Lightweight Cryptographic Group Key Management Protocols for the Internet of Things

Teklay Gebremichael

Department of Information Systems and Technology
Mid Sweden University

Licentiate Thesis No. 154
Sundsvall, Sweden
2019

Mittuniversitetet
Informationssystem och -teknologi
SE-851 70 Sundsvall
SWEDEN

ISBN 978-91-88527-91-2
ISNN 1652-8948

Akademisk avhandling som med tillstånd av Mittuniversitetet i Sundsvall framlägges till offentlig granskning för avläggande av teknologie licentiatexamen den 07 Mars 2019 klockan 11:00 i sal C326, Mittuniversitetet, Holmgatan 10, Sundsvall.

Tryck: Tryckeriet Mittuniversitetet

Abstract

The Internet of Things (IoT) is increasingly becoming an integral component of many applications in consumer, industrial and other areas. Notions such as smart industry, smart transport, and smart world are, in large part, enabled by IoT. At its core, the IoT is underpinned by a group of devices, such as sensors and actuators, working collaboratively to provide a required service. One of the important requirements most IoT applications are expected to satisfy is ensuring the security and privacy of users. Security is an umbrella term that encompasses notions such as confidentiality, integrity and privacy, that are typically achieved using cryptographic encryption techniques.

A special form of communication common in many IoT applications is group communication, where there are two or more recipients of a given message. In order to encrypt a message broadcast to a group, it is required that the participating parties agree on a group key a priori. Establishing and managing a group key in IoT environments, where devices are resources-constrained and groups are dynamic, is a non-trivial problem. The problem presents unique challenges with regard to constructing protocols from lightweight and secure primitives commensurate with the resource-constrained nature of devices and maintaining security as devices dynamically leave or join a group.

This thesis presents lightweight group key management protocols proposed to address the aforementioned problem, in a widely adopted model of a generic IoT network consisting of a gateway with reasonable computational power and a set of resource-constrained nodes. The aim of the group key management protocols is to enable the gateway and the set of resource-constrained devices to establish and manage a group key, which is then used to encrypt group messages. The main problems the protocols attempt to solve are establishing a group key among participating IoT devices in a secure and computationally feasible manner; enabling addition or removal of a device to the group in a security preserving manner; and enabling generation of a group session key in an efficient manner without re-running the protocol from scratch. The main challenge in designing such protocols is ensuring that the computations that a given IoT device performs as part of participating in the protocol are computationally feasible during initial group establishment, group key update, and adding or removing a node from the group.

The work presented in this thesis shows that the challenge can be overcome by

designing protocols from lightweight cryptographic primitives. Specifically, protocols that exploit the lightweight nature of crypto-systems based on elliptic curves and the perfect secrecy of the One Time Pad (OTP) are presented. The protocols are designed in such a way that a resource-constrained member node performs a constant number of computationally easy computations during all stages of the group key management process.

To demonstrate that the protocols are practically feasible, implementation result of one of the protocols is also presented, showing that the protocol outperforms similar state-of-the-art protocols with regard to energy consumption, execution time, memory usage and number of messages generated.

Acknowledgements

My supervisors, Mikael Gidlund and Ulf Jennehag, have greatly helped me during the writing of this thesis and the papers included herein. They've been a constant source of support and guidance and I'm incredibly grateful for that. Working with them has been an absolute pleasure, so much so that I wouldn't want to have it any other way.

Tingting Zhang has helped me significantly improve the thesis by reviewing the manuscript and providing me with helpful technical comments. I thank her a lot.

I extend my sincere thanks to all the colleagues at Mid Sweden University, some of whom I've worked with, and others I've had interesting discussions and friendships with. I'd like to particularly thank department head Patrik Österberg, and Annika Berggren, for their unreserved help regarding administrative aspects.

I'm grateful to Lisa Velander for reviewing the manuscript. She has pointed a number of errors out and provided me with suggestions for how to improve some of the text. Insofar as the thesis is error-free and readable, it's largely due to her. Of course, I take responsibility for errors which might have made it to the print.

This thesis was written under the SMART project. I'd like to thank all the companies who're behind the project in terms of funding.

I owe a debt of gratitude to Øyvind Ytrehus of the Simula research group at University of Bergen, Norway, and Gerhard Hancke at the City University of Hong Kong, Hong Kong, for hosting me and giving me opportunities to discuss and present my research to their respective research groups.

I'd like to express my gratitude to Andreas Jacobsson of Malmö University for accepting to be my thesis defence opponent.

Living in Sundsvall has been such a rewarding experience, and a huge part of the reason is my friends – of whom there are too many to mention. I'd like to thank them all for making me feel at home, putting up with my occasional propensity to become flippant on matters other people consider serious. I'd also like to thank all my friends outside Sundsvall.

I owe an undying debt of gratitude to my beloved family, specially my parents, who, even in their old age, continue to melt like a candle to shine light for me. I'd also like to thank Mizer Assefa for helping me during the initial stages of my study.

Contents

List of Papers

This thesis is mainly based on the following papers, herein referred to by their Roman numerals:

I **Lightweight IoT Group Key Establishment Scheme Using One-Way Accumulator**
Teklay Gebremichael, Ulf Jennehag, Mikael Gidlund,
In The International Symposium on Networks, Computers and Communications (ISNCC), Rome, Italy, June 2018.

II **Group-Key Establishment Protocol for IoT Devices: Implementation and Performance Analyses**
Nico Ferrari, **Teklay Gebremichael**, Ulf Jennehag, Mikael Gidlund,
In the Fifth International Conference on Internet of Things: Systems, Management and Security (IoTSMS), Valencia, Spain, October 2018.

III **Lightweight IoT Group Key Establishment Scheme From the One Time Pad**
Teklay Gebremichael, Ulf Jennehag, Mikael Gidlund,
To be presented *in the* 7^{th} *IEEE International Workshop of Security and Trust in Mobile Network*, San Francisco, California, USA, 2019.

IV **Survey of Proximity Based Authentication Mechanisms for the Industrial Internet of Things**
Umair Mujtaba Qureshi, Gerhard Petrus Hancke, **Teklay Gebremichael**, Ulf Jennehag, Stefan Forsström, Mikael Gidlund,
In the 44^{th} *Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Washington D.C., USA, October 21-23, 2018.

List of Figures

List of Tables

Terminology

Acronyms and Abbreviations

AES	Advanced Encryption Standard
CA	Certificate Authority
DDoS	Distributed Denial of Service
DoS	Denial of Service
DES	Data Encryption Standard
DH	Diffie-Hellman
DLP	Discrete Logarithm Problem
DS	Digital Signature
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
IFP	Integer Factorization Problem
IoT	Internet of Things
IIoT	Industrial Internet of Things
M2M	Machine-to-Machine
OTP	One Time Pad
TLS	Transport Layer Security
PKI	Public Key Infrastructure
PRF	Pseudorandom Function
6LowPan	IPv6 over Low Power Wireless Personal Area Network
WSN	Wireless Sensor Network

Mathematical Notations

$PrvKey, PubKey$	private key, public key pair
p	a prime number
F_q	a prime field of size p

$\oplus$	bit-wise XOR operator
P, Q	points on an elliptic Curve
n, s, y, v	random n-bit values
0	n-bit vector, with each bit set to 0
A	an array containing n-bit values in each entry
kP	scalar multiplication of an elliptic curve P by a scalar k

Chapter 1

Introduction

By embedding computational and communication capabilities into everyday objects [A+09], the conventional Internet is being extended in order that computation and intelligence become pervasive. Pervasive interconnection of everyday objects enables the creation of services which help realize notions such as smart cities, smart transport, and smart world [Sta14]. This move and paradigm of embedding intelligence and interconnecting everyday objects is generally called the *Internet of Things (IoT)* [WW10]. IoT represents a network of globally or locally identifiable everyday objects, their integration with the conventional Internet, the multiplicity of enabling protocols, infrastructures, applications built on top of the infrastructure, policies and regulations governing their operations [Sta14]. Some of the technologies that underlie IoT are Machine-to-Machine (M2M) communication [WSE14], Wireless Sensor Networks (WSN) [ASSC02] and RFID [Fin10]. IoT applications are built and deployed on top of these platforms to realize various applications such as home automation and home security management, smart energy monitoring and management, items and shipment tracking, surveillance and military, smart cities, health monitoring, and logistics monitoring and management [Raz13].

IoT applications consist mainly of a group of small devices with sensing and/or actuation capabilities, working collaboratively to provide a specific functionality. Collaboration is achieved by sending data from one or more devices in a network to another device or group of devices in the network. For instance, in a typical Industrial IoT (IIoT) application, a group of sensors monitoring a given component send their readings of the conditions of the machine to a control center. The control center, in collaboration with other entities, analyses the data and sends a command to a group of actuators to effect a desired outcome to ensure safe and normal operation of the component.

A common mode of communication in IoT applications is group communication whereby a group of two or more devices communicate with each other in such a way that a sender broadcasts a message to the group rather than sending unicast messages addressed to each device in the group. This has two advantages [KKT14]. First, since most IoT devices are resource-constrained, battery-powered, and with

limited computational capabilities, broadcasting a single message to a group is more economical to the sender than unicasting the message to each individual member. Second, group communication enables a fast delivery of a message to multiple recipients, a feature that is very important in time critical applications. For these reasons, group communication is a preferred mode of communication in many IoT applications.

1.1 Motivation

1.1.1 Group Communication in the IoT

In the conventional Internet, group communication is a common form of communication in various applications such as video conferencing [DGK+00], streaming applications [WK03], and other similar services [Mil99], where there are two or more recipients of a message from a given sender. The entities in a group are uniquely identified by an address, such as a multicast or broadcast address. The main rationale for using group communication is timely delivery of a message to multiple destinations while generating the minimum amount of traffic possible.

Group communication is particularly appealing in the IoT [HGMH+11]. IoT devices are resource-constrained, battery-powered and generally have limited computational capabilities. Since sending multiple unicast messages requires multiple processing, it is more economical for IoT devices to rely on broadcasting to save resources. Furthermore, by broadcasting a single message to multiple destinations, the problem of generating unnecessary traffic is avoided.

In addition to the aforementioned arguments in favor of group communication in the IoT, another reason why group communication is preferred is because of the peculiar requirements of IoT applications. A lot of IoT applications require that a message be sent to a group so that devices in the group act upon the message synchronously. Consider the following application scenarios:

- *Smart lighting*: A smart building may have its lighting devices grouped according to their location and connected to a switch, which acts as a gateway. It is important that the switch is able so send group messages to the devices to control lighting level and related functions.
- *Software update*: A gateway downloads a software update and simply broadcasts it to the group so that member nodes patch. The alternative is each device downloads the patch independently, which results in generating unnecessary traffic.
- *Emergency broadcast*: The control center of some automation may be forced to stop or start many devices in the process with a single command, minimizing time and resource requirements.
- *e-health*: A sensor implanted in a patient's body may broadcast readings to a group of receivers such as nurses, doctors and even chat servers.

This clearly shows that group communication in IoT is applicable in many application contexts. Given the nature of data that such IoT applications handle, certain security guarantees are required to maintain proper functioning of applications and maintain users' privacy. Security breaches in IoT applications could lead to a number of issues, ranging from exposing users' privacy to more serious dangers such as death in the case of applications which handle patients' medical data, or hazards in safety-critical IIoT applications.

1.1.2 Security Requirements in Group Communication

The security goals in group communication are versions of the security goals in a two party communication: confidentiality, integrity, non-repudiation, availability and entity authentication [KMVOV96]. Confidentiality in group communication deals with ensuring that a message intended to a group is not accessible to non-member parties [TVS07]; integrity is the requirement that every member of a given group receives a message unaltered [PBS+15]; non-repudiation service guarantees that a member of a group cannot, at a later time, deny having sent or received a message to/from the group [LX13]; entity authentication is a service that enables parties in communication to identify each other [KMVOV96]; and availability is giving a guarantee that the system continues to function even in the face of adversity and is usually achieved by having redundancy and other fault tolerance mechanisms.

Secure and privacy preserving group communication is achieved by satisfying one or more of the aforementioned security goals. The cryptographic tools employed to achieve the security goals such as encryption, digital signatures and message authentication codes typically require that a key or keying material be established between the parties in a group before secure communication can take place [KMVOV96]. Establishing a group key between IoT devices is a non-trivial problem, made even more complex due, in large part, to the resource-constrained and dynamic nature of IoT environments [RALS11]. A naive way to go about establishing keying material is to manually store the relevant information on each device in the communication group, an approach that becomes impractical for large and dynamic groups. A more realistic, albeit challenging, approach is to rely on cryptographic protocols that make it possible for a group of devices to establish a cryptographic key between themselves in a secure and computationally practical manner [RALS11].

1.1.3 Establishment and Management of Cryptographic Group Keys in IoT

One of the most significant works regarding key establishment is the Diffie-Hellman key exchange protocol [DH76], which enables two parties to share a secret value using public key cryptography. The protocol has been adapted to enable more than two parties to share a secret value [STW96]. However, the protocol uses computationally intensive public key cryptography that is not suitable for resource-constrained devices. Other key transport and management mechanisms based on symmetric

key encryption [RSSS16, MNG17], and others based on public key cryptography [KOO17, IOV+17, ÁBLLR16], have been proposed for the IoT. The key transport mechanisms based on symmetric key encryption [RSSS16, SO12] generally leverage a Trusted Third Party (TTP) to create, manage and securely send keys or keying information. Consequently, they do not scale well and are vulnerable to single point of attack and failure. Key establishment mechanisms based on public key cryptography [ÁBLLR16, IOV+17] partially address the above problems by relying on cryptographic primitives that enable collaborative secure key establishment.

