REPUBLIC OF TURKEY
AKDENIZ UNIVERSITY

A CONTRIBUTORY STUDY ON ACCESS CONTROL AND AUTHENTICATION MECHANISMS FOR INTERNET OF THINGS

Manolya ATALAY

INSTITUTE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

June 2019
ANTALYA

REPUBLIC OF TURKEY
AKDENIZ UNIVERSITY

A CONTRIBUTORY STUDY ON ACCESS CONTROL AND AUTHENTICATION MECHANISMS FOR INTERNET OF THINGS

Manolya ATALAY

INSTITUTE OF NATURAL SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

June 2019
ANTALYA

REPUBLIC OF TURKEY
AKDENIZ UNIVERSITY
INSTITUTE OF NATURAL SCIENCES

A CONTRIBUTORY STUDY ON ACCESS CONTROL AND AUTHENTICATION MECHANISMS FOR INTERNET OF THINGS

Manolya ATALAY

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

June 2019

REPUBLIC OF TURKEY

AKDENIZ UNIVERSITY

INSTITUTE OF NATURAL SCIENCES

A CONTRIBUTORY STUDY ON ACCESS CONTROL AND AUTHENTICATION MECHANISMS FOR INTERNET OF THINGS

Manolya ATALAY

DEPARTMENT OF COMPUTER ENGINEERING

MASTER THESIS

This thesis was unanimously approved by the jury on 27/06/2019.

Asst. Prof. Dr. Murat AK

Prof. Dr. Melih GÜNAY

Assoc. Prof. Dr. Cafer ÇALIŞKAN

ÖZET

A CONTRIBUTORY STUDY ON ACCESS CONTROL AND AUTHENTICATION MECHANISMS FOR INTERNET OF THINGS

Manolya ATALAY

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı Danışman: Dr. Öğr. Üyesi Murat AK

Haziran 2019; 139 sayfa

Nesnelerin İnterneti son on yılda çok popüler bir platform olup hayatımızda hızla yerini almaya başlamıştır. Bu platformu, belli bir hedefe yönelik çalışan birçok algılayabilir ağ düğümünün ortak çalışarak birleşik bir haberleşme ortamı olarak tanımlayabiliriz. Bu boğumlar, yani nesneler, farklı kapsamlarda güce ve yeteneğe sahip olabilirler. En yaygın kullanılan algılayabilir ağ boğumlarının birçoğu RFID etiketleriyle ilişkilendirilmiştir. Bu platformun asıl amacı modern yaşantının farklı alanlarında veri işleme verimini arttırmaktır. Nesnelerin interneti hergün gelişmektedir. Gelişmesinin kaynağı, kablosuz algılama haberleşmelerindeki gelişen özellik ve kalite, evrimleşen ağ standartları, insanların elektronik cihazlarla etkileşiminin giderek artması ve daha bir çok gelişmedir. Bu durum aynı zamanda Nesnelerin İnterneti kapsamına daha fazla ihtiyaç getirmektedir.

Paylaşımın yoğun olduğu ortamlardaki en büyük problemler hassas bilginin güvenliği, servis güvenirliliği ve sahteciliğin engellenmesidir. Bu durum, kimlik doğrulama ve giriş kontrol mekanizmalarının gerekliliğini ortaya çıkarmaktadır. Kullanıcı sayısı arttıkça daha zengin içerikli metodlar gerekmetedir. Ancak, Nesnelerin İnterneti mimarisi kısıtlı kaynağa sahip heterojen bir ortamdır. Bunun nedeni kısıtlı güç kaynağı ve depolama alanlarıdır. Aynı zamanda farklı uygulama alanlarının varlığı kısıtlamaların artışını tetiklemektedir.

Bu çalışmanın odağı Nesnelerin İnterneti iskeletlerinde kimlik doğrulama ve giriş kontrolü mekanizmalarıdır. Bu mekanizmalar farklı uygulama alanlarına yönelik, farklı rollerin verildiği farklı seviyelerde giriş ayrıcalıklarını kapsar. Burada, hali hazırda var olan kablosuz algılama ağlarında kimlik doğrulama mekanizmaları, sürekli veri akışı sistemlerinde giriş kontrol mekanizmaları, fiziksel olarak klonlanamayan fonksiyonlar, hafif siklet kriptografi yöntemleri, Nesnelerin İnterneti için yüksek seviyeli melez çözümler ve saldırı tespit sistemlerini özetle açıkladık. Daha sonra burada açıklanan çalışmaların bir çoğunun analizini gösterdik. Çalışmanın devamında ise bu analizlere dayanarak bir Nesnelerin İnterneti iskeleti sunduk. Son olarak, uygulama için gerekli ayrıntılar ve geliştirmek için ek teknikler önerdik.

ANAHTAR KELİMELER: 6LoWPAN, Düşük Güçlü ve Kayıplı Ağlarda IPv6 YÖnlendirmesi, Fiziksel Olarak Klonlanamayan Fonksiyonlar, Giriş Kontrol Mekanismaları, Hafif Siklet Kriptografi, Hizmet Reddi Saldırıları, IPv6, Kablosuz Algılama Ağları, Kimlik

Doğrulama Mekanizmaları, Nesnelerin İnterneti, RPL, Tek Yönlü Karma Fonksiyonlar, Veri Akışı Yönetim Sistemleri.

JÜRİ: Dr. Öğr. Üyesi Murat AK
Prof. Dr. Melih GÜNAY
Doç. Dr. Cafer ÇALIŞKAN

ABSTRACT

A CONTRIBUTORY STUDY ON ACCESS CONTROL AND AUTHENTICATION MECHANISMS FOR INTERNET OF THINGS

Manolya ATALAY

MSc Thesis in Computer Engineering

Supervisor: Asst. Prof. Dr. Murat AK June 2019; 139 pages

The Internet of Things (IoT) is a very popular platform that found its way into our lives very rapidly in the last decade. We can define this platform as an architecture consisting of many sensing nodes communicating with each other to collaborate given a certain common goal. These nodes, things, can have a wide range of power and capabilities. The most common sensing nodes are empowered with RFID tags. The main purpose of this platform is to increase productivity at many levels in our modern lives. The Internet of Things is getting improved every day. The power of its improvement comes from the growing quality and features of wireless sensor communications, evolving network standards, increscent of the human interaction with electronic devices, and many other developments which bring more requirements to the Internet of Things.

The biggest concerns arising from the shared platforms is the privacy of sensitive information, the confidentiality of service, prevention of repudiation. This brings the necessity of authorization and access control mechanisms. Since the number of users can increase, more sophisticated methods are required. However, IoT architecture is a resource-constrained heterogeneous environment due to limited computational power and storage. Also, specific application contexts can impose more constraints.

The focus of this work is authentication and access control mechanisms of the Internet of Things frameworks. These mechanisms encompass different levels of privileges of access to those assigned with different roles for specific application contexts. We provide brief descriptions on the existing solutions such as wireless sensor network authentication mechanisms, access control mechanisms for data stream management systems, physically unclonable functions, lightweight cryptographic schemes, high-level hybrid solutions for IoT, and intrusion detection systems. Later, we provide proper analysis for many of the described mechanisms. We further propose a framework based on our analysis. Finally, we propose implementation details and additional techniques for further improvement.

KEYWORDS: 6LoWPAN, Access Control Mechanisms, Authentication, Data Stream Management Systems, Denial of Service Attacks, Internet of Things, IPv6, IPv6 Routing Protocol for Low-power and Lossy-networks, Lightweight Cryptography, One-way hash functions, Physically Unclonable Functions, RPL, Wireless Sensor Networks.

