_3 and computes $r_2' = S \oplus k$, $ID_i' = M_2 \oplus H(r_2', r_1, 1)$. Then, the reader checks that the M_1 message is equal to $H(r_1, r_2', a_i)$ to authenticate the tag T_i. If the equality is confirmed, then the reader computes $M_3 = H(H(r_2', 1, 2), r_3, b_i)$ and sends r_3 and M_3 to the tag T_i.

Finally, the tag T_i checks that the M_3 message is equal to $H(h, r_3, b_i)$ to authenticate the reader. If equality is confirmed, the tag authenticates the reader too. Thus, mutual authentication is accomplished and the protocol is terminated successfully.

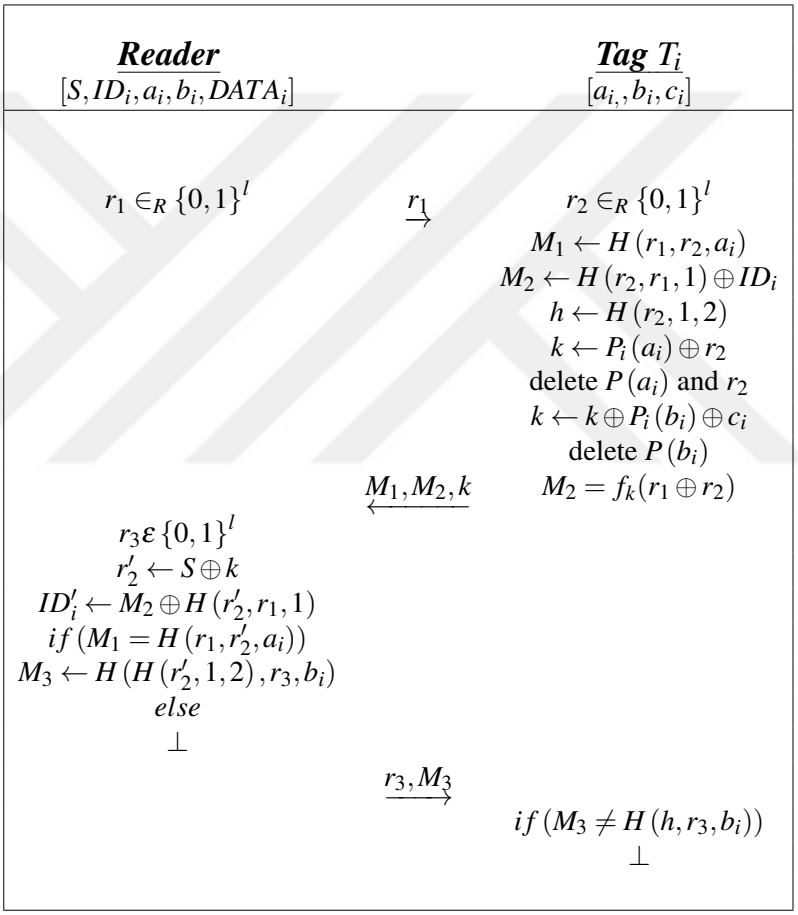

Figure 3.4. Akgün et al.'s authentication protocol [44]

Akgün et al. claimed that their protocol scheme provides destructive privacy according to Vaudenays privacy and security model with constant time identification property. Their protocol does not need key-updating mechanism on both, tags and back-end server. The authors use the common secret S to identify a tag with $O(1)$ time complexity. They base the security and privacy of their protocol on the PUFs that are regarded to have robustness, unclonability, unpredictability and tamper-evident properties [44]. We realized that there is a RNG misuse in their protocol design. We can prove that their protocol is neither destructive private nor secure. A RANDOMEYE adversary can trace the past and future

transactions of the tag as proven below.

Theorem 3.7.2 Akgün et al.'s protocol does not ensure the RANDOMEYE-WEAK privacy.

Proof 3.7.2 An adversary *Adv* can perform the following attack.

1. *Adv* creates two legitimate tags by using $\mathcal{O}^{CreateTag}(ID_1,1)$ and $\mathcal{O}^{CreateTag}(ID_2,1)$ oracles. Then, *Adv* draws two tags from the system by calling $\mathcal{O}^{DrawTag}\left(\frac{1}{2},2\right)$ oracle and obtains two pseudonyms T_1 and T_2. At this point, *Adv* does not know ID_1 and ID_2 that are the identifiers of the T_1 and T_2 tags respectively.

2. *Adv* calls $\mathcal{O}^{Execute}(T_1)$ two times and gets $\theta_\pi = \{(0,\pi_1),(0,\pi_2)\}$ for T_1.

3. Then, *Adv* requests $\mathcal{O}^{RNG}[\theta_\pi, \boldsymbol{T}_1]$. *Adv* obtains (RNG_1) and (RNG_2) respectively for T_1. For this protocol scheme, RNG_1 is equal to the random bit strings r_2 generated by the tag, T_1 for the first protocol instance and RNG_2 is the secondly generated random bit string r_2. $\mathcal{O}^{RNG}$ oracle performs the following procedures:

3.1 It generates all possible random strings for r_2 with respect to the seed of the RNG used in the tag. Let's call the list $R = \left[r_2^1, r_2^2, ..., r_2^j, ...r_2^{|K|}\right]$ where $|K|$ is the entropy of the seed.

3.2 It has the list of all the possible $X^1 = \left[ID_1^1, ID_1^2, ..., ID_1^j, ..., ID_1^{|K|}\right]$ values by computing $X^1 = M_2 \oplus H(R, r_1, 1)$ because M_2 and r_1 are obtained within the first protocol instance.

3.3 It has the second list of all the possible $X^2 = \left[ID_1^1, ID_1^2, ..., ID_1^j, ..., ID_1^{|K|}\right]$ values by computing $X^2 = M_2 \oplus H(R, r_1, 1)$ because M_2 and r_1 are obtained within the second protocol instance.

3.4 Then, it compares X^1 and X^2 and defines the identifier of the tag by finding the equal bit string of each list.

3.5 Finally, it obtains the random bit string r_2 by using the corresponding identifier of the tag ID_1 .

4. *Adv* obtains ID_1 for T_1 tag by computing $M_2 \oplus H(r_2, r_1, 1)$ using one of the protocol instances.

5. *Adv* performs step 2, step 3 and step 4 for the T_2 tag. *Adv* obtains ID_2 for T_2 .

6. *Adv* frees both tags with request $\mathcal{O}^{Free}(\boldsymbol{T}_1)$ and $\mathcal{O}^{Free}(T_2)$, then she re-affects only one of them using $\mathcal{O}^{DrawTag}\left(\frac{1}{2},1\right)$. She obtains a new T_3.

7. *Adv* performs step 2, step 3 and step 4 for the T_3 tag and obtains ID_3.

8. Then *Adv* compares ID_3 with ID_1 and ID_2.

9. If $ID_3 = ID_1$, *Adv* claims that $T_3 = T_1$ else she claims that $T_3 = T_2$.

Therefore, if the adversary *Adv* captures the *IDs*, she can trace the past and future transactions of the tags of the scheme using the unchanging *ID*. Hence, the scheme does not provide forward and backward untraceability properties.

Theorem 3.7.3 Akgün et al.'s protocol does not ensure the RANDOMEYE-DES TRUCTIVE privacy.

Proof 3.7.3 Akgün et al.'s protocol does not provide WEAK privacy. Hence, it is not DESTRUCTIVE private.

Theorem 3.7.4 Akgün et al.'s scheme is not secure against RANDOMEYE adversary.

Proof 3.7.4 It is clearly seen that the Akgün et al.'s scheme does not provide RANDOM-WEAK privacy and a passive adversary is able to reveal the *ID* of a tag. Let an adversary *Adv* reveals the *ID* of a tag and consequently has the random bit strings r_2. *Adv* also has the k value obtained during eavesdropping to the protocol session where $k = P_i(a_i) \oplus r_2 \oplus P_i(b_i) \oplus c_i$. The shared secret S is generated as $S = P_i(a_i) \oplus P_i(b_i) \oplus c_i$ in the initialization according to the protocol description. Thus, the adversary *Adv* obtains the shared secret S by computing $S = k \oplus r_2$. The scheme is no longer secure after the shared secret S is obtained and the whole system can be broken by the adversary *Adv*.

4. THE PROPOSED ECC BASED RFID AUTHENTICATION PROTOCOL

In this chapter, we first describe, analyze the recent ECC based RFID authentication protocols, and show their privacy vulnerabilities. Secondly, we present our proposed protocol and its security-privacy analysis. Then, we introduce our simulation and implementation environment where are theoretical and practical tests executed. Finally, we share our outcomes and acquaint a comprehensive comparison.

4.1. Analysis of Previous Authentication Schemes

In this section, we first briefly introduce four recent and relatively popular RFID protocols, namely ID17 [40], BDD17 [39], DB17 [45] and LZKZ18 [46]. Then, we present that these schemes do not ensure forward and/or backward privacy as they claimed.

4.1.1. Analysis of ID17 RFID authentication scheme

In this subsection, we first briefly describe ID17 [40] and then show our proposed attacks.

Protocol Description: ID17 scheme (illustrated in Figure 4.1) includes two phases: a setup phase and an authentication phase. In the setup phase, both reader and tag agree on elliptic curve domain (EC) parameters a, b, q, P, n and h, where P is a base point. Then, the reader randomly generates a private key $k_r^{'}$ and computes the public pair $k_R^{'} = k_r^{'}P$. Similarly, the tag randomly generates a private key $k_t^{'}$ and computes the public pair $k_T^{'} = k_t^{'}P$. Finally, both reader and tag share their public keys with each other.

In the authentication phase, the reader first randomly generates an ephemeral private key k_r and calculates its own ephemeral public key, where $k_R = k_r P$. Then, the reader signs k_R with its private key $k_r^{'}$ using ECDSA. The signature of k_R is (x, y), where $(x, y) = ECDSA_{k_r^{'}}(k_R)$. After signing, the reader sends k_R and its signature (x, y) to the tag.

When the tag receives the messages of the reader (k_R, x, y), the tag checks that (x, y) are integers in the range $[1, n-1]$. If not, the tag terminates the session. Else, it continues the verification process and verifies k_R using the public key of the reader $k_R^{'}$. If the verification is succeeded, the tag authenticates the reader. Otherwise, it rejects the session. In case of authentication, the tag also randomly picks k_t as an ephemeral private key and computes its own ephemeral public key, $k_{TT} = k_t P$. Then, the tag signs k_{TT} and gets the signature pair (w, v) using ECDSA, where $[w, v] = ECDSA_{k_t^{'}}(k_{TT})$. After signing, the tag calculates $K_{TR} = k_t k_R$ as an ephemeral shared secret key. Later on, the tag encrypts its ID with K_{TR} and obtains message C, where $C = AES_{K_{TR}}(ID)$. The tag sends k_{TT}, w, v, C to the reader.

Upon receiving the tag's response, the reader also checks that w, v are integers in the range $[1, n-1]$. If not, the reader drops the session. Else, the reader continues the verification process and verifies k_{TT}. If the verification is succeeded, the reader also computes the ephemeral shared secret key, where $K_{TR} = k_r k_{TT}$. The reader decrypts C using K_{TR} and obtains the ID of the tag. If the reader finds that the ID belongs to the tag registered in the database, the tag is authenticated, too.

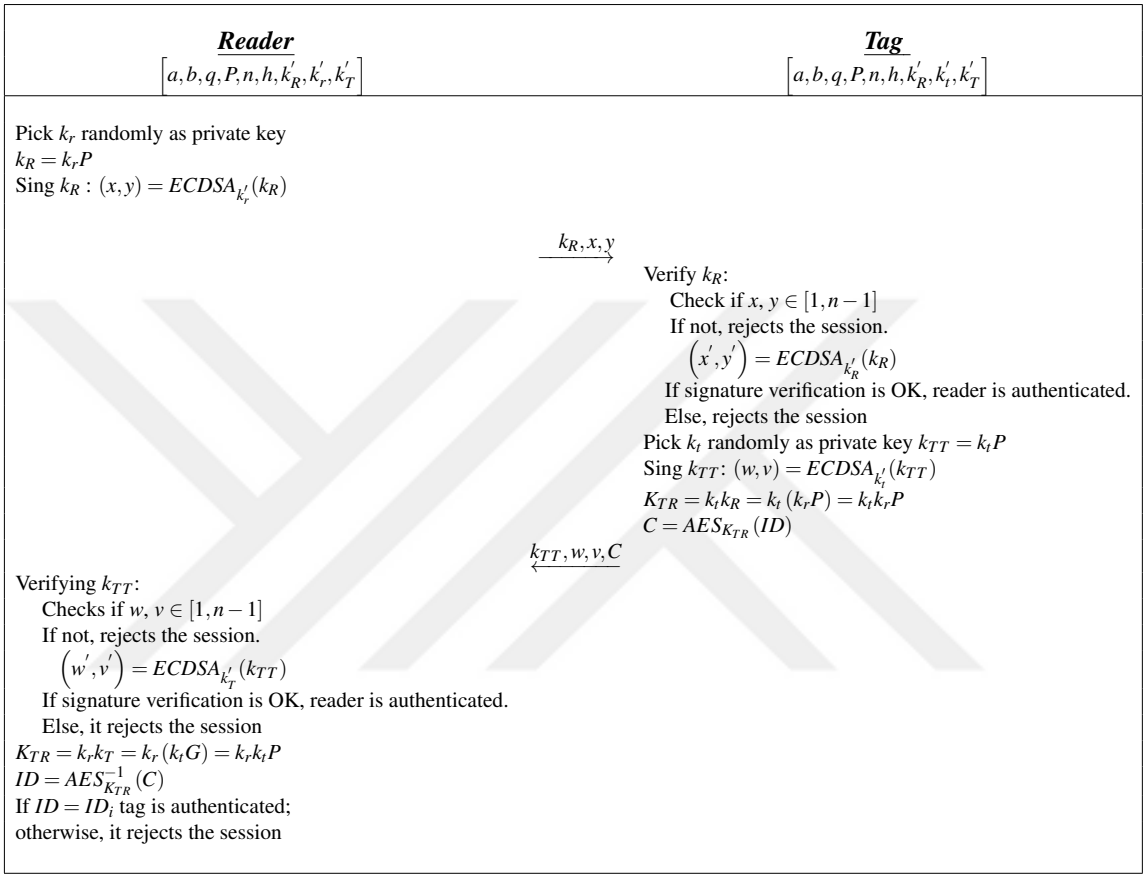

Figure 4.1. ID17 RFID authentication scheme [40]

Proposed Attacks on the Protocol: The authors claim that their protocol (ID17) provides forward and backward security but we prove that when an adversary corrupts a tag, she can distinguish the tag among the others using its past and future transactions [40]. The authors, in their analysis, state that an adversary cannot perform these attacks because all the transmitted messages are updated for each protocol session. We show that their design does not fulfill randomization in each session to prevent the untraceability since the adversary can verify every signature of the tag if she obtains the private key of the tag once. Therefore, the adversary can violate the backward and forward privacy. Formally, the adversary plays the following games to show how to break the forward and backward privacy properties.

Theorem 4.1.1 ID17 scheme does not provide backward privacy.

Proof 4.1.1 Let *Adv* be a STRONG adversary that plays a security game as below.

1. *Adv* calls $\mathcal{O}^{CreateTag}(ID_0,1)$ and $\mathcal{O}^{CreateTag}(ID_1,1)$ to create two valid tags T_0 and T_1, respectively.

2. *Adv* randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2},1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0,1\}$.

