ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

SECURITY ANALYSIS OF RFID AUTHENTICATION PROTOCOLS BASED ON SYMMETRIC CRYPTOGRAPHY AND IMPLEMENTATION OF A FORWARD PRIVATE SCHEME

M.Sc. THESIS

Muhammed Ali BİNGÖL

Department of Electronics and Communications Engineering

Telecommunications Engineering Programme

JANUARY 2012

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

SECURITY ANALYSIS OF RFID AUTHENTICATION PROTOCOLS BASED ON SYMMETRIC CRYPTOGRAPHY AND IMPLEMENTATION OF A FORWARD PRIVATE SCHEME

M.Sc. THESIS

Muhammed Ali BİNGÖL
(504081352)

Department of Electronics and Communications Engineering

Telecommunications Engineering Programme

Thesis Supervisor: Doç. Dr. S. Berna ÖRS YALÇIN

JANUARY 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

SİMETRİK KRİPTOGRAFİ TABANLI RFID PROTOKOLLERİNİN GÜVENLİK ANALİZİ VE İLERİ MAHREMİYETLİ BİR TASARININ GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ

Muhammed Ali BİNGÖL
(504081352)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Telekomünikasyon Mühendisliği Programı

Tez Danışmanı: Doç. Dr. S. Berna ÖRS YALÇIN

OCAK 2012

Muhammed Ali BİNGÖL, a M.Sc. student of ITU **Graduate School of Science, Engineering and Technology** student ID **504081352**, successfully defended the thesis entitled **"SECURITY ANALYSIS OF RFID AUTHENTICATION PROTOCOLS BASED ON SYMMETRIC CRYPTOGRAPHY AND IMPLEMENTATION OF A FORWARD PRIVATE SCHEME"**, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Doç. Dr. S. Berna ÖRS YALÇIN** ....
İstanbul Technical University

Jury Members : **Prof. Dr. İbrahim ALTUNBAŞ** ....
İstanbul Technical University

Yard. Doç. Dr. Selçuk BAKTIR ....
Bahçeşehir University

Date of Submission : **19 December 2011**
Date of Defense : **24 January 2012**

Aileme ve Burcu’ma,

FOREWORD

I would like to thank all people who have helped and inspired me during my M.Sc. study. I especially want to thank my supervisor S. Berna Örs Yalçın for her guidance and endless support during this thesis. I thank profoundly Gildas Avoine, the person who host me several times to Université catholique de Louvain and who gave me the resources to succeed. To me, his great heart and his scientific rigor served as an example and allowed me to progress during the periods I spent with him. I am happy to have such a supportive co-supervisor. I enjoyed his interest in my research as well as the fruitful discussions. I would like to sincerely thank to my thesis jury İbrahim Altunbaş and Selçuk Baktır for examining this thesis and providing me helpful comments. I also would like to thank my friend (a really good friend), Süleyman Kardaş, for his uncountable helps and contributions to this thesis. My deeply thanks to Atakan Arslan for his help while working on ZeitControl cardsystems and his great friendship over the years. Also I would like to thank to Mehmet Sabır Kiraz who always gave me the encouragement that I need. Also many thanks go to Xavier Carpent for the brainstorming discussions and valuable contributions to this work. Also thanks to Benjamin Martin and Tania Martin for their grate hospitality and helps during my stays in UCL. It was a pleasure for me to work with all the wonderful people in UCL.

My deepest gratitude goes to my family for their unflagging love and support throughout my life; this thesis is simply impossible without them.

Last but not least I would like to thank to all my colleagues in TUBITAK UEKAE for their support and strong friendship.

January 2012 Muhammed Ali BİNGÖL

TABLE OF CONTENTS

Page

ABBREVIATIONS

BF	**:** Bloom Filters
CRC	**:** Cyclic Redundancy Code
CTI	**:** Constant-Time Identification
DoS	**:** Denial of Service
EEPROM	**:** Electrically Erasable Read Only Memory
EPC	**:** Electronic Product Code
FIPS	**:** Federal Information Processing Standards
ISO	**:** International Organization for Standardization
LSB	**:** Least Significant Bit
MSB	**:** Most Significant Bit
PKC	**:** Public-Key Cryptography
PRF	**:** Pseudo-Random Function
PRNG	**:** Pseudo-Random Number Generator
RFID	**:** Radio Frequency IDentification
SEC	**:** Securities and Exchange Commission
TMTO	**:** Time-Memory Trade-off

LIST OF TABLES

LIST OF FIGURES

Page

SECURITY ANALYSIS OF RFID AUTHENTICATION PROTOCOLS BASED ON SYMMETRIC CRYPTOGRAPHY AND IMPLEMENTATION OF A FORWARD PRIVATE SCHEME

SUMMARY

This M.Sc. thesis is mainly two folds: First part includes the the theoretical security analysis of privacy-friendly radio frequency identification (RFID) protocols that are based on symmetric cryptography and sub-linear complexity. Second part is a practical part dedicated to an implementation of forward secure RFID authentication protocol.

RFID technology provides wireless communication with an object or someone to identify or authenticate by using radio waves with neither physical nor visual contact. RFID is one of the most promising technologies deployed in many applications such as contactless payment systems, public transportation, electronic passports, access cards, logistic tracking systems etc. In fact RFID has entered in our lives, however, security and privacy concerns have become controversial as a social demand. Moreover, the cost of RFID tags is an other obstacle to technological advance. Many works have been dedicated to this specific area to mitigate these issues. The large body of literature RFID Security and Privacy demonstrates that designing a privacy friendly and efficient protocol is still a challenging task and finding the appropriate one is quite awful for industrials. Indeed, although many protocols have been proposed over the years, none can be deemed as ideal. Motivated by this need, in this work we examine most of the proposals in the field, categorize them according to common features analyze them, compare their properties and discuss about which can be considered as the best ones to date. We also provide new attacks on several of these protocols and some patches.

First, this work includes a comprehensive analysis of privacy-friendly authentication protocols devoted to RFID that: (i) are based on well-established symmetric-key cryptographic building blocks; (ii) require a reader complexity lower than $O(N)$ where N is the number of provers in the system. These two properties are *sine qua non* conditions for deploying privacy-friendly authentication protocols in large-scale applications, e.g., access control in mass transportation. We describe existing protocols fulfilling these requirements and point out their drawbacks and weaknesses. We especially introduce new attacks and raise that some protocols are not resistant to timing attacks. We also suggest a number of new solutions to ameliorate some of the existing protocols and provide guidelines for those schemes. We have extensively evaluated and compared all the candidates according to their security, and performance. The security properties that we investigated include user privacy and as well as forward privacy, impersonation resiliency and desynchronization resistance. Furthermore, we examined thoroughly their performance, in terms of computational and storage cost. According to our analysis by means of security and efficiency, we selected the most appropriate candidates for practical uses.

Second, this thesis includes an implementation of a real RFID system which is efficient and secure with respect to the first part of this work. We implemented one of the best candidate that is, according to our analysis and criteria, the most appropriate one for practical uses. To the best of our knowledge, this is the first complete implementation of a forward-private RFID system based on time-memory trade-off. This method is already introduced but never tried to implemented in a real RFID system. We show that our implementation practically allows achieving a high performance by means of search complexity and memory usage without degrading privacy. We have run several experiments on the implemented real RFID system and we observed that the experimental outputs are very close to the theoretical bounds. Finally, the authentication speed and effective memory usage put forth that this forward-private RFID system is ready to be used for practical proposes.

SİMETRİK KRİPTOGRAFİ TABANLI RFID PROTOKOLLERİNİN GÜVENLİK ANALİZİ VE İLERİ MAHREMİYETLİ BİR TASARININ GERÇEKLENMESİ

ÖZET

Bu Yüksek Lisans tezi genel olarak iki kısımdan oluşmaktadır: Birinci kısımda "simetrik anahtarlı kriptografik sistem" tabanlı RFID (radyo frekansı tanımlama) protokollerinin teorik güvenlik ve mahremiyet analizi ele alınmıştır. İkinci kısımda ise ileri mahremiyet sağlayan bir RFID kimlik doğrulama protokolünün "zaman bellek ödünleşim" metodu kullanılarak gerçeklenmesi yapılmış ve sonuçları teorik sonuçlar ile karşılaştırılmıştır. Aşağıda öncelikle RFID teknolojisi hakkında kısa bilgiler verilerek bu konudaki güvenlik ve mahremiyet gereksinimlerine değinilmiş daha sonra bu tezdeki yapılan çalışmalar özetlenmiştir.

RFID teknolojisi, fiziksel temesa gerek olmaksızın radyo dalgalarıyla etiket taşıyan bir nesne ya da kişinin kimliğinin belirlenmesini veya doğrulanmasını sağlar. RFID sistemi temel olarak etiket (tag), okuyucu (reader) ve etiket hakkında bilgileri güvenli bir şekilde depolayan veri tabanı sunucusundan (back-end server) oluşmaktadır. RFID etiketi, okuyucudan gelen sorguları almaya ve cevaplamaya olanak tanıyan bir silikon yonga, anten ve kaplamadan meydana gelir. Yonga, etiketin üzerinde bulunduğu nesne ile ilgili bilgileri saklar. Anten, radyo frekansı kullanarak kimlik bilgilerini okuyucuya iletir. Kaplama ise etiketin bir nesne üzerine yerleştirilebilmesi için yonga ve anteni çevreler. Hafıza, okuma mesafesi, okuma/yazma kapasitesine göre farklılıklar göstermektedir. Etiketler okuma sırasında kullanılan frekans aralığına bağlı olarak da LF, HF, UHF ve mikrodalga frekans olmak üzere çeşitlendirilebilirler.

RFID ilk defa ikinci dünya savaşında dost savaş uçaklarını düşman savaş uçaklarından ayırmak için geliştirilmiş ve kullanılmış bir teknolojidir. Günümüzde ise çok geniş bir kullanım alanı vardır. RFID teknolojisi temassız ödeme, toplu taşıma, elektronik pasaport, giriş kontrol sistemleri, lojistik takip sistemleri, kütüphaneler, taşıt otomatik geçiş sistemleri, otomatik tanıma ve bilgi toplama sistemleri gibi birçok alanda yaygın olarak uygulanmış ve ileride de daha birçok alanda gelecek vaat eden bir teknolojidir.

Hayatımızın hemen hemen her alanına giren bu yeni teknoloji güvenlik ve kullanıcı mahremiyeti gibi toplumsal endişeleri de beraberinde getirmiştir. Bu teknoloji gün geçtikçe önem kazanıp üzerinde yapılan çalışmaların da arttırılmasına ragmen gizlilik ve güvenlik ile ilgili sorunları tam olarak çözülememiştir. İnsan mahremiyetinin ihlal edilmesi konusunda oluşan çekincelerle Eylül 2003'de bazı insan hakları ve sivil toplum organizasyonları RFID teknolojisi kullanan marketleri dava etmiştir. Bu sistemin kötü yollar için kullanılabilecegi öne sürülmüstür. RFID teknolojisinin bir parçası olan RFID etiketlerin her biri yalnızca kendine özgü ve ait olduğu kişiye yönelik bilgiler taşımaktadır. Bu durumda bu etiketleri taşıyan kişiler de adeta bu aygıtlarla birlikte etiketlenmiş olmaktadırlar. Ayrıca, yaygın olarak kullanılan RFID etiketleri sorgulandıklarından haberi olmayan ve her türlü sorguya yanıt veren

yapıdadırlar. Bunun sonucu olarak RFID etiketlerini taşıyan kişilerin habersiz olarak izlenmesi, özel hayatları hakkında istemedikleri bilgilerin ortaya dökülmesi durumu ortaya çıkmaktadır.

RFID teknolojisinin insanların özel yaşamlarının gizliliğine karşı oluşturduğu tehditler önemli bir sorun olmakla beraber asıl büyük tehdit ve problem bu sistemlerin kontrolünün, protokoldeki güvenlik açıklarından ve teknik savunma zafiyetlerinden istifade edilerek istenmeyen kişiler tarafından elde edilmesi sonucu ortaya çıkmaktadır. Çünkü yukarıda bahsedilen kullanım alanlarında da görüldüğü gibi RFID sistemleri artık insan hayatının önemli bir parçasını oluşturmakta ve insanlar kendileri için çok büyük önem taşıyan faaliyetlerini (ödemeler, sahip oldukları mülklerin korunması, kimlik denetim sistemleri ile kendilerini tanıtmaları vb.) bu sistemler üzerinden gerçekleştirmektedirler.

Tüm bu süreç esnasında ise RFID sistemlerinin güvenli olduğunu varsayarak hareket etmektedirler. Bir RFID sisteminin güvenliği sistemi oluşturan bileşenlerin (etiket, okuyucu ve veri tabanı) parçanın da güvenli olması ile doğrudan ilgilidir. RFID etiketlerinin, özellikle de daha yaygın kullanılan pasif RFID etiketlerinin devre alanı ve enerji tüketimi gibi kaynaklarının kısıtlı olduğu göz önünde bulundurulduğunda, bu cihazlarda mevcut kriptografik algoritmaları kullanarak güvenlik sağlamanın zorluğu ortadadır. RFID etiketlerinin düşük enerji tüketimi ile etkin çalışmasını sağlamak bu çalışmanın ortaya konulmasının başlıca hedeflerindendir.

Bu ihtiyaçlardan dolayı son zamanlarda akademik ve endustriyel çalışmalar bu özel alanda ortaya konulmuştur. Literatürdeki RFID güvenlik ve gizlilik çalışmalarının büyük çoğunluğu, hem kullanıcı mahremiyetini sağlayacak hem de verimli olacak bir protokol dizaynının oldukça zor olduğu konusunda ortak fikirdedirler. Bu kadar geniş ve hızlı değişen literatürde uygun protokolün seçilmesi işi sanayiciler için de bir problemdir. Gerçekten de yıllar boyunca önerilen birçok protokol olmasına rağmen, hiçbiri ideal olarak kabul edilmemiştir. Bu çalışmada bahsedilen ihtiyaçlardan yola çıkarak, bu alanda önerilen protokoller güvenlik ve verimlilik ortak özelliklerine göre kategorize edilmiş ve kendi aralarında karşılaştırılmıştır.

Bu çalışma ilk olarak mahremiyet özelliğini sağlayan protokollerin kapsamlı analizini içerir. Bu çalışmada ele alınan RFID kimlik doğrulama protokolleri şu iki özelliği taşımaktadır: (i) Simetrik kriptografi yapı taşları ile oluşturulmuş olması, (ii) N sistemdeki etiket sayısı olmak üzere $O(N)$'den daha düşük karmaşıklık ile kimlik doğrulama işlemlerini yapabilmesi. Büyük ölçekli gerçek hayattaki uygulamalar (örn. toplu taşıma vb.) göz önüne alındığında bu iki koşul RFID sisteminin taşıması gereken olmazsa olmaz özelliklerindendir. Bu çalışmada bu özellikleri sağlayan protokoller ele alınmış ve bunların kapsamlı olarak teorik güvenlik analizler yapılmış, eksiklikleri ve zayıf noktaları ortaya konulmuştur. Bu protokoller üzerine yeni kriptografik ataklar yapılmış, özellikle zamanlama ataklarının birçok protokol üzerinde nasıl gerçekleştirilebileceği bu çalışma ile ortaya konulmuştur. Ayrıca bazı mevcut protokolleri iyileştirmek için çözüm önerileri sunulmuş ve bu protokoller için bazı kılavuz bilgiler verilmiştir. Tüm aday protokolleri güvenlik ve performans kriterleri değerlendirilerek karşılaştırılmıştır. Protokollerin güvenlik olarak; kullanıcı mahremiyeti, taklit edilmeye karşı dayanıklılık, desenkronize edilip takip edilmeye karşı dayanıklılık ve iz sürülme tehlikesine karşı güvenirlik özellikleri ele alınmıştır.

Performans olarak, etiket ve veri tabanı üzerinde az işlem yapma ve düşük yer kaplama kriterleri göz önüne alınmıştır. Böylece ortaya konular çalışmalar sonucunda pratik dünyada kullanılabilecek en uygun adaylar seçilmiştir.

Bu çalışmada ikinci olarak, güvenliği ve performansı birinci bölümde değerlendirilen en uygun protokolün gerçek bir RFID sistemi üzerinde gerçeklenmesi yapılmıştır. Bildiğimiz kadarıyla zaman-hafıza ödünleşim metoduna dayalı ve ileri mahremiyet özelliği taşıyan ilk RFID sisteminin gerçeklenmesi bu çalışma ile ortaya konulmuştur. Daha önce teorik olarak tasarlanan bu sistemin şimdiye kadar gerçeklenmesi yapılmamıştı. Bu gerçeklemenin mahremiyet özellikler korunarak yüksek veri tabanı arama hızı ve düşük bellek kullanılarak yüksek performans sağladığı yapılan deneyler ile ortaya konulmuştur. Ayrıca deney sonuçlarının teorik sınırlara yakın olması bu çalışmanın doğruluğunu ve olumlu etkisini göstermektedir. Sonuç olarak bu çalışmanın pratik olarak da kullanılabilecek hazır bir sistem olduğu ortaya konulmuştur.

1. INTRODUCTION

In this chapter, we present essential concepts of this study and introduce a few basic definitions. These definitions will be useful to support ideas of the later chapters. First of all, we briefly introduce the background of RFID systems then we give some security and privacy definitions which we will establish other subsections over these basic informations. After that, we describe the major contributions of this thesis.

1.1 RFID Systems

Radio Frequency IDdentification (RFID) is a pervasive technology deployed in many applications to identify or authenticate objects and subjects with neither physical nor visual contact [4]. An RFID system usually consists of tags, i.e., a microcircuit with an antenna, carried by the object or subject, some readers that allow to remotely query the tags, and a back-end system [5]. Figure 1.1 illustrates a typical RFID system. It is generally assumed that the communication channel between the reader and its back-end database is secure while the channel between a reader and a tag is wireless and insecure.

A common idea is that an RFID tag is just a transponder that backscatters a unique identifier, used for supply chains, libraries, and pet identification. An RFID tag can actually do much more than simply backscattering an identifier, and it is even tricky to define the limits between RFID and the other evolved pervasive technologies. Consequently, we describe precisely in Sect. 2.1 the capabilities we confer to the tags in this paper.

The fact that no contact is needed to read an RFID tag allows to use it where traditional smartcards are not invited: pet identification, electronic passports, but also access control for ski lifts, ... RFID also brings advantages in access control applications by speeding up the flow of customers, typically in mass transportation. Such a kind

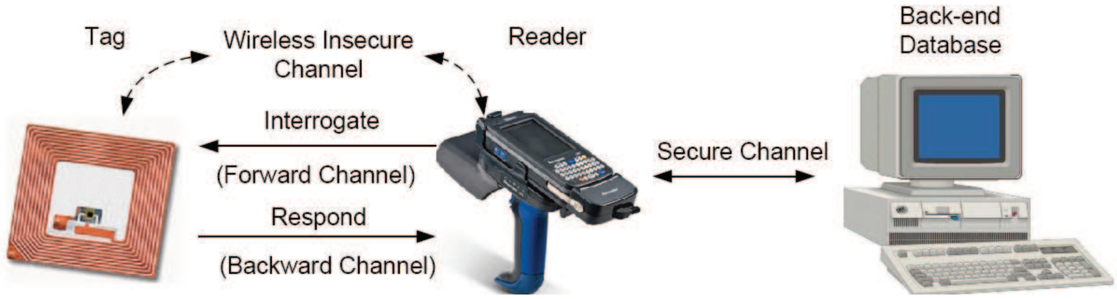


Figure 1.1: An RFID System [1]

of application requires authentication protocols that scale well when there is a large number of tags registered to the system.

While RFID has existed for several decades, it is its recent wide-spread that made privacy a major concern for everyone. Authorities are aware of the privacy issues and react accordingly. For example, in its recommendation SEC(2009) 585/586 [6] about RFID, the European Commission states: "Because of its potential to be both ubiquitous and practically invisible, particular attention to privacy and data protection issues is required in the deployment of RFID. Consequently, privacy and information security features should be built into RFID applications before their widespread use (principle of security and privacy-by-design)" [6]. Similar recommendations also araised in North America [7–9]. Fulfilling this recommendation may be partially done by designing authentication or identification protocols that ensure privacy against an external adversary. Among privacy, one may distinguish information leakage where the tag or the back-end reveals some personal information, from illicit tracking that consists in tracking a tag and so its holder.

1.2 Security Primitives

In this section some security primitives that will be used through this thesis are briefly introduced.

1.2.1 Authentication vs. identification

Authentication is a well-known terminology in cryptography and already defined in many classical textbooks. Definition 1.3 is excerpted from [10].

Definition 1.3 (Authentication). *An authentication is a process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated (*i.e., *is active at, or immediately prior to, the time evidence is acquired).*

Here the goal of corroborative evidence is that to provide a proposition that is already supported by some initial evidence e.g., something known, something possessed or something inherent (to a human individual). Moreover, mutual authentication refers to the provision of entity authentication for both parties [11].

