Cryptographic Approaches To Security and Privacy Issues In Pervasive Computing

Jihoon Cho

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

Information Security Group
Department of Mathematics
Royal Holloway, University of London

2013

Declaration

These doctoral studies were conducted under the supervision of Professor Chris J. Mitchell.

The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, whilst enrolled in the Information Security Group of Royal Holloway, University of London as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Jihoon Cho
April 2013

Acknowledgements

During my PhD programme, I had the most unforgettable experiences in my life. I've gone through my father's funeral and got married to a woman. I also look forward to meeting one amazing person this spring.

My father was my very best friend and great mentor. He was such a generous and lovely person, and I still remember that all his students and fellow teachers sent him away with tears at his retirement. When I left to Waterloo in Canada for my master, he was not happy with my plan. He was a bit like Helfgott's father in the film 'Shine', who wants his son to stay with him and family. However, he became supportive during my PhD programme, and used to encourage me with his great wisdom and love whenever I faced walls standing against me. He passed away at the second year of PhD. I had to spend a while until recovering from the shock. I lost him, but everything of him lives with me.

With my father's support I could start my PhD, but it was not possible to complete my thesis without another person. I met a girl when I joined an industrial project in Korea during the first year of PhD, and she became my wife in two years. She has been my a source of energy, always praying for me and my study and encouraging me with her lovely smile.

In this acknowledgement, I would like to give thanks to some special people. I greatly appreciate my supervisor Chris who has guided me with his experience and advice. I am thankful to my 'Good-Old-Days' roommates in MC256, Shriram and James. We used to have great discussions on everything, of course, including cryptography. After joining LG Electronics in 2009, my colleagues shared their experiences in the area of standardisation. Most importantly, Mom, I love you very much. I am what I am because of you!

I would like to conclude my acknowledgement with all my thanks to God, who is my Saviour, my Lord, and my King forever!

Seoul, Korea
March, 2011

Abstract

Technological innovation has enabled tiny devices to participate in pervasive computing. Such devices are particularly vulnerable to security and privacy threats, because of their limited computing resources and relatively weak physical security. We investigate possible cryptographic solutions to security and privacy problems arising in two kinds of emerging pervasive computing networks: Personal Area Networks (PANs) and the EPCglobal Network.

A number of key management schemes have been proposed for use in PANs, but these schemes only support key management within a PAN. However, as people are increasingly equipped with multiple wireless devices, PANs are likely to be interconnected to share information or services. We introduce a term, iPANs, to name such interconnected PANs. We define system models and design goals for key management in iPANs, and propose a novel security initialisation scheme for use in iPANs. The proposed scheme achieves desirable security and efficiency properties by making use of the unique characteristics of PANs.

The EPCglobal Network is designed to give efficiency and cost savings in and beyond the supply chain using Radio Frequency Identification (RFID) technology; however, privacy threats affecting such networks are particularly serious. We construct a formal privacy model for RFID systems accurately reflecting adversarial threats and power. We then give brief privacy analysis for the existing privacy-enhanced RFID schemes which have received wide attention in the literature. We then construct a secure refresh-based RFID system based on re-encryption techniques, and prove its privacy using the defined privacy model. Finally, we show that the proposed scheme can greatly enhance the security and privacy of EPC tags, making the maximum use of given tag functionalities as specified in the standards.

Contents

List of Figures

List of Tables

Abbreviations

AES	Advanced Encryption Standard
AODV	Ad hoc On-demand Distance Vector
CA	Certificate Authority
CCA	Chosen Ciphertext Attack
CDH	Computational Diffie-Hellman
CPA	Chosen Plaintext Attack
CRC	Cyclic Redundancy Checking
CRL	Certificate Revocation List
DDH	Decisional Diffie-Hellman
DKG	Distributed Key Generator
DoS	Denial of Service
DL	Discrete Logarithm
DPKG	Distributed Private Key Generator
DSR	Dynamic Source Routing
ECDL	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EPC	Electronic Product Code
EPCIS	EPC Information Service
Gen2	The Class-1 Generation-2 UHF RFID standard
GEs	Gate Equivalences
GGM	Goldreich-Goldwasser-Micali
GPRS	General Packet Radio Services
HF	High Frequency
IBC	Identity-based Cryptography
IBE	Identity-based Encryption
IBS	Identity-based Signature
IC	Integrated Circuit
ICAO	International Civil Aviation Organisation
IE-CCA	Indistinguishable Encryption under Chosen Ciphertext Attack
IE-CPA	Indistinguishable Encryption under Chosen Plaintext Attack
IEC	International Electrotechnical Commission
IK-CPA	Indistinguishability of Keys under Chosen Plaintext Attack
iPANs	interconnected Personal Area Networks
ISM	Industrial-Scientific-Medical
ISO	International Organisation for Standardisation
KDC	Key Distribution Centre
LAN	Local Area Network
LF	Low Frequency

LPN	Learning Parity in the Presence of Noise
LSB	Least Significant Bit
MAC	Message Authentication Code
MAC (address)	Media Access Control
MOV (Reduction)	Menezes-Okamoto-Vanstone
MRZ	Machine Readable Zone
MSB	Most Significant Bit
NFC	Near Field Communication
ONS	Object Name Services
OOB	Out-of-Band
PAN	Personal Area Network
PDA	Personal Digital Assistant
POWF	Physical One-Way Function
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
PKG	Private Key Generator
PPT	Probabilistic Polynomial-time
PRNG	Pseudo Random Number Generator
RE	Re-randomisable Encryption
RF	Radio Frequency
RFID	Radio Frequency Identification
SSP	Secure Simple Pairing
TA	Trusted Authority
TLS	Transport Layer Security
TPM	Trusted Platform Module
TTP	Trusted Third Party
UC	Universal Composability
UCC	Uniform Code Council
UHF	Ultra High Frequency
UMTS	Universal Mobile Telecommunications System
UPC	Universal Product Code
UR	Universal Re-encryption
URL	Universal Resource Locator
USB	Universal Serial Bus
USS	Universal Semantic Security
USSR	Universal Semantic Security under Re-encryption
VSS	Verifiable Secret Sharing
WAP	Wireless Application Protocol
WLAN	Wireless Local Area Network
WTLS	Wireless Transport Layer Security

Notation

$a \oplus b$	bitwise XOR of bit strings $a, b \in \{0,1\}^l$ for some $l \in \mathbb{Z}$
$a \mid b$	b is divisible by a
$a \nmid b$	b is not divisible by a
a^b	a to the power b
$a \| b$	concatenation of strings a and b
$a \in_{\mathsf{R}} A$	uniformly random selection of a from a finite set A
$a := b$	defining a value a as a value b
$a \approx b$	a is approximately equal to b
$a \bmod p$	a remainder of a modulo p
$a \equiv b \pmod{p}$	a is congruent to b modulo p
$\|a\|$	the length of a bit string $a \in \{0,1\}^*$
$\|a\|$	the binary length of $a \in \mathbb{N}$
$\|\|a\|\|$	the Hamming Weight of vector a
$\|\|A\|\|$	the Hamming Weight of matrix A
$\lfloor a \rfloor$	the greatest integer less than or equal to a
$\lceil a \rceil$	the smallest integer greater than or equal to a
$\|A\|$	the number of elements in a set A
$\perp$	an error message
1^n	a constant bit string $11\ldots1$ of length n
m	a plaintext message
ak	a tag-access key
pk	a public key
sk	a secret key
ϵ	an arbitrary negligible function
$\langle g \rangle$	a set of all powers of g
$\mathbb{N}$	a set of positive integers
$\mathbb{Z}$	a set of integers
$\mathbb{Z}_q$	a group $\{0, 1, \ldots, q-1\}$ under addition modulo q
$\mathbb{Z}_q^*$	the non-zero elements of $\mathbb{Z}_q$
$\mathbb{F}_p$	a finite field with p elements
$\mathbb{F}_p^*$	the non-zero elements of $\mathbb{F}_p$
$\mathcal{D}_{\mathsf{id}}$	a PAN device with an identifier id
$(Q_{\mathsf{id}}, \mathsf{d}_{\mathsf{id}})$	a public/private key pair for a device $\mathcal{D}_{\mathsf{id}}$
$\mathcal{P}_i$	a party engaging a secret sharing scheme
	a PKG from each PAN in the km-iPANs scheme
$\mathcal{I}$	a set of non-revoked PKGs in revocation scheme
$\mathcal{Q}$	a set of non-disqualified PKGs in secret sharing scheme
$\mathcal{G}$	see definition 2.3

$\mathcal{R}$	a tag
$\mathcal{T}$	a reader
$\hat{e}$	an admissible bilinear pairing
$E(\mathbb{F}_p)$	an elliptic curve group for an elliptic curve E defined over $\mathbb{F}_p$
$\{0,1\}^*$	a set of bit strings of arbitrary length
$\{0,1\}^l$	a set of bit strings of a length l
$\{0,1\}^{<d}$	a set of bit strings of a length that is less than d
$\{0,1\}^{\leq d}$	a set of bit strings of a length that is less than or equal to d
$\{0,1\}^{>d}$	a set of bit strings of a length that is greater than d
$\{0,1\}^{\geq d}$	a set of bit strings of a length that is greater than or equal to d
$O(\cdot)$	a big O notation which limits behaviour of a function when the argument tends towards a particular value or infinity

Chapter 1

Introduction

Contents

This chapter provides the motivation, contributions, and structure of the thesis.

1.1 Motivation

The term pervasive computing was first introduced by Schechter [125], as a way of describing the anticipated environment of computing services available anytime, anywhere, and on demand. In pervasive computing, devices are integrated into everyday objects and activities, and seamlessly communicate to share and exchange huge amounts of information. Technological innovation has enabled tiny devices to participate in pervasive computing; however, such devices are particularly vulnerable to security and privacy threats.

1.1.1 Security and privacy issues in pervasive computing

The term security includes the notions of confidentiality, integrity, availability, authenticity, etc. A large number of security mechanisms supporting these security goals have been developed, but these solutions are not all applicable to pervasive computing systems. Security mechanisms for a pervasive environment should be capable of handling (i) the diversity of computing resources available to devices, and (ii) the dynamics including the mobility, ubiquity, and decentralised nature of pervasive computing systems.

Pervasive computing technology is often described as a means of enabling constant surveillance of large parts of the population, because actions reflected in networked computing devices may allow personal profiling in great detail and to a high level of accuracy. Such public concerns are growing, mainly because of the combination of (i) the *ubiquity* of tiny computing devices, e.g. RFID tags, and (ii) their *invisibility*, i.e. the often uncontrolled wireless communications performed by such devices.

1.1.2 Topics in pervasive computing

We investigate possible cryptographic solutions to security and privacy issues arising in two pervasive computing systems: Personal Area Networks (PANs) and the EPCglobal Network.

Personal Area Networks (PANs)

A Personal Area Network (PAN) is a small wireless network that covers a personal work space, e.g. an office or a meeting room. A PAN only includes those components owned and controlled by a single user, and the components directly communicating with each other via a local interface such as Bluetooth or IrDA (Infrared Data Association). As the population is increasingly equipped with multiple wireless devices, PANs seem likely to become core elements of pervasive computing. Of course, PANs may act as stand-alone networks, but it seems likely that they will be interconnected to share information or services. We call such interconnected PANs iPANs.

Providing a robust and secure key management scheme for use in PANs remains a challenging task, in particular because of the unique characteristics of, and constraints on, such networks. That is, PAN devices are particularly susceptible to security and privacy threats because: (i) their computing resources are potentially limited, and thus often not possible to implement adequate cryptographic primitives; (ii) they are likely to be exposed to a wide range of physical attacks; and (iii) online trusted third parties (TTPs) are not always available. A variety of schemes have been proposed for securing PANs, but these schemes only support key management within a PAN.

The EPCglobal Network

Radio Frequency Identification (RFID) is a technology for automated identification of objects or people using radio communications. The EPCglobal Network is a standards-based technology designed to help realise automated global supply chain management using RFID technology.

The EPCglobal Network, like every technological innovation, is subject to potential information security risks. Security issues associated with the components of the EPCglobal Network other than the RFID system are similar to the concerns arising in other Internet applications. RFID technology, however, poses unique privacy and security concerns. In particular, a tag owner cannot physically control the communications of a tag, since most basic tags respond with their resident data to any reader queries without first authenticating the readers. Furthermore, tags do not store any communication history. RFID systems also suffer from threats similar to those that apply to iPANs. That is, the potentially limited computing capabilities of RFID tags cause serious security vulnerabilities, since standard cryptographic primitives are often beyond the capabilities of RFID tags. Also, since such tags will operate in hostile environments, they may be subject to a range of physical attacks, including fault induction or power analysis attacks.

Most previously proposed cryptographic solutions to the security and privacy issues of RFID technology are based on hardware-efficient hash functions or block ciphers. Such solutions, however, are not applicable to EPC tags[1], which cannot support most cryptographic primitives due to the limitations on their computing powers.

1.2 Contributions and Structure

In **Chapter 2**, we give the cryptographic preliminary necessary for the subsequent chapters of the thesis. We first describe the basic principles of modern cryptography. After discussing the computational approach to cryptography, we briefly present secret key and public key cryptography. We then discuss issues related to cryptographic key management.

[1] An EPC (Electronic Product Code) tag, the key component in the EPCglobal Network, is an RFID tag that is attached to, or embedded in, items.

The remainder of this thesis is divided into two distinct parts. In **Part I**, we present a study of key management schemes for PANs. This part of the thesis consists of chapters 3 and 4.

In **chapter 3**, We review previous research security mechanisms designed for use in PANs, focusing primarily on PAN security initialisation. After defining the notion of a PAN and giving a PAN security architecture, we present two existing approaches to PAN security initialisation.

In **chapter 4**, we propose a novel key management scheme for use both within and between PANs. We define a term, iPANs, to refer to interconnected PANs. We define a system model and give the design goals for key management in iPANs, and propose a security initialisation scheme for iPANs. We then show that the proposed scheme achieves desirable security and efficiency properties making use of the unique characteristics of PANs. Some of the research findings in this chapter have previously been published in [27, 28].

In **Part II**, we present a comprehensive study of security and privacy issues in RFID technology, and propose a solution to security and privacy problems in the EPCglobal Network. This part of the thesis consists of chapters 5–6.

In **chapter 5**, We discuss privacy issues of RFID technology, and construct a formal privacy model for RFID systems accurately reflecting adversarial threats and power. We then give brief privacy analysis for the existing privacy-enhanced RFID schemes which have received wide attention in the literature.

In **chapter 6**, we discuss security and privacy issues in the EPCglobal Network. After summarising the EPCglobal Network technology, we investigate the security and privacy issues arising in the RFID system of the EPCglobal Network, i.e. we focus on those issues exclusive to RFID technology. We also analyse the EPCglobal's current approach to such security and privacy concerns. We then propose a refresh-based RFID system, and analyse its privacy properties. Finally we discuss the application of the proposed RFID system and existing schemes to the EPCglobal Network. Some of the research findings in this chapter have previously been published in [29, 30].

In **chapter 7**, we conclude this thesis, and outline directions for further research.

Chapter 2

Cryptographic Preliminaries

Contents

This chapter provides cryptographic preliminaries necessary for the subsequent chapters of the thesis. We first describe the basic principles of modern cryptography. After discussing the computational approach to cryptography, we briefly present secret key and public key cryptography. We then discuss cryptographic key management issues.

2.1 Introduction

In this chapter, we give a brief overview of the cryptographic primitives relevant to this thesis. The material in this chapter is mostly derived from three books [35, 82, 102]. The rest of this chapter is organised as follows. In section 2.2, we describe the goals and basic principles of modern cryptography. In section 2.3, we briefly introduce computational security, covering cryptographic hardness assumptions and computational problems. We then describe secret key and public key primitives in sections 2.4 and 2.5. In section 2.6, we cover key management issues.

2.2 Modern Cryptography

In this section we give five main cryptographic goals, and describe basic principles of modern cryptography.

2.2.1 Cryptographic goals

The five main goals of cryptography can be defined as follows (ISO 7498-2).

- Authentication: can be subdivided into entity authentication, the corroboration that the entity at the other end of a communication link is the one claimed, and data origin authentication, the corroboration that the source of received data is as claimed.
- Access control: prevents unauthorised use of a resource.
- Data confidentiality: prevents disclosure of data to an unauthorised entity.
- Data integrity: prevents alteration or destruction of data by an unauthorised entity.
- Non-repudiation: prevents denial by an entity that it has taken a particular action, such as sending or receiving a message.

Elsewhere in the literature, e.g. 'Handbook of Applied Cryptography' [102], the term identification is used with the same meaning as entity authentication. Most cryptographic protocols require the identities of other entities in the protocol to be assured before starting cryptographic data processing. By contrast, in some RFID systems, e.g. the EPCglobal Network, a verifier (i.e. a reader) mostly obtains the identity declared by a claimant (i.e. a tag) without any corroboration, i.e. without entity authentication. We thus use the term identification with the following definition in this thesis: an identification is a process whereby one party (the verifier) obtains the identity that another party (the claimant) declares.

2.2.2 Basic principles

Classical cryptographic schemes were designed in an ad hoc manner and then evaluated on their perceived resistance to known attacks. This ad hoc approach involves providing informal arguments that any conceivable attack requires a resource level (e.g. time and space) greater than the fixed resources of a perceived adversary. Having survived such analysis, cryptographic primitives or protocols are said to possess heuristic security. Such claims of security, however, remain open to revision, since unforeseen attacks always remain a threat.

Modern cryptography to some extent rests on firmer and more scientific foundations by taking a rigorous approach. The following three principles distinguish modern cryptography from classical cryptography [82].

"*Principle 1 – The first step in solving any cryptographic problem is the formulation of a rigorous and precise definition of security.*"

In order to fully define the security of a cryptographic task, the attack model must be specified, i.e.

- what is considered to *break* the scheme, and
- what is assumed regarding the *power of adversary*: (i) the actions the adversary is assumed to be able to take; and (ii) the adversary's computational power.

Any definition of security will thus take the following form: *a cryptographic scheme for a given task is secure if no adversary of a specified power can break the scheme according to the given definition of security.*

"*Principle 2 – When the security of a cryptographic construction relies on an unproven assumption, this assumption must be precisely stated.*"

Most modern cryptographic constructions cannot be proven secure unconditionally. Constructing security proofs for today's schemes which do not depend on assumptions about the inherent difficulty of certain problems would require the resolution of fundamental questions in the theory of computational complexity, which seem far from being solved today. Those assumptions must be precisely stated, and they must be carefully studied. Thus modern cryptography rests on the heuristic assumption that the more an assumption is examined without it being successfully refuted, the greater confidence we can have that the assumption is true.

"*Principle 3 – Cryptographic constructions should be accompanied by a rigorous proof of security with respect to a definition formulated according to principle 1, and relative to an assumption of the form stated as in principle 2.*"

Giving an exact definition and a precise assumption is not in itself sufficient; without a *proof* that no adversary of the specified power can break the scheme, we have only our intuition that this is the case. Most proofs in modern cryptography use the reductionist approach: that is, given a theorem of the form, "*Given that assumption X is true, construction Y is secure according to the given definition,*" a proof typically shows that the problem of breaking construction Y is reduced to the problem of solving mathematical assumption X.

Remark. For some cryptographic solutions, e.g. complex schemes for key management, it is difficult to make a precise definition of security, i.e. it is difficult to give a mathematical statement capturing precisely what constitutes an attack, due to the complexity of the system. In such cases, we are obliged to use an ad hoc approach, achieving only heuristic security.

2.3 Computational Approach

In this section we briefly describe computational security, introducing cryptographic hardness assumptions and computational problems in a variety of mathematical settings.

2.3.1 Computational security

Cryptographic schemes are defined to be information-theoretically secure or perfectly secure when they can be proven mathematically secure (with respect to some particular definition of security) even against a computationally unlimited adversary. Most modern cryptographic constructions, however, aim to achieve computational security. Computational security is weaker than information-theoretic security, incorporating the following two relaxations:

- security is only preserved against *efficient* adversaries (algorithms) that run in a feasible amount of time; and
- adversaries can potentially succeed with some *very small probability*.

Asymptotic security is one common approach to capturing these notions. This approach views the running time of an adversary as well as its success probability as functions of a security parameter n.

Efficient algorithms

Efficient algorithms are defined to be *probabilistic* algorithms running in time *polynomial* in the security parameter n. We say that an algorithm with an input of size n is a polynomial-time algorithm if its worst-case running time is $O(n^c)$, for some constant c. A probabilistic algorithm is one that has the ability to toss *coins*, i.e. the algorithm has access to a source of randomness that yields unbiased random bits that are each independently equal to 1 with probability 1/2.

There are two reasons that in cryptography we consider probabilistic polynomial-time (PPT) algorithms rather than just deterministic polynomial-time algorithms. First,

randomness is essential to cryptography, e.g. for generating random keys, and thus it is also natural to consider a probabilistic adversary. Second, the capability to toss coins may provide additional power. Since we use the notion of efficient computation to model a realistic adversary, considering a probabilistic adversary is more desirable.

Negligible success probability

Cryptographic schemes that can be broken with a very small probability of success are still considered to be *secure* in modern cryptography. We define the notion of small probability of success by requiring the the success probability to be *smaller than any inverse-polynomial in n*. We call such a probability negligible, and formally define it as follows.

Definition 2.1 *A function f is negligible if, for every polynomial p, there exists an N such that $f(n) < 1/|p(n)|$ for all $n > N$.*

The following is an equivalent definition.

Definition 2.2 *A function f is negligible if, for every constant $c > 0$, there exists an N such that $f(n) < 1/n^c$ for all $n > N$.*

We typically denote an arbitrary negligible function by ϵ.

Asymptotic security definition

A definition of asymptotic security then takes the following general form.

> *A scheme is secure if every probabilistic polynomial-time adversary succeeds in breaking the scheme with only negligible probability.*

The above definition is *asymptotic* because it is possible that, for small values of n, an adversary can succeed with high probability.

2.3.2 Cryptographic hardness assumptions

In this section we introduce a class of cryptographic hardness assumptions regarding computations in *cyclic groups*.

Mathematical background

Let $\mathbb{G}$ be a finite multiplicative group of order n, and let $g \in \mathbb{G}$. The smallest positive integer t such that $g^t = 1$ is called the order of g; such a t always exists and must divide n. The set $\langle g \rangle = \{g^i | \ 0 \leq i \leq t-1\}$ of all powers of g is itself a group under the same operation as $\mathbb{G}$, and is called the cyclic subgroup of $\mathbb{G}$ generated by g. If $\mathbb{G}$ has an element g of order n, then $\mathbb{G}$ is called a cyclic group and g is called a generator of $\mathbb{G}$.

Analogous statements are true for additive groups. That is, if $\mathbb{G}$ is a finite additive group of order n and $g \in \mathbb{G}$, the order of g is the smallest positive divisor t of n such that $tg = 0$, and we write $\langle g \rangle = \{ig | \ 0 \leq i \leq t-1\}$. Here, tg denotes the element obtained by adding together t copies of g.

Menezes, van Oorschot, and Vanstone [101] give more detailed discussions of group and field.

The discrete logarithm (DL) and Diffie-Hellman (DH) assumptions

We now describe a number of computational problems that can be defined for any cyclic group. We first define an algorithm that generates cyclic groups as follows.

Definition 2.3 *Let $\mathcal{G}$ be a polynomial-time algorithm that, on input 1^n, outputs a description of a cyclic group $\mathbb{G}$ of order q and a generator $g \in \mathbb{G}$.*

The notation 1^n denotes constant bit string $11\ldots1$ of length n. Given a group-generating algorithm $\mathcal{G}$, algorithm $\mathcal{A}$, and parameter n, we consider the following experiment.

Experiment $\mathbf{Exp}^{\mathsf{dlog}}_{\mathcal{A},\mathcal{G}}(n)$

1. Run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$, as in the definition 2.3.
2. Compute $h := g^x$ for $x \in_{\mathsf{R}} \mathbb{Z}_q$.
3. Given $\mathbb{G}, q, g, h$, $\mathcal{A}$ outputs $x' \in \mathbb{Z}_q$.
4. The output of the experiment is defined to be 1 if $g^{x'} = h$, and 0 otherwise.

The notation $\mathbb{Z}_q$ denotes the group of integers $\{0, 1, \ldots, q-1\}$ under addition modulo q, and the notation $a \in_{\mathsf{R}} A$ denotes that a is selected uniformly at random from a finite set A. The aim of the algorithm $\mathcal{A}$ is, given $h \in \mathbb{G}$, is to find $x' \in \mathbb{Z}_q$ such that $g^{x'} = h$ $(\in \mathbb{G})$ in time polynomial in n. We denote $\mathbf{Exp}^{\mathsf{dlog}}_{\mathcal{A},\mathcal{G}}(n) = 1$ if the output of the experiment $\mathbf{Exp}^{\mathsf{dlog}}_{\mathcal{A},\mathcal{G}}(n)$ is 1, i.e. $\mathcal{A}$ is able to find such a x'. We then have the following.

Definition 2.4 (DL) *We say that the discrete logarithm (DL) problem is hard relative to $\mathcal{G}$ if, for all probabilistic polynomial-time algorithms $\mathcal{A}$, the function* $\Pr\left[\mathbf{Exp}^{\mathsf{dlog}}_{\mathcal{A},\mathcal{G}}(n) = 1\right]$ *is negligible.*

The discrete logarithm assumption is then that there exists a group-generating algorithm $\mathcal{G}$ for which the discrete logarithm problem is hard for all of the output groups. We next specify certain closely related problems, called the Diffie-Hellman problems. In particular we define two particularly important variants of the DH problem: the computational Diffie-Hellman (CDH) problem and the decisional Diffie-Hellman (DDH) problem.

Let $\mathbb{G}$ be defined as in Definition 2.3. Given two group elements h_1 $(= g^{x_1})$ and h_2 $(= g^{x_2})$, define $\mathsf{DH}_g(h_1, h_2) = g^{x_1 x_2}$. The CDH problem is then to compute $\mathsf{DH}_g(h_1, h_2)$ given $h_1, h_2 \in_{\mathsf{R}} \mathbb{G}$. The DDH problem is, given $h_1, h_2 \in_{\mathsf{R}} \mathbb{G}$ and a candidate solution $h' \in \mathbb{G}$, to decide whether or not $h' = \mathsf{DH}_g(h_1, h_2)$. More formally, given $\mathcal{G}$, the DDH problem is defined as follows.[1]

Definition 2.5 (DDH) *We say that the decisional Diffie-Hellman (DDH) problem is hard relative to $\mathcal{G}$ if, for all probabilistic polynomial-time algorithm $\mathcal{A}$, the function*

[1] We omit a formal description of the CDH problem since it is not used in the remainder of the thesis.

$$\mathsf{Adv\text{-}DDH} = \Big|\Pr\big[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^z) = 1\big] - \Pr\big[\mathcal{A}(\mathbb{G}, q, g, g^x, g^y, g^{xy}) = 1\big]\Big|$$

is negligible, where in each case the probabilities are taken over the experiment in which $\mathcal{G}(1^n)$ *outputs* $(\mathbb{G}, q, g)$ *and* $x, y, z \in_R \mathbb{Z}_q$ *are chosen at random.*

If the DL problem relative to some $\mathcal{G}$ is easy, then so is the CDH problem. It is not clear, however, whether the hardness of the DL problem necessarily implies the hardness of CDH problem. Again, if the CDH problem relative to some $\mathcal{G}$ is easy, then so is the DDH problem. The converse does not appear to be true; there are examples of groups in which the DL and CDH problems are believed to be hard, even though the DDH problem is easy [82]. The relationships between the problems is summarised in Figure 2.1.

DL problem is easy $\implies$ CDH problem is easy $\implies$ DDH problem is easy

Figure 2.1: Relationships between problems

2.4 Secret Key Cryptography

In this section, we briefly describe secret key primitives.

2.4.1 Secret key encryption

We give a definition of computational security for secret key encryption. We first define the notion of secret key encryption.

Definition 2.6 *A secret key encryption scheme is a triple of polynomial-time algorithms* (Gen,Enc,Dec) *such that:*

1. *The key generation algorithm Gen takes as input the security parameter* 1^n *and outputs a key* k *such that* $|k| \geq n$*. We write* $k \leftarrow \mathsf{Gen}(1^n)$*.*

2. *The* encryption algorithm Enc *takes as input* k *and a plaintext message* $m \in \{0,1\}^*$*, and outputs a ciphertext* c*. We write* $c \leftarrow \mathsf{Enc}(k, m)$*.*
3. *The* decryption algorithm Dec *takes as input a key* k *and a ciphertext* c*, and outputs a message* m *or a special symbol* $\perp$ *denoting failure. Assuming that* Dec *is deterministic, we write* $m \leftarrow \mathsf{Dec}(k, c)$*.*

It is required that, for every n*, every* $k \leftarrow \mathsf{Gen}(1^n)$*, and every* $m \in \{0,1\}^*$*,* $\mathsf{Dec}\big(k, \mathsf{Enc}(k, m)\big) = m$*.*

We next define the security notion for a secret key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, i.e. indistinguishability in the presence of an adversary. More specifically, given a secret key encryption scheme Π and an adversary $\mathcal{A}$, we consider the following experiment.

Experiment $\mathbf{Exp}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$

1. $\mathcal{A}$ is given input 1^n, and outputs a pair of messages (m_0, m_1) of the same length.
2. A key k is generated by running $\mathsf{Gen}(1^n)$, and a random bit $b \in_{\mathsf{R}} \{0,1\}$ is chosen. A ciphertext $c \leftarrow \mathsf{Enc}(k, m_b)$ is computed and given to $\mathcal{A}$.
3. $\mathcal{A}$ outputs a bit b'.
4. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

$\mathcal{A}$ is then said to *succeed* if $\mathbf{Exp}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1$. An encryption scheme is said to have the indistinguishability property if the success probability of any PPT adversary $\mathcal{A}$ is at most negligibly greater than $1/2$. It can be defined more formally as follows.

Definition 2.7 *A secret key encryption scheme* Π=(Gen,Enc,Dec) *is* indistinguishable in the presence of an adversary *if, for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the function* $\left|\Pr\left[\mathbf{Exp}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2}\right|$ *is negligible.*

2.4.2 Hash functions

Cryptographic hash functions (hereafter, simply hash functions) are functions that take arbitrary-length strings and *compress* them into shorter strings. Hash functions play a fundamental role in modern cryptography, and can be defined as follows.

Definition 2.8 *A hash function is a pair of probabilistic polynomial-time algorithms* (Gen,H) *with the following properties:*

- Gen *is a probabilistic algorithm which takes as input a security parameter* 1^n *and outputs a key* s*.*
- H *takes as input a key* s *and a string* $x \in \{0,1\}^*$*, and outputs a string* $H^s(x) \in \{0,1\}^{l(n)}$*, for a polynomial* l*.*

The input value x is a string of arbitrary length, but we assume that there exists some upper bound on the length of possible input strings. In practice, it is always the case that hash functions are only defined for strings of bounded length. The "key" s is not a key in the usual sense of the word. It is not kept secret, and is rather used to specify (or index) a particular function H^s from a *family* of hash functions. We define the following experiment, given a hash function $\Pi = (\mathsf{Gen}, H)$ and an adversary $\mathcal{A}$.

Experiment $\mathbf{Exp}^{\mathsf{coll}}_{\mathcal{A},\Pi}(n)$

1. A key s is generated by running $\mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given s and outputs x, x'.
3. The output of the experiment is defined to be 1 if $x \neq x'$ and $H^s(x) = H^s(x')$.

A hash function is said to be collision resistant if no efficient algorithm can find a collision in the above experiment except with negligible probability. The formal definition is as follows.

Definition 2.9 *A hash function* Π=(Gen,H) *is collision resistant if, for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the function* $\Pr\left[\mathbf{Exp}^{\mathsf{coll}}_{\mathcal{A},\Pi}(n) = 1\right]$ *is negligible.*

For simplicity and depending on the context, we refer to each of H, H^s, and $\Pi = (\mathsf{Gen}, H)$ as collision-resistant hash functions.

We now define the notion of security for a hash function. Collision resistance is a strong security requirement and is quite difficult to achieve, and thus in some applications we can relax the requirements somewhat. Typically, three security properties for a hash function are defined.

- Collision resistance: This is defined as above.
- Second pre-image resistance: Informally speaking, a hash function is second pre-image resistant if it is infeasible for a PPT adversary, given a randomly chosen x, to find x' ($\neq x$) such that $H^s(x) = H^s(x')$.
- Pre-image resistance: Informally, a hash function is pre-image resistant if it is infeasible for a PPT adversary, given y ($= H^s(x)$ for a randomly chosen x), to find a value x' such that $H^s(x') = y$.

It is easy to see that any collision-resistant hash function is second pre-image resistant, i.e. collision resistance implies second pre-image resistance. Collision resistance, however, does not guarantee pre-image resistance. Furthermore, second pre-image resistance does not guarantee pre-image resistance, nor does pre-image resistance guarantee second pre-image resistance [102].

2.4.3 Message authentication codes

In practice, it is often necessary to guarantee message integrity. That is, each communicating party should be able to verify that, when it receivers a message, the message is the exactly message sent by the other party. It is tempting to suggest that secret key encryption could also provide message authentication, since an adversary cannot possibly modify a ciphertext in a meaningful way. This reasoning is, however, false (as described in [82]), and thus a separate mechanism is required.

A message authentication code (MAC) is a mechanism enabling communicating parties to check whether or not a message has been tampered with. A MAC can be used between parties only when they share a secret. We now give a formal definition of a MAC.

Definition 2.10 (MAC) *A message authentication code (MAC) is a triple of polynomial-time algorithms (Gen,Mac,Ver) such that:*

1. *The key generation algorithm Gen takes as input the security parameter 1^n and outputs a key k with $|k| \geq n$. We write $k \leftarrow \mathsf{Gen}(1^n)$.*
2. *The tag generation algorithm Mac takes as input a key k and a message $m \in \{0,1\}^*$, and outputs a tag t. We write $t \leftarrow \mathsf{Mac}(k, m)$.*

3. *The verification algorithm Ver takes as input a key k, a message m, and a tag t. It outputs a bit $b = 1$ if t is valid, and $b = 0$ otherwise. We write $b \leftarrow \mathsf{Ver}(k, m, t)$.*

It is required that, for every n, every key $k \leftarrow \mathsf{Gen}(1^n)$, and every $m \in \{0,1\}^*$, $\mathsf{Ver}(k, m, \mathsf{Mac}(k, m)) = 1$.

A MAC can be used in the following way. When two parties wish to communicate in an authenticated manner, they run $\mathsf{Gen}(1^n)$ and share a secret k before they start communication. When one party wants to send a message m to the other, she computes a MAC tag $t \leftarrow \mathsf{MAC}(k, m)$, and transmits (m, t). Upon receipt of (m, t), the second party verifies that t is a valid tag on the message m with respect to k (by checking if $\mathsf{Ver}(k, m, t) = 1$).

A MAC is said to be *secure* if no PPT adversary is able to generate a valid tag on any *new* message that was not previously sent by one of the communicating parties. The formal security definition requires considering the following experiment for a message authentication code $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Ver})$ and an adversary $\mathcal{A}$.

Experiment $\mathbf{Exp}^{\mathsf{forge}}_{\mathcal{A},\Pi}(n)$

1. A random key k is generated by running $\mathsf{Gen}(1^n)$.
2. $\mathcal{A}$ is given input 1^n and oracle access $\mathsf{Mac}(k, \cdot)$, and eventually outputs a pair (m, t).
3. The output of the experiment is defined to be 1 if (i) $\mathsf{Ver}(k, m, t) = 1$ and (ii) $m \notin Q$, where Q is the set of queries that $\mathcal{A}$ has asked to the oracle.

An oracle $\mathsf{Mac}(k, \cdot)$, given a query $m' \in \{0,1\}^*$ from $\mathcal{A}$, returns $t \leftarrow \mathsf{Mac}(k, m')$ and adds m' to a set Q. As formally defined below, a MAC is said to be secure if no PPT adversary can succeed in the above experiment with non-negligible probability.

Definition 2.11 *A message authentication code $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Ver})$ is secure if, for all probabilistic polynomial-time adversaries $\mathcal{A}$, the function $\Pr\left[\mathbf{Exp}^{\mathsf{forge}}_{\mathcal{A},\Pi}(n) = 1\right]$ is negligible.*

2.5 Public Key Cryptography

In this section we briefly describe public key primitives.

2.5.1 Public key encryption

We first review public key encryption schemes, and, as an example, give the ElGamal encryption scheme that is referred to in Chapter 5 and 6.

Definition and security notions

A public key encryption scheme can be defined as follows.

Definition 2.12 *A public key encryption scheme is a triple of polynomial-time algorithms* (Gen,Enc,Dec) *with the following properties:*

1. *The key generation algorithm Gen takes as input the security parameter* 1^n *and outputs a public/private key pair* (pk, sk)*. We write* $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$*.*
2. *The encryption algorithm Enc takes as input a public key* pk *and a message* m *from some underlying plaintext space (that may depend on* pk*), and outputs a ciphertext* c*. We write* $c \leftarrow \mathsf{Enc}(pk, m)$*.*
3. *The decryption algorithm Dec takes as input a private key* sk *and a ciphertext* c*, and outputs a message* m *or a special symbol* $\perp$ *denoting failure. We assume without loss of generality that Dec is deterministic, and we write* $m \leftarrow \mathsf{Dec}(sk, c)$*.*

It is required that, for every n*, every* $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$*, and every message* m *in the appropriate underlying plaintext space,* $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m$*.*

We now describe two security notions for public key encryption schemes. First, given a public key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary $\mathcal{A}$, we define the following experiment.

Experiment $\mathbf{Exp}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$

1. $\mathsf{Gen}(1^n)$ is run to obtain keys (pk,sk).
2. Given pk, $\mathcal{A}$ outputs a pair of messages (m_0, m_1) of the same length, where these messages must be in the plaintext space associated with pk.
3. A random bit b is chosen, i.e. $b \in_{\mathsf{R}} \{0,1\}$, and the ciphertext $c \leftarrow \mathsf{Enc}(pk, m_b)$ is given to $\mathcal{A}$.
4. $\mathcal{A}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

We then have the following.

Definition 2.13 (IND-CPA) *A public key encryption scheme* Π=(*Gen,Enc,Dec*) *is indistinguishable under a chosen-plaintext attack* (*or is CPA secure*) *if, for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the function* $\left|\Pr\left[\mathbf{Exp}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] - \frac{1}{2}\right|$ *is negligible.*

Second, given a public key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary $\mathcal{A}$, we define the following experiment.