COMMITTEE: Asst. Prof. Dr. Murat AK
Prof. Dr. Melih GÜNAY
Assoc. Prof. Dr. Cafer ÇALIŞKAN

ACKNOWLEDGEMENTS

Ever since I was little I always had a great passion for anything related to computers. Along with this passion and the great efforts and support of my parents I was able to enter the Computer Science world. Although several times I had an occasional loss of motivation, the constant reminders from my family, friends, and professors pushed me to graduate with a degree in 2012.

My first motivation was the portrayal of cybersecurity people on popular media. The common misconceptions made me think the security was all about writing good codes and algorithms. So I constantly polished my skills in that area. However, along the way I noticed, in reality, there was much more and I could not find my way towards it despite my attempts. I saw the big picture when I first took the Cryptography course in my final year. I thought I was very late for that topic. Because of my existing skill-set in software development, I was always moved to the software industry. I have worked in many areas since I enjoyed it very much. However, nothing interested me as the security technology.

After graduating, until the last quarter of 2015, I worked in different companies, different countries, different jobs, and expanded my vision with meeting a great many people from all over the world. Those people I appreciated were always headstrong and knew clearly what they wanted to do with their lives. This way I decided to follow network security and started my researching career.

The last three years of my life was filled with continuous learning, researching, listening and watching other people, getting support from my professors, and expanding my understanding. I came back to university to become more humble and focus on science. I received more than I expected.

Because of this, I would like to thank my loving and patient parents for welcoming me back home and supporting me after studying and working all these years. I am grateful to my advisor professor, Murat AK, who supported me whenever I needed help and integrated me into this world. I'm also grateful to our chairman Melih Günay, who always supported me, helped me clearing my mind and giving me great advices. Our entire department is full of amazing professors who are truly passionate and gentle. I would like to thank my colleagues who always been great support and fun to work with. Our department not once made me feel like I am working. I would love to contribute to this serenity as long as I am present.

Finally, I would like to thank all of my close friends for their continuous emotional support and making my three years full of excitement. I hope to move forward even further with new research topics. My dream is to be a significant individual in network security society and contribute to its growth.

LIST OF CONTENTS

TEXT OF OATH

Yüksek Lisans tezi olarak sunduğum “A Contributory Study on Access Control and Authentication Mechanisms for Internet of Things” adlı bu çalışmanın, akademik kurallar ve etik değerlere uygun olarak bulunduğunu belirtir, bu tez çalışmasında bana ait olmayan tüm bilgilerin kaynağını gösterdiğimi beyan ederim.

27/06/2019
Manolya ATALAY
Signature

LIST OF ABBREVIATIONS

6LoWPAN	: IPv6 Low-Power Wireless Personal Area Networks
ACL	: Access Control List
CBC	: Cipher Block Chaining
CRP	: Challenge-Response Pair
DoS	: Denial of Service
DSMS	: Data Stream Management System
FPGA	: Field Programmable Gate Array
FSM	: Finite State Machine
IC	: Integrated Circuits
IDS	: Intrusion Detection Systems
IoT	: Internet of Things
IP	: Internet Protocol
IP Service	: Intellectual Property Service
IV	: Initial Vector
LFSR	: Linear Feedback Shift Register
MAC	: Medium Access Layer
MAC scheme	: Message Authentication Code Scheme
NFSR	: Non-linear Feedback Shift Register
PHY	: Physical Layer
PRNG	: Pseudo Random Number Generators
PUF	: Physically Unclonable Functions
RBAC	: Role-Based Access Control
RO	: Ring Oscillator
RPL	: IPv6 Routing Protocol for Low-power and Lossy-networks
TMD	: Time-Memory-Data Trade-off
TRNG	: True Random Number Generators
UID	: Ubiquitous Identifier
WSN	: Wireless Sensor Network
XOR	: Exclusive OR

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Kevin Ashton, the co-founder of Auto-ID Center of MIT coined the "Internet of Things" as a phenomenon in 1999, showing the potential of RFID technology when its integrated into our lives (Missbach et al. 2015). However, Internet of Things started off as a theoretical concept of an automated system where objects have location and status data. The objective was to improve quality of life through a real-time dynamic communication between users and objects.

The Internet of Things is not just a technological product or a scientific field to solve a certain set of problems. IoT is a collection of solutions that integrate the existing technologies efficiently. On the ground of this aim, the Internet creates the connectivity and productivity of not just humans but also the objects.

The Internet of Things is a technological revolution that provides a unified framework with a growing network of physical sensing and actuating devices. The physical devices have the ability to collect and share data over wired, wireless communication channels. Self-configuration on these sensing objects is a service that is in high demand.

One of the major goals of the Internet of Things is wide-scale connectivity. Development of wireless sensor networks, nanotechnology, and growing techniques and popularity of artificial intelligence increases the quality of IoT solutions.

There are mainly three issues of the Internet of Things. Firstly, the variety of devices in the framework highly relevant to the application context. These should be able to communicate and share information among themselves. This requires full interoperability.

Secondly, most of the devices should be autonomous and self-configuring even if the nature of the given architecture is fully centralized. We can achieve this with a random level of context-awareness and ubiquity. Lastly, the reliability of the entire framework is very crucial by means of the trust, privacy, authenticity, and overall security of devices and communication channels.

There are three distinct visions of the Internet of Things according to Miorandi et al.(2012). These visions are Internet-vision, Things-vision, and Semantic-vision. Semantic-oriented vision has arisen from messaging protocols among the connected objects. It's considered that the number of these objects will become larger in the future even if their architecture does not have many initially. Therefore, this vision focuses on message passing, data generating, data traffic, and analysis.

Things-oriented vision concern with object visibility, ubiquitous identifier (UID) architecture. It also offers context-awareness and collaborative communications. RFID-centric designs bring solutions to event transfer and network connectivity.

Internet-oriented vision deals with full connectivity over existing network standards. Web-of-Things apply web standards over the architecture. This vision intends to connect

everything from anywhere to everywhere, at any given time. The simplification of the current Internet addressing is aimed to work with various objects.

The visions mentioned above are like pivots for the Internet of Things framework design. They allow the developers to create an application-specific comprehensive backbone and provide specifications for user requirements.

The Internet of Things architecture has additional visions such as; system-level and service-level point-of-view. System-level point-of-view targets the system scalability and dynamicity as well as integration between the physical realm and processable data through system semantics. However, in service-level point-of-view, the focus is mainly on the functionalities and resources of objects individually. Their ubiquitous data exchange, energy-optimization, localization, self-configuration, self-organization, interoperability, etc. mechanisms are issues part of the service-level point-of-view.

Intel (Nath and Venkatesan 2013) envisions that in 2020 there will be 50 billion devices connected to the internet. These devices can facilitate services such as wireless audio streams, wireless charging units, 4K video to TV stream, intelligent thermostat control, weather detection units etc. The demands for better quality, smart maintenance, remote systems, comprehensive security, precise data analytics.

Since The Internet of Things is introduced, it fundamentally brings many security threats. Hence, the fact that heterogeneous devices characterizing the IoT and tasked with information handling, exchanging, and processing according to application-specific tasks; we are obliged to analyze the vulnerabilities at every level and dimension of the architecture.

According to Xu et al. (2013), the lowest security at any part of the system should characterize the overall security. When security issues are viewed, one should not only consider humans but also objects and objects those mimicking humans as adversaries. To be able to provide a valid security, privacy, and trust mechanism in this heterogeneous environment, we should consider the limitations such as computation. We also should not forget complex cryptographic techniques are unreliable in this context.