On the other hand, the term *identification* is used to simply refer to the process of claiming or stating an identity without providing the corroborating evidence required for entity authentication [11].

1.3.1 Forward Privacy

The term *privacy* can be defined as the ability of an entity to keep or control the confidentiality of identifying information and other personal information when desired [11].

In RFID context, *forward privacy*[1] is the property that guarantees the security of past interactions of a tag even when it is compromised at a later stage. Namely, the secret information of a tag $\mathcal{T}_i$ is corrupted by an adversary at time τ, the adversary can not associate any transactions with $\mathcal{T}_i$ at time τ', where $\tau' < \tau$. Therefore, past communications cannot be jeopardized by an adversary [12, 13].

1.3.2 Cryptographic protocol

In many papers related to cryptography, the term 'protocol' is generally used as a shorthand expression for 'cryptographic protocol'. A cryptographic protocol is a distributed algorithm describing precisely the interactions of two or more entities to achieve certain security objectives. The entities interact with each other by exchanging messages over private and/or public communication channels [10, 11]. In RFID

[1] It is also known as *backward untraceability*.

protocols these two interacting entities are mostly an RFID reader (verifier) and a RFID tag (prover).

1.3.3 Symmetric-key cryptosystem

Symmetric-key cryptosystem is a type of cryptographic operation system in which the same key is used for each of the encryption and decryption operations in the cryptosystem. The key shared by the parties (or a share can be computed trivially one from another) is called *symmetric-key* used for both encryption and decryption. The symmetric-key is typically kept secret by the parties [11]. The encryption and decryption algorithm A formal description is given in Definition 1.4 is excerpted from [10].

Definition 1.4 (Symmetric-key cryptosystem)**.** *Consider an encryption scheme consisting of the sets of encryption and decryption transformations* $\{E_e : e \in \mathcal{K}\}$ *and* $\{D_d : d \in \mathcal{K}\}$*, respectively, where* $\mathcal{K}$ *is the key space. The encryption algorithm is said to be symmetric-key if for each associated encryption/decryption key pair* $(e;d)$*, it is computationally 'easy' to determine d knowing only e, and to determine e from d.*

1.4.1 One-way functions

Many modern cryptographic applications rest on the use of *one-way functions*. Securities of many authentication mechanisms and various other cryptographic protocols depend on the hardness of inverting one-way functions which they are based on. The definition of one-way functions is as follows [10].

Definition 1.5 (One-way functions)**.** *A function* $H : A \to B$ *where* A *and* B *are non-empty finite sets is called one-way if for any* $a \in A$*, there exists a polynomial time algorithm to compute* $b = H(a)$*, but for an arbitrary* $b \in B$ *a polynomial time algorithm to find* $a = H^{-1}(b)$ *does not exist.*

In a one-way function it is easy to compute the image $b = H(a)$ but for a given b, the preimage a should be hard to compute. Here 'easy' and 'hard' are to be understood in the theory of polynomial time problems in the area of computational complexity theory [14]. Inverting these functions is equivalent to breaking the cryptosystem that depends on them.

1.6 Main Contributions

The summary of the contributions of this thesis can be stated in two main points as:

The large body of literature RFID Security and Privacy [15] demonstrates that designing a privacy-friendly protocol is still a challenging task and finding the appropriate one is quite awful for industrials. Indeed, although many protocols have been proposed over the years, none can be deemed as ideal. In this work, first, we examine most of the proposals in the field, categorize them according to common features, analyze them, compare their properties and discuss about which can be considered as the best ones to date. We also provide new attacks on several of these protocols and some patches. The security properties that we investigated include user privacy and as well as forward privacy, impersonation resiliency and desynchronization resistance. Furthermore, we examined thoroughly their performance, in terms of computational and storage cost.

Second, we implement the best candidate that is, according to our criteria, the most appropriate one for practical uses. To the best of our knowledge, this is the first complete implementation of a forward-private RFID system based on time-memory trade-off. It is shown that our implementation practically allows achieving a high performance by means of search complexity and memory usage without degrading the user privacy. Moreover, we run several experiments on the implemented real RFID system to show that it is close to theoretical results and ready to use in real-life applications.

1.7 Organization of the Thesis

The remaining body of the thesis is organized as follows.

Chapter 2 provides a comprehensive analysis of privacy-friendly authentication protocols devoted to RFID that are based on symmetric cryptography and has a sub-linear online search complexity on server side. This chapter includes attacks, improvements and some guidelines for the existing RFID protocols that are conforming to the pre-defined criteria. Also this chapter compares the most valuable protocols and provides a summary of their properties and performances. Our analysis finally yields the best protocols in terms of security, privacy, forward-privacy, desynchronization, reader and tag complexity, and memory.

In Chapter 3 we briefly recall the required background on Time-Memory Trade-off method. After that we describe the TMTO technique as well as the idea and results of perfect tables.

Chapter 4 demonstrates the implementation of the selected protocol (OSK/AO) based on time-memory trade-offs which provides forward private security. Also it gives the results for some specific settings on real RFID environment.

Chapter 5 presents conclusions and describes possible directions for future work.

2. SECURITY ANALYSIS OF SUB-LINEAR RFID PROTOCOLS BASED ON SYMMETRIC-KEY CRYPTOGRAPHY

In this Chapter, we provide a comprehensive analysis of privacy-friendly authentication protocols devoted to RFID that are based on symmetric cryptography and has a complexity lover than $O(N)$ where N is the number of provers in the system.

We describe existing protocols fulfilling these requirements and point out their drawbacks and weaknesses. We especially introduce attacks on CHT [2], CTI [16], YA-TRAP* [17, 18], and the variant of OSK/AO [19] with mutual authentication. We also raise that some protocols, such as O-RAP [20], O-FRAP [21] and OSK/BF [22] are not resistant to timing attacks. Finally, we select some candidates that are, according to our criteria, the most appropriate ones for practical uses.

We give in Sect. 2.1 the criteria we used to thoroughly select the protocols we analyze in this paper. We then categorize the selected protocols in three sections: protocols with shared secrets (Sect. 2.2), protocols using hash-chains (Sect. 2.3), and counter-based protocols (Sect. 2.4). We compare the most valuable protocols in Sect. 2.6 and provide a summary of their properties and performances in Tab. 2.3. Our analysis finally yields the best protocols in terms of security, privacy, forward-privacy, desynchronization, reader and tag complexity, and memory.

2.1 Protocol Selection Criteria

In this section, we list some characteristics that we consider relevant for the protocols to have in the problem at hand. We discard in the rest of our analysis all the protocols that do not meet these criteria. We emphasize that these characteristics do not form a partition, but should cover all existing solutions in RFID authentication, up to our knowledge.

In the following, we consider that the communication between tags and readers is insecure, meaning that it can be easily eavesdropped on, interrupted or modified on the fly by an external entity. However, the communication between readers and the database is secure, and we will in general refer to these two entities as a single one, since it makes no difference for an attacker. This is compliant with the model of Juels and Weis. We consider RFID tags as not tamper resistant. Therefore, solutions in which each tag has the same key, for instance, are discarded.

2.1.1 Time complexity of identification

The ISO-9798 defines challenge-response authentication protocols, which are commonly used in RFID. These are used in the MIFARE Classic for instance. Other standards are also in application, such as the ISO-11770, used for example in the Basic Access Control of e-passports.

To authenticate a tag, a reader first has to identify it, in order to determine its key. The two naive approaches to do that is either letting the tag send its identifier in the clear (but that eliminates any privacy in the system), or to let the reader "guess" the tag with which it is communicating. However, this latter solution takes $O(N)$ cryptographic operations (where N is the number of tags in the database), which is inefficient in large systems[1]. Note that linear complexity may be fine in some settings, for instance with car ignition, consisting of only one reader and one tag (the car "key"), and where identification is implicit. However, the problem remains important for most reasonably-sized systems, and we therefore restrict our analysis to protocols designed to reduce the complexity of the identification.

Some other proposals with linear reader complexity deserve to be mentioned. Some researchers try to design authentication protocols in which the computational load of the prover for an authentication is low. Hopper and Blum's HB protocol [23], for instance, has been quite influential in that area. It has notably inspired Juels and Weis' HB$^+$ protocol [24], designed specifically for RFID, which spawned numerous variants (see e.g. [25–28]). However, most of these proposals present security issues, and some of them present a *false acceptance rate* which could be problematic for large systems. Other protocols aim at satisfying properties such as forward-privacy using

[1] The linear complexity is problematic in large systems (e.g., public transportation), systems with many tags authenticated at the same time (e.g., libraries, logistics), and does not scale well (an application that works well with 100,000 tags might not be possible with 1,000,000 tags).

strong synchronization (see e.g. [29, 30]). We do not consider these protocols in this study for the reason stated above.

2.1.2 Public-key cryptography

Public-key cryptography (PKC) seems to be a solution to the identification problem stated above. The randomized Schnorr protocol [31], for instance, uses public-key encryption to provide both strong privacy and constant-time identification.

However, PKC is expensive, being in terms of gates required on the tag, or of time and especially energy necessary to perform the computations on a tag. Although some recent studies point otherwise (see, e.g., [32–34]), it is generally acknowledged that PKC is not affordable on low-cost tags, as most of the proposals for authentication in RFID use symmetric-key building blocks. We can hope that further research in that area will improve the feasibility of PKC for low-cost RFID, but there will always be a market for symmetric-key solutions.

For these reasons, we only consider symmetric-key schemes in the following.

2.1.3 Privacy

By trying to lower the identification procedure complexity, some solutions also lower the privacy or the security considerably. For instance, one could imagine a very simple scheme where each tag has a limited amount of ephemeral pseudonyms (or "coupons"), using one each time a reader wants to authenticate it. This solution is both private and efficient, but has a limited lifetime and an adversary could perform denial-of-service attacks very easily. Juels proposes in [35] a similar protocol in which each tag loops through a sequence of secrets to authenticate itself to a reader, again providing efficiency, but limited privacy. Henrici and Müller present in [36] a solution in which tags communicate to the reader the number of failed authentication attempts since the last legitimate authentication. While this allows the reader to efficiently identify the tags, it also allows an adversary to easily trace them, as pointed in [37].

In other proposals, such as [38–45], each tag uses pseudonyms that change after each successful authentication session. However, an adversary is able to trace its victim between two legitimate authentications, which is a serious threat in some applications.

Note that despite the fact that these solutions are not private strictly speaking, there might be scenarios where they can be applied, since *some* privacy is better than none at all. However, for the reasons argued in Sect. 1, we will only consider protocols that have no obvious privacy or availability issues in this study.

2.1.4 Building blocks

Finally, there are some other proposals that use non-classical cryptographic building blocks, deemed more lightweight than usual hash functions and ciphers, in order to lower the gate count on tags, and thus their price. An example is the family of so-called *ultralightweight* authentication protocols (see e.g. [38–45]). Although innovative and interesting, this branch is rather recent and to date, all proposals suffer from miscellaneous security and privacy weaknesses.

In addition, a number of works, such as [46–49], aim to provide secure protocols conforming to EPC Class-1 Gen-2 standards. Unfortunately, these attempts fall short of meeting the desired security objectives because EPC Class-1 Gen-2 supports only simple building blocks such as a 16-bit PRNG (Pseudo-Random Number Generator) and a 16-bit CRC (Cyclic Redundancy Code). Many analysis papers (see e.g., [50–53] show that, it seems that enforcing privacy and security under the EPC Class-1 Gen-2 specifications is an almost impossible task due to the “bad” properties of the building blocks used.

For these reasons, we only consider the protocols that use the classical cryptographic primitives, and we focus our analysis on the protocols, not on the underlying building blocks.

2.1.5 Remaining protocols

In the this work, we will consider all the protocols of which we are aware that match the criteria developed above. There are also distance bounding protocols which are related to RFID(e.g., [54–60]). However, this study area is out of the scope of this work since the main propose of these protocols are distance checking of the entity. For more further information about distance bounding protocols we recommend that you refer to [61].

2.1.6 Clarifying some protocol names

With time, some of the protocols were renamed, and to avoid confusion, we present in Table 2.1 the matches between the protocols, proposed with different names. The papers and the publication years are given in the first row. Each remaining row represents one protocol, showing names given in each paper. In what follows we will use the most recently appeared names (shown in bold).

Table 2.1: Matching the names of some protocols

[17] 2006	[62] 2006	[18] 2007	[20] 2009
YA-TRAP	-	YA-TRIP	**RIP**
-	-	YA-TRAP	**RIP+**
-	YA-TRAP+	-	**RAP**
-	O-TRAP	-	**O-RAP**
-	-	-	**O-RAKE**
-	-	**YA-TRAP***	-
-	-	**YA-TRAP*& fwd**	-

2.1.7 Considered security model

In order to analyze the privacy of the protocols we consider, several models are available in the literature [37, 63–65]. We decided to use Juels and Weis' model [64], which is based on Avoine's seminal work [37]. Although Juels and Weis' model is less powerful than Vaudenay's model [65], it is more intuitive and provides an adversary granularity more suited to our analysis than the one provided by [65]. Beyond the concept of *privacy* as defined in [64], we address in this paper the *forward privacy*, as described in [64] as well, and intuitively introduced in [66]. To complete our analyses, we also consider the *timing attacks* against the readers, as introduced in [67].

We emphasize that we do not consider in our work low-level criteria such as the gate count or the power consumption of tags, because, although important, these depend on the implementation of the building blocks. Instead, we focus on the protocols themselves, their efficiency, and the security and privacy level they achieve.

2.2 Protocols with Shared Secrets

Some recent protocols have the common feature that several tags in the system share their secrets (at least partially). They manage to lower the online complexity of the reader by storing tag secrets in a particular structure (a tree, a grid, etc.). While these protocols provide that very desirable property and bring new and interesting ideas, they all have traceability issues.

In this section, we describe Molnar and Wagner's tree-based protocol [68], Alomair, Clark, Cuellar, and Poovendran's protocol [16], Avoine, Buttyán, Holczer, and Vajda's group-based protocol [69], and Cheon, Hong, and Tsudik's meet-in-the-middle protocol [2]. We also discuss some attacks on these protocols, especially two new attacks we suggest against [2].

2.2.1 Tree-based and group-based protocols

As stated previously, privacy-friendly challenge-response protocols do not scale well: the reader must check $O(N)$ keys to authenticate a tag, where N is the total number of tags in the system.

Molnar and Wagner propose in [68] an approach that reduces the complexity from $O(N)$ to $O(\log N)$. The fundamental idea is to manage the tags' keys in a tree structure instead of using a flat structure. More precisely, the tags are assigned to the leaves of a balanced tree with branching factor b at each level of the tree. Each edge of the tree carries a random key. Each tag stores the keys along the path from the root to the leaf corresponding to the given tag, while the reader stores the whole tree. During the authentication process, the reader performs one challenge-response per tree level in order to identify the sub-tree the tag belongs to. Each challenge-response requires from the reader an exhaustive search in a set containing b keys only. The overall reader's complexity of the authentication is $b \log_b N$ in the worst case.

The significant complexity improvement due to Molnar and Wagner's technique (MW) has however an unacceptable drawback: the level of privacy provided by the scheme is quickly decreasing when an adversary tampers with some tags. Giving to the adversary the ability to tamper with some tags makes sense as MW is useless without this assumption: in such a case, the same key can be stored in all the tags and the complexity problem no longer occurs. On the other side, giving to the adversary the ability to tamper with tags significantly degrades the privacy-resistance of MW.

Avoine, Dysli, and Oechslin raise this attack in [19] and evaluate the trade-off between complexity and privacy according to the branching factor. Buttyán, Holczer, and Vajda in [70] also identified weaknesses of MW and introduce an improvement with variable branching factors. Nohl and Evans in [71] provided another approach to analyze MW. Later on, Halevi, Saxena, and Halevi [27] present a lightweight privacy-friendly authentication protocol that combines Hopper and Blum's HB protocol [23] and the tree-based key infrastructure suggested by Molnar and Wagner [68]. However, this protocol inherits from the weaknesses of MW as demonstrated by Avoine, Martin, and Martin in [72]. Finally, Beye and Veugen further analyze the improvement of Buttyán *et al.* in [73].

One may also cite some other attempts to design tree-based protocols, e.g., [74] or the saga [75–78]. The major distinctive property of those protocols comparing to MW is that they use key updating mechanisms. Although these protocols benefit the efficiency of tree-based designs in terms of authentication speed, all of them suffer from security flaws. Note that there is no published weakness so far on [75] but the security level it achieves is not sufficient in practical applications. In this protocol, compromising one tag causes compromising the root key that is shared by the all tags. After that, by only observing a legitimate authentication, the path secret can be obtained easily which is shared among a group of tags. According to Juels and Weis model, [75] does not ensure privacy because one can track any tag between two legitimate authentications (i.e. authentications between legitimate entities) as soon as one tag has been compromised in the system.

Shared secrets are definitely not suited when the adversary is capable of tampering with tags, and the tree structure is even not the best suited in such a case. Indeed,

	K_1^1	K_1^2	K_1^3	$\dots$	K_1^i	$\dots$	K_1^n
K_2^1	$\mathcal{T}_{1,1}$	$\mathcal{T}_{2,1}$	$\mathcal{T}_{3,1}$	$\dots$	$\mathcal{T}_{i,1}$	$\dots$	$\mathcal{T}_{n,1}$
K_2^2	$\mathcal{T}_{1,2}$	$\mathcal{T}_{2,2}$	$\mathcal{T}_{3,2}$	$\dots$	$\mathcal{T}_{i,2}$	$\dots$	$\mathcal{T}_{n,2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
K_2^j	$\mathcal{T}_{1,j}$	$\mathcal{T}_{2,j}$	$\mathcal{T}_{3,j}$	$\dots$	$\mathcal{T}_{i,j}$	$\dots$	$\mathcal{T}_{n,j}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
K_2^n	$\mathcal{T}_{1,n}$	$\mathcal{T}_{2,n}$	$\mathcal{T}_{3,n}$	$\dots$	$\mathcal{T}_{i,n}$	$\dots$	$\mathcal{T}_{n,n}$

Figure 2.1: Tags' secrets organized in a grid as proposed in [2].

Avoine, Buttyán, Holczer, and Vajda demonstrate in [69] that a simpler structure than the tree, namely when tags are grouped and each group share a same key, achieves a higher level of privacy and a better efficiency. Finding a better structure, that does not avoid the traceability problem but that mitigates it is still an open problem.

2.2.2 Cheon, Hong, and Tsudik's protocol

The protocol proposed by Cheon, Hong, and Tsudik in [2] is an innovative proposal to reduce the reader complexity. It uses a *meet-in-the-middle* strategy, similar to the one used in several famous attacks on double-encryption schemes [79].

2.2.2.1 Description

The prorocol steps are as follows. During the initialization, the system chooses two sets of keys $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $|\mathcal{K}_1| = |\mathcal{K}_2| = n$, where $N = n^2$ is the number of tags in the system, and $\mathcal{K}_1 \cap \mathcal{K}_2 = \emptyset$. It then initializes each tag $\mathcal{T}_{i,j}$ with a unique pair of keys $\langle K_1^i, K_2^j \rangle$, where $K_1^i \in \mathcal{K}_1$ and $K_2^j \in \mathcal{K}_2$, yielding an $n \times n$ grid in which each cell represents a tag, as depicted in Fig. 2.1.

The identification procedure, represented in Fig. 2.2, is as follows. The reader $\mathcal{R}$ first picks a nonce r and sends it to a tag $\mathcal{T}_{i,j}$ entering its field. The latter then picks another nonce r', and computes

$$C = PRF_{K_1^i}(r,r') \oplus PRF_{K_2^j}(r,r'),$$

where PRF is a pseudo-random function. The tag $\mathcal{T}_{i,j}$ then sends the pair $\langle C, r' \rangle$ to $\mathcal{R}$. In order to identify the tag, $\mathcal{R}$ computes $PRF_{K_1^x}(r,r')$ for $x \in [1,n]$, and then computes $C \oplus PRF_{K_2^y}(r,r')$ for $y \in [1,n]$, and tries to find a match between two values. This search requires $2n = 2\sqrt{N}$ PRF evaluations at worst, rather than N for a standard

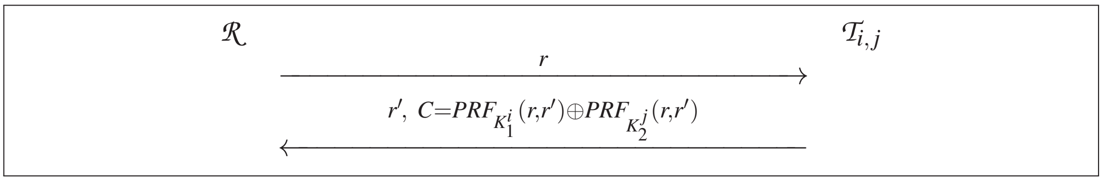


Figure 2.2: Cheon-Hong-Tsudik plain protocol.

linear search[2]. An adversary eavesdropping r, r', and C however would have to search the entire key space, since she does not know the key sets $\mathcal{K}_1$ and $\mathcal{K}_2$.