3. *Adv* chooses a time interval I_0. During I_0, she calls $\mathcal{O}^{Corrupt}(\psi_{T_i})$ and obtains the internal values of the tag with pseudonym ψ_{T_i}. These are $a,b,q,P,n,h,k'_{TT_i},k_{t'_i}$, ID_i, $k_{R'_i}$.

4. *Adv* frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.

5. *Adv* chooses another time interval I_1, where $I_1 > I_0$. During I_1, *Adv* calls $\mathcal{O}^{DrawTag}\left(\frac{1}{2},2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.

6. *Adv* arbitrarily selects one of the drawn tags $(e.g.\ \psi_{T_1})$ and calls $\mathcal{O}^{Execute}(\psi_{T_1})$ oracle. She gets $k_{R_1}^{I_1},x_1^{I_1},y_1^{I_1},k_{TT_1}^{I_1}$, $w_1^{I_1},v_1^{I_1},C_1^{I_1}$ as a protocol transcript.

7. *Adv* frees the tags by calling $\mathcal{O}^{Free}(\psi_{T_0})$ and $\mathcal{O}^{Free}(\psi_{T_1})$ oracle.

8. Then, $\boldsymbol{A}$ tries to verify the signature $\left(w_1^{I_1},v_1^{I_1}\right)$ of the ephemeral public key $k_{TT_1}^{I_1}$ of the tag with the pseudonym ψ_{T_1} by using the corrupted static key k'_{TT_i} of the tag.

9. If the signature is valid, she claims that $i=1$ $(i.e.\psi_{T_i}=\psi_{T_1})$. Otherwise, she claims that $i=0$ $(i.e.\psi_{T_i}=\psi_{T_0})$.

Obviously, the success probability of this adversary is 1 and she wins the game. This means that *Adv* can distinguish the future transactions of the tag. Therefore, this scheme does not provide backward privacy.

Theorem 4.1.2 ID17 scheme does not provide forward privacy.

Proof 4.1.2 Let *Adv* be a STRONG adversary that plays a security game as below.

1. *Adv* calls $\mathcal{O}^{CreateTag}(ID_0,1)$ and $\mathcal{O}^{CreateTag}(ID_1,1)$ to create two valid tags T_0 and T_1, respectively.

2. *Adv* randomly picks two tags by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2},2\right)$ oracle and gets two pseudonyms ψ_{T_0} and ψ_{T_1}.

3. *Adv* chooses a time interval I_0. During I_0, she arbitrarily selects one of the drawn tags $(e.g.\ \psi_{T_1})$ and calls $\mathcal{O}^{Execute}(\psi_{T_1})$ oracle. Then, *Adv* gets $k_{R_1}^{I_0},z_1^{I_0}$, $s_1^{I_0},k_{TT_1}^{I_0},g_1^{I_0}$, $h_1^{I_0},C_1^{I_0}$ as a protocol transcript for ψ_{T_1}.

4. *Adv* frees the tags by calling $\mathcal{O}^{Free}(\psi_{T_0})$ and $\mathcal{O}^{Free}(\psi_{T_1})$ oracle.

5. *Adv* chooses another time interval I_1, where $I_1 > I_0$. During I_1, *Adv* randomly chooses a tag T_i by calling $\mathcal{O}^{DrawTag}\left(\frac{1}{2},1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0,1\}$.

6. *Adv* calls $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle and gets $a, b, q, P, n, h, k'_{TT_i}, k_{t'_i}, ID_i$ and $k_{R'_i}$

7. *Adv* frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.

8. Then, *Adv* tries to verify the signature $\left(w_1^{I_0}, v_1^{I_0}\right)$ of the ephemeral public key $k_{TT_1}^{I_0}$ of the tag with the pseudonym ψ_{T_1} by using the corrupted static key k'_{TT_i} of the tag.

9. If the signature is valid, she claims that $i = 1$ $(i.e. \psi_{T_i} = \psi_{T_1})$. Otherwise, she claims that $i = 0$ $(i.e. \psi_{T_i} = \psi_{T_0})$.

The success probability of this adversary is 1 and she wins the game. This means that *Adv* has stored some past transcripts. Then, when she obtains the internal values of the tag, thereby she can verify the signature of the ephemeral public key and identify the tag using a previous transcript. Therefore, this scheme does not provide forward privacy.

4.1.2. Analysis of BDD17 RFID authentication scheme

Tn this subsection, we first briefly describe the BDD17 [39] scheme and then show our proposed attacks.

Protocol Description: BDD17 scheme shown in Figure 4.2 has three phases: setup phase, authentication phase, and update phase. In the setup phase, a trusted issuer generates the system parameters $< Z_{BS_j}, ID_{T_i}, x_{T_i}, SID_j, P_s, m >$, $< Z_{BS_j}, x_{R_i}, SID_j, P_s, V_k, W_k >$ and $< Z_{T_i}, Z_{R'_i}, ID_{T_i}, RID_{i'}, SID_j, x_{BS_j}, x_{R_{i'}}, P_s, m, IDs_{T_i}^{old}, IDs_{T_i}^{new} >$ to be stored by all involved entities (tags, readers and the back-end server, respectively).

In the mutual authentication and an updating phase, reader $(R_{i'})$ controls the user's password and checks whether $V'_k = V_k$. If it is held, then $R_{i'}$ generates a random number r_R and broadcast the request $(r_R, auth)$ to tag T_i. When the tag receives the request, it signs r_R with a pre-shared message m and a random scalar k using elliptic curve message recovering signature algorithm (ECMR). Then, the tag responds with an anonymous identity $ID_{s_{T_i}}$ and an ECMR signature (r, s). Upon receiving this response, the reader gets the current timestamp T_{r_1} and computes the message $V = h\left(x_{R_{i'}} \| r_R \| T_{r_1}\right)$. Then, the reader sends $r, s, r_R, V, T_{r_1}, ID_{s_{T_i}}$ to BS_j. BS_j firstly checks the validity of the timestamp and authenticates the reader checking the value V. BS_j finds the related tag's parameter using $ID_{s_{T_i}}$ in $O(1)$ time. Then, BS_j recovers message m' and verifies its validity by calculating $\gamma = h(r \| r_R)\left(Z_{T_i} + \left((Z_{T_i})_x + ID_{T_i}\right) P_s\right)$ and $\left(r_R \oplus m'\right) = r - \left((sG + \gamma) x_{BS_j}\right)_x mod(n)$.

If the signature is not correct, it rejects T_i. Otherwise, BS_j server gets the current timestamp T_{s_2} and performs the following calculations: $\beta = Z_{R'_i} + \left(\left(Z_{R'_i}\right)_x + RID_{i'}\right) P_s$, $r'_i = Data_i + h\left(r'_{i-1} \oplus (l(\beta))_x\right) mod(n)$, $R = h\left(r'_1 \| r'_2 \| \cdots \| r'_n \| T_{s_2}\right)$, $s' = l - Rx_{BS_j} mod(n)$ and $C = h\left(s \| SID_j \| m \| r_R\right)$.

Back-end Server (BS_j)		**Reader** $(R_{i'})$		**Tag** (T_i)
$< Z_{T_i}, Z_{R_i'}, ID_{T_i}, RID_{i'}, SID_j >$		$< Z_{BS_j}, x_{R_{i'}}, SID_j >$		$< Z_{BS_j}, ID_{T_i}, x_{T_i} >$
$< x_{BS_j}, x_{R_{i'}}, P_s, m, IDs_{T_i}^{old}, IDs_{T_i}^{new} >$		$< P_s, V_k, W_k >$		$< SID_j, P_s, m >$
		Checks whether $V_k^{'} = V_k$ $r_R \in [1, n-1]$	$\xrightarrow{Send\,(r_R, auth)}$	Generates : $k \in [1, n-1]$ $\psi = Z_{BS_j} + \left(\left(Z_{BS_j} \right)_x + SID_j \right) P_s$ $r \leftarrow (r_R \oplus m) + (k\psi)_x \, mod\,(n)$ $s = k - h(r \| r_R)\, x_{T_i} mod\,(n)$
		Gets the current timestamp T_{r_1} $V = h\left(x_{R_{i'}} \| r_R \| T_{r_1} \right)$	$\xleftarrow{r, s, ID_{s_{T_i}}}$	
Gets the current timestampT_{s_1} Verifies if $\mid T_{s_1} - T_{r_1} \mid < \triangle T$ Checks whether $h\left(x_{R_{i'}} \| r_R \| T_r \right) \stackrel{?}{=} V$ Searches IDs_{T_i} and recover $m^{'}$ $\gamma = h(r \| r_R)\left(Z_{T_i} + \left((Z_{T_i})_x + ID_{T_i} \right) P_s \right)$ $\left(r_R \oplus m^{'} \right) = r - \left((sG + \gamma) x_{BS_j} \right)_x mod\,(n)$ if $m = m^{'}$, then T_i is authenticated; Else, T_i is rejected. Picks a random number l, genaretes T_{s_2} and computes $r_0^{'} = 0$ $\beta = Z_{R_i'} + \left(\left(Z_{R_i'} \right)_x + RID_{i'} \right) P_s$ $r_i^{'} = Data_i + h\left(r_{i-1}^{'} \oplus (l(\beta))_x \right) mod\,(n)$ $R = h\left(r_1^{'} \| r_2^{'} \| \cdots \| r_n^{'} \| T_{s_2} \right)$ $s^{'} = l - R x_{BS_j} mod\,(n)$ $C = h\left(s \| SID_j \| m \| r_R \right)$	$\xleftarrow{r, s, r_R, V, T_{r_1}, ID_{s_{T_i}}}$			
$IDs_{T_i}^{old} \leftarrow IDs_{T_i}^{new}$ $IDs_{T_i}^{new} \leftarrow h(IDs_{T_i} \| m \| r_R \| x_{T_i})$	$\xrightarrow{T_{s_2}, C, R, r_1^{'}, r_2^{'}, \cdots, r_n^{'}}$	Gets the current timestampT_{r_2} if $\mid T_{r_2} - T_{s_2} \mid < \triangle T$ && $R^{'} = R$ $\varphi = \left(Z_{BS_j} \right)_x + SID_j$ $\xi = R^{'}\left(Z_{BS_j} + (\varphi) P_s \right)$ $\chi = r_i^{'} \oplus \left(\left(s^{'} G + \xi \right) x_{R_{i'}} \right)_x$ $Data_i = r_i^{'} - h(\chi) mod\,(n)$	$\xrightarrow{C}$	if $C = h\left(s \| SID_j \| m \| r_R \right)$,the $BS_j$is authenticated Else, it is rejected $IDs_{T_i}^{new} \leftarrow h(IDs_{T_i} \| m \| r_R \| x_{T_i})$

Figure 4.2. BDD17 RFID authentication scheme [39]

After BS_j sending the message $(C, R, r_1^{'}, r_2^{'}, \cdots, r_n^{'}, T_{s_2})$ to the reader, it updates $IDs_{T_i}^{new} \leftarrow h(IDs_{T_i} \| m \| r_R \| x_{T_i})$. When $R_{i'}$ receives the response of the server, it firstly verifies the validity of the timestamp, $\mid T_{r_2} - T_{r_1} \mid < \triangle T$. It also verifies the validity and integrity of the transmitted message by calculating: $\varphi = \left(Z_{BS_j}\right)_x + SID_j$, $\xi = R^{'}\left(Z_{BS_j} + (\varphi) P_s\right)$, $\chi = r_i^{'} \oplus \left(\left(s^{'} G + \xi\right) x_{R_{i'}}\right)_x$ and $Data_i = r_i^{'} - h(\chi) mod(n)$. If the verifications are succeeded, then the reader $R_{i'}$ relays the message C to the tag T_i for mutual authentication. When the tag receives C, it checks $C = h\left(s \| SID_j \| m \| r_R\right)$. If succeeded, T_i authenticates BS_j; else, it rejects. Finally, T_i updates its pseudonym $IDs_{T_i}^{new} \leftarrow h(IDs_{T_i} \| m \| r_R \| x_{T_i})$ and terminates the session.

Proposed Attacks on the Protocol: The authors claim that their protocol provides the forward security but we prove that when an adversary corrupts a tag, she can distinguish backward and forward transactions of the tag and destroy its privacy [39]. The authors, in their analysis, state that even if an attacker discovers the tag secret parameters, she cannot track the tag's past positions because she does not reach the timestamps and random values. However, we show that their scheme does not provide backward and forward privacy since an adversary can check the updates of the anonymous identifier IDs_{T_i} and break the tag's privacy. Formally, the adversary can perform the following attack.

Theorem 4.1.3 BDD17 protocol does not provide backward privacy.

Proof 4.1.3 Let *Adv* be a STRONG adversary that plays a security game as below.

1. *Adv* calls $\mathcal{O}^{CreateTag}(ID_{T_0}^0, 1)$ and $\mathcal{O}^{CreateTag}(ID_{T_1}^0, 1)$ to create two valid tags T_0 and T_1 with initial identifiers (the tags update their own identifier after authenticating the reader.), respectively.

2. *Adv* randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0,1\}$.

3. *Adv* chooses a time interval I_0. During I_0, she calls a $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle and gets $< Z_{BS_j}^{I_0}, ID_{\psi_{T_i}}^{I_0}, x_{\psi_{T_i}}^{I_0}, SID_j^{I_0}, P_s^{I_0}, m^{I_0} >$.

4. *Adv* frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.

5. *Adv* chooses another time interval I_1, where $I_1 > I_0$. During I_1, *Adv* calls $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.

6. *Adv* arbitrarily selects one of the drawn tags $(e.g.\ \psi_{T_1})$ and calls $\mathcal{O}^{Execute}(\psi_{T_1})$ oracle. She gets $r_R^{I_1^1}, auth^{I_1^1}, r^{I_1^1}, s^{I_1^1}, IDs_{\psi_{T_1}}^{I_1^1}, C^{I_1^1}$ as a protocol transcript.

7. *Adv* calls $\mathcal{O}^{Execute}(\psi_{T_1})$ oracle again and gets $r_R^{I_1^2}, auth^{I_1^2}, r^{I_1^2}, s^{I_1^2}, IDs_{\psi_{T_1}}^{I_1^2}, C^{I_1^2}$.

8. *Adv* frees the tags by calling $\mathcal{O}^{Free}(\psi_{T_0})$ and $\mathcal{O}^{Free}(\psi_{T_1})$ oracle.

9. Then, *Adv* tries to verify the message $IDs_{\psi_{T_1}}^{I_1^2}$ for the tag ψ_{T_i} by computing $IDs_{\psi_{T_1}}^{I_1^2} \stackrel{?}{=} h\left(IDs_{\psi_{T_1}}^{I_1^1} \| m^{I_0} \| r^{I_1^2} \| x_{\psi_{T_i}}^{I_0}\right)$.

10. If the verification is succeeded, she claims that $i = 1$ $(i.e. \psi_{T_i} = \psi_{T_1})$. Otherwise, she claims that $i = 0$ $(i.e. \psi_{T_i} = \psi_{T_0})$.

The success probability of this adversary is 1 and she wins the game. This means that *Adv* can distinguish the future interactions of T_i checking the updates of the anonymous identifier IDs_{T_i}. Therefore, this scheme does not provide backward privacy.

Theorem 4.1.4 BDD17 protocol does not provide forward privacy.

Proof 4.1.4 Let *Adv* be a STRONG adversary that plays a security game as below.