Experiment $\mathbf{Exp}^{\mathsf{cca}}_{\mathcal{A},\Pi}(n)$

1. $\mathsf{Gen}(1^n)$ is run to obtain keys (pk,sk).
2. $\mathcal{A}$ is given pk and access to a decryption oracle $\mathsf{Dec}(sk, \cdot)$. $\mathcal{A}$ outputs a pair of messages (m_0, m_1) of the same length, where these messages must be in the plaintext space associated with pk.
3. A random bit b is chosen, i.e. $b \in_{\mathsf{R}} \{0,1\}$, and the ciphertext $c \leftarrow \mathsf{Enc}(pk, m_b)$ is given to $\mathcal{A}$.
4. $\mathcal{A}$ continues to interact with the decryption oracle $\mathsf{Dec}(sk, \cdot)$, but may not request a decryption of c itself. Finally, $\mathcal{A}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

We then have the following.

Definition 2.14 (IND-CCA2) *A public key encryption scheme* Π=(*Gen,Enc,Dec*) *is indistinguishable under a chosen-ciphertext attack* (*or is CCA secure*) *if, for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the function* $\left|\Pr\left[\mathbf{Exp}^{\mathsf{cca}}_{\mathcal{A},\Pi}(n)=1\right]-\frac{1}{2}\right|$ *is negligible.*

Another commonly used means of formalising the above notions is to require that every adversary *behaves the same way* whether it sees an encryption of m_0 or an encryption of m_1. Since the adversary $\mathcal{A}$ outputs a single bit, "behaving the same way" means that $\mathcal{A}$ outputs 1 with almost the same probability in each case. Putting sec to either cpa or cca, we define the experiments $\mathbf{Exp}^{\mathsf{sec}}_{\mathcal{A},\Pi}(n,b)$ to be as above, except that a fixed bit b is used rather than being randomly chosen.

The following definitions then capture the notion that $\mathcal{A}$ cannot determine whether it is running experiment $\mathbf{Exp}^{\mathsf{sec}}_{\mathcal{A},\Pi}(n,0)$ or experiment $\mathbf{Exp}^{\mathsf{sec}}_{\mathcal{A},\Pi}(n,1)$.

Definition 2.15 *For* $\mathsf{sec}\in\{\mathsf{cpa},\mathsf{cca}\}$ *we define the advantage of an adversary* $\mathcal{A}$ *to be*

$$\mathbf{Adv}^{\mathsf{sec}}_{\mathcal{A},\Pi}(n)=\left|\Pr[\left(\mathbf{Exp}^{\mathsf{sec}}_{\mathcal{A},\Pi}(n,0)\right)=1]-\Pr[\left(\mathbf{Exp}^{\mathsf{sec}}_{\mathcal{A},\Pi}(n,1)\right)=1]\right|.$$

Definition 2.16 *A public key encryption scheme* $\Pi=(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ *is CPA secure* (*respectively, CCA secure*) *if the function* $\boldsymbol{Adv}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$ (*respectively,* $\boldsymbol{Adv}^{\mathsf{cca}}_{\mathcal{A},\Pi}(n)$) *is negligible for all probabilistic polynomial-time adversaries* $\mathcal{A}$*.*

Definition 2.16 can be shown to be equivalent to definitions 2.13 or 2.14 [82].

ElGamal encryption

We describe a well-known public key encryption scheme, namely the ElGamal encryption scheme, and discuss its security.

Definition 2.17 (ElGamal Encryption) *Let* $\mathcal{G}$ *be defined as in definition 2.3. The ElGamal encryption scheme is a triple of algorithms* (*Gen,Enc,Dec*) *with the following properties:*

- *Gen: takes as input* 1^n*, and obtains* $(\mathbb{G}, q, g)$ *by running* $\mathcal{G}(1^n)$*. It then chooses a random* $x \in_R \mathbb{Z}_q$*, computes* $y = g^x$*, and outputs a public/private key pair* (pk,sk)*, where* $pk = \langle \mathbb{G}, q, g, y \rangle$ *and* $sk = \langle \mathbb{G}, q, g, x \rangle$*.*
- *Enc: takes as input a public key* pk *and a message* $m \in \mathbb{G}$*. It chooses a random* $r \in_R \mathbb{Z}_q$ *and computes* $c_1 = g^r$ *and* $c_2 = my^r$*. Finally, it outputs the ciphertext* $c = (c_1, c_2)$*.*
- *Dec: takes as input a private key* sk *and a ciphertext* c*, and outputs* $m = c_2 \cdot c_1^{-x}$*.*

It is easy to see that, for every n, every $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$, and every message $m \in \mathbb{G}$, $\mathsf{Dec}\big(sk, \mathsf{Enc}(pk, m)\big) = m$. We then have the following result regarding the security of this scheme [134].

Theorem 2.1 *If the DDH problem is hard relative to* $\mathcal{G}$*, then the ElGamal encryption scheme has IND-CPA property (i.e. it is CPA secure).*

The ElGamal encryption scheme is, however, vulnerable to chosen-ciphertext attack as a result of the homomorphic property of ElGamal encryption scheme. That is, for any public key pk and any messages m_1 and m_2, we have

$$\mathsf{Enc}(pk, m_1) \cdot \mathsf{Enc}(pk, m_2) = \mathsf{Enc}(pk, m_1 m_2).^2$$

For example, in the experiment $\mathbf{Exp}^{\mathsf{cca}}_{\mathcal{A},\Pi}$, assume that $\mathcal{A}$ receives $c = \mathsf{Enc}(pk, m_b)$ in step 3. In step 4, $\mathcal{A}$ can send $\mathsf{Enc}(pk, m_b) \cdot \mathsf{Enc}(pk, m_1)$ to the decryption oracle and obtain $m_b m_1$. $\mathcal{A}$ is then able to correctly determine b with knowledge of m_0 and m_1.

2.5.2 Digital signature schemes

Digital signatures are the public key counterpart of MACs, in that they are used to ensure the integrity (or authenticity) of transmitted messages. Although typical implementations of digital signature schemes are 2–3 orders of magnitude less efficient than those of MACs, digital signatures have some advantages.

[2] The notation $\mathsf{Enc}(pk, m_1) \cdot \mathsf{Enc}(pk, m_2)$ means that the sequence of elements should be multiplied component-wise.

First, use of such a scheme simplifies key management when a sender communicates with multiple receivers. Instead of establishing distinct shared secrets with each recipient and computing a separate MAC tag for each secret key, the sender only needs to compute a single signature which can be verified by all recipients. Second, digital signatures provide the important security property of non-repudiation, i.e. once a signer signs a message, he/she cannot later deny having done so.

We next give a formal definition of a digital signature scheme [82].

Definition 2.18 (Signature Scheme) *A signature scheme is a triple of three polynomial-time algorithms* (Gen,Sig,Ver) *with the following properties:*

1. *The key-generation algorithm Gen takes as input a security parameter* 1^n *and outputs a pair of keys* (pk, sk)*, where* pk *is called the public key and* sk *is called the private key. We write* $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$.
2. *The signing algorithm Sig takes as input a private key* sk *and a message* $m \in \{0,1\}^*$*, and outputs a signature* σ*. We write* $\sigma \leftarrow \mathsf{Sig}(sk, m)$.
3. *The deterministic verification algorithm Ver takes as input a public key* pk*, a message* m*, and a signature* σ*. It outputs a bit* $b = 1$ *if* σ *is valid, and* $b = 0$ *otherwise. We write* $b \leftarrow \mathsf{Ver}(pk, m, \sigma)$.

It is required that, for every n*, every* $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$*, and every message* $m \in \{0,1\}^*$*,* $\mathsf{Ver}\big(pk, m, \mathsf{Sig}(sk, m)\big) = 1$.

A signature scheme is used in the following way. A sender S runs $\mathsf{Gen}(1^n)$ to obtain (pk, sk), and the public key pk is publicised as belonging to S. If S wishes to send a protected version of message m, S computes $\sigma \leftarrow \mathsf{Sig}(sk, m)$ and sends (m, σ). A receiver with the authentic copy of pk can verify the authenticity of m by checking whether the signature σ is *valid*. A signature σ is said to be valid on a message m if $\mathsf{Ver}(pk, m, \sigma) = 1$.

We now consider the security of signature schemes, following Goldwasser et al. [61]. An adversary is given a public key and allowed to repeatedly ask for signatures on multiple messages of his choice. The adversary succeeds if he can output a valid signature on any message which was not signed previously. The signature scheme is said to be secure against existential forgery under adaptive chosen-message attack if the success probability of any PPT adversary is negligible. The formal security

definition requires considering the following experiment for a digital signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ and an adversary $\mathcal{A}$.

Experiment $\mathbf{Exp}^{\text{e-forge}}_{\mathcal{A},\Pi}(n)$

1. $\mathsf{Gen}(1^n)$ is run to obtain keys (pk,sk).
2. $\mathcal{A}$ is given pk and access to an oracle $\mathsf{Sig}(sk, \cdot)$, and eventually outputs a pair (m, σ).
3. The output of the experiment is defined to be 1 if (i) $\mathsf{Ver}(pk, m, \sigma) = 1$ and (ii) $m \notin Q$, where Q is the set of queries that $\mathcal{A}$ has asked to the oracle.

An oracle $\mathsf{Sig}(sk, \cdot)$, given a query $m' \in \{0,1\}^*$ from $\mathcal{A}$, returns $\sigma \leftarrow \mathsf{Sig}(sk, m')$ and adds m' to a set Q. As formally defined below, a digital signature scheme is said to be existentially unforgeable under an adaptive chosen-message attack if no PPT adversary can succeed in the above experiment with non-negligible probability.

Definition 2.19 *A digital signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is existentially unforgeable under an adaptive chosen-message attack if, for all probabilistic polynomial-time adversaries $\mathcal{A}$, the function $\Pr\left[\mathbf{Exp}^{\text{e-forge}}_{\mathcal{A},\Pi}(n) = 1\right]$ is negligible.*

2.6 Key Management

Key management covers the roles, techniques, and procedures used to establish shared secret keys (as used by secret key cryptography) and to provide trustworthy copies of public keys for public key cryptosystems. We start by briefly describing the role of trusted third parties in key management. We then review key management frameworks, and in particular explain the notion of a public key infrastructure. Finally, we describe distributed key generation schemes which do not involve trusted third parties.

2.6.1 Trusted third parties

A trusted third party (TTP) is an entity in a system that is not under the control of that system's security authority and yet is trusted by that security authority to carry out certain security-related functions [35]. TTPs are often heavily involved in key management schemes. A TTP can be involved in key generation, key distribution, or the certification of public keys.

There are several types of TTPs, but they may be classified into two categories in the context of key management [102]: a TTP is said to be unconditionally trusted if it is trusted on all matters, i.e. it may have access to the secret and private keys of users, and functionally trusted if it is assumed to be honest and fair but it does not have access to the secret or private keys of users.

Dent and Mitchell [35] and Menezes, van Oorschot, and Vanstone [102] give more detailed discussions of TTPs, covering topics such as requirements, architectures, and related standards.

2.6.2 Key management frameworks

In this section, we describe basic key management frameworks.

Definitions and basic properties

A key is a sequence of symbols that controls the operation of a cryptographic transformation, and keying material is the data (e.g. keys and initialisation values) necessary to establish and maintain cryptographic keying relationships [35]. Key management is concerned with all operations related to keys *except* their actual use by cryptographic algorithms, and it can be defined as follows [102].

Definition 2.20 *Key management is the set of processes and mechanisms which support key establishment and maintenance of ongoing keying relationships between parties, including replacing older keys with new keys as necessary.*

In the above definition, key establishment means any process whereby a shared secret key becomes available to two or more parties for subsequent cryptographic use. Although there are mathematical security models for certain parts of key management systems such as key establishment protocols, it is difficult to define a formal security model for key management due to the complexity of the whole system. For this reason, key management principle have been developed as a series of statements about best practice obtained from practical experience. Dent and Mitchell [35] give the following main threats on key management systems.

(T1) The unauthorised disclosure of keys.

(T2) The unauthorised modification of keys.

(T3) The misuse of keys, i.e. (a) the use of a key by an unauthorised party, (b) the use of a key for an unauthorised purpose, or (c) the use of a key whose usage period has expired.

Finally, we briefly discuss one important key management principle, namely key separation. This requires that a key should be used for one purpose only, e.g. an encryption key should never be used with a MAC algorithm. A key hierarchy is a way of facilitating key separation by adding structure to a set of keys, and by defining the scope of the use of each key. More specifically, keys are classified in terms of levels of importance, and keys at one level are only used for protecting keys in the level directly below, except for the keys at the lowest level. Dent and Mitchell [35] give more detailed discussions of key management and related standards, such as ANSI X9.24.

Key management for secret key techniques

A major issue when using secret key cryptography is to establish pairwise secrets. In a network consisting of n parties, up to $\binom{n}{2} = \frac{n(n-1)}{2}$ pairwise shared secrets may need to be established. In practice, networks are often large, and thus more efficient and simple methods of handling this problem are required. One solution using secret key techniques involves an unconditionally trusted TTP, called a key distribution centre (KDC). Each entity is only required to share a unique secret key

with a KDC, and, when two entities wish to establish a shared secret key, the necessary secret can be transferred from the KDC to the two entities in encrypted form.

Key management for public key techniques

The key management problem is rather different for public key techniques. Whereas secret key cryptography requires shared secrets, the use of public key cryptography requires a means to provide assurance regarding the validity of public keys.

This can be achieved by using a functionally trusted TTP, called a certification authority (CA), which issues digitally signed statements (i.e. certificates) binding public keys to particular entities. More specifically, using a secure signature scheme, a CA signs a data structure containing an entity's identity and its public key, after having verified the identity of the entity. The original data structure, i.e. an entitiy's identity and its public key, combined with its signature generated by CA is called a certificate. Some assurance (the degree of which depends on the CA and the rigour of its operational procedures) in a public key can be now obtained by any entity that verifies the certificate using the CA's public key.

2.6.3 Public key infrastructures

When using a CA, a number of serious issues need to be addressed, including: how an entity obtains the necessary trusted copy of the CA's public key; how a CA can verify the binding between an entity's identity and its public key; how an entity decides whether or not to trust a CA; and how a CA can efficiently and securely revoke certificates when necessary. A public key infrastructure (PKI) includes the set of policy statements covering such details, and enables the generation and use of certified public keys in a community accepting the policy rules of the PKI.

The simplest PKI would involve a single CA which issues all the certificates for a particular PKI. However, use of a single CA can result in a range of problems, and thus multiple CAs can also be used. To avoid the need for every user of a PKI to hold trusted copies of the public keys of every CA, CAs can issue certificates

for each other's public keys, resulting in the use of sequences of certificates known as certificate chains. A multiplicity of CAs can be arranged hierarchically (making certificate chain construction simple) or in a 'flat' peer-to-peer scheme.

In some cases a certificate for an entity's public key may need to be withdrawn, e.g. if a user leaves an organisation or a user's private key is compromised. Handling these issues can be a non-trivial problem. We briefly discuss two simple approaches.

- Expiry: A certificate can be withdrawn by including an expiry date in the certificate. For example, a certificate for Bob's public key $pk_{\mathcal{B}}$ might have the form:

$$\mathsf{cert}_{\mathcal{B}} := \mathsf{Sig}(sk, \texttt{`Bob's key is } pk_{\mathcal{B}}\texttt{'} \,||\, \texttt{expiry date}),$$

  where Sig is a signing algorithm (as in definition 2.18) and sk is the private key of the CA. This approach gives a very coarse-grained solution. For example, if an employee receives a certificate that expires after one year, then it will remain valid for up to a year after an employee leaves. Issuing certificates on a daily basis, however, could be very costly.

- Revocation: A certificate can be withdrawn immediately by a CA explicitly *revoking* the certificate. For example, a CA might issue a certificate for Bob's public key $pk_{\mathcal{B}}$ with the following form:

$$\mathsf{cert}_{\mathcal{B}} := \mathsf{Sig}(sk, \texttt{`Bob's key is } pk_{\mathcal{B}}\texttt{'} \,||\, \texttt{serial number}).$$

  A certificate can be immediately revoked by the CA signing a certificate revocation list (CRL) containing the serial numbers of all currently revoked certificates. This CRL includes the current date, and needs to be made available to all PKI users.

More detailed discussions of PKI issues can be found in Katz and Lindell [82] and Menezes, van Oorschot, and Vanstone [102].

2.6.4 Distributed key generation

Pedersen [111] first proposed a solution to the distributed generation of private keys for threshold cryptosystems, and the variant schemes have subsequently appeared in

the literatures [25, 55, 56, 96, 129]. Such schemes involves the n parties choosing the secret key and distributing it verifiably among themselves, and thus it is infeasible for a single party to compute or reconstruct the private key. The distributed key generation schemes can also be used as a building block for other distributed protocols [48, 55, 68]. For example, Boneh and Franklin [19] propose the use Gennaro et al.'s scheme [56] to generate a master secret for use of the Distributed Private Key Generator (DPKG).[3] After briefly discussing the generic syntax and security model of the distributed key generation schemes, we describe example schemes considered in this thesis afterwards.

System model

We assume that a distributed key generation scheme involves a set of n parties, denoted by $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$. They have secure communication channels between them and have access to a dedicated broadcast channel. For simplicity, we assume a fully synchronous communication model, i.e. messages of a given round in the protocol are sent simultaneously by all the parties, and are delivered simultaneously to their recipients.

We also define a PPT adversary $\mathcal{A}$, which can corrupt up to t ($< n/2$) of the n parties in the system, and may cause the corrupted parties to deviate from the protocol specification. We assume that an adversary $\mathcal{A}$ is static, i.e. it chooses the corrupted parties at the beginning of the protocol.

Security requirements

Let $\mathbb{G}$ be a group of a large prime order q. A distributed key generation protocol is performed by n parties and generates (i) private outputs $x_1, x_2, \ldots, x_n$, called the shares of a secret $x \in \mathbb{Z}_q^*$, and (ii) a public output $y \in \mathbb{G}$. The protocol is called t-secure (or secure with threshold t) if, given an adversary that corrupts at most t out of n parties in the system, the following requirements are satisfied:

(R1) All subsets of $t+1$ shares provided by honest parties define the same unique secret key x.

[3] See Section 4.1.2 for the details.

(R2) All honest parties have the same public value y.

(R3) x is uniformly distributed in $\mathbb{Z}_q$, and y depends on x such that y will be uniformly distributed in $\mathbb{G}$.

(R4) No information regarding x can be learned by an adversary.

Gennaro et al. [56] call R1, R2, and R3 the correctness requirements, and R4 the secrecy requirement.

Pedersen's scheme

Pedersen [111] proposed a distributed key generation protocol which distributes a master secret without relying on a trusted third party. Pedersen's scheme involves n parallel executions of Feldman's scheme [45]. We give the detailed description as follows.

1. Each party $\mathcal{P}_i$ $(1 \leq i \leq n)$ chooses a polynomial $f_i(z) \in \mathbb{Z}_q[z]$ of degree t, i.e. $f_i(z) = a_{i0} + a_{i1}z + \cdots + a_{it}z^t$, where $a_{ik} \in_R \mathbb{Z}_q$.

2. Each $\mathcal{P}_i$ broadcasts $w_{ik} = g^{a_{ik}} \bmod p$ for $k = 0, 1, \ldots, t$.

3. Each $\mathcal{P}_i$ computes $x_{ij} = f_i(j) \bmod q$ for $j = 1, \ldots, n$, and sends x_{ij} to P_j via a secure channel, reserving x_{ii} for itself.

4. On receiving x_{ji} $(1 \leq j \leq n, j \neq i)$, each $\mathcal{P}_i$ verifies x_{ji} by checking that

$$g^{x_{ji}} = \prod_{k=0}^{t} (w_{jk})^{i^k} \bmod p. \tag{2.1}$$

   If this check fails for an index j, $\mathcal{P}_i$ broadcasts a *complaint* against $\mathcal{P}_j$.

   (a) If there are complaints against $\mathcal{P}_j$ from more than t parties, $\mathcal{P}_j$ is disqualified.

   (b) Otherwise, i.e. if there are at most t complaints against $\mathcal{P}_j$, $\mathcal{P}_j$ reveals (i.e. broadcasts) the share x_{ji} satisfying equation (2.1). If this revealed share does not satisfy the equation (2.1),[4] $\mathcal{P}_j$ is disqualified.

   The qualified parties then form a set $\mathcal{Q}$.

[4] This check is performed by all parties.

5. Each party $\mathcal{P}_i$ computes its share as $x_i = \sum_{j \in \mathcal{Q}} x_{ji} \bmod q$. The public value y $(= g^x \bmod p)$ is computed as $y = \prod_{i \in \mathcal{Q}} w_{i0} \bmod p$.

The master secret value x is not computed by any party, but is equal to $x = \sum_{i \in \mathcal{Q}} a_{i0} \bmod q$. If we define the polynomial $f(z) = \sum_{i \in \mathcal{Q}} f_i(z) \in \mathbb{Z}_q[z]$, it is easy to see that $x_i = f(i) \bmod q$ for every $i \in \mathcal{Q}$, and thus $x = f(0) \bmod q$.

Gennaro et al. [56], however, show that an adversary which corrupts a small number of parties can influence key generation. For example, an adversary can bias the distribution of a public key y, making its last bit significantly more likely to be 0 than 1. Suppose that an adversary corrupts two parties, say $\mathcal{P}_1$ and $\mathcal{P}_2$, and all the other parties are honest, i.e. they correctly follow the protocol specifications. In step 3, $\mathcal{P}_1$ sends $\mathcal{P}_i$ incorrect shares x_{1i} for $i = 3, \ldots, t+2$, which will result in t complaint in the subsequent step. One more complaint is now required for the disqualification of party $\mathcal{P}_1$. The adversary computes $\alpha = \prod_{i=1}^{n} w_{i0} \bmod p$ and $\beta = \prod_{i=2}^{n} w_{i0} \bmod p$. If α ends with 0, then $\mathcal{P}_2$ does not broadcast a compliant against $\mathcal{P}_1$. Otherwise, $\mathcal{P}_2$ broadcasts a complaint, and thus $\mathcal{P}_1$ is disqualified, in which case the probability that β ends with 0 is $1/2$. Hence, the probability that the public key y ends with 0 is $1/2 + (1/2)^2 = 3/4$ rather than $1/2$, and the security condition R3 of the secret sharing scheme is not satisfied.

Gennaro et al.'s scheme

Gennaro et al. [56] fixes the above problem and show that the proposed scheme satisfies the the requirements R1, R2, R3, and R4 with threshold t $(< n/2)$. Let p be a large prime and g be an element of large prime order q, where $q|(p-1)$. The $\tilde{g}$ is an element of a group generated by g. The secret key x can be generated as follows.

1. Each $\mathcal{P}_i$ $(1 \le i \le n)$ chooses two polynomials $f_i(z), \tilde{f}_i(z) \in \mathbb{Z}_q[z]$ of degree t:

$$f_i(z) = a_{i0} + a_{i1}z + \cdots + a_{it}z^t \quad \text{and} \quad \tilde{f}_i(z) = \tilde{a}_{i0} + \tilde{a}_{i1}z + \cdots + \tilde{a}_{it}z^t$$

   where $a_{ik}, \tilde{a}_{ik} \in_{\mathsf{R}} \mathbb{Z}_q$ $(0 \le k \le t)$. Each $\mathcal{P}_i$ broadcasts $w_{ik} = g^{a_{ik}} \tilde{g}^{\tilde{a}_{ik}} \bmod p$ for $k = 0, 1, \ldots, t$.

2. Each $\mathcal{P}_i$ computes the pair $\left(x_{ij} = f_i(j) \bmod q,\ \tilde{x}_{ij} = \tilde{f}_i(j) \bmod q\right)$ for $j =$

$1, \ldots, n$, and sends $(x_{ij}, \tilde{x}_{ij})$ to $\mathcal{P}_j$ via a secure channel, reserving $(x_{ii}, \tilde{x}_{ii})$ for itself.

3. On receiving $(x_{ji}, \tilde{x}_{ji})$, each $\mathcal{P}_i$ verifies it for $j = 1, 2, \ldots, n$ by checking that:

$$g^{x_{ij}} \tilde{g}^{\tilde{x}_{ij}} = \prod_{k=0}^{t} (w_{ik})^{j^k} \bmod p. \tag{2.2}$$

If this check fails for an index j, $\mathcal{P}_i$ broadcasts a *complaint* against $\mathcal{P}_j$.

4. If any $\mathcal{P}_i$ receives a complaint from $\mathcal{P}_j$, it broadcasts the pair $(x_{ij}, \tilde{x}_{ij})$.

5. A PKG $\mathcal{P}_i$ becomes *disqualified* if:

   - there are complaints against $\mathcal{P}_i$ from more than t parties; or
   - the pair broadcast by $\mathcal{P}_i$ in step 4 does not satisfy equation (4.1).

   The non-disqualified PKGs then form a set $\mathcal{Q}$.

6. Each $\mathcal{P}_i$ computes its share as the following pair:

$$\big(x_i = \sum_{j \in \mathcal{Q}} x_{ji} \bmod q,\ \tilde{x}_i = \sum_{j \in \mathcal{Q}} \tilde{x}_{ji} \bmod q\big).$$

   The master secret value x satisfies $x = \sum_{i \in \mathcal{Q}} a_{i0} \bmod q$, and is not available to any party.

The extraction process for the corresponding public value $y = g^x \bmod p$ can be performed as follows. Each $\mathcal{P}_i$ broadcasts $A_{ik} = g^{a_{ik}} \bmod p$ $(k = 0, 1, \ldots, t)$ and then every $\mathcal{P}_i$ verifies the broadcast values by checking whether

$$g^{x_{ji}} = \prod_{k=0}^{t} i^k A_{ik} \bmod p.$$

If the above check fails for party $\mathcal{P}_i$, then the other parties perform the *re-construction* phase, i.e. $\mathcal{P}_i$ rebroadcasts the values x_{ij}, A_{ik}, and the other parties verify them using the information $\tilde{x}_{ij}$ and w_{ik} (see [56, 57] for details). Each $\mathcal{P}_i$ from $\mathcal{Q}$ can compute $y = \prod_{i \in \mathcal{Q}} A_{i0} \bmod p$. If we define the polynomial $f(z) = \sum_{i \in \mathcal{Q}} f_i(z) \in \mathbb{Z}_q[z]$, it is easy to see that $x_i = f(i)$ $(\in \mathbb{Z}_q)$ for $i \in \mathcal{Q}$, and thus $x = f(0) = \sum_{i \in \mathcal{Q}} a_{i0}$ $(\in \mathbb{Z}_q)$ is the master secret of the system.

2.7 Conclusions

We have introduced the cryptographic primitives and infrastructures used in this thesis. We discuss some of them in greater detail in subsequent chapters.

Part I

Pervasive Computing: Personal Area Networks

Chapter 3

Securing A Personal Area Network

Contents

We review previous research security mechanisms designed for use in PANs, focusing primarily on PAN security initialisation. After defining the notion of a PAN and giving a PAN security architecture, we present two existing approaches to PAN security initialisation.

3.1 Introduction

A Personal Area Network (PAN) is a small wireless network that covers only a personal work space, e.g. an office or a meeting room. A PAN only includes those components owned and controlled by a single user, and the components directly communicating with each other via a local interface such as Bluetooth or IrDA (Infrared Data Association). Possible deployment scenarios for PANs include smart offices, smart homes, conference halls, hospitals, public areas, etc. Due to the unique characteristics of a PAN, we could take a different approach to that employed in other wireless networks, such as using human interface or trust relations within a PAN, in order to initialise the security contexts of the personal wireless devices.

In this chapter we give a formal definition of a PAN, and introduce a PAN security framework. We then discuss two different approaches to PAN security initialisation. The first involves establishing a shared secret between pairs of PAN devices with the active support of users. Since the PAN components are close to each other and there is, in most cases, at least one human that controls the components, the necessary security associations can be created with human assistance. The second approach uses public key cryptography and involves defining a special device called a Personal CA as part of a Personal PKI, which maintains the security context within a PAN.

3.2 Personal Area Networks

In this section we give a definition for a PAN, and describe a security framework for a PAN.

3.2.1 A Personal Area Network

We first define what we mean by a PAN [53, 133].

Definition 3.1 (PAN) *A PAN is a collection of fixed, portable, or moving components within or entering a Personal Area, which form a Network through local interfaces. A Personal Area is the space within a sphere around a person (stationary or in motion), typically having a radius of about 10 metres.*

We consider here PANs involving only wireless communications, following the IEEE standard [133]; however, other authors (see, for example, Gehrmann and Nyberg [53]) consider PANs using both wired and wireless communications. Wireless connections are expected to be the commonly used means of communications within a PAN, and, in fact, many PAN devices do not have a wired interface.

We also use the following PAN-specific terminology [53].

- Component (or device): A PAN consists of a collection of components. Each component is an independent computing unit, and has processing capabilities as well as digital memory. Possible PAN components include computers,

personal digital assistants (PDAs), printers, microphones, headsets, sensors, mobile phones, smartcards, etc.

- Service: A service is a communication or computing service offered by a component either *locally*, i.e. through a user interface, or remotely to other components. Each component maintains a list of services it offers, as well as policies for access and discovery/advertisement.
- Application: An application is a process running on a component.
- User: A user physically controls and operates a PAN component, complying with the policies configured in the component. The user of a component might change, but, at any given time, each component has a single user.
- Owner: Each component has a single owner. By specifying an appropriate policy, the component owner may allow users other than the owner to use the component.
- Local interface: Each component has at least one local communication interface which is suitable for direct connection to other PAN components. A direct connection between every pair of components is not required, but each component should be able to connect to at least one other PAN component.
- Global network interface: A component may have a global network interface such as a connection to an IP-backbone or a mobile network.
- Security policy: Each component has a security policy. There are two main types of security policy: local security policy and remote security policy. A local security policy specifies which resources of a component a user may access, and also determines whether or not authorisation for users is required. A remote security policy determines how to access a component, i.e. it specifies the relation between the service the component offers and the names of the PAN components that utilise the service.

In subsequent discussions we often assume that the owner and user of a PAN component are the same, but this is not necessarily always the case.

3.2.2 Security framework of Personal Area Networks

We next consider how the security context of personal wireless devices in a PAN can be initialised. We describe a PAN security architecture, following Gehrmann and Nyberg [53]. Since the administrator of a PAN may be a non-expert user, the security initialisation process should be simple and user-friendly. That is, the number and complexity of interactions should be minimised. We use the Dolev-Yao threat model [39] to characterise potential adversaries. We assume that the adversary controls the communication channels; thus an attacker can intercept and synthesise any message, and is only limited by the constraints of the cryptographic methods used. We define a security initialisation procedure in a PAN to involve the following three steps.

- Step 1: Establish an initial secure channel.
- Step 2: Create a security association, i.e. configure cryptographic parameters for autonomous secure PAN connection establishment.
- Step 3: Configure (default) security policies.

We next discuss each of these three steps in greater detail.

Initial secure channel establishment

In order to create a security association between two PAN components, an initial secure channel must first be established. Using this initial secure channel, one PAN component can be sure that it is exchanging information with the intended components and that the communication is not being intercepted or modified by any third party. The provision of an initial secure channel is thus particularly important. If tampering or eavesdropping is possible during this process, then the security of all subsequent communications could be compromised. Since wireless interfaces are potentially insecure, an alternative secure communications channel is required. A direct wired connection between two components, e.g. using USB or Ethernet, could provide sufficient protection, but many PAN devices do not have a wired interface.

However, in a PAN the components are close to each other and, in most cases, there

is at least one human that controls the components. We can thus use the human as a *secure channel*. That is, a human can be asked to read, check, and/or enter values into the PAN components.

Another way of establishing a secure channel arises if a component has an optical reader, e.g. a barcode reader, that could be used to input information to the component from a printed slip. In such a case, this reader could be used by a human administrator to load a public key, a hash of a public key, or a public key certificate into a component.

Security association creation

The security functions in a PAN, such as those necessary for communication security or access control, require cryptographic parameters, typically cryptographic keys, to be established in the PAN components. These parameters include secret keys for secret key cryptosystems and/or public/private key pairs for public key cryptosystems. Such shared secrets and/or trusted public keys form part of what is known as a security association. With a security association in place, authentication of other PAN devices becomes possible. Furthermore, the integrity and/or confidentiality of the information exchanged over an insecure channel between the components can be protected.

Security policy configuration

The configuration of security policy must be based on trust relationships between PAN components. It could cover the case where a new PAN device joins the network, or where devices from other PANs try to use a service. However, we do not discuss this topic further here, since the main focus of this chapter is the establishment of cryptographic keys in a PAN. A detailed discussion of PAN trust models and the configuration of security policies is given by Gehrmann and Nyberg [53].

3.3 Existing Security Schemes

In recent years, a wide variety of work has been performed on developing techniques for security initialisation in decentralised networks. We describe two types of widely studied security schemes, i.e. key establishment in peer-to-peer networks over a radio link [11, 26, 41, 47, 52, 53, 92, 98, 131, 132] and key management based on public key techniques [54, 104].

The **first class of solutions** involves establishing a shared secret between two PAN devices with the active involvement of users. Since the PAN components are close to each other and there is, in most cases, at least one human that controls the components, the necessary security associations can be created with human assistance.

For example, if the components possess *human interfaces* such as a key pad and/or a display, a human operator could be asked to copy data from one device to the other, compare the outputs of the two devices, or enter the same data into both devices. A variety of such solutions exist, with varying requirements on the display and input capabilities of the devices [26, 41, 52, 53, 92, 98].

Location limited channels can also be used to securely exchange secret information [11, 47, 132]. That is, security parameters can be exchanged using relatively secure channels, such as Infrared or NFC (Near Field Communication), prior to or during performing key establishment protocols using insecure channels. Since such channels have short communication ranges (typically less than 10cm), users need to locate two devices in close proximity to create a security association.

The **second class of solutions**, using public key cryptography, involve defining a special device called a Personal CA forming part of a Personal PKI [54, 104], responsible for maintaining a security context within a PAN. A PAN user managing its own local network environment will gain few benefits from employing a centralised CA, and the user may not want, for privacy reasons, to delegate the CA operation to a party outside its personal environment. Thus, a more localised PKI can provide a conventional means of creating security associations in a PAN.

3.3.1 DH key agreement via wireless channels

The Diffie-Hellman (DH) key agreement protocol [37] provides an efficient way of creating a security association between two PAN devices. We start by briefly reviewing this protocol. Two parties first generate common parameters using a group-generating algorithm $\mathcal{G}$, as given in definition 2.3. The first party generates $x \in_R \mathbb{Z}_q$ and computes g^x ($\in \mathbb{G}$). The second party generates $y \in_R \mathbb{Z}_q$ and computes g^y ($\in \mathbb{G}$). The two parties exchange the public parameters g^x and g^y, and calculate the shared secret key as $g^{xy} = (g^x)^y = (g^y)^x$. Assuming that the DH problem is hard relative to $\mathcal{G}$, the DH key agreement protocol is known to be secure against a passive adversary [102].

However, the DH key agreement protocol is vulnerable to an active attack, known as a man-in-the-middle attack. That is, when two devices attempt to pair with each other, an adversary device connects to the two devices and relays information between them, giving the illusion that they are directly connected. In the above protocol, an adversary generates $z \in_R \mathbb{Z}_q$, and exchanges g^z with the two devices; as a result of it establishes separate secret keys, g^{xz} and g^{yz}, with these two devices. The adversary can then eavesdrop on communications between the two devices, and is also able to insert or modify information on the connection.

Security using human interface

A variety of approaches have been proposed for verifying the integrity of Diffie-Hellman (DH) public parameters exchanged between two devices using a *human interface* [26, 41, 52, 53, 92, 98].

Maher and Windham [98] present several such methods, one of which involves comparing the truncated (four hexadecimal digit) hash values of DH public parameters in each device. Inspired by the work of Maher and Windham [98], Larsson [92] proposed the use of a temporary secret shared between the two users. Dohrmann and Ellison [41] suggest converting hash values to readable words or graphical representations. Comparing such words, however, could take as much as 24 seconds, and potentially limited PAN devices may not have the sophisticated displays necessary to support such graphical representation.

Gehrmann, Nyberg, and Mitchell [52, 53] have proposed a series of schemes which prevent man-in-the-middle attacks requiring the users to type in or compare relatively short strings of digits. These schemes, called the MANA protocols, are designed to enable wireless devices to authenticate one another via an insecure wireless channel with the aid of a manual transfer of data between the devices. The manual transfer refers to one of the following procedures: copying data output from one device into the other device; comparing the output of two devices; or, entering the same data into both devices. The MANA protocols have been standardised [46].

Security using location limited channels

The concept of *location limited channels* has also been proposed to protect against man-in-the-middle attack on the DH key agreement protocol [11, 47, 132]. Stajano and Anderson [132] discuss the use of a physical contact as a location limited channel for transmitting keying materials in plaintext. Balfanz et al. [11] suggest using a location limited channel, e.g. an infrared link, to exchange pre-authentication data, and, once such data has been transferred, users switch to a common radio channel and run a secure key exchange protocol.

Example scheme: Bluetooth SSP

The Bluetooth Alliance Core Specification V2.1 + EDR [130, 131] specifies a secure pairing procedure called Secure Simple Pairing (SSP), which provides four models for verifying the integrity of DH public parameters exchanged between two devices. The Numeric Comparison model makes use of a *human interface*, and the Out of Band model uses NFC as a *location-limited channel*. We next discuss these two models in greater detail.