The protocol presents an efficient search procedure and is allegedly private, but is not synchronized (i.e., the tag has no *state* that changes over time). This implies that it does not provide any forward-privacy, because an adversary having compromised a tag gets its two keys, and can thus recompute messages previously produced by the tag, in this way “tracing” the tag in the past.

Moreover, the authors themselves identify an important issue. Indeed, when a tag is compromised, its two sub-keys are disclosed, but this does not leak any information on other tags’ keys as the combination of subkeys is unique. However, when the adversary compromises several tags, she gains knowledge of key-pairs of legitimate tags. For instance, If the adversary compromises the tags $\mathcal{T}_{a,b}$ and $\mathcal{T}_{c,d}$, she also discovers the keys of the tags $\mathcal{T}_{a,d}$ and $\mathcal{T}_{b,c}$. These will respectively be referred as *directly compromised tags* and *indirectly compromised tags* hereafter (a *compromised tag* refers to either situation). We also name *partially compromised* the tags for which we only know one key.

The authors describe an extension to mitigate this problem by introducing proper authentication in the protocol. In this extension, represented in Fig. 2.3, each tag has a third, unique sub-key K_3. The key sets $\mathcal{K}_1$ and $\mathcal{K}_2$ have a size of N^α, with $0 \leq \alpha \leq \frac{1}{2}$ being a system parameter[3] and $\mathcal{K}_3$ has a size of N, such that $N^{1-2\alpha}$ tags have the same $\langle K_1, K_2 \rangle$ key-pair. The tag further computes

$$C' = PRF_{K_3}(r, r'),$$

[2] Note that in [2], the authors state that the search is $O(\sqrt{N}\log N)$. We consider only the cryptographic operations in the online time, so we suggest $O(\sqrt{N})$ instead.

[3] The authors recommend the value $\alpha = \frac{1}{3}$ for optimal search efficiency.

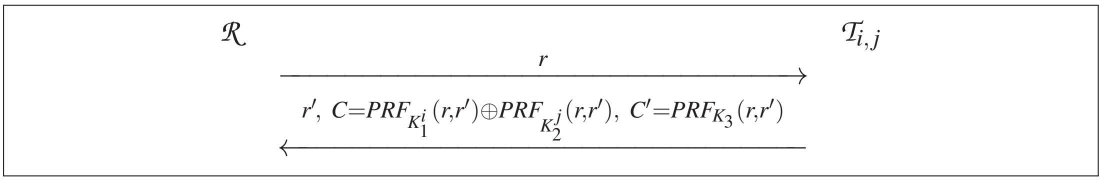

$\mathcal{R}$
$\mathcal{T}_{i,j}$
r
r', $C{=}PRF_{K_1^i}(r,r'){\oplus}PRF_{K_2^j}(r,r')$, $C'{=}PRF_{K_3}(r,r')$

Figure 2.3: Cheon-Hong-Tsudik protocol with authentication extension.

and sends it to the reader. After the usual search procedure, $\mathcal{R}$ checks the value C' to authenticate the tag. Since K_3 is unique to each tag, the aforementioned impersonation attack is prevented, but there is still a traceability issue, as detailed in Sect. 2.2.2.3.

2.2.2.2 Impersonation attack on the plain protocol

After having compromised some tags, an adversary can perform the following impersonation attack. The adversary listens to a legitimate authentication session between $\mathcal{R}$ and $\mathcal{T}_{i,j}$. When $\mathcal{T}_{i,j}$ outputs r' and C, she blocks the message. The adversary can now change C in order to authenticate another tag than $\mathcal{T}_{i,j}$. Because the protocol is stateless, $\tilde{C}$, the modified C, will be accepted as long as it is valid, and the corresponding tag will be identified. Two situations may occur for an adversary:

1. She wants to authenticate a tag that is compromised instead of $\mathcal{T}_{i,j}$.

2. She wants to authenticate a tag that is partially compromised.

In case 1, the adversary can replace the tag being authenticated with another compromised tag, say, $\mathcal{T}_{a,b}$, by simply replacing C by $\tilde{C} = PRF_{K_1^a}(r,r') \oplus PRF_{K_2^b}(r,r')$. This problem was already highlighted in [2].

In case 2, the adversary must at least know one of the keys of $\mathcal{T}_{i,j}$ to succeed (i.e. $\mathcal{T}_{i,j}$ must be partially compromised). Let us suppose that the adversary knows K_1^i but not K_2^j, and that she also knows another key K_1^k. She can then replace C by $\tilde{C} = PRF_{K_1^k}(r,r') \oplus PRF_{K_2^j}(r,r')$ by computing $\tilde{C} = C \oplus PRF_{K_1^i}(r,r') \oplus PRF_{K_1^k}(r,r')$, and by doing so, authenticate $\mathcal{T}_{k,j}$, which is only partially compromised. Of course, she does not know the keys of the victim in advance, so the attack is probabilistic. She can thus iterate on all the tags for which she knows the secrets partially. A side-effect of this is that when $\mathcal{R}$ accepts the authentication, the adversary gets $PRF_{K_2^j}(r,r')$, which can lead to a traceability attack.

In [2], the authors state that, when compromising t tags, the number of indirectly compromised tags is $t^2 - t$. This is actually rather optimistic (from an attacker viewpoint) and only accurate when t is small (because when t gets larger, the probability of corrupting a tag from which we do not know any sub-key becomes smaller). We provide a more precise result in Lemma 1.

Lemma 1. *Let T denote the number of* directly compromised *tags and S the total number of* compromised *tags, that is the ones for which we know both keys. Then, the expected number of compromised tags given that we compromised t tags is:*

$$\mathbb{E}[S|T=t] = N\left[1 - \frac{2\binom{N-n}{t} - \binom{N-2n+1}{t}}{\binom{N}{t}}\right],$$

where $n = \sqrt{N}$. A similar result applies for the authentication extension, and S here denotes the number of compromised cells*:*

$$\mathbb{E}[S|T=t] = n^2\left[1 - \frac{2\binom{N-N/n}{t} - \binom{N-2N/n+N/n^2}{t}}{\binom{N}{t}}\right],$$

with $n = N^{\alpha}$.

We present below the proof of Lemma 1. Let us consider the plain protocol first. We have $\mathbb{E}[\text{\# of indirectly compromised tags}] = \mathbb{E}[\text{\# of compromised tags}] - t$, where a compromised tag refers to a tag that is either directly or indirectly compromised. Let's denote by R_i the following random variable:

$$R_i = \begin{cases} 1 & \text{if at least one compromised tag has } K_1^i \\ 0 & \text{otherwise.} \end{cases}$$

Likewise, we define:

$$C_i = \begin{cases} 1 & \text{if at least one compromised tag has } K_2^i \\ 0 & \text{otherwise.} \end{cases}$$

Note that the total number of compromised tags can be expressed as:

$$S = \left(\sum_{i=1}^{n} R_i\right)\left(\sum_{i=1}^{n} C_i\right).$$

Indeed, this number corresponds to the number of tags ("cells" in the grid) for which we know K_1 and K_2. We thus have

$$\begin{aligned}\mathbb{E}[S] = \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} R_i C_j\right] = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}\left[R_i C_j\right] \\ = \sum_{i=1}^{n}\sum_{j=1}^{n} \Pr(R_i = 1 \wedge C_j = 1).\end{aligned}$$

Note that R_i and C_j are not completely independent. However, we have

$$\begin{aligned}\Pr(R_i = 1 \wedge C_j = 1) &= 1 - \Pr(R_i = 0 \vee C_j = 0) \\ &= 1 - [\Pr(R_i = 0) + \Pr(C_j = 0) \\ &\qquad - \Pr(R_i = 0 \wedge C_j = 0)].\end{aligned}$$

$\Pr(R_i = 0)$ is the probability that, after compromising t tags, none belong to the row i. That is, $\Pr(R_i = 0) = \binom{N-n}{t}/\binom{N}{t}$. Moreover, $\Pr(C_j = 0) = \Pr(R_i = 0), \forall i, j$ since the grid is symmetric. Likewise, $\Pr(R_i = 0 \wedge C_j = 0) = \binom{N-2n+1}{t}/\binom{N}{t}$. Finally,

$$\begin{aligned}\mathbb{E}[S|T = t] &= \sum_{i=1}^{n}\sum_{j=1}^{n} \Pr(R_i = 1 \wedge C_j = 1 | T = t) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \left[1 - 2\frac{\binom{N-n}{t}}{\binom{N}{t}} + \frac{\binom{N-2n+1}{t}}{\binom{N}{t}}\right] \\ &= N\left[1 - \frac{2\binom{N-n}{t} - \binom{N-2n+1}{t}}{\binom{N}{t}}\right].\end{aligned}$$

An example is presented in Fig. 2.4.

The demonstration for the authentication extension is very similar to the above, except that cells contain $N^{1-2\alpha}$ tags.

This result allows to quantify the probability of success of our attacks and confirms their feasibility, as we will see below.

2.2.2.3 Traceability attack on the authentication extension

Recall that in the authentication extension, the grid can now be seen as $N^{\alpha} \times N^{\alpha}$ "cells" of $N^{1-2\alpha}$ tags secrets. No two tags share the K_3 key, but each $\langle K_1, K_2 \rangle$ is shared among $N^{1-2\alpha}$ tags. As the authors mentioned, this leads to a traceability issue because if an attacker knows a $\langle K_1^i, K_2^j \rangle$ pair, she can track $\mathcal{T}_{i,j}$ with probability $1/N^{1-2\alpha}$ by using the fact that there are $N^{1-2\alpha}$ tags with the same pair.

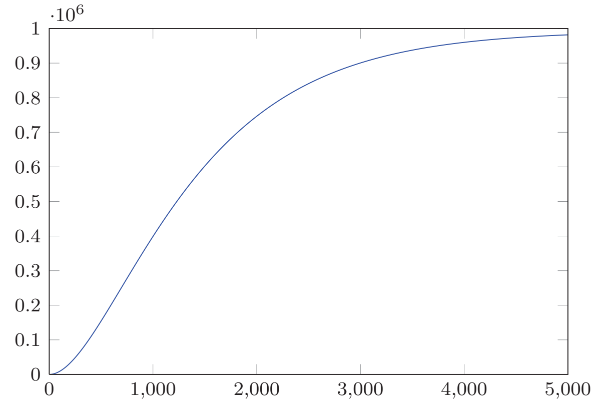


Figure 2.4: Average number of indirectly compromised tags in a system of $N = 10^6$ tags, with respect to an increasing number of directly compromised tags.

In this section, we point out a more dangerous issue. Let us assume that the adversary has obtained the keys related to s cells. For the sake of simplicity, we assume that the compromised tags are put back into circulation. Since this number is supposedly small compared to N, the number of tags in the system, this is a reasonable assumption.

Let X denote the set of tags which secrets belong to one of the s cells known by the adversary. In a Juels and Weis game [64], when two tags $\mathcal{T}_0$ and $\mathcal{T}_1$ are presented to her, the adversary is asked to answer which of these tags is her target. Several cases may occur:

- $E_1 = \mathcal{T}_0 \in X \wedge \mathcal{T}_1 \notin X$
- $E_2 = \mathcal{T}_0 \notin X \wedge \mathcal{T}_1 \in X$
- $E_3 = \mathcal{T}_0 \in X \wedge \mathcal{T}_1 \in X \wedge \langle K_1, K_2 \rangle_{\mathcal{T}_0} \neq \langle K_1, K_2 \rangle_{\mathcal{T}_1}$
- $E_4 = \mathcal{T}_0 \in X \wedge \mathcal{T}_1 \in X \wedge \langle K_1, K_2 \rangle_{\mathcal{T}_0} = \langle K_1, K_2 \rangle_{\mathcal{T}_1}$
- $E_5 = \mathcal{T}_0 \notin X \wedge \mathcal{T}_1 \notin X$

The obvious strategy for an adversary is, after choosing r, to query $\mathcal{T}_0$ and $\mathcal{T}_1$, and compare their answer with what would have answered the tags of which she knows the keys. If there is a match, then she identifies the tag and deduces its keys. In E_1 and E_2, only either of $\mathcal{T}_0$ and $\mathcal{T}_1$ is identified, and the adversary is able to determine correctly whether it is her target or not in all cases. If both tags are identified, the adversary succeeds only when they have a different key-pair (E_3, but not E_4). Finally, if neither is identified, the adversary is unable to tell her target apart in any better way than at random. Therefore, in the first three events, the adversary succeeds in the attack, and in the other two she fails. It is clear that the first two cases are symmetric:

$$\Pr(E_1) = \Pr(E_2) = \frac{NM - M^2}{N^2}, \tag{2.1}$$

where $M = sN^{1-2\alpha}$, that is the number of tags for which the adversary knows the secrets. Likewise, we have

$$\Pr(E_3) = \frac{M^2}{N^2}(1 - 1/s). \tag{2.2}$$

The overall probability that the adversary succeeds after corrupting s cells is thus

$$\begin{aligned}\Pr(E_1 \vee E_2 \vee E_3) &= \Pr(E_1) + \Pr(E_2) + \Pr(E_3)\\ &= 2\frac{M}{N} - \frac{M^2}{N^2}(1 + 1/s),\end{aligned}$$

because these events are mutually exclusive. This probability can become much higher than the one presented in [2]. For instance, in a system with $N = 10^6$ tags, configured with $\alpha = \frac{1}{3}$ (as suggested by the authors), an adversary having compromised $t = 300$ tags has roughly $s = 8750$ compromised cells (Lemma 1), and a probability of winning close to 0.984 which is quite high.

2.2.2.4 Discussion

We have introduced two important attacks on CHT. The first attack regards the plain protocol and allows an adversary to authenticate another tag than the one which initiated the authentication session. The targeted tag need not be completely indirectly compromised, as a probabilistic approach can be carried out.

The second attack regards the authentication extension, and allows an adversary to trace a tag in the system. The second attack is similar to the one proposed in [19] against Molnar and Wagner's protocol. Although both protocols are quite different technicallly, Molnar and Wagner's protocol and CHT have in common the fact that tags share parts of their secrets. This property yields efficient tag identification, but compromising tags becomes far more dangerous. We present in Fig. 2.5 a comparison

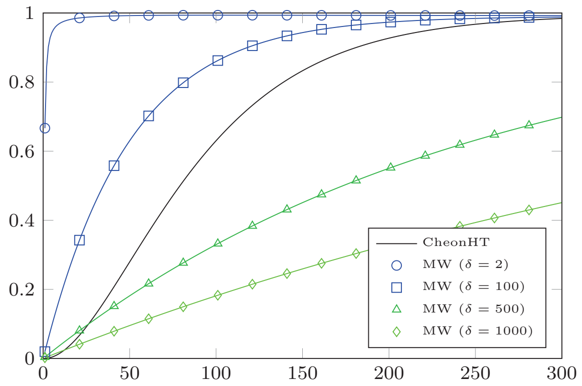


Figure 2.5: Probability of tracing a tag in CHT and in MW with respect to the number of (directly) compromised tags, in a system with $N = 10^6$ tags.

of the probabilities of tracing attacks in CHT and Molnar and Wagner's protocols (with different values for the branching factor).

2.2.3 Alomair, Clark, Cuellar, and Poovendran's protocol

The protocol introduced by Alomair, Clark, Cuellar, and Poovendran in [16] provides Constant-Time Identification (CTI). We classify this protocol in the shared-secret family in the sense that the system manages a pool of shared secret pseudonyms such that each tag is paired with a pseudonym for a while, and is reassigned to another one each time it is legitimately authenticated. Consequently, different tags may use the same pseudonym, at different times. Using re-usable pseudonyms was first introduced by Juels in [35] where each tag manages its own pool of pseudonyms and uses linear combination of them once all the pseudonyms have been used. However, tags do not exchange their pseudonym in [35], contrarily to [16].

2.2.3.1 Description

The protocol steps of [16] is as following. During the set up phase, each of the N_T tags is assigned with a secret key k, a cycling counter c that is incremented modulo C each time the tag is queried (initially $c = 0$), and an initial pseudonym ψ drawn from a pool $\mathscr{E}$ of size $N > N_T$. In the back-end database, all the possible hash values for all the pseudonyms and all the counter values are precomputed and stored. A sketch of CTI protocol is depicted in Fig. 2.6 and we refer the reader to the original paper [16] for a detailed description.

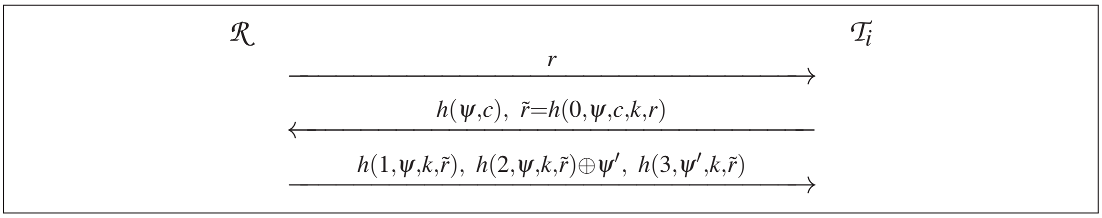

$\mathcal{R}$
$\mathcal{T}_i$
r
$h(\psi,c),\ \tilde{r}=h(0,\psi,c,k,r)$
$h(1,\psi,k,\tilde{r}),\ h(2,\psi,k,\tilde{r})\oplus\psi',\ h(3,\psi',k,\tilde{r})$

Figure 2.6: Alomair, Clark, Cuellar, and Poovendran's (CTI) Protocol.

The key-point of CTI is that each time a tag is legitimately authenticated, it releases its current pseudonym in order to get a new one from the reader randomly drawn from $\mathscr{E}$; it also updates its secret key k with the value $h(k)$ where h is a hash function. CTI provides constant time identification but this property is obtained after pre-calculation of all the NC possible answers from the tags. In that sense, CTI is not far from OSK [66]: the table of pairs (pseudonym, counter) in [16] is in some way similar to the table of pairs (identifier, counter) in [66]. A few differences can nevertheless be raised: (1) a denial of Service (DoS) occurs with OSK after M illegitimate authentications, while CTI is DoS-resistant; (2) OSK with authentication requires to compute between 3 (if there is no attack) and $2m+1$ hash calculations per identification, while CTI requires 4 hash calculations in any case; (3) CTI needs a larger memory than OSK and provides a lower privacy-resistance than OSK, as explained below. Note that both of them are resistant to timing attacks as stated in [67]. OSK will be studied in Sect. 2.3.1.

2.2.3.2 Intra-legitimate authentication attack

The main drawback of CTI, already mentioned in [16] is the cycling counter because a tag can be easily tracked between two legitimate authentications if an adversary is able to query it C times. Indeed, recording each of the answers $h(0,\psi,c,k,r)$ ($0 \leq c < C$), the adversary can definitely track the tag till the next legitimate authentication. This attack is especially meaningful when considering tags that are not frequently used, e.g., passports or tickets used for ephemeral event and kept by the customer as souvenir... Increasing C makes the attack harder, but this also significantly increases the memory consumption (and the reader's workload during the setup). This attack makes CTI not traceability-resistant in the Juels and Weis model [64].

2.2.3.3 Inter-legitimate authentication attack

The pseudonyms used in the system are originally secret and can only be revealed in case of tampering attack. In such a case the current pseudonym of the compromised tag is revealed (and the secret key as well) but the adversary can also obtain additional pseudonyms by impersonating the tag in the system. This attack is mentioned in [16] but we refine its analysis and show that its impact should not be underestimated. First of all, the number of pseudonyms obtained by the adversary after tampering with only one tag is [16]:

$$\left[N\left(1-\left(\frac{N-1}{N}\right)^{q}\right)\right] \qquad \textbf{(2.3)}$$

where q is the number of protocol executions[4]. Let $\mathscr{E}^q \subset \mathscr{E}$ the set of pseudonyms so revealed, the adversary can track a tag (even after legitimate authentications) as follows: in the learning phase as defined in the model of Juels and Weis [64], the adversary queries the targeted tag $\mathcal{T}_{\text{target}}$ once and so obtains a value $h(\psi_{\text{target}}, c_{\text{target}})$. Trying an exhaustive search on all values in $\mathscr{E}^q$ and all counter values, she obtains c_{target} if and only if $\psi_{\text{target}} \in \mathscr{E}^q$, which occurs with probability $|\mathscr{E}^q|/N$. In the challenge phase, given $\mathcal{T}_0$ and $\mathcal{T}_1$, the adversary must decide which one is $\mathcal{T}_{\text{target}}$. To do so, she applies the same technique and so possibly obtains c_0 and c_1. From c_0 and c_1, she could be able to decide which of $\mathcal{T}_0$ and $\mathcal{T}_1$ is $\mathcal{T}_{\text{target}}$. For example, if the adversary knows that her target is rather new while c_i ($i = 0$ or 1) is large, it may be safe to conclude that $\mathcal{T}_{\text{target}}$ is $\mathcal{T}_{1-i}$. To illustrate this attack, consider the following practical parameters: $N = 2N_T$, $N_T = 10^6$, $C = 10^3$, and $q = 10^3$. The probability to track a given tag is therefore 0.1%, assuming that one of the two tags only is rather new.