An Italian clothing brand, Benetton, brought a plan to tag a line of clothes (Miorandi et al. 2012). The company pursued the RFID innovation in their clothing line in order to provide a better service. They planned 15 million tags to be issued for the clothes of the season. Nevertheless, the idea caused mistrust among the customers who were worried about the disclosure of their personal information. Naturally, the customers did not wish to risk exploitation of their personal details.

In the work of Juels (2005), it is stated that tracking and clandestine inventorying have been a well-known security issue for a long time. RFID readers can read information without any authentication mechanism. This allowed adversarial readers to harvest data from honest tags without any security barrier. Also, the tag serial numbers included sensitive information such as personal information and RFID Supplier Chain data.

We should revisit several properties of the Internet of Things in order to create a reliable security backbone. According to Sicari et al. (2015), these properties are security, data anonymity, confidentiality, authentication and access control mechanisms, data protection, self-heal, flexibility, constant data transfer and sharing, and measurements against non-repudiation.

Perseverance of information during data communication between the nodes - which are either or both of humans and objects - is one of the main security expectations of the Internet of Things.

Pfitzmann and Hansen (2009), define the data minimization mechanism through its components such as anonymity, unlinkability, unobservability, and pseudonymity. The concentration of this topic is minimizing the possibility of clandestine inventorying. The duration of the data storage must be minimal according to this work. Instinctively, many of the users would not like to reveal their information without their consent. Therefore, the data minimization should be part of the security mechanisms while designing the IoT architectures.

CIA triad - confidentiality, integrity, and availability is a basis for many information security measures for a long time. Confidentiality ensures that only authorized users can access the sensitive information. Consequently, this is a crucial property even though a perpetual availability of all nodes in a sensor network is infeasible.

Waschke (2017) declared that authentication in terms of information security is a mechanism to prove the consistency between the action and the claim of a user or entity in a given system. In the Internet of Things architectures, there must be such a mechanism that is well-designed enough so malicious attackers will not be able to enter the system unnoticed to collect information or compromise it.

Access control mechanisms define the authentication levels of users and allow them to perform operations only within limits given by the system. In the Internet of Things, this does not only provide information security but gives an opportunity to optimize the workload on the system. Users with different authentication levels can have different roles. These mechanisms can affect the quality of the overall system. We will mainly focus on access control and authentication mechanisms in this work.

It is important for the system to adjust to the security parameters and requirements of new node added to the framework. The new node could be irrelevant to existing objects in the system as well as could be a newer version of active nodes. Different entities bring the variety of security protocols. It is aforementioned, the most vulnerable security measurements define the overall security level of the entire framework. The Internet of Things infrastructure requires flexible and scalable security mechanisms. There are many ad-hoc solutions proposed as countermeasures to adaptability issues.

Generally, dynamicity is important at every level and dimension of the Internet of Things infrastructures. It is crucial when the nodes were first introduced to a framework,

or when it should be removed. Additionally, it is important to keep communication streams running and make sure communication channels are not jeopardized in the light of dynamicity property. There are many mechanisms to keep the stability of data stream and transfer. The solutions originate from stream databases, online video and audio streaming, early wired and wireless sensor networks, distributed networks and so on. Nonetheless, the Internet of Things frameworks have the most constraints although the information flow and workload may not be as bulky as previously mentioned systems. Many of the existing solutions are not complying with the requirements and limitations of IoT. Whilst these drawbacks, some of the existing solutions can be applying to certain applications. Also, some mechanisms can be optimized for better performance or they may inspire newer solutions.

Finally, there is the issue of repudiation. Repudiation is when at least one of the communicating nodes to deny sending or receiving messages (Pfitzmann and Hansen 2009). In dynamic systems non-repudiation protocols are necessary. These protocols provide evidence of communications between two parties to the system. This way none of the communicating parties will be able to deny sending or receiving messages later.

Regarding the survey of Sicari et al. (2015), the security issues can be separated into eight main categories; Authentication, access control, privacy, confidentiality, secure middleware, mobile security, policy enforcement, and trust.

As declared above this work will focus mainly on authentication and access control mechanisms. To this day, diverse mechanisms developed, adapted, and implemented. These mechanisms varied from hardware implementations to very abstract solutions. We will give an insight to them by categorizing them into eight; existing solutions for wireless and distributed sensor networks, existing solutions for data stream systems, hardware solutions, software solutions, lightweight cryptography, capability based mechanisms evolved from role-based access control and attribute-based access control models, intrusion detection systems, and higher level solutions.

This work will extend the topic with analysis of given mechanisms. Then, we will introduce a multilayered security framework based on intrusion detection systems and anomaly-based access control models. Finally, an insight on a feasible extention will be provided.

2. LITERATURE REVIEW

2.1. Wireless Sensor Networks

The Internet of Things phenomena arisen from the development of wireless sensor networks. Therefore the early authentication mechanisms can be used in this environment. The nodes in traditional wireless sensor networks have similar features such as low computational power, limited bandwidth, close-knit organization, distributed but coordinated behavior. These features allow us to integrate similar technologies into the IoT. Nonetheless, there are more constraints regarding IEEE 802.15.4 standards. Inclusive of threats by various malicious attacks, the architecture of existing authentication mechanisms of wireless sensor networks are motivated by data and surrounding event sensing, computational and energy optimization, and overhead reduction (Lim and Korkishko 2005).

2.1.1. ZigBee Authentication

ZigBee introduces a backbone for Authentication and Access Control including mechanisms such as management of devices, generating and distributing secure keys, and frame protection. Their assumption is based on that each layer in the protocol stack of a single device secures its own frames and trust other layers in the same stack. Hence, the security among the layers in the stacks of the same device is not cryptographic. However, there are cryptographic security measures provided among different devices. Moreover, pseudo-random number generators and tamper-resistant hardware are available in the architecture.

The main architecture is separated into a security service provider, application framework, *ZigBee* device object(ZDO), and layers. The layers are application support sublayer(APS), network layer (NWK), Medium Access (MAC) layer, and physical (PHY) layer in the given order. Security service provider and ZDO are in direct communication with APS and NWK layers. On the other hand, the application framework communicates with ZDO while its application object is directly connected to the APS sublayer. The architecture is depicted in Figure 2.1.

ZigBee provides a Trust Center to manage devices, generation of the secure keys and their distribution. Trust center has three roles as first of them is trust manager role. Trust manager role deals with authentication of devices those attempt to join the network. The second role is network manager that is dedicated to generation, maintenance, and distribution of keys. Lastly, the configuration manager role where end-to-end security is provided among different devices.

There are two types of networks in *ZigBee*. These are centralized and distributed networks. In centralized networks, only Trust Centers can launch the network. Nodes receive network keys from Trust Centers in order to establish their own link keys to start communication with other nodes. On the other hand in distributed networks of *ZigBee*, there are no Trust Centers. Routers can initialize the network by themselves and nodes receive their network keys to communicate with them.