The Numeric Comparison model is designed for scenarios, in which both devices have a six-digit numeric display and an input method that can be used to indicate "yes" or "no". A six-digit number is displayed on the displays of both devices, and the user is asked whether the numbers are the same. If the user enters "yes" on both devices, the pairing is successful. This process serves two purposes. First, it confirms to the user that the correct devices are connected with each other; we note that many

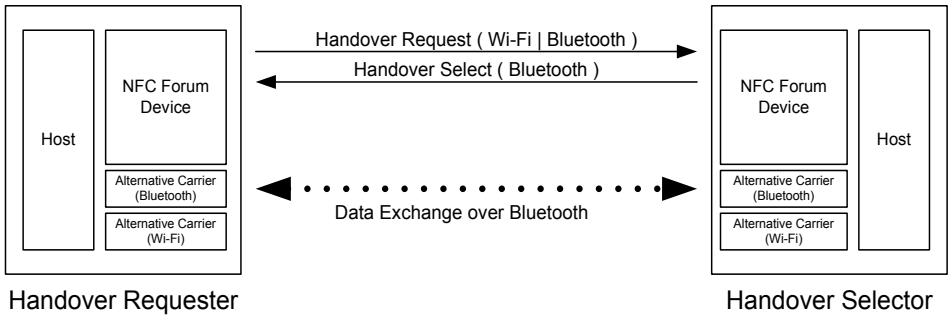


Figure 3.1: Example of Connection Handover [47]

devices do not have unique names. Second, it provides protection against man-in-the-middle attacks. This model looks similar to the PIN entry model adopted in the Bluetooth Core Specification V2.0 + EDR and in earlier versions. However, the six-digit number in SSP is not used as an input to the security algorithm; instead it forms a part of output of the security algorithm.

The Out of Band (OOB) association model is primarily designed for scenarios in which an OOB channel is used to both discover the correct device and to securely[1] exchange (or securely transfer) cryptographic values. A good example of OOB channel is provided by NFC, where a user(s) places a pair of devices so that they touch each other; the devices then exchange cryptographic information as well as discovery information such as Bluetooth addresses. One of the devices uses the received Bluetooth address to establish a connection with the other device, and the cryptographic information is used during a subsequent authentication process. Once the devices have completed their exchange, the user is asked if the device should pair with the other device; if the user enters "yes", the pairing procedure is successful.

The above type is standardised in the Connection Handover Specification [47]. SSP is used to verify the integrity of previously exchanged DH public parameters, and the Connection Handover exchanges necessary secret information via an NFC channel prior to a DH exchange using a Bluetooth or Wi-Fi channel. That is, the information included in the `Handover Request/Response` is used to prevent a man-in-the-middle attack on subsequent DH exchanges (see Figure 3.1). More specifically, assume that two devices are equipped with NFC, Bluetooth, and Wi-Fi communication capabilities. The 'Handover Requester' device first sends a `Handover Request` message,

[1]The OOB channels are assumed to be secure against passive and active attacks.

which includes information about the device's Wi-Fi and Bluetooth capabilities. The 'Handover Selector' device then replies with a `Handover Select` message, which indicates that it will use Bluetooth or Wi-Fi in subsequent communications. The `Handover Request/Select` messages include cryptographic secrets as well as the required configuration parameters.

3.3.2 Personal PKIs

In a conventional public key infrastructure (PKI), a certification authority (CA) issues a public key certificate. Prior to issuing such a certificate CA is responsible for checking that

- the user to which certificate is issued has the identity that is specified in the certificate; and
- the public key to be included in a certificate corresponds to a private key that the specified certificate subject possesses.

Operation and use of a large scale CA, however, has significant implementation issues.

- It is costly to implement and maintain a secure certification process for large numbers of users. Maintenance can be particularly problematic, especially if it is necessary to generate and distribute frequently updated certificate revocation lists (CRLs) containing the serial numbers of a large number of revoked certificates.
- A user managing its own local network environment, such as a PAN, will gain few benefits from employing a centralised CA, and the user may not want, for privacy reasons, to delegate the CA operation to a party outside its personal environment [54].

Nevertheless, a more localised PKI can provide an effective means of creating security associations in a PAN. Gehrmann et al. [54] and Mitchell et al. [104] adapt the notion of a PKI to a local PAN environment, introducing a concept called the personal PKI,

and give functional and security requirements for such a PKI [104]. In a personal PKI, one of the devices in a PAN acts as a personal CA that issues certificates for all PAN components. As with any other PKI, all PAN components share the public key of the personal CA, and use certificates issued by the personal CA as a basis for secure session key establishment and authentication between PAN components.

Security initialisation using a personal CA

We now briefly describe the operational processes of a personal PKI [54]. First, the personal CA must be initialised, which involves generating a signature key pair. Other PAN components can be initialised using the following procedure [104].

1. A PAN device generates any necessary key pairs. It also imports authentication material from its owner, which may require a modest number of keystrokes by the user.
2. The PAN device is informed of which other device is the personal CA, or discovers this device by communicating across the PAN.
3. The root public key of the personal CA is passed to the PAN device. This must be done in such a way that the PAN device can verify the integrity and origin of the CA public key.
4. The PAN device provides its public key to the personal CA. This must be done in such a way that the personal CA can verify the integrity and origin of the public key.
5. The personal CA generates a public key certificate for the mobile device.
6. The newly created public key certificate is passed to the PAN device.

As described above, during initialisation we need a means of verifying the integrity and origin of the public keys exchanged between the CA and the PAN component. That is, we require a method for two devices to exchange public keys in an authenticated manner. The Bluetooth SSP, described in Section 3.3.1, could, for example, be used for this purpose.

Public key status management

Once the PAN devices have been initialised, there is a need for ongoing management of key pairs and certificates. Three main management functions need to be supported in a PAN [54, 104].

- Certificate and key pair update, i.e. methods to be used when a PAN device wishes to use a new key pair or when the certificate for a public key has expired.
- Key status management, i.e. disseminating information regarding revoked public keys across the PAN.
- Trust management, i.e. managing a root public key update or managing the possible replacement of the personal CA.

We do not discuss key and certificate management issues for a personal PKI further here; more detailed discussions appear in Gehrmann et al. [54, 104].

3.4 Conclusions

We have investigated security initialisation schemes for PANs, focusing primarily on the security initialisation process. Personal wireless devices, however, are more likely to communicate with devices outside the PAN, because of the ubiquity of such devices and the convergence of communication technology. Inter-PAN secure communications poses significant additional requirements on the key management procedures. In the next chapter we propose a novel security initialisation scheme for use both within and between PANs.

Chapter 4

Securing Inter-PANs Communications

Contents

In this chapter we propose a novel security initialisation scheme for use both within and between PANs, and show that the proposed scheme achieves desirable security and efficiency properties making use of the unique characteristics of PANs.

4.1 Introduction

Although a PAN may act as a stand-alone network, PANs are likely to be interconnected to share information or services. We call such interconnected PANs iPANs. Network nodes exchange routing information in order to establish routes between nodes. Such information could be a target for an adversary who may inject erroneous

routing information in order to disrupt communications in iPANs. Cryptographic schemes could be used to protect routing information as well as data traffics, but they require the provision of cryptographic keys, i.e. security initialisation or key pre-distribution process is required.

In this section we first describe the interconnected Personal Area Networks, iPANs. We then investigate the candidates for key management schemes, which are potentially suitable for supporting security in iPANs. After discussing related work, we discuss the advantages of the proposed scheme compared to the conventional approaches.

4.1.1 Interconnected Personal Area Networks

We next describe how PAN devices communicate, and also how communications in iPANs is likely to become possible. Bluetooth technologies are potentially a key enabler of iPANs, although iPANs could also use a variety of other access technologies, e.g. wireless LAN (WLAN) interfaces within a PAN, and the IP-backbone (Internet) or a GPRS/UMTS mobile network between PANs.

Two or more piconets can be connected, forming a scatternet (Figure 4.1), through a device which is a member of both piconet. A device may be a slave in multiple piconets, but can be a master in only one piconet. A scatternet can then support inter-PAN communications. A PAN user may also wish to access a device in another PAN, where this device is not accessible via a direct connection (e.g. via Bluetooth).

Since a PAN can be IP-based, a PAN user can connect to other PANs via the IP-backbone network using a LAN access point or GPRS/UMTS mobile network. Figure 4.2 shows two scenarios of this type, where PANs are connected to an IP-backbone. The first figure of Figure 4.2 shows the interconnected PANs, where one PAN serves as a LAN access point for the other two PANs. In the other figure in Figure 4.2, one PAN serves as an access point to the IP-backbone via a mobile network.

We now discuss features which the two classes of network have in common.

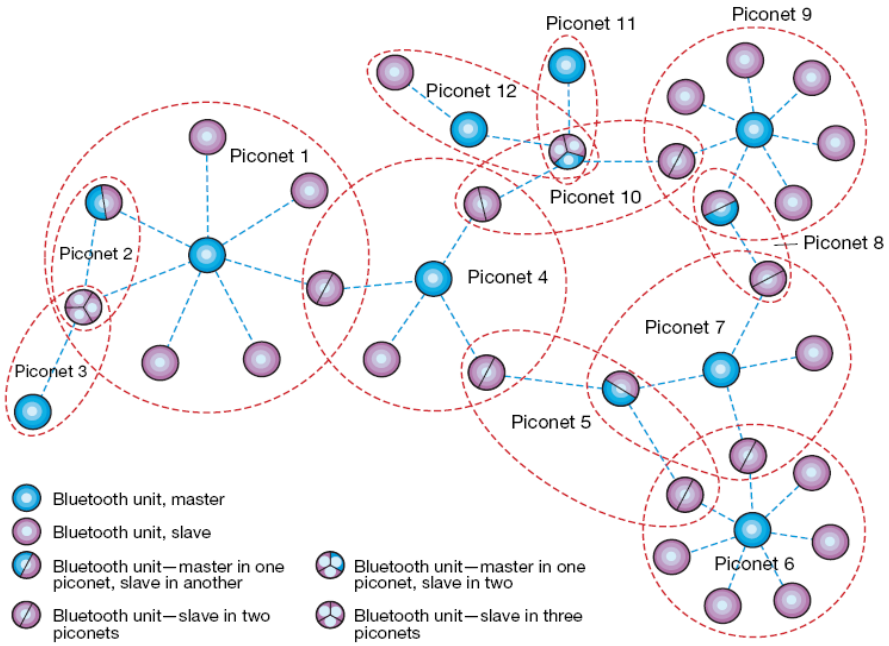


Figure 4.1: A Bluetooth scatternet [49]

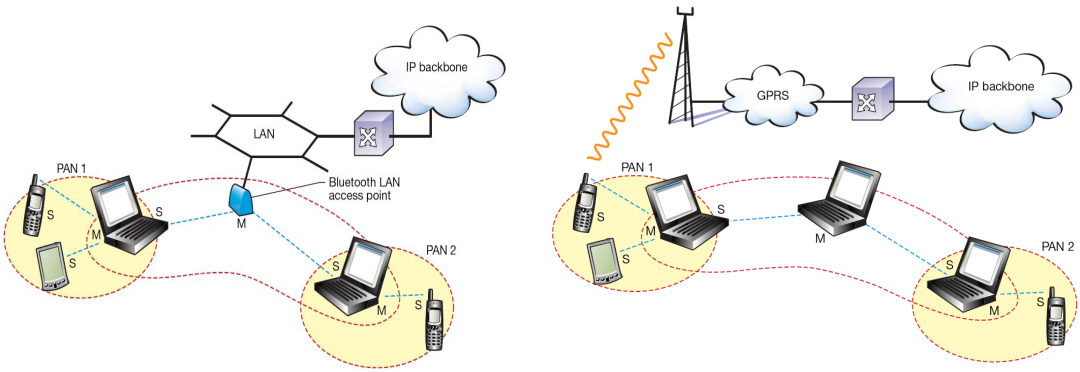


Figure 4.2: Three interconnected PANs [49]

- Shared physical medium: The wireless communication medium is accessible by any party with adequate equipment. Adversary is thus able to eavesdrop on communications and/or modify the transmitted messages.

- Poor physical security: Since network devices are not held in a secure location, such devices could be easily compromised, e.g. by being stolen or lost. Thus the insider adversary, who compromises one or more network devices and then performs attacks using the compromised devices, must be addressed.

- Limited resources: Wireless devices typically have limited computational, memory, and energy resources, for reasons of cost and portability. Limited computational capabilities may mean that a device is unable to perform public key crypto-algorithms. Limited battery power may restrict the energy available for communications, restricting to the available bandwidth and transmission range. Security protocols should thus be optimised to minimise use of network resources, and should also be able to cope with network node failures arising from battery exhaustion.

- Network topology: Network nodes are potentially mobile and wireless communication could be error-prone. These network features result in a dynamic and weakly connected topology. Security protocols should be designed so that security services remain available even in the presence of dynamic changes in network topology.

- Self-organisation: Network cannot rely on any form of central administrator, e.g. an on-line or off-line TTP. This is because such a cental node may not be accessible by all network nodes, and it could be a target of attack as a single point of failure [145]. This means that end users are obliged to participate in setting up security associations.

Apart from the above, iPANs may also have the following characteristics which differ from those of ad hoc networks.

- Network infrastructure: Ad hoc networks are designed to work in the absence of a fixed infrastructure, such as an IP-backbone or a mobile network, and all network nodes must be able to function autonomously. For example, because of the limited communications range of a single node, data transmission is

achieved in a multi-hop fashion, and each node serves as both a host and a router. This differs from the situation in iPANs, where we assume that external communications infrastructures, e.g. an IP-backbone or a mobile network, are likely to be available. Nevertheless, ad hoc networks which are integrated with conventional fixed networks, such as hybrid ad hoc networks [122] or wireless mesh networks [1, 22], have been considered. The network infrastructure of such networks can be regarded as similar to that of iPANs.

- Trust relationships: Unlike in ad hoc networks, all the devices within a PAN are used and controlled by a single user. Using this fact may enable both communication and computation efficiency to be significantly increased.
- Hierarchical structure: Ad hoc network protocols often distribute the responsibility of providing network functionality equally across the set of nodes [103, 145]. In reality, however, networks consist of devices with different computational and communications capabilities. It is thus natural to assign devices varying roles in providing security services.

As we discussed above, the iPANs share a variety of common features with ad hoc networks. Thus, key management schemes proposed for use in ad hoc networks could also be used in iPANs. However, there exist also differences between ad hoc networks and iPANs, which could make such an approach problematic.

4.1.2 Key management in iPANs

Symmetric cryptography is relatively straightforward to implement in resource-constrained devices, and necessary shared secrets can be established between any two devices in a variety of ways, as described in section 3.3. The use of public key cryptography (PKC), however, is more appropriate for an open environment, such as iPANs, where parties who have never previously interacted wish to communicate securely as described in Section 2.6.2. The use of a public key infrastructure (PKI) (see Section 2.6.3) enables end users to obtain verified public keys, but the identity-based cryptography (IBC) [18, 31, 127] has potential advantages over a conventional PKI.

The concept of identity-based cryptography (IBC) was introduced by Shamir [127]

in 1984.[1] Instead of using a random public/private key pair, Shamir proposed using a user's identity, e.g. an email address or IP-address, as that user's public key (hence avoiding the need of public key certificate). Such public keys are thus self-authenticating, and certificates are not required. Furthermore, since identities such as IP and/or MAC addresses can be propagated in transmitted messages. There is thus no need to generate and distribute public keys across the network.

One fundamental requirement for an IBC is that it should be infeasible to compute the private key from the public identity without access to certain global trapdoor information. For this reason, IBC requires an unconditionally trusted TTP, called a trusted authority (TA), which computes a user's private key using a private key generator (PKG) with the trapdoor information as input. As a means of reducing the reliance on a single PKG, Boneh and Franklin [18] introduce the concept of a distributed PKG (DPKG), where private key generation involves multiple PKGs.

The use of a DPKG is desirable in iPANs for the following reasons. A single PKG could be a point of vulnerability in the network, since iPANs nodes are likely be vulnerable to various (physical) attacks. Some network nodes may not be able to communicate with a single PKG because of the limited communications capacity of personal wireless devices. Furthermore, PAN users are unlikely to delegate security operations to an external trusted party for privacy reasons. We thus propose that the participating PAN users could share key management functionality among themselves. More specifically, a key management service for iPANs can be realised by distributing trust to a set of nodes from each PAN.

4.1.3 Related work

Zhou and Haas [145] propose key management scheme in ad hoc networks using certificates. They propose that a set of network nodes form a distributed CA and jointly perform key management service. The concept of a personal PKI was introduced in section 3.3.2; this involves one of the PAN devices acting as a personal CA, which issues and distributes certificates for the other PAN devices. In the setting of

[1] Although Shamir [127] showed that an identity-based signature (IBS) scheme can be constructed using the RSA [117] function, he was not able to construct an identity-based encryption (IBE) scheme. Much more recently, in 2001, Boneh and Franklin [18] introduced the first practical and secure IBE scheme, based on Weil parings.

iPANs, Personal CAs, one from each PAN, can form a distributed CA, and jointly sign certificates for devices, using the technique described by Zhou and Haas [145]. The certificates can be transferred from each Personal CA to the PAN devices within the PAN, using previously established secure channels.

Khalili, Katz, and Arbaugh [87] propose self-organised key pre-distribution schemes using identity-based cryptography, where all participating network nodes form a distributed PKG. They assume that there is no prior shared keying material or trust/security association between PKGs, and describe how to establish these at the time of network formation using the verifiable secret sharing schemes which does not involve a trusted third party. However, such a secret sharing scheme known to the author require secure channels between all network nodes for the distribution of master secret shares. We also note that they do not address the issue of protecting the confidentiality and integrity of the private key shares when transferred from the PKGs to the mobile devices. Providing such a secure channel is a non-trivial problem.

To address this issue, Deng and Agrawal [34] propose that each PKG encrypts the share using a requesting mobile device's temporary public key. Since this temporary public key is not certified, the adversary can spoof the public key of the requesting device, and then recover the distributed private key by combining decrypted private key shares. Both the Deng and Agrawal [34] and the Khalili et al. [87] schemes require the existence of secure channels between every pair of devices for secure key pre-distribution, which limits their practicality.

Liu et al. [144] separate non-PKG devices from PKG devices, which together form a distributed PKG. This scheme, however, requires substantial number of initial secure channels and only devices with public key functionality can use the key management service.

4.1.4 Contribution

We propose a security initialisation scheme for use both within and between PANs, making use of the unique characteristics of iPANs. The proposed scheme for iPANs has the following distinctive advantages over existing schemes.

- The proposed scheme greatly reduces the computation and communication burdens on resource-constrained devices. Unlike identity-based key management schemes previously proposed for use in ad hoc networks [34, 87, 144], which only support devices which can perform public key crypto-algorithms, the proposed scheme allows resource-constrained devices which only implement secret key crypto-algorithms to make use of the key management service.
- Previously proposed a distributed key management schemes for ad hoc networks [34, 87, 144, 145] assume *a priori* secure channels between all (or a substantial number of) participating the nodes, which reduces their applicability. The proposed scheme only requires the existence of a small number of such channels, relative to the network size.[2]

In the subsequent sections we describe a security initialisation scheme designed for iPANs.

4.2 Security Initialisation for iPANs

In this section we give a formal description on the proposed security initialisation scheme for iPANs, and provide security and efficiency analysis.

4.2.1 System models and design goals

The network scenario involves a limited number, say n, of users with multiple wireless personal devices who wish to form iPANs to share important information. In this section we define the system model and the capabilities of a potential adversary, and specify our design goals.

System definition

We assume that the system, i.e. iPANs, consists of n Personal Area Networks (PANs), interconnected using scatternets, the IP-backbone, and/or mobile networks,

[2] See Table 4.1 for the details.

as discussed in section 4.1.1. Each device has a unique identity, e.g. as provided by an IP-address and/or a MAC-address.

One[3] device from each PAN acts as a PKG for the devices within its PAN. It is reasonable to assume that each device in a PAN trust its PKG, since all devices within a PAN are under the control of a single user. The following minimum requirements must be satisfied for such a PKG.

- It should be computationally more powerful than other devices in the PAN. It must be capable of performing public key cryptography, while other PAN devices are assumed to be capable of performing secret key cryptography.
- It should have secure communication channels to all devices in its PAN. Such secure channels can be established using several schemes described in section 3.3.
- It should have a communication channel to the PKGs of at least $\lceil n/2 \rceil$ of the other PANs.

Such a set of n PKGs form a distributed private key generator (DPKG).

Adversary

Since the objective is devising a secure key management scheme, we only consider attacks against the key management scheme itself. We consider two kinds of attacks; compromise attacks and disruption attacks. If an adversary compromises a device, it has complete control over it, including learning its secret information and being able to change its intended behaviour. An adversary could also disrupt the system by making the compromised device affect the intended communication process.

We assume that an adversary cannot compromise an unlimited number of nodes, so that honest devices are always in the majority. Specifically, we assume that an adversary can corrupt up to t of the n PKG devices for any value of $t < n/2$. We assume that the adversarial computational power is a probabilistic polynomial time. Our adversary is static, i.e. it chooses the corrupted network nodes at the beginning of the protocol.

[3] More than one device may act as PKGs within a PAN, as discussed in section 4.3.3.

Design goals

Given the above system model and adversary types, we propose a security initialisation scheme, with the following security and efficiency requirements.

- Secrecy: The system master secret should be not disclosed to any party, and any information regarding the private key of a PAN device should not be learned by any party except for the PKG device within the same PAN.
- Correctness: Each PAN device should be equipped with the correct private key corresponding to its identifier, given the system parameter and master secret.
- Efficiency: The scheme should be efficient in terms of storage, computation, and communication, since PAN devices are often resource-constrained. In many cases, a wireless personal devices will be battery-operated, and devices such as RFID tags will even have no battery.
- Minimal initial secure channels: A priori initial secure channels are often assumed in key management schemes for ad hoc networks. Provision of such channels, however, is never simple in an ad hoc network, and, in some cases, a manual process is the only option. For practicality, the number of initial secure channels relative to the number of network nodes should be minimised.

In the subsequent sections we propose a security initailisation scheme for iPANs, which has been designed to meet the above requirements.

4.2.2 Proposed scheme

The security intialisation process involves distributing a master secret amongst n PKGs, and provides each network node with a public/private key pair. An adversary is assumed to be able to compromise at most t PKGs during network setup (i.e. during the security initialisation process). It is important to note that all secret sharing schemes require some kind of *a priori* security association, i.e. an initial secure channel, between the participating parties, and the proposed scheme requires $n(n-1)/2$ initial secure channels (between all the n PKGs).

System parameters

The group $\mathbb{G}_1$ of order q has a generator P, and the group $\mathbb{G}_2$ is of order q. The bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is the modified Weil pairing [18]. We also define a hash function $h : \{0,1\}^* \to \mathbb{G}_1^*$, as defined by Boneh and Franklin [18]. Since it is difficult to construct hash functions mapping directly from $\{0,1\}^*$ to $\mathbb{G}_1^*$, Boneh and Franklin [18] show that it suffices to have a hash function $h : \{0,1\}^* \to B$ ($\subset \{0,1\}^*$) and a deterministic encoding function to map from B to $\mathbb{G}_1^*$.

The parameters t and n need to be chosen to achieve an appropriate tradeoff between security and robustness. In particular, for a fixed n, a larger t means that adversaries need to compromise more PKGs (i.e. the system is more secure), but they only need to disrupt fewer PKGs (i.e. the system is less robust). It has been suggested to set $t = \lfloor \frac{n}{2} \rfloor$ [36, 128].

Distributing master secret

A set of n PKGs, $\mathcal{P}_i$ ($1 \leq i \leq n$), jointly generate a master secret and distribute the shares amongst the set using the Gennaro et al.'s scheme [56] (described in Section 2.6.4).[4]

1. Each $\mathcal{P}_i$ ($1 \leq i \leq n$) chooses two polynomials $f_i(z), \tilde{f}_i(z) \in \mathbb{Z}_q[z]$ of degree t:

$$f_i(z) = a_{i0} + a_{i1}z + \cdots + a_{it}z^t \quad \text{and} \quad \tilde{f}_i(z) = \tilde{a}_{i0} + \tilde{a}_{i1}z + \cdots + \tilde{a}_{it}z^t$$

   where $a_{ik}, \tilde{a}_{ik} \in_{\mathsf{R}} \mathbb{Z}_q$ ($0 \leq k \leq t$). Each $\mathcal{P}_i$ broadcasts $W_{ik} = a_{ik}P + \tilde{a}_{ik}\tilde{P}$ ($\in \mathbb{G}_1$) for $k = 0, 1, \ldots, t$, where $\tilde{P}$ ($= lP \in \mathbb{G}_1$ for some $l \in \mathbb{Z}_q^*$) and l are not known to any $\mathcal{P}_i$.

2. Each $\mathcal{P}_i$ computes the pair $\big(s_{ij} = f_i(j) \bmod q,\ \tilde{s}_{ij} = \tilde{f}_i(j) \bmod q\big)$ for $j = 1, \ldots, n$, and sends $(s_{ij}, \tilde{s}_{ij})$ to $\mathcal{P}_j$ via a secure channel, reserving $(s_{ii}, \tilde{s}_{ii})$ for itself.

[4] We use Gennaro et al's scheme to generate a master secret for the distributed private key generator (DPKG) as suggested by Boneh and Franklin [18]. When Gennaro et al's scheme is fully adopted, i.e. the public output is also generated (see Section 2.6.4), such a key pair can be used for the threshold cryptosystems for PKG devices [56].

3. On receiving $(s_{ji}, \tilde{s}_{ji})$, each $\mathcal{P}_i$ verifies it for $j = 1, 2, \ldots, n$ by checking that:

$$s_{ji}P + \tilde{s}_{ji}\tilde{P} = \sum_{k=0}^{t} i^k W_{ik} \ (\in \mathbb{G}_1). \tag{4.1}$$

   If this check fails for an index j, $\mathcal{P}_i$ broadcasts a *complaint* against $\mathcal{P}_j$.

4. If any $\mathcal{P}_i$ receives a complaint from $\mathcal{P}_j$, it broadcasts the pair $(s_{ij}, \tilde{s}_{ij})$.

5. A PKG $\mathcal{P}_i$ becomes *disqualified* if:

   - there are complaints against $\mathcal{P}_i$ from more than t parties; or
   - the pair broadcast by $\mathcal{P}_i$ in step 4 does not satisfy equation (4.1).

   The non-disqualified PKGs then form a set $\mathcal{Q}$.

6. Each $\mathcal{P}_i$ computes its share as the following pair:

$$\big(s_i = \sum_{j \in \mathcal{Q}} s_{ji} \bmod q, \ \tilde{s}_i = \sum_{j \in \mathcal{Q}} \tilde{s}_{ji} \bmod q\big).$$

The master secret value s satisfies $s = \sum_{i \in \mathcal{Q}} a_{i0} \bmod q$, and is not available to any party. If we define the polynomial $f(z) = \sum_{i \in \mathcal{Q}} f_i(z) \in \mathbb{Z}_q[z]$, it is easy to see that $s_i = f(i)$ $(\in \mathbb{Z}_q)$ for $i \in \mathcal{Q}$, and thus $s = f(0) = \sum_{i \in \mathcal{Q}} a_{i0}$ $(\in \mathbb{Z}_q)$ is the master secret of the system. Any disqualified PKG, i.e. any PKG $\mathcal{P}_j$ such that $j \notin \mathcal{Q}$, is excluded from use of the key management service along with the PAN that includes $\mathcal{P}_j$.

Private key extraction and secure transfer

Once the master secret is shared between the n PKGs, they can jointly generate the private keys for iPANs network nodes.

Any set of at least $t+1$ PKGs can jointly perform private key extraction. We denote by $\mathcal{D}_{\mathsf{id}}$ the PAN device with identifier $\mathsf{id} \in \{0,1\}^*$. A PKG in the same PAN as $\mathcal{D}_{\mathsf{id}}$, denoted by $\mathcal{P}$, can extract the private key for $\mathcal{D}_{\mathsf{id}}$ using the following procedure.

1. $\mathcal{D}_{\mathsf{id}}$ sends $\mathcal{P}$ its identifier id along with a private key extraction request, together with any information required by the key issuance policy.

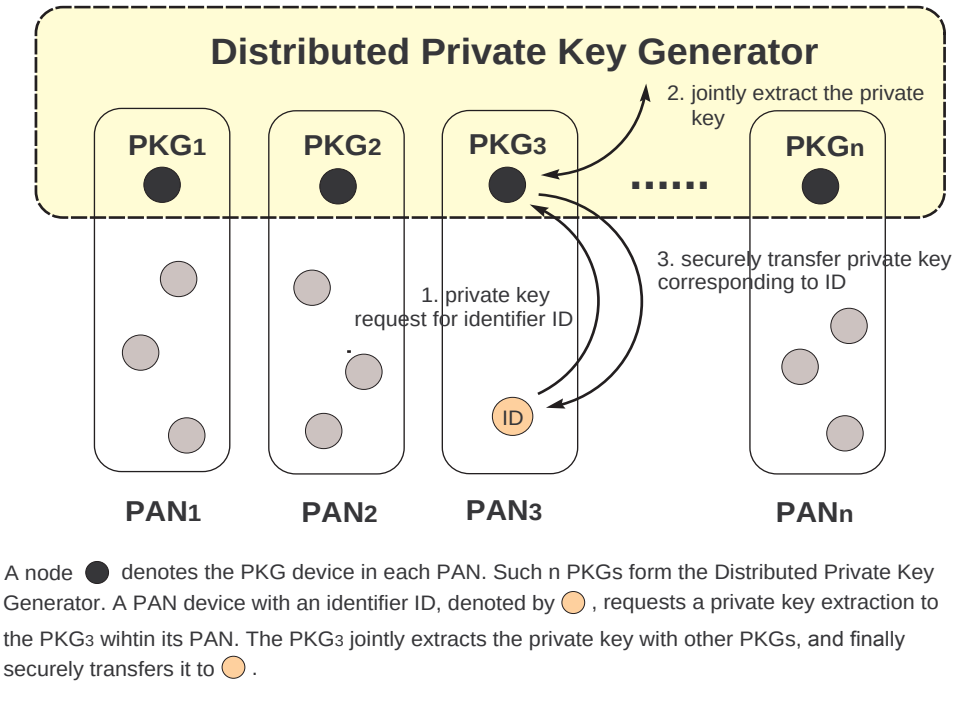


Figure 4.3: A distributed private key generator (DPKG)

2. $\mathcal{P}$ relays the request information to t (or more[5]) other PKGs: $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_t$, say.

3. Each $\mathcal{P}_i$ $(1 \leq i \leq t)$ replies with a private key share $d_{\mathsf{id}}^{(i)} = s_i Q_{\mathsf{id}}$ $(\in \mathbb{G}_1)$ via a secure channel, where $Q_{\mathsf{id}} = h(\mathsf{ID})$ and $\mathsf{ID} = \mathsf{id} \,||\, \texttt{expiry-date}$. Assuming that s_0 is $\mathcal{P}$'s share of the master secret, $\mathcal{P}$ also computes the share $d_{\mathsf{id}}^{(0)} = s_0 Q_{\mathsf{id}}$.

4. On receiving t private key shares $d_{\mathsf{id}}^{(i)}$ $(1 \leq i \leq t)$, $\mathcal{P}$ computes $\mathcal{D}_{\mathsf{id}}$'s private key as $d_{\mathsf{id}} = \sum_{k=0}^{t} \lambda_i s_i Q_{\mathsf{id}}$ $(\in \mathbb{G}_1)$, where the λ_i's are the appropriate Lagrange coefficients for the polynomial $f(z)$ [19].

It is straightforward to see that $\mathcal{P}$ is able to compute its private key with the help of t other PKGs. Once a private key has been computed by $\mathcal{P}$, it must be securely transferred to the requesting PAN device $\mathcal{D}_{\mathsf{id}}$. This secure channel must provide data origin authentication as well as confidentiality. Such a channel can be established using Bluetooth SSP or the MANA protocols as described in section 3.3.

[5]Bearing in mind the error-prone nature of wireless links, sending the request information to greater than t other PKGs will reduce the risk of $\mathcal{P}$ not receiving enough private key shares.

4.2.3 Analysis

We give an analysis on how the security initialisation scheme satisfies the design goals, i.e. secrecy, correctness, efficiency, and minimal initial secure channels.

Secrecy and correctness

The master secret distribution makes use of the Gennaro et al.'s scheme [56] described in Section 2.6.4. The Gennaro et al. [56] prove that their scheme satisfies the followings: (i) all subset of $t+1$ master secret shares provided by honest parties define the same unique master secret; and (ii) no information regarding the master secret can be learned by the adversary. The proposed scheme thus satisfies the the secrecy and correctness requirements for the process of master secret distribution.

We now discuss the private key extraction and transfer process. Any information regarding the private key of a PAN device belonging to the honest, i.e. not corrupted, PKG devices, is never learned by the adversary, because it is infeasible to compute the private key from less than $t+1$ private key shares and the computed private key is transferred to the requesting device in encrypted form. It is, of course, possible for an adversary to learn the private key of a device belonging to a PAN with a compromised PKG.

Given each node's public key, the private key is computed from the correct $t+1$ private key shares and the correctness of the private key shares is verified by the PKG device on behalf of the requesting PAN device, using the fact that the DDH problem is easy in $\mathbb{G}_1$ [18]. More specifically, $\mathcal{P}$ first computes U_i for each $\mathcal{P}_i$ as follows.[6]

[6]We note that a pair (s_i, U_i) can be used as a private/public key pair, e.g. for Schnorr's signature scheme [126] or Hashed ElGamal [40], a variant of the ElGamal encryption scheme, as proposed by Saxena [124]. In this case, these cryptographic schemes could be viewed as examples of IBC (without using pairings), since a node can send an encrypted message and verify signatures with the knowledge of the identifier of a particular node and the public system parameters. However, unlike other identity-based schemes, these schemes become insecure once more than t PKGs are compromised. These schemes would be appropriate for a private network with a relatively small size. We note that using the secret shares in multiple schemes would violate the principle of key separation and may jeopardise the security of the schemes, depending on how they interact [102].

$$\begin{array}{rcccl} U_i & := & s_iP & = & \sum_{i\in\mathcal{Q}} a_{i0}P + \sum_{i\in\mathcal{Q}} a_{i1}iP + \cdots + \sum_{i\in\mathcal{Q}} a_{it}i^tP, \\ & & & = & \sum_{i\in\mathcal{Q}} A_{i0} + \sum_{i\in\mathcal{Q}} iA_{i1} + \cdots + \sum_{i\in\mathcal{Q}} i^tA_{it} \ \ (\in \mathbb{G}_1), \end{array} \tag{4.2}$$

The identifier i and the values A_{ik} ($i \in \mathcal{Q}$ and $k = 0, 1, \ldots, t$) are public values. The private key share $d_{\mathsf{id}}^{(i)}$ can then be verified by checking whether

$$\hat{e}(d_{\mathsf{id}}^{(i)}, P) = \hat{e}(Q_{\mathsf{id}}, U_i) \ \text{ in } \mathbb{G}_2. \tag{4.3}$$

Equation (4.3) holds because

$$\hat{e}(d_{\mathsf{id}}^{(i)}, P) = \hat{e}(s_iQ_{\mathsf{id}}, P) = \hat{e}(Q_{\mathsf{id}}, P)^{s_i} = \hat{e}(Q_{\mathsf{id}}, s_iP) = \hat{e}(Q_{\mathsf{id}}, U_i) \ \text{ in } \mathbb{G}_2.$$

If equation (4.3) does not hold, $\mathcal{P}$ discards the private key share $d_{\mathsf{id}}^{(i)}$ and broadcasts a complaint against $\mathcal{P}_i$.

Finally, the authenticity and integrity of transferred private key is guaranteed by the secure channel established by using the schemes described in Section 3.3.

Efficiency

By delegating private key extraction to a PKG device in the same PAN, both the computations and communications that non-PKG devices need to perform are decreased. As a result, a PAN device will only need to contact its PKG instead of communicating with t PKGs for private key generation, as required by other identity-based DPKG proposals for ad hoc networks [34, 87, 144]. Indeed, personal wireless devices, which generally have a local communications interface with limited capacity, will not necessarily be able to connect to as many as t PKGs. Finally, observe that the overall energy efficiency of the novel scheme is significantly greater than that of rival schemes, because wireless transmission of a bit can require over 1,000 times more energy than a single 32-bit computation [144].[7]

[7] Transmitting a sufficiently powerful signal or decoding a received spread-spectrum signal involves considerable energy consumption, equivalent to that used by several thousand cycles a CPU; see, e.g. http://xbow.com

Table 4.1: Comparison of the number of required initial secure channels

	Deng et al. [34]	Khalili et al. [87]	Liu et al. [144]	Proposed scheme
Number of initial secure channels	$O(m^2)$	$O(m^2)$	$O(n^2) + O(mn)$	$O(m)$ or $O(n^2) + O(m)$

$*$ The value m denotes the number of network nodes in an iPAN, and n denotes the number of PKG devices (clearly $m \geq n$).

Minimal initial secure channels

Finally, the proposed scheme is potentially efficient in its use of secure channels during initialisation, as shown in Table 4.1. The related work quoted in this table, [34, 87, 144], is discussed in section 4.1.3.

4.3 Further Discussion

In this section we briefly describe the example use case of pre-distributed key pairs, and discuss the subsequent key management process after security initialisation in iPANs.

4.3.1 Use of pre-distributed keys

Any two network nodes, with identifiers id_a and id_b say, can agree a shared secret key without any communication [121]:

$$\begin{aligned} K_{ab} &= \hat{e}(d_{\mathsf{id}_a}, Q_{\mathsf{id}_b}) = \hat{e}(Q_{\mathsf{id}_a}, Q_{\mathsf{id}_b})^s \\ &= \hat{e}(Q_{\mathsf{id}_b}, Q_{\mathsf{id}_a})^s = \hat{e}(d_{\mathsf{id}_b}, Q_{\mathsf{id}_a}) = K_{ba} \in \mathbb{G}_2 \end{aligned} \tag{4.4}$$

where d_{id_i} and Q_{id_i} are the private/public keys respectively of the network node with identifier id_i.

In fact, the nodes only need to compute K_{ab} (or K_{ba}) once and can cache the result, obviating the need for an expensive pairing computation which (after the initial key computation) makes the system as efficient as would be the case if secret key

cryptography was used for key establishment. A PAN device could ask its PKG to compute the shared secret on its behalf if pairing computations are beyond its computing capabilities. The shared secrets can be used as long-term shared secrets between a pair of network nodes.

The iPAN devices, however, subject to physical attacks which would compromise secure data stored in the devices. The use of the non-interactive key establishment protocols between PAN devices is thus vulnerable to attacks in which the attacker corrupts the devices, extracts the keys, and then retroactively reads communications between the devices which had occurred before the devices were corrupted.

The concept of the forward security [38] protects against the threats of this kind by ensuring that the security of past uses of keys is not compromised by the exposure of the currently stored keys. Bellare and Yee [13] dicuss a comprehensive treatment of forward-security in the context of shared-key based cryptographic primitives, as a practical means to mitigate the damage caused by key exposure, showing how forward-secure message authentication schemes and symmetric encryption schemes can be built based on standard schemes.