Group key management protocols designed to be used in the conventional Internet do not generally take into account the peculiar nature of IoT devices and hence are not convenient for deployment in IoT. The need to design key management protocols tailored towards IoT domains has spurred a research in protocols based on lightweight cryptography [EKP+07]. Despite the significant amount of work in the literature to address this need [ÁBLLR16, IOV+17, RSSS16, SO12], there is still a need for secure key management and establishment schemes that rely on lightweight cryptography that address various complex issues related to key management that will be discussed in Chapter 3. This thesis work is an attempt to bridge some of the research gap in this regard regard by designing and implementing lightweight group key establishment and management schemes from various lightweight cryptographic constructions.

1.2 Purpose and Scope

The main purpose of the work presented in this thesis is to design secure and lightweight group key management schemes that could be used in many of the IoT applications mentioned previously. Designing a group key management scheme for IoT has many aspects, including functional requirements such as how a key or keying material is established between devices in a group, how session keys are securely generated and how nodes are added or removed from a group in a security preserving manner [JVW+14]. There are also other design aspects regarding what cryptographic primitives are used as building blocks, how lightweight and secure they are and how they are implemented and deployed on devices. There are protocols in the literature that address one or more of these issues to varying levels of degrees [KOO17, IOV+17, ÁBLLR16, ÁBLLR16, IOV+17]. The core of this thesis is a presentation of lightweight and secure group key management schemes that enable IoT devices in a group to establish, manage and generate session keys.

While IoT networks can be set up in different ways [HBK+14], the most commonly deployed architecture is one that consists of a gateway or a device with similar functionality and a set of end nodes connected to the gateway [ZWC+10]. As a result, this thesis considers the scenario where there is an IoT network consisting of a trusted entity such as a gateway and a group of two or more end devices. The choice of this particular model is motivated by the fact that it is the most common architecture in real world IoT application deployments [GBMP13]. For instance, in a typical industrial IoT (IIoT) application, there would be a controller and a group

of sensors and/or actuators sending sensed data to the controller and acting on data received from the controller. Furthermore, the proposed schemes under this model can easily be adapted to other models, such as a group of IoT devices without a gateway, all connected to the Internet or other external network, by having one of the devices assume the role of a gateway.

Problems related to ensuring reliable communication between devices, physical safety of devices, and issues related to how to practically implement and deploy cryptographic primitives on devices are beyond the scope of this thesis.

1.3 Research Questions and Objectives

Within the context of the purpose and scope stated in Section 1.2, the overall aim of this thesis is to construct lightweight and secure group key management schemes [1] that could be used in IoT environments. As discussed previously, this is a multi-faceted problem that involves making choices at non-technical levels regarding approach, and at technical levels regarding what cryptographic primitives protocols should be constructed from.

Motivated by the standard practice in the field [FSK11], the research objectives and questions this thesis deals with are geared towards how one can construct lightweight key management schemes from existing cryptographic primitives that are considered secure. To this end, the following goals and research questions have been formulated:

- **Research Goal 1 (RG1):** To construct a lightweight group key management scheme from lightweight public key cryptographic primitives. The goal here is to investigate possibilities for secure and lightweight key management constructions using public key primitives that are lightweight, such as elliptic curve based crypto-systems. As stated above, any key management construction should address one or more of the issues mentioned; that is, what underlying primitives are used and how lightweight they are, how security of the construction is proved, how the construction would be implemented, and what security assumptions and models are considered. To achieve this goal, the following research question is posed:
  - **Research Question 1 (RQ1)**: What existing public cryptographic primitives can be used, and in what way can they be combined, so that a secure and lightweight IoT group key management scheme is realized?
- **Research Goal 2 (RG2):** To construct a lightweight group key management scheme from lightweight symmetric key primitives. The One Time Pad (OTP) is a lightweight cryptographic primitive, owing to the fact that bit-wise XOR is the only operation it relies upon. Moreover, the OTP provides perfect secrecy [Ver26]. However, the OTP is not practically useful due to the requirement

[1] The words *scheme* and *protocol* are used interchangeably in this thesis.

that the key must at least be as long as the message to be encrypted [Sha49]. The research goal is to find ways of constructing IoT group key management schemes that rely on the lightweight nature and perfect secrecy guarantee of the OTP. The challenge is to find a workaround to the practical and theoretical limitations that are inherent to the OTP, while still exploiting the desirable features of it. To this end, the following research question is posed:

  - **Research Question 2 (RQ2)**: How can one use the perfect secrecy security guarantee provided by the OTP to construct a secure and lightweight IoT group key management scheme?

- **Research Goal 3 (RG3):** To evaluate the various authentication mechanisms used in IoT environments today and suggest new secure authentication mechanisms. To achieve **RG1** and **RG1** , it was assumed that there is a mechanism for IoT devices to authenticate each other. This assumption is valid to show theoretical constructions of unauthenticated key management schemes for IoT environments. In practice, however, one also needs to have a mechanism that enables two or more devices to mutually authenticate to each other. All other security objectives can only be achieved if the parties involved in communication know each other. The goal is to study and analyze all the authentication mechanisms in use today. To achieve this goal, the following research question is posed:

  - **Research Question 3 (RQ3):** What are the most commonly used proximity-based authentication mechanisms used in IoT environments today? What are the benefits and drawbacks of each mechanism and what kinds of authentication mechanisms should be used in the future?

1.4 Research Methodology

The research methodology used in designing the group key management protocols follows the standard research methodology in the design of security protocols. It was based on a combination of analytical, theoretical and experimental research. The analytical aspect of the research dealt with understanding the literature on group key management protocols and identifying problems, which were then formulated as research problems. It also included studying different security and adversarial models, identifying the ones that fit the security requirements of IoT applications and the kind of security attacks these applications are potentially subject to.

The theoretical part of the research included proposing and designing key management schemes from lightweight cryptographic primitives and justifying that the construction are secure in the following sense: given a security model consisting of security definitions, requirements and an adversarial model showing what a potential attacker can do, we consider a scheme is secure if an attacker with reasonable computational power cannot break it. That is the standard notion of security of any protocol [Yao82]. To construct secure schemes in this context, primitives that have been proved to be secure or whose security relies on mathematical problems

believed to be computationally hard [CKT91] were used as building blocks. Furthermore, proofs and justifications for why constructions consisting of more than one cryptographic primitives are considered secure are provided.

All the key management solutions proposed in this thesis were validated in terms of their performance, through implementation (**Paper II**) or through theoretical analysis (**Papers I** and **III**). Moreover, for the security schemes in all the publications, mathematical analysis is used to show the correctness of the protocols.

To demonstrate that our theoretical constructions are practically feasible, implementation was written on a simulated IoT network using the Coooja simulator in Contiki [DGV04] (**Paper II**).

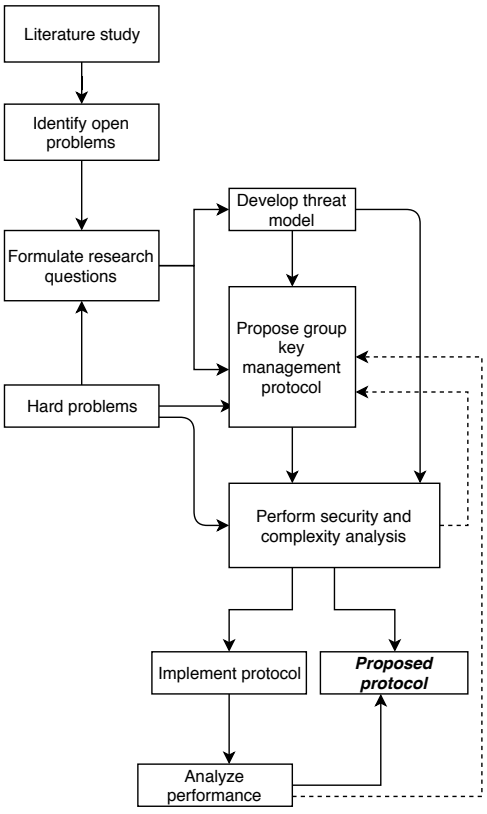

Figure 1.1: Research work flow.

The overall research work flow that was adopted in designing the protocols is depicted in Fig. 1.1.

1.5 Contributions

This thesis is based on the four papers listed previously, also included in full at the end of this work. The papers address different aspects of the issues stated in section 1.3.

Paper I presents a lightweight group key management scheme that uses elliptic curve based cryptography and the notion of cryptographic one-way accumulator. The main contribution presented in this paper is a demonstration of a secure construction of a lightweight group key management scheme which uses point multiplication on elliptic curves as the main underlying cryptographic computation. The novelty of this contribution is showing that a secure and lightweight key management scheme can be built that enables IoT devices in a group to share a secure key, generate a group digital signature, generate group session keys and enable addition or deletion of a node from a group. **Paper** II is an extension of **Paper** I. **Paper** II presents an implementation of the scheme proposed in **Paper** I in Contiki [DGV04]. The paper shows the feasibility of the scheme with regard to energy consumption, memory usage, execution time and number of messages generated. **Paper** III presents a lightweight and secure group key management scheme from the OTP. The novelty of the scheme is showing that despite the intrinsic weakness of the OTP, it could be used as an underlying primitive to construct group key management, resulting in a scheme that is lightweight and unconditionally secure. **Paper** IV presents an overview of proximity-based authentication mechanisms in use today in IoT environments. It presents an analysis of strengths and drawbacks of various authentication schemes, and a suggestion for more secure and convenient authentication mechanisms.

A mapping of all the contributions and how they fit into the overall goal and research questions is depicted in Fig. 1.2.

1.5.1 The Authors' Roles

As the first author of papers I and III, I was responsible for the ideas, methods, security analyses and proofs, as well as presentations. For **Paper** II, I came up with the implementation ideas, and Nico Ferrari – the first author of the paper – implemented the scheme and wrote the paper. All authors of **Paper** IV equally contributed towards the ideas presented and the writing process.

The co-authors have helped me in terms of providing guidance, technical suggestions, corrections, and reviewing the manuscripts.

1.6 Thesis Outline

The remainder of the thesis is organized into four chapters, with the content of each chapter as follows:

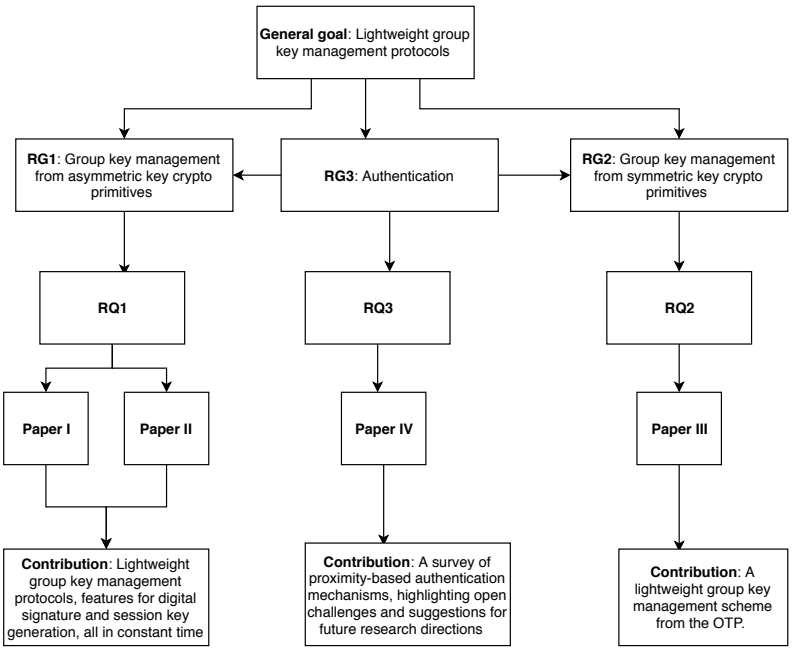

Figure 1.2: A mapping of the contributions to the overall goal and sub-goals and the novelty of each contribution

– Chapter 2 introduces the relevant mathematical foundation required to understand the cryptographic primitives that serve as building blocks of the protocols proposed.

– Chapter 3 presents a discussion of the broad topic of group key management in the domain of the IoT, with focus on state-of-the-art and open challenges.

– Chapter 4 presents new lightweight group key management protocols proposed to address some of the challenges discussed in Chapter 4.

– Chapter 5 contains conclusions, providing overview, outcome, impact of the research presented and directions for future work.

Chapter 2

Background

This chapter briefly introduces the cryptographic primitives used as building blocks in the group key management schemes that will be described in Chapter 4. The chapter also introduces the hardness assumptions of the mathematical problems that underlie the security of the primitives.

2.1 Mathematical Background

Definition 2.1. An algebraic structure is a set S together with one or more binary operations, such that the following two conditions are true [Fra03]:

1. Each binary operation assigns exactly one element to each possible ordered pair elements of S.

2. For each ordered pair of elements in S, the element assigned to it is again in S.

Definition 2.2. Let $S, *$ and $S^{'}, *^{'}$ be two binary algebraic structures. An isomorphism of S with $S^{'}$ is a one-to-one mapping(function) ϕ mapping S onto $S^{'}$ such that

$$\phi(x * y) = \phi(x) *^{'} \phi(y), \text{ for all } x, y \in S.^{1}$$

The implication (and advantage) of showing that two binary algebraic structures S and $S^{'}$ are isomorphic is that one can reason about S by only reasoning about $S^{'}$, since they are structurally the same. For example, by showing that a less studied algebraic structure is isomorphic to $\{\mathbb{Z}, +\}$, one gets the benefit of relying on a well studied algebraic structure to talk about another algebraic structure that is less known.

[1] This is called homomorphism property.

There are a lot of algebraic structures, but here we focus only on groups since they are important for our purposes.