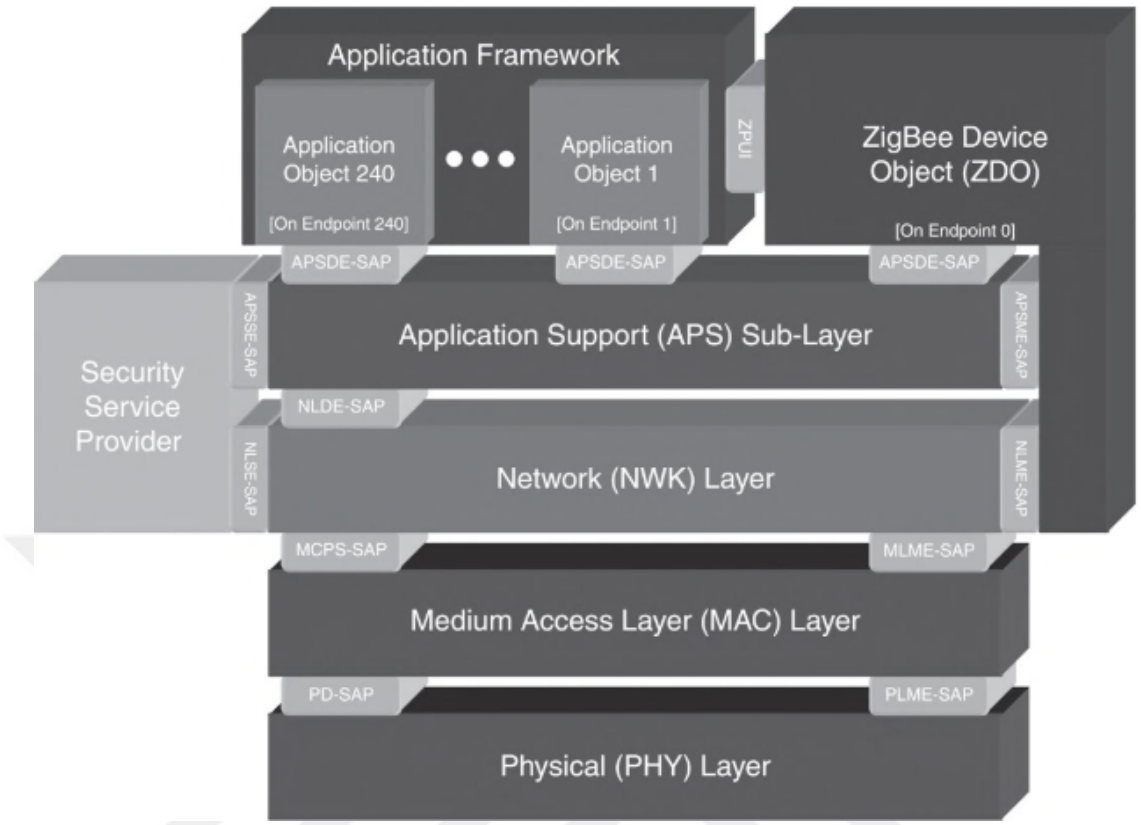


Figure 2.1. The ZigBee architecture (Gislason 2008)

Despite the fact that distributed networks of *ZigBee* have lower security, they are simpler than centralized structure. Routers manage the new devices and issue a single network key. All the nodes communicate with encrypted messages using this network key. Before the nodes attempt to join the network, they already come with installed link keys.

The link key is used in peer-to-peer communication. It is applied by APS of the *ZigBee* stack while network keys are applied by APL. In addition to key-transport and pre-installation, link keys can also be obtained through key-establishment. Link keys used by trust centers usually come pre-configured. There are two types of link keys; global and unique. They decide how the device will interpret certain messages coming from trust center.

Finally, the third key used in *ZigBee* is master key. The master key is responsible for long-term security between devices. It keeps the link key exchange secure through Symmetric-Key Key Exchange protocol (SKKE). Apart from key transport and early installation, the master key can be obtained from user-entered data such as PIN or pass-code.

Keys are managed through either pre-installation, or key establishment, or key transport. Manufacturers initially install a set of keys and jumpers to select those keys for users to select one. However, to update and establish link-keys from Trust Centers, SKKE pro-

tocol is used. A link-key is generated based on the master key using one-way functions. In certain situations, key-transport is required. Here the network device makes a request to Trust Center to generate a new key. In centralized networks in *ZigBee* keys are distributed through Certificate-Based Key Establishment (CBKE).

In IEEE 802.15.4, AES-128 is used to secure the communication from other networks. The data payload is encrypted with this encryption. Integrity is provided through Message Integrity Code (MIC) or Message Authentication Code (MAC). These codes are appended to the encrypted message. According to IEEE 802.15.4, in order to use the Auxiliary Security Header, Security Enabled Subfield from Frame Control Field should be active. This header consists of three fields; security control, frame counter, and key identifier.

In Security Control Header global security parameters such as key length and the specification of the part of data to be encrypted are determined. These parameters describe the security level. Consequently, this header defines the security of the network. The second field, Frame Counter is a security measure against replay attacks. During communication, the source of frame issues this field. Boyle and Newe (2007) used this mechanism in symmetric encryption in AES-CTR mode. The last field, Key Identifier specifies the type of key for communication at the time. The information required for these fields are all stored in Access Control List (ACL) which is located in MAC PAN Information Base (PIB).

ZigBee uses AES-CCM* for symmetric encryption and authentication. AES-CCM is a CBC-MAC mode of operation with the addition of a counter. AES-CCM* is a modification of AES-CCM with only the difference of support for encryption required only messages (IEEE Standard for Local and metropolitan area networks 2011). It generates a Message Integration Code (MIC). Receivers authenticate the frame by comparing the MIC they have with the MIC they decrypted from the frame with AES-CCM*. This mechanism allows detection of changes in data, while those changes might originate from an attacker or an error in the system.

In *ZigBee*, device authentication has two modes: Residential and commercial. These modes deal with distribution, storage, and encryption of keys by Nabeel et al. (2012). In the residential mode, the master and network keys are maintained by either Trust Center or the devices themselves. Here new devices do not have network keys, therefore Trust Center sends them over an unprotected link. That creates a security breach within a short time period. In the beginning, the Trust Center address is not known to the device. After receiving the network key, the device sets the source address as Trust Center address.

Centralized control is more distinguished in commercial mode since Trust Center manages every key used in the network and never uses an unreliable channel to distribute them. When a new device requests to join the network, Trust Center sends master key. They start a communication with SKKE protocol using the master key. Here device attempts to generate a link-key and wait for the approval of Trust Center within a given time period. If the time period expires or the link-key is rejected, the new device either has to leave the network or restart the entire session. If the link-key is approved, Trust Center sends the network key.

ZigBee introduced touch-link protocol with version 3.0. It is a device-unique authentication with ease of implementation. Nevertheless, Morgner et al. (2017) states that several active and passive threat models are tested. The work aimed to make the system unavailable and controllable. They succeeded and recommended to upgrade the "commissioning" mechanism.

Over-the-air (OTA) firmware upgrade mechanism allows manufacturers to provide updates and patches for current systems. However this imposes vulnerability without proper security measures. *ZigBee* provides logical link-based encryption with virtual private links and AES-128 encryption.

ZigBee has many mechanisms to prove security over interference such as Signal-spreading method DSSS, Dynamic RF output power control, mesh networking, location-aware routing through path diversity, frequency channel selection, frequency agility capability, and adaptive packet length selection. *ZigBee* provides considerably many security mechanisms for different application contexts and the servicecs are easy to configure.

2.1.2. SPINS

The widespread use of RFID technologies with sensor networks led to many security concerns due to its existential properties. These properties are limitations of energy, memory, and computational power. The communication links were as slow as 10Kbps and TinyOS took almost half of memory remaining 4500 Bytes for security. The most energy consumption originates from message exchange, therefore keeping the overhead at minimum is crucial. Moreover, limited energy requires frequent updates on keys. These constraints left security neglected while the number of applications of sensor networks is increased. However, the use of sensor networks is spread over more information sensitive areas such as emergency, military, health, energy, and data inventory applications. The data freshness and authentication triad; confidentiality, integrity, and availability were immediately required.