2.2.4 Discussion

While protocols using shared secrets all aim mainly to decrease the identification time on the reader, they all have issues when facing adversaries capable of compromising

[4] Note that [16] suggests to limit the number of requests to a reader per tag, but bounding q to a value less than 1000 does not seem realistic in most applications as the adversary can avoid being detected, using a slow attack.

tags. One could argue that a protocol using only one "master key" is the extreme case in that direction: it has constant-time identification, but no privacy/security as soon as one tag is compromised.

All of the proposals we analyzed in this section have important problems, mostly due to the fact that compromising one tag reveals information on other tags too. However, we have no element showing that sharing secrets between tags is a definitely flawed way of reducing identification time. It remains an open question whether it is possible to design such a protocol without any loss of security or privacy.

2.3 Protocols Based on Hash-Chains

An early family of sub-linear protocols uses hash-chains to update the internal state of the tags. In this section, we describe Ohkubo, Suzuki, and Kinoshita's (OSK) [66] two of its improvements, OSK/AO [19, 80] and OSK/BF [22]. Then we describe the PFP [81], and O-RAP [20] protocols.

We show a traceability attack on the mutual authentication extension of OSK/AO protocol, and we suggest a solution to overcome this problem. We also show new weaknesses of O-RAP and OSK/BF.

2.3.1 OSK protocol

OSK [66] is a well-known synchronized identification protocol, and was one of the earliest of its kind. Each tag $\mathcal{T}_i$ of the system is initialized with a randomly chosen secret s_i^0. When queried by a reader, a tag answers with the hash of its *current* secret, that is $\sigma = G(s_i^k)$, and immediately updates it using another hash function: $s_i^{k+1} = H(s_i^k)$. When receiving an answer, the reader looks in its database for an initial secret s_j^0 that leads to σ, in other words, it checks whether there exists i and j such that $G(H^i(s_j^0)) = \sigma$. To do that, from each of the N initial secrets s_j^0, the reader computes the hash chains as shown in Fig. 4.1 until it finds a value matching σ, or until it reaches a given maximum limit M on the chain length. An overview of the protocol is shown in Fig. 4.2.

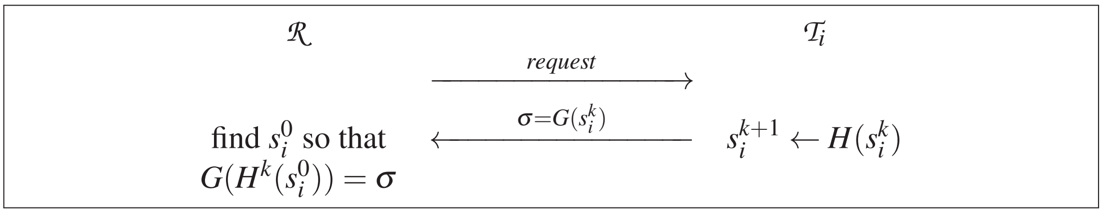


Figure 2.7: OSK protocol.

The value $\sigma = G(s_i^k)$ does not allow an eavesdropper to learn the identity of $\mathcal{T}_i$. However, since a tag updates its secret regardless of the success of the identification, a rogue reader initiating the protocol with $\mathcal{T}_i$ will make it update its secret. An adversary initiating lots of instances of the protocol with $\mathcal{T}_i$ will perform a *desynchronization* denial-of-service attack. Indeed, the reader would then need to compute a lot of hashes to identify $\mathcal{T}_i$. To prevent this, the length of the hash chains have to be bounded, i.e., the reader stops its search after M hashes per tag. This protection has the following drawback: an adversary skimming a tag M times makes it unable to be identified by the system, and thus traceable.

Beside this traceability issue, and although the protocol is very efficient when all tags are synchronized, the worst-case complexity of the search makes the protocol unsuitable for most practical systems.

The authors later introduced in [82] some ideas to improve the efficiency of the search at the cost of lowering privacy. Since strong privacy is one of the design goals of OSK, we will not consider them further.

2.3.2 OSK/AO protocol

Avoine and Oechslin propose in [80] to apply Hellman's time-memory trade-offs [83] to the search procedure of OSK, which has two main implications. First, the complexity of the search procedure varies from $O(1)$ to $O(N)$, depending on the amount of memory we are willing to devote to the time-memory trade-off[5]. Moreover, the search is intrinsically randomized, which prevents timing attacks [67].

[5] The authors mention that, for instance, a complexity of $O(N^{2/3})$ can be reached with a memory of size $O(N^{2/3})$.

$$\begin{array}{ccc} \mathcal{R} & & \mathcal{T}_i \\ & \xrightarrow{\quad r \quad} & \\ \text{search for } i & \xleftarrow{H_1(s_i^k \oplus r),\ H_1(s_i^k)} & s_i^{k+1} \leftarrow H_0(s_i^k) \\ s_i^{k+1} \leftarrow H_0(s_i^k) & \xrightarrow{\quad H_2(s_i^{k+1}) \quad} & \end{array}$$

Figure 2.8: Patched OSK with replay-attack protection and reader authentication.

Avoine, Dysli, and Oechslin also suggest in [19] a variant of OSK that ensures authentication as OSK is originally designed to provide private identification only (i.e., it does not resist to replay attacks). To do so, they suggest using nonces: instead of simply sending a *request* message, the reader sends a nonce r, and the tag answers $G(s_i^k \oplus r)$ along with $G(s_i^k)$.

Finally, Avoine proposes in [84] an extended version of OSK that provides reader authentication to the tag: the reader sends a last message $G(s_i^{k+1} \oplus w)$, where w is a public static value.

However, we point out a traceability issue in this extension: an adversary can eavesdrop a legitimate authentication between $\mathcal{R}$ and $\mathcal{T}_i$, and record the last message (i.e. $G(s_i^{k+1} \oplus w)$); after a while, she sends w as a nonce to a tag, and if the tag answers with the previously recorded value, it means that this tag is almost certainly $\mathcal{T}_i$, and that it has not been queried since then.

Preventing this attack can be done easily using a third hash function for the last message. In practice, a single hash function is implemented and an additional input enables to derive it into several functions, for instance, by concatenating 0, 1, or 2 to the value to hash. Fig. 2.8 shows our modification to the mutual authentication extension of OSK/AO.

2.3.3 OSK/BF protocol

Nohara, Inoue and Yasuura propose in [22] another innovative time-memory trade-off for OSK, which we denote OSK/BF in the following. They use Bloom Filters [85], a space-efficient data structure, to store all the hash-chains of each tag.

2.3.3.1 Description

In order to identify a tag, the reader first queries all the Bloom Filters for the received σ, and then computes the whole hash-chain of each candidate to confirm the identity of the tag. Once identified, the corresponding Bloom Filter is re-computed to match the next hash-chain. On that point OSK/BF contrasts with OSK/AO, in which updates of the database occur less frequently but are more expensive.

As presented in [22], OSK/BF is an identification protocol and does not resist impersonation. However, we point out that it can be easily adapted to an authentication scheme using the same construction as the one in [19].

In [86], Nohara and Inoue present an analogous protocol using a similar architecture but a different data structure, d-left Hash Tables [87], an extension of Bloom Filters. The resulting protocol has, according to the authors, a better update efficiency than OSK/BF, but it turns out to be the same. Furthermore, the identification time seems to be very comparable to that of [22], and it has the further disadvantage of being less parameterizable.

2.3.3.2 Traceability timing attacks

We point out two potential traceability weaknesses in this OSK/BF due to timing analysis, not mentioned in [22]. The first one uses the fact that the search is linear in [22], meaning that $\mathcal{T}_1$ will on average be authenticated much faster than $\mathcal{T}_N$, for instance. The reason for this is that when a tag has a record (and a corresponding Bloom Filter) at the start of the table, the reader will have to go through few false positives invalidations before actually confirming the identity of the tag, whereas when it has a record near the end of the table, it might go through several of them. The second attack uses the fact that it is possible to trace a tag being desynchronized more than M times by observing whether the identification time remains constant (it should be the case when the reader refuses identification, but not the case when the Bloom Filters get updated). Countermeasures might exist against these attacks (simply shuffling the search seems to be a solution to the first one), but in any case, OSK/BF is more fragile regarding timing analysis than OSK/AO, and avoiding them without artificially waiting for $O(N)$ cryptographic operations does not seem to be trivial.

2.3.3.3 Comparison OSK/BF with OSK/AO

As in OSK/AO, the time of identification can be lowered by increasing the memory of the reader. In OSK/BF, this is done by tuning the false positive rate of the Bloom Filters. Doing so results in more time needed to compute the hash-chains in order to infirm false positives, increasing identification time, but also in a decrease of the size of Bloom Filters and thus of memory. In OSK/AO, this is done by tuning the size of the Rainbow table, and also determining the amount of intermediate columns stored.

A slight advantage of OSK/BF over OSK/AO is that, despite it also has a probabilistic nature, the successful identification rate is of 100% while being *close to* 100% (fixed by parameters) in OSK/AO. However, the two protocols have the same disadvantage regarding desynchronization, i.e., a tag desynchronized more than M times is lost.

Regarding the trade-off efficiency, OSK/AO seems slightly more efficient than OSK/BF, although comparable. We used numbers from [19], i.e. a system of 2^{20} tags and chains of 2^7 hashes, to provide a comparison between the two protocols, which we depict in Fig. 2.9. The saturation in OSK/BF after some point comes from the fact that the update part takes $2M$ cryptographic operations, no matter how much memory is dedicated to the trade-off. Note also that we did not take the random hash calculations into account, and that depending on the functions used, they could potentially increase the identification time significantly.

2.3.4 PFP protocol

Berbain, Billet, Etrog, and Gilbert present in [81] an authentication protocol strongly inspired by OSK. The main additional claim with respect to the latter is the lightweight nature of their proposal. They point out the fact that the collision-resistance property is useless in the hash functions used in OSK. Their solution makes use of a strongly universal hash function family and a pseudo-random generator.

The protocol is essentially the same as the OSK protocol with the replay attack countermeasure (nonce produced by the reader). The function used for the update of

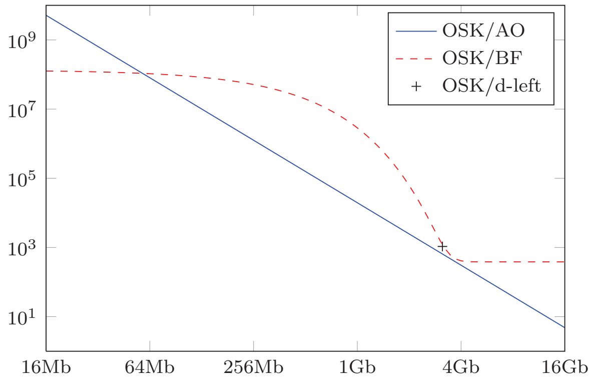
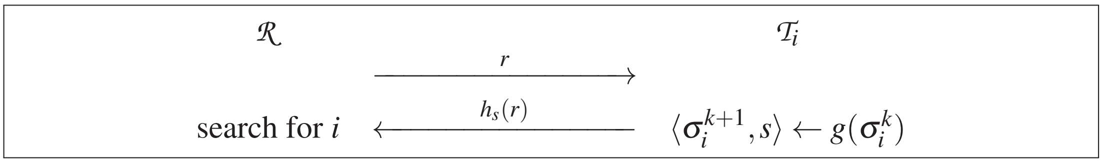


Figure 2.9: Average number of cryptographic hashes during identification depending on the memory dedicated to the trade-off in a system with $N = 2^{20}$ tags, and chains of $M = 2^7$ hashes.



$\mathcal{R}$
$\mathcal{T}_i$
r
search for i
$h_s(r)$
$\langle \sigma_i^{k+1}, s \rangle \leftarrow g(\sigma_i^k)$

Figure 2.10: PFP protocol.

the secret state σ_i^k is no longer a hash function but a pseudo-random number generator $g : \{0,1\}^n \to \{0,1\}^{n+k}$. The first n bits of $g(\sigma_i^k)$ are used as the next state σ_i^{k+1}, and the last k ones are used to select one hash function in the universal hash function family $\{h_s\}_{s\in\{0,1\}^k}$. This hash function is then used to hash the nonce sent by the reader, in order to allow the latter to identify $\mathcal{T}_i$ in a way much similar to OSK. The PFP protocol is illustrated in Fig. 2.10.

As is, the protocol has a linear reader complexity. However, the authors propose a solution to accommodate it for time-memory trade-off, allowing the search to be sub-linear, as in OSK/AO [19]. The building blocks in PFP are different than the ones in OSK, and they are used in a different way, but the global scheme is the same, and the security and privacy properties of the two protocols are equivalent. Hence, we will not detail PFP further.

2.3.5 O-RAP protocol

O-RAP, which stands for *Optimistic RFID Authentication Protocol* has been originally introduced in [62] (its former name was O-TRAP — see Table 2.1) and a slightly modified version is re-presented in [20]. They call the protocol "optimistic" for the reason that the security overhead is minimal when the system is not under attack.

2.3.5.1 Description

The steps of O-RAP are shown in Fig. 2.11. The reader contains a hash table indexed by r_{tag} with entries K_i (the static keys of the tags). When starting an authentication, the reader sends a random number r_{sys} to the tag. The tag computes the hash of r_{sys} and r_{tag} with its key K_i and gets r and h output values. Then the tag sends h and r_{tag} values to the reader. The tag also updates r_{tag} with r value. The system searches r_{tag} to find the corresponding K_i in the database, and if found, it checks the correctness of the hash. If r_{tag} is not found, then it exhaustively searches among all the keys. If found, it validates the tag and updates r_{tag} with r value. This allows the reader to re-synchronize the tag automatically.

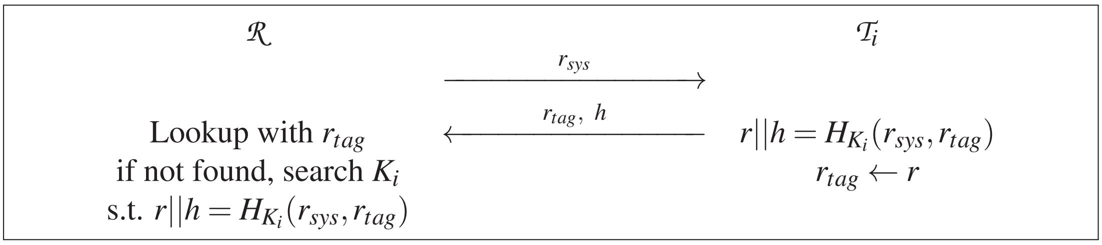

$\mathcal{R}$
$\mathcal{T}_i$
r_{sys}
$r_{tag},\ h$
Lookup with r_{tag}
if not found, search K_i
s.t. $r||h = H_{K_i}(r_{sys}, r_{tag})$
$r||h = H_{K_i}(r_{sys}, r_{tag})$
$r_{tag} \leftarrow r$

Figure 2.11: O-RAP Protocol.

2.3.5.2 Attack by Ouafi and Phan

In [88], Ouafi and Phan propose a traceability attack on O-RAP based on the desynchronization of a tag. The idea is that an adversary can make enough queries to a tag in order to make it update its secret r_{tag} a lot of times to the point that a legitimate reader is unable to authenticate it anymore.

However, we point out that this attack is erroneous. Indeed, the tag always sends r_{tag} in its answer so the resynchronization is trivial, and because K_i does not change, the authentication is always correct, regardless of how many queries the attacker has performed.

2.3.5.3 Forward-privacy issue and O-FRAP

Although the authors raise the problem in [62], no particular attention has been drawn on the forward-privacy of O-RAP. An attacker compromising $\mathcal{T}_i$ at some point can recover r_{tag} and K_i. This allows him to trace $\mathcal{T}_i$ in the past, because r_{tag} is sent in the clear and is updated by r. This update can be computed by the adversary, since K_i does not change.

The authors propose in [21] the O-FRAP protocol, adding the forward-privacy to O-TRAP. This comes at the cost of an extra pass in order to authenticate the reader to the tag, as well as a memory overhead for storing previous keys. However, we point out that the protocol is not forward-private strictly speaking. Indeed, suppose than an adversary queries the tag some times without answering to it. Afterwards, she compromises the tag, and if the tag has not been authenticated since, she will be able to trace it in the past. This is the same idea as protocols using pseudonyms for identification, discarded in Sect. 2.1.3.

Also note that in [20] and [21], the authors propose key exchange extensions to O-RAP and O-FRAP respectively, namely O-RAKE and O-FRAKE. Their goal is to provide features outside of authentication, which is beyond the scope of our paper.

2.3.5.4 Traceability timing attack

The fact that O-RAP behaves differently according to synchronization makes it work very efficiently in “normal” situations, but allows an adversary to carry out the following timing attack. The adversary first sends a random number to a tag and ignores its answer. The tag will thus be desynchronized with the system, and the next legitimate reader trying to authenticate it will take much more time, because in that case, the search is linear. The adversary can easily notice that by measuring time differences, and can thus trace the tag she desynchronized. A possible countermeasure is to artificially add time for the search in a normal situation, but this would be equivalent to a protocol with linear complexity.

2.3.6 Discussion

OSK and O-RAP are two convincing proposals with a simple design and interesting properties. As pointed by Avoine and Oechslin in [80] and by Nohara *et al.*

in [22], OSK can be easily accommodated to using time-memory trade-offs, which make the identification procedure efficient. It also provides forward-privacy to the tags. However, the synchronization issue present in OSK and its variants, although mitigatable, remains significant. In that regard, the O-TRAP protocol has no such synchronization issue because tags automatically “re-synchronize” with each authentication attempt. It is also the reason why the identification procedure is constant-time in normal situations. However, it is very easy to make the next search linear by querying the tag once. This also leads to traceability issues using reader-side timing analysis. Additionally, it provides no forward-privacy. Despite their respective weaknesses, these protocols are nonetheless probably the most solid solutions we analyzed.

2.4 Counter-Based Protocols

The *counter-based protocols* all share the same characteristics: they use a strictly increasing number[6] and maintain a periodically updated hash table for each counter. The idea is to pre-compute the table at each counter tick, in order to reduce the online search to a constant time on the server-side.

In this section we examine a family of counter-based protocols, namely RIP, RIP+, RAP, and YA-TRAP* (see Table 2.1 for the names given in different papers). We show a traceability attack on the most advanced protocol proposed in [18], namely YA-TRAP*, based on timing analysis.

2.4.1 YA-TRAP family

A family of tag identification and authentication protocols that use strictly increasing counters is proposed in the papers [17, 18, 20, 62]. The first protocol, RIP, stands for *RFID Identification Protocol*. It is followed by authentication protocols called RIP+, YA-TRAP*, and a variant of YA-TRAP* with forward-privacy (we call this protocol YA-TRAP*&fwd).

[6] In some previous papers [17, 18, 20, 62] the name “*timestamp*” is used to denote a strictly increasing number. Since the tags do not have any clock and this number is not a cryptographic timestamp, we prefer using the more generic term *counter*.

2.4.1.1 Description

We describe below the RIP [20] protocol, which is the simplest and earliest proposal in the family.

Each tag $\mathcal{T}_i$ is initialized with a starting counter T_0 and a maximum counter value $T_{\max}$, as well as with a unique secret key K_i. When initiating an authentication, the reader sends its current counter T_r. The tag checks that T_r is less than $T_{\max}$ and that the received counter is bigger than the one it currently stores, T_t, which it received during the last successful identification. If these conditions hold, it stores the new counter and computes and sends the hash of T_r with its key K_i. Otherwise, the tag sends a random number to prevent an adversary from drawing any conclusion. The authors added that to avoid timing attacks against a tag at this point, the nonce generation must be designed to take approximately the same time as the hash computation.

As stated above, every now and then, the server increases the value of the counter, and re-computes the table accordingly. This allows for a constant time identification online, but takes time offline.