Since implementing full PKI functionality in constrained devices appears likely to be problematic, a number of authors have considered how TLS might be modified to use shared secrets, especially in mobile environments. These methods typically avoid expensive public-key operations involving the exchange of certificates in the TLS handshake, while providing an equivalent level of security using shared secret secret keys. Gutmann [63] suggests seeding the TLS session cache with the shared key and using session resumption functionality without changing the TLS protocol. Eronen and Tschofenig [42] propose three sets of new ciphersuites for the TLS protocol to support authentication based on shared secrets. In particular, the first set of ciphersuites uses only symmetric algorithms and is thus suited to performance-constrained devices.

4.3.2 Revocation and update

We have proposed a security initialisation scheme for iPANs, but the subsequent key management should be provided. In this section we briefly discuss such process, i.e.

key revocation and update.

Revocation

Cryptographic schemes themselves cannot protect against attacks from compromised but non-revoked network nodes, i.e. an internal adversary [84]. To minimise the possible damage from compromised nodes, the public keys of such nodes should be revoked immediately the problem becomes known. It is not sufficient to associate an expiry date with a public key to support key revocation (as described in [18]) in iPANs, since the public keys of malicious or compromised devices may need to be revoked prior to their expiry.

In the key revocation we briefly described here, PKGs are given the right to generate an accusation of misbehaviour against any iPAN node. If a non-PKG device detects misbehaviour by another network node, then the device reports the misbehaviour to its PKG. The PKG generates a signed accuasation against a device, $\mathcal{D}_{\mathsf{id}}$ say, by computing $\mathsf{Sig}(sk, \tilde{\mathsf{id}})$, where $\tilde{\mathsf{id}} = \mathsf{id}||\texttt{revoke\_request}||t$ and Sig is the signing algorithm in Definition 2.18. The value t denotes the time when the signature is generated. The PKG then propagates the signed accusation to the other PKG devices. For the generation and verification of the signed accusations, each PKG may use the private/public key pair provided in the initial secure channels[8] with any appropriate secure signature schemes [102].

On receiving an accusation, a PKG discards it if the PKG issuing the accusation has been revoked. Otherwise, the PKG adds the accusation to its accusation list. If the number of accusations against an iPAN device reaches a revoke threshold r during the predefined period, the device is deemed to be *revoked*. To prevent an adversary from attempting to revoke an honest PKG, the value r should be no less than $t + 1$ (i.e. $r \geq t + 1$)[9]. The list of the revoked devices is propagated within a PAN by a PKG device using the secure channels. The choice for the threshold r is a trade-off between false-accusation tolerance and compromise detectability. A larger choice for r will give a greater tolerance to false accusations, but a reduced ability

[8] The security initialisation process assumes that there exist the initial secure channels between PKGs (see Section 4.2.1).

[9] The adversary is assumed to be able to compromise up to t PKGs as described in the security initialisation phase (see Section 4.2.2).

to detect compromised nodes. The reasonable choice of the revoke threshold value would be $r = t + 1$.

We note that naive broadcast of a signed accusation, however, may be insecure, since it could alert the accused network node to temporarily behave normally, as pointed out by Zhang et al. [144]. By misbehaving selectively, the misbehaving device could keep the number of accusations below the threshold r. To avoid this possibility, the signed accusation may be sent to the other PKG devices in the encrypted form.

We note that it requires considerable computation and communication activities to participate the revocation process. Such activities, however, are performed by the PKG devices, and the other resource-constrained PAN devices can detect the revoked devices using the list received from its PKG device. The revocation scheme also provides a greater efficiency in storage. PKG devices should maintain the list of any received signed accusations, but the other PAN devices only have to store the list of identifiers whose keys have been revoked.

Update

We first describe the share update process, and then how a network node updates its public/private key pair. We also describe how the share update technique could be used to support membership update, i.e. when an existing PAN leaves or a new PAN joins an iPAN.

Share update: Herzberg et al. [68] proposed a share update technique called proactive secret sharing to cope with a mobile adversary of the type defined in section 4.2.1. The idea is to *refresh the shares* at the beginning of each time period in such a way that: (i) the updated shares are independent of the shares in the previous time period; and (ii) the master secret remains the same. Assume that each $\mathcal{P}_i$ has a share s_i of a master secret s, i.e. $f(i) = s_i$ and $f(0) = s$, as in section 4.2.2. All the PKGs $\mathcal{P}_i$ ($i \in \mathcal{I}$) can engage in proactive secret sharing as follows.

- The PKGs $\mathcal{P}_i$ ($i \in \mathcal{I}$) jointly generate $g(z) = \sum_{k=0}^{t} b_k z^k \in_R \mathbb{Z}_q[z]$ such that $b_0 = 0$ ($\in \mathbb{Z}_q$), by performing the Pedersen's DPKG (see Section 2.6.4) scheme, i.e. they compute a new master secret equal to 0 ($\in \mathbb{Z}_q$).

- Each $\mathcal{P}_i$ then updates its share to $s_i + g(i) \bmod q$, where s_i is its old share. The value $g(i)$ can be computed locally at each $\mathcal{P}_i$, as described in section 2.6.4.

The previous polynomial $f(z)$ is now implicitly updated to $f(z) + g(z)$, but the master secret remains the same since $g(0) = 0$ ($\in \mathbb{Z}_q$).

Key update: In line with generally accepted key management principles, key pairs should be updated periodically. The key management scheme defined here enables this by including the expiry date within public keys, as originally proposed in [18, 19]. When the public key of a node expires, a new public key can be automatically computed by any entity in the network, i.e. the previous public key $h(\mathsf{ID})$ is updated to $h(\mathsf{ID}')$, where $\mathsf{ID} = \mathsf{id}||\,\texttt{expiry-date}$ and $\mathsf{ID}' = \mathsf{id}||\,\texttt{expiry-date} + \texttt{predetermined period}$. The corresponding private key can be extracted using the method described in section 4.2.2.

Membership update: To accommodate PANs joining and leaving iPANs, any necessary changes of configuration, such as from use of an (n, t) threshold scheme to an (n', t) scheme, will need to be securely managed. When a PAN leaves iPANs, the remaining PKGs must perform the share refreshing process without the involvement of the PKG device in the leaving PAN. As a result, the leaving PKG will be excluded from any future key management procedures. The public keys of all nodes in the leaving PAN will need to be revoked; to achieve this the leaving PKG could broadcast the list of pairs $(\mathsf{id}_i, \sigma_i)$, where id_i is the identifier of device in its PAN and σ_i is a signature on $\mathsf{id}_i||\,\texttt{revoke-request}$. On the other hand, when a new PAN joins an iPAN, the PKG device in the joining PAN must perform the share refreshing procedure with the existing PKGs. The private keys for the other PAN devices can then be generated as described in section 4.2.2.

4.3.3 Other security issues

A fundamental assumption for the use of IBC is that the PKG is secure. This is a reasonable assumption, because it is typically reasonable to assume that the PKG can be kept physically secure. This is, however, not the case in an iPAN; a PKG device could be compromised (e.g. mobile phones are prone to theft) or simply be

unavailable (e.g. a laptop or a mobile phone might be switched off).

The proposed scheme (i.e. use of a DPKG) is resilient to failures of one or more PKGs, and also tolerant of faulty or malicious behaviour by up to t PGKs. However, a PKG in a PAN represents a single point of failure for the communications of the other devices within this PAN. To avoid this issue, two or more devices[10] in the same PAN could act as PKGs. One device would be nominated as the primary PKG, and the others as secondary PKGs. All such PKGs would participate in the master secret sharing protocols as defined in section 4.2.2, and each PKG would be assigned a different share. Assuming that all the PKGs in a PAN are known to the PAN devices, one of the secondary PKGs can replace the primary PKG in the event of the failure of the primary PKG.

Commonly discussed routing protocols, such as AODV [112] and DSR [74], allows adversaries to locate PKGs easily by eavesdropping on identifiers sent in data packets. Zhang et al. [144] point out that an adversary could launch a pinpoint compromise or disruption attack given knowledge of the identity of a PKG. Such an attack could be prevented by using the MASK [143] routing protocol, which uses dynamically changing pseudonyms in the routing process without disclosing the real identities of packet sources, packet destinations, and all intermediate nodes.

4.3.4 Generalisation of the proposed scheme

The proposed scheme can be applied to an ad hoc network. An ad hoc network is a network formed without any central administration, which consists of (mobile) nodes that use a wireless interface to send data packets [49]. So far, ad hoc packet-radio network have been mainly considered as military applications, where decentralised configurations are usually required. However, as capacity of personal wireless devices increases in the commercial sectors, the needs of ad hoc networking of such devices are rising. Furthermore, ad hoc network in commercial sectors are more likely to form iPANs, because a single (master) user could deploy more than two wireless devices to maximise the resources within the ad hoc network. The proposed scheme can be applied to any ad hoc network which can be divided into a collection of

[10] Such devices should be computationally powerful and physically secure by comparison with other devices in the PAN.

smaller sub-networks, where a certain degree of trust exists between devices within each sub-networks.

4.4 Conclusions

Since personal wireless devices are becoming ubiquitous, the need for secure communications within and between PANs is becoming increasingly common. Providing robust and secure key management scheme remains a challenging task, in particular because of the unique characteristics of, and constraints on, such networks. Whilst it is possible to employ existing key management schemes for ad hoc networks to secure intra/inter-PAN communication, we have proposed a novel scheme which achieves a greater degree of security and efficiency by taking advantage of the unique characteristics of PANs.[11]

[11] See Section 4.1.1 for the details of the unique characteristics of iPANs

Part II

Pervasive Computing: Radio Frequency Identification

Chapter 5

Privacy in RFID Systems

Contents

We discuss privacy issues of RFID technology, and construct a formal privacy model for RFID systems accurately reflecting adversarial threats and power. We then give brief privacy analysis for the existing privacy-enhanced RFID schemes which have received wide attention in the literature.

5.1 Privacy Issues in RFID Systems

Radio Frequency Identification (RFID) is a technology for automated identification of objects or people using radio communications. An RFID system consists of tags and readers. A Radio Signal Transponder, commonly known as an *RFID tag* or simply a *tag*, consists of a chip containing identity information and an antenna for wireless data transmission. Such a tag is typically attached to an object, and transmits resident data when the tag passes through a radio frequency (RF) field generated by a compatible *reader*.

The use of radio identification technology dates back at least to the second world war, when the British attached transponders to aircraft to differentiate their own planes from others [118]. Only recently, however, has such a technology been seriously considered for commercial applications. Use of this technology has been pioneered by large organisations, such as WalMart, Proctor & Gamble, and the United States Department of Defense, who have attempted to implement RFID technology in their supply chains [77]. A combination of falling tag costs and global RFID standardisation makes rapid growth in adoption likely in the near future.

RFID technology, however, poses unique privacy and security concerns. In particular, the owner of a tag cannot physically control tag communications because: (i) radio communications are non-contact and non-line-of-sight, (ii) the tag itself typically does not maintain any history of past readings, and (iii) the tag does not possess a human interface. The potentially limited computing capabilities of RFID tags render security threats more serious, since standard cryptographic primitives are beyond the capabilities of most RFID tags. Also, because of their small size, people can carry RFID tags without their consent or even knowledge. Furthermore, since such tags will operate in hostile environments, they may be subject to a variety of physical attacks [138], including fault induction and power analysis attacks [17].

5.1.1 Malicious tag readings in RFID systems

Unauthorised tag reading is probably the most serious security and privacy threat to an RFID system, since it could either directly infringe privacy and/or security, or facilitate further malicious activities such as tag cloning. We divide unauthorised

tag reading into two types, i.e. passive and active attacks. An unauthorised reader in the field could surreptitiously listen to tag-reader communications. Alternatively, it could actively attempt to read tags, since passive tags send their identifiers without further security verification to an interrogating reader.

Tag-read ranges are an important factor when discussing security and privacy issues. The standards specify a range of frequency bands, along with nominal read ranges, for passive tags [77]. LF (Low-Frequency) tags operate at 124–135 kHz and have nominal read ranges up to half a metre; HF (High-Frequency) tags operate at 13.56 MHz and have read ranges up to a metre or more; Ultra High-Frequency (UHF) tags operate at 860–960 MHz (sometimes 2.45 GHz) and have read ranges up to tens of metres. The following issues also affect reading ranges.

Forward link vs. reverse link in a passive attack

Since passive tags communicate using backscattering energy sent by readers, a very high energy signal is used on the *forward link* from readers to tags; tags respond by backscattering a very small portion of that energy on the *reverse link*. As a result, the high power forward link can be read up to hundreds, or even thousands, of metres away, depending on the regulations, antenna, and environment. The reverse link, on the other hand, can be typically observed only within tens of metres [70]. Sending tag-sensitive information on the forward link should thus be avoided.

Nominal read ranges vs. rogue read ranges in an active attack

The read ranges for the reverse link can be further sub-divided. The nominal read range indicates the maximum distance at which tags can be scanned reliably by a normally operating reader, i.e. a reader with an ordinary antenna and power output. A rogue reader, however, may be equipped with more a sensitive antenna and could generate power beyond the legal limits. Such a rogue read range could be up to five times the nominal read range [86].

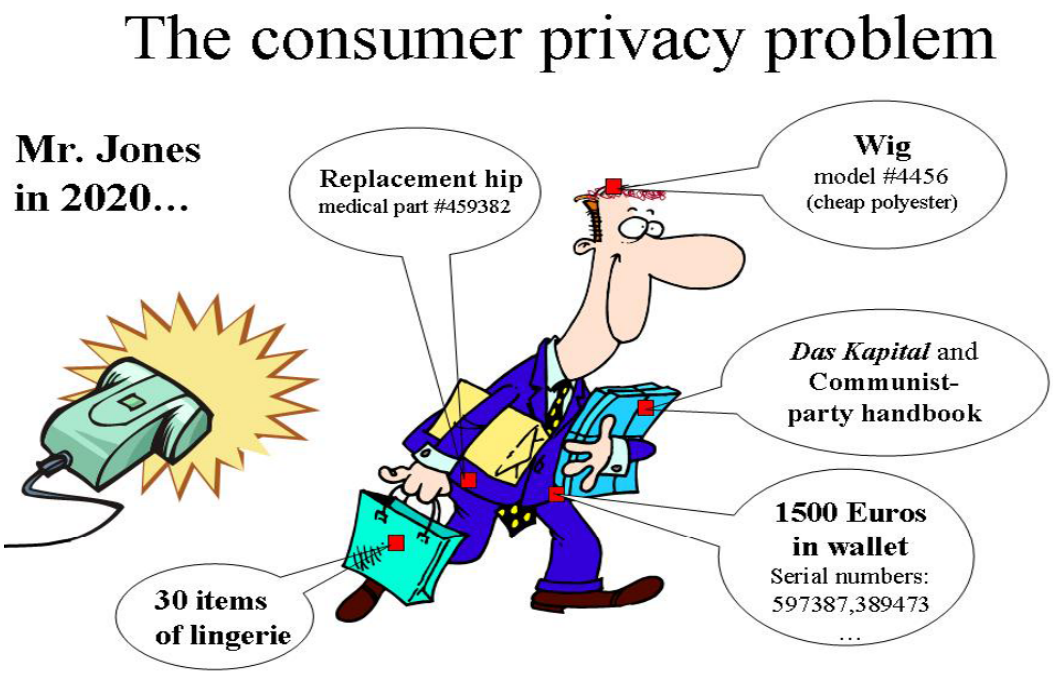


Figure 5.1: Potential RFID consumer privacy problems [77]

5.1.2 Threats in practical scenarios

Indeed, RFID technology has already raised privacy concerns in a range of practical scenarios as follows.

EPC tags: The EPCglobal Network [71] is a standards-based approach designed to help realise automated global supply chain management. An EPC (Electronic Product Code) tag is the key component in the EPCglobal Network. The EPC tag is an RFID tag, which is attached to, or embedded in, items. EPC tags may eventually replace conventional barcodes for the items used in supply chains, but they will introduce serious privacy issues. For example, mobile RFID devices [88, 89, 110], i.e. tag readers embedded within a mobile phone and capable of communicating with the EPCglobal Network via a mobile service provider, are being developed. Such devices, when combined with the network of fixed tag readers, would significantly increase threats of unauthorised tag reading. Figure 5.1 illustrates the potential privacy threats arising from unauthorised tag reading. We further discuss privacy issues on EPC tags in the following chapter.

Banknotes: The European Central Bank planned to embed RFID tags in ban-

knotes as an anti-counterfeiting measurement.[1] Juels and Pappu [79] proposed a practical cryptographic banknote protection scheme using both Optical and RFID systems. However, Avoine [6] demonstrated that the Juel-Pappu scheme [79] severely compromises the privacy of banknotes bearers.

Libraries: Some libraries have implemented RFID systems to facilitate book check-out and inventory control. This may be the first major deployment of item-level tagging, where each individual item is given its own tag. Molnar and Wagner [106] discuss privacy issues for RFID-enabled library users.

e-Passports: The International Civil Aviation Organization (ICAO) has published a guideline for RFID-enabeld passports, so called the e-Passport [78]. An Integrated Circuit (IC) embedded in the cover pages of an e-Passport contains: a copy of the Machine Readable Zone (MRZ)[2], a digital facial image, and other optional biometric data for the passport holder. RFID technology is used to read the information stored in the e-Passport. The security and privacy threats of e-Passport have been widely discussed [8, 66, 78, 90].

Human-implantable chips: The possible use of VeriChip[3], a human-implantable RFID tag, has fuelled major privacy concerns arising from possible physical tracking of individuals. Its proposed uses include in healthcare and emergencies, as well as security applications such as physical access control. Halamka et al. [64] show that Verichip is vulnerable to a simple cloning attack, and also suggest that, for bearer safety, such tags should be exclusively used for identification and not authentication.

5.1.3 Privacy as fundamental requirement

Privacy, in particular, seems to be a *fundamental property* required in all RFID systems, since, if industry fails to address privacy concerns, RFID technology is likely to be deployed in environments in which legislation will limit its operation. There is also likely to be public resistance to deployment as described below.

[1] See http://www.eetimes.com/story/OEG20011219S0016

[2] The MRZ is the set of optically-readable encoded lines at the bottom of the first page of a passport, and includes information such as document type, full name, passport number, nationality, date of birth, and gender.

[3] See http://www.verichipcorp.com

The European Union has created a working group[4] to protect consumer privacy in connection with personal data gathered using RFID technology. In the United States, several states have initiated RFID privacy legislation [77]. Consumer privacy advocates have mounted campaigns against RFID deployment by retailers. For example, in 2003 a boycott caused Benetton to abandon RFID plans for its products.[5] In the same year, a group of privacy organisations signed a position statement on the use of RFID technology in consumer products.[6]

Malicious tag reading is not just a consumer privacy concern; it also poses a security threat to the supply chain. For example, identifying individual tagged items could make it easier for competitors to learn about a company's stock turnover rates in its supply chains. This threat is potentially even more serious in a military supply chain,[7] e.g. because enemy forces might learn about troop movements by monitoring RFID communications.

5.2 Defining Privacy of RFID system

It is important to note that the cryptographic security models only capture the security properties of the application layer communications protocols used between tags and readers. Avoine and Oechslin [10] point out that there are multilayer privacy issues; for example, if a tag has a distinct radio fingerprint (e.g. due to its use of a different underlying standard at the communication/physical layer), then the most secure cryptographic privacy-preserving identification protocol running at the application layer may be of no use.

Several privacy models of RFID systems have been proposed in the literature [4, 7, 23, 75, 80, 137], but most privacy models fail to cover all major attack scenarios, threats to the system, possible platforms, and attack strategies. We define a security model for privacy in RFID systems, which captures multiple security levels in many different categories.

[4] See http://ec.europa.eu/justice_home/fsj/privacy.
[5] See http://www.boycottbenetton.com.
[6] See http://www.privacyrights.org/ar/RFIDposition.htm.
[7] The Department of Defence in the United States has ordered that all shipments to its armed forces be equipped with RFID tags [77].

5.2.1 Related work

The definition of *privacy* used in the existing RFID systems varies depending on the system components and the adversarial capabilities, but the fundamental property is summarised as follows: *An adversary should not be able to link two different sessions involving the same tag.*

In most cryptographic security models, an adversary is assumed to have more-or-less unfettered access to system components. Such access, however, will be a sporadic event in most RFID systems, since, in order to scan a tag, an adversary must have physical proximity to the tag. Moreover, because most inexpensive RFID tags cannot perform standard cryptographic functions, they cannot provide a meaningful level of security against too strong an adversary. This provides an additional motive to consider security models involving a less powerful adversary as the security model described below.

Juels [75] points out the inevitable computational limitations of low-cost RFID tags, describes likely attack scenarios in real-world settings, and provides a practical formalisation of *minimal* security requirements for a low-cost RFID tag. More specifically, the minimalist security model assumes that an adversary can read a given tag only a specified number of times; once this number is reached, the tag is assumed to interact with a valid reader outside the eavesdropping range of the adversary. Le et al. [93] define that a refresh-based RFID system is said to provide privacy if the adversary cannot link two reads of the same tag if the tag has been refreshed outside the eavesdropping range of the adversary between the two read events. Juels and Weis [80] focus on a *strong* privacy requirement for an RFID system in which secret key cryptographic operations are assumed to be possible in an RFID tag, but they exclude the adversarial capability of corrupting a certain class of tags, i.e. the two tags that the adversary is challenged to distinguish in the privacy experiment.

Security, however, is not a binary state, so the security model may have many levels in many different categories of RFID systems. Vaudenay [137] presents a security model by defining adversaries with a range of strengths and introducing the eight levels of privacy; each level of privacy considers the adversary with different strength.

5.2.2 Defining RFID system

An RFID system consists of the following two entities.

- A tag, denoted by $\mathcal{T}$, is a *passive* transponder; tags have no on-board power source and are powered from an interrogating reader. Such use of an external power source limits communication range of tags up to only few metres. A tag also has limited size of storage and computational capabilities.
- A reader, denoted by $\mathcal{R}$, consists of one or more *transceivers* and a *back-end server*. The transceivers send the captured data from a tags to the back-end server in order to determine if such tags are legitimate, i.e. if tags are registered in the database $D_{\mathcal{R}}$, and further identify the legitimate tags, i.e. recover tag identifiers id's.

An RFID system consists of a set $\mathbf{P}$ of polynomial-time algorithms of the following types.

- Setup-Reader takes as input a security parameter 1^n, and returns the system parameter $parm$ and a key $K_{\mathcal{R}}$. We write $\mathcal{R}(parm, K_{\mathcal{R}}) \leftarrow \mathsf{Setup\text{-}Reader}(1^n)$.
- Setup-Tag takes as inputs $parm$, a key $K_{\mathcal{R}}$, and a tag identifier id. It returns a tag-specific secret $K_{\mathcal{T}}$ and the initial state S of a tag. The pair $(id, K_{\mathcal{T}})$ is stored in $D_{\mathcal{R}}$, and S in a tag $\mathcal{T}$. $K_{\mathcal{T}}$ is not necessarily stored in the tag, but the initial state S may be defined to include $K_{\mathcal{T}}$. We write $(\mathcal{R}(id, K_{\mathcal{T}}), \mathcal{T}(S)) \leftarrow \mathsf{Setup\text{-}Tag}(id)$.
- Protocol is a polynomial-time interactive protocol between a tag and a reader. A reader initiates the protocol, and it ends with a tape output. We write $\mathsf{output} \leftarrow \mathsf{Protocol}(\mathcal{R}, \mathcal{T})$.

We now give a formal definition of an RFID system.

Definition 5.1 (RFID system) *An RFID system consists of a tuple $(\mathcal{T}, \mathcal{R}, \boldsymbol{P})$, as defined above, satisfying the following viability condition. That is, for the following experiment*

$\mathcal{R}(parm, K_{\mathcal{R}}) \leftarrow \mathsf{Setup\text{-}Reader}(1^n)$;
$(\mathcal{R}(id, K_{\mathcal{T}}), \mathcal{T}(S)) \leftarrow \mathsf{Setup\text{-}Tag}(id)$;
$\mathit{output} \leftarrow \mathsf{Protocol}(\mathcal{R}, \mathcal{T})$.

we have $\mathit{output} = id$.

5.2.3 Defining privacy

Providing rigorous definition of the notions of privacy in an RFID system involves constructing a formal model that characterises the capabilities of potential adversaries. Such a model typically takes the form of an *experiment*, which intermediates in communications between an adversary and a runtime environment, also called a *challenger*, which contains the system components. In an RFID system, the system components are tags and readers.

We now define the notion of privacy required in RFID systems based on the notion of indistinguishability. Following the security model of Vaudenay [137], we define the multiple levels of privacy depending on the adversarial capability of corrupting the tags and the availability of side channels. We, however, further generalise the existing privacy model by reflecting the fact that the adversarial access to the system components would be a sporadic event in RFID systems. More specifically, we describe the types and capabilities of adversary as follows.

- **Accessibility to tag internal states:** Adversary may be able to recover the internal states of tags, including the tag secrets, and such a capability is characterised by the `Corrupt` query.
- **Accessibility to side channels:** Adversary may be able to access the side channels such as the result of RFID protocols, e.g. whether the interrogated tags are successfully identified or not, and this could greatly affect tag privacy as discussed in the following sections. Such a capability is characterised by the `Result` query.
- **Pervasiveness of adversary:** In order to either scan a tag or listen to messages from a reader, an adversary must be in physical proximity to the tag or the reader. The most powerful adversary may listen to tag-reader communication

or send messages to tags and reader anytime. Otherwise, a tag may be assumed to have communication with a reader outside the eavesdropping range of the adversary between every two consecutive adversarial access to the tag. There may be a certain upper limit on the number that the adversary is allowed to access tags during the lifetime of tags. An adversary may be assumed to have an access to a tag once in a particular length of time. We characterise such a adversarial capability by defining the *access* types.

Reflecting the adversarial capabilities described above, we now construct a formal privacy model.

Definition 5.2 (Adversary) *An adversary $\mathcal{A}$ is a probabilistic polynomial-time (PPT) algorithm, which can make the following types of oracle queries:*

- `Create-Tag`(id): *creates a tag $\mathcal{T}$ with a tag identifier id using* Setup-`Tag`(id)*, and $(id, K_{\mathcal{T}})$ is added to the database of the backend server.*
- `Launch`$(protocol) \rightarrow \pi$: *makes a reader initiate the protocol instance π.*
- `Send-Reader`$(m, \pi) \rightarrow m'$: *sends a message m to a reader for a protocol instance π, and receives the answer m'.*
- `Send-Tag`$(m, \pi, \mathcal{T}) \rightarrow m'$: *sends a message m to a tag for a protocol instance π, and receives the answer m'.*
- `Execute`$(protocol, \mathcal{T}) \rightarrow (\pi, trasncript)$*: performs one* `Lauch` *query and successive use of* `Send-Reader` *and* `Send-Tag` *to execute a complete protocol between the reader and the tag $\mathcal{T}$. It returns the transcript of the protocol, i.e. the list of successive protocol messages.*
- `Result`$(\pi) \rightarrow x$: *at the end of execution of π, returns 1 if the tag has been successfully identified and 0 otherwise.*
- `Corrupt`$(\mathcal{T})$: *returns the current state S of $\mathcal{T}$.*

The adversaries are classified by the capability of using the oracle queries. The weak adversary cannot use `Corrupt` *query. The destructive adversary cannot use $\mathcal{T}$ again after the* `Corrupt`$(\mathcal{T})$ *query. The strong adversary, however, still has an access to*

the queries involving $\mathcal{T}$ after the `Corrupt`$(\mathcal{T})$ *query. The* wide *adversary has access to the* `Result` *query, but the* narrow *adversary does not.*

Apart from the adversarial capabilities to use the particular oracle queries, we also define three access *types of adversary. The* universal *adversary make queries involving any tag anytime during the experiment. The* discrete *adversary, prior to making any query, should send the* `Execute` *query, where the* transcript *is not given to the adversary. The* discrete-τ *adversary cannot make more than one oracle query during the time interval τ.*

We next define the experiments given the RFID system and the adversary $\mathcal{A}^{\delta}$, where the parameter δ denotes the adversarial capability of using oracle queries and the access type.

Experiment $\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^{\delta},\mathsf{RFID}}(n,\omega,b)$

Learning Stage:

(1) A single reader or two readers are set up.

(2) $\mathcal{A}^{\delta}$ makes the allowed oracles queries, without exceeding ω overall queries.

Challenge Stage:

(3) $\mathcal{A}^{\delta}$ outputs two tags, $\mathcal{T}_0$ and $\mathcal{T}_1$ say, and one of the tags, $\mathcal{T}_b$ say, is selected.

(4) $\mathcal{A}^{\delta}$ makes the allowed oracle queries involving $\mathcal{T}_b$ depending on their *access* types, without exceeding overall ω queries.

(5) $\mathcal{A}^{\delta}$ outputs a guess bit $d \in \{0,1\}$.

The variable ω is polynomial in the security parameter n. Most RFID schemes assume a single reader within the system, but some schemes aim to provide privacy of tags which have been set up by different readers [2, 27, 28, 62]. Privacy of such schemes can be analysed by setting up two readers as in step (1) and investigating if the given adversary can distinguish the two tags which have been set up by different readers.

In step (3) the adversary cannot output the tags which have been set up by the corrupted readers. In step (4) the universal adversary can make any allowed oracle queries after $\mathcal{T}_b$ is selected. The discrete adversary should first make the Execute(protocol, $\mathcal{T}_b$) query without the transcript given, before making the allowed oracle queries. Finally, the discrete-τ adversary cannot make more than one oracle query during the time interval τ.

We define privacy to mean that every adversary *behaves in the same way* in the privacy experiment regardless of whether it interacts with $\mathcal{T}_0$ or $\mathcal{T}_1$ in the challenge stage. Since the adversary $\mathcal{A}^\delta$ outputs a single bit, "behaving in the same way" means that it outputs "1" with almost the same probability in both cases. For the privacy experiment defined above, the advantage of $\mathcal{A}^\delta$ can be defined as

$$\mathbf{Adv}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID}}(n,\omega) = \left|\Pr\left[\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID}}(n,\omega,1)=1\right] - \Pr\left[\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID}}(n,\omega,0)=1\right]\right|.$$

The following definition captures the notion that the adversary $\mathcal{A}^\delta$ cannot determine whether it is running the experiment $\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID}}(n,\omega,0)$ or the experiment $\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID}}(n,\omega,1)$.

Definition 5.3 ((δ,ω)-privacy) *An RFID system is said to provide (δ,ω)-privacy if the function $\mathbf{Adv}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID}}(n,\omega)$ is negligible, where $\delta \in \{x,y,z\}$, $x \in \{$universal, discrete, discrete-$\tau\}$, $y \in \{$wide, narrow$\}$, and $z \in \{$strong, destructive, weak$\}$.*

We note that, if the system provides *universal* privacy, it also provides *discrete* and *descrete-τ* privacy; *wide* privacy means *narrow* privacy; *strong* privacy means *destructive* privacy, which in turns means *weak* privacy.

5.3 Secret key Cryptographic Solutions

Since inexpensive RFID tags cannot perform public key cryptography, most privacy-enhancing RFID schemes have proposed to use secret key cryptography. A technique known as key search[8] has been most widely discussed in the literature, as surveyed

[8]The term *key* means bit-strings that are used by the tags to prove their identity. They are *not keys* in the classical cryptographic definition, in that they do not control the operation of a commonly-accepted cryptographic primitive (see Section 2.6.2).

by Avoine [5]. In this section we first discuss the general idea of RFID security protocols based on key search, and then describe the variants of key-search RFID schemes which aim to enhance security and efficiency.

5.3.1 Protocols based on key search

In privacy-enhanced RFID schemes, we need to find a means for a tag to identify itself to a reader without revealing its identifier in cleartext. In order to achieve this property using secret key cryptography, the majority of published privacy-preserving RFID protocols use a method called key search, first discussed by Weis et al. [140].

A tag and a reader share a tag-specific key k, and perform a protocol as follows. Given a pseudo-random function f, a tag chooses a random number r, computes $s = f(k, r)$, and sends (r, s) to a reader. A reader is then able to identify a tag by finding a key k^* such that $s = f(k^*, r)$. The value r should be different at every session so that the output s would be different from session to session.

It is easy to see that the scheme cannot achieve *strong* privacy, since the adversary can determine the value b in the challenge stage after recovering the keys stored in $\mathcal{T}_0$ and $\mathcal{T}_1$ in the learning stage. The scheme satisfies (*universal*, *wide*, *destructive*, ω)-privacy, provided that the a pseudo-random function f is a random oracle. This can be proven by showing that $\mathcal{A}^\delta$ gains no knowledge from its interaction with $\mathcal{T}_b$. More specifically, a simulator Sim for $\mathcal{T}_b$ can be constructed in the experiment $\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID}}(n, \omega, b)$ such that (a) Sim does not have knowledge of the value b or any key k, and (b) $\mathcal{A}^\delta$'s interaction with Sim would be computationally indistinguishable from the one with $\mathcal{T}_b$. The probability that the adversary distinguishes Sim from $\mathcal{T}_b$ can be shown to be $p(n)/2^n$, where p is a polynomial in the security parameter n. The detailed security proof can be found in Weis et al. [140].

A potential issue associated with key search is the computation/storage cost for the reader, which is a linear function of the number of tags in the system. Based on Weis et al. [140], several variants of key-search RFID scheme have been proposed in the literature [5], aiming to enhance privacy and/or efficiency. Such improvement makes use of the following methods; (a) use of pre-computed table, (b) use of time-stamp, (c) use of tree-architecture, and (d) randomised internal secret update.

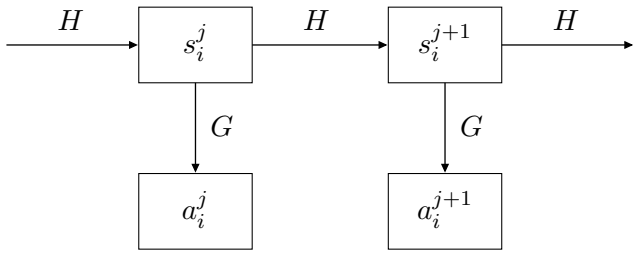


Figure 5.2: The action of tag $\mathcal{T}_i$ in the OSK scheme

In the following sections, we describe such variant schemes along with brief privacy analysis. The strategy of the formal security proof for such schemes is similar to Weis et al. [140], and we rather focus on analysing how security and/or efficiency improvements specifically affected privacy levels compared to the original key-search RFID scheme [140].

5.3.2 Use of pre-computed table (OSK/ADO)

Ohkubo, Suzuki, and Kinoshita (OSK) [108] propose a simple privacy-preserving tag identification scheme. Suppose the set of tag is $\{\mathcal{T}_i\}_{1\leq i\leq n}$. Suppose that a tag $\mathcal{T}_i$ with identifier id_i is initialised with a secret s_i^1, and let H and G be independent one-way hash functions. As shown in Figure 5.2, in the j-th transaction with a reader, $\mathcal{T}_i$ replies to a reader query with $a_i^j = G(s_i^j)$, and then updates its secret s_i^j to $s_i^{j+1} = H(s_i^j)$. A reader determines id_i using a pre-computed table containing $\{a_i^j, (id_i, j)\}_{1\leq i\leq n, 1\leq j\leq m}$ for rapid lookup, where n is the number of tags and m is a fixed upper bound on the number of times a tag can be queried before it needs to be re-initialised.

Hellman [67] studied the problem of searching for secret keys for symmetric ciphers. He considered the resource requirements for an attacker seeking to recover a secret key k from a ciphertext $c = e_k(m)$ on a predetermined message m. He showed how to reduce the computational effort involved in searching for a key from $O(n)$ to $O(n^{2/3})$ by constructing a pre-computed table known as a Helllman table, where n is the size of a key space.

Avoine, Dysli, and Oescslin (ADO) [7, 9] observe that the problem of table searching

Protocol: YA-TRIP

1.	$\mathcal{R} \longrightarrow \mathcal{T}$	:	T_r
2.	$\mathcal{T}$	:	if $T_r \leq T_t$ or $T_r > T_{\mathsf{max}}$, $H_r \leftarrow \mathsf{PRNG}_i^j$;
3.			else, $T_t \leftarrow T_r$ and $H_r \leftarrow \mathsf{HMAC}(k_i, T_t)$
4.	$\mathcal{T} \longrightarrow \mathcal{R}$	:	H_r
5.	$\mathcal{R}$	:	$s \leftarrow \mathsf{TableLookup}(T_r, H_r)$

Figure 5.3: The YA-TRIP Protocol [136]

task for readers in RFID systems using a key search approach is similar to the problem breaking keys, i.e. a reader needs to be able to determine the secret key k_i of a tag $\mathcal{T}_i$, in order to authenticate $\mathcal{T}_i$. They thus apply a variant of the Hellman technique to the OSK scheme, yielding a scheme with considerably reduced complexity of server table searching.

The OSK/ADO schemes are conjectured to provide (*universal*, *narrow*, *destructive*, ω)-privacy if G is a random oracle and H is collision-resistant. Ohkubo et al. [108] discuss that the schemes could provide privacy since it is infeasible to link a_i^j and a_i^{j+1} because of the assumed pre-image resistance of G. We, however, point out that the OSK/ADO schemes fail to provide privacy if the hash function H is not collision-resistant. For example, once two collided hash chains are found, e.g. $\{s_i^j\}_{j=1}^{\infty}$ and $\{s_{i'}^j\}_{j=1}^{\infty}$ say, and one of the chains is used in the system, privacy of the tag using the chain is not guaranteed.

As Weis et al. [140], the OSK/ADO schemes cannot satisfy privacy against *strong* adversary, since, once the adversary obtains the internal state s_i^j for $\mathcal{T}_i$ in the learning stage, it is possible to compute the all valid future a_i^j. The OSK/ADO schemes, however, also fail to provide *wide* privacy, since the adversary is able to determine the value b exploiting the upper bound m described by Juels and Weis [80]. The adversary may send the $\texttt{Execute}(\mathsf{protocol}, \mathcal{T})$ query m times to exhaust $\mathcal{T}$'s valid outputs, and submits $\mathcal{T}$ and any other tag in the challenge stage. By sending $\texttt{Execute}(\mathsf{protocol}, \mathcal{T}_b)$ and `Result` query, the adversary determine whether $\mathcal{T}$ is $\mathcal{T}_b$ or not.