Definition 2.3. A **group** $\langle \mathbb{G}, * \rangle$ is a non-empty set $\mathbb{G}$, with a binary operation $(a, b) \mapsto a * b : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ satisfying the following conditions [Fra03]:

1. For all $a, b, c \in \mathbb{G}$, we have

$$(a * b) * c = a * (b * c), \textbf{ associativity of } *.$$

2. There is an element e in $\mathbb{G}$ such that for all $x \in \mathbb{G}$,

$$e * x = x * e = e, \textbf{ identity element } e \text{ for } *.$$

3. Corresponding to each $a \in \mathbb{G}$, there is an element $a^{'}$ such that

$$a * a^{'} = a^{'} * a = e, \textbf{ inverse } a^{'} \text{ of } a.$$

A group is called **abelian** or commutative if
$a * b = b * a$ for all $a, b \in G$.

Definition 2.4. The number of elements in group $\mathbb{G}$, denoted $|\mathbb{G}|$, is called the order of $\mathbb{G}$. A group $\mathbb{G}$ is finite if $|\mathbb{G}|$ is a positive integer.

A group $\mathbb{G}$ is said to be cyclic if there is an element $g \in \mathbb{G}$ such that for for any $a \in \mathbb{G}$, there is an $n \in \mathbb{Z}$ such that $a = g^n$, where $g^n = \underbrace{g * g * \cdots * g}_{n \text{ times}}$. Such an element g is called a *generator* of $\mathbb{G}$.

Example 2.1. For prime p, the set of integers $\{0, 2, \cdots, p-1\}$ forms a cyclic group under addition modulo p . The set of nonzero integers modulo p also form a multiplicative group denoted by ${\mathbb{Z}_p}^*$ [Fra03].

In this thesis, we let $\mathbb{G} \subset {\mathbb{Z}_p}^*$ be a cyclic group of prime order q, where g is its generator. The security parameters p and q are such that $q \mid (p-1)$ and the order of g is q, that is, g is a generator of $\mathbb{G}$.

2.2 Hard Computational Problems and Assumptions

This section briefly introduces some computational problems that are believed to be intractable. The presumed computational hardness of the problems is related to the complexity of the algorithms that are known solve them. Therefore, we first briefly discuss algorithm complexity classes.

Definition 2.5. (**Negligible Functions**): a function $f : \mathbb{N} \rightarrow \mathbb{R}$ is called negligible if it approaches zero faster than the reciprocal of any polynomial. In other words, for every $c \in \mathbb{N}$ there is an integer n_c such that $g(n) \leq n^{-c}$ for all $n \geq n_c$ [Bel02].

The concept of negligible functions is important when talking about the success probability of an attacker, as a function of some security parameters, such as key size. We say that the success probability is too small to matter if it is a negligible function of some security parameter.

Definition 2.6. (**Polynomial Time Algorithm**): A polynomial time algorithm is an algorithm whose worst-case running time function is of the form $\mathcal{O}(n^c)$, where n is the input size, in some reasonable encoding, and c is a constant. Polynomial time algorithms are also informally called efficient algorithms [Bel02].

A computational problem for which there is a polynomial time algorithm to solve it is considered an easy problem. A problem for which there is no such algorithm to solve it is considered hard or intractable. There are a lot of computational problems that are believed to be hard [Opp11], but here we focus on the the Discrete Logarithm Problem (DLP) and its variant, the Elliptic Curve Discrete Logarithm Problem (ECDLP).

Definition 2.7. (**Discrete Logarithm Problem**): Given a finite group $\mathbb{G}$, with generator g, the discrete logarithm problem asks to find an a between 0 and $|\mathbb{G}| - 1$ from g^a, i.e, $DLog_g(g^a)$ [BL96]. This problem is considered to be hard and forms the basis of many public key cryptographic systems.

Generally, all the algorithms that are hitherto known to solve the DLP take exponential time in the size of the input [Sho94]. However, depending on the underlying algebraic structure on which the group operation is defined, the level of difficulty of solving the the DLP varies [KMV00]. The DLP defined on the multiplicative group of integers modulo a prime p is easier than when it is defined on the additive group of points on an elliptic curve due to the disparity inherent in the algebraic structures and operations defined on them [HMV06]. The implication is that a given security level provided by DLP on a multiplicative group of integers modulo prime p can be achieved by formulating DLP on an elliptic curve with smaller parameters [BMS+06]. This makes cryptography based on elliptic curves a good fit for IoT environments.

2.3 Elliptic Curve Based Cryptography

For our purposes, an elliptic curve E defined over a finite field $\mathbb{F}$ is an equation of the of the form

$$y^2 = x^3 + ax + b \tag{2.1}$$

where a and b are elements of a finite field $\mathbb{F}$ with p^n elements for some large prime p [Kob87]. There are also elliptic curves of other forms, but this is the one we

use in our constructions and implementations following standard practice [KAS08]. The set of points (x, y), $x, y \in \mathbb{F}$, that satisfy equation (2.1) plus an extra point, referred to as a point at *infinity* and denoted by $\mathcal{O}$, and an "addition" operation defined as follows form a group [Kob87]. Geometrically speaking, to add two points $\mathbb{Q}_1$ and $\mathbb{Q}_2$ together, one draws a straight line that passes through $\mathbb{Q}_1$ and $\mathbb{Q}_2$ and looks for the third intersection with the curve, $\mathbb{R}_1$. Then reflecting the point $\mathbb{R}_1$ along the x-axis yields $\mathbb{Q}_1 + \mathbb{Q}_1$. To add a point Q to itself, draw a line tangent to $\mathbb{Q}$ and look for the second point $\mathbb{Q}^{'}$ at which the line crosses the curve E. The reflection of $\mathbb{Q}^{'}$ across the x-axis is the sum $\mathbb{Q} + \mathbb{Q}$. The elliptic curve group operation is depicted graphically in Fig. 2.1 [Lau04]. The symbol $\mathcal{O}$ serves as the additive identity element. A related group operation is scalar point multiplication, whereby a given point is added to itself a given number of times. The computation is effected via the common *square and multiply* method for efficiency reasons [GLV01].

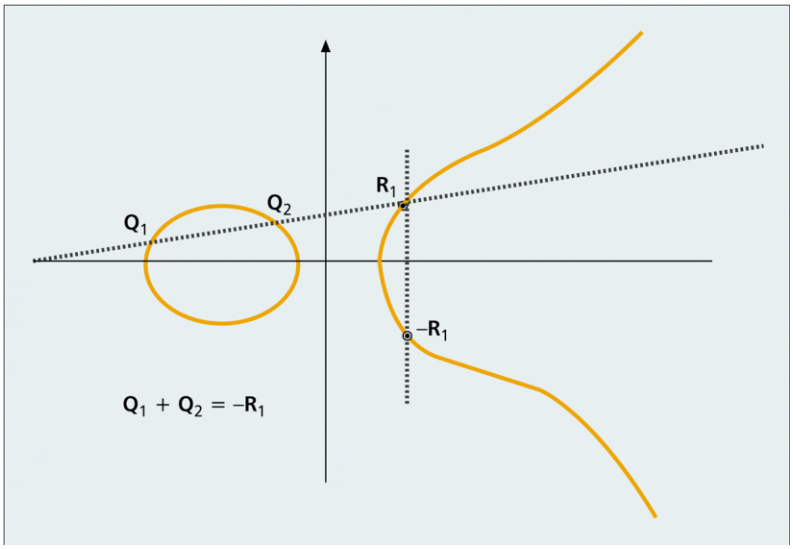

Figure 2.1: Group law on an elliptic curve

We can now state the Elliptic Curve Discrete Logarithm Problem (ECDLP).

Definition 2.8. (**ECDLP**:) Given an elliptic curve $\mathbb{E}$ over a finite field $\mathbb{F}$, and two points P and Q such that $Q = kP$ for some integer k, the ECDLP is to compute k [Mil85]. It is conjectured that this is a computationally hard problem.

Since the ECDLP appears to be harder than the DLP, the strength-per-key-bit is substantially greater in elliptic curve systems than in conventional discrete logarithm systems [KMV00]. Consequently, smaller parameters, but with equivalent security levels of security, can be used with an ECDLP-based crypto-system than with conventional discrete logarithm based system [JMV01]. Using smaller parameters means faster computation and less storage requirement, making elliptic curve based crypto-systems convenient for environments where processing power, storage space, bandwidth or power consumption are constrained.

2.4 Cryptographic One-way Accumulator

A cryptographic one-way accumulator [BDM93] is a way to combine a set of values into one accumulator value, in such a way that each participant whose value was used in the computation is able to produce a witness that it has participated in generating the accumulator value. Formally, A one-way cryptographic accumulator is defined as a one-way hash function $f : \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{X}$ such that $\forall x \in \mathbb{X}$ and $\forall y_1, y_2 \in \mathbb{Y}$

$$f(f(x, y_1), y_2) = f(f(x, y_2), y_1).$$

The function f can then be used to compute an accumulator value z for a set of values $\{y_1, y_2, \ldots, y_n\} \in \mathbb{Y}$ given a base value $x \in \mathbb{X}$ by applying f repeatedly to each y_i. It can also be used to generate a witness z_j for a value y_j in the set, by accumulating all y_i such that $i \neq j$. Since the order in which accumulation was done is not relevant, one can recover $z = h(z_j, y_j)$, and this holds for all $y_j \in \mathbb{Y}$.

In our proposed scheme, we will use this property to generate a witness and also to complete the computation of a shared key as will be shown in Chapter 4. We use point multiplication on an elliptic curve as a one-way accumulator in our proposed scheme. The basic function f is defined as

$$f(s, \mathbb{P}) = s \times \mathbb{P} = \mathbb{Q}$$

It takes an integer value s and a point $\mathbb{P}$ on the curve and outputs another point $\mathbb{Q}$ on the same curve. It is one-way since point multiplication can be done easily, using repeated point addition [Kob87], while computing the reverse direction, i.e, computing s from $s\mathbb{P}$ is hard due to the ECDLP assumption. Moreover, the function is quasi-commutative since we have

$$f(f(s_1, \mathbb{P}), s_2) = f(f(s_2, \mathbb{P}), s_1) = (s_1 s_2)\mathbb{P}.$$

We exploit this property twice in our scheme. First, during group key establishment and, second, when each node produces a proof of group membership. During key establishment, a device multiplies a point by a sequence of integers in any arbitrary sequence.

We end our brief discussion of mathematical background by introducing the OTP.

2.5 One Time Pad

The OTP is a pair of encryption algorithm E and decryption algorithm D over message space $\mathcal{M}$, cipher space $\mathcal{C}$ and key space $\mathcal{K}$ with the following properties:

1. The message space $\mathcal{M}$, cipher space $\mathcal{C}$ and key space $\mathcal{K}$ are all of length n-bits. In other words, $\mathcal{M}$, $\mathcal{C}$, $\mathcal{K}$ = $\{0, 1\}^n$.
2. E takes a random key $k \in \mathcal{K}$ and a message $m \in \mathcal{M}$ and outputs $c = E(k, m) = k \oplus m$ in $\mathcal{C}$.

3. The decryption algorithm D takes a cipher message $c \in \mathcal{C}$ and a key $k \in \mathcal{K}$ and outputs $D(k, c) = k \oplus c$. The consistency requirement holds because $D(k, E(k, m)) = k \oplus E(k, m) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0 \oplus m = m$.

The OTP has two desirable properties. First, it is fast since the only computation is bit-wise XOR. Second, it is unconditionally secure [Sti05]. This is true because if k is uniformly sampled from the key space $\mathcal{K}$, given a cipher $c \in \mathcal{C}$ and given two distinct messages $m_1, m_2 \in \mathcal{M}$, the probability that c is the encryption of m_1 is the same as the probability that c is the encryption of m_2 [Sha49]. This means that given a cipher c, an attacker can learn absolutely nothing about the corresponding plain text. This fact is alternatively called information-theoretic security.

The OTP also has two limitations. First, the unconditional security guarantee holds if and only if the key length is the same as the message length [Sha49]. Second, as the name implies, it can only be used once, as using it to encrypt more than one message is insecure [DL04]. Due to these limitations, the OTP is not commonly used in practise. Despite these limitations, we will show that the OTP can be put to good use in constructing a secure and lightweight group key management scheme.

Chapter 3

Cryptographic Group Key Management in the Internet of Things

This chapter discusses the security requirements in IoT group communication, the various aspects unique to the IoT that need to be taken into account when designing group key management protocols, and the state-of-the-art of lightweight group key management protocols.

3.1 Security and Privacy Requirements in Group Communication

Whether in two party or group communication, IoT applications must provide certain security guarantees to ensure the privacy and safety of users. Security is a multi-faceted concept that includes a lot of aspects, but generally, a typical IoT application would be required to provide some or all of the following security services [KMVOV96]:

- *Confidentiality* is a service used to keep the content of information from all but those who are authorized. Various mechanisms such as physical protection of information can be employed to provide confidentiality, but we are mainly interested in mechanisms that employ cryptography to achieve confidentiality.
- *Data Integrity* is a service that deals with the unauthorized alteration of data. In order to provide such a service, one must have a mechanism for detecting data tampering, such as unauthorized alteration, deletion or insertion.
- *Authentication* is a service that deals with the identification of some or all of

the entities involved in a communication and the data that is communicated. As a result, authentication has two dimensions. Entity authentication deals with identifying the parties involved in a communication. Data origin authentication deals with ensuring that data is from the right source and that it has not been tampered with. Obviously, data origin authentication implies data integrity.

- *Non-repudiation* is a service that prevents an entity from denying having participating in a communication, either as a sender or receiver. This service prevents the possibility of dispute arising as a result of one or more entities denying previous actions.