The authors identified several drawbacks in this protocol. First, it is vulnerable to a trivial DoS attack: the adversary can temporarily or permanently incapacitate a tag by sending a future counter. Although the authors point out that DoS resistance is not the main goal of this protocol, the attack is very easy to perform and very hard to recover from. Second, it is implicitly assumed that a tag is never identified more than once between two consecutive counter ticks. A short time interval (e.g., a second) between two counter updates makes this assumption realistic, but it causes heavy computational burden for the server. RIP is also vulnerable to replay attacks: an adversary can send a counter slightly ahead to a tag and wait until this counter is sent by the server. She can repeat this attack and thus impersonate its victim for a long time without the original tag being present. RIP is depicted in Fig. 2.12.

In RIP+, the protocol is modified in order to provide authentication. The reader sends a random nonce R_r along with the counter, and the tag chooses its random nonce R_t and compute a second hash for authentication, i.e., $h_{auth} = H_{K_i}(R_t, R_r)$. The reader first identifies the tag then checks the correctness of the authentication message.

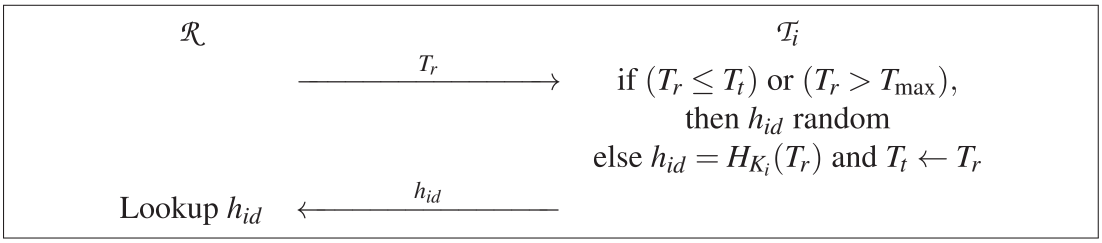

$\mathcal{R}$
$\mathcal{T}_i$
T_r
if $(T_r \leq T_t)$ or $(T_r > T_{\max})$,
then h_{id} random
else $h_{id} = H_{K_i}(T_r)$ and $T_t \leftarrow T_r$
h_{id}
Lookup h_{id}

Figure 2.12: RIP protocol.

Note that although this prevents the replay attack, the two aforementioned issues are still present.

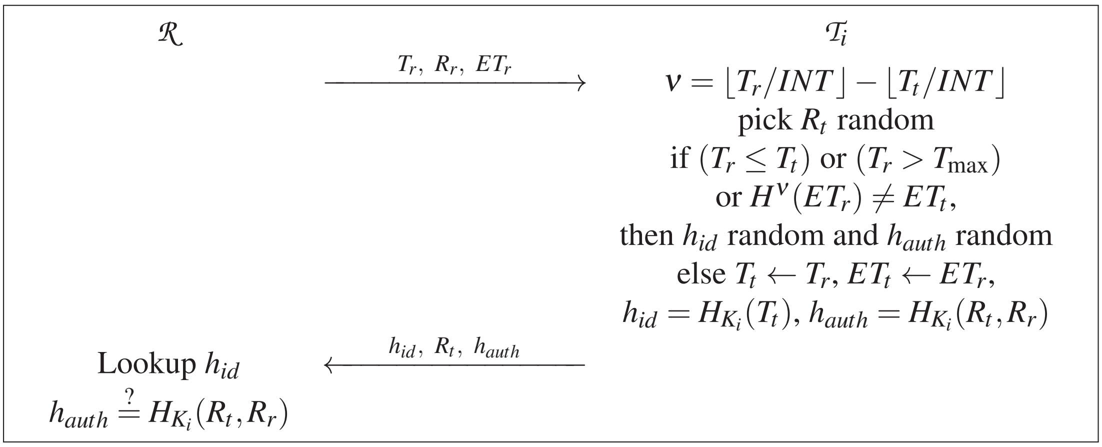

$\mathcal{R}$
$\mathcal{T}_i$
$T_r,\ R_r,\ ET_r$
$\nu = \lfloor T_r/INT \rfloor - \lfloor T_t/INT \rfloor$
pick R_t random
if $(T_r \leq T_t)$ or $(T_r > T_{\max})$
or $H^{\nu}(ET_r) \neq ET_t$,
then h_{id} random and h_{auth} random
else $T_t \leftarrow T_r$, $ET_t \leftarrow ET_r$,
$h_{id} = H_{K_i}(T_t)$, $h_{auth} = H_{K_i}(R_t, R_r)$
$h_{id},\ R_t,\ h_{auth}$
Lookup h_{id}
$h_{auth} \overset{?}{=} H_{K_i}(R_t, R_r)$

Figure 2.13: YA-TRAP* protocol.

In order to cope with DoS attacks, Tsudik proposed YA-TRAP*, which is illustrated in Fig. 2.13. DoS resistance is achieved by using a system-wide hash-chain. At setup, the system initializes a long Lamport-chain [89] of hashes, and sets the value ET_t of all tags to the last hash computed. Every INT counter ticks, a value of the hash-chain is popped, and the next one is used as ET_r. During an authentication session, a tag receiving T_r, R_r and ET_r will compute the number of intervals skipped since the last authentication (i.e. $\nu = \lfloor T_r/INT \rfloor - \lfloor T_t/INT \rfloor$), and will verify that the hash ET_r is the corresponding predecessor of ET_t by checking whether $H^{\nu}(ET_r) = ET_t$[7].

Note that DoS resistance in YA-TRAP* is limited by the magnitude of INT value. When ET_r is sent by the system it is no longer secret and it can be easily snooped on by the adversary. Therefore, the adversary can still incapacitate tags up to the upper

[7] Note that in [18], the authors mistakenly stated this check was $H^{\nu}(ET_t) = ET_r$.

duration of INT by querying the tag with the maximum possible T_r value within the current epoch.

All the aforementioned protocols do not provide forward-privacy because the long-term key of the tags are static. Tsudik introduces an additional operation for updating the keys of the tags. In this extension, which we denote YA-TRAP*&fwd hereafter, a tag takes ν times hash of the key for each authentication namely $K_i^\nu = H^\nu(K_i)$. With this modification, the tag's key is changed once per INT interval, and this brings ν additional hash operations on the tag-side.

2.4.1.2 Attacks on YA-TRAP*

In YA-TRAP*, the tag computes ν times the hash function depending on the difference between the T_t and T_r values. If the received T_r value is within the same interval than T_t, the tag computes no hash function for the interval check. If the difference between these two counters is large, the tag has to compute many hash functions. This leads to two potential attacks.

The first one is a traceability attack. It is simply that if a tag has not been authenticated in a long time, it is traceable due to the amount of time it spends computing the hashes. Distinction is thus possible between two tags in some situations.

The second one is a DoS. If an adversary sends a big T_r and whatever ET_r to a tag, the latter needs to compute a lot of hashes, even if it will eventually discard the request since the ET_r is not correct.

Depending on INT, this can make the authentication impossible due to the amount of time needed by the tag to complete its calculation. The parameter INT must be carefully chosen: the bigger it is, the less it mitigates the DoS already present in RIP+; and the smaller it is, the more it imposes a lot of computation on tags, leading to the two problems described above.

2.4.1.3 Other protocols

Another counter-based protocol is proposed by Burmester *et al.* called YA-TRAP+ in 2006 [62, 90]. A slightly modified version of this protocol is presented in [20] with a new name "RAP"). This protocol is very similar to O-RAP in terms of security properties. In [20] it is also stated that “O-RAP is simpler than RAP, at the cost of not supporting kill-keys.

The security for O-RAP is similar to that of RAP.” In particular, the two issues mentioned in Sect. 2.3.5 are also applicable to RAP. Additionally, in O-RAP a desynchronized tag is resynchronized automatically after each legitimate authentication, however RAP does not support automatic resynchronization. For these reasons, we only analyses O-RAP among those two similar protocols.

2.4.2 Discussion

Counter-based protocols, embodied by the YA-TRAP family, provide an interesting approach to constant-time identification. However, since the counter must be provided in the clear and, as such, is not authenticated, DoS attacks are extremely easy to accomplish and hard to prevent. YA-TRAP* attempts to alleviate this problem but at the same time introduces other weaknesses as indicated in Section 2.4.1.2.

2.5 Notes on Timing Attacks

Before starting to analyze the protocols we first want to highlight some facts about the *timing attack*. As emphasized in [67], in some cases it is possible for an adversary to use a timing attack to deduce information on a tag being authenticated. This kind of side channel attack is very easy to carry on, and although it lacks accuracy, it can very well be used for *tracking*.

For instance, in the OSK protocol that we mention later [66], tag identification time will depend on the desynchronization level of the target. Tags thus have a "fingerprint" which will only slightly change over time. This allows an adversary to easily trace its target.

For example, in the OSK protocol, an adversary can perform this attack without desynchronizing T:

- query T k times (k "reasonably" large),
- release T,

Table 2.2: IDs and next-IDs for C^2 protocol.

ID_1	ID'_1
ID_2	ID'_2
ID_3	ID'_3
$\vdots$	$\vdots$
ID_i	ID'_i
$\vdots$	$\vdots$
ID_N	ID'_N

- draw T_1, T_2,
- query T_1 and T_2 and output whichever takes the longest to be identified.

2.5.1 A case study on C^2: Countermeasures against timing attacks

In the original article [30], the search procedure on the reader-side is described as follows. For each tag the reader computes $H_1(ID_i \| N_T \| N_R)$ and compares it with the received value σ until it finds a match for $i = 1, \ldots, N$ where N is the number of tags in the database. In case no match is found, for each tag it generates the next-ID $ID'_i = H_2(ID_i)$, computes $H_1(ID'_i \| N_T \| N_R)$ and compares with σ until it founds a match. Table 2.2 depicts the possible IDs of the tags for a single authentication. Namely, if the $\mathcal{T}_i$ is synchronized, its ID is one of those in the first column, otherwise in the second column. According to the original paper, the reader only stores the first column. This search procedure leads the timing attacks as described in the previous subsection.

2.5.1.1 Search procedures on server-side to avoid timing attack

In the following we discuss some potential methods to mitigate timing attacks. In general, the objective is to avoid the adversary to predict the time spent for a given tag authentication.

Constant-time search: The most straightforward method is waiting until the worst-case execution time for each identification. Taking the example of C^2 protocol, even the identification of a tag is completed the reader waits until a certain constant-time to send the third message. This solution obviously mitigate the time

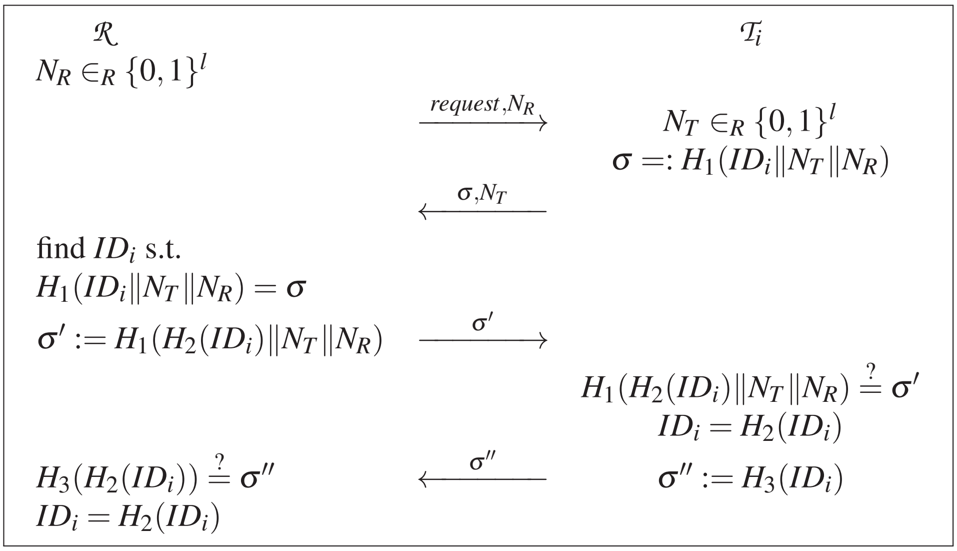

$\mathcal{R}$
$\mathcal{T}_i$
$N_R \in_R \{0,1\}^l$
$\xrightarrow{request, N_R}$
$N_T \in_R \{0,1\}^l$
$\sigma =: H_1(ID_i \| N_T \| N_R)$
$\xleftarrow{\sigma, N_T}$
find ID_i s.t.
$H_1(ID_i \| N_T \| N_R) = \sigma$
$\sigma' := H_1(H_2(ID_i) \| N_T \| N_R)$
$\xrightarrow{\sigma'}$
$H_1(H_2(ID_i) \| N_T \| N_R) \overset{?}{=} \sigma'$
$ID_i = H_2(ID_i)$
$H_3(H_2(ID_i)) \overset{?}{=} \sigma''$
$\xleftarrow{\sigma''}$
$\sigma'' := H_3(ID_i)$
$ID_i = H_2(ID_i)$

Figure 2.14: C^2 Protocol

attack however, the identification efficiency is worst. If we consider the reader does not store the second column, the wost-case execution time (i.e., the identification time for a single tag) is $3\tau_f \cdot N$ otherwise $2\tau_f \cdot N$, where τ_f denotes the execution time of the PRF functions. This method is also proposed in several papers like [17,18] as a simple solution to avoid the timing attacks.

Randomly starting-point selection: Another solution is to randomizing the "starting-point" of the search [67]. Namely, the reader chooses a random row in the first column to start the exhaustive search. Also preferably, it continuously choses random numbers uniformly in [1, N] (without choosing the same number). In this randomize search one can propose two different search procedure (i) first searching in the first column then if not found continuing with the second column, (ii) searching row by row. For the first method, (i) the average authentication time for a synchronized tag is $\tau_f \cdot N/2$, and for a desynchronized tag $3\tau_f \cdot N/2$. Thus, this method still allow tracing by using timing attack if an adversary desynchronizes a tag. For the second method, (ii) the average authentication time (i.e.,$3\tau_f \cdot N/2$) for a synchronized and desynchronized tag is not so distinguishable like previous one. However, timing attack is still an issue for this search method. To show that, let us first consider the existing of two tags in the system. The adversary desynchronize one of the tags and observes an authentication protocol. The adversary can distinguish the target tag if (a) the execution time is $2\tau_f$, she knows it is the desynchronized tag, (b) the execution time is τ_f she knows that it is the other one. If the execution time is $3\tau_f$ she cannot distinguish.

Random ID search: Another method is proposed in [67]. In this method the reader stores the IDs in the first and second column. During the online computation reader tests a random ID without considering whether it is a current or a next-ID. This searching procedure obviously prevent the timing attack. For this procedure the database storage increase from N to $2N$, and the average online computation is $\tau_f \cdot N$.

Random search with marked tags: To decrease both the database storage and the online computation we come-up with a new search procedure to thwart the timing attacks. In this method, reader does not store the all the ID's of the second column. For an authentication protocol, if reader receives the fourth message it is sure that the ID is updated by the tag so reader do so. Otherwise, the tag possible desynchronized, and reader stores the next-ID. So, the reader only stores $N+m$, where m is the number of marked tags as desynchronized. During an identification, reader randomly chooses an ID from among $N+m$ IDs. Using this searching procedure the timing attack obviously avoided thus an adversary cannot obtain any information by desynchronizing the tags. The average online computation is $\tau_f \cdot (N+m)/2$. If the number of desynchronized tags are not many in the system our method brings noticeable efficiency over the previous search procedures.

2.6 Comparison

In this section we summarize most of the protocols we analyzed and compare them on several criteria, as shown in Table 2.3. We evaluate the schemes that provide sub-linear complexity, at least during the normal case online interaction, and that provide at least user privacy (not necessarily forward-privacy). We also include those for which we highlight new weaknesses in this paper. For clarity reasons, we provide additional remarks (superscripted capital letters in the table) which is given after the table.

The protocols using shared secrets, although presenting alluring identification efficiency, have important security and privacy problems, as stated earlier. Nonetheless, future ideas might lower the impact of tag compromise, and this approach remains interesting. The counter-based protocols, embodied by the YA-TRAP family, seem to be promising as well, but also have issues regarding privacy. Their usability (a maximum of one authentication per counter tick) might be a problem in some applications too. Although not ideal, the protocols based on hash-chains seem to be the most solid solutions to date, among the protocols we analyzed. OSK/AO and OSK/BF provide forward-privacy but have desynchronization issues due to the finite size of the chains. O-RAP is also somewhat easier to manage on the server-side. However, if we consider an adversary capable of performing timing analysis, it has a lower privacy. O-FRAP also brings some forward-privacy to O-RAP (but not completely as we point out in Sect. 2.3.5.3).

Table 2.3: Comparison of the protocols analyzed in this article. Letters in brackets link to comments described below.

CLASS	SHARED SECRETS			HASH-CHAINS					COUNTER-BASED		
PROTOCOL	CHT plain	CHT with auth	CTI	OSK	OSK/AO with Auth	OSK/BF with Auth	O-RAP	O-FRAP	RIP+	YA-TRAP*	YA-TRAP* &fwd
MAIN REFERENCE	[2]	[2]	[16]	[66]	[19, 80, 84]	[22]	[62]	[21]	[18]	[18]	[18]
YEAR OF PUBLICATION	2009	2009	2010	2003	2005	2008	2006	2007	2007	2007	2007
Identification/ Authentication	auth.[A]	auth.	auth.	id.	auth.	auth.	auth.	auth.	auth.	auth.	auth.
Off-line Computation Complexity[B]	0	0	$O(NC)$[C]	$2N$[C]	$\frac{NM^2}{2}$ ([19])[D]	$2NM$[C]	0	0	N/counter update[E]	N/counter update[E]+ Lamport Chain	N/counter and key update[E] + Lamport Chain
Normal case online complexity	$O(\sqrt{N})$	$O(N^a)$	4	2	$O(N^{2/3})$[F]	$M(\varepsilon N+3)$ on average	1	2	0[E]+1[G]	0[E]+1[G]	0[E]+1[G]
Desynchronized case online complexity	N/A	N/A	N/A	lower than $2N(M-1)$[H]	$O(N^{2/3})$[F]	$M(\varepsilon N+3)$ on average	$O(N)$	$O(N)$	out of order after desync.[H]	out of order after desync.[I]	out of order after desync.[I]
Memory Complexity	$2\sqrt{N}$	$2N^a+N$	$O(N)$[J]	N	$O(N^{2/3})$[F]	$\frac{NM\log\varepsilon}{-\log^2 2}$	$2N$	$3N$	N[E]	N[E]	N[E]
Tag Computation	2 PRFs + 1 Nonce	3 PRFs + 1 Nonce	5 hashes	2 hashes	3 hashes	3 hashes	2 hashes	4 hashes	2 hashes + 1 Nonce	$v+2$ hashes + 1 Nonce	$2v+2$ hashes + 1 Nonce
Tag Resources	PRF, PRNG	PRF, PRNG	PRNG, hash fun.	Hash func.	Hash func.	Hash func.	Hash func.	Hash func.	PRNG, hash fun.	PRNG, Hash func.	PRNG, hash fun.
Privacy	no	no★	no★	yes[K]	yes◊,[K]	no★	no★,[K]	no★,[K]	not private after desync.	not private after desync.★,[L]	not private after desync.★,[L]
Forward-privacy	no	no	no[M]	yes[K]	yes[K]	no[M]	no	no★	no	no	no[N]
Desynchronization resistance	N/A	N/A	yes	yes up to M consecutive[O]	yes up to M consecutive[P]	yes up to M consecutive[O]	no[Q]	no[Q]	no[R]	yes[S]	yes[S]
Impersonation Resistance	no★	no	yes	N/A	yes	yes	yes	yes	yes	yes	yes

★ : Weaknesses discovered in this paper. ◊ : Weaknesses discovered and fixed in this paper.

[A] Although the authors implicitly consider it to be an identification protocol (because of the existence of an *authentication* extension), we denote it by an authentication protocol, since the reader sends a nonce, and since the protocol is at first designed to cope with impersonation.

[B] Excluding key generation.

[C] Done during the setup.

[D] Done each time one tag reaches M authentications since the last table update.

[E] The whole hash table is periodically updated or can be precomputed for forecoming counters.

[F] Using rainbow tables. The complexity provided is an example, but the identification complexity can be set anywhere between $O(1)$ and $O(N)$ according to the memory available for the trade-off (see Sect. 4.1.1 for discussion).

[G] Additional hash for authentication.

[H] The tag might not be identified/authenticated by the reader.

[I] The tag cannot be authenticated within the time interval. It gets resynchronized in the next one.

[J] Can be big due to constant terms (see Sect. 2.2.3 and [16]).

[K] Private if and only if the adversary is not able to tell whether the protocol session was successful.

[L] However, private after resynchronization. The tag resynchronized automatically at each interval start.

[M] Since it is not private.

[N] Not forward private until the last ET update.

[O] After M desynchronizations, the tag owner can go to some central office to fix the issue.

[P] Including legitimate authentications. If the tag owner goes to the office the tag can be resynchronized in the next precomputation. However, if the number of the illegitimate authentications is less than M, then the resynchronization of the tag will be done automatically during the next update.