1. *Adv* calls $\mathcal{O}^{CreateTag}(ID_{T_0}^0, 1)$ and $\mathcal{O}^{CreateTag}(ID_{T_1}^0, 1)$ to create two valid tags T_0 and T_1 with initial identifiers (the tags update their own identifier after authenticating the reader), respectively.

2. *Adv* randomly picks two tags by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and gets two pseudonyms ψ_{T_0} and ψ_{T_1}.

3. *Adv* chooses a time interval I_0. During I_0, she arbitrarily selects one of the drawn tags $(e.g. \psi_{T_1})$ and calls $\mathcal{O}^{Execute}(\psi_{T_1})$ oracle. Then, *Adv* gets $r_R^{I_0^1}, auth^{I_0^1}, r^{I_0^1}, s^{I_0^1}, IDs_{\psi_{T_1}}^{I_0^1}, C^{I_0^1}$ as a protocol transcript for ψ_{T_1}.

4. *Adv* calls $\mathcal{O}^{Execute}(\psi_{T_1})$ oracle again and gets $r_R^{I_0^2}, auth^{I_0^2}, r^{I_0^2}, s^{I_0^2}, IDs_{\psi_{T_1}}^{I_0^2}, C^{I_0^2}$

5. *Adv* frees the tags by calling $\mathcal{O}^{Free}(\psi_{T_0})$ and $\mathcal{O}^{Free}(\psi_{T_1})$ oracle.

6. *Adv* chooses another time interval I_1, where $I_1 > I_0$. During I_1, *Adv* randomly chooses a tag T_i by calling $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0,1\}$.

7. *Adv* calls $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle and gets $< Z_{BS_j}^{I_1}, ID_{\psi_{T_i}}^{I_1}, x_{\psi_{T_i}}^{I_1}, SID_j^{I_1}, P_s^{I_1}, m^{I_1} >$

8. *Adv* frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.

9. Then, *Adv* tries to verify the message $IDs_{\psi_{T_1}}^{I_0^2}$ for the tag ψ_{T_i} by computing $IDs_{\psi_{T_1}}^{I_0^2} \stackrel{?}{=} h\left(IDs_{\psi_{T_1}}^{I_0^1} \| m^{I_1} \| r^{I_0^2} \| x_{\psi_{T_i}}^{I_1}\right)$.

10. If the verification is succeeded, she claims that $i = 1$ $(i.e. \psi_{T_i} = \psi_{T_1})$. Otherwise, she claims that $i = 0$ $(i.e. \psi_{T_i} = \psi_{T_0})$.

The success probability of this adversary is 1 and she wins the game. This means that *Adv* has stored some past transcripts. Then, when she obtains the internal values of the tag, she can identify the tag using previous transcripts by verifying the message $IDs_{T_i}^{new}$. Therefore, this scheme does not provide forward privacy as claimed.

4.1.3. Analysis of DB17 RFID authentication scheme

In this subsection, we first briefly describe the DB17 [45] scheme and then show our proposed attacks.

Protocol Description: DB17 scheme (illustrated in Figure 4.3) consists of 3 phases: setup phase, authentication phase, and updating phase. Before the authentication, public and private key pairs, ECC domain, and some system parameters are securely shared to the readers and the tags in the system. The authentication and updating phases are described below.

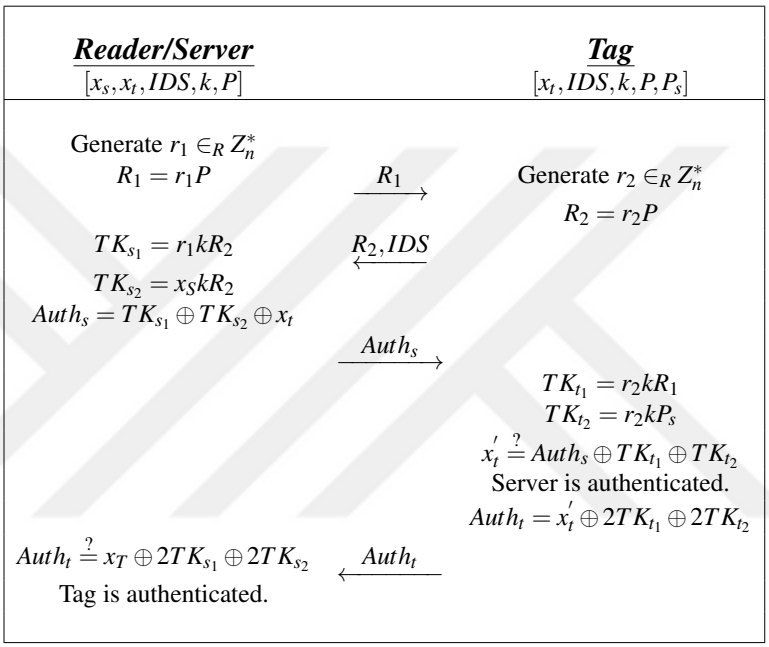

Figure 4.3. DB17 RFID authentication scheme [45]

Authentication Phase. In this phase, mutual authentication is provided. Firstly, the reader picks a random number r_1, computes R_1 and sends it to the tag. When the tag receives the challenge of the reader, the tag also picks a random number r_2, computes R_2 and sends R_2, and pseudonym IDS back to the reader. When the reader receives the response of the tag, it searches IDS in the database. If the reader does not find it, the reader terminates the protocol. Otherwise, the reader obtains the corresponding identifier (x_t) and key (k) corresponding to IDS^{new} or IDS^{old}. Then, the reader computes $TK_{S_1} = r_1 k R_2$, $TK_{S_2} = x_S k R_2$ and $Auth_s = TK_{s_1} \oplus TK_{s_2} \oplus x_t$. After receiving $Auth_s$, the tag computes $TK_{t_1} = r_2 k R_2$, $TK_{s_2} = x_s k R_2$ and checks if $x_t^{'} \stackrel{?}{=} Auth_s \oplus TK_{t_1} \oplus TK_{t_2}$. If the obtained identifier $x_t^{'}$ does not match, the tag terminates the session. Otherwise, the tag authenticates the reader, computes $Auth_t = x_t^{'} \oplus 2TK_{t_1} \oplus 2TK_{t_2}$ and sends $Auth_t$ to the reader. When the reader receives the message, it checks if $Auth_t \stackrel{?}{=} x_t \oplus 2TK_{s_1} \oplus 2TK_{s_2}$. If checking succeeds, the reader authenticates the tag. Otherwise, the reader rejects the $Auth_t$ and terminates the protocol.

Updating Phase. When the authentication is successfully accomplished, both the reader and the tag refresh their secret keys k and the pseudonyms (IDS). The reader also keeps old and new IDS. The tag performs the following updates:
$IDS^* = X(TK_{t_1}) \oplus IDS \oplus k, k^* = X(TK_{t_2}) \oplus 2k$ and $IDS = IDS^*, k = k^*$.

The reader performs the following updates: If IDS^{old} is received, the reader computes $IDS^{new} = X(TK_{S_1}) \oplus IDS^{old} \oplus k$ and $k^{new} = X(TK_{S_2}) \oplus 2k^{old}$. If IDS^{new} is received, the reader updates $IDS^{old} = IDS^{new}$ and $k^{old} = k^{new}$. The reader, then, computes $IDS^{new} = X(TK_{S_1}) \oplus IDS^{old} \oplus k$ and $k^{new} = X(TK_{S_2}) \oplus 2k^{old}$.

Proposed Attacks on the Protocol: Dinarvand and Barati [45] claim that their protocol (BD17) provides forward privacy. However, they do not mention backward privacy in their paper. In this subsection, we show that their scheme does not achieve backward privacy which is one of the well-known privacy requirements. In other words, we prove that when an adversary obtains the secrets of a tag once, she can distinguish the tag with using its future transactions. An adversary can directly reveal the identifier of the tag x_t with sending P_s to the tag instead of R_1 after obtaining the secrets of the tag. Formally, the adversary plays the following game to show how to break the forward untraceability property.

Theorem 4.1.5 BD17 scheme does not provide backward privacy.

Proof 4.1.5 Let Adv be a STRONG adversary that plays a security game as below.

1. Adv calls $\mathcal{O}^{CreateTag}(x_t^{T_0}, 1)$ and $\mathcal{O}^{CreateTag}(x_t^{T_1}, 1)$to create two valid tags T_0 and T_1, respectively, where $x_t^{T_0}$ denotes the identifier of a tag.

2. Adv randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0,1\}$.

3. Adv chooses a time interval I_0. During I_0, calls a $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle. She obtains all internal values of the tag with pseudonym ψ_{T_i}. These are $x_t^{T_i}, IDS_i, k_i, P$ and P_s.

4. Adv frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.

5. Adv chooses another time interval I_1, where $I_1 > I_0$. During I_1, Adv calls $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.

6. Adv arbitrarily selects one of the drawn tags $(e.g.\ \psi_{T_1})$ and calls $\mathcal{O}^{Launch}()$ oracle. She starts a new protocol execution with π_1.

7. Adv calls $\mathcal{O}^{SendTag}(P_s, \pi_1)$ oracle and she sends P_s message instead of R_1 message. The tag ψ_{T_1} responses with $R_2^{I_1}, IDS^{I_1}$but Adv does not need these messages.

8. Adv sends $x_t^{T_i}$ to the tag instead of $Auth_s^{I_1}$ message (in step-3) by calling $\mathcal{O}^{SendTag}(x_t^{T_i}, \pi_1)$ oracle and waits for the response of the tag.

9. If the tag ψ_{T_1} responds with $Auth_t^{I_1}$, Adv directly gets $x_t^{T_1}$ and claims that $i = 1$ $(\psi_{T_i} = \psi_{T_1})$, since the response means that the tag authenticates Adv. In fact, $Auth_t^{I_1} = x_t^{T_1}$ because of $Auth_t^{I_1} = x_t^{T_1} \oplus 2(r_2 k_1 P_s) \oplus 2(r_2 k_1 P_s)$.

10. If the tag ψ_{T_1} does not respond, this means that the tag does not authenticate and terminates the session. Therefore, Adv claims that $i = 0$ $(i.e. \psi_{T_i} = \psi_{T_0})$.

Obviously, the success probability of this adversary is 1 and she wins the game. Therefore, BD17's scheme does not provide forward untraceability property.

4.1.4. Analysis of LZKZ18 RFID Authentication Scheme

In this subsection, we first describe LZKZ18 [46] scheme and then show our proposed attacks.

Protocol Description: LZKZ18 scheme (illustrated in Figure 4.4) includes two processes: a setup process and an implementation process. In the setup process which is also divided into initialization and bidirectional authentication phases, the server and the reader securely share and store the needed keys. The reader, server and tag also agree on the ECC domain parameters.

Server $[T_D, R_D, k_{AB}, k_{AC}, a, P_S = aP]$		**Reader** $[R_D, k_{AB}, b, P_R = bP]$		**Tag** $[T_D, k_{AC}, c, P_T = cP, P_S]$
		$x_R \in_R Z_q$	(1) $\xrightarrow{Query, R_1}$	$x_T \in_R Z_q$
		$R_1 = x_R P$		$T_1 = x_T P$
				$T_2 = H(x_T R_1)$
$x_S \in_R Z_q$, $S_1 = x_S P$		$R_2 = H(x_R T_1)$	(2) $\xleftarrow{T_1, T_2, T_3}$	$T_3 = T_D + (x_T + c) P_S$
$S_2 = H(R_1 \| k_{AB} \| t_R)$	(3) $\xleftarrow[T_1, T_3, t_R]{R_1, R_3, R_4}$	Judge: $R_2 \stackrel{?}{=} T_2$		
Judge: $S_2 \stackrel{?}{=} R_3$		$R_3 = H(R_1 \| k_{AB} \| t_R)$		
$S_3 = R_4 - aR_1 - k_{AB}$		$R_4 = R_D + (x_R + b) P_S$		
Judge: $S_3 \stackrel{?}{=} R_D$				
$S_4 = T_3 - aT_1 - k_{AC}$				
Judge: $S_4 \stackrel{?}{=} T_D$				
$S_5 = x_S R_1 + k_{AB}$				
$S_6 = x_S T_1 + k_{AC}$	(4) $\xrightarrow{S_1, S_5, S_6}$	$R_5 = x_R S_1 + k_{AB}$	(5) $\xrightarrow{S_1, S_6}$	$T_4 = x_T S_1 + k_{AC}$
		Judge: $R_5 \stackrel{?}{=} S_5$		Judge: $T_4 \stackrel{?}{=} S_6$

Figure 4.4. LZKZ18 RFID authentication scheme [46]

In the implementation process, at first, the reader picks a random x_R, computes R_1 and initiates a new protocol session sending the query request $Query$ and R_1. When the tag receives a request, the tag picks a random x_T and computes T_2 and T_3. The tag sends T_1, T_2 and T_3 to the reader. When the reader receives the response of the tag, the reader computes R_2 and checks if $R_2 \stackrel{?}{=} T_2$. If the checking is false, the authentication fails and the session drops. Otherwise, the reader considers that the tag is legitimate and computes R_3 and R_4. Then, the reader sends R_1, R_2, R_3, T_1, T_3 and t_R to the server. After receiving the

message, the server first checks the timestamp t_R. If the t_R exceed the time limit, the server finishes the authentication. Otherwise, the server picks a random number x_S and computes S_1 and S_2. If $S_2 \neq R_3$, then the authentication fails. Otherwise, the server authenticates the reader and calculates S_3. If $S_3 \neq R_D$, then the authentication fails again. Otherwise, the server obtains the reader's authorization the identifier and computes S_4 and checks if $S_4 \stackrel{?}{=} T_D$. If $S_4 \neq T_D$, then the authentication fails. Otherwise, the server obtains the tag's authorization identifier and calculates S_5 and S_6. Later on, the server sends S_1, S_5 and S_6 to the reader. When the reader gets this, it computes R_5 and checks if $R_5 \stackrel{?}{=} S_5$. If $R_5 = S_5$, the reader authenticates the server. Otherwise, the authentication fails. After the successful authentication, the reader sends S_1, S_6 to the tag. The tag then computes T_4. Finally, the tag checks if $T_4 \stackrel{?}{=} S_6$. If $T_4 \neq S_6$, the tag rejects the authentication and terminates the session. Otherwise, the tag authenticates the reader and the back-end server, too.

Proposed Attacks on the Protocol: The authors claim that LZKZ18 protocol provides forward security without presenting any analysis [46]. In this thesis, we show that their scheme does not fulfill both backward and forward privacy because an adversary can destroy the privacy of a tag by sending P_s instead of R_1 and checking if $T_2 \stackrel{?}{=} H(T_3 - T_D - cP_S)$.

Theorem 4.1.6 LZKZ18 scheme does not provide backward privacy.

Proof 4.1.6 Let *Adv* is a STRONG adversary that plays a security game as below.

1. *Adv* calls $\mathcal{O}^{CreateTag}(T_{D_0}, 1)$ and $\mathcal{O}^{CreateTag}(T_{D_1}, 1)$ to create two valid tags T_0 and T_1, respectively.

2. *Adv* randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0,1\}$.

3. *Adv* chooses a time interval I_0. During I_0, calls a $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle. She obtains all internal values of the tag with pseudonym ψ_{T_i}. These are $T_{D_i}, k_{AC}, c_i, P_{T_i}$ and P_S.

4. *Adv* frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.

5. *Adv* chooses another time interval I_1, where $I_1 > I_0$.

   During I_1, *Adv* calls $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.