5.3.3 Use of time-stamp (YA-TRIP/YA-TRIP*)

A number of schemes have been proposed which use monotonically increasing time-stamps, thus facilitating efficient key search via a pre-computed table for each time-stamp [33, 135, 136]. The YA-TRIP (Yet Another Trivial RFID Identification Protocol) scheme [136] was the first proposed protocol of this type (see Figure 5.3).

A reader $\mathcal{R}$ pre-computes a table containing entries of the form $\{k_i, \mathsf{MAC}(k_i, T_r)\}_{i,r}$ for all tag keys k_i and time-stamps T_r (for a pre-specified range of values r). The choice of the interval between time-stamps is thus an important factor in determining the table size. A relatively long interval, e.g. of an hour, would decrease the cost of computation and storage for a reader, but tags could only be identified once in any hour. On the other hand, a short interval, e.g. of a few seconds, would be more realistic, but increases the maintenance cost at the reader.

Each tag $\mathcal{T}_i$ is initialised with the triple $(k_i, T_0, T_{\mathsf{max}})$, where k_i serves as both a tag identifier and a secret key for $\mathcal{T}_i$, and T_0 and T_{max} are the initial and final possible time-stamps, respectively. We use T_t to denote the current time-stamp stored in $\mathcal{T}_i$. PRNG is a pseudo-random number generator which could, for example, be constructed using a cryptographic hash function, and PRNG_i^j denotes the j-th invocation of PRNG in tag $\mathcal{T}_i$.

To authenticate a tag $\mathcal{T}_i$, a reader $\mathcal{R}$ sends it the current time-stamp T_r. $\mathcal{T}_i$ responds with $\mathsf{MAC}(k_i, T_r)$ if $T_t < T_r \leq T_{\mathsf{max}}$, i.e. $\mathcal{T}_i$ ensures that that T_r is a fresh and valid time-stamp before computing and sending a response. Otherwise, $\mathcal{T}_i$ responds with PRNG_i^j. The algorithm $\mathsf{TableLookup}(T_r, H_r)$ returns k_i if there exists an entry $\big(k_i, \mathsf{MAC}(k_i, T_r)\big)$ such that $H_r = \mathsf{MAC}(k_i, T_r)$; otherwise, it returns $\perp$. T_r is not tag-specific and could be the real (current) time value.

However, tags do not have an internal clock, and thus can only check the validity of T_r by means of a comparison with the most recently received time-stamp. This causes significant security vulnerabilities as pointed out by Tsudik [135].

- YA-TRIP does not provide tag authentication,[9] since an adversary could easily impersonate a tag by querying it in advance to obtain pairs (T_r, H_r).

[9] Tsudik [136] first proposes this protocol as YA-TRAP, and later re-names it as YA-TRIP in [135].

Protocol: YA-TRIP*

1. $\mathcal{R} \longrightarrow \mathcal{T}$: T_r, w_r
2. $\mathcal{T}$: $\gamma \leftarrow T_r - T_t$; $\delta \leftarrow \lfloor T_r/\texttt{int} \rfloor - \lfloor T_t/\texttt{int} \rfloor$;
3. if $\gamma \leq 0$ or $T_r > T_{\mathsf{max}}$ or $H^{\delta}(w_r) \neq w_t$, then $H_r \leftarrow \mathsf{PRNG}_i^j$
3′. else, $T_t \leftarrow T_r$; $w_t \leftarrow w_r$; $H_r \leftarrow \mathsf{MAC}(k_i, T_t)$

4 & 5 are the same as YA-TRIP

Figure 5.4: The YA-TRIP* protocol [135]

- YA-TRIP is vulnerable to an obvious Denial of Service (DoS) attack; an adversary could send a tag T_r' such that $T_r' \gg T_r$ or $T_r' \approx T_{\mathsf{max}}$, and thereby incapacitate it either temporarily or permanently.

To mitigate the above DoS attack, Tsudik [135] introduced a modified version of the YA-TRIP scheme, which we refer to as the YA-TRIP* scheme (see Figure 5.4). The YA-TRIP* scheme incorporates a hash chain $\{w_i\}_{i=0}^{z}$, such that $w_j = H(w_{j+1})$ for $j = 0, 1, \ldots, z-1$, where $z = T_{\mathsf{max}}/\texttt{int}$ for the time interval `int` between values w_j. Each tag now stores the triple (T_t, w_t, k_i). The choice of the length of the time interval `int` has a direct effect on the degree of vulnerability to a DoS attack. That is, an adversary can disable tags for at most `int`, by intercepting (T_r, w_r) and sending (T_r', w_r), where T_r' is the maximum time-stamp such that $\lfloor T_r'/\texttt{int} \rfloor = \lfloor T_r/\texttt{int} \rfloor$. Too small a choice for `int`, however, will impose an excessive computational burden on both reader and tags.

The YA-TRIP provides (*universal*, *narrow*, *destructive*, ω)-privacy if the underlying MAC function is a random oracle. The scheme fails to provide *wide* privacy similarly to the OSK/ADO schemes, since the adversary is able to *mark* a tag by querying T_{max}. The tag sets $T_t \leftarrow T_{\mathsf{max}}$, and outputs random values for all future queries. The reader would be reject this *marked* tag in all future sessions, allowing the adversary to distinguish such a tag from *unmarked* tags.

On the other hand, the YA-TRIP* provides either (*universal*, *narrow*, *destructive*, ω)-privacy or (*discrete*-2`int`, *wide*, *destructive*, ω)-privacy. That is, the scheme provides privacy against the *wide* adversary, if the adversary cannot access a tag more than once during the time interval 2×`int`. By adopting the hash chain for reader authentication, the attack of *marking* the target tags only works during the

time internal `int`.

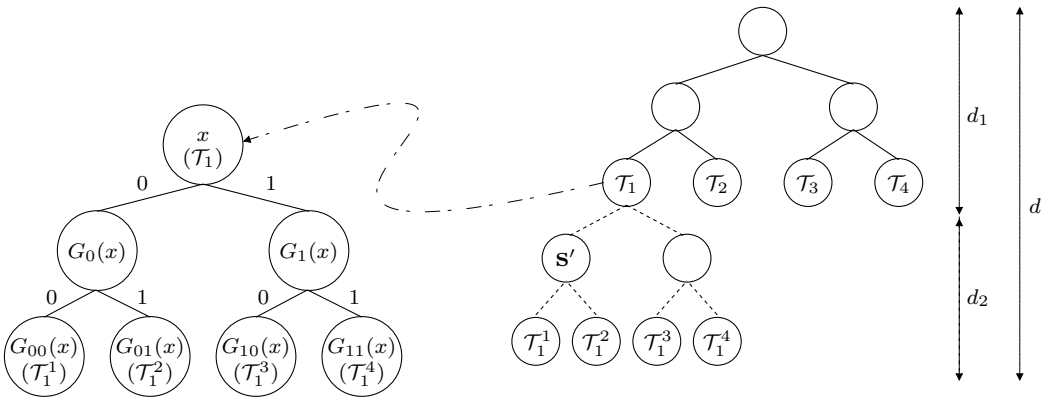
5.3.4 Tree-based approach (MW/MSW)

Molnar and Wagner (MW) [106] propose an RFID system in which each tag contains multiple secret keys. These keys are arranged hierarchically as defined by a tree T, where every node other than the root is associated with a unique key and each tag is assigned to a unique leaf. Each tag stores the keys associated with each of the nodes on the path from the root to its leaf. A reader can authenticate a tag using its stored secret keys.

That is, the reader runs several rounds of an authentication protocol, e.g. the protocol of Weis et al. [140], where the tag uses a different shared key in each round, starting with the key for the root of a tree T and successively using keys on the path to the leaf corresponding to the tag. This means that, in each round, the reader only needs to search through the keys immediately adjacent (in T) to the key used in the previous round, thereby significantly reducing the size of the key search performed by the reader.

For example, if T has depth d and branching factor b, then each tag contains d keys and the scheme can accommodate up to b^d tags in total. The reader is able to authenticate a tag after searching through at most db keys. The brute-force key search, however, would require a reader to search the complete space of b^d tag keys.

Molnar, Soppera, and Wagner (MSW) [105] further develop the MW scheme, introducing the notion of time-limited delegation. Unlike the MW scheme, the MSW scheme uses a *binary* tree of secrets with depth $d = d_1 + d_2$, as shown in Figure 5.5. The nodes at each of the first d_1 levels of the tree are assigned secrets that are chosen uniformly at random by the Trusted Centre (TC), and each node at depth d_1 corresponds to a unique tag. The next d_2 levels of the tree contains node secrets that are derived from the secret stored in the node at depth d_1 (e.g. the secret x in the case of the tag $\mathcal{T}_1$ in Figure 5.5) using the Goldreich-Goldwasser-Micali (GGM) construction [59], and each tag keeps a counter c that identifies a leaf at level d in the tree. We give a formal description below.



The nodes connected with solid edges correspond to secrets shared between tags $\mathcal{T}_1, \ldots, \mathcal{T}_4$ and the TC, while the nodes with dashed edges correspond to secrets which are generated using the GGM construction from the secret at their immediate parent. On each read, a tag increases its counter and updates its state to use the secrets in the next leaf. For example, a tag $\mathcal{T}_1$ first uses the secrets from the root to the leaf $\mathcal{T}_1^1$, and when queried next it uses the secrets from the root to the leaf $\mathcal{T}_1^2$. If a reader is delegated to identify a tag $\mathcal{T}_1$ during the interval $[1, 2]$, it is provided with the secret $G_0(x)$ associated with node $\mathbf{s}'$. This reader can then identify tag $\mathcal{T}_1$ only twice using the secrets $G_0(G_0(x)) = G_{00}(x)$ and $G_1(G_0(x)) = G_{01}(x)$.

Figure 5.5: A toy example of the tree of secrets (MSW)

Let $\{0,1\}^l$ be a set of binary strings of length l. A node s of depth e is labelled with a binary string of length e defined by the unique path form the root to that node. For example, in Figure 5.5 the node $\mathbf{s}'$ is labelled with 000. A value $h(s)$ denotes the key associated with node s. The TC defines a function $h : \{0,1\}^{\leq d} \rightarrow \{0,1\}^k$ for a security parameter k as follows:

- $h : \{0,1\}^{\leq d_1} \rightarrow \{0,1\}^k$ is chosen uniformly at random; and
- $h : \{0,1\}^{>d_1} \rightarrow \{0,1\}^k$ is defined as $h(s||b) = G_b(h(s))$ for all $s \in \{0,1\}^{\geq d_1}$ and $b \in \{0,1\}$,

where $G : \{0,1\}^k \rightarrow \{0,1\}^{2k}$ is a pseudo-random number generator, and $G_0(s)$ and $G_1(s)$ are the k MSBs and k LSBs of $G(s)$, respectively. By using the GGM construction [59], both tags and the TC can reduce key storage space, i.e. they only need to store keys up to d_1 levels in the tree.

A tag responds to a reader query by generating a random number r and sending the

following pair of values back to the reader

$$\big(r, p\big) = \big(r, (p_1, p_2, \ldots, p_d)\big) = \big(r, (F_{h(c_{1..1})}(r), F_{h(c_{1..2})}(r), \ldots, F_{h(c_{1..d})}(r))\big). \quad (5.1)$$

F denotes a pseudo-random function, and $c_{1..i}$ $(1 \leq i \leq d)$ denotes the sequence of nodes on the path in the tree of secrets from the root to the tag's current leaf $\mathcal{T}^c$. Finally, the tag increases its counter c, i.e. it sets $c \leftarrow c + 1$. Note that each leaf is assigned with a counter c such that $1 \leq c \leq 2^d$, and that each tag can update its counter at most $2^{d_2} - 1$ times.

On receipt of (r, p), the reader can conduct a depth-first search to find a path in the tree that matches p, as in the MW scheme. That is, at node s with a depth i, the reader can check whether the left child $s||0$ or the right child $s||1$ matches an entry p_{i+1} by checking whether $F_{h(s||0)}(r) = p_{i+1}$ or $F_{h(s||1)}(r) = p_{i+1}$.

The MSW scheme also enables delegation of the ability to identify a tag for a limited period of time, i.e. for a specific number of read operations. Suppose that the TC decides to delegate the right to access tag $\mathcal{T}$ to a reader $\mathcal{R}$ for the interval $[L, R]$, where $1 \leq L \leq R \leq 2^d$. The TC then sends $\mathcal{R}$ the secret $h(s^*)$ of the node $s^* \in \{0,1\}^{\geq d_1}$, where $s^*||00\cdots0$ of depth d has a counter L and $s^*||11\cdots1$ of depth d has a counter R. It is straightforward to verify that, for any node s of depth d satisfying that its counter is within $[L, R]$, there exists an s^* such that s^* is a prefix of s. $\mathcal{R}$ is thus able to compute the secrets $h(s)$ for $|s^*| \leq |s| \leq d$ using the GGM construction, where $|s|$ and $|s^*|$ denote the bit-lengths of s and s^*, respectively. Also, given (r, p) as a tag response, a reader can identify the tag $R - L + 1$ times by checking the entries $p_{|s|}, \ldots, p_{d-1}, p_d$ of p.

Such a delegation can be useful in a variety of scenarios. For example, hand-held readers may be given temporary scanning privileges within systems with intermittent connectivity. In the strict sense, the scheme does not provide transfer of ownership, but enables the distribution of access control rights by the TC.

There is a price to be paid for the efficiency of key search in tree-based approach. First, tags must store $\lceil \log_b n \rceil$ keys, where n is the number of tags and b is the branching factor, and must also perform $\lceil \log_b n \rceil$ protocol executions. Furthermore, since the tree structure creates an overlap among the sets of keys, compromising one tag or a small number of tags can lead to significant privacy infringements for other tags, as analysed in [7, 107].

More specifically, the MW/MSW schemes fails to provide *destructive* privacy, conjectured to only satisfying (*universal*, *wide*, *weak*, ω)-privacy if the underlying pseudo-random functions are random oracles. We give a simple example that the *destructive* adversary wins the privacy experiment $\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^{\delta},\mathsf{MSW}}$. The adversary corrupts a tag $\mathcal{T}^*$ and then tries to find two tags, $\mathcal{T}_0$ and $\mathcal{T}_1$, where one of the two tags, $\mathcal{T}_0$ say, has the same c_1 as $\mathcal{T}^*$ and the other tag, $\mathcal{T}_1$ say, does not. Finding such tags become possible because $\mathcal{T}^*$ and $\mathcal{T}_0$ will respond with same p_1 given the same r in the equation (5.1). By selecting $\mathcal{T}_0$ and $\mathcal{T}_1$ in the step (3) of $\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^{\delta},\mathsf{RFID}}$, the adversary is able to output the correct guess bit d.

5.3.5 The Lim-Kwon (LK) scheme

Lim and Kwon (LK) [97] propose a modified version of the OSK scheme, realising tag authentication, forward secrecy, and secure ownership transfer. In the LK scheme, forward secrecy and secure ownership transfer become possible by evolving the tag secrets during every protocol execution. More specifically, they use the term refresh to mean probabilistic evolution, and update to mean deterministic evolution. If the protocol is successfully completed, the tag and backend database refresh the tag secret probabilistically using exchanged random numbers; otherwise, the tag updates its secret deterministically, as in the OSK scheme. The 'refresh' process makes secure ownership transfer possible. The LK scheme's use of a pre-computed table is more efficient than that of the OSK scheme, as described below.

Parameters. We first define the parameters used in the scheme: m denotes the maximum number of allowable authentication failures between two valid sessions; n denotes the length of the backward key chain used for server authentication; l denotes the bit-length of a tag secret; l_1 denotes the bit-length of random challenges; l_2 denotes the bit-length of the tag secret transmitted in clear when the server identifies a tag ($l_2 \leq l$). The following functions are used in the protocol:

- $f : \{0,1\}^l \times \{0,1\}^{2l_2} \rightarrow \{0,1\}^{2l_1}$, which is used to generate authenticators;
- $g : \{0,1\}^l \rightarrow \{0,1\}^l$, which is used to generate the forward key chain;
- $h : \{0,1\}^{2l_1} \rightarrow \{0,1\}^{2l_1}$, which is used to generate the backward key chain;

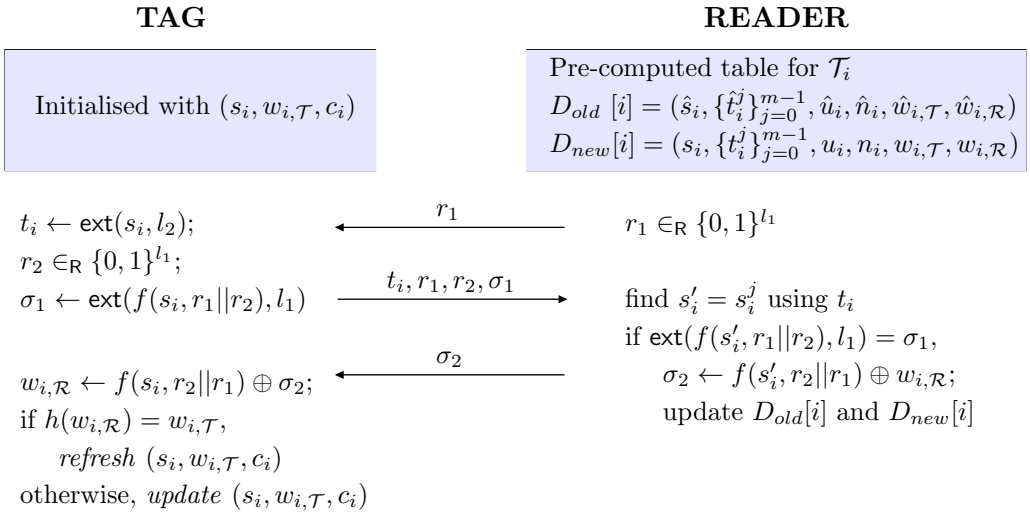


Figure 5.6: The LK authentication protocol

- $\mathsf{ext}(x, l^*)$ denotes a simple extract function that returns l^* bits of x, e.g. $x \bmod 2^{l^*}$.

The functions f, g, and h are pseudo-random functions and can be constructed from a single lightweight block cipher [97].

Initialisation. Tag $\mathcal{T}_i$ is initialised by a reader $\mathcal{R}$ in the following way.

- $\mathcal{R}$ chooses a secret $s_i \in_R \{0,1\}^l$ for $\mathcal{T}_i$, and computes $\{s_i^j\}_{j=1}^{m-1}$ and $\{t_i^j\}_{j=1}^{m-1}$, where $s_i^0 = s_i$, $s_i^j = g(s_i^{j-1})$, and $t_i^j = \mathsf{ext}(s_i^j, l_2)$.
- $\mathcal{R}$ also chooses $u_i \in_R \{0,1\}^{l_1}$ for each $\mathcal{T}_i$, and computes a hash chain $\{w_i^j\}_{j=0}^{n-1}$, of length n, where $w_i^n = u_i$ and $w_i^j = h(w_i^{j+1})$ for $0 \leq j < n$.
- $\mathcal{T}_i$ stores $(s_i, w_{i,\mathcal{T}})$, where $w_{i,\mathcal{T}} = w_i^0$, and initialises the failure counter c_i to 0.
- $\mathcal{R}$ makes the following entries for $\mathcal{T}_i$ in its tag database, $D[i] = D_{old}[i] \cup D_{new}[i]$; $D_{old}[i]$ is initially empty and $D_{new}[i] = \big(s_i, \{t_i^j\}_{j=0}^{m-1}, u_i, n_i, w_{i,\mathcal{T}}, w_{i,\mathcal{R}}\big)$, where $w_{i,\mathcal{R}} = w_i^1$ and $n_i = n$. Note that $w_{i,\mathcal{T}} = h(w_{i,\mathcal{R}})$.

Protocol. The protocol is summarised in Figure 5.6. $\mathcal{R}$ maintains two tag databases $D_{old}[i]$ and $D_{new}[i]$. The retention of *old* tag data prevents $\mathcal{T}_i$ becoming desynchronised if it does not receive the last protocol message σ_2, e.g. because of a commu-

nication failure. The authentication protocol is performed between $\mathcal{T}_i$ and $\mathcal{R}$ as follows.

1. $\mathcal{R}$ chooses $r_1 \in_{\mathsf{R}} \{0,1\}^{l_1}$ and sends it to $\mathcal{T}_i$.

2. $\mathcal{T}_i$ chooses $r_2 \in_{\mathsf{R}} \{0,1\}^{l_1}$, computes $t_i \leftarrow \mathsf{ext}(s_i, l_2)$ and $\sigma_1 \leftarrow \mathsf{ext}(f(s_i, r_1||r_2), l_1)$, and send $(t_i, r_1, r_2, \sigma_1)$ to $\mathcal{R}$.

3. $\mathcal{R}$ searches its database to find an entry containing t_i. If no match is found, $\mathcal{R}$ responds with $\sigma_2 = \perp$ (denoting 'failure') and stops; otherwise, i.e. if there exists $D[i]$ containing $\left(s_i, \{t_i^j\}_{j=0}^{m-1}, u_i, n_i, w_{i,\mathcal{T}}, w_{i,\mathcal{S}}\right)$ such that $t_i^j = t_i$ for some j, $\mathcal{R}$ computes $s_i' = g^j(s_i)$ and checks that $\mathsf{ext}(f(s_i', r_1||r_2), l_1) = \sigma_1$. If this checks fails, $\mathcal{R}$ responds with $\sigma_2 = \perp$ and stops; otherwise, i.e. if the check succeeds, $\mathcal{R}$ computes $\sigma_2 \leftarrow f(s_i', (r_2||r_1)) \oplus w_{i,\mathcal{R}}$, sends σ_2 to $\mathcal{T}_i$, and updates $D[i]$ as follows:

   (a) $D_{old}[i]$ is updated to $\left(\hat{s}_i, \{\hat{t}_i^j\}_{j=0}^{m-1}, \hat{u}_i, \hat{n}_i, \hat{w}_{i,\mathcal{T}}, \hat{w}_{i,\mathcal{R}}\right)$, where $\hat{s}_i = g(s_i')$, $\hat{t}_i^k = \mathsf{ext}(g^k(\hat{s}_i), l_2)$ for $0 \le k \le m-1$, and $\hat{u}_i \leftarrow u_i, \hat{n}_i \leftarrow n_i, \hat{w}_{i,\mathcal{T}} \leftarrow w_{i,\mathcal{T}}, \hat{w}_{i,\mathcal{R}} \leftarrow w_{i,\mathcal{R}}$.

   (b) $D_{new}[i]$ is updated to $\left(s_i, \{t_i^j\}_{j=0}^{m-1}, u_i, n_i, w_{i,\mathcal{T}}, w_{i,\mathcal{R}}\right)$, where $s_i \leftarrow g(s_i' \oplus (w_{i,\mathcal{R}}||r_1||r_2))$, $t_i^j = \mathsf{ext}(g^j(s_i), l_2)$ for $0 \le k \le m-1$, $n_i \leftarrow n_i - 1$, $w_{i,\mathcal{T}} \leftarrow w_{i,\mathcal{R}}$ and $w_{i,\mathcal{R}} \leftarrow h^{n_i}(u_i)$, and u_i and n_i are unchanged.

4. $\mathcal{T}_i$ computes $w_{i,\mathcal{R}}' = \sigma_2 \oplus f(s_i, r_2||r_1)$ and checks whether $h(w_{i,\mathcal{R}}') = w_{i,\mathcal{T}}$. If the check succeeds, then $\mathcal{T}_i$ sets $c_i = 0$ and refreshes $(s_i, w_{i,\mathcal{T}})$ as follows: $w_{i,\mathcal{T}} \leftarrow w_{i,\mathcal{R}}'$ and $s_i \leftarrow g(s_i \oplus (w_{i,\mathcal{T}}||r_1||r_2))$. If the check fails, then: if $c_i < m$ then $\mathcal{T}_i$ increments the failure counter, i.e. sets $c_i \leftarrow c_i + 1$ and updates s_i as $s_i \leftarrow g(s_i)$;[10] alternatively, if $c_i \ge m$ then $\mathcal{R}$ does nothing.

The LK scheme provides either (*universal*, *wide*, *destructive*, m)-privacy or (*diescrete*, *wide*, *strong*, m)-privacy if the underlying pseudo-random functions are random oracles. We observe that the scheme fails to provide privacy if the adversary can make more than m oracle queries, since the tag does not update the secret s_i once the tag reaches the upper bound for the authentication failures; in such a case a tag

[10] Once s_i is updated, the tag will emit the updated t_i in the subsequent session, where $t_i = \mathsf{ext}(s_i, l_2)$ and s_i is the updated version.

will emit a static identifier t_i. To minimise the probability of the described privacy infringes, a relatively large m could be used in practice.

The parameter m determines the size of the pre-computed table held by the reader, like the upper bound m in the OSK scheme. The size of the parameter m in the LK scheme, however, can typically be considerably smaller than the corresponding parameter in the OSK scheme. This is because m in the LK scheme is an upper bound on the number of permitted protocol failures between two successful protocol executions for a tag, whereas m in the OSK scheme defines the maximum number of protocol executions in the entire tag life cycle.

Differently from the previous key-search schemes, the LK scheme provides *strong* privacy given that the adversarial access type is *discrete*. This is because the protocol provides a measure of protection even if the tag secret $(s_i, w_{i,\mathcal{T}})$ is compromised. Such secret becomes useless when the compromised tag engages the successful run of protocol outside the eavesdropping range of the adversary; the tag refreshes the internal secrets, i.e. $w_{i,\mathcal{T}} \leftarrow w'_{i,\mathcal{R}}$ and $s_i \leftarrow g(s_i \oplus (w_{i,\mathcal{T}}||r_1||r_2))$. That is, the adversary cannot determine the updated value $w_{i,\mathcal{T}}$ provided that the constructed hash function h satisfies the one-wayness, and also the updated value s_i because the adversary has no knowledge of the values r_1 and r_2.

Apart from privacy, Lim and Kown [97] claim that the protocol provides a measure of *strong authentication* and *secure ownership transfer*. A compromised tag secret could be used to construct a counterfeit tag. Suppose that a fake tag $\tilde{\mathcal{T}}_i$ is constructed by equipping it with the secret $(s_i, w_{i,\mathcal{T}})$. The fake tag $\tilde{\mathcal{T}}_i$ becomes useless as soon as the genuine tag $\mathcal{T}_i$ engages in the protocol with $\mathcal{R}$ causing the secret to be refreshed. Unfortunately, it is also true that the genuine tag $\mathcal{T}_i$ will become permanently desynchronised if the fake tag $\tilde{\mathcal{T}}_i$ engages in the protocol with the reader.

Ownership of a tag $\mathcal{T}_i$ can be securely transferred from A to B in the following way. The reader information for the tag, i.e. D_i $(= D_{old}[i] \cup D_{new}[i])$, is sent to B via a secure channel, and B performs the above protocol with $\mathcal{T}_i$ in an environment in which A cannot eavesdrop upon the exchanged messages. As a result, the secret key in $\mathcal{T}_i$ and the secret tag record D_i held by B will both be refreshed. The previous owner then can no longer identify $\mathcal{T}_i$.

5.4 Public key Cryptographic Solutions

Use of public key cryptography can securely address privacy issues in RFID systems as discussed in the literatures [85, 91, 99, 100, 141]. Unfortunately, implementing public-key primitives is beyond the computational capability of passive RFID tags. External agents, however, could refresh the pseudonym resident in a tag frequently enough to ensure privacy.

A number of refresh-based approaches have been proposed which support the use of public key primitives in an RFID system, where a tag delegates expensive cryptographic operations to a reader. Such refresh-based approaches have been discussed by a number of authors [2, 62, 79], in which public key encryption is used to support refresh techniques, i.e. tags store data in encrypted form which is subsequently refreshed (i.e. re-encrypted) by more powerful readers without requiring any cryptographic functionality in a tag.

In this section we describe Juels and Pappu (JP) [79], which first proposed to use public key encryption as a refresh technique in RFID systems. We then investigate other refresh-based schemes of Golle et al. [62] and Ateniese et al. [2], which suggest the methods of refreshing ciphertexts when multiple public keys are used in the system.

5.4.1 Juels-Pappu scheme (JP)

The European Central Bank planned to embed RFID tags into banknotes in order to prevent forgeries, and to also provide tracking mechanism for use by law enforcement agencies.[11] Embedding a tag in a banknote, however, may infringe individual privacy when improperly deployed [79]. To address this issue Juels and Pappu [79] proposed a cryptographic banknote protection scheme, which both protects against counterfeiting and enables banknote tracking by a law enforcement agency, whilst protecting the privacy of banknote bearers.

More specifically, Juels and Pappu [79] first proposed use of the refresh approach to enhance privacy of RFID-enabled banknotes. The authors proposed storing an

[11] See http://www.eetimes.com/story/OEG20011219S0016.

encrypted version of a banknote's serial number in an RFID tag embedded in the banknote. To prevent tracking, the ciphertext is periodically re-encrypted by readers programmed with the public key system, thereby rendering multiple interactions with the same RFID tag unlinkable. Juels and Pappu also proposed that the banknote should carry optical write-access keys, to prevent unauthorised writing to a tag. Thus a reader must scan this optical key before re-encrypting a ciphertext.

'Re-encryption' here means transforming a ciphertext c into a new unlinkable ciphertext c' using a public key pk, without changing the underlying plaintext, i.e. c and c' decrypt to the same plaintext under the system private key sk. This is different from other uses of the term re-encryption, such as proxy re-encryption, which involves generating a new ciphertext that is an encrypted version of the original plaintext under a different public key.

We now describe Juels and Pappu's scheme [79], and first describe the entities involved.

- *Central bank* ($\mathcal{B}$): The central bank creates banknotes and wishes to prevent banknote forgery.
- *Law enforcement agency* ($\mathcal{L}$): This agency wishes to trace flows of banknotes, and in addition wishes to efficiently detect counterfeit banknotes with high assurance.
- *Merchant* ($\mathcal{M}$): The merchant is an entity that handles banknotes, accepting them for payment and possibly anonymising them to help protect client privacy. Most merchants will comply with requirements of the law enforcement agency by reporting irregularities in banknote data and helping to protect client privacy; however, some merchants may attempt to compromise the privacy of their clients.
- *Consumers* ($\mathcal{C}$): The consumers are banknote bearers, and they may wish to protect their privacy. They may attempt to corrupt the information stored in the banknote in order to avoid tracing by $\mathcal{L}$.

Juels and Pappu [79] next define the RFID-enabled banknote scheme.

Definition 5.4 (JP Scheme) *Suppose that PE=(EKG,Enc,Dec) is a public key encryption scheme, as in definition 2.12, and DS=(SKG,Sig,Ver) is a digital signature scheme, as in definition 2.18.*[12] *The JP scheme consists of the following procedures.*

- *Setup: Given a security parameter τ, a central bank $\mathcal{B}$ and a law enforcement agency $\mathcal{L}$ generate their own public/private key pairs $(PK_{\mathcal{B}}, SK_{\mathcal{B}}) \leftarrow \mathsf{SKG}(\tau)$ and $(PK_{\mathcal{L}}, SK_{\mathcal{L}}) \leftarrow \mathsf{EKG}(\tau)$, respectively. The public keys, i.e. $PK_{\mathcal{B}}$ and $PK_{\mathcal{L}}$, and a collision-resistant hash function $h : \{0,1\}^* \rightarrow \{0,1\}^k$ are published, where k is chosen appropriately.*

- *Banknote creation: For every banknote i, $\mathcal{B}$ selects a unique serial number S_i and computes the signature $\Sigma_i = \mathsf{Sig}(SK_{\mathcal{B}}, S_i||D_i)$, where D_i is the denomination (value) of the banknote. $\mathcal{B}$ then generates an access key $A_i = h(\Sigma_i)$, and prints S_i and Σ_i as an optical barcode on the banknote.*[13] *$\mathcal{B}$ then generates a random number $r_i \in_R \{0,1\}^l$ for an appropriately chosen l, and computes*

$$C_i = \mathsf{Enc}(PK_{\mathcal{L}}, \Sigma_i||S_i; r_i),$$

  where r_i is written into the δ-cell memory and C_i is written into the γ-cell memory, as indicated in Figure 5.7.

- *Banknote verification and anonymisation: On receiving a banknote i, a merchant $\mathcal{M}$ first verifies the stored data and then re-encrypts it, using the following procedure.*

  1. *$\mathcal{M}$ reads the data S_i and Σ_i printed on the banknote, and computes $A_i = h(\Sigma_i)$.*
  2. *$\mathcal{M}$ reads C_i from γ-cell, and keyed-reads r_i from δ-cell using A_i.*
  3. *$\mathcal{M}$ checks if $C_i = \mathsf{Enc}(PK_{\mathcal{L}}, \Sigma_i||S_i; r_i)$.*
  4. *$\mathcal{M}$ chooses a random number r_i' and key-writes it into δ-cell.*
  5. *$\mathcal{M}$ computes $C_i' = \mathsf{Enc}(PK_{\mathcal{L}}, \Sigma_i||S_i; r_i')$ and keyed-writes it into γ-cell.*

  If any of the above steps fails, then the merchant reports to $\mathcal{L}$.

[12] We use notations EKG and SKG to denote key generation algorithms for public key encryption scheme and digital signature scheme, respectively.

[13] Juels and Pappu [79] state that the access key A_i should be derived by hashing Σ_i rather than S_i, in order to avoid an attack in which an adversary computes the access-key by guessing the serial number without needing to see the banknote.

- *Banknote tracking: $\mathcal{L}$ obtains a ciphertext C_i by reading γ-cell in the banknote, and recovers the plaintext by computing $\Sigma_i||S_i = \mathsf{Dec}(SK_{\mathcal{L}}, C_i)$. $\mathcal{L}$ then checks if Σ_i is a valid signature on S_i by seeing $\mathsf{Ver}(PK_{\mathcal{B}}, \Sigma_i, S_i||D_i) = 1$. If Σ_i is valid, then $\mathcal{L}$ obtains the banknote serial number S_i.*

While the security of a conventional banknote relies on visible features (possibly including features only visible under an ultraviolet source), Juels and Pappu use both optical and electronic features, summarised as in Figure 5.7. *Optical* data is encoded in a human-readable form and/or in a machine-readable two-dimensional barcode. *Electronic* data is stored in an RFID tag, whose memory consists of two types of cell; universally-readable/keyed-writable γ-cell and keyed-readable/keyed-writable δ-cell (see Figure 5.7).[14]

Given the entities and definition of the scheme, Juels and Pappu [79] describe the security goal of the proposed scheme as follows.

- *Consumer privacy*: Only the law enforcement agency $\mathcal{L}$ should be able to track banknotes; even central bank $\mathcal{B}$ should not be able to track banknotes.
- *Strong tracking*: The law enforcement agency $\mathcal{L}$ should be able to identify and track a banknote even without visually inspecting it.
- *Forgery resistance*: A forger should not be able to create a new banknote with a previously unseen serial number; the forger should also not be able to alter the denomination of a banknote.
- *Fraud detection*: If invalid information is written to the RFID tag in a banknote, this should be readily detectable by a merchant $\mathcal{M}$.

For the specific construction of the JP scheme, which we call the Juels-Pappu scheme, Juels and Pappu choose ElGamal encryption scheme [40] and the Boneh-Shacham-Lynn signature scheme [20] as cryptographic algorithms. Let ElGmal encryption is defined as a tuple (Gen,Enc,Dec), as in definition 2.17, where for $r \in_{\mathsf{R}} \mathbb{Z}_q$ and $PK_{\mathcal{L}} = y$,

$$\mathsf{Enc}(PK_{\mathcal{L}}, m; r) = (my^r, g^r).$$

[14] The terms *keyed-read/write* mean that the reader could read or write into the tag only if sending the correct keys along with the read/write commands.

RFID Tag	
γ-cell (universally-readable/keyed-writable)	δ-cell (keyed-readable/keyed-writable)
$C = \mathsf{Enc}(PK_L, \Sigma \| \| S; r)$	r

Optical	
S	$\Sigma = \mathsf{Sig}(SK_{\mathcal{B}}, S \| \| D)$

Figure 5.7: RFID-enabled banknote data

Since ElGamal encryption scheme is not CCA secure, Juels and Pappu propose to use the secure integration method of Fujisaki and Okamoto [51].[15]

More specifically, for $r \in_{\mathsf{R}} \mathbb{Z}_q$, Juels and Pappu define

$$\mathsf{Enc}^*(PK_{\mathcal{L}}, m; r) = \big(\mathsf{Enc}(PK_{\mathcal{L}}, r; h_1(r||m)), h_2(r) \oplus m\big),$$

where h_1 and h_2 are hash functions from $\{0,1\}^*$ to $\{0,1\}^n$ $(n = |m|)$.

The JP scheme, however, cannot satisfy any type of privacy defined in the Definition 5.3 as discussed by Avoine [6]. First, the static access key could be used to track it when each banknote has a unique access key. This is because the adversary can readily obtain a tag-access key by making the `Send-Tag` query and a banknote tag will only respond if the access-key sent by a reader is valid. An adversary could thus track the banknote with a particular access key, by attempting to read the δ-cell memory of any banknote.

Even if such access keys are not unique across banknotes, the adversary can distinguish every banknote by making data recovery attack [6]. The adversary can readily obtain a random number r for each banknote by making the `Execute` query. In such a case, the integration method of Fujisaki and Okamoto becomes insecure, since, given

$$\mathsf{Enc}^*(PK_{\mathcal{L}}, m; r) = (\omega_1, \omega_2, \omega_3) = \big(ry^{h_1(r||m)}, g^{h_1(r||m)}, h_2(r) \oplus m\big),$$

the adversary can recover the plaintext by simply computing $m := \omega_3 \oplus h_2(r)$, since $m = \Sigma || S$.

[15] Although the authors propose to use a specific method, any IND-CCA2 secure public key encryption scheme could be used.

5.4.2 Universal re-encryption (UR)

While a single key pair may suffice for the RFID-enabled banknote scheme, multiple public keys are likely to be necessary in other RFID systems. In such a scenario, in order to randomise the ciphertext within an RFID tag, it is necessary to know under which public key the ciphertext has been encrypted. However, including a public key on a tag along with the ciphertext would itself permit a certain degree of tracking and profiling, because the public key could be used as a static identifier. To solve this anonymity problem, Golle et al. [62] introduced a cryptographic technique known as universal re-encryption, which permits re-encryption without any need to know which public key was used to encrypt the ciphertext.