Typically, cryptography is the tool used to achieve the aforementioned security services. Encryption is used to achieve *confidentiality*, digital signatures are used to achieve *authentication* and *non-repudiation*. In order to encrypt and digitally sign messages, encryption keys are required. A question that immediately emerges is how cryptographic keys are established and managed among the entities that participate in a secure communication.

Before delving into the discussion of group key management protocol, it is important to make a distinction between key establishment and key management protocols.

Definition 3.1. Key management is the set of processes which support key establishment and maintenance of ongoing keying relationships between parties, including replacing older keys with new keys, revoking keys, generating session keys and other similar tasks [Cho06].

Definition 3.2. Key establishment is a subset of key management dealing with making a shared secret become available to two or more parties, for subsequent cryptographic purposes [Cho06].

Key establishment protocols deal only with making a shared secret available to relevant parties only once, whereas key management schemes also include mechanisms for key management, key revocation, node addition and deletion and key revocation. Therefore, key establishment is a subset of key management. As a result, key management is the focus of this thesis.

3.2 Challenges in Designing IoT Group Key Management Protocols

Construction and implementation of secure and lightweight group key management schemes is a non-trivial problem with a lot of challenges. Some of the issues involved are the following:

- Identifying the group security services that need to be provided. These services may vary from application to application, but usually one would like to

provide confidentiality of a group message so that only legitimate group members can access the message. There may also be scenarios where validating the source and integrity of a group message is required. This is achieved by group digital signatures and adding a message authentication code (MAC) to group messages.

- Since a cryptographic key is a prerequisite for providing one or more of the above security services, a mechanism for establishing and managing a key between the set of devices is required. There are generally three ways to do this [Gol09]. In small and static networks key management can be done manually, by storing keys or keying information on each device and updating keys when necessary. This approach is not feasible in large and dynamic networks where keys need to be changed and updated frequently. The second approach is to design key management protocols from scratch. This approach is generally not recommended since a crypto system that has not been publicly tested and vetted is vulnerable. The third and most common approach is to build key management construction from well studied existing cryptographic primitives. Therefore, an important issue when embarking on a research related to key management is to decide a convenient approach regarding how and what kinds of primitives are to be used.

- It is not enough to care only about primitives that satisfy a stated security objective in the domain of the IoT. Given that IoT devices are computationally limited and resource constrained, it is important that the primitives employed in the construction of any key management protocol are lightweight. The challenge in designing lightweight cryptographic protocols is to ensure that security levels are not sacrificed in an effort to force protocols to be lightweight.

- Designing secure protocols underpinned by sound mathematical proofs is no good if the protocol is implemented incorrectly or insecurely. There are a lot of attacks that exploit weaknesses in implementation [ZG13]. Therefore, to achieve a given security objective, not only must a protocol have a secure design, but it must also be implemented securely.

- One has to show or prove that a security construction is secure. There are generally two approaches: heuristic analysis and provable security approach [Ng05]. In the former, a security protocol is assumed to be secure if it withstands known attacks. This is problematic because there could be unforeseen attacks. Furthermore, it is not feasible to list all possible attacks and prove that a protocol withstands them. The latter approach is to prove a security claim by demonstrating that if the protocol can be attacked, then a problem believed to be hard can be solved, using the attack scheme as a subroutine. This is what is technically called a reduction argument [KM07]. While this approach is theoretically appealing, it is hard in practice, as proofs could get extremely messy [Ng05]. In either case, it is important that there is a clearly specified security model, security definitions, clearly stated assumptions and security goals.

3.3 Lightweight Cryptography

Lightweight cryptography refers to a set of design principles and techniques for designing and implementing cryptographic primitives, algorithms and protocols tailored for resource-constrained environments such as RFID tags, sensors, contactless smart cards, implantable devices, and others [KM08]. What all these have in common is that they mainly consist of IoT devices which are constrained in terms of computational capability, memory space and battery power. Conventional cryptographic primitives and protocols were not designed with the objective of being implemented in resource constrained devices. As a consequence, the cryptographic primitives and protocols commonly used in the conventional Internet are not feasible for IoT. Lightweight cryptography is an attempt to bridge this gap.

The main challenge in designing lightweight cryptography is finding an acceptable trade-off between guaranteeing acceptable security levels on the one hand and performance and cost on the other hand [Pos09]. Security level is generally a function of some security parameter, such as key length or group size in the case of public key cryptography [CBCM12]. A naive approach to realize lightweight cryptography is to take an existing cryptographic primitive and run it with smaller security parameters. Although this meets the low cost and high performance requirement, small security parameter means that the security level is reduced, making attacks easier. As a result, this approach is not generally recommended. A second approach is to build cryptographic algorithms specifically designed for IoT, taking into account the limited nature of devices. This involves carefully designing cryptographic primitives that are underpinned by computations that are inherently lightweight.

Generally, lightweight primitives can be designed from symmetric key or public key cryptographic constructions. Constructions that mainly rely on the binary XOR operation are considered lightweight since the XOR operation is not computationally intensive both in hardware and software implementations [EKP+07]. As a result, symmetric key cryptographic lightweight constructions based on bit-wise XOR, such as the OTP, are suitable for IoT environments. However, since such constructions have limitations with respect to key length and distribution, they do not answer all the security requirements in various IoT application scenarios, such as digital signatures, key management and other related security aspects. To address problems related to key management and digital signatures in IoT, public key cryptosystems based on lightweight mathematical underpinnings are employed. One such construction is public key crypto systems based on elliptic curves. The inherent supposed hardness of the Discrete Logarithm Problem (DLP) defined on elliptic curves has made it possible for the creation of many lightweight public key crypto systems that achieve various security objectives in IoT environments. Implicit digital certificates [SCP+15] are one such lightweight crypto systems which make authentication possible without relying on the otherwise computationally complex PKI Internet solution. Elliptic Curve Digital Signature Algorithm (ECDSA) [JMV01] is a lightweight digital signature scheme widely deployed in resource constrained environments. The Identity Based Encryption (IBE) schemes proposed in [BF01] are also lightweight encryption schemes that are suitable for IoT as they are underpinned by

algebra of elliptic curves.

3.3.1 Lightweight Group Key Management Protocols

A cryptographic key management protocol for IoT addresses issues ranging from enabling three or more IoT devices to agree on a key or keying material (key establishment) to management aspects such as secure addition and removal of a node to/from the group and generating session keys securely. In some cases, a key management protocol needs to have a mechanism to authenticate devices to each other, in which case the protocol is called authenticated key management protocol [XMH06]. An authenticated key management protocol is desired in situations where IoT devices do not generally trust each other, or in situations where an attacker can feasibly masquerade as a legitimate member of a given group. In other words, a cryptographic group key management needs to have a mechanism for authentication, group key establishment and secure group key generation.

In the following section, we briefly discuss various lightweight cryptographic group key management constructions from both symmetric and asymmetric key cryptographic primitives.

Lightweight Group Key Management Protocols from Symmetric Key Constructions

Symmetric key primitives can be used to address different aspects of group key management. We briefly discuss how symmetric key primitives are used to to provide authentication, key establishment and session key generation.

- *Authentication* : By having a group of IoT devices share a common group key, device authentication can easily be achieved by having a device send an encrypted message to any other device or group of devices in the network. Since, by assumption, only legitimate devices have the secret key, if an encrypted message decrypts to an intelligible message, it can reasonably be assumed that it was sent for a legitimate member, effectively authenticating the device. This kind of authentication mechanism is employed in many key establishment protocols. Such authentication mechanisms have an obvious weakness in that they do not address the important problem of how to make the secret key available in all the devices in the first place. In practice, a key or a keying material is manually stored on the relevant devices and manually updated as required. This is practical in small and stable networks, such as home automation application, but it becomes increasingly impractical in large and dynamic networks.
- *Key establishment*: [RSSS16] proposes a mechanism using symmetric key encryption to establish a group key among IoT devices in a network. The basic set-up is that all devices in the group have pre-shared keys, and a carefully designed symmetric key encryption algorithm is run in such a way that at the end of the protocol all the devices in the group have the same secret key. There

Algorithm family	Crypto-systems	Security level (bit)			
		80	128	192	256
Integer factorization	RSA	1024	3072	7680	15360
Discrete logarithm	DH, DSA, Elgamal	1024	3072	7680	15360
Elliptic curves	ECDH, ECDSA	160	256	384	512
Symmetric-key	AES, 3DES	80	128	192	256

Table 3.1: Computationally equivalent key sizes expressed in bits.

are several different variants of this basic pre-shared key set-up, relying on pre-shared keys and various lightweight cryptographic symmetric key constructions [PP18, WGL00, MP04, Bri99].

- *Session key generation*: To the best of the author's knowledge, there are no symmetric key based session key management mechanisms, with support for secure removal and addition of a node to the group, without running the key establishment protocol from scratch.

Lightweight Group Key Management Protocols from Asymmetric Key Constructions

Lightweight group key management schemes can also be constructed from asymmetric cryptographic primitives. Public key primitives inherently involve more complex computations when compared to symmetric key primitives. Due to the inherent weakness of symmetric key cryptography regarding key establishment, key management solutions based on asymmetric key cryptography are deployed, at the cost of incurring the complex computational requirements.

Traditional secret key establishment mechanisms such as the Diffie-Hellman key exchange protocol and its variants rely on public key primitives whose security relies on the supposed hardness of the Discrete Logarithm Problem (DLP) or Integer Factorization (IF) problem. The supposed level or hardness of these problems depends on the nature of the underlying algebraic structures on which they are defined. Due to the reasons discussed in Chapter 2, DLP defined on the group structure induced by the set of points on an elliptic curve is inherently harder than the same problem defined on a conventional finite field [MOV93]. Consequently, crypto-systems based on elliptic curve cryptography with smaller parameters provide comparable levels of security to crypto-systems based on RSA or other conventional public key crypto-systems. This means elliptic curve based cryptographic primitives result in faster and fewer computations, require less memory and consume less power as a result.

Table 3.1 shows a comparison of the different cryptographic primitives in terms of key lengths required for a specific security level [PP09]. For example, an elliptic curve crypto-system defined over a finite field over prime p of size 160-bits is equivalent to AES-80 or RSA with modulus n of size 1024-bits. The key length of elliptic

curve based crypto-systems is significantly smaller, making them good candidates for lightweight public key crypto-systems for many IoT applications.

A discussion of how elliptic curve based crypto-systems can be used to address the various aspects of key management, including authentication and session key generation is provided below:

- *Authentication:* authentication, both data and entity, is mainly achieved using the Public Key Infrastructure (PKI) in the Internet. It is, however, difficult to use the same solution in many IoT environments due to the complexities inherent to the PKI. This has necessitated designing lightweight authentication mechanisms specifically tailored for IoT environments. One such mechanism is implicit certificates which, unlike conventional digital certificates, do not require complex certificate verification process [VMQ98]. Moreover, implicit certificates rely on elliptic curve based cryptography, which makes them lightweight. Raw public key based authentication mechanisms are also employed in some IoT applications [SSG13]. Some of the open issues regarding authentication using public key cryptographic primitives are how to manage certificates (renewal, revocation, expiry) and entity identification in the absence of a central trusted entity [LXC12].

- *Key establishment:* in a two-party setting, the most common public key based key establishment mechanism is the Diffie-Hellman Key exchange protocol, whose security relies on the presumed hardness of the DLP [DH76]. The original two-party protocol can be extended to a group key establishment protocol, as demonstrated in [STW96]. A variant of the original Diffie-Hellman key exchange protocol whose security relies of ECDLP is used in IoT settings [MWS04]. The Diffie-Hellman key exchange protocol can also be extended by adding an authentication mechanism through certificates to prevent MITM attacks, in which case it is *authenticated key exchange* [BMP00]. Other group key establishment protocols have been proposed such as MIKEY [ACL+04], designed for multimedia distribution; TESLA [PSC+05], designed for source authentication in broadcast communication; and other lightweight key establishment protocols based on public key cryptography [PBS+15, MWS04, MNG17].

3.4 The State of the Art

This section discusses the approaches and techniques that have been adopted to design group key management schemes in IoT.

Since key establishment is a precursor to key management, it is important to explore key establishment protocols. There are generally two approaches to key establishment [Cho06]. The first is *key agreement*, whereby the parties agree on a key in such a way that each participating party influences the outcome. The second approach is *key transport*, whereby one party creates or otherwise obtains a secret value and securely transfers it to the other parties.

Key transport protocols are convenient in IoT networks consisting of a set of end nodes connected to a trusted entity, such as a gateway, which serves as a key distribution center (KDC) [SO12]. In such settings, the KDC unilaterally constructs the key or keying material and securely sends it to each member of the secure group. The group key is then a function of the keying material. When a new key is needed, the process is repeated from scratch. This is practical in stable networks, i.e., low node addition and removal rates, with a trusted entity with sufficient computational capabilities and where communication happens only sporadically. This is obviously not the typical IoT application scenario.

In a typical IoT application, a group of IoT devices connected to each other and to a designated entity, commonly called a gateway, continuously collaborate with each other to provide a specific functionality. In such a scenario, the above model with a single KDC is not practical for the following reasons:

- There may not exist a dedicated KDC with enough computational power and battery-life to sustain a network for a long period of time. Even when there is one, it may sometimes fail or malfunction because of technical failure or tampering [KKA13].
- Continuous generation of session keys is required in most IoT applications that require devices to send secure messages more than once. Since using the same key to encrypt more than one message is generally insecure [WGL00], having a mechanism for generating session keys from already established parameters is important.