[Q] Tags can be desynchronized, but resynchronize automatically after each authentication.

[R] Either up to $T_{\max}$ (tag becomes useless) or with a smaller counter (for to traceability and replay).

[S] Tags can be easily desynchronized within a time interval (e.g. one day), making them unusable and/or traceable. Tags are automatically resynchronized at the beginning of each interval. Tags have limited lifetime because the Lamport chain must have a start (although the system can be initialized with a really big hash-chain).

3. TIME-MEMORY TRADE-OFF METHOD

In this chapter, we briefly recall the required background on *Time-Memory Trade-off* (TMTO) method proposed by Hellman in [83]. After that we explain the TMTO technique but make no attempt at providing a complete survey of it. In particular, the analysis of perfect table trade-offs is described in this section, but we recommend to read [91] an advanced introduction.

3.1 From Extreme Cases to Time-Memory Trade-off Method

The basic idea of the time-memory trade-off method is to find a trade-off that has a lower online computation complexity than exhaustive key search, and lower memory complexity than a table look-up(exhaustive storage). In what follows we describe exhaustive search and table look-up method to warm-up the idea of TMTO.

3.1.1 Exhaustive search method

The first naive method to find a preimage of a given value is trying all possible inputs, and calculating the output values by using the H function and checking whether they yield the given value. This method, which is known as *exhaustive search*, requires $N/2$ operations in average to find a preimage where $N = |A|$. However if the size of the problem (i.e., N) is large enough the inversion could be almost infeasible for polynomial time.

3.1.2 Table look-up method

To mitigate the online computing time issue of the exhaustive search method, one may first construct a look-up table including all the preimage values (exhaustive storage). Afterwards, any preimage finding task can be accomplished via one table look-up operation which requires a negligible amount of time. Also, the precomputation process requires an effort equal to exhaustive search and is to be performed once.

Although table look-up is quite fast, it requires extreme amounts of memory for the functions acting on large sets. The comparison of exhaustive search and table look-up methods are depicted in Table 3.1. As it can be seen in the table there is a huge gap between the solution time and the required memory. To over come this problem, a method that provide trading between memory and time would be particularly useful. In what follows we go through the Time-Memory Trade-off method.

Table 3.1: Comparison of exhaustive search and table look-up methods.

	Exhaustive search	Table look-up
Precomputation	0	N
Online computation	$N/2$	0
Memory (storage)	0	N

3.2 Time-Memory Trade-off Method and Perfect Tables

Many searching problems allow time-memory trade-offs. That is, if there are N possible solutions to search over, the time-memory trade-off allows the solution to be found in T operations (time) with M words of memory, provided the time-memory product $T \times M$ equals N [11]. The concept of TMTO method [83] was originally suggested by Hellman in 1980. The time-memory trade-off described in Hellman's work applies to inverting any one-way function. In TMTO method a *pre-computation table* is constructed only once and only a subset of generated values is kept.

The significant point about the storage is that only the first and the last elements of each chain are stored, providing so a table. However different than the exhaustive search and table look-up methods, TMTO is a probabilistic method, i.e., the search operation may not find a preimage even if there exists one.

A major improvement over Hellman's original TMTO method [83] was given by Oechslin [3]. The precomputation table for this method is structurally different than the Hellman's TMTO. Oechslin suggested to use a single table of size $\rho W \times W$ satisfying $\rho W^2 = N$ which he called a rainbow table. Unlike the Hellman's reduction functions which are constant for each table in rainbow table different reduction functions in each column. By doing so, the online search computations decreases by a factor of 2 compared to Hellman's TMTO. Another significant advantage is that, two different

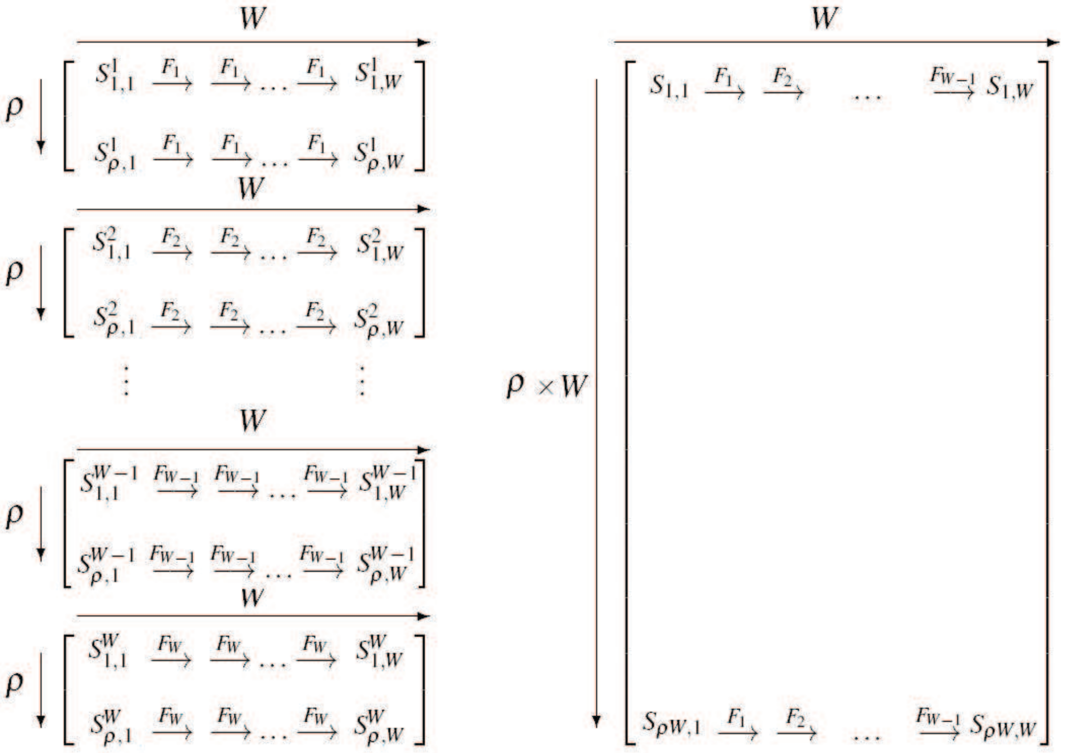

W
ρ
$S^1_{1,1} \xrightarrow{F_1} \xrightarrow{F_1} \dots \xrightarrow{F_1} S^1_{1,W}$
$S^1_{\rho,1} \xrightarrow{F_1} \xrightarrow{F_1} \dots \xrightarrow{F_1} S^1_{\rho,W}$
$S^2_{1,1} \xrightarrow{F_2} \xrightarrow{F_2} \dots \xrightarrow{F_2} S^2_{1,W}$
$S^2_{\rho,1} \xrightarrow{F_2} \xrightarrow{F_2} \dots \xrightarrow{F_2} S^2_{\rho,W}$
$S^{W-1}_{1,1} \xrightarrow{F_{W-1}} \xrightarrow{F_{W-1}} \dots \xrightarrow{F_{W-1}} S^{W-1}_{1,W}$
$S^{W-1}_{\rho,1} \xrightarrow{F_{W-1}} \xrightarrow{F_{W-1}} \dots \xrightarrow{F_{W-1}} S^{W-1}_{\rho,W}$
$S^{W}_{1,1} \xrightarrow{F_W} \xrightarrow{F_W} \dots \xrightarrow{F_W} S^{W}_{1,W}$
$S^{W}_{\rho,1} \xrightarrow{F_W} \xrightarrow{F_W} \dots \xrightarrow{F_W} S^{W}_{\rho,W}$
$\rho \times W$
$S_{1,1} \xrightarrow{F_1} \xrightarrow{F_2} \dots \xrightarrow{F_{W-1}} S_{1,W}$
$S_{\rho W,1} \xrightarrow{F_1} \xrightarrow{F_2} \dots \xrightarrow{F_{W-1}} S_{\rho W,W}$

Figure 3.1: Structural differences between Classical Hellman's tables (on the left) and Rainbow tables (on the right) [3].

chains can merge only if they have the same value at the same position in the chain. In a single table this ensures to detect the collisions easily by only looking at the end of the table and makes possible to generate much larger tables. In our implementation we also use the rainbow TMTO tables. Figure 3.1 shows the structural differences between classical Hellman's tables and rainbow tables.

Perfect tables: In [92], Borst et al. suggested to clean the tables by discarding the colluding and cycling chains since merging chains decreases the efficiency of the trade-off tables significantly. This kind of tables, called perfect table, considerably degrades the required memory. The more efficient usage of storage leads to better performance during the online TMTO calculations, at the expense of higher pre-computation cost. The removal of redundancies cannot be done as easily with the classical Hellman algorithm, but the rainbow table method provides a perfect table version much more easily. Subsequently, in [91] Avoine et al. provides the characterization and some improvements on perfect TMTO tables.

3.3 Optimal Configurations for Perfect Tables

For our implementation in Chapter 4 we use some theoretical results to set the optimal configurations and get the most appropriate settings. We first give following results based on perfect tables are studied in [84, 91, 93, 94]. We start with the success rate of a single rainbow table which is not perfect is given in [3].

$$1-\prod_{i=1}^{W}\left(1-\frac{\rho_i}{N}\right),$$

where ρ_i is the number of different "key values" (or later we say tag responses) in column i, and W is the window size (i.e., the chain length in a single rainbow row). With perfect rainbow tables, we have $\rho_i = \rho$ for all i such that $1 \leq i \leq W$. Therefore, the success rate of a single perfect rainbow table can be given as

$$P_{rbw} = 1-\left(1-\frac{\rho}{N}\right)^{W}, \qquad \textbf{(3.1)}$$

The largest possible perfect tables results fastest cryptanalysis and better success rate. Thus the un-merged rainbow chain rows should be the maximum as much as possible in a table. We call $\rho_{max}(W)$ as the maximum number of rainbow rows that can be generated without merges with a given window size W. In [3] and [91] it is stated that $\rho_{max}(W)$ can be obtained by calculating the number of independent elements at W^{th} column if we start with N elements in the first column. For the sake of notation clarity we say $\rho_{max}(W) = \rho_W$. Thus in order to generate perfect table we choose $\rho_1 = \rho(W = 1) = N$. The following equations are excerpted from [3].

$$\rho_2 = N\left(1-\left(1-\frac{1}{N}\right)^{\rho_1}\right) \approx N\left(1-e^{-\frac{\rho_1}{N}}\right),$$

also

$$\rho_{j+1} = N\left(1-e^{-\frac{\rho_j}{N}}\right), \qquad \textbf{(3.2)}$$

where $0 < j < W$.

By using Taylor approximation of the exponential it can be deduced as

$$\rho_{j+1} \approx N\left(\frac{\rho_j}{N} - \frac{\rho_j^2}{2N^2}\right) = \rho_j - \frac{\rho_j^2}{2N^2}.$$

After transforming this expression into a differential equation and doing some calculations the following approximation can be obtained as stated in [91].

$$\frac{d\rho_j}{dj} = \frac{\rho_j^2}{2N},$$

the solution of ρ_j is

$$\rho_j = \frac{2N}{j+c},$$

where c is the constant from differential solution. When $\rho_1 = N$ we get $c = 1$ which yields the following approximation [1] that is the maximum number of rainbow rows that can be generated without merges

$$\rho_{max}(W) = \rho_W \approx \frac{2N}{j+1}. \qquad \textbf{(3.3)}$$

The number of preimages that can be found in a table is the number of distinct entries in the first W columns which we call the *coverage* of the table. The expected maximum coverage (i.e., expected maximum probability of success) for a single perfect rainbow table that have ρ_{max} columns can be obtained from (**3.1**) and (**3.3**) as follows

$$T(\rho_{max}) = 1 - \left(1 - \frac{\rho_{max}}{N}\right)^W = 1 - \left(1 - \frac{2}{W+1}\right). \qquad \textbf{(3.4)}$$

The Formula (**3.4**) can be also represented by using the following approximation which gives the theoretical bound for success probability of a table

$$T(\rho_{max}) \approx 1 - e^{-\frac{2W}{W+1}} = 1 - e^{-2} \approx 0.8647. \qquad \textbf{(3.5)}$$

[1] In [91] Formula (**3.3**) corrects the erroneous done in [93, 94].

From Equation **3.5** we know that if we use only one table the maximum achievable coverage is bounded approximately by 86%. To have a higher coverage (e.g., %99) we need to generate more tables. We denote $T(\rho_W, k)$ the overall coverage of k perfect rainbow tables in which the number of un-merged rows at each table is ρ_W at window size W. The following theorem states the coverage of k tables under the assumption that each reduction function defines an independent random function.

Theorem 3.3.1. *The overall coverage of a single table of size $\rho_W \times W$ is $T(\rho_W, 1)$, then the overall coverage of k independent tables of size $\rho_W \times W$ is $(1-(1-T(\rho_W, 1)))^k$.*

Proof. The probability that a point is not in one table is $1-T(\rho_W, 1)$. The probability that a point is not in k different tables is $(1-T(\rho_W, 1))(1-T(\rho_W, 1))\dots(1-T(\rho_W, 1))$, which is equal to $(1-T(\rho_W, 1))^k$. If we subtract this expression from 1 we obtain the probability that a point is in one of the k tables, which is: $(1-(1-T(\rho_W, 1)))^k$. □

To choose the most appropriate settings for TMTO we should first decide on the desired total coverage when we generate more than one table. In [91] it is shown that the smallest number of tables needed for a trade-off only depends on the desired success rate. The number of tables needed for a given desired total coverage is

$$k = \left\lceil \frac{-ln(1-T(\rho_W, k))}{2} \right\rceil \qquad \textbf{(3.6)}$$

Finally window size can be calculated in terms of the other parameters as follows

$$W = \left\lceil -\frac{N}{k\rho_W} ln(1-T(\rho_W, k)) \right\rceil . \qquad \textbf{(3.7)}$$

4. IMPLEMENTATION OF A FORWARD SECURE AND EFFICIENT RFID AUTHENTICATION PROTOCOL

In this chapter, we first briefly recall the protocol proposed by *Ohkubo, Suzuki and Kinoshita* (OSK) [66] already mentioned in Sect.2. After that we re-introduce the improved protocol called OSK/AO suggested by Avoine *et al.* in [19, 80] and the specific time-memory trade-off technique that removes the scalability issue of OSK scheme. After that we give the notations and a generic pseudo-algorithm of software part of our implementation. Then we define the environment and the gadgets used in our implementation. Finally, we demonstrate our implementation for some specific settings and give their results according to.

4.1 Ohkubo, Suzuki, and Kinoshita's Protocol

OSK [66] is a well-known synchronized identification protocol, and was one of the earliest of its kind.

The basic idea of this protocol is to modify the seed (an identifier key) of the tag each time it is queried by a reader. The tag updates its seed autonomously even it is queried by an rogue reader, using two hash functions $\mathcal{G}$ and $\mathcal{H}$ as described below.

System Setup: Each tag $\mathcal{T}_i$ of the system is initialized with a randomly chosen secret S_i^0 which is the identifier seed of the tag. We do not assume that the tags are tamper resistant. The back-end system also stores all the seeds of the tags in its database.

Interrogation: When queried by a reader, a tag answers with the hash of its *current* secret, such that $\sigma = \mathcal{G}(S_i^\ell)$, and immediately updates it using another hash function: $S_i^{\ell+1} = \mathcal{H}(s_i^k)$.

Search & Identification: When receiving an answer, the reader sends it to the back-end system. Then it searches in its database for an initial secret S_j^0 that leads to σ, in other words, it checks whether there exists i and j such that $\mathcal{G}(\mathcal{H}^i(S_j^0)) = \sigma$. To do that, from each of the n initial secrets S_j^0, the reader computes the hash chains as shown in Fig. 4.1 until it finds a value matching σ, or until it reaches a given maximum limit L on the chain length. An overview of the protocol is shown in Fig. 4.2.

The value $\sigma = \mathcal{G}(S_i^\ell)$ does not allow an eavesdropper to learn the identity of $\mathcal{T}_i$. However, since a tag updates its secret regardless of the success of the identification, a rogue reader initiating the protocol with $\mathcal{T}_i$ will make it update its secret. An adversary initiating lots of instances of the protocol with $\mathcal{T}_i$ will perform a *desynchronization* denial-of-service attack. Indeed, the reader would then need to compute a lot of hashes to identify $\mathcal{T}_i$. To prevent this, the length of the hash chains have to be bounded, i.e., the reader stops its search after L hashes per tag. This protection has the following drawback: an adversary skimming a tag L times makes it unable to be identified by the system, and thus traceable.

Beside this traceability issue, and although the protocol is very efficient when all tags are synchronized, the worst-case complexity of the search makes the protocol unsuitable for most practical systems.

$$\begin{array}{ccccccccc}
S_1^0 & \longrightarrow & r_1^0 & r_1^1 & r_1^2 & \dots & r_1^{L-1} & r_1^L \\
S_2^0 & \longrightarrow & r_2^0 & r_2^1 & r_2^2 & \dots & r_2^{L-1} & r_2^L \\
\dots & \longrightarrow & \dots & \dots & \dots & \dots & \dots & \dots \\
S_i^0 & \longrightarrow & \dots & \dots & \dots & \boxed{r_i^\ell = \mathcal{G}(\mathcal{H}^\ell(S_i^0))} & \dots & r_i^L \\
\dots & \longrightarrow & \dots & \dots & \dots & \dots & \dots & \dots \\
S_n^0 & \longrightarrow & r_n^0 & r_n^1 & r_n^2 & \dots & r_n^{L-1} & r_n^L
\end{array}$$

Figure 4.1: OSK table: Chain of hashes in the OSK protocol.

The authors later introduced in [82] some ideas to improve the efficiency of the search at the cost of lowering privacy. Since strong privacy is one of the design goals of OSK, we will not consider them further.

4.1.1 OSK/AO protocol

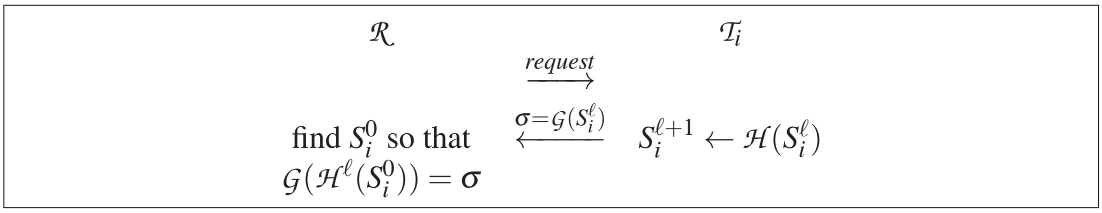

$\mathcal{R}$
$\mathcal{T}_i$
$\xrightarrow{request}$
find S_i^0 so that
$\mathcal{G}(\mathcal{H}^\ell(S_i^0)) = \sigma$
$\xleftarrow{\sigma=\mathcal{G}(S_i^\ell)}$
$S_i^{\ell+1} \leftarrow \mathcal{H}(S_i^\ell)$

Figure 4.2: OSK protocol.

Avoine and Oechslin propose in [80] to apply Hellman's time-memory trade-offs [83] to the search procedure of OSK. The complexity of the search procedure varies from $O(1)$ to $O(N)$, depending on the amount of memory we are willing to devote to the time-memory trade-off[1].

Avoine, Dysli, and Oechslin also suggest in [19] a variant of OSK that ensures authentication as OSK is originally designed to provide private identification only (i.e., it does not resist to replay attacks). To do so, they suggest using nonces: instead of simply sending a *request* message, the reader sends a nonce r, and the tag answers $\mathcal{G}(S_i^\ell \oplus r)$ along with $\mathcal{G}(S_i^\ell)$.

TMTO Approach Now we briefly describe the specific time-memory trade-off technique introduced in [19, 80].

In this technique there are two main functions namely a *response generating function* $\mathcal{F}$ and a *reduction function* $\mathcal{R}$. $\mathcal{F}$ takes two indexes as an input (i.e., tag identifier index and life time index) and outputs a tag response coresponding to the input such that

$$\mathcal{F} : (i,\ell) \mapsto \mathcal{G}(\mathcal{H}^\ell(S_i^0)) = r_i^\ell$$

$\mathcal{R}$ works as complement of $\mathcal{F}$ which takes a response value as the input and produces arbitrary indexes. So that

$$\mathcal{R} : r_i^\ell \mapsto (i',\ell')$$

where $1 \leq i,i' \leq n$, and $0 \leq \ell,\ell' \leq L$

[1] The authors mention that, for instance, a complexity of $O(N^{2/3})$ can be reached with a memory of size $O(N^{2/3})$.

By alternating these two functions starting with a initial value, a chain of tag responses and indexes can be built. Several chains of a given window size are generated, most outputs of F will appear at least once in any chain. A TMTO table constructed by only storing the initial and last elements of each chain. As stated Sect. 3 this table includes most of the responses of the tags but not all and limited to approximately 86% coverage. To provide a higher coverage close to, but not exactly 100%, many rainbow tables should be generated by using different reduction functions.