Perrig et al. (2002) designed a two-party communication protocol by combining two distinct building blocks called SPINS. These two building blocks are called $SNEP$ and $\mu TESLA$. In addition to aforementioned security and design concerns, broadcast authentication was a challenge as existing solutions were impractical in these constrained environments. Finally, asymmetric cryptography is not suitable as well since they operate computationally intensive and result in a large data overhead during communication. Asymmetric cryptographic operations shorten the battery life.

The architecture of SPINS mainly consists of sensor nodes and base stations. Base stations have better computational power, larger memory, and a longer battery life compared to sensor nodes. The communications in the sensor networks are RF based, hence broadcast mechanisms are fundamentally required. The nodes in these networks form a routing forest connected to a base station as their root, while base stations interface their forests with external networks.

Beacon transmission initiates the formation of the routing forests. The communica-

tion patterns are separated into three. The first pattern is sensor information messages from sensor nodes to the base station. Secondly, there are requests for specific tasks from base stations to a certain node or a group of nodes sent as messages. And lastly, the broadcasting messages from base station to all of the nodes connected to it. These messages can be beacon transmission, queries or reforming of the network.

The key management of *SPINS* protocol assumes that the network may be located in an unreliable environment. Additionally, the integrity of each node is unrealistic considering the constraints. Since generalization is impossible in sensor networks, the protocol assumes some of the nodes are compromised and can acquire some keys. The aim of the proposed protocol is that the compromised nodes will not affect the overall security of the network. There is also the fact that the wireless communications are prone to eavesdrop and injection.

SPINS provides mechanisms for parties to recognize each other as well as bootstrapping to re-initiate the counters and keys when necessary to secure the channel during communication sessions. Point-to-point communications are secured with symmetric cryptographic applications, while broadcast communications use public key schemes to prevent impersonation attacks. During data authentication, integrity is also provided.

Data freshness is important in wireless communications, so the system knows the messages exchanged are recent consequently not replayed or resent after modification. *SPINS* provide weak and strong freshness mechanisms in building blocks. $SNEP$ provides weak freshness by partially ordering the messages exchanged, while $\mu TESLA$ provides strong freshness that not only ordering all the messages but also estimates the delays. Weak freshness is enough for sensor nodes when sending sensor readings, however, counter sharing requires more sensitive properties. Hence, strong freshness is implemented in $\mu TESLA$.

The two protocols in the system provide distinct functions in the system. Secure Network Encryption Protocol ($SNEP$) defines the authenticity of two-party communication while Time-Efficient Loss-Tolerant Authentication ($\mu TESLA$) authenticates the broadcast communications. $SNEP$ uses DES algorithm to encrypt the messages with cipher block chaining (CBC) mode of operation. Every node shares the same counter to generate message authentication codes (MAC). By synchronizing the counters from time to time and allow parties to generate their own MACs to authenticate the message they received we avoid energy draining from large data transfer. The fundamental feature of using CBC code with CTR is to achieve semantic security. Semantic security is based on the presence of eavesdroppers in the communication channel that they should not distinguish two identical plaintexts from the ciphertext. The counter in CBC mode of operation can provide the semantic security with negligible distinguishability. The replay attackers should be able to mimic the counter to be able to generate MAC.

$SNEP$ can provide weak freshness using its counter mode by simply ordering the messages exchanged on the network. Also by sharing the counter, there is less overhead and only 8 Bytes of addition to the message. This is satisfactory for sensor nodes to send their information to base stations. On the other hand, $\mu TESLA$ provides more sophistica-

ted solutions for more sensitive information such as counter synchronization, topological updates, and other specific messages sent to nodes by the base station.

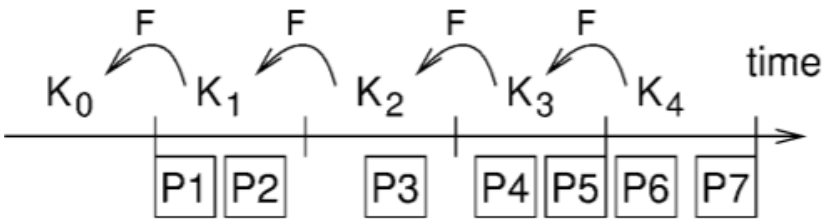


Figure 2.2. The $\mu TESLA$ one-key chain (Perrig et al. 2002)

$\mu TESLA$ is a purely symmetric mechanism derived from its hybrid version, TESLA proposed by Perrig et al. (2000). TESLA is not efficient enough regarding the overhead in sensor networks despite its performance in multicasting lossy channels. In $\mu TESLA$, initial packets are authenticated through delayed disclosure of secret keys. This provides the necessary asymmetry and more efficient broadcast authentication. The Figure 2.1. shows the one-key chain mechanism where the time processed left-to-right and the key chain is applied right-to-left. The admission of the key is associated with time intervals. The key is broadcasted in a separate message by the sender.

There is a loose time synchronization between the base station and nodes. Each node in the communication knows the threshold for maximum synchronization error. The time is split into epochs where each epoch is paired with a key from key-chain used in message authentication code mechanism. The key-chain is generated from a one-way function while each key is generated in reverse order.Receivers buffer the packets until key chain disclosure is completed and packet contents are authenticated. The key disclosure takes place after a reasonable round-trip delay to prevent forgery attacks. Additionally the receivers are required to know the disclosure schedule. This also is necessary against forgery attacker with knowledge of a disclosed key and epoch pair.

Broadcasting the disclosed keys to every node will drain the batter life. *SPINS* provides two approaches where first is for the node to use $SNEP$ to broadcast through base station. The second approach is node to broadcast data while base station keeps the key-chain.

In *SPINS* mechanism, the costs are directly related with communication load. There is no additional cost. End-to-end authentication is provided with strong freshness, confidentiality and authentication. Nodes in system have watchdog behavior for anomaly detection. The most of the work load is on base station with less constraints and better computational power. This mechanism is convenient for ad-hoc networks. Design is simple, extensible and effective.

SPINS has several drawbacks such as reliance of central and more less constrained

devices which limits the mobility and the security assumption is based on time synchronization. Time synchronization can be mitigated with a counter but requires overhead. There is also the consequences of clock drift. Many security issues are not covered such as non-repudiation and hardware attacks.

2.1.3. TinySec

Previously in SPINS (Perrig et al. 2002) provided very solid design choices considering the limitations and capabilities of sensor networks. Although it was not fully implemented, it exposed that the security at link-layer provides more sophisticated and higher level security mechanisms. Karlof et al. (2004) proposed the *TinySec* which is based a two-tier design on an efficient broadcast mechanism in resource-constrained environments. It also aimed to prove that purely software-based security is sufficient with good design. This is the first fully implemented protocol for link-layer cryptography for sensor networks at the time. The authors of *TinySec* measured the mechanism in terms of the bandwidth, latency, and power consumption. It is meant to interoperate with higher level protocols.

TinySec is based on TinyOS (Hill et al. 2000) packets. TinyOS is a micro-threaded operating system that can fit into 178 bytes of memory that supports two-level scheduling, concurrency-intensive operations, and different hardware platforms. However, *TinySec* removes the CRC and group field in its packets replacing it with 4 bytes of message authentication code, CBC-MAC.

Aforementioned works listed out many issues in the sensor networks. Additionally, *TinySec* pointed out that sensor networks are different from other distributed networks. Hence, they should be regarded more delicately in light of bandwidth and energy consumption since they are vulnerable to resource consumption attacks where adversaries can send packets to drain batteries of sensor nodes and waste the bandwidth. Along with the CIA triad and resource consumption attacks, one should not forget wireless sensor networks are broadcast environments. This allows adversaries to eavesdrop on communication channels, inject and intercept the packets. There are also costly and compelling radio transceivers and workstations that can interact with the network from a distance. Aside from the fact these attacks are considered in this work, the most of them are not addressed in terms of security measures. Authenticity, confidentiality, and integrity are the main issues concerned.