The universal re-encryption makes use of the ElGamal encryption scheme. Let ElGmal encryption be defined as a tuple (Gen,Enc,Dec), as in definition 2.17. The output of an encryption operation for a message m consists of a pair made up of an ElGamal encryption of m together with an ElGamal encryption of 1. That is, the output of the encryption operation is:

$$\big(\mathsf{Enc}(y,m),\mathsf{Enc}(y,1)\big) = \big((my^{k_0},g^{k_0}),(y^{k_1},g^{k_1})\big),$$

for the public/private key pair $(\mathbb{G},q,g,y)$ and $(\mathbb{G},q,g,x)$. Universal re-encryption then operates as follows. Randomly choose $r' = (k'_0,k'_1) \in_{\mathsf{R}} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$, randomise $\mathsf{Enc}(y,1)$ by computing $\mathsf{Enc}'(y,1) = (y^{k_1k'_1}, g^{k_1k'_1})$, and then use $\mathsf{Enc}(y,1)$ to conceal $\mathsf{Enc}(y,m)$, exploiting the homomorphic property of ElGamal. That is, compute

$$\mathsf{Enc}'(y,m) = (my^{k_0}y^{k_1k'_0}, g^{k_0}g^{k_1k'_0}) = (my^{k_0+k_1k'_0}, g^{k_0+k_1k'_0}).$$

The re-encrypted ciphertext is then the pair $\big(\mathsf{Enc}'(y,m),\mathsf{Enc}'(y,1)\big)$, where $\mathsf{Enc}'(y,m)$ decrypts to the message m using the private key x.

Definitions

We now give the definition of Golle et al.'s universal re-encryption scheme [62], which we call the UR scheme.

Definition 5.5 (UR Scheme) *The UR scheme is a tuple of polynomial-time algorithms* (UK,UE,UD,URe) *with the following properties:*

- *The key generation algorithm UK takes as input a security parameter* 1^τ*, and returns a public/private key pair* (pk, sk)*. We write* $(pk, sk) \leftarrow \mathsf{UK}(1^\tau)$*.*
- *The encryption algorithm UE takes as input a public key* pk *and a plaintext* m*, and returns a ciphertext* c*. We write* $c \leftarrow \mathsf{UE}(pk, m)$*.*
- *The decryption algorithm UD takes as input a secret key* sk *and a ciphertext* c*, and outputs a message* m *or a special symbol* $\perp$ *denoting failure. We write* $m \leftarrow \mathsf{UD}(sk, c)$*.*
- *The re-encryption algorithm URe takes as input the public parameter and a ciphertext* c*, and returns* c' *such that both* c *and* c' *decrypt to the same plaintext* m*. We write* $c' \leftarrow \mathsf{URe}(c)$*.*

In order to model the security of the UR scheme, Golle et al. [62] also introduce a new security concept, universal semantic security under re-encryption (USSR). Given the UR = (UG,UK,UE,URe,UD) scheme and an adversary $\mathcal{B}$, we consider the following experiment.

Experiment $\mathbf{Exp}^{\mathsf{ussr}}_{\mathsf{UR},\mathcal{B}}(\tau)$

1. $\mathsf{UK}(1^\tau)$ is run to obtain key pairs (pk_i, sk_i) for $i = 0, 1$.
2. Given (pk_0, pk_1), $\mathcal{B}$ outputs ciphertexts (c_0, c_1).[16]
3. A random bit $b \in_{\mathsf{R}} \{0, 1\}$ is chosen, and a ciphertext $c' \leftarrow \mathsf{URe}(c)$ is computed. Finally, the ciphertext c' is given to $\mathcal{B}$.
4. $\mathcal{B}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

Golle et al. give the following definition [62].

Definition 5.6 (USSR) *We say that the UR scheme has universal semantic security under re-encryption (USSR) if, for any probabilistic polynomial-time adversary* $\mathcal{B}$*, a function* $\left|\Pr[\mathbf{Exp}^{\mathsf{ussr}}_{\mathcal{B},\mathsf{UR}}(\tau) = 1] - \frac{1}{2}\right|$ *is negligible.*

Golle et al. [62] construct the example scheme of the UR scheme using ElGamal encryption scheme as follows.

[16] The corresponding plaintexts must be in the plaintext space associated with the public keys.

Definition 5.7 (GJJS Scheme) *The scheme is a tuple of polynomial-time algorithms* (UK,UE,UD,URe) *with the following properties:*

- *The key generation algorithm UK takes as input a security parameter* 1^τ*, and returns a public/private key pair* $(pk, sk) = \big((\mathbb{G}, q, g, y), (\mathbb{G}, q, g, x)\big)$*, where* $x \in_R \mathbb{Z}_q^*$ *and* $y = g^x$*. We write* $(pk, sk) \leftarrow \mathsf{UK}(1^\tau)$*.*

- *The encryption algorithm UE takes as input a public key pk and a plaintext* $m \in \mathbb{G}$*, and returns a ciphertext*

$$c = \big((\alpha_0, \beta_0), (\alpha_1, \beta_1)\big) = \big((my^{k_0}, g^{k_0}), (y^{k_1}, g^{k_1})\big),$$

  where $r = (k_0, k_1) \in_R \mathbb{Z}_q^* \times \mathbb{Z}_q^*$*. We write* $c \leftarrow \mathsf{UE}(pk, m)$*.*

- *The decryption algorithm UD takes as input a secret key sk and a ciphertext* $c = \big((\alpha_0, \beta_0), (\alpha_1, \beta_1)\big)$*. It first checks whether* $\alpha_k, \beta_k \in \mathbb{G}$ *for* $k = 0, 1$*; if not, it returns a special symbol* $\perp$ *indicating that the ciphertext is invalid. It then checks whether* $\alpha_1/\beta_1^x = 1$*. If so, the algorithm outputs* $m := \alpha_0/\beta_0^x$*; otherwise the decryption fails and the output is* $\perp$*. We write* $m \leftarrow \mathsf{UD}(sk, c)$*.*

- *The re-encryption algorithm URe takes as input* $(\mathbb{G}, p, g)$ *and a ciphertext c, and returns*

$$c' = \big((\alpha_0', \beta_0'), (\alpha_1', \beta_1')\big) = \big((\alpha_0\alpha_1^{k_0'}, \beta_0\beta_1^{k_0'}), (\alpha_1^{k_1'}, \beta_1^{k_1'})\big),$$

  where $r' = (k_0', k_1') \in_R \mathbb{Z}_q^* \times \mathbb{Z}_q^*$ *is an randomly chosen pair of values. We write* $c' \leftarrow \mathsf{URe}(c)$*.*

Application to RFID systems

Golle et al. [62] propose using the GJJS scheme to enhance privacy of RFID tags. Each tag stores a ciphertext c that is an encrypted version of its identifier id under some public key pk of its owner, i.e. $c \leftarrow \mathsf{UE}(pk, id)$. Since tags themselves are not capable of performing the re-encryption process, more powerful computing agents, such as readers, re-encrypt the ciphertexts c stored in tags.

In practice, when a customer has bought a number of items with RFID tags attached in the market, readers in the shop or public area can re-encrypt the ciphertexts

in tags as a public service to prevent unauthorised tracking of the tags. More specifically, a reader obtains c from the tag, re-encrypts it to c', i.e. $c' \leftarrow \mathsf{URe}(I, c)$, and writes c' to the tag. Re-encryption can be performed without knowledge of the underlying public key pk, and thus a reader is able to randomise any tag that have been encrypted (or re-encrypted) using the shared system parameter.

Privacy analysis

The USSR security property plays an essential role to enhance privacy *against passive eavesdropping adversary.* That is, even though an adversary knows the currently stored ciphertext and the associated identifier of a tag, the adversary cannot identify the tag any longer once the ciphertext is re-encrypted using the GJJS scheme.

Golle et al.'s scheme, however, is not secure *against active adversary*,[17] thus not satisfying any type of privacy in the Definition 5.3. Golle et al. [62] identify basic vulnerability. On seeing a ciphertext $c = \big((\alpha, \beta), (\alpha', \beta')\big)$ stored in a tag, an adversary could re-write the ciphertext $c' = \big((\alpha, \beta), (1, 1)\big)$ to the tag. It is straightforward to verify that such a ciphertext will not change after subsequent re-encryptions. They thus suggest that readers should always check whether a ciphertext read from tags has this degenerate form.

Even such a security method is performed in readers, the adversary is still able to trace tags as pointed out by Siato et al. [120]. Suppose an adversary generates a public/private key pair $(\tilde{y} = g^{\tilde{x}}, \tilde{x}) \in \mathbb{G} \times \mathbb{Z}_q^*$. The adversary then constructs the following ciphertext $\tilde{c}$ and writes it to a target tag $\tilde{\mathcal{T}}$:

$$\tilde{c} = \big((\tilde{m}\tilde{y}^{k_0}, g^{k_0}), (\tilde{y}^{k_1}, g^{k_1})\big),$$

where $\tilde{m}$ is the adversary's chosen message and $(k_0, k_1) \in_{\mathsf{R}} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$. It follows that even after the ciphertext $\tilde{c}$ in $\tilde{\mathcal{T}}$ has been re-encrypted, the adversary can recognise $\tilde{\mathcal{T}}$ by decrypting the ciphertext resident in $\tilde{\mathcal{T}}$ using its private key. More specifically, even if the ciphertext $\tilde{c}$ in $\tilde{\mathcal{T}}$ is re-encrypted to

$$\tilde{c}' = \big((\alpha_0, \beta_0), (\alpha_1, \beta_1)\big) = \big((\tilde{m}\tilde{y}^{k_0}\tilde{y}^{k_1 k_0'}, g^{k_0} g^{k_1 k_0'}), (\tilde{y}^{k_1 k_1'}, g^{k_1 k_1'})\big),$$

where $(k_0', k_1') \in_{\mathsf{R}} \mathbb{Z}_q^* \times \mathbb{Z}_q^*$, an adversary can still recognise $\tilde{\mathcal{T}}$ by decrypting $\tilde{c}'$ using the private key $\tilde{x}$, i.e. by checking whether $\alpha_0/\beta_0^{\tilde{x}} = \tilde{m}$ and $\alpha_1/\beta_1^{\tilde{x}} = 1$.

[17] Active adversary can write into tags the data that is selected by the adversary.

5.4.3 Insubvertible encryption (IE)

As described in previous section, Golle et al.'s universal re-encryption allows an adversary to *mark* RFID tags so that they can be recognised later even after they have been re-encrypted [120]. To address this limitation, Ateniese, Camenisch, and Medeiros [2] proposed the notion of an insubvertible encryption scheme, which uses a special type of universal re-encryption technique. Ateniese et al. [2] propose storing a (ciphertext, certificate) pair in an RFID tag, where the ciphertext is an encrypted version of the tag identifier under the public key of the tag issuer, and a certificate is a signature on the issuer's public key. Their scheme permits such a pair to be randomised with only the system parameter, but without any keying material, and the certificate ensures that the ciphertext can only be decrypted by the tag issuer.

Definition

We give the definition of Ateniese et al.'s insubvertible encryption scheme [2], which we call the IE scheme.

Definition 5.8 *The IE scheme is a tuple of polynomial-time algorithms with the following properties:*

- *GenerateCAKey generates a public/private signature key pair* $(\mathcal{C}_{PK}, \mathcal{C}_{SK})$ *for a certification authority* $\mathcal{C}$*.*
- *GenerateKey generates a public/private key pair* (pk, sk) *for a reader* $\mathcal{R}$*.*
- *RegisterPublicKey takes as input a public key* pk *and the private key* $\mathcal{C}_{SK}$ *of a certificate authority* $\mathcal{C}$*, and generates a certificate* u *on* pk*.*
- *InitiateTag takes as input the identifier* m *of a tag* $\mathcal{T}$ *and the pair* (pk, u)*, and encrypts the pair* (m, u) *using the public key* pk*, generating a ciphertext* d *that is written to the tag* $\mathcal{T}$*.*
- *ReadAndDecrypt takes as input the ciphertext* d *for a tag* $\mathcal{T}$ *and the pair* $(\mathcal{C}_{PK}, sk)$*, and determines the identifier of* $\mathcal{T}$*: If* d *is the output of the decryption of* (u, m) *and* u *corresponds to* sk*, output* m*; otherwise, return* $\perp$*.*
- *ReadAndRandomise takes as input* d *and* $\mathcal{C}_{PK}$*, and randomises* d *to* d'*.*

Ateniese et al. [2] construct an example scheme of the IE scheme to prevent the malicious writing attack described in section 5.4.2. Let $E(\mathbb{F}_p)$ be an elliptic curve defined over the field $\mathbb{F}_p$ of low embedding degree, and let $\mathbb{G}_1$ be a large subgroup of prime order q in $E(\mathbb{F}_p)$. Suppose also that its embedding degree l be the smallest integer such that a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathrm{T}$ $(= \mathbb{F}_{p^l})$ exists, where $\mathbb{G}_2$ is a subgroup of $E(\mathbb{F}_{p^l})$. Let g be a generator of $\mathbb{G}_1$, and $\tilde{g}$ be a generator of $\mathbb{G}_2$. As previously, we use multiplicative group notation, instead of the additive notation often used in elliptic curve settings.

The certification authority $\mathcal{C}$ first generates a public/private key pair $(\mathcal{C}_{PK}, \mathcal{C}_{SK})$, where $\mathcal{C}_{PK} = (\tilde{g}^s, \tilde{g}^t)$ and $\mathcal{C}_{SK} = (s,t) \in_\mathsf{R} \mathbb{F}_q \times \mathbb{F}_q$. The reader $\mathcal{R}$ then generates a public/secret key pair $(pk, sk) = (y, x)$, where $x \in_\mathsf{R} \mathbb{F}_p$ and $y = g^x \in \mathbb{G}_1$, and $\mathcal{C}$ issues a certificate $u = (a_1, a_2, a_3, a_4, a_5) = (a, a^t, a^{s+xst}, a^x, a^{xt}) \in \mathbb{G}_1^5$ for the public key y, where $w \in_\mathsf{R} \mathbb{F}_q$, $a = g^w \in \mathbb{G}_1$, and

$$\left.\begin{array}{c} e(a_1, \tilde{g}^t) = e(a_2, \tilde{g}), \quad e(a_4, \tilde{g}^t) = e(a_5, \tilde{g}) \\ \text{and} \quad e(a_3, \tilde{g}) = e(a_1 a_5, \tilde{g}^s). \end{array}\right\} \quad (*)$$

The reader $\mathcal{R}$ then initiates a tag $\mathcal{T}$ with identifier m by writing it the value $d = (u, c) \in \mathbb{G}_1^7$, where $r \in_\mathsf{R} \mathbb{F}_q$, $u \leftarrow (a_1^r, a_2^r, a_3^r, a_4^r, a_5^r) \in \mathbb{G}_1^5$, and $c = (c_1, c_2) = (g^k, my^k) \in \mathbb{G}_1^2$ for $k \in_\mathsf{R} \mathbb{F}_q$.

$\mathcal{R}$ can identify $\mathcal{T}$ in the following way. $\mathcal{R}$ first verifies the certificate u by checking (i) the equations in $(*)$ and (ii) whether $a_1^x = a_4$ $(\in \mathbb{G}_1)$ to ensure that u is a certificate for its own public key y. $\mathcal{R}$ then returns $m \leftarrow c_2 / c_1^x$ $(\in \mathbb{G}_1)$ if u is valid, and $x \leftarrow \perp$ otherwise.

Any reader can then verify and randomise $\mathcal{T}$ in the following way. $\mathcal{R}'$ verifies the certificate u by checking the equations in $(*)$, and writes d' to $\mathcal{T}$, where

- $d' \leftarrow \rho$, where ρ is a *dummy* value, if u is invalid,
- $d' \leftarrow (a_1^v, a_2^v, a_3^v, a_4^v, a_5^v, a_1^z c_1, a_4^z c_2) \in \mathbb{G}_1^7$ for $v, z \in_\mathsf{R} \mathbb{F}_q$, otherwise.

The above process enhances tag privacy by either writing into dummy value when the certificate is not valid, or randomising the ciphertext otherwise. We note that the randomising process applies the principle of Golle et al.'s universal re-encryption

[62]. Specifically, we note that $a_1 = g^w$ and $a_4 = y^w$ for some $w \in \mathbb{F}_q$, where y is the public key for which the certificate is issued. We thus have the following: $(a_1^z c_1, a_4^z c_2) = (g^{wz} g^k, y^{wz} m y^k) = (g^{w'}, m y^{w'})$ for $w' = wz + k$.

Privacy analysis

Ateniese et al. [2] define the notion of privacy to mean that if a tag has been read and randomised by an honest reader, it cannot be tracked by an adversary. This security definition is equivalent to (*discrete*, *wide*, *strong*, ω)-privacy.

Ateniese et al. [2] prove privacy of the scheme using the Universal Composability (UC)/Reactive Simulatability frameworks [24, 113]. They first provide a rigorous specification of the ideal functionality of the IE scheme (the ideal world scenario), where an ideal world adversary $\mathcal{S}$ cannot break privacy. They then propose a specific construction of the IE scheme (the real world scenario). They finally show that the inputs/outputs of all parties, including adversaries in the real and ideal world, are identically distributed, and thus a real world adversary $\mathcal{A}$ cannot break privacy, as an ideal world adversary $\mathcal{S}$ cannot by definition of the ideal functionality.

The IE scheme, however, is vulnerable against *insider attack*. When any legitimate subscriber tries to track some target tags owned by other subscribers by writing into the tags its certificate and the ciphertext which is encrypted under its public key. The randomising process, of course, cannot prevent its malicious tracking, and, even seriously, it is not possible to identify the insider attacker. This is because, the certificate contains the public key with its randomised version. Ateniese et al. [2] state that such a threat is mitigated in practice by the need for subscribers to maintain their reputation within the system. However, in supply chains, which is the main application of the IE scheme, the most serious adversary is the insiders, e.g. competitor suppliers or superstores.

5.5 Lightweight Protocols

There has been a considerable volume of work devoted to designing cryptographic mechanisms with particularly small computational requirements. Juels and Weis

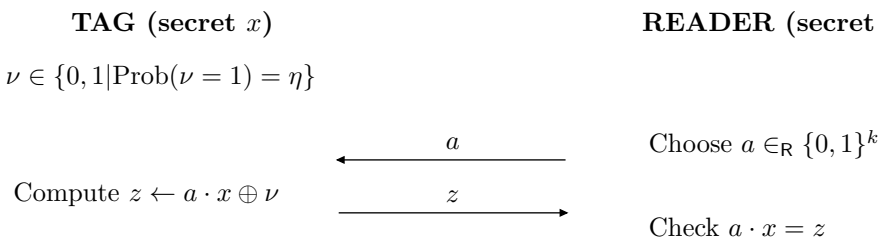


Figure 5.8: One round of the HB protocol

[81, 139] propose two versions of lightweight authentication protocols, called the HB/HB$^+$ protocols, whose security can be reduced to a problem called Learning Parity in the Presence of Noise (LPN). These protocols have been widely discussed as seen in other variants [21, 50, 65, 83, 95, 109, 142].

The LPN problem requires an adversary to recover a k-bit secret vector x after being given a number of bits of the form $b_i = a_i \cdot x \oplus \nu_i$ with unknown noise bits ν_i's, where ν_i is equal to 1 with probability $\eta \in (0, 1/2)$. The LPN problem is known to be NP-hard [14] and is formally defined as follows. We write $\|x\|$ and $\|A\|$ to denote the Hamming Weight of a vector x and matrix A, respectively.

Definition 5.9 (LPN Problem) *Let A be a random $q \times k$ binary matrix, x be a random k-bit vector, $\eta \in (0, 1/2)$ be a constant noise parameter, and ν be a random q-bit vector such that $\|\nu\| \leq \eta q$. Given A, η, and $z = (A \cdot x) \oplus \nu$, find a k-bit vector x' such that $\|(A \cdot x') \oplus z\| \leq \nu q$.*

The HB protocol (see Figure 5.8) is a tag authentication protocol. The message exchange is repeated r times, and the tag is deemed authenticated if the check, i.e. whether $a \cdot x = z$, fails at most ηr times. The HB protocol is secure against a passive adversary under the LPN hardness assumption, but not secure against an active attack [81]. An active adversary could challenge a tag with a chosen value of a, such as e.g. $||a|| = 1$, multiple times, and thereby recover the value $a \cdot x$. Once k linearly independent values a haven been collected, the adversary can recover x using Gaussian elimination [3].

The HB$^+$ protocol (see Figure 5.9) is a somewhat more sophisticated protocol designed to prevent the extraction of tag secrets by the type of attacks described

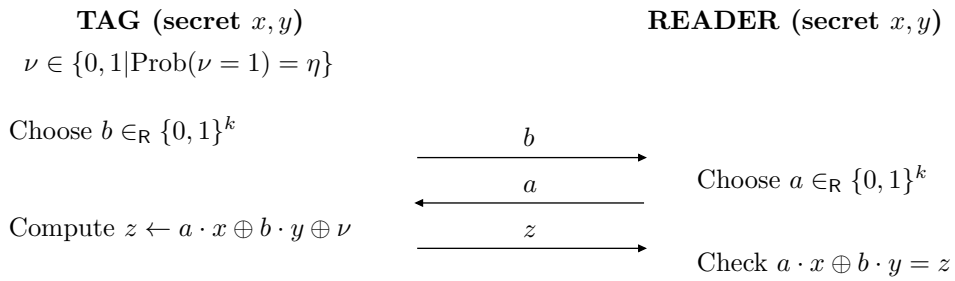


Figure 5.9: One round of the HB^+ protocol

above. The secret key shared between a tag and a reader consists of a pair (x, y) of k-bit vectors. The HB^+ protocols is repeated r times, and the tag is deemed to be successfully authenticated if the check, i.e. whether $a \cdot x \oplus b \cdot y = z$, fails at most ηr times.

The HB^+ protocol provides (*universal*, *narrow*, *destructive*, ω)-privacy under the LPN hardness assumption [58, 81]. In particular, Gilbert et al. [58] has discussed that HB^+ is vulnerable to a *wide* adversary, showing that each "accept" or "reject" outcome from a reader reveals one bit of secret information.

Specifically, suppose that an adversary intercepts the second message a and instead sends $a \oplus \delta$ to a tag, where the k-bit vector δ is chosen so that $\|\delta\| = 1$ (i.e. δ has a single non-zero bit). The same vector δ is used in each of the r rounds of the protocol. If the authentication is successful, $\delta \cdot x = 0$ with overwhelming probability; otherwise, $\delta \cdot x = 1$ with overwhelming probability. Acceptance or rejection by the reader thereby reveals one bit of secret key x, and a k-bit secret vector x can be recovered after k instances of this attack.

5.6 Conclusions

In this chapter we gave a formal privacy definition, and discussed the existing RFID schemes which have received very wide attention in the literature. In the following chapter we discuss EPC tags which seem likely to be used in the majority of RFID application, mostly affecting consumer privacy as well as supply chain security. We

then propose a privacy-enhancing RFID scheme, and discuss the feasibility of the practical deployment of the proposed scheme and existing schemes to the EPC tags.

Chapter 6

Proposed RFID Systems

Contents

We describe the EPCglobal Network and its associated RFID system, which looks set to become the most widely deployed RFID system in the near future. We define the security requirements arising from threats that arise from the RFID systems within the EPCglobal Network. We then construct an example of a refresh-based RFID system, and analyse its security and privacy properties. Finally, we investigate the applicability of the proposed system to the EPCglobal Network.

6.1 Introduction

The EPCglobal Network [71] is a standard-based approach designed to help realise automated global supply chain management. EPCglobal[1], a subscriber-driven organisation, is leading the development of the EPCglobal Network. The EPCglobal Network uses RFID technology to obtain information for individual objects, and uses Internet technology to create a network for sharing the information captured from tags among authorised trading partners.

An EPC (Electronic Product Code) tag is the key component in the EPCglobal Network. The EPC tag is an RFID tag, which is attached to, or embedded in, items. These technologies could usefully be extended to consumers beyond the supply chain to maximise the benefits of RFID technology. However, security and privacy threats for such applications are potentially serious.

In a privacy-enhanced RFID system, an RFID tag identifies itself to an authorised reader using a sequence of pseudonyms, and these pseudonyms should appear random to any entity other than the authorised reader. EPC tags, however, are incapable of either computing or storing such a sequence of pseudonyms. The refresh-based RFID schemes discussed in section 5.4 could be used to ensure privacy of EPC tags, delegating expensive computations to external transceivers.

In section 6.2 we briefly describe the EPCglobal Network [71] and its associated RFID standard, and define security and privacy requirements. In section 6.3 we then propose a refresh-based RFID system which complies with the underlying RFID standard, and analyse privacy of the proposed scheme using the privacy model defined in the previous chapter. Finally, in section 6.4, we investigate the feasibility of practical deployment of the proposed scheme and the existing schemes to the EPCglobal Network [71].

[1] http://www.epcglobalinc.org

6.2 Security and privacy in the EPCglobal Network

We briefly describe the EPCglobal Network and the standard of its associated RFID system, and then investigate the security and privacy issues for the RFID system.

6.2.1 The EPCglobal Network

Efficiency gains in supply chain management can deliver significant savings to businesses, and RFID technology has been introduced as a way of achieving this. EPC tags provide two attractive features by comparison with conventional barcodes:

- *Automation*: EPC tags can transmit information to RFID readers via RF without requiring line-of-sight or physical contact. This feature reduces the need for potentially costly manual intervention in the scanning process.
- *Unique Identification*: While a barcode typically specifies the type of a product, an EPC tag assigns a unique serial number to individual items. This unique identifier associated with an object can be used as a pointer to a database containing a detailed history of the object.

The EPCglobal Network then provides two fundamental capabilities to support supply chain management [73]. It allows companies both to know where a product is at any time within the supply chain (*tracking*), and to see exactly where a product has been throughout the entire supply chain process (*tracing*).

We now briefly describe the architecture, components, functionality, and implementation of the EPCglobal Network.

Architecture framework

As illustrated in Figure 6.1, the EPCglobal architecture framework [71] describes the activities carried out by EPCglobal subscribers and the role that components of the EPCglobal architecture framework play in facilitating those activities. It also

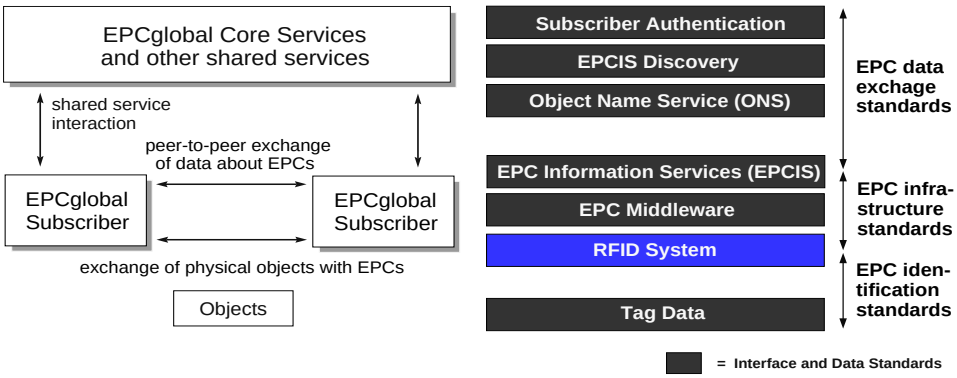


Figure 6.1: The EPCglobal Network architecture framework [71]

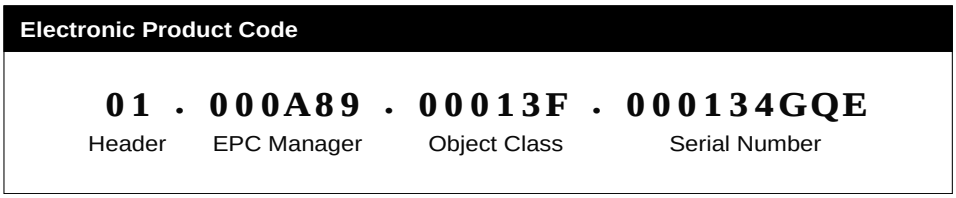


Figure 6.2: Electronic Product Code [71]

defines three broad standards, each supported by a group of minor standards. We first describe the three main groups of standards.

- The **Electronic Product Code (EPC) data exchange standards** provide a means for subscribers to share data about EPCs within defined user groups, or with the general public, in order to enable sharing of information about the movement of physical objects through the network.
- The **EPC Infrastructure standards** define interface standards for subscribers' internal systems to enable them to share EPC data.
- The **EPC identification standards** are designed to ensure that, when one subscriber delivers a physical object to another subscriber, the recipient will be able to determine the EPC of the object and interpret it properly.

The EPCglobal Network consists of the following components [71], which are used to realise its three major activities, i.e. shared service interaction, peer-to-peer exchange

of data about EPCs, and exchange of physical objects with EPCs, as shown in Figure 6.1.

- **Electronic Product Codes:** EPCs are analogous to the Universal Product Codes (UPCs) used in bar codes. Unlike UPCs, however, EPCs uniquely identify individual product items via a *serial number*, as shown in Figure 6.2. The EPC Manager, which is typically a manufacturer in the supply chain, is an organisation that is granted the right to use one or more blocks of EPCs within a designated coding scheme, in order to independently assign EPCs to physical objects or other entities. We make a distinction between an EPC and an EPC tag. An EPC tag is an RFID device attached to an object, while an EPC is a bit string stored in an EPC tag.

- **RFID System:** The RFID system consists of tags and readers. EPCs are stored on the tags, which are applied to cases, pallets, and/or individual items. Readers communicate with tags via radio, and deliver the captured EPCs to the local business information systems using the EPC Middleware.

- **EPC Middleware:** This is a subscriber's internal infrastructure, including the readers, data collection software, and enterprise applications. It manages the captured tag information by communicating with EPCIS and other information systems at the business site.

- **EPC Information Services (EPCIS):** This is a data repository which stores EPC information about unique items. Each company has its own EPCIS, and designates which trading partners have access to its EPCIS.

- **Discovery Services:** This is a collective term for the Object Name Service (ONS) and the EPCIS Discovery Service. The ONS is a simple lookup service that, given an EPC as input, outputs the address (in the form of a Uniform Resource Locator (URL)) of EPCIS which issued the EPC. The EPCIS Discovery Service provides a directory of the EPCISs of the parties which participated in the supply chain for a particular EPC.

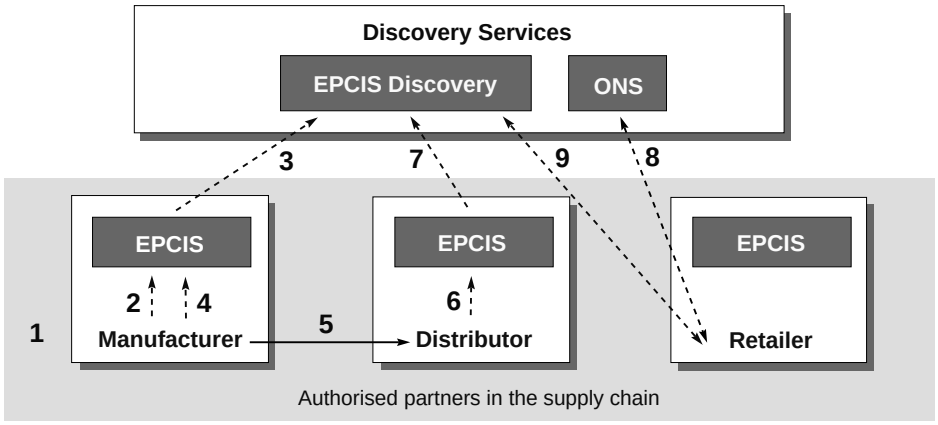


Figure 6.3: The EPCglobal Network in action [72]

The EPCglobal Network in action

The EPCglobal Network enables information to be disseminated across the entire supply chain. As shown in Figure 6.3, it involves the following steps.

1. A product is first given a tag that includes an EPC.
2. The information on this particular product is added to the manufacturer's EPCIS.
3. The location of this information is passed to the Discovery Services.

4-5. When the product leaves the manufacturer's premises, the EPCIS is updated with the departure information of the product.

6-7. The arrival of the product is registered with the distributor's EPCIS, and the location information of the product is updated with the Discovery Services.

8-9. A retailer asks the ONS for the location of the manufacturer's EPCIS in order to obtain the product information, and also asks the EPCIS Discovery Service to obtain the history of the whereabouts of the arrived product.

A more detailed description is given in [72].

Implementation

Adoption of EPCglobal Network technology is still at an early stage. Companies in the Fast Moving Consumer Goods (FMCG) industry are currently testing the components in the EPCglobal Network at the pallet and case level [69]. Item-level tagging, however, is expected to only be implemented on a large scale when tag cost drops to $0.05 and the standards are well established [77].

6.2.2 Gen2 Standards

The Auto-ID centre first developed the EPCglobal Network as a way of bringing the benefits of RFID technology to the global supply chain. The original Class-1 Generation-1 RFID standard (Gen1 standard) was developed by a small number of commercial companies, and was not an open standard. The Auto-ID center also started the development of the Class-1 Generation-2 (Gen2 standard), eventually transferring the responsibility for its further development to EPCglobal, which was formed by EAN International and the Uniform Code Council (UCC).

In this section we describe the standard of RFID systems for the EPCglobal Network, known as the Class-1 Generation-2 UHF (860–960 MHz) RFID standard (ISO/IEC 18000-6C) [70].

Memory

The memory of EPC tags is logically separated into four distinct banks, as shown Figure 6.4. The memory banks are defined as follows.

- EPC memory contains a CRC-16, Protocol-Control (PC) bits, and a code (e.g. an EPC) that identifies the object to which the tag is or will be attached. A CRC-16 is a cyclic-redundancy check, and it conforms with ISO/IEC 13239. The PC bits contain an EPC length field, which supports up to 416 bits of EPC length.
- User memory allows to store user-specific data.

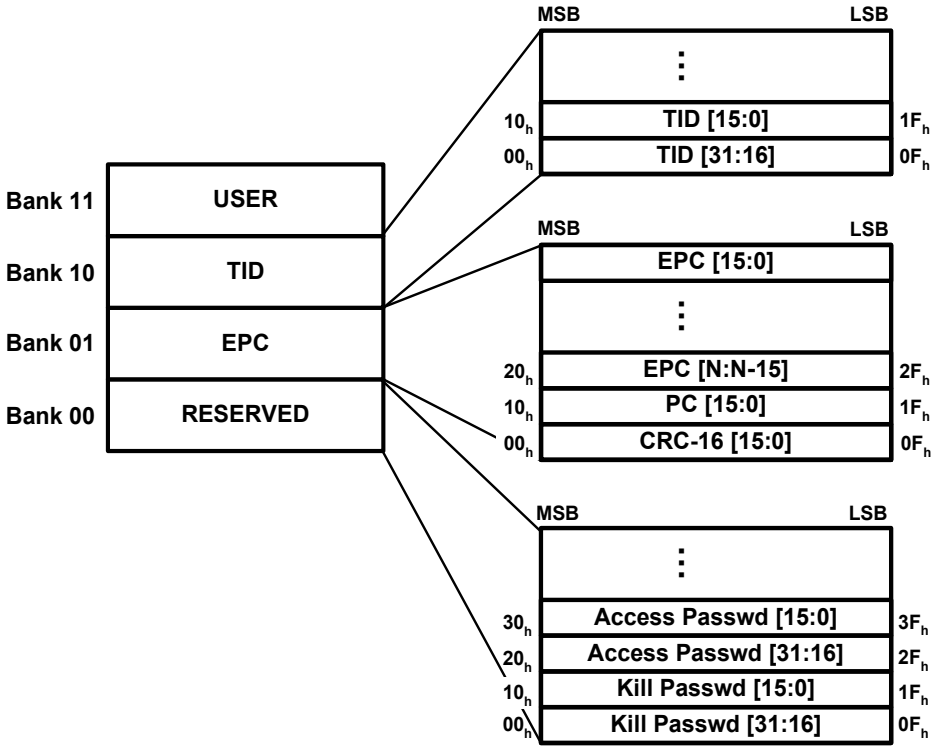


Figure 6.4: Operations between tag and reader [70]

- TID memory contains sufficient information for a reader to uniquely identify the custom commands and/or optional features that a tag supports.
- Reserved memory contains two 32-bit passwords, a kill password and an access password.

A reader may lock, permanently lock, unlock, or permanently unlock memory, thereby preventing or allowing subsequent changes. The kill and/or access passwords can be individually locked, as can the EPC, TID, and User memory. If the kill and/or access passwords are locked, they are rendered both unreadable and unwritable, unlike other memory banks, which are always readable regardless of their lock status. We note that the EPC memory bank can be read during 'inventory' operation, but the other memory banks may be read during 'access' operation, as described in the next section.

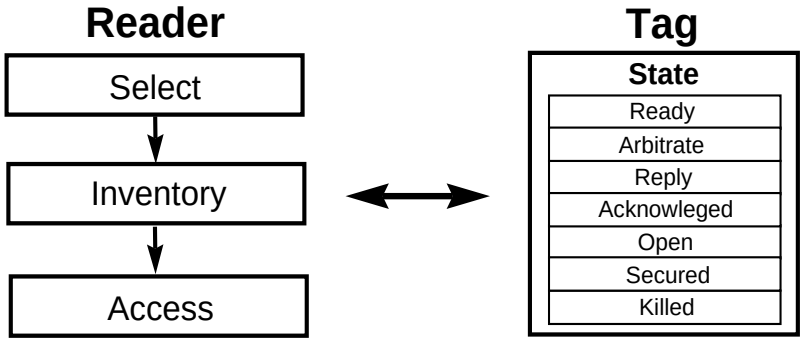


Figure 6.5: Tag-reader operations and tag state changes [70]

Air interface

We now describe the air interface between a tag and a reader, following 'Specification for RFID Air Interface by EPCglobal' [70]. Readers manage tag populations using the three basic operations, while tags implement the states, as shown in Figure 6.5. The operations are defined in three steps: select, inventory, and access.

In the select process a reader selects a particular tag population prior to use of the inventory process. It involves issuing multiple identical `Select` commands to select the tags matching user-defined criteria.

In the inventory process a reader uniquely identifies, using commands of `Query`, `QueryAdjust`, `QueryRep`, `ACK`, and `NAC`. Upon receiving a `Query` participating tags pick a random value in $(0, 2^Q - 1)$, inclusive, and load this value into their slot counter. The slot counter parameter Q is an integer in $(0, 15)$, inclusive. If tags pick a zero, they transition to the *ready* state and reply immediately; otherwise, they transition to the *arbitrary* state, and await a `QueryAdjust`/`QueryRep` command. Assuming that a single tag replies, the query-response protocol proceeds between a tag $\mathcal{T}$ and a reader $\mathcal{R}$ as follows.