6. *Adv* arbitrarily selects one of the drawn tags $(e.g.\ \psi_{T_1})$ and calls $\mathcal{O}^{Launch}()$ oracle. She starts a new protocol execution with π_1

7. *Adv* calls $\mathcal{O}^{SendTag}(P_S, \pi_1)$ oracle so she sends P_S message instead of R_1 message. The tag ψ_{T_1} responses with $T_1^{I_1}, T_2^{I_1}, T_3^{I_1}$.

8. *Adv* checks $T_2^{I_1} \stackrel{?}{=} H\left(T_3^{I_1} - T_{D_i} - c_i P_S\right)$.

   If succeeds, *Adv* claims that $i = 1$ $(i.e. \psi_{T_i} = \psi_{T_1})$. Otherwise, *Adv* claims that $i = 0$ $(i.e. \psi_{T_i} = \psi_{T_0})$.

Obviously, the success probability of this adversary is 1 and she wins the game. Therefore, LZKZ's scheme does not ensure backward privacy.

Theorem 4.1.7 LZKZ18 scheme does not provide forward privacy.

Proof 4.1.7 Let *Adv* be a STRONG adversary that plays a security game as below.

1. *Adv* calls $\mathcal{O}^{CreateTag}(T_{D_0}, 1)$ and $\mathcal{O}^{CreateTag}(T_{D_1}, 1)$ to create two valid tags T_0 and T_1, respectively.

2. *Adv* randomly picks two tags by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and gets two pseudonyms ψ_{T_0} and ψ_{T_1}.

3. *Adv* chooses a time interval I_0. During I_0, she arbitrarily selects one of the drawn tags $(e.g.\ \psi_{T_1})$ and calls $\mathcal{O}^{Launch}()$ oracle. *Adv* starts a new protocol execution with π_1

4. *Adv* calls $\mathcal{O}^{SendTag}(P_S, \pi_1)$ oracle so she sends P_S message instead of R_1message. The tag ψ_{T_1} responses with $T_1^{I_0}, T_2^{I_0}, T_3^{I_0}$.

5. Then, *Adv* frees the tags by calling $\mathcal{O}^{Free}(\psi_{T_0})$ and $\mathcal{O}^{Free}(\psi_{T_1})$ oracle.

6. *Adv* chooses another time interval I_1, where $I_1 > I_0$. During I_1, *Adv* randomly chooses a tag T_i by calling $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0,1\}$.

7. *Adv* calls $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle and gets $T_{D_i}, k_{AC}, c_i, P_{T_i}$ and P_S.

8. *Adv* frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.

9. *Then, Adv* checks if $T_2^{I_0} \stackrel{?}{=} H\left(T_3^{I_0} - T_{D_i} - c_i P_S\right)$. If succeeds, *Adv* claims that $i = 1$ $(i.e. \psi_{T_i} = \psi_{T_1})$. Otherwise, *Adv* claims that $i = 0$ $(i.e. \psi_{T_i} = \psi_{T_0})$.

Obviously, the success probability of this adversary is 1 and she wins the game. Therefore, LZKZ's scheme does not provide forward privacy.

4.2. Our Improved Protocol

We propose a new privacy-friendly ECC based RFID authentication protocol depicted in Figure 4.5 by enhancing ID17 scheme [40]. Our focus is to overcome the privacy weaknesses of their protocol. We consider that transmitting the ephemeral public key and its signature in an insecure channel causes privacy issues. Therefore, we claim that if an ephemeral public key is broadcasted with an indistinguishable encrypted signature, then an

attacker cannot track the past and future interactions of any tag so that both forward and backward untraceability properties are provided.

We consider that both the reader and the back-end server (BS) are trusted entities but a tag might be corrupted, compromised or illegitimate. For the sake of simplicity, we also consider both BS and reader as a single entity and the tag is the second entity of our scheme. Note that this does not affect the generality since most of the applications accept that the communication of tag−reader is not secure but the communication of reader−BS is secure (as shown in Figure 1.1). Before describing the protocol, we present the following notations in Table 4.1 to improve the intelligibility.

Table 4.1. Notations of our proposed protocol

p,a,b,G,n,h	ECC domain parameters
$k_R^{'},k_r^{'}$	Static key pairs (public, private) of the reader
$k_T^{'},k_t^{'}$	Static key pairs (public, private) of the tag
ID_i	Unique identifier of i^{th} tag
k_R,k_r	Ephemeral key pairs (public, private) of the reader
(z,s)	Signature of the ephemeral public key of the reader
k_{TT},k_t	Ephemeral key pairs (public, private) of the tag
(g,f)	Signature of the ephemeral public key of the tag
k_{TR}	Established ephemeral shared secret key after ECDH key agreement protocol
$Hash(.)$	A secure cryptographic hash function

4.2.1. Protocol description

We present a brief description of our scheme below. Figure 4.5 also elaborately shows the details. Our proposed protocol consists of a setup phase and an authentication phase.

Setup Phase. Reader and tags must agree on ECC domain parameters of the scheme to use elliptic curve cryptosystem. Hence, in the setup, both tags and readers firstly agree on a curve with ECC domain parameters. In our scheme, we prefer ECC brainpoolP160r1, a standard curve, to be used for the domain parameter values [98]. In this phase, all unique identifiers ID_i of the tags are stored in BS. An integer $k_t^{'}$ is randomly chosen as the private key of the tag, where $1 \leq k_t^{'} \leq n-1$ and its public key is computed as $k_T^{'} = k_t^{'}G$. Then, the key pairs are stored in the tag. $k_T^{'}$ is shared with the reader. On the other hand, an integer $k_r^{'}$ is randomly chosen as the private key of the reader, where $1 \leq k_r^{'} \leq n-1$ and its public key is computed as $k_R^{'} = k_r^{'}G$. Then, the key pairs are stored in BS, while $k_R^{'}$ is shared with all tags.

Authentication Phase. In this phase, the mutual authentication is accomplished in two rounds with the following steps. Note that Figure 4.5 also depicts each step of our protocol execution.

(Step-1): First, the reader randomly generates an ephemeral private key k_r and calculates its own ephemeral public key, where $k_R = k_rG$.

(Step-2): The reader signs k_R with its private key $k_r^{'}$ using ECDSA, $(z,s) = ECDSA_{k_r^{'}}(k_R)$.

2.1 $e = Hash(k_R)$

2.2 Select k randomly

2.3 $z = x_1 mod\,(n)$; if $z = 0$ then go to (2.2)

2.4 $s = k^{-1}\left(e + k_r^{'} z\right) mod(n)$; if $s = 0$ then go to (2.2)

(Step-3): The reader sends k_R and the signature (z,s) to the tag.

(Step-4): The tag firstly verifies k_R using the public key of the reader $k_R^{'}$.

4.1 Check if $z,s \in [1,n-1]$; If not, rejects the session

Reader $\left[a,b,p,G,n,h,k_R^{'},k_r^{'},k_T^{'}\right]$		**Tag** $\left[a,b,p,G,n,h,k_R^{'},k_t^{'},k_T^{'},ID\right]$
Pick k_r randomly as private key (1) $k_R = k_r G$ (2) Sing k_R: $\{(z,s) = sign(k_R)\}$ (2.1) $e = Hash(k_R)$ (2.2) Select k randomly (2.3) $z = x_1 mod\,(n)$; if $z = 0$ then go to (2.2) (2.4) $s = k^{-1}\left(e + k_r^{'} z\right) mod(n)$; if $s = 0$ then go to (2.2)		
	(3) $\xrightarrow{k_R,z,s}$	
		(4) Verifying k_R (4.1) Check if $z,s \in [1,n-1]$ If not, rejects the session. (4.2) $e = Hash(k_R)$ (4.3) $w = s^{-1}\ mod\,(n)$ (4.4) $u_1 = ew\ mod\,(n)$ and $u_2 = zw\ mod\,(n)$ (4.5)$(x_1,x_2) = u_1 G + u_2 k_R^{'}$ (4.6) If $x_1 \stackrel{?}{=} z\ mod\,(n)$, reader is authenticated; else, it is not and rejects the session (5) Pick k_t randomly as private key $k_{TT} = k_t G$ (6) Sing k_{TT}: $\{(g,f) = sign(k_{TT})\}$ (6.1) $e = Hash(k_{TT})$ (6.2) Select γ randomly (6.3) $g = x_2 mod\,(n)$; if $g = 0$ then go to (6.2) (6.4) $f = \gamma^{-1}\left(e + k_t^{'} g\right) mod(n)$; if $f = 0$ then go to (6.2) (7) $K_{TR} = k_t k_R = k_t\,(k_r G) = k_t k_r G$ (8) $C = AES_{K_{TR}}\,(ID \| g \| f)$
	(9) $\xleftarrow{k_{TT},C}$	
(10) $K_{TR} = k_r k_{TT} = k_r\,(k_t G) = k_r k_t G$ (11) $[ID \| g \| f] = AES^{-1}_{K_{TR}}\,(C)$ (12) Verifying k_{TT} (12.1) Check if $g,f \in [1,n-1]$ If not, rejects the session. (12.2) $e = Hash(k_{TT})$ (12.3) $i = f^{-1}\ mod\,(n)$ (12.4) $j_1 = ei\ mod\,(n)$ and $j_2 = gi\ mod\,(n)$ (12.5)$(x_2,y_2) = j_1 G + j_2 k_T^{'}$ (12.6) If $x_2 \neq g\ mod\,(n)$, reader rejects the session. (13) If $x_2 \stackrel{?}{=} g\ mod\,(n)$ and $ID = ID_i$ tag is authenticated; else, it is not and rejects the session		

Figure 4.5. Our proposed scheme

4.2 $e = Hash(k_R)$
4.3 $w = s^{-1} \; mod \, (n)$
4.4 $u_1 = ew \; mod \, (n)$ and $u_2 = zw \; mod \, (n)$
4.5 $(x_1, x_2) = u_1 G + u_2 k_R^{'}$
4.6 If $x_1 \stackrel{?}{=} z \; mod \, (n)$, Reader is authenticated; Else, rejects the session

(Step-5): If the verification is succeeded, the tag will authenticate the reader. Otherwise, it rejects the session. In case of authentication, the tag also randomly picks k_t as an ephemeral private key and computes its own ephemeral public key, $k_{TT} = k_t G$.

(Step-6): The tag signs k_{TT} and gets its signature (g, f) by computing
$(g, h) = ECDSA_{k_t^{'}} (k_{TT})$.
6.1 $e = Hash(k_{TT})$
6.2 Select γ randomly
6.3 $g = x_2 mod \, (n)$; if $g = 0$ then go to (6.2)
6.4 $f = \gamma^{-1} \left(e + k_t^{'} g \right) mod(n)$; if $f = 0$ then go to (6.2)

(Step-7): The tag calculates $K_{TR} = k_t k_R$ as an ephemeral shared secret key.

(Step-8): The tag encrypts ID and the signature (g, f) together as a plaintext using the ephemeral key K_{TR}, where $C = AES_{K_{TR}} (ID \| g \| f)$.

(Step-9): The tag transmits only C and k_{TT} messages to the reader.

(Step-10): When the reader receives the message of the tag, the reader computes the ephemeral shared secret key $K_{TR} = k_r k_{TT}$, where $k_r k_{TT} = k_r (k_t G) = k_r k_t G$.

(Step-11): Then, the reader can meaningfully decrypt message C using K_{TR}, obtain ID and signature (g, f) if the shared key is valid. Otherwise, the reader has a garbage message.

(Step-12): The reader verifies the decrypted messages. It checks if g and f are integers in the range $[1, n-1]$. If not, it rejects the session. After that, the reader also verifies the k_{TT} with using the decrypted signature (g, f). If the verification is not succeeded, it rejects the session.
12.1 Check if $g, f \in [1, n-1]$ If not, rejects the session.
12.2 $e = Hash(k_{TT})$
12.3 $i = f^{-1} \; mod \, (n)$
12.4 $j_1 = ei \; mod \, (n)$ and $j_2 = gi \; mod \, (n)$
12.5 $(x_2, y_2) = j_1 G + j_2 k_T^{'}$
12.6 If $x_2 \neq g \; mod \, (n)$, reader rejects the session.

(Step-13): The reader checks if $x_2 \stackrel{?}{=} g \; mod \, (n)$ and searches if the ID belongs to a tag

registered in the database ($ID = ID_i$), the tag is authenticated. Otherwise, the reader rejects the session.

4.3. Security Analysis of Our Proposed Protocol

In this section, we give the security and privacy analysis of our proposed protocol and prove that our scheme provides all essential security and privacy properties.

Theorem 4.3.1 Our protocol provides confidentiality.

Proof 4.3.1 In our protocol, the sensitive information is the identity of tag ID and the private keys of the reader and the tag. The private keys are protected well and are not transmitted. Furthermore, ID is transmitted as ciphertext encrypted by AES. The key of AES is ephemerally derived using elliptic-curve DiffieHellman mechanism by both the reader and the tag. Therefore, an adversary Adv who collects k_R, z, s, k_{TT} and C transcripts, cannot obtain any confidential information without breaking AES or ECDHE in polynomial time.

Theorem 4.3.2 Our protocol provides integrity.

Proof 4.3.2 In our protocol, we use the ECDSA signatures that are basically used to provide the integrity of k_R and k_{TT} messages. An adversary Adv cannot change the content of the protocol transcripts because both the reader and the tag verify the transmitted signatures (z, s) and (g, f). Adv can modify the transmitted message or forge the related signatures if she solves the elliptic curve discrete logarithm problem (ECDLP) but ECDSA is computationally secure and it is a hard problem for polynomial time attackers. Consequently, the protocol guarantees the integrity of transmitted messages.

Theorem 4.3.3 Our protocol provides availability.

Proof 4.3.3 In our protocol, the tag identifier ID and the pre-shared keys are securely stored and protected well. Hence, it is not needed to synchronously refresh these values for our scheme. In fact, there is no update mechanism between the tag and reader. Therefore, the protocol can be executed all the time between the reader and the tag. Hence, our scheme provides availability.

Theorem 4.3.4 Our protocol provides tag anonymity.

Proof 4.3.4 In the authentication phase, the tag responds when it receives challenges from the reader. Hence, anonymity is becoming one of the utmost important and imperative security requirement for privacy. In our protocol, an adversary Adv collects the only k_R, z, s, k_{TT} and C transcripts and cannot reach the tag identifier ID because Adv is not able to ECDLP in polynomial time and gain k_{TR}. Adv also cannot break C without having k_{TR} because AES-128 is considered computationally secure. In fact, Theorem 4.3.1 also shows

that *Adv* can never obtain any confidential information. Moreover, if k_{TT} and C messages were not randomly generated for each session, the adversary can ruin the anonymity. However, all messages of the tag in our scheme are randomized for each protocol session and *Adv* cannot even distinguish any tag's messages sent in different sessions. Therefore, the protocol achieves tag's anonymity property and the adversary cannot attain any indicator to point out a tag anymore.

Theorem 4.3.5 Our protocol provides mutual authentication.

Proof 4.3.5 Mutual authentication (two-way authentication) is an important property in which both entities in a protocol link authenticate each other. In the authentication phase of our protocol, the reader sends randomly generated k_{TR} and its signature z, s by using ECDSA. The tag can verify k_{TR} using the pre-shared public key of the reader $k_R^{'}$, herewith the reader can be authenticated. Likewise, the reader authenticates the tag after verifying k_{TT}. For this authentication, the reader firstly decrypts C, gets the unique tag identifier ID and the ECDSA signature of k_{TT} which is g, f. Secondly, the reader verifies g, f using the pre-shared public key of the tag $k_T^{'}$. If the verification is successful, the reader finally searches ID in its database. If the reader finds it (matches $ID = ID_i$), the tag is authenticated, too. Therefore, the proposed protocol provides mutual authentication.