4.2 Notations

The notations used in the pseudo-algorithm are given below.

- n : The number of tags in the system.
- L : The length of life time of a tag in the system.
- $\mathcal{H}$, $\mathcal{G}$: Collision resistant one-way hash functions.
- S_i^ℓ : The secret seed of the i-th tag at ℓ-th life time where $1 \leq i \leq n$, and $0 \leq \ell \leq L$.
- W : The length of window size of a rainbow table.
- $rapid\mathcal{H}(i,\ell)$: A function which computes the ℓ-th seed of the i-th tag such that $S_i^\ell = \mathcal{H}^\ell(S_i^0)$. This function uses a pre-computed seed table to compute hashes faster. The construction of this function is demonstrated in Algorithm 1.
- κ : The length of interval between hash indexes. This parameter is needed for computing rapid hashes.
- $seed[i][j]$: A pre-computed two-dimensional array which stores $j \times \kappa$-th hash value of i-th tag's initial seed ($\mathcal{H}^{j\times\kappa}(S_i^0)$). For instance, let $\kappa = 6$, $i = 1$ and $j = 6$, then $seed[1][6]$ stores $S_1^{36} = \mathcal{H}^{36}(S_1^0)$. This array is used during evaluation of $rapid\mathcal{H}(i,\ell)$.
- $\mathcal{F}(i,\ell)$: The response generating function inputs two parameters the tag identifier index and the life time of the tag. This function uses the rapid-hash function. It outputs a tag response such that $\mathcal{F}(i,\ell) = \mathcal{G}(rapid\mathcal{H}(i,\ell))$.

- $Table_t$: The t-th TMTO table which stores the initial and end points (indexes) of the TMTO table. The table construction is shown in Algorithm 2.
- $\mathcal{R}_w^t(val)$: For w-th column of the t-th table, a simple reduction function which maps input val into a output with smaller size.

4.3 Our Algorithm

First, the system randomly generates the initial seeds for all the tags such that $S_i^0 \in_R \{0,1\}^\gamma$ where $1 \leq i \leq n$, and γ is the length of seeds. The system defines a κ parameter then computes the interval seed values of all the tags. After that all the seed values are stored into a two dimensional array such that $seed[i][j] := \mathcal{H}^{j\times\kappa}(S_i^0)$ where $j = 0,1,2,\ldots$ and $0 \leq j \times \kappa \leq L$.

Now, for a given i-th seed, the ℓ-th rapid-hash computation of the seed is presented in Algorithm 1. The algorithm requires only at most κ hashes by the help of the precomputed seed table. Whenever κ decreases, the memory usage increases but the on-line computation decreases.

Algorithm 1 Compute $y = rapid\mathcal{H}(i,\ell)$

```
Require: $1 \leq i \leq n, 0 \leq \ell \leq L$
Ensure: $y = S_i^\ell$
  $y \leftarrow seed[i][\ell \div \kappa]$
  $a \leftarrow \ell \ mod \ \kappa$
  while $a \neq 0$ do
    $y = \mathcal{H}(y)$
    $a \leftarrow a - 1$
  end while
  return $y$
```

Algorithm 2 shows the processes to construct a single rainbow TMTO table. For the construction, only two parameters are needed i.e., the number of trials and table number. The starting points of TMTO table (i.e., the index numbers of the tags and the life time) are fed into the $\mathcal{F}$ function sequentially. The output is actually a response of a tag in the system is fed into the reduction function which outputs arbitrary indexes (i, j), where $1 \leq i \leq n$, and $1 \leq \ell \leq L$. For a single chain this process is repeated consecutively up to a pre-defined window size, then the initial and end-points (i.e., an arbitrary index) are stored in the table. Each generated end-point is compared in the table, if a merge is not found the end point and the corresponding initial point are inserted to the table. To generate a perfect table $n \times L$ chains should be tried. Finally the resulting table is sorted in order to make the identification process faster.

Algorithm 2 Construction of Table (ℓ, TableNo)

```
Require: $1 \leq \ell \leq L$ , TableNo $\geq 1$
  $table \leftarrow \{\emptyset\}$
  for $i = 1$ to $n$ do
    for $j = 0$ to $\ell$ do
      $nextResp \leftarrow \mathcal{F}(i, j)$
      for $w = 1$ to $W - 1$ do
        $z[\,] \leftarrow \mathcal{R}_w^{TableNo}(nextResp)$
        $nextResp = \mathcal{F}(z[0], z[1])$
      end for
      $z[\,] \leftarrow \mathcal{R}_W^{TableNo}(nextResp)$
      if $z \notin table$ then
        add the record $\{(i, j); (z[0], z[1])\}$ into $table$
      end if
    end for
  end for
  sort $table$ by $z$ values
  return $table$
```

Finally, Algorithm 3 shows the identification process of a tag by extracting the pre-image of a given response by the help of TMTO tables. This part of the system runs during the online interaction with tags. Assume that we are searching a pre-computation table ($Table_t$) and for a response from a tag ($TagResp$). First, $TagResp$ is fed to the reduction function R_W^t and search among the end points of the TMTO table. (i) If a match is found the corresponding initial point is iterated as explained in Algorithm 2 upto R_W^t and get a candidate the respons. If the candidate response is equal to $TagResp$ then identification is completed otherwise (ii) $TagResp$ fed into R_{W-1}^t , $\mathcal{F}$, R_W^t consecutively. As previously done, the output value search among the end points of the TMTO table and the same process is carried as mentioned above.

4.4 Implementation Environment and Some Experiments

In this section, we define the experimental environment and the gadgets used in the experiments for the implementation. As a server, we use a Windows Vista machine

Algorithm 3 Identify ($Table_t$, TagResp)

```
Require: TagResp $\in \{0,1\}^\gamma$, $t \geq 1$
Ensure: TagResp $\leftarrow \mathcal{G}(y)$
  for $l = W$ down to 1 do
    $nextResp \leftarrow$TagResp
    for $i = l$ to $W - 1$ do
      $z[\,] \leftarrow \mathcal{R}_i^t(nextResp)$
      $nextResp \leftarrow \mathcal{F}(z[0], z[1])$
    end for
    $z[\,] \leftarrow \mathcal{R}_W^t(nextResp)$
    if $z \in Table_j$ then
      $\{z'; z\} \leftarrow Table_t(z)$
      $nextResp \leftarrow \mathcal{F}(z'[0], z'[1])$
      for $w = 1$ to $l - 1$ do
        $tz[\,] \leftarrow \mathcal{R}_w^t(nextResp)$
        $nextResp \leftarrow \mathcal{F}(tz[0], tz[1])$
      end for
      if $nextResp = TagResp$ then
        return $true$
      end if
    end if
  end for
  return $false$
```

having Intel 3.16GHz Core2 Duo processor and 4GB RAM as a RFID server. The information of the tags and data tables are stores in this part. As an RFID reader we have the OMNIKEY 5321 dual interface PC-linked reader that reads/writes to both a 13.56 MHz contactless smart card. The reader supports contactless smart cards with up to 848 kbps in the ISO 14443 transmission mode. For the tags, we work on professional version of ZeitControlers basic card $ZC7.5$ $(ZC - Basic)$ which is a programmable processor card as hardware environment for protocol implementation. It has a microcontroller with 32kB EEPROM that holds its own operating system (OS) and tags data and 2.9kB RAM It supports, $T = CL$-type a contactless protocol, as defined in ISO/IEC 14443. The compiler compiles the source codes into P-code which is machine code and machine-independent. We used Java language to run the protocols and provide the communication between the RFID reader and the server.

We work on a RFID system, where the initial settings are chosen as below.

- $n = 2^{20}$, $L = 2^7$, $\kappa = 6$.

- $\mathcal{H}(S_i)$: The first 64 bits of the standard SHA1 (S_i).

- $\mathcal{G}(S_i)$: The first 64 bits of the standard SHA1 $(S_i||1)$.

- $\mathcal{R}_i^t(val)$: It is defined in Algorithm 4.

To construct a rainbow table each column uses a different reduction function so that when two chains collide in different columns, they do not merge. If they collide at the same column it is easy to detect the merge and to discard the chain. In our system, for each reduction function of a column in each table, a random nonce is generated. These nonce are stored into two dimensional array (tnonce[ ][ ]). The construction of our reduction functions are given in Algorithm 4.

Algorithm 4 Compute $\mathcal{R}_i^t(val)$

```
Require: $t \geq 1, i \geq 1$
Ensure: $z[0] \in \mathbb{Z}_N, z[l] \in \mathbb{Z}_L$
  $z[0] \leftarrow val$ & 0xffffffff
  $z[1] \leftarrow (val >> 32)$
  $z[0] = z[0] \oplus$ tnonce$[i][t]$ $mod\ n$
  $z[1] = z[1] \oplus$ tnonce$[i][t]$ $mod\ L$
  return $z$
```

In order to ensure that our experimental outcomes are compatible with theoretical results we give the following example. Let us aim to identify a tag with a probability of at least 0.9996, so we first compute the number of perfect tables according to Equation **3.6**. We get the required number of tables as follows.

$$\begin{aligned} \#oftables &= \left\lceil \frac{-ln(1-0.9996)}{2} \right\rceil \\ &= 4 \end{aligned}$$

Now, we calculate the optimal windows size W according to Equation **3.7**. We want to occupy approximately 128 MB memory (RAM) for raw data of all tables in order to make our system to be realized in many devices. A row in a table consists of four indexes, two of them 20-bit and two of them 7 bit. One 20-bit and 7-bit can be stored in a 32-bit integer, so the row requires only two 32-bit integers (8 Byte) [2]. Thus, a table

[2] For this example, there are at least four unused bits. It could be used to increase life time or number of tags as needed.

can store $\rho_W = \frac{32\times 2^{20}}{8} = 2^{22}$ distinct rows. The windows size for a table is computed as (see Equation **3.7**):

$$\begin{aligned} W &= \left\lceil -\frac{2^{27}}{4\times 2^{22}} ln(1-0.9996) \right\rceil. \\ &= 64 \end{aligned}$$

We implement some perfect TMTO tables with different window sizes. To generate perfect tables we compute $n \times L = 2^{27}$ rainbow chains which is our total problem size. We test the coverage of the each settings by simulating Algorithm 3 (i.e., identification proses of OSK/AO with random tags). For each settings, the simulation is run $100,000$ times. The experimental results are depicted in Table 4.1. $T(\rho_W, k)$ denotes the overall coverage of k perfect rainbow tables in which the number of un-merged distinct rows in each table is ρ_W at window size W.

Table 4.1: Simulation results

	W	ρ_W (10^6)	Memory (MB)	Time (ms)	$T(\rho_W, 1)$	$T(\rho_W, 4)$
Settings-I	16	14.29	436	0,219	0.8322	0.9992
Settings-II	32	7.67	234	0,859	0.8467	0.9994
Settings-III	64	3.99	122	3.077	0.8552	0.9996
Settings-IV	128	2.21	67	9.929	0.8596	0.9996
Settings-V	256	1.03	31	39.055	0.8608	0.9996
Theoretical	$*$	-	-	-	0.8647	0.9997

Table 4.1 shows that by increasing the window size the coverage increases as natural and approaches the theoretical bound shown in **3.5**. When the window size increases, the memory for pre-computation decrease but on-line computation increase. On the other hand, provided the fact that the window size is 64 or more, the effect of window size is not significant to the overall coverage in the aggregation of four tables. Hence, choosing a window size of 64 could be reasonable.

Also one may aim to reduce the overall pre-calculation effort. Thus, the number of rows that are tried in the first column can be reduced during the pre-computation of a table. Figure 4.3 depicts the effect of increasing the trials in the first column (i.e., ρ_1) to the number of un-merged distinct chains (i.e., ρ_W) at the column W. From this figure

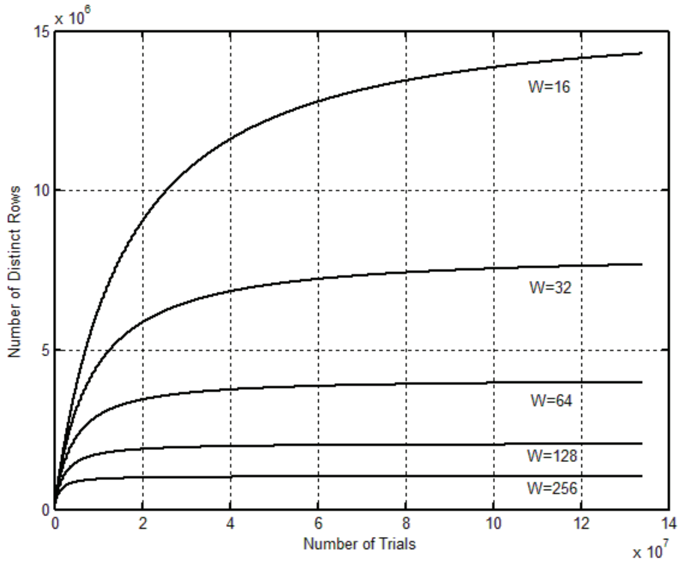


Figure 4.3: Number of Trials vs Number of Distinct Chains for Several Windows Sizes

it can be seen that after some points the advantage of increasing ρ_1 does not effect the final size of the table significantly.

5. CONCLUSION

The recent advent of ubiquitous technologies has raised an important concern for citizens: the need to protect their privacy. So far, this wish was not heard of industrials, but national and international regulation authorities, as the European Commission recently published some guidelines to enforce customers' privacy in RFID systems: "Privacy by design" is the way to be followed as stated in EC Recommendation of 12.5.2009. Research on privacy is an active domain but there is still a wide gap between theory and everyday life's applications. Filling this gap will require academia to design protocols and algorithms that fit the real life constraints.

In this work, first we studied a number of identification and authentication protocols based on classical symmetric-key cryptographic building blocks (e.g. hash functions) and providing sub-linear online complexity to identify users. We have evaluated each of the schemes by examining whether they satisfy a set of security properties under a well-known adversarial model [64]. We have shown two new attacks on the CHT protocol [2] which is a very efficient protocol in terms of key search complexity (i.e., $O(\sqrt{N})$). We also introduced two new traceability attacks on the CTI protocol [16]. Furthermore, we have shown a traceability weakness of the mutual authentication version of OSK/AO [19] protocol, and shown a possible way to repair this problem with no additional cost. We also introduce traceability attacks on OSK/BF [22], O-RAP [90] and YA-TRAP* [18], which emphasize the importance of timing attacks [67] on the reader side. Finally, we have extensively evaluated and compared all the candidates according to their security, and performance. The security properties that we investigated include user privacy and as well as forward privacy, impersonation resiliency and desynchronization resistance. Furthermore, we examined thoroughly their performance, in terms of computational and storage cost.

Second, we have implemented OSK/AO [19] based on time-memory trade-off method which is the first implementation according to our best knowledge. Our implementation practically allows achieving a high performance by means of online search complexity and memory usage without degrading the user privacy. We have run several experiments on the implemented real RFID system. The experimental outputs are fascinating since they are very close to the theoretical bounds. Finally, the authentication speed and effective memory usage put forth that this forward-private RFID system is ready to be used for practical proposes.

REFERENCES

[1] **Liang, B.**, 2010, Security and Performance Analysis for RFID Protocols.

[2] **Cheon, J.H.**, **Hong, J. and Tsudik, G.**, 2009, Reducing RFID Reader Load with the Meet-in-the-Middle Strategy, Cryptology ePrint Archive, Report 2009/092.

[3] **Boneh, D.**, editor, 2003. Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, volume2729 of *Lecture Notes in Computer Science*, Springer, Santa Barbara, California, USA.

[4] **Finkenzeller, K.**, 2003. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, John Wiley & Sons, Inc., New York, NY, USA, 2 edition.

[5] **Juels, A.**, 2006. RFID security and privacy: a research survey, *IEEE Journal on Selected Areas in Communications*, **24(2)**, 381–394.

[6] **Reding, V.**, 2009, Commission Recommendation of 12.05.2009 - SEC(2009) 585/586, on the implementation of privacy and data protection principles in applications supported by radio-frequency identification.

[7] **Cavioukan, A.**, 2006, Privacy Guidelines for RFID Information Systems (RFID Privacy Guidelines).

[8] **Karygiannis, T.**, **Eydt, B.**, **Barber, G.**, **Bunn, L. and Phillips, T.**, 2007, Guidelines for Securing Radio Frequency Identification (RFID) Systems.

[9] **Simitian, J.**, 2005, Californian Senate Bill No.682.

[10] **Menezes, A.**, **van Oorschot, P.C. and Vanstone, S.A.** Handbook of Applied Cryptography, CRC Press, Boca Raton, FL. year = 1996,.

[11] **van Tilborg, H.C.A.**, editor, 2005. Encyclopedia of Cryptography and Security, Springer.

[12] **Kardaş, S.**, **Levi, A. and Murat, E.**, 2011. Providing Resistance against Server Information Leakage in RFID Systems, New Technologies, Mobility and Security – NTMS'11, IEEE, IEEE Computer Society, Paris, France, pp.1–7.

[13] **Peris-Lopez, P.**, 2008. Lightweight Cryptography in Radio Frequency Identification (RFID) Systems, Ph.D. thesis, Computer Science Department, Carlos III University of Madrid.

[14] **Hartmanis, J.**, 1988. Computational Complexity Theory, volume 38, American Mathematical Society, Atlanta, Georgia.

[15] **Avoine, G.**, 2012, RFID Security & Privacy Lounge, http://www.avoine.net/rfid/.

[16] **Alomair, B.**, **Clark, A.**, **Cuellar, J. and Poovendran, R.**, 2010. Scalable RFID Systems: a Privacy-Preserving Protocol with Constant-Time Identification, the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – DSN'10, IEEE, IEEE Computer Society, Chicago, IL, USA.

[17] **Tsudik, G.**, 2006. YA-TRAP: Yet Another Trivial RFID Authentication Protocol, International Conference on Pervasive Computing and Communications – PerCom 2006, IEEE, IEEE Computer Society, Pisa, Italy, pp.640–643.

[18] **Tsudik, G.**, 2007. A Family of Dunces: Trivial RFID Identification and Authentication Protocols, **N. Borisov and P. Golle**, editors, Workshop on Privacy Enhancing Technologies – PET 2007, volume4776 of *Lecture Notes in Computer Science*, Springer, Ottawa, Canada,, pp.45–61.

[19] **Avoine, G.**, **Dysli, E. and Oechslin, P.**, 2005. Reducing Time Complexity in RFID Systems, **B. Preneel and S. Tavares**, editors, Selected Areas in Cryptography – SAC 2005, volume3897 of *Lecture Notes in Computer Science*, Springer, Kingston, Canada, pp.291–306.

[20] **Burmester, M.**, **Le, T.v. and Medeiros, B.d.**, 2009. Universally Composable RFID Identification and Authentication Protocols, *ACM Transactions on Information and System Security – TISSEC'09*, **12(4)**, 21:1–21:33.

[21] **Van Le, T.**, **Burmester, M. and de Medeiros, B.**, 2007. Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange, **F. Bao and S. Miller**, editors, ACM Symposium on Information, Computer and Communications Security – ASIACCS 2007, ACM, ACM Press, Singapore, Republic of Singapore, pp.242–252.

[22] **Nohara, Y.**, **Inoue, S. and Yasuura, H.**, 2008. A secure high-speed identification scheme for RFID using Bloom filters, Third International Conference on Availability, Reliability and Security – AReS 2008, Barcelona, Spain, pp.727–722.

[23] **Hopper, N. and Blum, M.**, 2001. Secure Human Identification Protocols, *Advances in Cryptology – Asiacrypt 2007*, 52–66.

[24] **Juels, A. and Weis, S.**, 2005. Authenticating Pervasive Devices with Human Protocols, **V. Shoup**, editor, Advances in Cryptology – CRYPTO'05, volume3126 of *Lecture Notes in Computer Science*, IACR, Springer, Santa Barbara, California, USA, pp.293–308.

[25] **Bringer, J.**, **Chabanne, H. and Emmanuelle, D.**, 2006. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks, IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and

Trust in Pervasive and Ubiquitous Computing – SecPerU 2006, IEEE, IEEE Computer Society, Lyon, France.

[26] **Duc, D.N. and Kim, K.**, 2007. Securing HB+ against GRS Man-in-the-Middle Attack, Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security, pp.23–26.

[27] **Halevi, T.**, **Saxena, N. and Halevi, S.**, 2009. Using HB Family of Protocols for Privacy-Preserving Authentication of RFID Tags in a Population, Workshop on RFID Security – RFIDSec'09, Leuven, Belgium.

[28] **Munilla, J. and Peinado, A.**, 2007. HB-MP: A further step in the HB-family of lightweight authentication protocols, *Computer Networks*, **51(9)**, 2262–2267.