Key establishment mechanisms based on key transport usually fall short of providing comprehensive solutions for establishing group keys in IoT. To remedy this, many key establishment protocols based on key agreement mechanisms have been proposed [RSSS16, PBS+15, MNG17, KH18, RALS11]. In key agreement based key establishment protocols, all the relevant parties participate in the protocol and influence the value that is finally established as the shared secret or key. This bodes well for IoT because the work is distributed among the set of devices constituting a group and there is no single point of failure. Furthermore, since there is no single entity where all keying information is stored, a potential attacker cannot target a device, and hence the network is more resistant to attacks.

While key agreement schemes without a central entity are good in terms of being distributed, the problem of trust naturally emerges. If there is no central entity that acts as a trust anchor vouching for the claimed identity of other nodes, there is no straightforward way for any two entities to verify the identity each other. [KPT04] proposed a fully distributed key management protocol by extending the Diffie-Hellman key exchange protocol for Internet group applications, such as multicasting in the Internet. However, such protocols are not practical for most IoT group applications because of the heavyweight computations they involve and reliance on identity vouching infrastructure such as PKI [AL03], which is infeasible for IoT networks. In general, key agreement protocols based on complex computations and reliant on security solutions that do not take into account the limited nature of IoT

devices are not feasible for IoT environments. As a result, key management mechanisms based on lightweight cryptography are generally preferred in environments consisting of resource constrained devices.

3.4.1 Open Problems

Key management in IoT presents peculiar challenges owing in part to the fact that constituent devices are resource constrained, deployed in environments where physical access to devices is possible, and there is usually no designated trust anchor [KH18]. Moreover, IoT group applications could sometimes be highly dynamic in that they involve high rate of devices joining or leaving the group, necessitating the need to guarantee forward and backward secrecy and secure group session key management mechanisms [EPG18].

The standard way to design a cryptographic protocol suitable for resource constrained IoT devices is to rely on lightweight cryptographic primitives. As discussed in Section 3.3, a crypto-system based on elliptic curves on a finite field with prime p of size 163-bits is considered lightweight, although lightweight constructions may still be computationally intensive for tiny IoT devices, such as sensors powered by 8-bit processors. Therefore, one aspect of designing a lightweight group key management protocol is maintaining an acceptable balance between performance and security. Performance can be enhanced by using cryptographic primitives with smaller security parameters but that comes at the cost of reduced security guarantees, which is not desired in environments where an attacker has greater computational power than the devices in the network. Designing a group key management protocol that is both lightweight and secure is one of the open challenges that will be presented in Chapter 4.

Device authentication is another open problem in designing key management protocols. Before devices establish a cryptographic key among themselves, it is important that each device has a mechanism to ensure that it is communicating with authentic devices. Authentication in IoT is particularly difficult because conventional authentication mechanisms such as those based on the PKI are computationally intensive. Password or PIN based authentication mechanisms are also not suitable for IoT because small IoT devices do not generally have an input/output interface. Authentication mechanisms based on lightweight cryptographic constructions [shi, YCW+16], and other mechanisms which take environmental context, such as proximity into account [MMV+11, LH14] have been proposed. Some of the open problems regarding authentication, such as device authentication in a group setting, and establishing trust without trust anchor are discussed in [QHG+18].

Chapter 4

Proposed Lightweight Group Key Management Protocols

In this chapter, we delve into our proposed lightweight group key management protocols that are the main contributions of this thesis. In previous sections, we discussed the unique challenges that key management in IoT presents and the challenges that call for more research. Our proposed protocols address some of the open issues with regard to group key establishment, group signature, group session key generation and addition and removal of IoT devices to a group.

4.1 Network Model and Assumptions

We consider a network of k nodes all connected to a common gateway G, which may itself be connected to the Internet or another network, as depicted in Fig. 4.1. Formally, the network can be modeled as a graph $G = \langle G, N, E \rangle$, where G is a gateway that acts as a group initiator and manager and N is a set of nodes $N_1, N_2, \cdots, N_k$, and $\mathbb{E}$ is the set of edges from G to N_i representing bi-directional communication links. The model is a reasonable abstraction of many IoT applications. For example, in a typical IIoT setting, the gateway could represent a controller and the nodes represent a set of sensors and/or actuators that receive command from the controller. Similarly, in an e-health application, the gateway could represent a smart phone App whereas the nodes represent a set of implants in a patient's body. In either case, one would like the entity represented by the gateway to send a group message to a group of some or all entities represented by the nodes. Generally, the gateway itself would be connected to an external network but that is not relevant for our purposes.

In all the protocols we propose, we make the following assumptions regarding communication protocol, security associations between and among devices and other network deployment aspects.

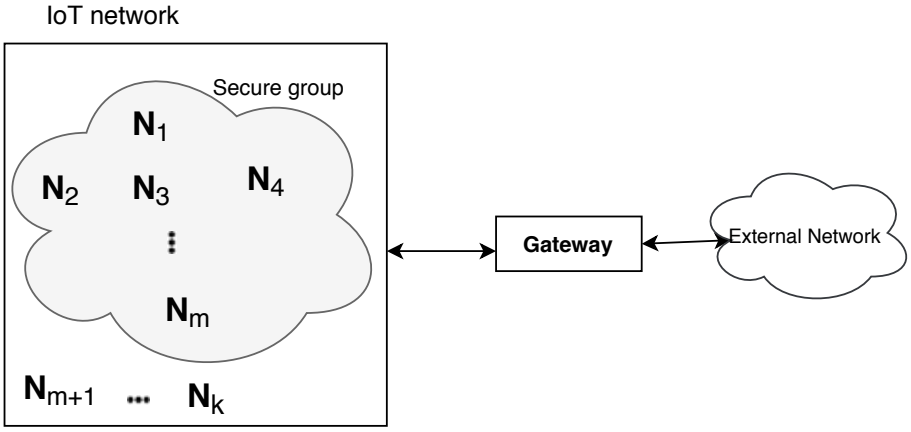

Figure 4.1: Network Model. The network consists of a gateway and a set of nodes, supported by a communication infrastructure. All or a part of the nodes may be members of a group as shown in the figure (m of the k nodes are in the group).

- The proposed group key management protocols work at the application layer and as such are agnostic to the specifics of underlying communication protocol(s) used. It is assumed that when we say *entity A sends a message to entity B*, the underlying network has a mechanism for delivering the message, regardless of what the actual communication protocol is. Similarly, when we say *entity A broadcasts a message to the network*, it is assumed that the underlying network can reliably deliver the message to all the recipients in the network.
- Security associations between devices, whether it be setting up private/public key pairs or pre-shared secrets, is done through other off-band mechanisms that we do not discuss in this thesis. It is assumed that each node on the network has a way of getting and verifying the public key of another device in the network.
- IoT group applications generally run on a group of devices uniquely identified by a specific address. Except when talking about implementation, we do not generally talk about what specific addressing mechanism is employed to identify a group.
- The main focus in this thesis work is designing lightweight group key management mechanisms. We also provide implementation of one of the proposed protocols to demonstrate feasibility of the protocol. However, we do not generally focus on implementation related aspects such as how to securely implement the protocol to prevent side-channel attacks [BSP+11], or how to optimize implementation to minimize power consumption or other computational resources.

The protocols we propose here address the following aspects of group key man-

agement in IoT.

- *Group key kstablishment* deals with designing a secure and lightweight scheme that enables the gateway and a set of the nodes (not necessarily all) to establish a keying material among themselves. The keying material securely shared among the legitimate devices is then used to generate group session keys that are used to encrypt group messages. The objective is to minimize the number of messages generated by all nodes and the number of computations at each node.

- *A group session key generation mechanism* is required in situations where there are more than one group messages being broadcast, one after the other. Using a group key to encrypt more than one group message is generally insecure due to many well known attacks [Ble98], and therefore a key management protocol should have a mechanism for generating session keys independent from each other. Independence of session keys from each other means that a given session key carries no information about previous or future session keys. Generating independent session keys is not a hard problem in the conventional Internet because a new session key can be generated by running the key establishment anew since there is not strict computational or power constraint. In the IoT context, however, it is costly to run a group key establishment protocol anew every time a session key is required. We use the Psedo-Random Function (PRF) primitive to generate session keys which are computationally independent from each other. The key securely established using the session key establishment protocol is used as a seed input to the PRF. Details of the construction can be found in papers I and III.

- *A group digital signature* is required in situations where there is no designated sender in a group. In such cases where any member of a group can broadcast a message, it may be required that recipients have a mechanism that lets them know who the sender is. Since a group key does not uniquely identify a single member node, the key can not be used to produce a digital signature to be attached to a group message. In the proposed protocols, we use a combination of the notion of cryptographic one-way accumulator and elliptic curve point multiplication to produce a construct that effectively serves as a digital signature. The digital signature is then broadcast along with the group message. The construction is such that it is computationally easy for the legitimate node to produce the signature and for the recipients to verify the signature. Forging the signature amounts to solving the ECDLP.

- *Secure management of group membership* is required in a dynamic group where adding a new node or removing a member node from the group may be required. The exact definition of secure management of group membership may vary from context to context, but any reasonable definition has to generally include two important properties. First, the group membership management has to ensure that a newly added node does not have access to messages or session keys used before it joined the group. This implies that session keys should not

be stored in places accessible to a new node and that it should be computationally hard to compute past session keys from a current session key. Second, the group membership management has to make sure that a node that has been evicted from the group does not learn future session keys or group messages. In our proposed scheme, we use ephemeral session keys. This makes it impossible for a new node or adversary to access past session keys. To make computing session keys computationally hard for an attacker, we designed our protocols in such a way that succeeding in computing a session key without being a member amounts to solving a problem supposed to be computationally hard.

4.2 Papers

In previous sections and chapters, the necessity for lightweight group key management protocols for IoT and the challenges that make designing such protocols difficult have been discussed. In this section, group key management protocols proposed to address some of the challenges regarding group key establishment and session management will be presented. The research contributions presented here are part of an attempt to answer the research questions posed in Chapter 1. The presentation here is essentially a summary of the protocols proposed in the papers attached at the end of this book. The contributions that underpin this thesis are the following:

1. Paper I - Lightweight IoT Group Key Establishment Scheme Using One-Way Accumulator.
2. Paper II - Lightweight Group-Key Establishment Protocol for IoT Devices: Implementation and Performance Analyses.
3. Paper III - Lightweight IoT Group Key Establishment Scheme From the One Time Pad.
4. Paper IV - Survey of Proximity-Based Authentication Mechanisms for the Industrial Internet of Things.

4.2.1 Paper I - Lightweight IoT Group Key Establishment Scheme Using One-Way Accumulator

Paper I presents a lightweight group key management scheme from a lightweight cryptographic primitive based on algebra on elliptic curves and the notion of cryptographic one-way accumulator. The proposed scheme answers the first research question (**RQ1**) posed in Chapter 1. The scheme enables a gateway and a set of end nodes to establish a keying material among themselves, manage session keys and node membership securely. Three defining characteristics of the scheme are that it is built from lightweight cryptographic primitives; that it enables nodes to establish a keying material in $\mathcal{O}(1)$, meaning that the amount of computation and rounds on

each node is independent of the number of nodes in the group; and that it makes it possible for each member node to generate an equivalent of a digital signature when sending a group message.

The entire security of the scheme emanates from the fact that ECDLP is supposedly computationally hard and the observation that elliptic curve point multiplication can effectively be used as a cryptographic one-way accumulator function. The first point is a standard assumption in cryptography and has been briefly discussed in Chapter 2. The second point regarding using elliptic curve point multiplication as a cryptographic one-way accumulator and the rationale for using it warrants further explanation.

First, let us see that a function f that takes as inputs an arbitrary integer n and a point on P on an elliptic curve E defined over a finite field F_q, for prime q, and outputs the scalar multiplication of n and P, is indeed a cryptographic one-way accumulator. By the ECDLP assumption, we have that $f(n, P) = nP$ is a one-way function [HILL99]. By definition of elliptic curve point multiplication,

$$f(n, P) = \underbrace{P + P + \cdots + P}_{n} = Q$$

and by the closure property of the group $E(F_q)$, Q is another point on the elliptic curve. Therefore, for any integer m, not necessarily distinct from n, we have,

$$\begin{aligned}
f(m, Q) = f(m, f(n, P)) &= \underbrace{Q + Q + \cdots + Q}_{m} \\
&= \underbrace{\underbrace{P + P + \cdots + P}_{n} + \underbrace{P + P + \cdots + P}_{n} + \cdots + \underbrace{P + P + \cdots + P}_{n}}_{m} \\
&= \underbrace{P + P + \cdots + P}_{mn} \\
&= \underbrace{\underbrace{P + P + \cdots + P}_{m} + \underbrace{P + P + \cdots + P}_{m} + \cdots + \underbrace{P + P + \cdots + P}_{m}}_{n} \\
&= f(n, f(m, P))
\end{aligned}$$

So that $f(m, f(n, P)) = f(n, f(m, P))$, satisfying the quasi-commutative property that a cryptographic one-way accumulator needs to satisfy. The third requirement for f to be a one-way accumulator is that it be efficiently computable. The *repeated square-and-multiply* [GLV01] algorithm is a fast and efficient method to do elliptic curve scalar point multiplication. Therefore, f is efficiently computable.