Many of the high-level internet authentication systems have end-to-end security mechanisms to preserve confidentiality and integrity. Despite their convenience, they bring vulnerability for denial of service and inject attacks. Therefore, the many-to-one nature of the sensor networks posed in this work communicates through the multihop topology. There are many repeated patterns in sensor networks, and many nodes send similar packets. In order to prevent waste of energy and bandwidth, the duplicate packets are aggregated and removed. Regarding these facts, *TinySec* builds a link-layer security.

TinySec sets three main security goals: Access control and message integrity, confi-

dentiality, and replay protection. The first goal requires authentication mechanism with each packet sent on the communication channel. This will allow the detection of unauthorized senders. Furthermore, integrity can be preserved through message authentication code (MAC). Secondly, confidentiality is achieved through sound semantic security as in SPINS architecture. The proposed scheme should prevent adversaries to extract information from any part of packets. And lastly, the replay protection can be provided with counters. However, conventional counter mechanisms require a certain amount of information and table for counters on receiver devices. Thus, sensor nodes have mainly RAM and it is impossible to keep the required information on link-layer.

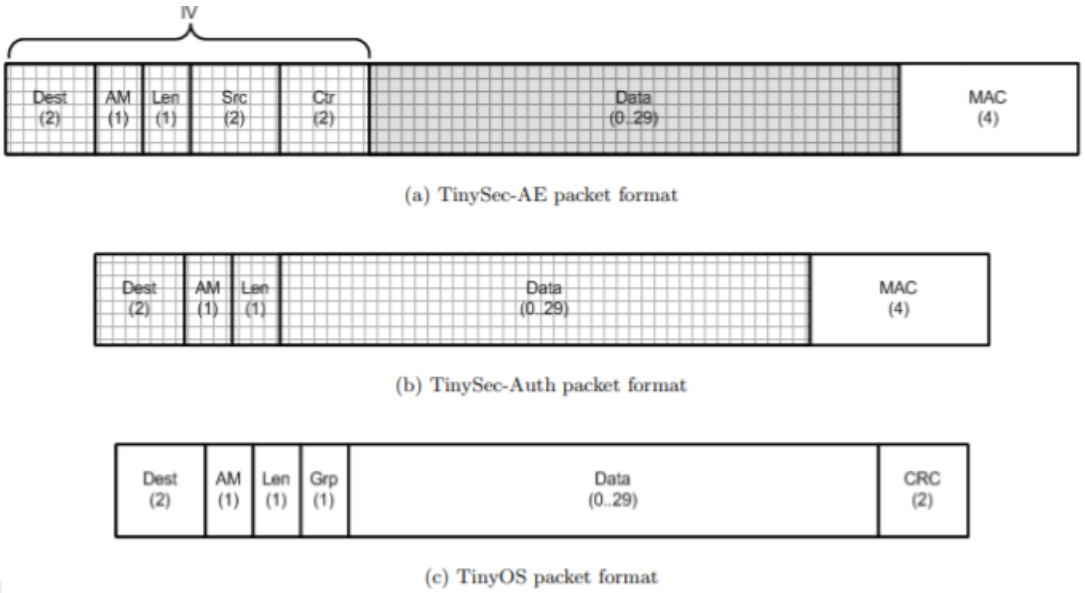
TinySec design is based on the burglar alarm model. As indicated earlier, it is critical to authenticate the packet sources. However, encryption is not always necessary. Given the security assumption, the unauthorized source cannot trigger false alarms. This prevents the cut-and-paste attacks where the adversary can replace another meaningful encrypted text with original one to be decrypted at the destination.

Unnecessary encryption increases the latency, computation, and power consumption. Hence, there are two different packet formats in *TinySec* where one is *TinySec-Auth* and the other is *TinySec-AE* (Figure 2.3). *TinySec-Auth* has only authentication properties while *TinySec-AE* has both encryption and authentication properties. These packet formats slightly different than original TinyOS packets. They have common fields such as destination address, active message, and length. Group field and CRC is replaced with 4 bytes of MAC, which provide sufficient security for the sensor node lifetime. Since the group field is a weak access control mechanism that is not intended for environments with the presence of adversaries, it is rather reliable to use MAC. However, *TinySec* increases the packet size by 1 byte. Also, *TinySec-AE* differentiates from *TinySec-Auth* with the source address and counter value. The encryption scheme used in *TinySec-AE* is Skipjack with CBC mode of operation. Although RC5 performed better results with precomputed key scheduling, it is patented. *TinySec* includes ciphertext stealing in its encryption scheme that preserves the ciphertext length same as plaintext.

Denial of service threat is also present on authentication mechanisms. Nonetheless, at 19kbps communication rate successful brute force attack will take longer than the battery life while occupying the channel for so long is almost impossible. Thus, no strong measures have taken against such attacks.

Link-layer security requires effective keying mechanisms. Per-link keys between neighboring nodes and group keys have both shown effective with several drawbacks while former limits passive participation and local broadcast, latter reduce the robustness.

TinySec has shown good performance in implementation. Although it negatively affects the performance of TinyOS Stack by latency, overhead, bandwidth, power consumption, and size, TinySec provides an effective authentication mechanism.



(a) TinySec-AE packet format

(b) TinySec-Auth packet format

(c) TinyOS packet format

Figure 2.3. The TinySec packets (Karlof et al. 2004)

2.1.4. LEAP

Many traditional sensor networks rely on sink devices to process the real-time data collected from the sensor nodes. Repeated events from similar nodes yield overhead which drains the battery life. Despite the numerous limitations of sensor nodes, they have a certain degree of computation and communication power. In order to reduce the overhead, some of the processing can be done within the network while some of the unnecessary or duplicated data are eliminated and aggregated. This method causes less computational load on the sink device and communication overhead among the sensor nodes, thus lifetime is increased. Chen et al. (2006) proposed this technique called in-network data processing. Here, an intermediate proxy node is chosen to have processing function for data streams and forward processed data to sink device.

Passive participation is similar to in-network processing, where the sensor node decides its own action upon the transmission from its immediate neighbors. This requires messages between two nodes in the channel to be verified by a third node. Existing keying mechanism in TinySec does not support this technique. Zhu et al. (2006), provided a key management mechanism, *LEAP*. It is a mechanism that is composed of several building blocks as shown in Figure 2.4 which will be explained in this section.

LEAP supports four types of keys to provide security for different communications. Individual keys are used for communication with the base station by every sensor node in the network. Pairwise keys secure communication between two sensor nodes which require privacy. Cluster keys are provided for local broadcasting to support data aggregation, passive participation for sensor messages and routing information. And finally, group keys are used by the base station to broadcast messages among a group of nodes.

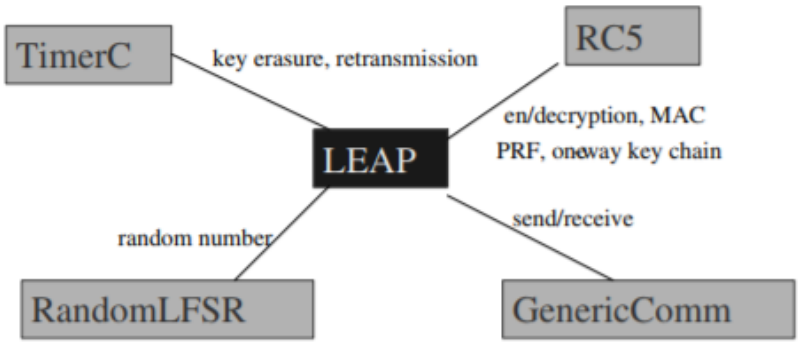


Figure 2.4. The building blocks of LEAP (Zhu et al. 2006)

LEAP is defined over static sensor networks where the base station has complex computational capabilities and sufficient energy supply. Sensor nodes have similar features with storage for hundreds of bytes provided for key management mechanism. The immediate neighbors are not known at joining. Adversaries have capabilities such as injecting packets, eavesdropping on the channel, compromising nodes to retrieve all the information on it. However, the base station should not be compromised.