Theorem 4.3.6 Our protocol provides scalability.

Proof 4.3.6 The scalability is a crucial property that reduces the computational cost, searching time of a tag in the database and authentication time. In most cases, the searching time linearly increases proportionally proliferating the registered tags in the database with search complexity $O(N)$, where N is the number of valid tags. In our protocol, the reader decrypts C and gets the ID_i (where $1 \leq i \leq N$). Then, the reader searches the matched entry in the database with search complexity $O(1)$ because each entry ID_i matches only one tag in DB. Therefore, our proposed protocol is scalable.

Theorem 4.3.7 Our protocol provides forward privacy.

Proof 4.3.7 Forward security is explained in Definition 5. In our proposed protocol, the reader freshly sends k_R and its signature z, s for each protocol session. The tag also generates a new fresh k_{TT} and C messages. The ephemeral K_{TR} ensures that C is randomized. Because of randomization of all session messages, if a probabilistic polynomial-time (ppt) adversary *Adv* corrupts a tag T, discloses the secrets $ID, k_t^{'}$ and collects the past protocol transcripts, *Adv* can distinguish the corrupted tag and its transactions with a negligible probability. *Adv* never gets any advantage to overcome the previous indistinguishable transactions of our scheme. Therefore, our protocol provides backward untraceability property.

Theorem 4.3.8 Our protocol provides backward privacy.

Proof 4.3.8 As mentioned in the proof of Theorem 4.3.7, if the same adversary *Adv* collects the future protocol transcripts, *Adv* can distinguish the corrupted tag and its transactions with a negligible probability. *Adv* never gets any advantage to overcome the future indistinguishable transaction of our scheme. Therefore, our protocol provides forward untraceability property.

Theorem 4.3.9 Our protocol provides location privacy, traceability privacy and withstands the tracking attack.

Proof 4.3.9 In Theorem 4.3.7 and Theorem 4.3.8, we prove that future and backward untraceability property of our protocol. An adversary *Adv* cannot destroy the privacy of a tag T, even if *Adv* has the secrets of T and the past/future protocol transcripts in related protocols. Hence, *Adv* certainly cannot ruin location privacy of T without any confidential information of the tag and track T. In other words, untraceability properties imply this result. Therefore, our protocol provides location privacy, traceability privacy and it is resistant against the tracking attack.

Theorem 4.3.10 Our protocol withstands the tag impersonation and reader spoofing attacks.

Proof 4.3.10 An adversary *Adv* can impersonate a tag T only by obtaining ID and $k_t^{'}$ but solving ECDLP is computationally infeasible in polynomial time. Hence, *Adv* cannot impersonate T. Similarly, *Adv* can never produce valid C, z, s messages without having K_{TR}, ID and $k_t^{'}$ because of the aforementioned computational infeasibilities. Thus, *Adv* cannot spoof the reader.

Theorem 4.3.11 Our protocol withstands the replay attack.

Proof 4.3.11 In a replay attack, an adversary *Adv* imitates a tag *Adv* or a reader R by reusing the intercepted past protocol messages. In our proposed protocol, *Adv* cannot generate and reuse valid k_R, z, s messages because they are randomly changed for each session. Similarly, *Adv* cannot generate and reuse valid k_{TT}, C messages because they are ephemerally generated random transcripts. This attack can be succeeded, only if *Adv* reveals the tag secrets $k_t^{'}, ID$ and reader private key $k_r^{'}$. Therefore, the proposed protocol is resistant against the replay attack.

Theorem 4.3.12 Our protocol withstands the denial-of-service (DoS) and de-synchronization attack.

Proof 4.3.12 We prove that our proposed protocol provides availability in Theorem 4.3.3 which shows that both a tag and a reader always remain synchronized during each protocol execution. An adversary cannot desynchronize both entities and execute DoS attack. Thus, the scheme is resistant against the denial-of-service and de-synchronization attack.

Theorem 4.3.13 Our protocol withstands the man-in-the-middle attack (MiTM).

Proof 4.3.13 According to Theorem 4.3.5, our proposed protocol provides mutual authentication between the tag and the reader. Therefore, it is resistant to MiTM attack.

Theorem 4.3.14 Our protocol withstands the cloning attacks.

Proof 4.3.14 In our proposed protocol, each tag has its own identity ID_i and the secret key $t^{'}$. Even if an adversary can obtain some tags' *IDs* and their private keys, she cannot reach the other tags' *IDs* and their secret keys. Thus, the protocol is resistant to the cloning attack.

Theorem 4.3.15 Our protocol provides unforgeability.

Proof 4.3.15 In our scheme, only the valid tag and the reader can generate a legitimate signature. An adversary can never perform a forgery attack without having the private keys as their security leans to the hardness of ECDLP. Therefore, the proposed protocol provides unforgeability.

Theorem 4.3.16 Our protocol withstands modification attack

Proof 4.3.16 According to Theorem 4.3.2, our proposed protocol provides integrity property. Therefore, it is resistant to any modification attacks.

4.4. Our Test Environment

In this section, we would like to introduce our test environment to express our simulation and implementation outcomes more clearly. We hope that these succinct explanations will be useful for new implementors.

We prefer the BasicCard RFID environment because of several reasons. The first and important one is the BasicCard RFID tags support many standard cryptographic algorithms and primitives. Secondly, this environment presents a flexible and easy simulation and realization structure. Thirdly, the environment can be set up by on-the-self standard products, since it supports communication standards (ISO-14443, ISO 15693, etc.) and ISO-defined commands for programming. Fourthly, its program development software is updated and clear to understand. Finally, the reader and the RFID tags are inexpensive with respect to their robust and efficient platform facilities.

Table 4.2. Security and Privacy Comparison

Security and Privacy Properties	LH14 [73]	Z14 [125]	C14 [113]	ZQ14 [129]	BDD17 [39]	ID17 [40]	DB17 [45]	LZKZ18 [46]	Our Protocol
Mutual authentication	x	✓	x	✓	✓	✓	✓	✓	✓
Confidentiality	x	x	✓	✓	✓	✓	✓	✓	✓
Integrity	–	–	–	–	✓	✓	✓	✓	✓
Availability	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tag anonymity	x	x	x	✓	✓	✓	✓	✓	✓
Location privacy	x	x	x	x	✓	✓	✓	✓	✓
Scalability	✓	✓	✓	✓	✓	✓	✓	✓	✓
Forward privacy	x	x	x	x	x	x	✓	x	✓
Backward privacy	x	x	x	x	x	x	x	x	✓
Tag impersonation att. res.	x	✓	x	✓	✓	✓	✓	✓	✓
Reader spoofing att. res.	x	✓	✓	✓	✓	✓	✓	✓	✓
Replay attack res.	✓	✓	✓	✓	✓	✓	✓	✓	✓
DoS attack res.	✓	✓	✓	✓	✓	✓	✓	✓	✓
MiTM attack resistance	–	–	x	✓	✓	✓	✓	✓	✓
Desynchronization att. res.	–	–	✓	✓	✓	✓	✓	✓	✓
Cloning attack resistance	x	✓	x	✓	✓	✓	✓	✓	✓

✓: provide, x: do not provide, –: not mentioned

4.4.1. The setup

We implemented our proposed scheme in ZeitControl's BasicCard environment [140]. The implementation environment mainly consists of a personal computer as as RFID back-end server (or database), an RFID reader and a ZeitControl's BasicCard RFID tag. The environment is illustrated in Figure 4.6. Note that the back-end server, the reader, and the tag are called as terminal, card reader, programmable processor card in the BasicCard's manual, respectively [140].

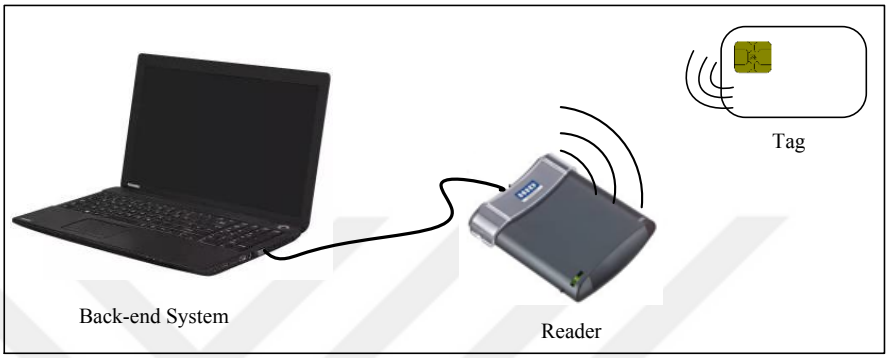

Figure 4.6. The setup of our implementation environment

We use a personal computer as a back-end server which has Intel Core i5 CPU processor @2.5GHz, 6GB RAM and 64-bit Windows 7 operating system to run simulations and implementation tests. The computer basically controls the reader and stores the system data.

We use the OMNIKEY 5321 device as an RFID reader. The reader complies with ISO 15693 and ISO 14443 standards and can communicate 13.56 MHz RFID tags. The interface of the reader and the terminal (back-end server) is standard universal serial bus (USB).

We implement our proposed scheme in professional version BasicCard ZC7.5 RFID card supporting ISO-14443 standard as RFID tag. The tag contains $32K$ of EEPROM (Electrically Erasable, Programmable Read-Only Memory) and $4.3K$ RAM (Random Access Memory). In the tag, there are also three processors such as CPU, RSA/ECC, and DES/AES co-processors. The overview of the RFID reader and tag used in the test is depicted in Figure 4.7.

The tag supports RSA (4096 bits), EC-167, EC-211, EC-p (up to 544 bits) as public-key algorithms, DES (ANSI X3.92-1981), AES (FIPS 197), EAX (a conventional authenticated encryption mode using AES), OMAC (One-Key CBC MAC using AES), ISO secure messaging as private key encryption schemes, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 as hash functions, both T=0 and TL=1 (byte-level or block transmission protocol ANSI X3.92-1981), and Mifare (NXP Semiconductors RFID protocol) as communication

protocols. The tag has P-code interpreter that compiles the programs coded with a high-level programming language Java or Basic into P-code. P-code is a machine independent language of a hypothetical CPU. Interpret can execute the P-code after downloading it to the tag.

Figure 4.7. RFID reader and RFID tag

Moreover, the used chip hardware in the tag is certified according to Common Criteria level EAL5+ with certification id "BSI-DSZ-CC-0555-2009". Note that the software (BasicCard operating system) is not included in this certification.

4.4.2. Simulation and implementation

The ZC-Basic language was designed for cryptographic protocol designers to easily develop their programs, applications or codes, simulate, and execute them in a real-world platform. It is a good opportunity that a developer can run his/her programs in PC with or without a real RFID reader attached to the serial port.

We can test our codes even if we do not have RFID reader and tag, by simulating the BasicCard environment in the computer. This feature is quite useful for protocol designers before testing their schemes in real-wold applications. The development software of the BasicCard environment, which is free and functional, supports a higher level language such as Java or ZC-Basic (dialect of Basic). We write our programs with ZC-Basic language that a procedure-oriented language because using ZC-Basic is easy to program and there is a detailed library about its usage. Additionally, the heart of a BasicCard processor is its P-Code (like Java programming language) interpreter and written codes are compiled into a machine-independent language called P-Code which is similar to machine code [140].

The development software basically composed of software support packages BCDevEnv which is the BasicCard development environment, ZCMDTerm which is debugger for the terminal program, ZCMDCard which is debugger for the tag program, ZCMBasic which is the compiler for the ZC-Basic language, and BCLoad which is downloading P-Code to

the tag.

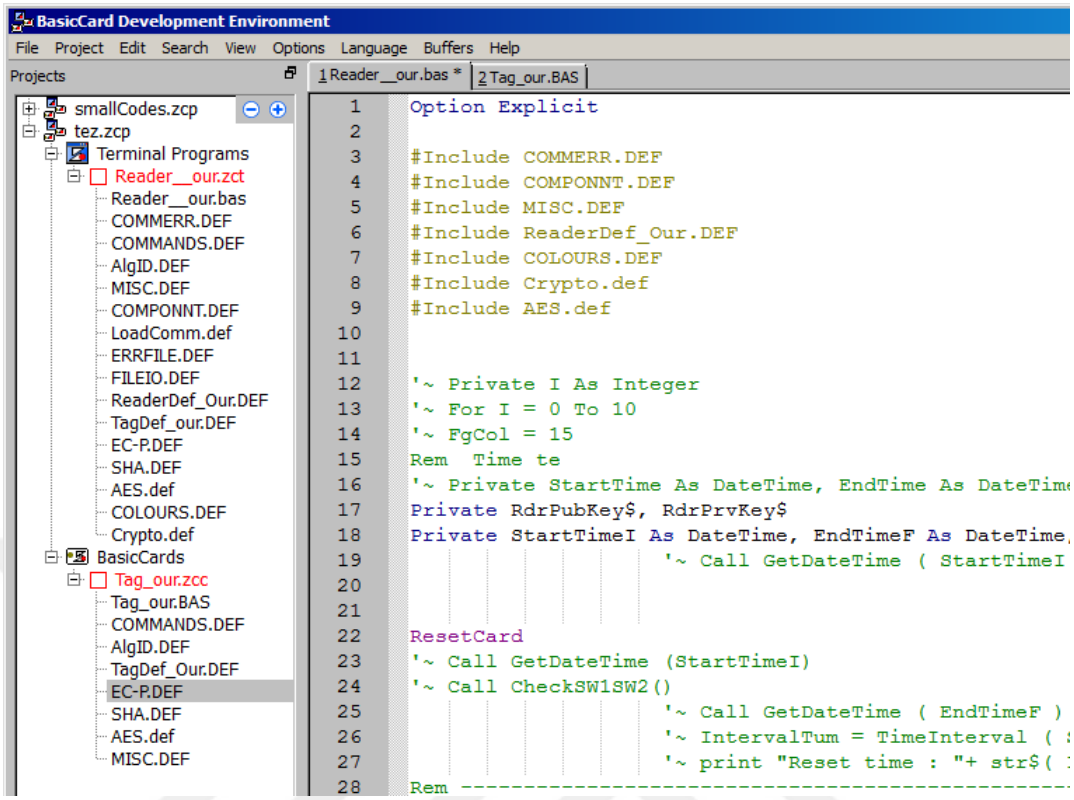

Figure 4.8. The BasicCard development environment

At this point, without going into further detail, we should explain a couple of files used in our BasicCard environment to present a clear and essential idea on how to effectively develop the protocol in the environment. They might be arranged in three different hierarchies: source files, program files, and project files. The three main software. Correspondingly, the project files (*.zcp) are related to the development environment and provide to list and manage all the program files that belong for a specific project. There are two types of program files such as terminal program files (with *.zct extension) and BasicCard program files (with *.zcc extension) The program files (with *.zcp) that both contains compiler and run-time options for virtual running (simulation) and real-time working. Lastly, the source and definition files (with *.bas and *.def extensions) contain sources codes, definitions, and declarations written by the program developer.

In fact, the three main software (BCDevEnv, CMDTerm/ZCMDCard, and ZCMBasic) can be said to be corresponding to the above files hierarchy. The project files can be managed (creating and editing) using by BCDevEnv (depicted in Figure 4.8). On the other hand, ZCMDTerm allows the terminal to communicate with ZCMDCard debuggers, and one or more physical RFID readers. It stores the required information for compiling and running the terminal programs. Finally, ZCMDCard is another debugger that waits and evaluates the commands from ZCMDTerm and then executes them and relays the responses back to ZCMDTerm.