[29] **Billet, O.**, **Etrog, J. and Gilbert, H.**, 2010. Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher, **S. Hong and T. Iwata**, editors, Fast Software Encryption – FSE'10, volume6147 of *Lecture Notes in Computer Science*, Springer, Seoul, Korea, pp.55–74.

[30] **Canard, S. and Coisel, I.**, 2008. Data Synchronization in Privacy-Preserving RFID Authentication Schemes, Workshop on RFID Security – RFIDSec'08, Budapest, Hungary.

[31] **Bringer, J.**, **Chabanne, H. and Icart, T.**, 2008. Cryptanalysis of EC-RAC, a RFID identification protocol, **M.K. Franklin**, **L.C.K. Hui and D.S. Wong**, editors, 7th International Conference on Cryptology And Network Security – CANS'08, volume5339 of *Lecture Notes in Computer Science*, Springer, Hong Kong, China, pp.149–161.

[32] **Hein, D.**, **Wolkerstorfer, J. and Felber, N.**, 2008. ECC is Ready for RFID – A Proof in Silicon, Workshop on RFID Security – RFIDSec'08, Budapest, Hungary.

[33] **Hutter, M.**, **Feldhofer, M. and Plos, T.**, 2010. An ECDSA Processor for RFID Authentication, **S.O. Yalcin**, editor, Workshop on RFID Security – RFIDSec'10, volume6370 of *Lecture Notes in Computer Science*, Springer, Istanbul, Turkey, pp.189–202.

[34] **Lee, Y.K.**, **Sakiyama, K.**, **Batina, L. and Verbauwhede, I.**, 2008. Elliptic-Curve-Based Security Processor for RFID, *IEEE Transactions on Computers*, 1514–1527.

[35] **Juels, A.**, 2004. Minimalist Cryptography for Low-Cost RFID Tags, **C. Blundo and S. Cimato**, editors, International Conference on Security in Communication Networks – SCN 2004, volume3352 of *Lecture Notes in Computer Science*, Springer, Amalfi, Italy, pp.149–164.

[36] **Henrici, D. and Müller, P.**, 2004. Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers, **R. Sandhu and R. Thomas**, editors, International Workshop on Pervasive

Computing and Communication Security – PerSec 2004, IEEE, IEEE Computer Society, Orlando, Florida, USA, pp.149–153.

[37] **Avoine, G.**, 2005. Adversary Model for Radio Frequency Identification, Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland.

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

[39] **David, M. and Prasad, N.R.**, 2009. Providing Strong Security and High Privacy in Low-Cost RFID Networks, **O. Akan**, **P. Bellavista**, **J. Cao**, **F. Dressler**, **D. Ferrari**, **M. Gerla**, **H. Kobayashi**, **S. Palazzo**, **S. Sahni**, **X.S. Shen**, **M. Stan**, **J. Xiaohua**, **A. Zomaya**, **G. Coulson**, **A.U. Schmidt and S. Lian**, editors, Security and Privacy in Mobile Information and Communication Systems, volume 17 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer Berlin Heidelberg, Turin, Italy, pp.172–179.

[40] **Lee, Y.C.**, **Hsieh, Y.C.**, **You, P.S. and Chen, T.C.**, 2009. A New Ultralightweight RFID Protocol with Mutual Authentication, WASE International Conference on Information Engineering – ICIE '09, IEEE, IEEE Computer Society, Taiyuan, Shanxi, pp.58–61.

[41] **Peris-Lopez, P.**, **Hernandez-Castro, J.C.**, **Estevez-Tapiador, J.M. and Ribagorda, A.**, 2006. EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags, OTM Federated Conferences and Workshop: IS Workshop – IS'06, volume4277 of *Lecture Notes in Computer Science*, Springer, Montpellier, France, pp.352–361.

[42] **Peris-Lopez, P.**, **Hernandez-Castro, J.C.**, **Estevez-Tapiador, J.M. and Ribagorda, A.**, 2006. LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags, Workshop on RFID Security – RFIDSec'06, Ecrypt, Graz, Austria.

[43] **Peris-Lopez, P.**, **Hernandez-Castro, J.C.**, **Estevez-Tapiador, J.M. and Ribagorda, A.**, 2006. M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags, **J. Ma**, **H. Jin**, **L.T. Yang and J.J.P. Tsai**, editors, International Conference on Ubiquitous Intelligence and Computing – UIC'06, volume4159 of *Lecture Notes in Computer Science*, Springer, Wuhan and Three Gorges, China, pp.912–923.

[44] **Peris-Lopez, P.**, **Hernandez-Castro, J.C.**, **Estevez-Tapiador, J.M. and Ribagorda, A.**, 2008. Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol, **K.I. Chung**, **K. Sohn and M. Yung**, editors, Workshop on Information Security Applications – WISA'08, volume5379 of *Lecture Notes in Computer Science*, Springer, Jeju Island, Korea, pp.56–68.

[45] **Yeh, K.H.**, **Lo, N. and Winata, E.**, 2010. An Efficient Ultralightweight Authentication Protocol for RFID Systems, Workshop on RFID Security – RFIDSec Asia'10, volume 4 of *Cryptology and Information Security*, IOS Press, Singapore, Republic of Singapore.

[46] **Chen, C.L. and Deng, Y.Y.**, 2009. Conformation of EPC Class 1 Generation 2 standards RFID system with mutual authentication and privacy protection, *Engineering Applications of Artificial Intelligence*, **22**, 1284–1291.

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

[48] **Nguyen Duc, D.**, **Park, J.**, **Lee, H. and Kim, K.**, 2006. Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning, Symposium on Cryptography and Information Security, Hiroshima, Japan.

[49] **Yeh, T.C.**, **Wang, Y.J.**, **Kuo, T.C. and Wang, S.S.**, 2010. Securing RFID systems conforming to EPC Class 1 Generation 2 standard, *Expert System Applications*, **37(12)**, 7678–7683.

[50] **Han, D. and Kwon, D.**, 2009. Vulnerability of an RFID authentication protocol conforming to EPC Class 1 Generation 2 Standards, *Comput. Stand. Interfaces*, **31**, 648–652.

[51] **Peris-Lopez, P.**, **Hernandez-Castro, J.C.**, **Tapiador, J.E. and van der Lubbe, J.C.A.**, 2011. Cryptanalysis of an EPC Class-1 Generation-2 standard compliant authentication protocol, *Engineering Applications of Artificial Intelligence*, **24(6)**, 1061–1069.

[52] **Peris-Lopez, P.**, **Hernandez-Castro, J.C.**, **Estevez-Tapiador, J.M.**, **Li, T. and van der Lubbe, J.C.**, 2009. Weaknesses in Two Recent Lightweight RFID Authentication Protocols, Workshop on RFID Security – RFIDSec'09, Leuven, Belgium.

[53] **Peris-Lopez, P.**, **Hernandez-Castro, J.C.**, **Estevez-Tapiador, J.M. and Ribagorda, A.**, 2007. Cryptanalysis of a Novel Authentication Protocol Conforming to EPC-C1G2 Standard, Workshop on RFID Security – RFIDSec'07, Malaga, Spain.

[54] **Brands, S. and Chaum, D.**, 1993. Distance-Bounding Protocols, Advances in Cryptology – EUROCRYPT'93, volume765 of *Lecture Notes in Computer Science*, Springer-Verlag, Lofthus, Norway, pp.344–359.

[55] **Hancke, G. and Kuhn, M.**, 2005. An RFID Distance Bounding Protocol, Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005, IEEE, Athens, Greece.

[56] **Kara, O.**, **Kardaş, S.**, **Bingöl, M.A. and Avoine, G.**, 2010. Optimal Security Limits of RFID Distance Bounding Protocols, **S.O. Yalcin**, editor, Workshop on RFID Security – RFIDSec'10, volume6370 of *Lecture Notes in Computer Science*, Springer, Istanbul, Turkey, pp.220–238.

[57] **Kardaş, S.**, **Kiraz, M.S.**, **Bingöl, M.A. and Demirci, H.**, 2012. A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions, **A. Juels and C. Paar**, editors, 7th International Workshop, on RFID Security –RFIDSec'11, volume7055 of *Lecture Notes in Computer Science*, Springer, Amherst, Massachusetts, USA.

[58] **Kim, C.H. and Avoine, G.**, 2009. RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks, **J.A. Garay**, **A. Miyaji and A. Otsuka**, editors, 8th International Conference on Cryptology And Network Security – CANS'09, volume5888 of *Lecture Notes in Computer Science*, Springer-Verlag, Kanazawa, Ishikawa, Japan, pp.119–133.

[59] **Munilla, J. and Peinado, A.**, 2008. Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels, *Wireless Communications and Mobile Computing*, **8(9)**, 1227–1232.

[60] **Trujillo Rasua, R.**, **Martin, B. and Avoine, G.**, 2010. The Poulidor Distance-Bounding Protocol, **S.O. Yalcin**, editor, Workshop on RFID Security – RFIDSec'10, volume6370 of *Lecture Notes in Computer Science*, Springer, Istanbul, Turkey, pp.239–257.

[61] **Avoine, G.**, **Bingöl, M.A.**, **Kardaş, S.**, **Lauradoux, C. and Martin, B.**, 2011. A Framework for Analyzing RFID Distance Bounding Protocols, *Journal of Computer Security – Special Issue on RFID System Security*, **19(2)**, 289–317.

[62] **Burmester, M.**, **Le, T.v. and de Medeiros, B.**, 2006. Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols, Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2006, IEEE, IEEE Computer Society, Baltimore, MD, USA, pp.1–10.

[63] **Canard, S.**, **Coisel, I. and Girault, M.**, 2010. Security of Privacy-Preserving RFID Systems, IEEE International Conference on RFID-Technology and Applications – RFID-TA'10, Sun Yat-sen University, IEEE, Guangzhou, China, pp.269–274.

[64] **Juels, A. and Weis, S.**, 2007. Defining Strong Privacy for RFID, International Conference on Pervasive Computing and Communications – PerCom 2007, IEEE, IEEE Computer Society, New York City, NY, USA, pp.342–347.

[65] **Vaudenay, S.**, 2007. On Privacy Models for RFID, **K. Kurosawa**, editor, Advances in Cryptology – Asiacrypt 2007, volume4833 of *Lecture Notes in Computer Science*, Springer, Kuching, Malaysia, pp.68–87.

[66] **Ohkubo, M.**, **Suzuki, K. and Kinoshita, S.**, 2003. Cryptographic Approach to "Privacy-Friendly" Tags, RFID Privacy Workshop, MIT, MA, USA.

[67] **Avoine, G.**, **Coisel, I. and Martin, T.**, 2010. Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols, **S.O. Yalcin**, editor, Workshop on RFID Security – RFIDSec'10, volume6370 of *Lecture Notes in Computer Science*, Springer, Istanbul, Turkey, pp.138–157.

[68] **Molnar, D. and Wagner, D.**, 2004. Privacy and Security in Library RFID: Issues, Practices, and Architectures, **V. Atluri**, **B. Pfitzmann and P.D. McDaniel**, editors, Conference on Computer and Communications Security – ACM CCS'04, ACM, ACM Press, Washington, DC, USA, pp.210–219.

[69] **Avoine, G.**, **Buttyán, L.**, **Holczer, T. and Vajda, I.**, 2007. Group-Based Private Authentication, IEEE International Workshop on Trust, Security, and Privacy for Ubiquitous Computing – TSPUC, IEEE, IEEE Computer Society, Helsinki, Finland, pp.1–6.

[70] **Buttyán, L.**, **Holczer, T. and Vajda, I.**, 2006. Optimal Key-Trees for Tree-Based Private Authentication, **G. Danezis and P. Golle**, editors, Workshop on Privacy Enhancing Technologies – PET 2006, volume4258 of *Lecture Notes in Computer Science*, Springer, Cambridge, United Kingdom, pp.332–350.

[71] **Nohl, K. and Evans, D.**, 2006. Quantifying Information Leakage in Tree-Based Hash Protocols, **P. Ning**, **S. Qing and N. Li**, editors, International Conference on Information and Communications Security – ICICS'06, volume4307 of *Lecture Notes in Computer Science*, Springer, Raleigh, NC, USA, pp.228–237.

[72] **Avoine, G.**, **Martin, B. and Martin, T.**, 2010. Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly, **S.O. Yalcin**, editor, Workshop on RFID Security – RFIDSec'10, volume6370 of *Lecture Notes in Computer Science*, Springer, Istanbul, Turkey, pp.103–122.

[73] **Beye, M. and Veugen, T.**, 2011, Improved Anonymity for Key-Trees, Cryptology ePrint Archive, Report 2011/395.

[74] **Yao, Q.**, **Qi, Y.**, **Han, J.**, **Zhao, J.**, **Li, X. and Liu, Y.**, 2009. Randomizing RFID private authentication, Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on, IEEE, pp.1–10.

[75] **Akgün, M.**, **Caglayan, M.U. and Anarim, E.**, 2009. Secure RFID Authentication with Efficient Key-lookup, Proceedings of the 28th IEEE conference on Global telecommunications, GLOBECOM'09, IEEE Press, Piscataway, NJ, USA, pp.4777–4784.

[76] **Dimitriou, T.**, 2005. A Lightweight RFID Protocol to protect against Traceability and Cloning attacks, Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005, IEEE, IEEE Computer Society, Athens, Greece, pp.59–66.

[77] **Lu, L.**, **Han, J.**, **Hu, L.**, **Liu, Y. and Ni, L.**, 2007. Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems, International Conference on Pervasive Computing and Communications – PerCom 2007, IEEE, IEEE Computer Society, New York City, NY, USA, pp.13–22.

[78] **Wang, W.**, **Li, Y.**, **Hu, L. and Lu, L.**, 2007. Storage-awareness: RFID private authentication based on sparse tree, Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2007. SECPerU 2007. Third International Workshop on, IEEE, pp.61–66.

[79] **Diffie, W. and Hellman, M.**, 1977. Special Feature Exhaustive Cryptanalysis of the NBS Data Encryption Standard, *Computer*, **10(6)**, 74–84.

[80] **Avoine, G. and Oechslin, P.**, 2005. A Scalable and Provably Secure Hash Based RFID Protocol, International Workshop on Pervasive Computing and Communication Security – PerSec 2005, IEEE, IEEE Computer Society, Kauai Island, HI, USA, pp.110–114.

[81] **Berbain, C.**, **Billet, O.**, **Etrog, J. and Gilbert, H.**, 2009. An Efficient Forward Private RFID Protocol, **E. Al-Shaer**, **S. Jha and A.D. Keromytis**, editors, Conference on Computer and Communications Security – ACM CCS'09, ACM, ACM Press, Chicago, IL, USA, pp.43–53.

[82] **Ohkubo, M.**, **Suzuki, K. and Kinoshita, S.**, 2004. Efficient Hash-Chain Based RFID Privacy Protection Scheme, International Conference on Ubiquitous Computing – Ubicomp, Workshop Privacy: Current Status and Future Directions, Nottingham, England.

[83] **Hellman, M.**, 1980. A cryptanalytic time-memory trade-off, *Information Theory, IEEE Transactions on*, **26(4)**, 401–406.

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

[85] **Bloom, B.H.**, 1970. Space/time trade-offs in hash coding with allowable errors, *Communications of the ACM*, **13(7)**, 422–426.

[86] **Nohara, Y. and Inoue, S.**, 2010. A Secure and Scalable Identification for Hash-based RFID Systems Using Updatable Pre-computation, **S. Wetzel**, **C. Nita-Rotaru and F. Stajano**, editors, Proceedings of the 3rd ACM Conference on Wireless Network Security – WiSec'10, ACM, ACM Press, Hoboken, New Jersey, USA, pp.65–74.

[87] **Broder, A. and Mitzenmacher, M.**, 2001. Using multiple hash functions to improve IP lookups, Procedings of the twentieth Annual Joint Conference of the IEEE Computer and Communications Societies – INFOCOM 2001, volume 3, IEEE, pp.1454–1463.

[88] **Ouafi, K. and Phan, R.C.W.**, 2008. Privacy of Recent RFID Authentication Protocols, **L. Chen**, **Y. Mu and W. Susilo**, editors, 4th International Conference on Information Security Practice and Experience – ISPEC

2008, volume4991 of *Lecture Notes in Computer Science*, Springer, Sydney, Australia, pp.263–277.

[89] **Lamport, L.**, 1981. Password authentication with insecure communication, *Communications of the ACM*, **24(11)**, 770–772.

[90] **Chatmon, C.**, **van Le, T. and Burmester, M.**, 2006. Secure Anonymous RFID Authentication Protocols, Technical Report TR-060112, Florida State University, Department of Computer Science, Tallahassee, Florida, USA.

[91] **Avoine, G.**, **Junod, P. and Oechslin, P.**, 2008. Characterization and Improvement of Time-Memory Trade-Off Based on Perfect Tables, *ACM Trans. Inf. Syst. Secur.*, **11**, 17:1–17:22, a preliminary version has appeared in *Progress in Cryptology – Indocrypt 2005* volume 3797, Springer.

[92] **Borst, J.**, **Preneel, B. and Vandewalle, J.**, 1998. On the Time-Memory Tradeoff Between Exhaustive Key Search and Table Precomputation, Proceeding of the 19th Symposium in Information Theory in the Benelux, WIC, Veldhoven, The Netherlands, pp.111–118.

[93] **Avoine, G.**, **Junod, P. and Oechslin, P.**, 2005. Time-Memory Trade-Offs: False Alarm Detection Using Checkpoints, Progress in Cryptology – Indocrypt 2005, volume3797 of *Lecture Notes in Computer Science*, Cryptology Research Society of India, Springer-Verlag, Bangalore, India, pp.183–196.

[94] **Avoine, G.**, **Junod, P. and Oechslin, P.**, 2005. Time-Memory Trade-Offs: False Alarm Detection Using Checkpoints, Technical Report LASEC-REPORT-2005-002, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland.

CURRICULUM VITAE

Name Surname: Muhammed Ali BİNGÖL

Place and Date of Birth: Erzurum / 03.03.1986

E-Mail: mabingol@gmail.com, muhammedalib@uekae.tubitak.gov.tr

B.Sc.: Istanbul Technical University, Telecommunications Engineering, June 2008.

List of Publications

- Gildas Avoine, **Muhammed Ali Bingöl**, Süleyman Kardaş, Cedric Lauradoux, and Benjamin Martin., 2010:A Framework for Analyzing RFID Distance Bounding Protocols *Journal of Computer Security – Special Issue on RFID System Security*, volume 19, number 2/2011, pages 289-317, IOS Press, 2011. Available at http://sites.uclouvain.be/security/download/papers/AvoineBKLM-2010-jcs.pdf.

- Gildas Avoine, **Muhammed Ali Bingöl**, Xavier Carpent, S. Berna Örs Yalçın, 2011: Privacy-friendly Authentication in RFID Systems: On Sub-linear Protocols based on Symmetric-key Cryptography(submitted to IEEE Transactions on Mobile Computing (TMC)).

- Orhun Kara, Süleyman Kardaş, **Muhammed Ali Bingöl**, Gildas Avoine, 2010: Optimal Security Limits of RFID Distance Bounding Protocols, *In 6th International Workshop on RFID Security - RFIDSec'10*, Istanbul, Turkey, June 2010. Lecture Notes in Computer Science vol. 6370, pages 220-238, Springer Verlag. Available at http://sites.uclouvain.be/security/download/papers/KaraKBA-2010-rfidsec.pdf

- **Muhammed Ali Bingöl**, Ali Özhan Gürel, Orhun Kara, Süleyman Kardaş, 2010: A New RFID Distance Bounding Protocol, *4th International Information Security*

& Cryptology Conference (ISCTurkey'10), Ankara, Turkey, May 2010. Available at http://www.projectice.eu/en/images/papers/01.pdf

- Süleyman Kardaş, **Muhammed Ali Bingöl**, 2010: Attacks on a Mutual Authentication Scheme Conforming to EPCglobal C1 GEN 2 RFID System,*4th International Information Security & Cryptology Conference - ISCTurkey'10*, Ankara, Turkey, May 2010. Available at http://www.projectice.eu/en/images/papers/02.pdf

- Süleyman Kardaş, Mehmet Sabır Kiraz, **Muhammed Ali Bingöl**, Hüseyin Demirci, 2010: A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions, *In 7th International Workshop on RFID Security - RFIDSec'11*, Amherst, USA, June 2011. Lecture Notes in Computer Science vol. 7055, pages. 78-93, Springer Verlag.

- Mehmet Sabır Kiraz, Süleyman Kardaş, **Muhammed Ali Bingöl**, Fatih Birinci, 2011: An Improved Internet Voting Protocol (submitted to a journal) *Available at IACR Cryptology ePrint Archive, Report 2011/341* http://eprint.iacr.org/2011/341.pdf.