The observation that point multiplication on elliptic curves can be used as a one-way accumulator can be exploited to enable a group of nodes and a trusted gateway to establish a common keying material easily and securely. Assume that P

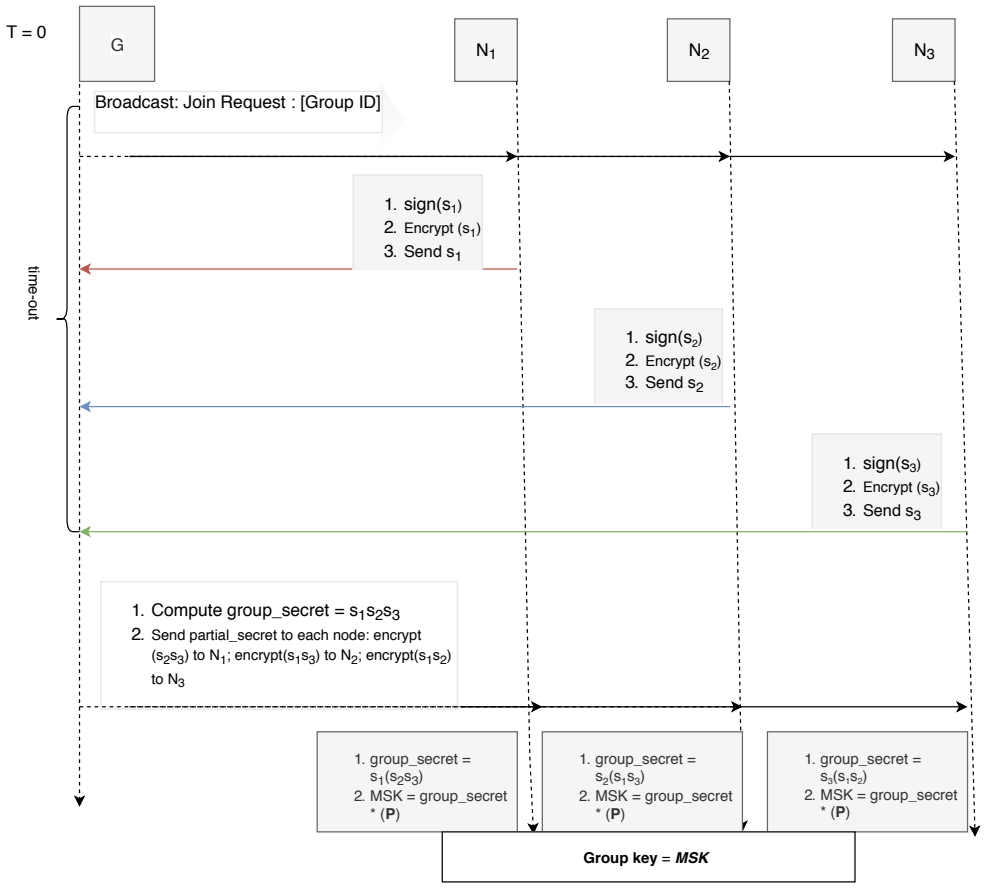

Figure 4.2: Group Key Initialization: The figure shows the messages exchanged between G and three end nodes replying to the join request before the specified timeout.

is a generator of the additive group of points of the points of an elliptic curve E over F_q. If $s_1, s_2, \cdots, s_n$ are secretly shared integers between gateway G and nodes $N_1, N_2, \cdots, N_n$. For each node N_i, G simply sends the value we call $partial\_secret = (s_1 * s_2 * \cdots s_{i-1} * s_{i+1} * \cdots s_n)P$. Node N_i then recovers the group keying material, which we call $group\_secret$ by simply multiplying $partial\_secret$ by s_i. A run of the protocol for the case when $n = 3$ is shown in Fig. 4.2.

The detailed security proofs and the other aspects of the protocol – digital signature, session key generation and group membership management – can be found in **Paper** I.

The novelty of this protocol lies in its reliance on a single lightweight cryptographic primitive to provide the aforementioned group key management services. Crucially, the protocol makes it possible for resource-constrained IoT nodes to es-

tablish a keying material in constant time regardless of the number of devices in the group. Furthermore, a node can be securely added or removed from the group in constant time. The fact that all these operations and computations can be effected in constant time makes the protocol a good solution for secure group establishment in constrained environments with dynamic groups. To to best of the authors' knowledge, there is no other group key management protocol that provides the aforementioned services in constant time.

4.2.2 Paper II - Lightweight Group-Key Establishment Protocol for IoT Devices: Implementation and Performance Analyses

As the tittle indicates, **Paper** II in an extension of Contribution I. **Paper** II presents implementation results of the protocol proposed in I. The main purpose of this experiment was to show that the proposed protocol is feasible for implementation in tiny IoT devices and to demonstrate that the protocol provides considerable improvements when compared to state-of-the-art group key management solutions with regard to energy and memory consumption and number of computations performed at each node. It is worth mentioning here that strict comparison with other protocols is not possible because other solutions use other primitives and other implementation mechanisms.

Experimental Setup

The network model depicted in Fig. 4.1 was simulated in Contiki OS, using the Cooja simulator [Seh13]. The ecc-nano [iSE] implementation of ECDH [LN08]. The protocol was implemented in a group of Tmote Sky devices, whose description is shown in Fig. 4.1.

Resource	
Operating Voltage	3 V
Microcontroller	(16 bit)8 MHz
RAM	10 KB
ROM	48 KB
Low Power Mode (LPM)	0.0545 mA
Current consumption TX mode	19.5 mA
Current consumption RX mode	21.8 mA
Ticks/second	327680

Table 4.1: Tmote Sky

Evaluation and Results

The objective of the experiment was to test the practicality of the protocol when implemented on tiny IoT devices, such as those with 8-bit microprocessors. In small IoT devices, sometimes battery powered, the energy that a device expends to run a protocol, the amount of computations performed and the amount of storage required to store data needed by the protocol are, among other things, important parameters that protocol designers need to minimize [HJFA13]. Therefore, we measured the performance of the protocol with respect to these three parameters.

Task	Energy (mJ)	Duration (s)
Generate signature (ECDSA)	2.197	19.68
Verify signature	2.006	17.112
ECDH	2.197	19.206
EC Point Multiplication	2.085	19.199
AES Encrypt	≈ 0	0.010
AES Dencrypt	≈ 0	0.014

Table 4.2: Energy and execution time consumed by each cryptographic primitive that is part of the protocol.

Tables 4.2 and 4.3 show the total amount of resources (memory, energy and time) that the protocol requires. In terms of memory, each node needs to allocate only less than 150 bytes of memory to store both the secret and network parameters. Only 30763 bytes are needed to store and run the program code. Even extremely tiny devices, like the one described in Table 4.1, can feasibly run the protocol as far as memory is concerned. The time each node took to execute the constituent cryptographic primitives is also shown in Table 4.2, with the elliptic curve related

Resource	Dimension (Bytes)
$PubKey$	48
$PrivKey$	24
$GroupID$	1
$GatewayPubKey$	48
MSK	24

Table 4.3: The amount of memory required to store the protocol parameters.

Table 4.4: Computational overhead for each step. PM =Elliptic Curve Point Multiplication. V = Signature Verification; S= Signing. PA = Elliptic Curve Point Addition. AES_E = AES Encryption. DES_D = AES Decryption

	Node Computation Overhead	
Phase	**Our Protocol**	**Protocol 2 [PBS$^+$15]**
Key establishment	$V + S + PM$	$4PM + 3PA$
Adding new node	$V + S + PM$	$4PM + 3PA$
Removing nodes	$V + S + (AES_D \approx 0) + PM$	$4PM + 3PA$

computations inevitably taking longer time to execute, but still within an acceptable range [SOS$^+$08]. It is worth noting that no attempt was made to optimize the implementation by doing precomputation of elliptic curve point multiplication [SOS$^+$08] or by choosing elliptic curves that are known to be fast for implementation [LD99].

The aforementioned measures are device and implementation specific in that a more optimized implementation in a different hardware or simulation would yield a completely different result. Therefore, they can not be used to compare the complexity of our protocol with other similar protocols. The sole purpose of presenting these results is to show that the protocol is lightweight and that it could be reasonably deployed in a real world IoT application scenario.

In Table 4.4, a comparison of the protocol proposed here to another protocol which uses similar cryptographic primitives proposed in [PBS$^+$15] is provided. Instead of focusing on how much resource each computation consumes, which is hardware and implementation dependent, this table shows how many times each cryptographic primitive is computed for each task in our protocol: key establishment, adding a new node and removing a node. The most resource intensive computation, point multiplication, is required only once in each task of our protocol, whereas in [PBS$^+$15], elliptic curve point multiplication is invoked 4 times, in addition to 3 elliptic curve points additions.

4.2.3 Paper III - Lightweight IoT Group Key Establishment Scheme From the One Time Pad

In **Paper** III, a group key management scheme from symmetric key cryptographic constructions is proposed. This contribution is an attempt to address the second research question (**RQ2**) posed in Chapter 1. The security of key management protocols based on public key primitives relies on the supposed hardness of one or more of the problem discussed in Chapter 2. The reliance on supposedly hard problems has two security implications. First, it is not known for a fact that the problems are indeed computationally hard and therefore in the unlikely event that they are not, the security of protocols based on them breaks. Second, the supposed computational hardness of the problems is a function of some security parameter such as order of

a group [Mil85]: the bigger the security parameter the more computationally hard the problem supposedly is. To keep up with the increasing computational capability, problems need to be defined with a sufficiently large security parameters [BP97]. As a result, a crypto-system defined on a supposedly computationally hard problem becomes computationally intensive for tiny IoT devices. Although symmetric key based primitives are inherently more lightweight, they are not a natural fit for key management purposes. In fact, it is because symmetric key based primitives do not generally provide a mechanism for key management that public key cryptography was invented in the first place [DH76]. In **Paper** III, a group key management protocol whose security is underpinned by the perfect secrecy guarantee of the OTP is proposed to reconcile this apparent conundrum. The routine computations related to key establishment and session key management are simple bit XOR computations, whereas tasks related to authentication are left to public key based solutions. The result is that unnecessary reliance on computationally intensive public key computations is avoided.

The protocol construction is simple. Each node N_i and the gateway G secretly share an n-bit value through some off-band mechanism. The value n should be sufficiently large so that the value $\frac{1}{2^n}$ is negligible, as defined in Chapter 2. A one-round run of the protocol in a group consisting of a gateway and three nodes is depicted in Fig. 4.3.

The protocol has two novelties. First, the protocol is lightweight since OTP involves only bitwise XOR operation, which is fast. Second, the protocol is not just computationally but also unconditionally secure. The second point is too strong of a claim and as such warrants further explanation. The proposed protocol addresses only aspects related to group group key establishment and session key management, not issues related to authentication and identification. However, designing device identification and authentication is part of key management, and it is a complex problem in and of itself [BS11]. To the best knowledge of the author, there is no key management protocol that solves all key management related security problems in an information-theoretic (unconditionally) secure way. The caveat to the claim that the protocol proposed in **Paper** III is unconditionally secure is that the claim does not include aspects related to authentication. In a security model where devices trust each other (which is not practical) the claim that the protocol is unconditionally secure holds, although it is not true in general in other models.

Complexity Analysis

Regardless of the number of nodes in a group, each node computes two XOR operations of n-bit values, resulting in $\mathcal{O}(1)$ computational overhead to complete participation in the protocol. Similarly, node addition or removal is effected with each member node performing one XOR operation of two n-bit values. The number of XOR operations of n-bit values at the gateway grows linearly with the number of nodes in the group. This is an acceptable overhead since most gateways in real-word IoT deployments are capable of doing much more complex computations [DBN14].

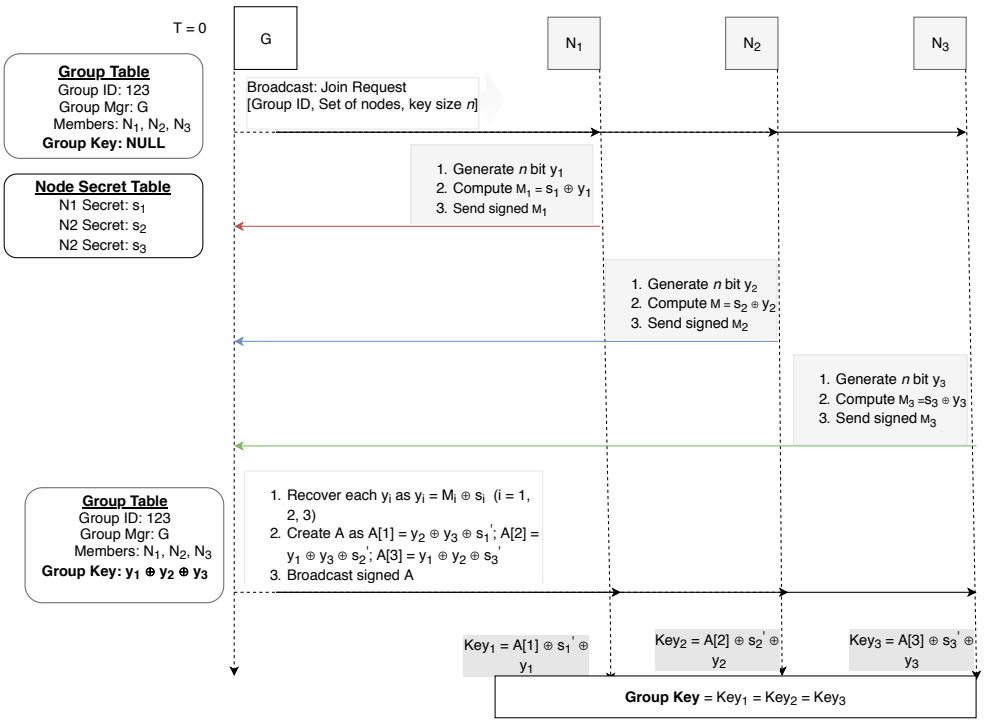

Figure 4.3: Initialization process. The figure shows control messages in the initialization stage between three end nodes and the gateway to establish a group key. s_i is an n-bit secret shared between node N_i and G. y_i is a n-bit value randomly picked from n-bit message space $\mathcal{M}$. Each $s_i^{'}$ value is derived from s_i, by flipping every other bit of s_i. The details, including security proof, can be found in **Paper** III.