From the implementation perspective, we have chosen a binary 160-bit elliptic curve complying with brainpoolP160r1 standard to elliptic curve functions such as ECDH and ECDSA. The related curve parameters (see Section 1.3.2 for details) and some preparations can be set calling subroutine ECpSetCurve with corresponding curve index. Moreover, we use AES algorithm in our protocol with 128 bit of key length. We have written our own function of AES algorithm to implement the desired modes of operation. We have also declared a command that provides a challenge and response mechanism for the communication of RFID reader and RFID tag. Lastly, one critical tip for efficient realization is that the development software allows you to adjust the speed of the cores. For further and comprehensive information, and utilization of the related functions and subroutines, please see the BasicCard Developer Manual [140].

4.5. Comparison and Implementation Results

In this section, we compare our proposed scheme with other existing ECC-based RFID authentication works in terms of security and performance.

4.5.1. Security comparison

We enumerate the security and privacy comparison of our proposed scheme and related protocols in Table 4.2. It can be obviously seen that our scheme provides all essential security and privacy requirements of an RFID system and is more secure than the previously proposed protocols [39, 40, 45, 46, 73, 113, 125, 129].

Furthermore, we proved in Section 4.1 that the state-of-the-art protocols [39, 40, 45, 46] cannot provide forward and/or backward privacy. Our scheme not only guarantees the related security and privacy requirements but also provides additional properties such as mutual authentication and efficiency in search of the tags during the identification process.

4.5.2. Performance comparison

Although security and privacy properties are indispensable for RFID schemes, the performance of these schemes is vital to effectively use RFID systems in real applications. While our priority is proposing an authentication scheme that solves all essential security and privacy issues existing in RFID systems; we target to design an efficient scheme for practical applications. In this section, we first analyze our protocol and compare it to previous prominent ECC-based RFID authentication protocols [39, 40, 45, 46, 73, 113, 125, 129] in terms of computational and communication costs. A detailed performance comparison (including computation and communication costs) in the literature are summarized in Table 4.3.

Communication Cost Comparison. Communication cost is crucial because of determining availability delays. Increase in delays usually obstacles the effective usage of the entire system. In terms of communication cost, there are two dominant factors to determine the effects, i.e. the number of protocol rounds and communication overhead. According to our analysis, only our protocol and the inspired ID17 have two rounds. DB17 scheme has four rounds and the other protocols require three rounds to provide authentication.

Furthermore, the communication overhead of our protocol from reader-to-tag is 80 bytes (the public key and its signature), and 88 bytes (the public key and 3 blocks of AES encryption) transmitted from tag-to-reader. Hence, the total overhead of our protocol is 168 bytes.

As seen in Table 4.3, ZQ14's scheme achieves the lowest communication overhead. However, they use SHA-1 algorithm for hashing the messages but SHA-1 is cryptographically insecure [141, 142]. Their communication overhead will be greater if they prefer a secure alternative hash function in their scheme. In fact, two works [40, 143] evaluate that CH14's and ZQ14's schemes have $184-186$ bytes and $160-165$ bytes communication overhead, respectively. Therefore, we deduce that our protocol provides the minimum communication cost considering the aforementioned factors.

Table 4.3. Performance comparison

	Protocol Rounds	Comm. Overhead (B)	**Scalability**	Tag's Comp. Cost	Reader's Comp. Cost	Total Comp. Cost
LH14 [73]	3	168	$O(1)$	$5T_{ecm}$	$5T_{ecm}$	$10T_{ecm}$
Z14 [125]	3	168	$O(1)$	$5T_{ecm}$	$5T_{ecm}$	$10T_{ecm}$
C14 [113]	3	160	$O(1)$	$2T_{ecm}$	$2T_{ecm}$	$4T_{ecm}$
ZQ14 [129]	3	140	$O(1)$	$2T_{ecm}$	$2T_{ecm}$	$4T_{ecm}$
BDD17 [39]	3	>255	$O(1)$	$2T_{ecm}$	$7T_{ecm}$	$9T_{ecm}$
ID17 [40]	2	176	$O(1)$	$4T_{ecm}$	$4T_{ecm}$	$8T_{ecm}$
DB17 [45]	4	180	$O(1)$	$3T_{ecm}$	$3T_{ecm}$	$6T_{ecm}$
LZKZ18 [46]	3	>220	$O(1)$	$4T_{ecm}$	$9T_{ecm}$	$13T_{ecm}$
Our Protocol	2	168	$O(1)$	$4T_{ecm}$	$4T_{ecm}$	$8T_{ecm}$

Moreover, ECC point compression methods could be applied during sending public keys in the channel so that the communication efficiency might be increased in terms of communication overhead. For instance, our protocol can gain roughly 38 bytes in transmission and the communication overhead will be only 130 bytes in this case. However, this compression causes extra computations on both tag and reader sides. Note that, this point compression load might be delegated to only the reader since it has higher computational

capabilities.

Computational Cost Comparison. In this section, we compare the computational cost of our protocol with existing ECC-based RFID authentication protocols [39,40,45,46,73,113, 125, 129]. Table 4.5 summarizes the results in more detail. At first, to make an appropriate comparison, we will consider the primary operations which directly affects and determine the computation efficiency of an authentication protocol such as $T_{ecm}, T_{eca}, T_{inv}, T_{mul}, T_h$ and T_{aes}. Kobliz et al. [144] and Wu and Chen [145] analyze the time complexity of various operations in terms of T_{mul}. Also, these metrics are accepted by [39, 143]. Table 4.4 depicts their running time comparison of these primary operations.

Table 4.4. The running time of primary operations in terms of T_{mul} [145]

T_{mul}	Running time of a modular multiplication in $\mathbb{F}_{2^{163}}$	1
T_{add}	Running time of a modular addition in $\mathbb{F}_{2^{163}}$	negligible
T_{aes}	Running time of encrypting with AES-128	$\approx 0.15T_{mul}$
T_h	Running time of hashing with SHA (512-bit)	$\approx 0.36T_{mul}$
T_{inv}	Running time of a modular inversion in $\mathbb{F}_{2^{163}}$	$\approx 3T_{mul}$
T_{eca}	Running time of an EC point addition in $E(\mathbb{F}_{2^{163}})$	$\approx 5T_{mul}$
T_{ecm}	Running time of an EC point multiplication in $E(\mathbb{F}_{2^{163}})$	$\approx 1200T_{mul}$

We calculate the computation cost of our proposed protocol and the related works based on the above analysis in terms of T_{mul}. The tag and reader computational cost of our protocol are separately around $4T_{ecm} + 1T_{eca} + 2T_{inv} + 4T_{mul} + 2T_h + 2AES = 4817T_{mul}$, so the total cost is roughly $8T_{ecm} + 2T_{eca} + 4T_{inv} + 8T_{mul} + 4T_h + 2AES = 9634T_{mul}$. According to Table 4.5, it is clearly seen that our scheme performs an acceptable computational cost. The schemes [113, 129] have better computational efficiency, however, they have serious security and privacy issues. In fact, these results show us that EC point multiplication T_{ecm} is a dominant and decisive operation to determine the computational cost of a protocol. Hence, we claim that calculating T_{ecm} is enough for evaluating the computational cost of an ECC based authentication protocol, in general. We presented this interpretation in Table 4.3 to intelligibly demonstrate our performance comparison.

Our Implementation Environment and Results. To explore the practical usage of our proposed design, we implemented our scheme in a real-world RFID system. The overwhelming majority of works [39, 45, 73, 113, 125, 127, 129, 143], except [40], present computational cost of their protocols by referencing previous simulation results [146, 147] in their performance evaluations. Hence, a real-world implementation is valuable.

We first simulate an RFID system and run several simulations to accelerate and mature our implementation using BasicCard development environment (v8.55). Then, we run tens of realizations and take the average time of all. In the end, we obtain the results presented in Table 4.6. According to the table, our proposal uses 488 bytes as code size and 3,278

Table 4.5. Computational cost comparison

Protocols	Tag's Computations	Reader's Computations	Total Cost
LH14 [73]	$5T_{ecm}+3T_{eca}$ $\backsimeq 6015T_{mul}$	$5T_{ecm}+3T_{eca}$ $\backsimeq 6015T_{mul}$	$\backsimeq 12030T_{mul}$
Z14 [125]	$5T_{ecm}+3T_{eca}+2T_{mul}$ $\backsimeq 6017T_{mul}$	$5T_{ecm}+3T_{eca}+2T_{mul}$ $\backsimeq 6017T_{mul}$	$\backsimeq 12034T_{mul}$
C14 [113]	$2T_{ecm}+3T_{mul}+2T_h$ $\backsimeq 2403T_{mul}$	$2T_{ecm}+2T_{inv}+1T_{mul}+2T_h$ $\backsimeq 2408T_{mul}$	$\backsimeq 4811T_{mul}$
ZQ14 [129]	$2T_{ecm}+1T_{eca}+2T_h$ $\backsimeq 2405T_{mul}$	$2T_{ecm}+1T_{eca}+2T_h$ $\backsimeq 2405T_{mul}$	$\backsimeq 4810T_{mul}$
BDD17 [39]	$2T_{ecm}+1T_{eca}+3T_h$ $\backsimeq 2406T_{mul}$	$7T_{ecm}+6T_{eca}+9T_h$ $\backsimeq 8433T_{mul}$	$\backsimeq 10839T_{mul}$
ID17 [40]	$4T_{ecm}+1T_{eca}+2T_{inv}+4T_{mul}$ $+2T_h+1AES \backsimeq 4817T_{mul}$	$4T_{ecm}+1T_{eca}+2T_{inv}+4T_{mul}$ $+2T_h+1AES \backsimeq 4817T_{mul}$	$\backsimeq 9634T_{mul}$
DB17 [45]	$3T_{ecm}+5T_{mul}$ $\backsimeq 3605T_{mul}$	$3T_{ecm}+5T_{mul}$ $\backsimeq 3605T_{mul}$	$\backsimeq 7210T_{mul}$
LZKZ18 [46]	$4T_{ecm}+3T_{eca}+1T_h$ $\backsimeq 4810T_{mul}$	$9T_{ecm}+6T_{eca}+1T_h$ $\backsimeq 10819T_{mul}$	$\backsimeq 15629T_{mul}$
Our Protocol	$4T_{ecm}+1T_{eca}+2T_{inv}+4T_{mul}$ $+2T_h+1AES \backsimeq 4817T_{mul}$	$4T_{ecm}+1T_{eca}+2T_{inv}+4T_{mul}$ $+2T_h+1AES \backsimeq 4817T_{mul}$	$\backsimeq 9634T_{mul}$

bytes as data size on the reader side. Besides, it has a 567 bytes EEPROM usage and 1510 bytes RAM usage on the tag side. Also, the running time of our protocol is on average 442 ms. Actually, we realize that a remarkable amount of the time is consumed for wireless channel communication.

Table 4.6. Time-memory cost of our proposal in BasicCard environment

Code Sizes (B)	Data Sizes (B)	EEPROM Usage (B)	RAM Usage (B)	Total Running Time (ms)
488	3278	567	1510	442

At this point, we would like to emphasize that implementers might obtain different realization results because of some reasons: implementation platform and implementation approach (pipelining the algorithms in FPGA or using processors, etc.). For instance, the running time of ID17 scheme, in WISP platform, is roughly $12,742$ ms. Its FLASH/FRAM usage is $29,450$ bytes for code size and $3,296$ bytes for data size. The RAM usage is $1,595$ bytes. Thus, our implementation has better results than their WISP realization.

In our implementation, an EC point multiplication T_{ecm} takes on average 27 ms. But, this running time includes some extra operations that are used to prevent the RFID tag against side channel attacks.

On the other hand, it is obtained that T_{ecm} takes on average $1,471$ ms in the implementation of ID17 scheme [40]. The authors implement only the main components units (ECMR signature unit and ECMR recovery unit) in FPGA but they do not give any numerical results

about the running time of BDD17 scheme [39]. They just present the usage hardware resources for these units such as number of flip flops, slice registers, and LUTs. Finally, the other related papers use Gódor et al.'s [146,147] simulation results in their works. They accept that T_{ecm} takes averagely 64 ms which is also slower than our result.

5. CONCLUSIONS

In this dissertation, we firstly focus on the improper usages of RNGs in privacy-friendly RFID authentication protocols and show that misusing RNGs in an RFID protocol design might cause serious security and privacy weaknesses. To prove our claim, we first have revisited and enhanced RFID privacy and security model proposed by Vaudenay by modeling a new attack based on misusing of the RNGs. In this context, we extend the model by introducing a new RNG oracle and RANDOMEYE adversary class. Then, we apply our improved model on recently published RFID authentication protocols. We exhibit that Song-Mitchell's [43] and Akgün et al.'s [44] schemes are vulnerable to RNG attacks. In our point of view, RNGs should only be utilized to increase the security and privacy level of the protocols instead of becoming a brittle point of the scheme. It is known that a chain is only as strong as its weakest link and we point out that misusing RNGs might be the weakest link in a protocol design. Moreover, for future analysis, a completely new RFID privacy model can be constructed.

Secondly, we concentrated on both theoretical and practical aspects of ECC based RFID authentication protocols. First, we investigated vulnerabilities of the existing protocols and showed that ID17 [40], BDD17 [39], DB17 [45] and LZKZ18 [46] schemes did not provide forward and/or backward privacy. We presented our attacks against these schemes under Vaudenay's privacy model. Then, we enhanced ID17 scheme and proposed a new and practical ECC based authentication RFID protocol to efficiently satisfy all essential security and privacy properties. Thereafter, we analyzed our improved protocol in terms of security and performance perspectives. We also compared it with recent ECC-based assertive schemes and give an in-depth comparison.

Considering the practicality, we explored the realization of the existing protocols. To the best of our knowledge, the overwhelming majority of ECC based RFID protocols have not yet been implemented and tested so far in a real-world RFID system. Among the previous protocols, the conservative approach for evaluating the performance was utilizing only previous simulation results [146, 147]. Contrary to this approach, we implemented and tested our proposed protocol in ZeitControl's BasicCard environment, and presented the implementation results. Finally, we evaluated our realization outcomes especially in terms of communication and computational cost to show the performance of our proposed scheme in practice. We demonstrated that our proposed scheme had higher performance providing all common security and privacy features including backward and forward privacy rather than the ECC based RFID authentication protocols implemented in a real-world environ-

ment. Also, we believe that this work will shed light on future designs and evaluations of ECC based RFID protocol designers.

REFERENCES

[1] Bello O., Zeadally S., Badra M., Network Layer Inter-Operation of Device-to-Device Communication Technologies in Internet of Things (IoT), *Ad Hoc Networks*, 2017, **57**(C), 52-62.

[2] Eteng A.A., Abdul Rahim S.K., Leow C.Y., *RFID in the Internet of Things*, John Wiley Sons, 2018, 135-152.

[3] Daya Priyanka D., Jayaprabha T., Daya Florance D., Jayanthi A., Ajitha E., A Survey on Applications of RFID Technology, *Indian journal of Science and Technology*, 2016, **9**(2), 1-5.

[4] Finkenzeller K., *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, Wiley Publishing, 2nd ed., 2003.

[5] Zhang D., Huang H., Jo M., Future RFID Technology and Applications: Visions and Cchallenges, *Telecommunication Systems*, 2015, **58**(3), 193-194.