The communication patterns are defined as unicast, local broadcast, and global broadcast to define the level of reliability required in for exchanged messages. The effect of attacks should be at the minimum and the system should not be completely compromised. Message fragmentation proved to cause difficulty and complexity in implementation, therefore the system requires the size of the packets to be sufficient. Lastly, one of the main goals is to reduce the heavy computation and number of transmissions to increase the efficiency of power consumption.

Further, into the design of *LEAP*, there are separate key generation and distribution schemes for each of four keys. Every node in the network has an individual key, they are either pre-loaded or generated. They are generated by pseudo-random functions using master keys. Secondly, pairwise keys are mostly used for immediate neighbors but for multihop neighbors *LEAP* offers multiple identity models designed by Douceur (2002) for distributed systems those absent from central entity. The newly added nodes generate pairwise keys with all of their neighbors with four steps: pre-distribution, neighbor discovery, pairwise key establishment, erasure. There is a lifespan for these keys, and the new node does not need to authenticate itself, however, neighbors use MAC to verify their identity to the new node. The lifespan of the keys should be as optimal as possible considering the transmission rates and packet sizes. Since the new node does not authenticate itself, an adversary can inject many unauthenticated joining messages and flood the network. However, *LEAP* offers solutions to make these attacks as infeasible as possible.

Cluster key generation is followed by the pairwise key establishment. The node can generate a key for immediate neighbors and encrypt the key with its pairwise key to es-

tablish. Finally, the group keys are used by the base station to send information to all nodes. It is possible that base station encrypts and broadcasts a message with its cluster key and sends it to its immediate neighbors. The neighbors decrypt the message for themselves and re-encrypting it to their neighbors with their own cluster key. However, this means a lot of encryption and decryption that uses too much energy. Here an efficient group rekeying is proposed. It has two steps: Authenticated node revocation and secure key distribution. $\mu TESLA$ is used for node revocation and TinyOS beaconing protocol is used for key distribution.

Authors propose a one-way key chain based authentication to provide local broadcast. Instead of delayed key disclosure and time synchronization as in $\mu TESLA$, authors propose one-way function based key chains. The node only needs to authenticate itself to immediate neighbors. However, the scheme is prone to inject and jamming attacks.

LEAP is implemented on TinyOS and uses its timer component, TimerC for key erasure in pairwise key establishment and retransmissions. Pseudo-random numbers are generated by a linear-feedback shift register (RandomLFSR). Message encryptions, MAC, pseudo-random functions, and one-way key chain are based on RC5 with CBC-MAC. GenericComm interface of TinyOS is used for send and receive operations.

LEAP provides some critical and solutions for reliable and efficient communications among the sensor network devices. Local broadcasting, data aggregation, and passive participation made possible here without significant loss of energy and memory.

2.1.5. Authentication Framework Using Identity-Based Signatures

There is a long list of solutions for broadcasting and multicasting mechanisms for WSNs. However, the most of the proposed works do not emphasize on the importance of quick authentication of messages in realistic settings. This work of Yasmin et al. (2010) brings light to three main topics. Firstly, a fast authentication method for broadcast and multicast messages is provided. Secondly, the work continues on the verification of the sender and the content integrity of the messages. And lastly, they proposed a verification method for messages from users outside the network. Authors point out that in untrusty environments the user authentication is at least as important as the confidentiality of incoming data.

Previously proposed $\mu TESLA$ is regarded as a good solution for broadcast authentication and resource handling. Nonetheless, the nature of delayed mechanism and limitations on the number of senders bring complications. Furthermore, the public key mechanisms and certificates of $\mu TESLA$ increases the processing time.

User authentication fundamentally requires mechanisms for access control and session key establishment. The centralized solutions results in single-point of failure, DoS attacks and injection attacks. Also the communication with a central unit affects battery life.

The solution in this topic consists of two cryptographic schemes: Identity-Based Signature (IBS) and Identity-Based Online/Offline Signature (IBOOS). IBS provides a de-

centralized verification to all sensor nodes during broadcast to eliminate injection attacks. IBOOS allows every sensor node to identify intruders without loss of memory space.

The proposed framework is partially-centralized. It relies on a base station for complex computations such as private key generator for sensor node and user registration, partial signature generation, and sensor node revocation. The base station is a trusted node that initializes the system, maintains the public system parameters, and manages the private keys. It uses an identity-based one-pass key establishment protocol that reduces the communication overhead by minimizing the number of messages. This protocol protects the framework against man-in-the-middle attacks.

The sensor nodes can finalize their own signatures to send their messages. They can also verify the messages they receive. The system parameters and unique IDs acts are used as public keys. The illegitimate nodes can be detected by sensor nodes through these parameters and time stamps. The signature schemes are confidentially sound. IBS is based on Elliptic Curve Discrete Logarithm Problem while IBOOS is based on Discrete Logarithm Problem.

The security analysis has shown that the authentication, verification, integrity, freshness, and session key mechanisms give satisfying results. The attacks considered in this framework are active, DoS, node compromise, and false injection. Finally, the performance analysis prove quick broadcast, storage, computational, and communication efficiency, multiple senders, and scalability

2.2. Access Control with Data Stream Engines

Before elaborating on the idea of determining the roles to actors on the Internet of Things, we need to understand where access control stands. Access Control is defined as a measure of restrictions to the resources of an organization regarding the identities of persons or other subjects (Ferraiolo et al. 1993). In terms of information security, access control is a collection of solutions to limit the access to computational systems through physical or logical mechanisms to both authorized and unauthorized subjects. These mechanisms should be able to determine roles and distribute credentials to access the information system.

Specifically, the Internet of Things defines two essential roles in the network: data holders and data collectors. Data holders are entities those can exchange data privately and exposes data according to the credentials of requesting target. Data collectors authenticate, verify, store, process, analyze, and visualize the data in real-time according to target requirements (Alcaide et al. 2013).

Data Stream Systems have more constraints than DBMS (Database Management Systems) in terms of computational power, real-time processing of a large amount of data, unpredictable transmission rates, and temporal factors as stated by Sicari et al. (2015). Regarding these constraints IoT and data stream systems have common issues, thus access control mechanisms designed for data stream architectures could inspire IoT frameworks.

Nonetheless, we must not forget the disparate constraints such as computational power, low battery power, and small memory.

2.2.1. FT-RC4

Aside from the need for sensor networks, there is a wide range of demand for on-line data stream systems such as, network monitoring, health care systems, tracking, telecommunications which require real-time processing. Data streams have unpredictable and continuous nature where they enter their dedicated system at high rates. The channels where the data streams are communicated are lossy and there is not enough storage for buffering. Since due to synchronization issues it is infeasible to keep track of lost data and retransmit it without keeping the channel busy. Since many of the real-time systems require security, the solution must be fault-tolerant.

Data stream systems manage large volumes of data without boundaries, therefore it is almost impossible to store all data at once and process it afterward. Aside from storing, querying these stream systems have also similar nature as input data. The queries required to be continuous and continue to evaluate incoming data.