4.2.4 Paper IV - Survey of Proximity-Based Authentication Mechanisms for the Industrial Internet of Things

Paper IV presents a survey of various proximity-based authentication mechanisms used in different IoT application contexts today. The work presented in this paper is not a core component of the thesis but complements the previously discussed works on group key management. In previous works, devices' (mutual) authentication was taken for granted; in that the statement "Node A sends a message to node B" meant that A is indeed communicating with B. However, there is no reason to take this claim for granted, since impersonation attacks are commonplace in IoT [TBH14]. Therefore, key management protocols need to have authentication mechanisms as an integral part.

There are generally various authentication mechanisms in use on the Internet, such as those based on PKI, traditional passwords and biometric information. However, none of these are suitable for IoT devices due to the computationally limited nature of IoT devices and their lack of input interface to enter credentials such as passwords. The unsuitability of conventional authentication mechanisms has spurred the need to devise new authentication mechanisms that take the unique nature of IoT devices into account. One such authentication mechanism is based on the physical proximity of IoT devices, termed proximity-based authentication mechanism.

Paper IV presents a survey of seven proximity-based authentication mechanisms used in IoT: wire-, radio-, acoustic-, light-, gesture- and biometric-based authentication mechanisms. The novelty of this contribution is that it is, to the best of our knowledge, the first comprehensive survey of such schemes, highlighting open problems in each authentication mechanism and suggesting possible research directions.

Chapter 5

Conclusions

In chapters 1 through 4, the need for having group key management protocols for IoT, the unique challenges in designing such protocols, and the contributions as part of this thesis work to address some of the challenges have been presented. This chapter presents an overview of the work done, evaluation of the presented work vis-á-vis the research goals stated in Chapter 1, as well as a discussion of the direction for future work.

5.1 Overview

The main purpose of this work is to contribute to the literature on the design and implementation of lightweight group key management protocols that can be deployed in various IoT group applications. The work focused on understanding and presenting the challenges involved in designing lightweight group key management protocols and proposing new schemes that address some of the challenges identified.

In general, a key management protocol solves three closely interrelated problems: devices authentication, key establishment and session key management. The main challenge in designing a protocol that addresses these problems is ensuring that the protocol is lightweight; that is, that the protocol is constituted by cryptographic primitives that are computationally feasible for resource limited devices.

In an effort to partially address the aforementioned challenge, two lightweight group key management protocols were proposed, each following a different approach. In the first work, a key management protocol was proposed from a lightweight public key construction: based on elliptic curves. The protocol enables resource limited IoT devices in a group to establish a cryptographic key among themselves, generate session group keys, manage device leave/join operation and create the equivalent of digital signature. The main contribution of this work is demonstrating that the supposed hardness of ECDLP can be combined with the the notion

of cryptographic one-way accumulator to create a group key management protocol in such a way that each task in the protocol can be achieved in $\mathcal{O}(1)$ time. To show the protocol is practically feasible, it was implemented in a simulated environment on Contiki. The results show that the protocol is indeed lightweight, but also confirms that there is an inherent level of complexity associated with public key based protocols. The second work is an attempt to explore the possibility of designing a group key management systems from symmetric key cryptographic primitives so that the complexities inherent to public key based protocols can be avoided. To this end, a protocol whose security is underpinned by the perfect secrecy of the OTP was proposed. The main contribution of this work is demonstrating that the lightweight and perfect secrecy nature of the OTP can be harnessed for use in IoT group management.

5.2 Outcome

Two new lightweight group key management protocols were proposed and one of them also implemented to achieve the three research goals outlined in Chapter 1. The research goals are listed again here for ease of readability.

1. *Research Goal 1 (RG1):* Construct a lightweight group key management protocol from lightweight public key cryptographic primitives.
2. *Research Goal 2 (RG2):* Construct a lightweight group key management protocol from lightweight symmetric key primitives.
3. *Research Goal 3 (RG3):* Evaluate the various authentication mechanisms used in IoT environments and suggest new secure authentication mechanisms for future use.

Roughly, the contribution presented in **Papers** I and II achieves *RG1*; the contribution presented in **Paper** III achieves *RG2*; and the contribution presented in **Papers** IV achieves *RG3*. Below, a discussion of how each presented contribution achieves the stated goals is provided.

***RG1**: Construct a lightweight group key management protocol from lightweight public key cryptographic primitives.*

Achieving **RG1** required studying and investigating various public key based cryptographic primitives. By design, public key primitives are convenient for solving key management related problems, with the sometimes undesirable property of being computationally complex. The challenge involved in achieving **RG1** was to investigate and find a combination of lightweight public key based primitives to construct a protocol that addresses the various aspects of group key management discussed in previous chapters. This challenge was addressed by exploiting the supposed computational hardness of ECDLP and using it as a cryptographic one-way accumulator to build a group key management protocol that enables devices to share a group key, generate group session keys and digital signatures. Implementation

of the protocol showed that it is feasible for practical purposes but that there is an inherent level of complexity associated with public key cryptography based primitives. For example, scalar point multiplication on an elliptic curve took roughly 19 seconds on a simulated Tmote Sky with a 16-bit processor. Although the time and other resources consumed by the protocol execution can be reduced by optimizing the implementation and using faster curves, there is still a considerable amount of inherent slowness and complexity that cannot be avoided.

***RG2**: Construct a lightweight group key management protocol from lightweight symmetric key primitives.*

Symmetric key based cryptographic systems are naturally more lightweight, owing to the fact that they are not generally reliant on heavyweight mathematical computations. This fact is specially true of the OTP, which involves only computing XOR or bits. The purpose of pursuing this goal was to exploit the lightweight nature and perfect secrecy of OTP by designing a group key management protocol such that the OTP is the core of it. To this end, a protocol was designed underpinned by the OTP that enables tiny devices in a group to establish a group key among themselves and generate session group keys, all in constant time regardless of the number of devices in the group.

The literature on lightweight group key management protocols is rich with works that rely on public key cryptographic systems. This is no coincidence. Public key crypto-systems were invented to solve what is generally considered to be the Achilles heel of symmetric key crypto-systems: key management. The protocol proposed to satisfy **RG2** shows that lightweight and secure key management protocols can be designed from symmetric key primitives, and particularly from the OTP. The benefits of this are twofold: first, the protocol is lightweight and, second, the protocol enables the devices in a group to share a keying material of any bit length without it being computationally complex.

***RG3**: Evaluate the various authentication mechanisms used in IoT environments and suggest new secure authentication mechanisms for future use.*

This research goal was pursued to complement the other two with regard to device authentication. To achieve **RG1** and **RG2**, it was assumed that there is an authenticated channel between any two devices in a network. This means that the problem the protocols proposed are not authenticated key management protocols. The purpose of setting this goal was to investigate various authentication mechanisms that could feasibly be used in IoT environments, paving the way for future design of authenticated key management protocols. The outcome of this particular research endeavor is a comprehensive and thorough presentation of commonly used proximity-based authentication mechanisms, pointing out open challenges, as well as suggestions for future design of secure authentication mechanisms.

5.3 Impact

The research presented in this thesis work dealt with designing new security protocols that could be used in various IoT applications that are part of our daily lives. The novelty introduced in the proposed protocols will contribute to the body of literature on lightweight cryptographic protocols. Given that the domain of the research is IoT applications used in various consumer and commercial domains, the research has social impact as well.

5.3.1 Scientific Impact

The scientific impact of the contributions comes from the novelty of the proposed key management protocols. As part of this thesis work, new lightweight group key management protocols for IoT showing new ways of constructing protocols from secure and lightweight cryptographic primitives, new ways of generating a digital signature in group communication, and efficient ways of managing session keys were proposed. The novelty of the protocols will add to the literature on how to securely and effectively design security protocols suitable for resource-constrained devices. Furthermore, the new design approach followed to construct the protocols will hopefully serve as an input for other researchers when designing other protocols, which will further push the research frontier on lightweight cryptographic protocols. Furthermore, the fact that the the key management protocols were built from cryptographic primitives not typically designed for that purpose will open new alternatives for other researchers to explore. The contributions presented here have also laid the groundwork for future work by this author.

5.3.2 Social Impact

IoT applications are becoming part and parcel of our daily lives in various areas such as home automation, industrial automation and smart transport. The key management protocols presented in this thesis could become integral components of the security protocol suites that provide security services such as confidentiality, safety and privacy in the aforementioned IoT applications. For IoT applications to be trusted by users, it is important that the applications have strong security mechanisms built-in. Given that key management is a prerequisite for most security aspects of IoT applications that we rely on in our daily lives, it is easy to see that the research presented in thesis is highly applicable.

5.4 Ethical Considerations

Ethical aspects of designing cryptographic protocols include deliberately weakening an algorithm with a malign intention of attacking the user at a later time [SFKR15], introducing a backdoor to an algorithm [YY05], designing cryptographic protocols

that could easily be broken by third parties, such as governments, by way of key escrow [DS94], and similar nefarious practices that make cryptographic protocols fail to do what they purport to.

The protocols presented here are based on publicly vetted cryptographic primitives and the security services they claim to provide are duly proved. Moreover, a justification of why and how they are useful to secure IoT applications is thoroughly provided. Therefore, there is no room left for the aforementioned types of ethical shenanigans.

In terms of the bigger picture of upholding scientific integrity, the research has been conducted with utmost honesty. No part of the thesis has been plagiarized and all other works that have served as input to this research have been properly credited.

5.5 Future Work

The proposed protocols presented in this thesis rely on a network model where there is at least one trust anchor that coordinates the key establishment process. This model has two obvious drawbacks: (i) as the number of devices in the network increases, the entity that acts as a trust anchor may prove to be a bottleneck and (ii) there is generally a need to deploy IoT networks in an ad hoc manner without a central entity. One research direction for future work is to devise a mechanism to develop key management protocols for distributed, peer-to-peer IoT networks without a central entity. The challenge to be solved in this setting is to solve the problem of how two or more devices with no prior interaction can establish trust between themselves. In IP-connected IoT networks, trust between two IoT devices can be established with the help of the PKI infrastructure, but this does not generally work, since IoT networks typically run on non-IP networks, and the inherent complexity of PKI is not suitable for IoT devices even when they are IP-enabled. Therefore, it would be interesting to study and design other mechanisms of establishing trust, and relying on that, design key management protocols for IoT networks without a central trusted entity. One such mechanism that exploits physical proximity of devices is proposed in [MMV+11], but it does not work in general, since two IoT devices are generally not expected to be within a fixed physical proximity to each other.

Another future research direction relates to designing quantum-safe [CCJ+16] versions of the protocols proposed here. The protocols presented in this thesis are underpinned by classical cryptographic primitives. The working assumption in designing security protocols that rely on classical cryptography is that a practical quantum computer is a distant possibility. However, as Bernstein argues in [Ber09], it is important that researchers develop quantum safe versions of the security protocols to fall back to in the invent that quantum computers become a reality. Against this background, it would be interesting to explore the possibility of designing lightweight and quantum-safe group key management protocols that are practical for resource-constrained IoT devices.

Bibliography

[A+09] Kevin Ashton et al. That "internet of things" thing. *RFID journal*, 22(7):97–114, 2009.

[ÁBLLR16] José Antonio Álvarez-Bermejo, Antonela Lodroman, and Juan Antonio López-Ramos. Distributed key agreement for group communications based on elliptic curves. An application to sensor networks. *Mathematical Methods in the Applied Sciences*, 39(16):4797–4809, 2016.

[ACL+04] Jari Arkko, Elisabetta Carrara, Fredrik Lindholm, Mats Naslund, and Karl Norrman. Mikey: Multimedia internet keying. Technical report, 2004.

[AL03] Carlisle Adams and Steve Lloyd. *Understanding PKI: concepts, standards, and deployment considerations*. Addison-Wesley Professional, 2003.

[ASSC02] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[BDM93] Josh Benaloh and Michael De Mare. One-way accumulators: A decentralized alternative to digital signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 274–285. Springer, 1993.

[Bel02] Mihir Bellare. A Note on Negligible Functions. *Journal of Cryptology*, 15(4), 2002.

[Ber09] Daniel J Bernstein. Introduction to post-quantum cryptography. In *Post-quantum cryptography*, pages 1–14. Springer, 2009.

[BF01] Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology — CRYPTO 2001*, pages 213–229, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[BL96] Dan Boneh and Richard J Lipton. Algorithms for black-box fields and their application to cryptography. In *Annual International Cryptology Conference*, pages 283–297. Springer, 1996.

[Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1. In *Annual International Cryptology Conference*, pages 1–12. Springer, 1998.

[BMP00] Victor Boyko, Philip MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 156–171. Springer, 2000.

[BMS+06] Lejla Batina, Nele Mentens, Kazuo Sakiyama, Bart Preneel, and Ingrid Verbauwhede. Low-cost elliptic curve cryptography for wireless sensor networks. In *European Workshop on Security in Ad-hoc and Sensor Networks*, pages 6–17. Springer, 2006.

[BP97] Niko Barić and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 480–494. Springer, 1997.

[Bri99] Bob Briscoe. MARKS: Zero side effect multicast key management using arbitrarily revealed key sequences. In *International Workshop on Networked Group Communication*, pages 301–320. Springer, 1999.

[BS11] Debasis Bandyopadhyay and Jaydip Sen. Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69, 2011.

[BSP+11] Sachin Babar, Antonietta Stango, Neeli Prasad, Jaydip Sen, and Ramjee Prasad. Proposed embedded security framework for internet of things (iot). In *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, pages 1–5. IEEE, 2011.

[CBCM12] Luca Costantino, Novella Buonaccorsi, Claudio Cicconetti, and Raffaella Mambrini. Performance analysis of an LTE gateway for the IoT. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–6. IEEE, 2012.