[6] Kardas S., Celik S., Bingöl M.A., Levi A., A New Security and Privacy Framework for RFID in Cloud Computing, *IEEE 5th International Conference on Cloud Computing Technology and Science*, 171-176, Bristol, United Kingdom, 2013.

[7] Bingöl M.A., Birinci F., Kardaş S., Kiraz M.S., Anonymous RFID Authentication for Cloud Services, *International Journal of Information Security Science*, 2012, **1**(2), 32-42.

[8] Roberti M., When RFID Becomes Obsolete, RFID Journal Blog, http://www.rfidjournal.com/articles/view?16608 (Accessed date: 24 May 2019).

[9] Want R., Schilit B.N., Jenson S., Enabling the Internet of Things, *IEEE Computer*, 2015, **48**(1), 28-35.

[10] Bilal Z., Addressing Security and Privacy Issues in Low-Cost RFID Systems, Ph.D. Thesis, Royal Holloway, University of London, London, UK, 2015.

[11] Ghaeini, H. R. and Tippenhauer, N. O., HAMIDS: Hierarchical Monitoring Intrusion Detection System for Industrial Control Systems, *Proceedings of the 2Nd ACM Workshop on Cyber-Physical Systems Security and Privacy*, New York, NY, USA, 103-111, 2016.

[12] Armknecht F., Hamann M., Mikhalev V., Lightweight Authentication Protocols on Ultra-Constrained RFIDs - Myths and Facts, *Radio Frequency Identification: Security and Privacy Issues*, 1-18, 2014.

[13] Juels A., Minimalist Cryptography for Low-Cost RFID Tags, *International Conference on Security in Communication Networks*, Amalfi, Italy, 149-164, 2004.

[14] Avoine G., Bingöl M.A., Carpent X., Kardaş S., Deploying OSK on Low-Resource Mobile Devices, *Radio Frequency Identification: Security and Privacy Issues 9th International Workshop, RFIDsec*, Graz, Austria, July 9-11, 2013

[15] Avoine G., RFID Lounge, http://www.avoine.net/rfid/index.php (Accessed date: 24 May 2019).

[16] Bilal Z., Martin K., Saeed Q., Multiple Attacks on Authentication Protocols for Low-Cost RFID Tags, *Applied Mathematics and Information Sciences*, 2014, **9**(2), 561-569.

[17] Radványi T., Biró C., Király S., Szigetváry P., Takács P., Survey of Attacking and Defending in the RFID System, *Annales Mathematicae et Informaticae*, 2015, **44**, 151-164.

[18] Alavi S.M., Baghery K., Abdolmaleki B., Security and Privacy Flaws in a Recent Authentication Protocol for EPC C1 G2 RFID Tags, *Advances in Computer Science : an International Journal*, 2014, **3**(5), 44-52.

[19] Hein D., Wolkerstorfer J., Felber N., ECC Is Ready for RFID - A Proof in Silicon, *Selected Areas in Cryptography*, Canada, August 14-15, 2008.

[20] Hutter M., Feldhofer M., Plos T., An ECDSA Processor for RFID Authentication, *Radio Frequency Identification: Security and Privacy Issues*, Istanbul, Turkey, June 8-9, 2010.

[21] Lee Y.K., Sakiyama K., Batina L., Verbauwhede I., Elliptic-Curve-Based Security Processor for RFID, *IEEE Transactions on Computers*, 2008, **57**(11), 1514-1527.

[22] Batina L., Guajardo J., Kerins T., Mentens N., Tuyls P., Verbauwhede I., Public-Key Cryptography for RFID-Tags, *International Workshop on Pervasive Computing and Communication Security - PerSec 2007*, New York, USA, 2007.

[23] Bringer J., Chabanne H., Icart T., Cryptanalysis of EC-RAC, a RFID identification protocol, M.K. Franklin, L.C.K. Hui, D.S. Wong, eds., *7th International Conference on Cryptology And Network Security - CANS'08*, Springer, Hong Kong, China, 2008.

[24] Karaoğlan Altop D., Bingöl M.A., Levi A., Savaş E., DKEM: Secure and Efficient Distributed Key Establishment Protocol for Wireless Mesh Networks, *Ad Hoc Networks*, 2017, **54**(C), 53-68.

[25] Avoine G., Cryptography in Radio Frequency Identification and Fair Exchange Protocols, Ph.D. Thesis, EPFL, Lausanne, Switzerland, 2005.

[26] Juels A., Weis S., Defining Strong Privacy for RFID, *International Conference on Pervasive Computing and Communications - PerCom 2007*, New York, 2007.

[27] Vaudenay S., On Privacy Models for RFID, *Advances in Cryptology ASIACRYPT*, Kuching, Malaysia, December 2-6, 2007

[28] Avoine G., Adversary Model for Radio Frequency Identification, Technical report, Swiss Federal Institute of Technology (EPFL), *Security and Cryptography Laboratory (LASEC)*, 2005.

[29] Avoine G., Coisel I., Martin T., Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols, *Workshop on RFID Security - RFIDSec'10*, Springer, Istanbul, Turkey, 2010

[30] Ha J., Moon S., Zhou J., Ha J., A New Formal Proof Model for RFID Location Privacy, *Proceding of the 13th European Symposium on Research in Computer Security - ESORICS 2008*, Springer, Malaga, Spain, 2008

[31] Lai J., Deng R.H., Li Y., Revisiting Unpredictability-Based RFID Privacy Models, *Proceedings of the 8th International Conference on Applied Cryptography and Network Security - ACNS 2010*, Beijing, China, 2010

[32] Akgün M., Çağlayan M., Extending An RFID Security and Privacy Model by Considering Forward Untraceability, *Security and Trust Management*, Copenhagen, Denmark, 2011.

[33] Kardaş S., Çelik S., Bingöl M.A., Kiraz M.S., Demirci H., Levi A., *k*-Strong Privacy for Radio Frequency Identification Authentication Protocols Based On Physically Unclonable Functions, *Wireless Communications and Mobile Computing*, 2014, **15**(18), 1-17.

[34] Hermans J., Peeters R., Preneel B., Proper RFID Privacy: Model and Protocols, *IEEE Transactions on Mobile Computing*, 2014, **13**(12), 2888-2902.

[35] Peinado A., Munilla J., Fúster-Sabater A., EPCGen2 Pseudorandom Number Generators: Analysis of J3Gen, *Sensors (Basel)*, 2014, **14**(4), 65006515.

[36] Meli Segu J., Garcia-Alfaro J., Herrera-Joancomart J., A Practical Implementation Attack on Weak Pseudorandom Number Generator Designs for EPC Gen2 Tags, *Wireless Personal Communications*, 2011, **59**(1), 27-42.

[37] Garcia F., de Koning Gans G., Muijrers R., van Rossum P., Verdult R., Schreur R., Jacobs B., Dismantling MIFARE Classic, *Computer Security - ESORICS 2008*, Malaga, Spain, October 6-8, 2008.

[38] Bayon P., Bossuet L., Aubert A., Fischer V., Poucheret F., Robisson B., Maurine P., Contactless Electromagnetic Active Attack on Ring Oscillator Based True Random Number Generator, *Constructive Side-Channel Analysis and Secure Design*, Darmstadt, Germany, May 3-4, 2012

[39] Benssalah M., Djeddou M., Drouiche K., A Provably Secure RFID Authentication Protocol Based on Elliptic Curve Signature with Message Recovery Suitable for m-Health Environments, *Transactions on Emerging Telecommunications Technologies*, 2017, **28**(11), e3166.

[40] Ibrahim A., Dalkılıç G., An Advanced Encryption Standard Powered Mutual Authentication Protocol Based on Elliptic Curve Cryptography for RFID, Proven on WISP, *Journal of Sensors*, 2017, **2017**, 1-10.

[41] From CoreRFID A.W.P., The Internet of Things: Practical Thoughts for Bussiness, http://www.corerfid.com/wp-content/uploads/2017/12/The-IoT-White-Paper.pdf (Accessed date: 24 May 2019).

[42] Gueulle P., Contactless A Discreet Alternative, http://www.basiccard.com/elektor-zc75rfid.pdf (Accessed date: 24 May 2019).

[43] Song B., Mitchell C.J., RFID Authentication Protocol for Low-cost Tags, *Proceedings of the 1st ACM Conference on Wireless Network Security - ACM WiSec'08*, Virginia, USA, March 31 - April 02 2008.

[44] Akgün M., Çağlayan M.U., Providing Destructive Privacy and Scalability in RFID Systems Using PUFs, *Ad Hoc Networks*, 2015, **32**, 32-42.

[45] Dinarvand N., Barati H., An Efficient and Secure RFID Authentication Protocol Using Elliptic Curve Cryptography, *Wireless Networks*, 2017, **25**(1), 415428.

[46] Liu G., Zhang H., Kong F., Zhang L., A Novel Authentication Management RFID Protocol Based on Elliptic Curve Cryptography, *Wireless Personal Communications*, 2018, **101**(3), 1445-1455.

[47] Arslan A., Kardaş S., Çolak S.A., Ertürk S., Are RNGs Achilles' Heel of RFID Security and Privacy Protocols?, *Wireless Personal Communications*, 2018, **100**(4), 1355-1375.

[48] Avoine G., Bingöl M.A., Carpent X., Yalcin S.B.O., Privacy-Friendly Authentication in RFID Systems: On Sublinear Protocols Based on Symmetric-Key Cryptography, *IEEE Transactions on Mobile Computing*, 2013, **12**(10), 2037-2049.

[49] He D., Zeadally S., An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography, *IEEE Internet of Things Journal*, 2015, **2**(1), 72-83.

[50] Ibrahim A., Dalkılıç G., Review of Different Classes of RFID Authentication Protocols, *Wireless Networks*, 2019, **25**(3), 961-974.

[51] Avoine G., Bingöl M.A., Kardaş S., Lauradoux C., Martin B., A Framework for Analyzing RFID Distance Bounding Protocols, *J. Comput. Secur.*, 2011, **19**(2), 289-317.

[52] Kardaş S., Çelik S., Arslan A., Levi A., An Efficient and Private RFID Authentication Protocol Supporting Ownership Transfer, *Lightweight Cryptography for Security and Privacy*, Gebze, Turkey, May 6-7, 2013.

[53] Menezes A.J., Vanstone S.A., Oorschot P.C.V., *Handbook of Applied Cryptography*, 1st ed., CRC Press, USA, 1996.

[54] Piramuthu S., RFID Mutual Authentication Protocols, *Decision Support Systems Elsevier*, 2010, 50(2), 387-393.

[55] Chien H.Y., SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity, *IEEE Transactions on Dependable and Secure Computing*, 2007, **4**(4), 337-340.

[56] Baashirah R., Abuzneid A., Survey On Prominent RFID Authentication Protocols for Passive Tags, *Sensors*, 2018, **18**(10), 3584.

[57] Zheng L., Xue Y., Zhang L., Zhang R., Mutual Authentication Protocol for RFID based on ECC, *IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, 21-24 July 2017.

[58] Lauter K., The Advantages of Elliptic Curve Cryptography for Wireless Security, *IEEE Wireless Communications*, 2004, **11**(1), 62-67.

[59] Van Deursen T., Radomirović S., Insider Attacks and Privacy of RFID Protocols, *Proceedings of the 8th European Conference on Public Key Infrastructures, Services, and Applications*, Leuven, Belgium, September 15-16 2011.

[60] Yih-Chun H., Perrig A., A Survey of Secure Wireless Ad Hoc Routing, *IEEE Security Privacy*, 2004, **2**(3), 28-39.

[61] K R., Hansdah R.C., Symmetric Key-Based Lightweight Authentication Protocols for RFID Security, *32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, Krakow, Poland, 16-18 May 2018.

[62] Peris-Lopez P., Hernandez-Castro J.C., Estevez-Tapiador J.M., Ribagorda A., LMAP: A Real Lightweight Mutual Authentication Protocol for Low-Cost RFID Tags, *Workshop on RFID Security - RFIDSec'06*, Graz, Austria, July 12-14 2006.

[63] Peris-Lopez P., Hernandez-Castro J.C., Estevez-Tapiador J.M., Ribagorda A., M2AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags, *International Conference on Ubiquitous Intelligence and Computing - UIC'06*, Wuhan and Three Gorges, China, 2006.

[64] Peris-Lopez P., Hernandez-Castro J.C., Estevez-Tapiador J.M., Ribagorda A., EMAP: An efficient mutual-authentication protocol for low-cost RFID tags, *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, Montpellier, France, October 29 - November 3 2006.

[65] Peris-Lopez P., Hernandez-Castro J.C., Estevez-Tapiador J.M., Ribagorda A., Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol, *Workshop on Information Security Applications - WISA'08*, Jeju Island, Korea, September 23-25 2008.

[66] EPC Global, UHF Air Interface Protocol Standard Generation2/Version2, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2 (Accessed date: 2 March 2018).

[67] Peris-Lopez P., Tong Lee L., Li T., Providing Stronger Authentication at a Low-Cost to RFID Tags Operating Under the EPCglobal Framework, *Embedded and Ubiquitous Computing - Volume 02 - EUC'08*, IEEE, IEEE Computer Society, Shanghai, China, December 17-20. 2008.

[68] Chien H.Y., Chen C.H., Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 standards, *Computer Standars & Interfaces, Elsevier*, 2007, **29**(2), 254-259.

[69] Wolkerstorfer J., Is Elliptic-Curve Cryptography Suitable to Secure RFID Tags, *ECRYPT Workshop RFID and Lightweight Crypto*, Graz, Austria, 2005.

[70] Tuyls P., Batina L., RFID-Tags for Anti-counterfeiting, *Topics in Cryptology - CT-RSA*, San Jose, USA, February 13-17 2006.

[71] Bosmans J., Roy S.S., Jarvinen K., Verbauwhede I., A Tiny Coprocessor for Elliptic Curve Cryptography over the 256-bit NIST Prime Field, *29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, Kolkata, India, January 4-8 2016.

[72] Juels A., RFID Security and Privacy: A Research Survey, *IEEE Journal on Selected Areas in Communications*, 2006, **24**(2), 381-394.

[73] Liao Y.P., Hsiao C.M., A Secure ECC-Based RFID Authentication Scheme Integrated with ID-Verifier Transfer Protocol, *Ad Hoc Networks*, 2014, **18**, 133-146.

[74] Peris-Lopez P., Hernandez-Castro J.C., Estevez-Tapiador J.M., Li T., Li Y., Vulnerability Analysis Of RFID Protocols For Tag Ownership Transfer, *Computer Networks, Elsevier*, 2010, **54**(9), 15021508.

[75] Sun H.M., Chen S.M., Wang K.H., Cryptanalysis on the RFID ACTION protocol, *Proceedings of the International Conference on Security and Management (SAM)*, Las Vegas, USA, July 16-19 2012.

[76] Bono S.C., Green M., Stubblefield A., Juels A., Rubin A.D., Szydlo M., Security Analysis of a Cryptographically-enabled RFID Device, *Proceedings of the 14th Conference on USENIX Security Symposium*, Berkeley, USA, July 31 - August 5 2005.

[77] Katz J., Lindell Y., *Introduction to Modern Cryptography*, 2nd ed., Chapman & Hall/CRC, 2014.

[78] Wachsmann C., Trusted and Privacy-preserving Embedded Systems: Advances in Design, Analysis and Application of Lightweight Privacy-preserving Authentication and Physical Security Primitives, Ph.D. Thesis, Technische Universität, Darmstadt, 2014.