(a) Upon receiving a `Query` from $\mathcal{R}$, $\mathcal{T}$ backscatters a randomly generated 16-bit RN16, and enters the *reply* state.

(b) $\mathcal{R}$ acknowledges $\mathcal{T}$ by sending an `ACK`, which is the same as the received RN16.

(c) On receiving `ACK`, $\mathcal{T}$ checks that RN16 = `ACK`; if so, $\mathcal{T}$ backscatters its PC, EPC, and CRC-16.

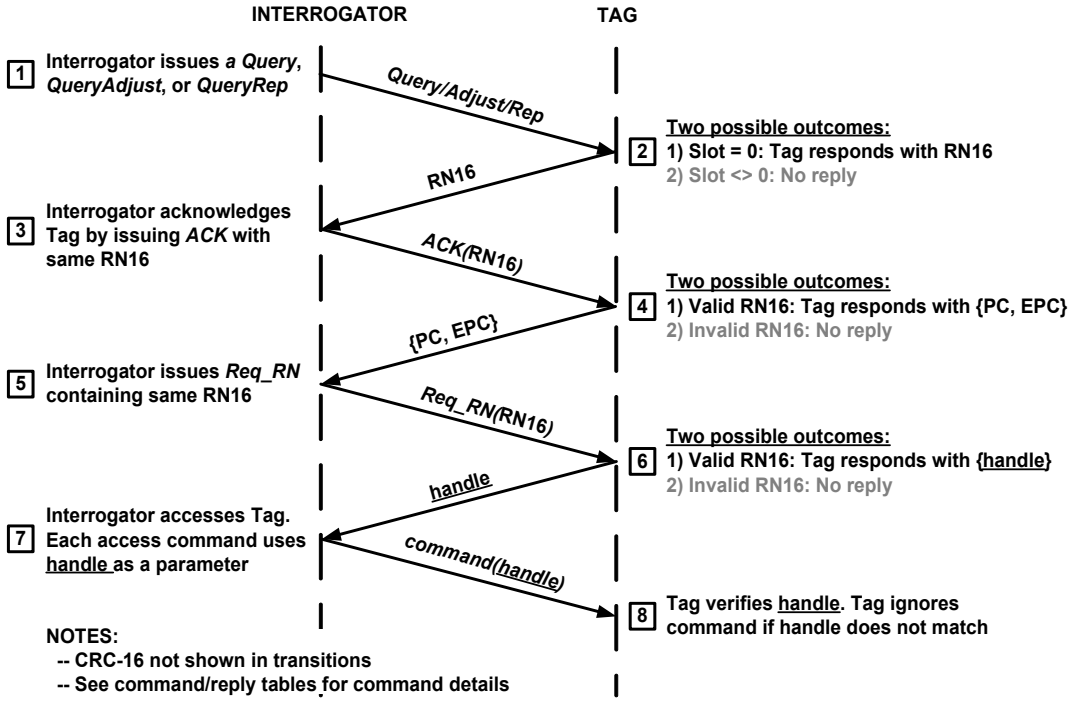


Figure 6.6: Example of tag inventory and access [70]

After acknowledging $\mathcal{T}$, $\mathcal{R}$ may issue a `QueryAdjust`/`QueryRep` command, causing $\mathcal{T}$ to transition to the *ready* state. If $\mathcal{T}$ fails to receive `ACK` within a pre-defined timeout, or receives `ACK` with an erroneous RN16 in step (c), $\mathcal{T}$ transitions to the *arbitrary* state. $\mathcal{T}$ in the *arbitrary* or *ready* state that receives a `QueryAdjust` first adjusts Q and picks a random value in $(0, 2^Q - 1)$, inclusive, and loads this value into its slot counter. $\mathcal{T}$ in the *arbitrary* state that receives a `QueryRep` decrements its slot counter, and transitions to the *reply* state and backscatters RN16 when its slot counter reaches a zero. At any point, $\mathcal{R}$ may issue a `NAK` to cause $\mathcal{T}$ to transition to the *arbitrary* state.

When multiple tags reply in step (a), a reader can resolve an RN16 from one of the tags by detecting and resolving collisions at the waveform level. Unresolved tags receive erroneous RN16 and return to the *arbitrary* state. Also, two or more readers can independently inventory the common tag population. The more detailed description can be found in the Gen2 standard [70].

After acknowledging a tag, a reader may choose to access it during the access process. The specification defines mandatory commands such as `Req_RN`, `Read`, `Write`, `Kill`,

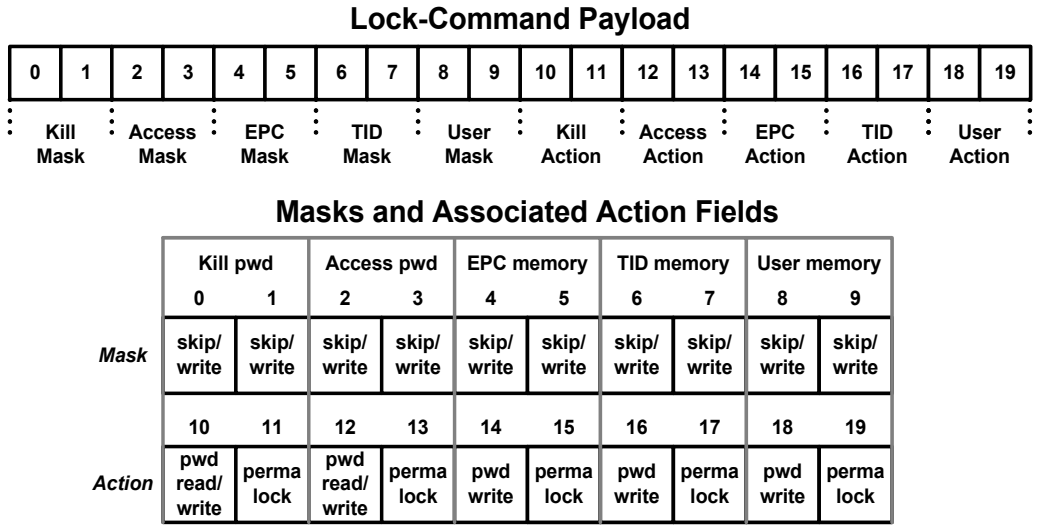


Figure 6.7: Lock command payload and usage [70]

and `Lock`, and optional commands such as `Access`, `BlockWrite`, and `BlockErase`.

A reader $\mathcal{R}$ and a tag $\mathcal{T}$ in the *acknowledge* state start access process as follows.

(a) $\mathcal{R}$ issues a `Req_RN` to $\mathcal{T}$.

(b) $\mathcal{T}$ generates and stores a new RN16, which is called a handle, and backscatters the handle. $\mathcal{T}$ transitions to the *open* state if its access password is nonzero; otherwise, it transitions to the *secured* state.

$\mathcal{R}$ may now issue further access commands, or may terminate the access sequence by issuing a `Query`, `QueryAdjust`, `QueryRep`, or `NAK`. All access commands issued to $\mathcal{T}$ in the *open* or *secured* states include the tag's handle, which the tag verifies prior to executing the received access commands. This handle is fixed for the entire duration of an access sequence.

We particularly describe a set of commands, `Lock`, `Write`, and `Access`. The description of the other commands used in access operation can be found in the Gen2 standard [70].

A reader uses the `Lock` command in order to:

- *lock* individual passwords (i.e. kill and access passwords), thereby preventing or allowing subsequent reads and/or writes of that password;

Table 6.1: Lock Action-field functionality [70]

pwd-write	permalock	Description
0	0	Associated memory bank is writeable from either the **open** or **secured** states.
0	1	Associated memory bank is permanently writeable from either the **open** or **secured** states and may never be locked.
1	0	Associated memory bank is writeable from the **secured** state but not from the **open** state.
1	1	Associated memory bank is not writeable from any state.
pwd-read/write	**permalock**	**Description**
0	0	Associated password location is readable and writeable from either the **open** or **secured** states.
0	1	Associated password location is permanently readable and writeable from either the **open** or **secured** states and may never be locked.
1	0	Associated password location is readable and writeable from the **secured** state but not from the **open** state.
1	1	Associated password location is not readable or writeable from any state.

Table 6.2: The `Lock` command and response [70]

	Command	Payload	RN	CRC-16
# of bits	8	20	16	16
description	11000101	Mask and Action Fields	handle	

	Header	RN	CRC-16
# of bits	1	16	16
description	0	handle	

- *lock* individual memory banks, thereby preventing or allowing subsequent writes to that bank; and
- *permalock* (i.e. make permanently unchangeable) the lock status for a password or memory bank.

The syntax of the `Lock` command and response is defined as Table 6.2. The handle is used as the temporary identifier of the tag during the access sequence. The `Lock` command contains a 20-bit payload (as shown in Figure 6.7) defined as follows.

- The first 10-bit payload bits are Mask bits, and a tag interprets these bits as follows.
  - If Mask = 0, a tag ignores (i.e. skips) the associated Action field and retains the current lock setting.

  - If Mask = 1, a tag implements the associated Action field and overwrites the current lock setting.

- The last 10 payload bits are Action bits, and are defined as in Table 6.1.

Permalock bits, once they have been set, cannot be changed. Depending the success of `Lock` command operation, a tag replies by backscattering its handle if the operation has been successful; otherwise, the tag backscatters an error code. If a reader does not observe any reply within 20ms, it assumes that the lock operation has not been successful.

We next describe the use of the `Write` and `Access` commands. When a reader $\mathcal{R}$ sends these commands, it sends a 16-bit word (either data or half-passwords) to a tag $\mathcal{T}$ at a time, using a technique called cover-coding to prevent sensitive words from being transmitted in cleartext in high-power forward link. The sequence is defined as follows.

(a) $\mathcal{R}$ issues a `Req_RN`, to which $\mathcal{T}$ responds by backscattering a new RN16. $\mathcal{T}$ stores this RN16.

(b) $\mathcal{R}$ generates and transmits a ciphertext that is a bit-wise XOR of a 16-bit word (to be transmitted) with the received RN16.

(c) $\mathcal{T}$ decrypts the received ciphertext string by performing a bit-wise XOR of the received 16-bit ciphertext with the stored RN16.

$\mathcal{R}$ must not use re-use an RN16 for cover-coding, and it shall first issue `Req_RN` for sending another data. The handle is not used for cover-coding. We give an example of tag inventory and access operations in Figure 6.6.[2] We now describe the `Write` and `Access` commands more in detail.

The `Access` command is used to cause a tag with a non-zero access password to transition form the *open* state to the *secured* state.[3] The syntax of the `Access` command and response is defined as Table 6.3. Since the `Access` command contains a half-password (i.e. 16 bits), a reader issues two `Access` commands using the cover-coding technique described above, and a tag incorporates the necessary logic to successively accept two 16-bit sub-portions of a 32-bit access password.

[2] The Gen2 standard uses a term interrogator to denote a reader.
[3] A tag with a zero access password is never in the *open* state as described earlier in this section.

Table 6.3: The `Access` command and response [70]

	Command	Password	RN	CRC-16
# of bits	8	16	16	16
description	11000110	(½ access password) ⊕ RN16	handle	

	RN	CRC-16
# of bits	16	16
description	handle	

Table 6.4: The `Write` command and response [70]

	Command	MemBank	WordPtr	Data	RN	CRC-16
# of bits	8	2	EBV	16	16	16
description	11000011	00: Reserved 01: EPC 10: TID 11: User	Address pointer	RN16 ⊕ word to be written	handle	

	Header	RN	CRC-16
# of bits	1	16	16
description	0	handle	

More specifically, a reader issues two `Access` commands, the first containing the MSBs of the tags's access password XORed with an RN16, and the second containing the LSBs of the tags's access password XORed with a different RN16.[4] For the first `Access` command, the tag backscatters its handle to acknowledge that it received the command. When the second `Access` command has been received and the entire 32-bit access password is correct (by applying the step (c) above), then the tag backscatters its handle to acknowledge that it has executed the command successfully and transitioned to the *secured* state; otherwise, the tag does not reply.

The `Write` command allows a reader to write a word in the Reserved, EPC, TID, and User memories of a tag in the *open* or *secured* state. The syntax of the `Write` command and response is defined as Table 6.4. The `Write` command has the following fields.

- MemBank specifies the memory bank where a word is to be written.

[4]The notation MSB denotes the most significant bits, and the LSB the least significant bits.

- WordPtr specifies the 16-bit word address for the memory write. For example, $\mathsf{WordPtr} = 00_{\mathsf{h}}$ means the first 16-bit memory word, $\mathsf{WordPtr} = 01_{\mathsf{h}}$ means the second 16-bit memory word, etc.
- Data contains a 16-bit word to be written.

Depending the success of `Write` command operation, the tag backscatteres its handle if the operation has been successful; otherwise, the tag backscatters an error code. If the reader does not observe any reply within 20ms, it means that the operation has not been successful.

6.2.3 Security and privacy requirements

In this section we discuss security and privacy requirements in the EPCglobal Network. Since security issues for many components of the EPCglobal Network, other than the RFID system, are similar to those arising in other Internet applications, we only assess the corresponding security and privacy properties required in the RFID system.

Privacy is most important requirement for the deployment of the EPCglobal network. An EPC stored in a tag can permit surreptitious inventorying of an object the tag is attached to. This is because the field 'EPC manager' in an EPC represents the manufacturer, and the 'object class' is typically a product code. A reader could surreptitiously harvest personal information such as: the type of medication a person is carrying, clothing size, accessory preference, etc. Also, by surreptitiously scanning tagged items, an organisation could learn about stock turnover rates in the supply chains of its competitors. Confidentiality or data privacy is thus required.

Requirement 6.1 (Confidentiality (or Data Privacy)) *Reading EPC tags should not give any product information of the tagged item to the unauthorised entities.*

A unique tag identifier, such as an EPC or other tag-specific string, can be used to track an object or a person carrying a tag in terms of both time and space. The collected information can be merged and linked to create a personal profile, or generate critical information about inbound and outbound flows to/from a corpo-

rate warehouse. In a military supply chain, enemy forces could learn about troop movements by monitoring RFID communications. Anonymity or location privacy of objects/people carrying tags is thus required.

Requirement 6.2 (Anonymity (or Location Privacy)) *It should be infeasible for unauthorised entities to have knowledge of the whereabout of the products attached with EPC tags.*

While identification of items is the main purpose of using RFID technology in the supply chain, authentication[5] is also required for certain items, such as expensive products or drugs. Drug counterfeiting is a particularly serious issue [77]; confirming the authenticity of drug supplies is challenging because of the complexity of modern supply chains, i.e. it is difficult to ensure the provenance of delivered items. Tag authenticity is thus required.

Requirement 6.3 (Authenticity) *Once an EPC has been programmed into a specific tag by the EPCglobal Network, only should this tag be able to claim that it possesses that specific EPC value.*

A Denial of Service (DoS) attack could result in a permanent or temporary loss of the ability of a reader to communicate with a tag. Like any system using wireless communications, RFID systems can be easily disturbed by RF jamming, but this is not an issue specific to RFID systems. We thus only consider DoS attacks arising in the application layer. Availability is thus required.

Requirement 6.4 (Availability) *An authorised reader within reading range of an EPC tag should have uninterrupted access to that tag.*

6.2.4 Gen2 standard and related variant schemes

In this section we give security and privacy analysis for Gen2 standard [70] to be used for RFID system of the EPCglobal Network.

[5]The authentication of an item is performed by authenticating the tag which is attached to the item.

Privacy

The EPCglobal Network addresses confidentiality and anonymity in the following ways [69, 70, 71]. An EPC is simply an identifier for a specific object, and no other information is contained in it. The information associated with an EPC is only accessible to authorised subscribers, and data is transferred between subscribers via secure channels, i.e. making the networks virtually private. At the point of sale, RFID tags are permanently deactivated (killed) to ensure consumer privacy; when a tag receives the kill command, it renders itself permanently inoperable. To prevent inadvertent or malicious disabling of tags, the Gen2 standards requires readers to use a tag-specific 32-bit password.

However, the use of the kill command at the point of sale blocks a range of post-purchase applications of RFID technology, including their use in smart appliances and receiptless return of goods [77], as described below.

- The EPCglobal Network could provide benefits to customers if the network could be extended to smart home systems; such a system could retrieve detailed product information from the EPCglobal Network using the EPC as an identifier. For example, a washing machine could select an appropriate wash cycle and temperature to avoid damage to delicate fabrics by reading a tag in a garment, or a refrigerator might warn its owner when a tagged item of foodstuff has expired.
- People who are physically or mentally impaired might benefit from RFID-enabled smart home systems [114]. For example, a voice-enabled aid to a blind person could recognise the content of an object by reading the embedded tag.
- Customers could return an item without the need for the sales receipt, since a tag could act as an index into the database payment records and help retailers track defective or contaminated items.

Although killing tags at the point of sale may provide consumer privacy, it fails to address security issues for corporate supply chains; since EPC tags emit static identifiers, they are vulnerable to malicious tracking by competitors. As described in Section 6.2.3, unauthorised reading in military supply chains is a particularly serious threat, and killing tags cannot address this.

Authenticity

The EPCglobal Network expects RFID technology to help combat counterfeiting [72], e.g. the authenticity of drug shipments could be checked by using the electronic pedigree provided by Discovery Services. EPC tags, however, may provide only very limited assurance of authenticity, as described below.

A cloning attack involves a counterfeit tag attempting to convince a reader that it is exchanging information with a genuine tag. Since EPC tags emit their resident EPC to any querying reader, an adversary could easily learn the data resident in a tag by simply scanning it. Furthermore, field-programmable EPC tags are available today,[6] and readers typically have no way of checking the validity of the EPCs they scan. These features make EPC tags vulnerable to elementary cloning attacks.

Juels [76] describes a simple way of using the kill functionality in EPC tags to achieve limited counterfeit resistance. The kill password is normally used to authenticate a reader to a tag, i.e. a kill password is used as a means to enable a tag to authenticate a reader prior to self-deactivation. However, the unique kill password shared between a tag and a reader can also be used to enable a reader to authenticate a tag.

More specifically, when an EPC tag receives a kill command including a valid kill password, but the received power is insufficient, it remains operational and emits an error code. Juels thus suggests that, given that it possesses this functionality, a tag could be modified to emit "yes" or "no" indicating the validity of a kill password. Such a protocol, however, has basic vulnerability; any cloned tag, which is not compliant with the Gen2 standard, could simply accept any password, in which case the protocol will always output "`valid`".

To address the problem described above, Juels [76] proposes the modified protocol, called BasicTagAuth (see Figure 6.8). The protocol uses *spurious* passwords, which functions similarly to *winnowing* as introduced by Rivest [116].[7] The number ϱ is a security parameter that defines the number of spurious passwords to be generated (line 4). The function GenPWSet generates a set of $\varrho - 1$ spurious passwords uniformly at random, and inserts the correct password k_i in a random position j (line

[6] See http://www.ti.com/rfid/docs/manuals/pdfSpecs/epc_inlay.pdf.

[7] Rivest suggests inserting false packets into a data stream to achieve confidentiality; a receiver can extract the transmitted message by picking out the correct ones.

Protocol: BasicTagAuth

1. $\mathcal{T} \longrightarrow \mathcal{R}$: id
2. $\mathcal{R}$: if $id = id_x$ for some $1 \leq x \leq n$, then $i \leftarrow x$;
3. else output "`unknown tag`" and halt
4. $\mathcal{R}$: $(j, \{P_i^{(1)}, \ldots, P_i^{(\varrho)}\}) \leftarrow \mathsf{GenPWSet}(i)[\varrho]$;
5. $\lambda \leftarrow$ "`valid`";
6. for $k = 1$ to ϱ do
7. $\mathcal{R} \longrightarrow \mathcal{T}$: $\mathsf{PW\text{-}test}(P_i^{(k)})$
8. $\mathcal{T} \longrightarrow \mathcal{R}$: b
9. $\mathcal{R}$: if $b = 1$ and $k \neq j$, then $\lambda \leftarrow$ "`invalid`"
10. if $b = 0$ and $k = j$, then $\lambda \leftarrow$ "`invalid`"
11. $\mathcal{R}$: output λ

Figure 6.8: The BasicTagAuth protocol [76]

4).

The BasicTagAuth protocol, however, is vulnerable to the adversary who obtains the tag identifier id and the true password $P_i^{(p)}$ by either passively or actively as follows [76]. The adversary first

(i) eavesdrops on the BasicTagAuth protocol run and thus obtains id and $P_i^{(p)}$; or

(ii) obtains id by sending the target tag a read-query, interacts with the reader and obtains the password set $\{P_i^{(k)}\}_{k=1}^{\varrho}$, and, finally, actively tests the keys in the set on the tag to determine $P_i^{(p)}$.

It is then impossible to detect such counterfeit tags from genuine tags.

Availability

As described in Section 6.2.2, Gen2 standard [70] implements an access control for tag-writing and tag-deactivation using tag-specific passwords. If such an access control is not present, e.g. an access password is set to be zero, the rewritable tag

memory, such as the EPC memory, could be manipulated by an adversary so that the tag becomes permanently desynchronised with the authorised reader(s). That is, an adversary may attempt to write into the EPC memory a garbage value, in which case the tag is not recognised by a reader in the subsequent communications.

6.3 Proposed RFID system

We now construct a refresh-based RFID system, namely what we call the RFID-R system, and give privacy analysis using the privacy model in Definition 5.3.

6.3.1 Construction of algorithms

We first define an encoding scheme and then a set of algorithms for RFID system.

Encoding scheme

Following Ateniese et al. [2], we present an encoding scheme which maps bit-strings of fixed length b into elements of an elliptic curve group. Let E be an elliptic curve such that $E(\mathbb{F}_p)$ contains a cyclic subgroup of large prime order, q say. Let $\mathsf{MAC} = (\mathsf{m\text{-}Gen}, \mathsf{Mac}, \mathsf{Ver})$ be a secure message authentication code, as in definition 2.10. The MAC length will be denoted by t,[8] and we also require a w-bit counter ct, where $w = \lfloor \log_2 q \rfloor - b - t - 1$. Unlike the scheme of Ateniese et al. [2], the encoding scheme presented here makes use of the counter ct for authentication purposes.

The encoding algorithm Encode-to-Group is shown in Figure 6.9. The input to Encode-to-Group, is a b-bit tag identifier id and a key K_{MAC} ($\leftarrow \mathsf{m\text{-}Gen}(1^{n'})$). Given a b-bit identifier id and a w-bit ct, the algorithm first computes $M \leftarrow id||ct||\mathsf{Mac}(K_{\mathsf{MAC}}, id||ct)$. Since $|M| = \lfloor \log_2 q \rfloor - 1 \leq \lfloor \log_2 p \rfloor - 1$,[9] M may be interpreted as the binary representation of an integer X smaller than p, and thus as a unique value in $\mathbb{F}_p$.

[8] A reasonable choice for t would be in the range 64–128 bits, depending on the security requirements.

[9] The notation $|M|$ denotes the bit length of M.

Algorithm: Encode-to-Group $(K_{\mathsf{MAC}}, id, ct)$

1. while $ct < 2^w$ do;
2. $\quad m \leftarrow id||ct$; $\tilde{m} \leftarrow \mathsf{Mac}(K_{\mathsf{MAC}}, m)$; $M \leftarrow m||\tilde{m}$;
3. $\quad X \leftarrow \mathsf{O2I}(M)$;
4. $\quad$ if there exists Y such that $E(X, Y) = 0$, then
5. $\quad\quad P \leftarrow (X, Y)$; return (P, ct) and stop;
6. $\quad$ else, increment ct
7. return $\perp$

Figure 6.9: Encoding algorithm

Exploiting this fact, the function O2I maps octet-strings (byte strings) to numbers in $\mathbb{F}_p$, i.e. maps M to X. Regarding X as the x-coordinate of a point in $E(\mathbb{F}_p)$, the algorithm checks whether there exists a value Y satisfying the equation $E(X, Y)$ mod $p = 0$. This equation has either two solutions or none. If two solutions exist, then the algorithm chooses one in a systematic manner, e.g. it could choose the smaller integer. If no solution is found, the algorithm increments ct and repeats the above process.[10] Finally, the algorithm outputs a point $P = (X, Y)$ and the current counter value ct.

Algorithm: Decode-from-Group (K_{MAC}, P)

1. $X \leftarrow \mathsf{X\text{-}coordinate}(P)$;
2. $M \leftarrow \mathsf{I2O}(X)$; $id||ct||\tilde{m} \leftarrow M$;
3. $b \leftarrow \mathsf{Ver}(K_{\mathsf{MAC}}, id||ct, \tilde{m})$;
4. if $b = 1$, return (id, ct); else, return $\perp$

Figure 6.10: Decoding algorithm

The Decode-from-Group algorithm, shown in Figure 6.10, can be used to invert the Encode-to-Group encoding operation. Given input of a key K_{MAC} and a point P, the algorithm processes P's x-coordinate X using the X-coordinate function. Using the function I2O that maps numbers in $\mathbb{F}_p$ to octet-strings, the binary representation of X is recovered, and then parsed as $id||ct||\tilde{m}$. If $\tilde{m}$ is a valid MAC, i.e. $\mathsf{Ver}(K_{\mathsf{MAC}}, id||ct, \tilde{m}) = 1$, then the algorithm returns id and ct. If this check fails,

[10] Since approximately 50% of the values in a finite field possess a square root, the probability that the encoding will fail is 2^{-2^w}.

the algorithm returns the error message $\perp$.

Algorithms

We first define two basic protocols between $\mathcal{R}$ and $\mathcal{T}$. $\mathcal{R}$ reads a pseudonym, i.e. the encrypted version of the tag identifier, resident in $\mathcal{T}$ by performing the Tag-Read protocol. $\mathcal{R}$ updates a pseudonym in $\mathcal{T}$ by performing the Tag-Write(c, ak) protocol. This has the effect of causing $\mathcal{T}$ to replace its existing pseudonym with the value c only if ak is equal to the tag-access key stored in $\mathcal{T}$. $\mathcal{R}$ and $\mathcal{T}$ may also perform the Tag-Write(c) protocol, when $\mathcal{T}$ does not enforce access control for tag writing.

We use multiplicative group notation instead of the additive notation often used in elliptic curve settings. The algorithms of the RFID-R system we define here make use of the ElGamal encryption scheme PE = (e-Gen,Enc,Dec), as given in definition 2.17, and a message authentication code MAC=(m-Gen,Mac,Ver), as given in definition 2.10. We construct the algorithms of RFID-R system using the notations defined in Section 5.2.2.

- Setup-Reader takes as input a security parameter 1^n and returns the system parameter $parm$ and a master key $K_{\mathcal{R}} = (K_{\mathsf{PE}}, K_{\mathsf{MAC}})$, where $K_{\mathsf{PE}} = (pk, sk) \leftarrow$ e-Gen(1^n), $K_{\mathsf{MAC}} \leftarrow$ m-Gen$(1^{n'})$, and $n' = l(n)$ for some polynomial l. We write $\mathcal{R}(parm, K_{\mathcal{R}}) \leftarrow$ Setup-Reader(1^n).

- Setup-Tag is a two-party protocol conducted by $\mathcal{T}$ and $\mathcal{R}$. It takes as inputs $(parm, K_{\mathcal{R}})$ and a tag identifier $id \in \{0,1\}^b$. $\mathcal{R}$ initialises $\mathcal{T}$ as follows.

  $\mathcal{R}$: $ct \leftarrow 00\cdots0$ (w bits);
  $(P, ct) \leftarrow$ Encode-to-Group$(K_{\mathsf{MAC}}, id, ct)$;
  $c \leftarrow$ Enc(pk, P);
  $\mathcal{R} \longleftrightarrow \mathcal{T}$: Tag-Write(c)

  We write $(\mathcal{R}(id, K_{\mathcal{T}}), \mathcal{T}(c)) \leftarrow$ Setup-Tag(id), where $K_{\mathcal{T}} = K_{\mathcal{R}}$.

- Idt-Tag is two-party protocol conducted by $\mathcal{T}$ and $\mathcal{R}$. It takes as inputs $(parm, K_{\mathcal{R}})$ and a ciphertext $c = (\alpha, \beta)$. $\mathcal{R}$ determines the identifier of $\mathcal{T}$.

$\mathcal{R} \longleftrightarrow \mathcal{T}$: Tag-Read

$\mathcal{R}$: $P \leftarrow \mathsf{Dec}(sk, c)$;
if $\perp \leftarrow$ Decode-from-Group(K_{MAC}, P),
then output $\leftarrow$ unknown_tag; return output and halt
else (i.e. if $(id, ct) \leftarrow$ Decode-from-Group(K_{MAC}, P))
output $\leftarrow id$; return output

We write output $\leftarrow$ Idt-Tag$(\mathcal{R}(parm, K_{\mathcal{R}}), \mathcal{T}(c))$.

- Ref-Tag is two-party protocol conducted by $\mathcal{T}$ and $\mathcal{R}$. It takes as input $(parm, pk)$ and a ciphertext $c = (\alpha, \beta)$. $\mathcal{R}$ refreshes the pseudonym stored in $\mathcal{T}$ in the following way.

  $\mathcal{R} \longleftrightarrow \mathcal{T}$: Tag-Read

  $\mathcal{R}$: $c' \leftarrow (\alpha(pk)^{k'}, \beta g^{k'})$ for $k' \in_{\mathsf{R}} \mathbb{Z}_q^*$

  $\mathcal{R} \longleftrightarrow \mathcal{T}$: Tag-Write(c')

  We write $\mathcal{T}(c') \leftarrow$ Ref-Tag$(\mathcal{R}(parm, pk), \mathcal{T}(c))$.

In the Setup-Tag algorithm, a tag identifier id is encoded after concatenated with its MAC value, which can be generated with the knowledge of K_{MAC}. This ensures that $\mathcal{R}$ will be able to correctly identify $\mathcal{T}$ which has been setup by itself. We assume that the reader and tag perform both the Idt-Tag and Ref-Tag for privacy-enhanced tag identification, but two protocols can be used separately for its own purpose.

6.3.2 Privacy analysis

It is straightforward to see that the proposed RFID-R scheme fails to provide privacy against the *wide* adversary. This is because such a adversary can distinguish the tag $\mathcal{T}_b$ by writing a garbage value into one of the tags $\mathcal{T}_0$ and $\mathcal{T}_1$ in the privacy experiment. We now prove that the proposed RFID-R scheme satisfies (*discrete*, *narrow*, *strong*, ω)-privacy.

Theorem 6.1 *The RFID-R system provides (discrete, narrow, strong, ω)-privacy if the DDH is hard relative to $\mathcal{G}$.*

Proof. We provide a sketch of proof as below. Suppose that a simulator, SIM0 say, performs all its actions as in privacy experiment $\mathbf{Exp}^{\mathsf{privacy}}_{\mathcal{A}^\delta,\mathsf{RFID\text{-}R}}$ and the advantage of the adversary $\mathcal{A}^\delta$ is ϵ.

We then construct a simulator, SIM1 say, that performs all its actions as in SIM0 except for the production of the final ciphertext; it is replaced with a random value. We now show that the advantage of $\mathcal{A}^\delta$ in SIM1 is at most $\epsilon + \mathsf{Adv\text{-}DDH}$.[11] We first need to address how SIM1 knows which ciphertext would be the last ciphertext to be requested. Since the adversary can send at most ω queries, there has to be q_e queries which require the production of ciphertext. SIM1 can thus set to replace with a random value the q_e-th ciphertext that has to be produced. The queries that require to produce ciphetexts are `Create-Tag`, `Send-Reader`, and `Execute`.

First, the adversary can distinguish with the probability at most Adv-DDH between a random number and a ciphertext created from a tag identifier when the `Create-Tag` is queried, due to the fact that the security of underlying ElGamal encryption scheme can be reduced to the DDH problem [134]. The `Execute` query is a set of other oracle queries, where `Send-Reader` is only query that requires to produce ciphertexts using the Ref-Tag algorithm. When the ciphertext is replaced with a random number, the adversary can distinguish between them with the probability at most Adv-DDH. Since Ref-Tag outputs $c' = (my^{k+k'}, g^{k+k'})$ given $c = (my^k, g^k)$ as an input, where $k' \in_{\mathsf{R}} \mathbb{Z}_q^*$, the adversary has the same advantage as when it is challenged when the ciphertext is produced using the `Create-Tag` query. We note that the `Result` query is not allowed to the given type of adversary, and the `Corrupt` query simply reveals the ciphertext stored in a tag.

We next construct a simulator, SIM2 say, that performs all its actions as in SIM1 except for the production of the next-to-last ciphertext; it is replaced with a random value. As described above, we can show that the advantage of $\mathcal{A}^\delta$ in SIM2 is at most $\epsilon + 2\mathsf{Adv\text{-}DDH}$.

All the ciphertexts can be eventually replaced with random values, in which case the adversary never receives a ciphertext that depends on a tag identifier. Furthermore, the *discrete* type of adversary always gets two different random numbers for two consecutive reads of the same tag. The adversary thus has no advantage in $\mathsf{SIM}q_e$.

[11] See Definition 2.5 for Adv-DDH.

We now see that $\epsilon + q_e \cdot \mathsf{Adv\text{-}DDH}$ is negligible and thus ϵ is negligible. □

The security of the MAC scheme does not affect the privacy of the RFID-R scheme. That is, even if unsecure MAC scheme is used, the privacy of the RFID-R scheme is preserved. By using secure MAC scheme, a reader can determine if an interrogated tag has been issued by itself by checking $\tilde{m}$ (see Figure 6.10). The use of MAC scheme will be further discussed in Section 6.4.

6.3.3 Further discussions

Secret key encryption scheme, e.g. AES-CTR, may also be used as a refresh-based RFID system, which could deliver a significant reduction in computation cost for the reader and the reduced size of ciphertexts to be stored in the tags.

Public key encryption schemes, however, give the following advantages when used as refresh-based RFID schemes. That is, the keys used for identification and refresh can be separately managed. The private key used for identification remains in the central server, and the public key used for refresh process can be transferred to the third party or a stand-alone transceiver which is entitled to enhance privacy of tags. For example, items tagged with RFID tags can be frequently refreshed by transceivers using the public key during transport.

On the other hand, the secret key should not be known to the third party when using secret key encryption scheme. When such a key is transferred to a stand-alone transceiver, the secret key could be compromised because the transceivers might be often deployed in the hostile environments. Even though the secret key only remains in the central server, the connectivity may be not always guaranteed between the server and transceivers, and the central server could be the communication bottleneck.

6.4 Application to the EPCglobal Network

A key issue is whether or not RFID tags in the EPCglobal Network can perform onboard cryptographic operations, in order to enhance privcy. The main obstacle

to providing such functionality is cost; EPC tags are likely to be deployed on a very large scale once the unit price drops to \$0.05 [77]. In this case, Moore's law will not necessarily deliver the increases in computing power necessary to enable cryptographic functionality, since commercial pressures will push the industry to use one-cent tags when they become available, rather than continuing to pay five cents per tag and adding advanced security features.

The Gen2 standard [70] specifies the physical and logical requirements for the RFID system used in the EPCglobal Network. Given their small size, the most inexpensive EPC tags are likely to only have between 250 to 1,000 gates available for security features [123]. Another estimate indicates that no more than 2,000 gate equivalents (GEs) are available for security functionality in general EPC tags [81]. In this section we investigate the applicability of the discussed RFID scheme to the EPCglobal Network.

6.4.1 Existing schemes

The key-search based schemes described in Section 5.3 use either hash functions or secret key encryption algorithms, since these functions can be readily used to produce other necessary crypto-fuctions, such as message authentication codes (MACs) or pseudorandom number generators (PRNGs). In particular, an on-tag hash function is often assumed in RFID security protocols, perhaps because the high throughput possible for dedicated hash functions suggests that hash functions could be implemented even on very computationally limited platforms such as RFID tags. However, current hash functions are not suitable for implementation on low-cost basic tags, as shown in [16, 44].

Even one of the most compact dedicated hash functions, namely MD4, requires as many as 7350 GEs for 80-bit security.[12] Thus, since current dedicated hash functions are either too complex or broken, it would be desirable to consider hash functions built from compact block ciphers. Bogdanov et al. [16] investigate the performance of a variety of hash functions based on direct application of the most compact block cipher known to the author (the block cipher PRESENT), and the results are summarised in Table 6.5. Replacing PRESENT with a different block cipher is likely

[12] MD4 is, however, considered broken by the vast majority of the cryptographic community.

Table 6.5: Performance of secret key crypto-algorithms [16]

Block ciphers	Key size	Block size	Cycles per block	Throughput (Kbps)	Logic process	Area GEs
PRESENT-80 [15]	80	64	32	200	0.18μm	1570
PRESENT-80 [119]	80	64	563	11.4	0.18μm	1075
DES [94]	56	64	144	44.4	0.18μm	2309
PRESENT-128 [15]	128	64	32	200	0.18μm	1886
AES-128 [43]	128	128	1032	12.4	0.35μm	3400
Hash functions	**Output size**	**Data path size**	**Cycles per block**	**Throughput (Kbps)**	**Logic process**	**Area GEs**
MD4 [44]	128	32	456	112.28	0.13μm	7350
MD5 [44]	128	32	612	83.66	0.13μm	8400
SHA-1 [44]	160	32	1274	40.19	0.35μm	8120
SHA-256[44]	256	32	1128	45.39	0.35μm	10868
H-PRESENT-128 [16]	128	128	32	200	0.18μm	4256
H-PRESENT-128 [16]	128	8	559	11.45	0.18μm	2330
C-PRESENT-128 [16]	192	192	108	59.26	0.18μm	8048
C-PRESENT-128 [16]	192	12	3338	1.9	0.18μm	4600

to increase the space required for an implementation.

Table 6.5 shows that, with today's technology, compact block ciphers are more efficient than hash functions in RFID implementations. Furthermore, when using hash functions in a security protocol, care should be taken regarding the security properties required for hash functions; a hash function with an n-bit output can offer a 2^n-bit security level for pre-image and second pre-image resistance, and a $2^{n/2}$-bit security level for collision resistance.

Even though compact block ciphers such as PRESENT-80[13] could be implemented on EPC tags, the use of secret key cryptographic primitives in RFID systems often requires an (optimised) exhaustive key search by the reader, and this could be not appropriate for potentially large-scale RFID applications such as the EPCglobal Network. Furthermore, the use of all the RFID schemes discussed in Section 5.3 and Section 5.5 in the EPC tags require the significant changes of the current standard [70].