[CCJ+16] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[Cho06] Kim-Kwang Raymond Choo. *Key Establishment: Proofs and Refutations*. PhD thesis, Queensland University of Technology, 2006.

[CKT91] Peter C Cheeseman, Bob Kanefsky, and William M Taylor. Where the really hard problems are. In *IJCAI*, volume 91, pages 331–340, 1991.

[DBN14] Soumya Kanti Datta, Christian Bonnet, and Navid Nikaein. An IoT gateway centric architecture to provide novel M2M services. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, pages 514–519. IEEE, 2014.

[DGK+00] Antonio DeSimone, Joseph Golan, Ashok K Kuthyar, Bryant Richard Parent, Ram S Ramamurthy, and David Hilton Shur. Method for managing multicast addresses for transmitting and receiving multimedia conferencing information on an internet protocol (IP) network, January 4 2000. US Patent 6,011,782.

[DGV04] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE, 2004.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[DL04] Fu-Guo Deng and Gui Lu Long. Secure direct communication with a quantum one-time pad. *Physical Review A*, 69(5):052319, 2004.

[DS94] Dorothy E Denning and Miles Smid. Key escrowing today. *IEEE Communications Magazine*, 32(9):58–68, 1994.

[EKP+07] Thomas Eisenbarth, Sandeep Kumar, Christof Paar, Axel Poschmann, and Leif Uhsadel. A survey of lightweight-cryptography implementations. *IEEE Design & Test of Computers*, (6):522–533, 2007.

[EPG18] Mohamed H Eldefrawy, Nuno Pereira, and Mikael Gidlund. Key Distribution Protocol for Industrial Internet of Things without Implicit Certificates. *IEEE Internet of Things Journal*, 2018.

[Fin10] Klaus Finkenzeller. *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons, 2010.

[Fra03] John B Fraleigh. *A first course in abstract algebra*. Pearson Education India, 2003.

[FSK11] Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno. *Cryptography engineering: design principles and practical applications*. John Wiley & Sons, 2011.

[GBMP13] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[GLV01] Robert P Gallant, Robert J Lambert, and Scott A Vanstone. Faster point multiplication on elliptic curves with efficient endomorphisms. In *Annual International Cryptology Conference*, pages 190–200. Springer, 2001.

[Gol09] Oded Goldreich. *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[HBK+14] Jan Höller, David Boyle, Stamatis Karnouskos, Stefan Avesand, Catherine Mulligan, and Vlasios Tsiatsis. *From machine-to-machine to the internet of things*. Elsevier, 2014.

[HGMH+11] Tobias Heer, Oscar Garcia-Morchon, René Hummen, Sye Loong Keoh, Sandeep S Kumar, and Klaus Wehrle. Security Challenges in the IP-based Internet of Things. *Wireless Personal Communications*, 61(3):527–542, 2011.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HJFA13] Chong Han, Josep Miquel Jornet, Etimad Fadel, and Ian F Akyildiz. A cross-layer communication module for the Internet of Things. *Computer Networks*, 57(3):622–633, 2013.

[HMV06] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone. *Guide to elliptic curve cryptography*. Springer Science & Business Media, 2006.

[IOV+17] SK Hafizul Islam, Mohammad S Obaidat, Pandi Vijayakumar, Enas Abdulhay, Fagen Li, and M Krishna Chaitanya Reddy. A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs. *Future Generation Computer Systems*, 2017.

[iSE] iSECPartners. A very small ECC implementation for 8-bit microcontrollers. 2018-11-26.

[JMV01] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security*, 1(1):36–63, 2001.

[JVW+14] Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the Internet of Things: perspectives and challenges. *Wireless Networks*, 20(8):2481–2501, Nov 2014.

[KAS08] Vivek Kapoor, Vivek Sonny Abraham, and Ramesh Singh. Elliptic curve cryptography. *Ubiquity*, 2008(May):7, 2008.

[KH18] Yi-Hsuan Kung and Hsu-Chun Hsiao. GROUPIT: Lightweight Group Key Management for Dynamic IoT Environments. *IEEE Internet of Things Journal*, 2018.

[KKA13] Arun Kanuparthi, Ramesh Karri, and Sateesh Addepalli. Hardware and embedded security in the context of internet of things. In *Proceedings of the 2013 ACM workshop on Security, privacy & dependability for cyber vehicles*, pages 61–64. ACM, 2013.

[KKT14] Sye Loong Keoh, Sandeep S Kumar, and Hannes Tschofenig. Securing the internet of things: A standardization perspective. *IEEE Internet of Things Journal*, 1(3):265–275, 2014.

[KM07] Neal Koblitz and Alfred J Menezes. Another look at" provable security". *Journal of Cryptology*, 20(1):3–37, 2007.

[KM08] Masanobu Katagi and Shiho Moriai. Lightweight cryptography for the internet of things. *Sony Corporation*, pages 7–10, 2008.

[KMV00] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. *Designs, codes and cryptography*, 19(2-3):173–193, 2000.

[KMVOV96] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. *Handbook of applied cryptography*. CRC press, 1996.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[KOO17] Adnan Kiliç, Ertan Onur, and Cansu Betin Onur. Revisiting shamirâs no-key protocol: Lightweight key transport. In *Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence & Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), 2017 IEEE 15th Intl*, pages 573–580. IEEE, 2017.

[KPT04] Yongdae Kim, Adrian Perrig, and Gene Tsudik. Tree-based group key agreement. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):60–96, 2004.

[Lau04] Kristin Lauter. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless communications*, 11(1):62–67, 2004.

[LD99] Julio López and Ricardo Dahab. Fast multiplication on elliptic curves over GF (2 m) without precomputation. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 316–327. Springer, 1999.

[LH14] Yi-Pin Liao and Chih-Ming Hsiao. A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. *Ad Hoc Networks*, 18:133–146, 2014.

[LN08] An Liu and Peng Ning. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 245–256. IEEE Computer Society, 2008.

[LX13] Fagen Li and Pan Xiong. Practical secure communication for integrating wireless sensor networks into the internet of things. *IEEE Sensors Journal*, 13(10):3677–3684, 2013.

[LXC12] Jing Liu, Yang Xiao, and CL Philip Chen. Authentication and access control in the internet of things. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 588–592. IEEE, 2012.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 417–426. Springer, 1985.

[Mil99] C Kenneth Miller. *Multicast networking and applications*. Addison-Wesley Reading, 1999.

[MMV+11] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. Proximate: proximity-based secure pairing using ambient wireless signals. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 211–224. ACM, 2011.

[MNG17] Zahid Mahmood, Huansheng Ning, and AtaUllah Ghafoor. A polynomial subset-based efficient multi-party key management system for lightweight device networks. *Sensors*, 17(4):670, 2017.

[MOV93] Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *iEEE Transactions on information Theory*, 39(5):1639–1646, 1993.

[MP04] Daniele Micciancio and Saurabh Panjwani. Optimal communication complexity of generic multicast key distribution. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 153–170. Springer, 2004.

[MWS04] David J Malan, Matt Welsh, and Michael D Smith. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 71–80. IEEE, 2004.

[Ng05] Eddie M Ng. Security models and proofs for key establishment protocols. Master's thesis, University of Waterloo, 2005.

[Opp11] Rolf Oppliger. *Contemporary cryptography*. Artech House, 2011.

[PBS+15] Pawani Porambage, An Braeken, Corinna Schmitt, Andrei V Gurtov, Mika Ylianttila, and Burkhard Stiller. Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications. *IEEE Access*, 3(1):1503–1511, 2015.

[Pos09] Axel York Poschmann. Lightweight cryptography: cryptographic engineering for a pervasive world. In *PH. D. THESIS*. Citeseer, 2009.

[PP09] Christof Paar and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[PP18] Chang-Seop Park and Wang-Seok Park. A Group-Oriented DTLS Handshake for Secure IoT Applications. *IEEE Transactions on Automation Science and Engineering*, (99):1–10, 2018.

[PSC+05] Adrian Perrig, Dawn Song, Ran Canetti, JD Tygar, and Bob Briscoe. Timed efficient stream loss-tolerant authentication (TESLA): Multicast source authentication transform introduction. Technical report, 2005.

[QHG+18] Umair Mujtaba Qureshi, Gerhard Petrus Hancke, Teklay Gebremichael, Ulf Jennehag, and Mikael Gidlund. Survey of Proximity Based AuthenticationMechanisms for the Industrial Internet of Things. In *The 44th Annual Conference of the IEEE Industrial Electronics Society (IECON2018), Washington DC, October 21-23, 2018.*, 2018.

[RALS11] Rodrigo Roman, Cristina Alcaraz, Javier Lopez, and Nicolas Sklavos. Key management systems for sensor networks in the context of the Internet of Things. *Computers & Electrical Engineering*, 37(2):147–159, 2011.

[Raz13] Shahid Raza. *Lightweight security solutions for the internet of things*. PhD thesis, Mälardalen University, Västerås, Sweden, 2013.

[RSSS16] Shahid Raza, Ludwig Seitz, Denis Sitenkov, and Göran Selander. S3K: Scalable Security With Symmetric Keys-DTLS Key Establishment for the Internet of Things. *IEEE Transactions on Automation Science and Engineering*, 13(3):1270–1280, 2016.

[SCP+15] Savio Sciancalepore, Angelo Capossele, Giuseppe Piro, Gennaro Boggia, and Giuseppe Bianchi. Key management protocol with implicit certificates for IoT systems. In *Proceedings of the 2015 Workshop on IoT challenges in Mobile and Industrial Systems*, pages 37–42. ACM, 2015.

[Seh13] Anuj Sehgal. Using the contiki cooja simulator. *Computer Science, Jacobs University Bremen Campus Ring, Technical Report*, 2013.

[SFKR15] Bruce Schneier, Matthew Fredrikson, Tadayoshi Kohno, and Thomas Ristenpart. Surreptitiously weakening cryptographic systems. *IACR Cryptology ePrint Archive*, 2015:97, 2015.

[Sha49] Claude E Shannon. Communication theory of secrecy systems. *Bell system technical journal*, 28(4):656–715, 1949.

[shi] one time password authentication scheme based on elliptic curves for internet of things (iot).

[Sho94] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. Ieee, 1994.

[SO12] Yosra Ben Saied and Alexis Olivereau. D-HIP: A distributed key exchange scheme for HIP-based Internet of Things. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a*, pages 1–7. IEEE, 2012.

[SOS+08] Piotr Szczechowiak, Leonardo B Oliveira, Michael Scott, Martin Collier, and Ricardo Dahab. NanoECC: Testing the limits of elliptic curve cryptography in sensor networks. In *Wireless sensor networks*, pages 305–320. Springer, 2008.

[SSG13] Ludwig Seitz, Göran Selander, and Christian Gehrmann. Authorization framework for the internet-of-things. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*, pages 1–6. IEEE, 2013.

[Sta14] John A Stankovic. Research directions for the internet of things. *IEEE Internet of Things Journal*, 1(1):3–9, 2014.

[Sti05] Douglas R Stinson. *Cryptography: theory and practice*. CRC press, 2005.

[STW96] Michael Steiner, Gene Tsudik, and Michael Waidner. Diffie-Hellman key distribution extended to group communication. In *Proceedings of the 3rd ACM conference on Computer and communications security*, pages 31–37. ACM, 1996.

[TBH14] Muhamed Turkanović, Boštjan Brumen, and Marko Hölbl. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Networks*, 20:96–112, 2014.

[TVS07] Andrew S Tanenbaum and Maarten Van Steen. *Distributed systems: principles and paradigms*. Prentice-Hall, 2007.

[Ver26] Gilbert S Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Transactions of the American Institute of Electrical Engineers*, 45:295–301, 1926.

[VMQ98] Scott Vanstone, Alfred John Menezes, and Minghua Qu. Key agreement and transport protocol with implicit signatures, June 2 1998. US Patent 5,761,305.

[WGL00] Chung Kei Wong, Mohamed Gouda, and Simon S Lam. Secure group communications using key graphs. *IEEE/ACM Transactions on Networking (TON)*, 8(1):16–30, 2000.

[WK03] Jonathan Walpole and Charles Krasic. Priority progress multicast streaming for quality-adaptive transmission of data, December 25 2003. US Patent App. 10/177,864.

[WSE14] Michael Weyrich, Jan-Philipp Schmidt, and Christof Ebert. Machine-to-machine communication. *IEEE Software*, 31(4):19–23, 2014.

[WW10] Rolf H Weber and Romana Weber. *Internet of things*, volume 12. Springer, 2010.

[XMH06] Sen Xu, Manton Matthews, and Chin-Tser Huang. Security issues in privacy and key management protocols of ieee 802.16. In *Proceedings of the 44th annual Southeast regional conference*, pages 113–118. ACM, 2006.

[Yao82] Andrew C Yao. Protocols for secure computations. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, pages 160–164. IEEE, 1982.

[YCW+16] Yanjiang Yang, Haibin Cai, Zhuo Wei, Haibing Lu, and Kim-Kwang Raymond Choo. Towards lightweight anonymous entity authentication for IoT applications. In *Australasian Conference on Information Security and Privacy*, pages 265–280. Springer, 2016.

[YY05] Adam Young and Moti Yung. Malicious cryptography: Kleptographic aspects. In *Cryptographersâ Track at the RSA Conference*, pages 7–18. Springer, 2005.

[ZG13] Kai Zhao and Lina Ge. A survey on the internet of things security. In *Computational Intelligence and Security (CIS), 2013 9th International Conference on*, pages 663–667. IEEE, 2013.

[ZWC+10] Qian Zhu, Ruicong Wang, Qi Chen, Yan Liu, and Weijun Qin. Iot gateway: Bridgingwireless sensor networks into internet of things. In *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on*, pages 347–352. Ieee, 2010.