[79] Lim C.H., Kwon T., Strong and Robust RFID Authentication Enabling Perfect Ownership Transfer, *Information and Communications Security*, Raleigh, USA, December 4-7 2006.

[80] Song B., Mitchell C.J., RFID Authentication Protocol for Low-Cost Tags, *Proceedings of the First ACM Conference on Wireless Network Security*, New York, USA, March 31 - April 02, 2008

[81] Phan R.C.W., Wu J., Ouafi K., Stinson D.R., Privacy Analysis of Forward and Backward Untraceable RFID Authentication Schemes, *Wireless Personal Communications*, 2011, **61**(1), 69-81.

[82] Alomair B., Poovendran R., Privacy versus Scalability in Radio Frequency Identification Systems, *Computer Communication, Elsevier*, 2010, **33**(18), 2155-2163.

[83] Hsi C.T., Lien Y.H., Chiu J.H., Chang H.K.C., Solving Scalability Problems on Secure RFID Grouping-Proof Protocol, *Wireless Personal Communications*, 2015, **84**(2), 10691088.

[84] Wu J., Stinson D.R., A Highly Scalable RFID Authentication Protocol, *Information Security and Privacy*, Brisbane, Australia, July 1-3 2009.

[85] Gill III J.T., Computational Complexity of Probabilistic Turing Machines, *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, Washington, USA, April 30 - May 02 1974.

[86] Schoenmakers B., Lecture Notes Cryptographic Protocols Version 1.32, http://www.win.tue.nl/ berry/2DMI00/LectureNotes.pdf, (Accessed date: 24 May 2019).

[87] FIPS PUB 197, Advanced Encryption Standard (AES), *U.S.Department of Commerce/National Institute of Standards and Technology*, 2001.

[88] Ferguson N., Kelsey J., Lucks S., Schneier B., Stay M., Wagner D., Whiting D., Improved Cryptanalysis of Rijndael, *Fast Software Encryption*, New York, USA, April 10-12 2000.

[89] CNSS Policy No. 15, Fact Sheet No. 1 , National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information, *Committee on National Security Systems*, 2003.

[90] Verbauwhede I., Schaumont P., Kuo H., Design and performance testing of a 2.29-GB/s Rijndael processor, *IEEE Journal of Solid-State Circuits*, 2003, **38**(3), 569-572.

[91] Daemen J., Rijmen V., *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer Science & Business Media, 2013.

[92] Dai W., Crypto++ 5.6.0 Benchmarks, https://www.cryptopp.com/benchmarks.html (Accessed date: 24 May 2019).

[93] Hankerson D., Menezes A., *Elliptic Curve Cryptography*, Springer Science Business Media LLC, 397-397, 2011.

[94] Koblitz N., Elliptic Curve Cryptosystems, *Mathematics of Computation*, 1987, **48**(177), 203-209.

[95] Miller V.S., Use of Elliptic Curves in Cryptography, *Advances in Cryptology*, 1986, (218)1, 417-426.

[96] Lauter K., The Advantages of Elliptic Curve Cryptography for Wireless Security, *IEEE Wireless Communications*, 2004, **11**, 62-67.

[97] Koblitz A.H., Koblitz N., Menezes A., Elliptic Curve Cryptography: The Serpentine Course of a Paradigm Shift, *Journal of Number theory*, 2011, **131**(5), 781-814.

[98] RFC 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, *The British Standards Institution (BSI)*, 2010.

[99] SP 800-57 Part 1 Rev. 4, Recommendation for Key Management, *National Institute of Standards and Technology NIST*, 2016.

[100] Harkanson R., Kim Y., Applications of Elliptic Curve Cryptography: A Light Introduction to Elliptic Curves and a Survey of Their Applications, *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, Tennessee, USA, April 04 - 06 2017.

[101] Ravikumar K., Udhayakumar A., Secure Multiparty Electronic Payments Using ECC Algorithm: A Comparative Study, *2014 World Congress on Computing and Communication Technologies*, Trichirappalli, India, February 27 - March 1 2014.

[102] Bingöl M.A., Biçer O., Kiraz M.S., Levi A., An Efficient 2-Party Private Function Evaluation Protocol Based on Half Gates, *The Computer Journal*, 2019, **62**(4), 598613.

[103] Biçer O., Bingöl M.A., Kiraz M.S., Levi A., Highly Efficient and Reusable Private Function Evaluation with Linear Complexity, https://eprint.iacr.org/2018/515 (Accessed date: 24 May 2019).

[104] Bingöl M.A., Efficient and Secure Schemes for Private Function Evaluation, Ph.D.Thesis, Sabanci University, Istanbul, 2019, 531586.

[105] Schindler W., Killmann W., Evaluation Criteria for True (Physical) Random Number Generators Used in Cryptographic Applications, *4th International Workshop on Cryptographic Hardware and Embedded Systems CHES*, CA, USA, August 1315 2002.

[106] hashcat, Performance, http://hashcat.net/oclhashcat/ (Accessed date: 30 August 2015).

[107] Avoine G., Dysli E., Oechslin P., Reducing Time Complexity in RFID Systems, *Selected Areas in Cryptography - SAC'05*, Kingston, Canada, August 11-12 2005.

[108] Lim C.H., Kwon T., Strong and Robust RFID Authentication Enabling Perfect Ownership Transfer, *International Conference on Information and Communications Security - ICICS'06*, NC, USA, December 4-7 2006.

[109] Van L. T., Burmester M., Medeiros B., Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange, *ACM Symposium on Information, Computer and Communications Security - ASIACCS*, SMU, Singapore, March 20-22 2007.

[110] Schnorr C.P., Efficient Identification and Signatures for Smart Cards, *Advances in Cryptology - CRYPTO' 89 Proceedings*, NY, USA, July 06 2001.

[111] Lee Y.K., Batina L., Verbauwhede I., EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID Authentication Protocol, *2008 IEEE International Conference on RFID*, NV, USA, April 16-17 2008.

[112] Okamoto T., Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes, *Advances in Cryptology - CRYPTO' 92*, CA, USA, August 1620 1992.

[113] Chou J.S., An Efficient Mutual Authentication RFID Scheme Based on Elliptic Curve Cryptography, *The Journal of Supercomputing*, 2014, **70**(1), 75-94.

[114] van Deursen T., Radomirović S., Algebraic Attacks on RFID Protocols, *Workshop on Information Security Theory and Practice - WISTP'09*, Brussels, Belgium, September 1-4 2009.

[115] Van Deursen T., Radomirović S., EC-RAC: Enriching a Capacious RFID Attack Collection, *Workshop on RFID Security - RFIDSec'10*, Istanbul, Turkey, June 8-9 2010

[116] Lee Y.K., Batina L., Singelee D., Preneel B., Verbauwhede I., Anti-counterfeiting, Untraceability and Other Security Challenges for RFID Systems: Public-Key-Based Protocols and Hardware, *Towards Hardware-Intrinsic Security Springer*, 237-257, 2010.

[117] Lv C., Li H., Ma J., Zhang Y., Vulnerability Analysis of Elliptic Curve Cryptography-Based RFID Authentication Protocols, *Transactions on Emerging Telecommunications Technologies*, 2012, **23**(7), 618-624.

[118] Lee Y.K., Batina L., Verbauwhede I., Untraceable RFID Authentication Protocols: Revision of EC-RAC, *2009 IEEE International Conference on RFID*, FL, USA, April 27-28 2009.

[119] Zhang X., Li L., Wu Y., Zhang Q., An ECDLP-Based Randomized Key RFID Authentication Protocol, *2011 International Conference on Network Computing and Information Security*, Guilin, China, May 14-15 2011.

[120] Chien H.Y., Elliptic Curve Cryptography-Based RFID Authentication Resisting Active Tracking, *Wireless Personal Communications*, 2017, **94**(4), 2925-2936.

[121] An R., Feng H., Liu Q., Li L., Three Elliptic Curve Cryptography-Based RFID Authentication Protocols for Internet of Things, *Proceedings of the 11th International Conference On Broad-Band Wireless Computing, Communication and Applications (BWCCA'16)*, Asan, Korea, November 57 2016.

[122] Moosavi S.R., Nigussie E., Virtanen S., Isoaho J., An Elliptic Curve-based Mutual Authentication Scheme for RFID Implant Systems, *Procedia Computer Science*, 2014, **32**, 198-206.

[123] He D., Kumar N., Chilamkurti N., Lee J.H., Lightweight ECC Based RFID Authentication Integrated with an ID Verifier Transfer Protocol, *Journal of Medical Systems*, 2014, **38**(10), 116.

[124] Farash M.S., Nawaz O., Mahmood K., Chaudhry S.A., Khan M.K., A Provably Secure RFID Authentication Protocol Based on Elliptic Curve for Healthcare Environments, *Journal of Medical Systems*, 2016, **40**(7), 165.

[125] Zhao Z., A Secure RFID Authentication Protocol for Healthcare Environments Using Elliptic Curve Cryptosystem, *Journal of Medical Systems*, 2014, **38**(5), 46.

[126] Peeters R., Hermans J., Attack on Liao and Hsiao's Secure ECC-based RFID Authentication Scheme integrated with ID-Verifier Transfer Protocol, Cryptology ePrint Archive, https://eprint.iacr.org/2013/399 (Accessed date: 24 May 2019).

[127] Alexander P., Baashirah R., Abuzneid A., Comparison and Feasibility of Various RFID Authentication Methods Using ECC, *Sensors*, 2018, **18**(9), 2902.

[128] Farash M.S., Cryptanalysis and Improvement of an Efficient Mutual Authentication RFID Scheme Based on Elliptic Curve Cryptography, *J. Supercomput.*, 2014, **70**(2), 987-1001.

[129] Zhang Z., Qi Q., An Efficient RFID Authentication Protocol to Enhance Patient Medication Safety Using Elliptic Curve Cryptography, *Journal of Medical Systems*, 2014, **38**(5), 47.

[130] Jin C., Xu C., Zhang X., Zhao J., A Secure RFID Mutual Authentication Protocol for Healthcare Environments Using Elliptic Curve Cryptography, *Journal of Medical Systems*, 2015, **39**(3), 24.

[131] Jin C., Xu C., Zhang X., Li F., A Secure ECC-based RFID Mutual Authentication Protocol to Enhance Patient Medication Safety, *J. Med. Syst.*, 2016, **40**(1), 1-6.

[132] Peris-Lopez P., Hernandez-Castro J.C., Estevez-Tapiador J.M., Ribagorda A., LAMED - A PRNG for EPC Class-1 Generation-2 RFID specification , *Computer Standards and Interfaces*, 2009, **31**(1), 88-97.

[133] Meli Segu J., Garcia-Alfaro J., Herrera-Joancomart J., J3Gen: A PRNG for Low-Cost Passive RFID, *Sensors*, 2013, **13**(3), 3816-3830.

[134] Garcia-Alfaro J., Herrera-Joancomart J., Meli Segu J., Remarks on Peinado et al.al's Analysis of J3Gen, *Sensors*, 2015, **15**(3), 6217-6220.

[135] Che W., Deng H., Tan W., Wang J., A Random Number Generator for Application in RFID Tags, *Networked RFID Systems and Lightweight Cryptography*, Springer, Berlin, 279-287, 2008.

[136] ISO/IEC Standard 18000 RFID Air Interface Standard, http://www.hightechaid.com/standards/18000.htm (Accessed date: 24 May 2019).

[137] Sarma S., Weis S., Engels D., RFID Systems and Security and Privacy Implications, *Cryptographic Hardware and Embedded Systems - CHES'02*, CA, USA, August 1315 2002.

[138] Barak B., Shaltiel R., Tromer E., *True Random Number Generators Secure in a Changing Environment*, Cologne, Germany, September 0810 2003.

[139] Song B., Mitchell C.J., RFID Authentication Protocol For Low-cost Tags, *ACM Conference on Wireless Network Security*, VA, USA, March 31 - April 02 2008.

[140] Guilfoyle T., BasicCard Developer Manual V8.15, http://www.basiccard.com/download.htm (Accessed date: May 2019).

[141] Stevens M., Bursztein E., Karpman P., Albertini A., Markov Y., The First Collision for Full SHA-1, Cryptology ePrint Archive, https://eprint.iacr.org/2017/190 (Accessed date: 24 May 2019).

[142] Wang X., Yin Y.L., Yu H., Finding Collisions in the Full SHA-1, *Advances in Cryptology - CRYPTO'05*, CA, USA, August 14-18 2005.

[143] He D., Zeadally S., An Analysis of RFID Authentication Schemes for Internet of Things in Healthcare Environment Using Elliptic Curve Cryptography, *IEEE Internet of Things Journal*, 2015, **2**(1), 72-83.

[144] Koblitz N., Menezes A., Vanstone S., The State of Elliptic Curve Cryptography, *Designs Codes and Cryptography*, 2000, **19**(2-3), 173-193.

[145] Wu S., Chen K., An Efficient Key-Management Scheme for Hierarchical Access Control in E-Medicine System, *Journal of Medical Systems*, 2012, **36**(4), 2325-2337.

[146] Gódor G., Giczi N., Imre S., Elliptic Curve Cryptography Based Mutual Authentication Protocol for Low Computational Capacity RFID systems - Performance Analysis by Simulations, *IEEE International Conference on Wireless Communications, Networking and Information Security*, Beijing, China, June 25-27 2010.

[147] Gódor G., Imre S., Elliptic Curve Cryptography Based Authentication Protocol for Low-Cost RFID tags, *IEEE International Conference on RFID-Technologies and Applications*, Sitges, Spain, September 15-16 2011.

PUBLICATIONS

Arslan A., Çolak S.A., Ertürk S., A Secure and Privacy Friendly ECC Based RFID Authentication Protocol for Practical Applications, *Wireless Personal Communications*, (under reviewing)

Arslan A., Kardaş S., Çolak S.A., Ertürk S., Are RNGs Achilles' Heel of RFID Security and Privacy Protocols?, *Wireless Personal Communications*, 2018, **100**(4), 1355-1375.

Arslan A., Çelik S., Kartal M.: The Performance Bounds Of Protocols In Wireless Channel For ISO 18000-3 Standard, *International Conference on Software, Telecommunications and Computer Networks (SoftCom2012)*, Split, Croatia, September 11-13 2012.

Eksim A., **Arslan A.**, Celik S., Kartal M.: Performance Improvement of Multiple-Antenna RFID Tags Using Limited Feedback Schemes In ISO 18000-7 Standard, *International Conference on Software, Telecommunications and Computer Networks (SoftCom2012)*, Split, Croatia, September 11-13 2012.

Kardaş S., Çelik S., **Arslan A.**, Levi A. An Efficient and Private RFID Authentication Protocol Supporting Ownership Transfer, *Lightweight Cryptography for Security and Privacy. LightSec 2013*, Gebze, Turkey, May 6-7 2013.

Gurel A., **Arslan A.**, Akgun M., None-Uniform Stepping Approach to RFID Distance Bounding Problem, *Data Privacy Management (DPM 2010)*, Athens, Greece, September 23 2010.

CURRICULUM VITAE

Atakan ARSLAN was born in Eskişehir in 1986. He completed his primary and secondary education there in 2000. He is graduated from TEV İnanç Türkeş Private High School (TEVİTÖL) in 2004. He received the B.Sc. and M.Sc. degrees in Control Engineering and Telecommunication Engineering from the Faculty of Electrical and Electronics Eng., Istanbul Technical University, Turkey in 2008 and 2013, respectively. He also received the B.Sc. in Business Administration from Anadolu University in 2012.