The previously proposed refresh-based schemes, i.e. Golle et al. [62] and Ateniese et al. [2], comply to the Gen2 standard [70], i.e. the encrypted version of the EPC can

[13]This block cipher is designed to have a 64-bit block size and 80-bit key length.

Table 6.6: Security and privacy comparisons of RFID systems

	Class-1 Gen-2 Standard [70]	Golle et al.'s UR scheme [62]	Ateniese et al.'s IE scheme [2]	The RFID-R system
Data Privacy	X	O	O	O
Location Privacy	X	X	△	O
Authenticity	△	X	X	O
Availability	O	X	X	O

The notations X, △, and O denote *no protection*, *partial protection*, and *full-protection*, respectively. Privacy is only satisfied against the *discrete* adversary. Gelle et al. and Ateniese et al. sacrifice authenticity and availability in order to enable randomisation of tag pseudonyms by any transceivers with the underlying system parameters, i.e. without knowledge of private/public key pairs; neither the Gen2 standard nor our proposed scheme cannot provide this property.

be written into the EPC partition and the access password can be set to be zero to allow any reader to refresh the encrypted EPC. Golle et al.'s scheme, however, cannot provide location privacy as described in section 5.4.2, and Ateniese et al.'s scheme also fails to provide location privacy against insider attackers, e.g. competitors in supply chains, which are practically most serious adversary as discussed in Section 5.4.3.

6.4.2 Proposed scheme

In this section we investigate the compatibility of the proposed scheme to Gen2 standard [70], and then discuss how the RFID-R scheme uses and extends the core set of features of the Gen2 standard [70] to satisfy privacy and security requirements discussed in Section 6.2.3. We summarise the comparison between the proposed scheme, Gen2 standard, and existing refresh-based RFID systems in Table 6.6.

Compatibility to Gen2 standard

Similarly to other refresh-based RFID schemes [2, 62], the tag identifier *id* corresponds to the EPC, and the encrypted version of EPC can be written into the EPC partition of the tag storage instead of EPC. Given that 80-bit security appears to be a reasonable target for RFID tag applications,[14] the RFID-R system can use an

[14]This security level is adopted by the eSTREAM project, details of which are available at http://www.ecrypt. eu.org/stream.

underlying finite field $\mathbb{F}_p$ with $|p| = 192$ or $\mathbb{F}_{2^m}$ with $|m| = 163$.[15] The result of such a choice would mean that the length of an encrypted EPC would be less than 416 bits, which the Gen2 standard sets as the maximum length for an EPC. The tag-access key ak corresponds to the access password of the standard, which is stored in the Reserved partition of tag storage. The Tag-Read and Tag-Write protocols, two basic building blocks of the protocols of the RFID-R system, corresponds to the protocol used during inventory process and the combination of the `Access` and `Write` commands in the access process, respectively.

Example use case of the RFID-R scheme

Unlikely previous refresh-based RFID schemes [2, 62], we suggest to implement access control for tag-writing to enhance authenticity and availability. Otherwise, an adversary could maliciously write into tags garbage values making attacked tags permanently desynchronised from readers. The tag-specific access keys could also be used for tag authentication discussed in Section 6.2.

Naive introduction of such tag-specific access key, however, directly infringes privacy. If the association between a tag identifier id (or an item bearing the tag) and a tag-access key ak is compromised, an adversary could trace the associated tag by broadcasting the tag-access key and, if any tag responds, that tag can be identified as the owner of the tag identifier id.

We thus suggest to use the RFID-R scheme in order to enhance security and privacy of EPC tags as follows. The EPC partition is configured to be universally readable and keyed writable, i.e. pwd-write field set to 1 (see Table 6.1), and the Reserved partition is configured to be keyed readable/writable, i.e. pwd-read/write field set to be 1 (see Table 6.1). The central database maintains the list of (EPC, ak), and the tag-identification process can be implemented as follows.

The reader identifies the tag using the Idt-Tag protocol and recovers (EPC, ak). The reader then refreshes the interrogated tag using Ref-Tag algorithm with the following extension. Instead of performing the Tag-Write(c'), a reader could authenticate a tag by testing whether the tag can be successfully written using its tag-access key,

[15] See http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf.

i.e. performing Tag-Write(c', ak) and checking if c' is successfully written. However, in order to detect an attack in which a cloned tag always indicates that the wiring process has been successful, the reader may send the tag a series of write instructions, one of which contains the correct tag-access key and the rest of which involve invalid tag-access keys, as described in Section 6.2. Finally, if the tag is successfully authenticated, the reader writes into the Reserved partition the tag-access key ak' which is newly generated uniformly at random. After receiving from the tag the acknowledgement that writing of ak' has been successful, the reader also updates (EPC, ak') in the database.

For a large number of write instructions, i.e. the number of spurious passwords, the protocol can be time-consuming; however, small number, even $\varrho = 2$, would suffice to detect the casual introduction of cloned tags (see Figure 6.8). For example, as described in section 6.2.3, drug counterfeiting is a particularly serious issue in supply chains [77]. The detection of a single counterfeit tag among the tags associated with a batch of drugs would be sufficient to cast doubt about the provenance of the entire batch.

Privacy analysis

Confidentiality (or data privacy), as defined in Requirement 6.1, is guaranteed through the universal semantic security (USS) property of the underlying ElGamal encryption scheme. That is, if an EPC is encrypted using the ElGamal scheme, then the USS property guarantees that no information about the EPC can be learned by an adversary.

With respect to anonymity (or location privacy), as in Requirement 6.2, the adversary should be able to link either a tag-access key or the encrypted version of EPC between two reads of the same tag. Since the tag-access key is always randomly generated, the adversary is now left to link two ciphertexts. This, however, is not possible as shown in Section 6.3.2.

It is, however, clear that tags can be tracked between refreshes, since the tags store universally-readable static values. This issue is inherent to refresh-based RFID systems. Such threats, however, can be mitigated in practice. Correlations between inbound/outbound flows in supply chains could reveal sensitive information that

compromises corporate privacy; however, the tracing of tags at an isolated location poses relatively little threat. A certain level of *corporate* location privacy can be provided by refreshing ciphertexts in tags before they leave the supply chain premises.[16] For *individual* location privacy, personal mobile RFID devices can reduce the risk of malicious tracing by refreshing tags frequently, with the precise refresh rate depending on the user's security policy.

When the RFID-R system is applied to the EPCglobal Network, we assume that an adversary cannot gain access to a *decryption oracle query* for a chosen ciphertext. To see why this is true, we first observe that we can reasonably assume that an authorised entity only has access to the decryption algorithm Dec (see Section 6.3). Now suppose that an adversary is somehow able to make a decryption oracle query with a chosen ciphertext. However, if the submitted ciphertext has not been constructed by the legitimate issuer, i.e. the tag owner, the output from the Idt-Tag algorithm will be an error message. This is because, when a ciphertext is decrypted, the plaintext should contain a valid MAC value.

Authenticity

In any refresh-based RFID system it is always possible to perform a cloning attack by obtaining tag pseudonyms from target tags either passively or actively and then writing these pseudonyms into field-programmable tags. It is then impossible to detect such counterfeit tags, since readers deem a tag to be legitimate as long as the pseudonym resident in a tag is valid.

We thus introduced a tag-authentication scheme using the technique proposed by Juels [76]. Our proposed protocol, however, provides a robust mechanism for detecting counterfeit tags produced by an attack of the type described by Juels [76]. Suppose that an adversary obtains the pseudonym c and the tag-access key ak of $\mathcal{T}$, e.g. by eavesdropping on the Ref-Tag algorithm performed between the tag $\mathcal{T}$ and a reader. The adversary can now produce a counterfeit tag $\tilde{\mathcal{T}}$ by writing c and ak into it. However, once the genuine tag $\mathcal{T}$ performs the suggested authentication protocol,

[16] RFID tags using ultra high frequency (UHF) communications are likely to be used for item-level tagging in supply chains, and the radio signal for such tags cannot penetrate metal. Thus leakage of information regarding inbound/outbound flows can be greatly reduced if items are carried in metal containers.

the access key stored in $\tilde{\mathcal{T}}$ becomes invalid. It is, of course, possible that the genuine tag $\mathcal{T}$ could be deemed to be a cloned tag if $\tilde{\mathcal{T}}$ performs the authentication protocol first. However, whenever a tag is identified as having been cloned, $\mathcal{R}$ could choose to classify as *tainted* the decrypted identifier from the pseudonym stored in the cloned tag, and then track the tags having this tainted identifier.

We point out that the proposed authentication scheme is not a *standard* challenge-response authentication protocol. Specifically, the proposed authentication mechanism cannot prevent the tag counterfeiting attacks, i.e. not fully satisfying Requirement 6.3, but can only detect that such attacks have been attempted. The proposed scheme thus would fail to satisfy the security definitions designed for authentication. Nevertheless, for inexpensive tags incapable of implementing such standard challenge-response authentication protocols, the proposed algorithm provides a pragmatic approach to enhancing tag authenticity.

Availability

As the Gen2 standard [70], our proposed scheme provides availability, as in Requirement 6.4, by implementing access-control for tag writing and thus preventing an adversary from maliciously writing into tags garbage values for permanent desynchronisation with the reader. The proposed scheme, however, provides a more robust level of protection even when the database of subscribers is compromised. That is, with the compromised list of EPC and its access/kill passwords the adversary could maliciously write into the tags the garbage values or even permanently deactivate the tags by simply reading the EPC from the tag and searching the corresponding passwords from the compromised database. With the proposed scheme, since EPCs are now stored as an encrypted form, an adversary cannot determine which access/kill passwords to use.

6.5 Conclusion

EPC tags are likely to become very widely used in the very near future, despite the fact that current tags have very limited computational capabilities and hence

cannot implement strong cryptographic primitives. Hence, there is a need to achieve the optimum level of security and privacy possible realisable with the capabilities of current tags. The proposed RFID system is directed at this endeavour.

Chapter 7

Concluding Remarks

Contents

In this chapter we describe main findings in this thesis and outline some possible directions for further research.

7.1 Main Research Findings

We have studied security and privacy issues in two classes of pervasive networks, namely Personal Area Networks and RFID-enabled EPCglobal Networks.

A number of key management schemes have been proposed for use in PANs, but these schemes only support key management within a PAN. We defined system models and design goals for key management within and between PANs, and proposed a novel security initialisation scheme for use in such networks. The proposed scheme achieves desirable security and efficiency properties by making use of the unique characteristics of iPANs.

We also constructed a formal privacy model for RFID systems accurately reflecting adversarial threats and power. We then gave brief privacy analysis for the existing privacy-enhanced RFID schemes which have received wide attention in the literature. We then constructed a secure refresh-based RFID system based on re-encryption techniques, and proved its privacy using the defined privacy model. Finally, we showed that the proposed scheme can greatly enhance the security and privacy of EPC tags, making the maximum use of given tag functionalities as specified in the

standards.

7.2 Future Research Directions

More networks seamlessly connect to other networks in pervasive computing, rather than existing as isolated networks. As society is becoming more densely connected, new research opportunities are arising to investigate the provision of security and privacy in future networks.

In this connection the EU has launched a number of research projects on the Future Internet [32, 115], aiming to overcome the current limitations of the Internet and to provide emerging integration services accommodating various types of devices. Clearly, security, privacy, and trust must be included in all aspects of the design and development of the Future Internet.

Since 2007, the EU's FP7 (7th Framework Programme) has supported a range of projects on the Future Internet [32]. Trustworthy ICT is a security project under the work programme Pervasive and Trusted Network and Service Infrastructures, and focuses on trustworthy network/services infrastructure, standardisation, and authentication models. More recently, the EU Future Internet PPP (Public Private Partnership) project started with the active involvement of industry [115]. This project focuses on delivering transparent application services on a common consolidated platform. Such applications include Smart Grid, Intelligent Transportation, e-Health, and m-commerce. The PPP project covers the development of a strategy for security provisioning, where the infrastructure must deliver optimal levels of security, privacy, and trust that match the dynamic context of the Future Internet.

However, while it is widely recognised that the future Internet requires built-in security mechanisms, the appropriate adversary model is far less clear. That is, while the current problems are relatively well known, it is not obvious which threats the Future Internet might face. Thus identifying the adversarial model and anticipating the emerging threats is a fundamentally important first step in building a secure Future Internet. Only when the community has a solid understanding of the threats in the Future Internet can appropriate countermeasures be designed.

Bibliography

[1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.

[2] G. Ateniese, J. Camenisch, and B. Medeiros. Untraceable RFID tags via insubvertible encryption. In V. Atluri, C. Meadows, and A. Juels, editors, *Proceedings of the 12th ACM Conference on Computer and Communications Security* (CCS 2005), pages 92–101. ACM, 2005.

[3] K. A. Atkinson, editor. *An Introduction to Numerical Analysis* (*2nd ed.*). John Wiley & Sons, New York, 1989.

[4] G. Avoine. Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049, 2005, Available at http://eprint.iacr.org/2005/049.

[5] G. Avoine. RFID security & privacy lounge. Available at http://www.avoine.net/rfid/.

[6] G. Avoine. Privacy issues in RFID banknotes protection schemes. In J-J Quisquater, P. Paradinas, Y. Deswarte, and A. A. E. Kalam, editors, *Proceedings of 6th International Conference on Smart Card Research and Advanced Applications* (CARDIS 2004), pages 33–48. Kluwer, 2004.

[7] G. Avoine, E. Dysli, and P. Oechslin. Reducing time complexity in RFID systems. In B. Preneel and S. E. Tavares, editors, *Proceedings of 12th International Workshop on Selected Areas in Cryptography* (SAC 2005), volume 3897 of *LNCS*, pages 291–306. Springer, 2006.

[8] G. Avoine, K. Kalach, and J-J Quisquater. ePassport: Securing international contacts with contactless chips. In G. Tsudik, editor, *Financial Cryptography 2008*, volume 5143 of *LNCS*, pages 141–155. Springer, 2008.

[9] G. Avoine and P. Oechslin. A scalable and provably secure hash-based RFID protocol. *Proceedings of third IEEE International Conference on Pervasive Computing and Communications Workshops* (PerComW05), pages 110–114. IEEE Computer Society, 2005.

[10] G. Avoine and P. Oechslin. RFID traceability: A multilayer problem. In A. S. Patrick and M. Yung, editors, *Financial Cryptography 2005*, volume 3570 of *LNCS*, pages 125–140. Springer, 2005.

[11] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of the Network and Distributed System Security Symposium* (NDSS 2002). The Internet Society, 2002.

[12] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In C. Boyd, editor, *Proceedings of Advances in Cryptology — ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 566–582. Springer, 2001.

[13] M. Bellare and B. Yee. Forward-security in private-key cryptography. In J. Marc, editor, *Proceedings of the 2003 RSA conference on The cryptographers' track* (CT-RSA'03), volume 2612 of *LNCS*, pages 1–18. Springer, 2003.

[14] E. R. Berlekamp, R. J. McEliece, and V. Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24:384–386, 1978.

[15] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In P. Paillier and I. Verbauwhede, editors, *Proceedings of 9th International Workshop on Cryptographic Hardware and Embedded Systems* (CHES 2007), volume 4727 of *LNCS*, pages 450–466. Springer, 2007.

[16] A. Bogdanov, G. Leander, C. Paar, and A. Poschmann. Hash functions and RFID tags: Mind the gap. In E. Oswald and P. Rohatgi, editors, *Proceedings of 10th International Workshop on Cryptographic Hardware and Embedded Systems* (CHES 2008), volume 5154 of *LNCS*, pages 283–299. Springer, 2008.

[17] D. Boneh, R. A. DeMillo, and R. J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In W. Fumy, editor,

Proceedings of Advances in Cryptology — EUROCRYPT 1997, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.

[18] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Proceedings of Advances in Cryptology — CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001.

[19] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[20] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *Proceedings of Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 514 – 532. Springer, 2001.

[21] J. Bringer, H. Chabanne, and E. Dottax. HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks. *Proceedings of Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing* (SecPerU 2006), pages 28–33, IEEE Computer Society, 2006.

[22] R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3):123–131, 2005.

[23] M. Burmester, T. Le, and B. Medeiros. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. *Proceedings of Second International Conference on Security and Privacy in Communication Networks* (SecureComm), pages 1–9, IEEE Computer Society, 2006.

[24] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. *Proceedings of 42nd Annual Symposium on Foundations of Computer Science* (FOCS 2001), pages 136–145, IEEE Computer Society, 2001.

[25] M. Cerecedo, T. Matsumoto, and H. Imai. Efficient and secure multiparty generation of digital signatures based on discrete logarithms. *IEICE Trans. Fundamentals*, E76-A(4):532–545, 1993.

[26] M. Čagalj, S. Čapkun, and J. P. Hubaux. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE*, 94(2):467–478, 2006.

[27] J. Cho. Bootstrapping security in personal area networks. *Journal of Information Assurance and Security*, 2(4):303–310, December 2008.

[28] J. Cho. Practical and robust self-keying scheme for personal area networks. *Proceedings of 2th IEEE International Conference on Digital Information Management* (ICDIM 2007), pages 493–499, IEEE, 2007.

[29] J. Cho. On refresh-based tag identification schemes. *Proceedings of 6th IEEE Consumer Communications and Networking Conference* (CCNC 2009), pages 1–5, IEEE, 2009.

[30] J. Cho. Strengthening Class1 Gen2 RFID tags. *Proceedings of 6th IEEE International Conference on Mobile Ad-hoc and Sensor Systems* (MASS 2009), pages 818–824, IEEE Computer Society, 2009.

[31] C. Cocks. An identity based encryption scheme based on quadratic residues. In B. Honary, editor, *Proceedings of 8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 360–363. Springer, 2001.

[32] European Commission. FP7 (7th Framework Programme). Available at http://www.future-internet.eu.

[33] M. Conti, R. D. Pietro, L. V. Mancini, and A. Spognardi. RIPP-FS: An RFID Identification, Privacy Preserving Protocol with Forward Secrecy. *Proceedings of 4Fifth Annual IEEE International Conference on Pervasive Computing and Communications* (PerCom 2007), pages 229–234, IEEE Computer Society, 2007.

[34] H. Deng and D. P. Agrawal. TIDS: Threshold and identity-based security scheme for wireless ad hoc networks. *Ad Hoc Networks*, 2(3):291–307, 2004.

[35] A. W. Dent and C. J. Mitchell, editors. *User's Guide To Cryptography And Standards*. Artech House, Inc., Norwood. MA. USA, 2004.

[36] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In G. Brassard, editor, *Proceedings of Advances in Cryptology - CRYPTO '89*, volume 435 of *LNCS*, pages 307–315. Springer, 1990.

[37] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions in Information Theory*, 22(6):644–654, November 1976.

[38] W. Diffie, P. C. van Oorschot, and M. J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2), June 1992.

[39] D. Dolev and A. C. Yao. On the security of public key protocols. *Proceedings of the IEEE 22nd Annual Symposium on Foundations of Computer Science*, pages 350–357, IEEE, 1981.

[40] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[41] C. M. Ellison and S. Dohrmann. Public-key support for group collaboration. *ACM Transactions on Information and System Security*, 6(4):547–565, 2003.

[42] P. Eronen and H. Tschofenig. Pre-shared key ciphersuites for Transport Layer Security (TLS). RFC 4279, December 2005.

[43] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In M. Joye and J-J Quisquater, editors, *Proceedings of 6th International Workshop Cryptographic Hardware and Embedded Systems* (CHES 2004), volume 3156 of *LNCS*, pages 357–370. Springer, 2004.

[44] M. Feldhofer and C. Rechberger. A case against currently used hash functions in RFID protocols. In R. Meersman, Z. Tari, and P. Herrero, editors, *Prceedings of On the Move to Meaningful Internet Systems 2006* (OTM 2006 Workshop), volume 4277 of *LNCS*, pages 372–381. Springer, 2006.

[45] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. *Proceedings of 28th Annual Symposium on Foundations of Computer Science* (FOCS), pages 427–437, IEEE, 1987.

[46] International Organization for Standardization. ISO/IEC 1st CD 9798-6, Information technology — Security techniques — Entity authentication — Part 6: Mechanisms using manual data transfer. December, 2003.

[47] NFC Forum. Connection Handover Technical Specification. Available at http://www.nfc-forum.org/specs/spec_list/.

[48] Y. Frankel, P. Gemmell, P. Mackenzie, and M. Yung. Optimal resilience proactive public-key cryptosystems. *Proceedings of 38nd Annual Symposium on Foundations of Computer Science* (FOCS 1997), pages 384–393, IEEE Computer Society, 1997.

[49] M. Frodigh, P. Johansson, and P. Larsson. Wireless ad hoc networking — the art of networking without a network (Ericsson Review). Available at http://www.ericsson.com/ericsson/corpinfo/publications/ review/2000_04/ 124.shtml, April 2000.

[50] D. Frumkin and A. Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. *Workshop on RFID Security – RFIDSec'09*, July 2009.

[51] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption scheme.

[52] C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *Cryptobytes*, 7(1):29–37, Spring 2004.

[53] C. Gehrmann and K. Nyberg. Securty in personal area networks. In C. J. Mitchell, editor, *Security for Mobility*, chapter 9, pages 191–230. IEE, London, 2004.

[54] C. Gehrmann, K. Nyberg, and C. Mitchell. The personal CA — PKI for a Personal Area Network. *Proceedings of IST Mobile & Wireless Communications Summit 2002*, pages 31–35, June 2002.

[55] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. In U. Maurer, editor, *Proceedings of Advances in Cryptology — EUROCRYPT '96*, volume 1070 of *LNCS*, pages 353–371. Springer, 1996.

[56] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystem. In J. Stern, editor, *Proceedings of Advances in Cryptology — EUROCRYPT '99*, LNCS, pages 295–310. Springer, 1999.

[57] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, Winter 2007.

[58] H. Gilbert, M. Robshaw, and H. Sibert. Active attack against HB^+: a provably secure lightweight authentication protocol. *Electronics Letters*, 41(21):1169–1170, 2005.

[59] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[60] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[61] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, 17(2):281–308, 1988.

[62] P. Golle, M. Jakobsson, A. Juels, and P. F. Syverson. Universal re-encryption for mixnets. In T. Okamoto, editor, *Proceedings of Topics in Cryptology — CT-RSA 2004*, volume 2964 of *LNCS*, pages 163–178. Springer, 2004.

[63] P. Gutmann. Use of shared keys in the TLS protocol. Internet draft (expired), draft-ietf-tls-sharedkeys-02, October 2003.

[64] J. Halamka, A. Juels, A. Stubblefield, and J. Westhues. The security implications of VeriChip cloning. *Journal of the American Medical Informatics Association* (JAMIA), 13(5):601–607, November 2006.

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

[66] M. Halváč and T. Rosa. A Note on the Relay Attacks on e-Passports: The Case of Czech e-Passports. Cryptology ePrint Archive, Report 2007/244, 2007.

[67] M. Hellman. A cryptographic time-memory tradeoff. *IEEE Transactions on Information Theory*, IT-26:401–406, 1980.

[68] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. In D. Coppersmith, editor, *Proceedings of Advances in Cryptology — CRYPTO '95*, volume 963 of *LNCS*, pages 339–352. Springer, 1995.

[69] EPCglobal Inc. The EPCglobal Network: Overview of design, benefits, and security. Available at http://www.epcglobalinc.org/about/media_centre/ Network_Security_Final.pdf, September 2004.

[70] EPCglobal Inc. Specification for RFID air interface. Available at http://www.epcglobalinc.org/standards/uhfc1g2/uhfc1g2_1_1_0-standard-20071017.pdf, December 2005.

[71] EPCglobal Inc. The EPCglobal architecture framework. Available at http://www.epcglobalinc.org/standards/architecture/architecture_1_2-framework-20070910.pdf, September 2007.

[72] VeriSign Inc. The EPCglobal Network: Ehancing the supply chains. Available at http://www.verisign.com/stellent/groups/public/documents/white_paper/002109.pdf, June 2005.

[73] VeriSign Inc. The EPCglobal Network: Maximizing the business benifits of RFID. Available at http://www.verisign.com/static/014025.pdf, June 2005.

[74] D. Johnson and D. Maltz. Dynamic sourcee routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Ad Hoc Wireless Networks*. Kluwer Academic Publishers, New York, 1996.

[75] A. Juels. Minimalist cryptography for low-cost RFID tags. In C. Blundo and S. Cimato, editors, *Proceedings of 4th International Conference on Security in Communication Networks* (SCN 2004), volume 3352 of *LNCS*, pages 149–164. Springer, 2005.

[76] A. Juels. Strengthening EPC tags against cloning. In M. Jakobsson and R. Poovendran, editors, *Proceedings of the ACM Workshop on Wireless Security* (WiSe 2005), pages 67–76. ACM, 2005.

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

[78] A. Juels, D. Molnar, and D. Wagner. Security and privacy issues in e-passports. *Proceedings of First International Conference on Security and Privacy for Emerging Areas in Communications Networks* (SecureComm 2005), pages 74–88, IEEE Computer Society, 2005.

[79] A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In R. N. Wright, editor, *Financial Cryptography 2003*, volume 2742 of *LNCS*, pages 103–121. Springer, 2003.

[80] A. Juels and S. Weis. Defining strong privacy for RFID. *Proceedings of International Conference on Pervasive Computing and Communications* (PerCom 2007), pages 342–347. IEEE Computer Society, 2007.

[81] A. Juels and S. A. Weis. Authenticating pervasive devices with human protocols. In V. Shoup, editor, *Proceedings of Advances in Cryptology — CRYPTO 2005*, volume 3621 of *LNCS*, pages 293–308. Springer, 2005.

[82] J. Katz and Y. Lindell. *Introduction to Modern Cryptogaphy.* Champman & Hall/CRC, 2008.

[83] J. Katz and J. Shin. Parallel and concurrent security of the HB and HB$^+$ protocols. In S. Vaudenay, editor, *Proceedings of Advances in Cryptology - EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 73–87. Springer, 2006.

[84] J. Katz and J. Shin. Modelling insider attacks on group key exchange protocols. *Proceedings of the 12th ACM Conference on Computer and Communications Security* (CCS 2005), pages 180–189, ACM, 2005.

[85] S. V. Kaya, E. Savas, A. Levi, and Ö. Erçetin. Public key cryptography based privacy preserving multi-context RFID infrastructure. *Ad Hoc Networks*, 7(1):136–152, 2009.

[86] Z. Kfir and A. Wool. Picking virtual pockets using relay attacks on contactless smartcard. *Proceedings of First International Conference on Security and Privacy for Emerging Areas in Communications Networks* (SecureComm 2005), pages 47–58. IEEE Computer Society, 2005.

[87] A. Khalili, J. Katz, and W. A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. *Proceedings of IEEE Security and Assurance in Ad-Hoc Networks at International Symposium on Applications and the Internet* (SAINT '03), pages 342–346. IEEE, 2003.

[88] I. Kim, E. Choi, and D. Lee. Secure mobile RFID system against privacy and security problems. *Proceedings of Third International Workshop onSecurity, Privacy and Trust in Pervasive and Ubiquitous Computing* (SecPerU 2007), pages 67–72. IEEE Computer Society, 2007.

[89] D. M. Konidala and K. Kim. Mobile RFID applications and security challenges. In M. Rhee and B. Lee, editors, *Proceedings of 9th International Conference on Information Security and Cryptology* (ICISC 2006), volume 4296 of *LNCS*, pages 194–205. Springer, 2006.

[90] E. Kosta, M. Meints, M. Hensen, and M. Gasson. An analysis of security and privacy issues relating to RFID enabled ePassports. In H. Venter, M. Eloff, L. Labuschagne, J. Eloff, and R. Von Solms, editors, *IFIP International Federation for Information Processing, New approaches for Security, Privacy and Trust in Complex Environments*, pages 467–472. Springer, 2007.

[91] S. Kumar and C. Paar. Are standards compliant elliptic curve cryptosystems feasible on RFID. *Workshop on RFID Security* (RFIDSec06), 2006. Available at http://events.iaik.tugraz.at/RFIDSec06.

[92] J. Larsson. Private communication with M. Jakobsson.

[93] T. V. Le, M. Burmester, and B. Medeiros. Universally composable and forward-secure RFID authentication and authenticated key exchange. In F. Bao and S. Miller, editors, *Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security* (ASIACCS 2007), pages 242–252. ACM, 2007.

[94] G. Leander, C. Paar, A. Poschmann, and K. Schramm. New lightweight DES variants. In A. Biryukov, editor, *Proceedings of the 14th International Workshop on Fast Software Encryption* (FSE 2007), volume 4593 of *LNCS*, pages 196–210. Springer, 2007.

[95] X. Leng, K. Mayes, and K. Markantonakis. HB-MP+ protocol: An improvement on the HB-MP protocol. *IEEE International Conference on RFID*, pages 118–124, April 2008.

[96] C. Li, T. Hwang, and N. Lee. (t, n) threshold signature schemes based on discrete logarithm. In R. Rueppel, editor, *Proceedings of Advances in Cryptology — EUROCRYPT '94*, volume 950 of *LNCS*, pages 191–200. Springer, 1994.

[97] C. Lim and T. Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In P. Ning, S. Qing, and N. Li, editors, *Proceedings of 8th International Conference on Information and Communications Security* (ICICS 2006), volume 4307 of *LNCS*, pages 1–20. Springer, 2006.

[98] D. P. Maher and N. H. Windham. Secure communication method and apparatus. United States Patent, No. 5450493, Sep. 1995.

[99] S. Martinez, M. Valls, C. Roig, F. Gine, and J. Miret. An elliptic curve and zero knowledge based forward secure RFID protocol. *Workshop on RFID Security* (RFIDSec07), 2007. Available at http://rfidsec07.etsit.uma.es/.

[100] M. McLoone and M. J. B. Robshaw. Public key cryptography and RFID tags. In M. Abe, editor, *Proceedings of Topics of Cryptology — CT-RSA 2007*, volume 4377 of *LNCS*, pages 372–384. Springer, 2007.

[101] A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[102] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001. Available at http://www.cacr.math.uwaterloo.ca/hac/.

[103] J. V. D. Merwe, D. Dawoud, and S. McDonald. A survey on peer-to-peer key management for mobile ad hoc networks. *ACM Computing Surveys*, 39(1):Article No. 1, 2007.

[104] C. J. Mitchell and R. Schaffelhofer. The personal PKI. In C. J. Mitchell, editor, *Security for Mobility*, chapter 3, pages 35–60. IEE, London, 2004.

[105] D. Molnar, A. Soppera, and D. Wagner. A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In B. Preneel and S. E. Tavares, editors, *12th International Workshop On Selected Areas in Cryptography* (SAC 2005), volume 3897 of *LNCS*, pages 276–290. Springer, 2006.

[106] D. Molnar and D. Wagner. Privacy and security in library RFID: issues, practices, and architectures. In V. Atluri, B. Pfitzmann, and P. D. McDaniel, editors, *Proceedings of 11th ACM Conference on Computer and Communications Security* (CCS 2004), pages 210–219. ACM, 2004.

[107] K. Nohl and D. Evans. Quantifying information leakage in tree-based hash protocols (short paper). In P. Ning, S. Qing, and N. Li, editors, *Proceedings of 8th International Conference on Information and Communications Security* (ICICS 2006), volume 4307 of *LNCS*, pages 228–237. Springer, 2006.

[108] M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic approach to privacy-friendly tags. *In RFID Privacy Workshop*, available at http://www.rfidprivacy.org/papers/ohkubo.pdf, 2003.

[109] K. Ouafi, R. Overbeck, and S. Vaudenay. On the security of HB# against a man-in-the-middle attack. In J. Pieprzyk, editor, *Proceedings of Advances in Cryptology - ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 108–124. Springer, 2008.

[110] N. Park, Y. Song, D. Won, and H. Kim. Multilateral approaches to the mobile RFID security problem using web service. In Y. Zhang, G. Yu, E. Bertino, and G. Xu, editors, *Proceedings of 10th Asia-Pacific Web Conference on Progress in WWW Research and Development* (APWeb 2008), volume 4976 of *LNCS*, pages 331–341. Springer, 2008.

[111] T. P. Pedersen. A threshold cryptosystem without a trusted party (extended abstract). In D. W. Davies, editor, *Proceedings of Advances in Cryptology - EUROCRYPT '91*, volume 547 of *LNCS*, pages 522–526. Springer, 1991.

[112] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, July 2003.

[113] B. Pfitzmann and M. Waidner. Composition and integrity preservation of secure reactive systems. *Proceedings of the 7th ACM Conference on Computer and Communications Security* (CCS 2000), pages 245–254, ACM, 2000.

[114] M. Philipose, K. P. Fishkin, D. Fox, H. Kautz, D. Patterson, and M. Perkowitz. Guide: Towards understanding daily life via auto-identification and statistical analysis. *Proceedings of the International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications* (Ubihealth 2003), Springer Verlage, 2003.

[115] European Future Internet Portal. White paper on the Future Internet PPP Definition. Available at http://www.future-internet.eu.

[116] R. L. Rivest. Chaffing and winnowing: Confidentiality without encryption. *CryptoBytes*, 4(1):12–17, 1998.

[117] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[118] C. M. Roberts. Radio Frequency Identification (RFID). *Computers & Security*, 25(1):18–26, 2006.

[119] C. Rolfes, A. Poschmann, G. Leander, and C. Paar. Ultra-lightweight implementations for smart devices — security for 1000 gate equivalents. In G. Grimaud and F-X Standaert, editors, *Proceedings of 8th IFIP WG 8.8/11.2 International Conferenc on Smart Card Research and Advanced Applications* (CARDIS 2008), volume 5189 of *LNCS*, pages 89–103. Springer, 2008.

[120] J. Saito, J. Ryou, and K. Sakurai. Enhancing privacy of universal re-encryption scheme for RFID tags. In L. T. Yang, M. Guo, G. R. Gao, and N. K. Jha, editors, *Proceedings of International Conference on Embedded and Ubiquitous Computing* (EUC 2004), volume 3207 of *LNCS*, pages 879–890. Springer, 2004.

[121] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. *Proceedings of the 2000 Symposium on Cryptography and Information Security* (SCIS 2000), pages 26–28, January 2000.

[122] N. B. Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson. Node cooperation in hybrid ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(4):365–376, 2006.

[123] S. E. Sarma, S. A. Weis, and D. W. Engels. Radio Frequency Identification: Security risks and challenges. *CryptoBytes*, 6(1):2–9, 2003.

[124] N. Saxena. Public key cryptography sans certificates in ad hoc networks. In J. Zhou, editor, *Proceedings of 4th International Conference on Applied Cryptography and Network Security* (ACNS 2006), volume 3989 of *LNCS*, pages 375–389. Springer-Verlag, 2006.

[125] B. Schechter. Seeing the light: IBM's vision of life beyond the PC. *IBM research*, No. 2, 2000.

[126] C-P Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[127] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and D. Chaum, editors, *Proceedings of Advances in Cryptology — CRYPTO '84*, volume 196 of *LNCS*, pages 47–53. Springer, 1985.

[128] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[129] V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In K. Nyberg, editor, *Proceedings of Advances in Cryptology — EUROCRYPT '98*, volume 1403 of *LNCS*, pages 1–16. Springer, 1998.

[130] Bluetooth SIG. Core specification version 2.1 + edr. Available at http://www.bluetooth.com/English/Technology/Works/Pages/Core_Specification_v21__EDR.aspx.

[131] Bluetooth SIG. Secure simple pairing whitepaper (2006-08-03). Available at http://www.bluetooth.com/NR/rdonlyres/0A0B3F36-D15F-4470-85A6-F2CCFA26F70F/0/SimplePairing_WP_V10r00.pdf.

[132] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In B. Christianson, B. Crispo, J. A. Malcolm, and M. Roe, editors, *Proceedings of 7th International Workshop on Security Protocols*, volume 1796 of *LNCS*, pages 172–194. Springer, 2000.

[133] IEEE standard 802.15.1. IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks – Specific requirements Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs).

[134] Y. Tsiounis and M. Yung. On the security of ElGamal based encryption. In H. Imai and Y. Zheng, editors, *Proceedings of First International Workshop on Practice and Theory in Public Key Cryptography* (PKC 1998), volume 1431 of *LNCS*, pages 117–134. Springer, 1998.

[135] G. Tsudik. A Family of Dunces: Trivial RFID Identification and Authentication Protocols. In N. Borisov and P. Golle, editors, *Proceedings of 7th International Symposium in Privacy Enhancing Technologies* (PEC 2007), volume 4776 of *LNCS*, pages 45–61. Springer, 2007.

[136] G. Tsudik. YA-TRAP: Yet Another Trivial RFID Authentication Protocol. *Proceedings of 4th IEEE Conference on Pervasive Computing and Communications Workshops* (PerCom 2006), pages 640–643, IEEE Computer Society, 2006.

[137] S. Vaudenay. On privacy models for RFID. In K. Kurosawa, editor, *Proceedings in Advances in Cryptology - ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 68–87. Springer, 2007.

[138] S. H. Weingart. Physical security devices for computer subsystems: A survey of attacks and defences. In Ç. K. Koç and C. Paar, editors, *Proceedings of 9th International Workshop on Cryptographic Hardware and Embedded Systems* (CHES 2000), volume 1965 of *LNCS*, pages 302–317. Springer, 2000.

[139] S. A. Weis. Security parallels between people and pervasive devices. *Proceedings of 3rd IEEE Conference on Pervasive Computing and Communications Workshops* (PerCom 2005), pages 105–109, IEEE Computer Society, 2005.

[140] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In D. Hutter, G. Müller, W. Stephan, and M. Ullmann, editors, *First International Conference on Security in Pervasive Computing* (SPC 2003), volume 2802 of *LNCS*, pages 201–212. Springer, 2004.

[141] J. Wolkerstorfer. Is elliptic-curve cryptography suitable to secure RFID tags? *ECRYPT Workshop on RFID and Lightweight Crypto*, 2005. Available at http://events.iaik.tugraz.at/RFIDandLightweightCrypto05.

[142] B. Yoon. HB-MP++ protocol: An ultra light-weight authentication protocol for RFID system. In *IEEE International Conference on RFID – RFID 2009*. IEEE, April 2009.

[143] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Anonymous communications in mobile ad hoc networks. *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM 2005), pages 1940–1951.

[144] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Securing mobile ad hoc networks with certificateless public keys. *IEEE Transactions on Dependable and Secure Computing*, 3(4):386–399, 2006.

[145] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.