Traditional security requirements are all applied here. However, it is infeasible to answer all of them. There are also application-specific requirements. Conventional stream ciphers are not sufficient in these systems since it will be impossible to retrieve the message when synchronization between key stream and cipher is failed. Desynchronization is likely to appear in communication and application layer. During high transmission rates, small memory, low computational power, and energy failures may cause desynchronization.

There have been several works to provide a solution to security issues of data stream systems. Fault-tolerant solutions for stream databases, role-based access control models, designated block ciphers for confidentiality and integrity.

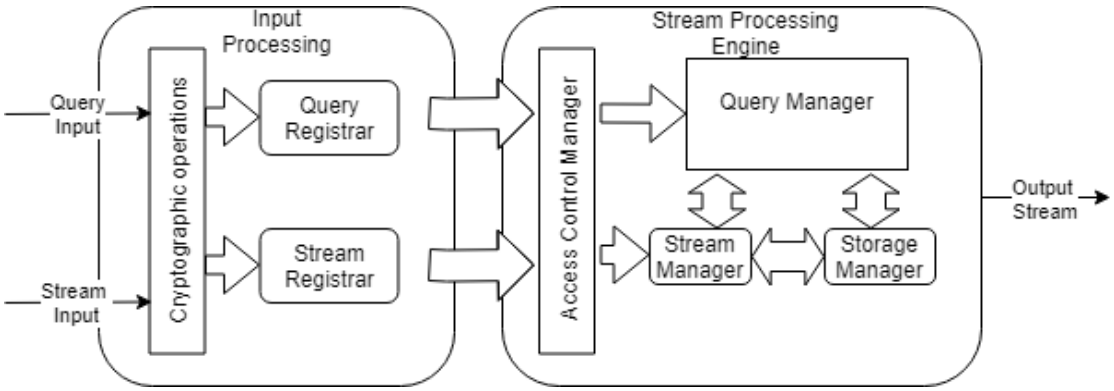


Figure 2.5. The Nile (Hammad et al. 2004)

Ali et al. (2005) proposed the FT-RC4 as an extended security architecture based on the Nile data stream engine (Hammad et al. 2004). The Nile has several components

for streams and query management (Figure 2.5): Stream registration, query registration, stream manager, query engine, and storage manager. While registration component facilitates weak access control mechanisms; the stream manager handles multiple streams and acts as a buffer, query engine processes the incoming queries, and finally, storage manager provides computational efficiency by summarizing data.

FT-RC4 builds two additional components. The general structure of the FT-RC4 can be seen in Figure 2.6. The first component provides cryptographic operations such as confidentiality, integrity, authentication, non-repudiation while operating outside the engine. The second component manages access control policies of authorized clients regarding their access to target data. This component resides in the engine, operating after registration components.

Aside from providing new components to Nile architecture, FT-RC4 extends the RC4 stream cipher scheme which does not operate when desynchronization occurs. This is overcome by breaking the stream into separate cycles while each cycle is encrypted separately. Both the sender and receiver have three steps before sending data.

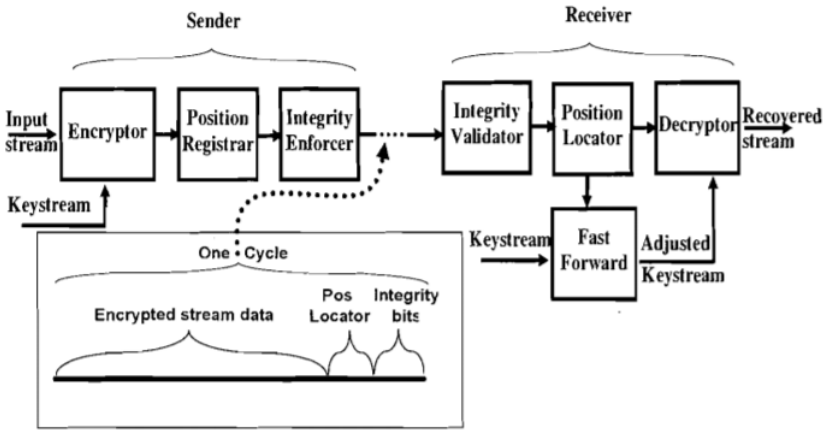


Figure 2.6. The general structure of FT-RC4 (Ali et al. 2005)

Sender by using stream position bits submits the input stream that is XOR-ed with key-stream into the position registrar. Uses hashing to generate an integrity value as a digest at integrity enforcer. One cycle is transmitted to the receiver after both position and integrity digest is appended at the end of the encrypted data. The receiver operates in reverse steps with the additional fast forward step. Firstly, it validates the integrity using the integrity digest and passes it to the position locator. Next, the position locator uses the position bit to adjust the position on key-stream at fast forward. Fast forward validate the location of the position, if it is not the correct position notifies the decryptor to discard the message to terminate the communication for the missing cycle. Fast forward also make sure if the message is corrupted if there is no loss. Finally, decryptor retrieves message using the correct key-stream. These steps are repeated until the last cycle has been successfully decrypted.

FT-RC4 requires more processing time than RC4 with additional data. Furthermore, as the cycle size increases the data loss is increased. Since, when desynchronization occurs, the whole cycle is lost. There are granularity solutions with additional computational overhead to recover the data.

2.2.2. CADS

Previously, FT-RC4 addressed many issues in data stream security. However, a mechanism for fast updating and temporal completeness was not present. Papadopoulos et al. (2007) propose Continuous Authentication on Data Streams (CADS), addressed this issue by providing an extensive and efficient framework through database outsourcing.

Database outsourcing systems have three main actors: Data owner (DO), the service provider (SP), and client (see Figure 2.7). DO outsources its database to at least one SP with better computational power. This way it does not need to have a fully capable database management system. SP has tools for advanced query processing. SP can operate for more than one data owner to improve the efficiency of the data stream system.

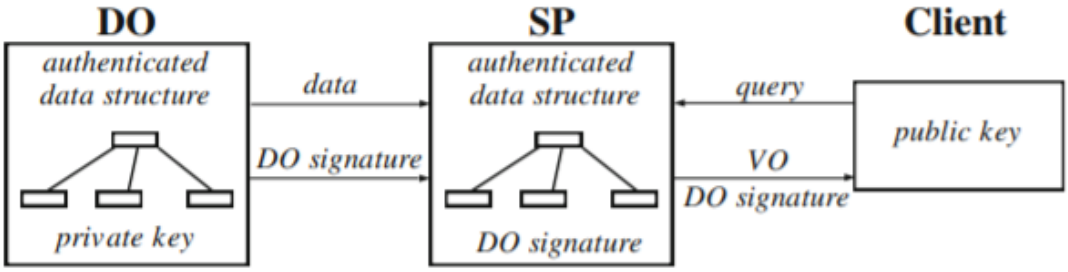


Figure 2.7. Outsourcing operations (Papadopoulos et al. 2007)

CADS eliminates the single point of failure with separating the operations. It also provides completeness through authenticating the records coming from DO and presenting the desired results from queries. Clients obtained the ability to verify the correctness of result with signature and public key acquired from DO. CADS finally focuses on issues such as a large number of intensive queries and provides mechanisms that increase the productiveness of query processing, reducing the communication overhead and temporal correctness.

The organization of the query processing is followed by communication between DO, SP, and Client. Firstly, DO generates a private key for itself and a public key that is available to the client. DO signs the dataset and transmit it with signature to SP. SP stores index the dataset with the aid of a mechanism called Authenticated Data Structure (ADS). Later, when the client issues a query, SP generates a verification object(VO) then sends VO and signature of DO to the client. The client verifies the correctness using the public key acquired from the target DO.

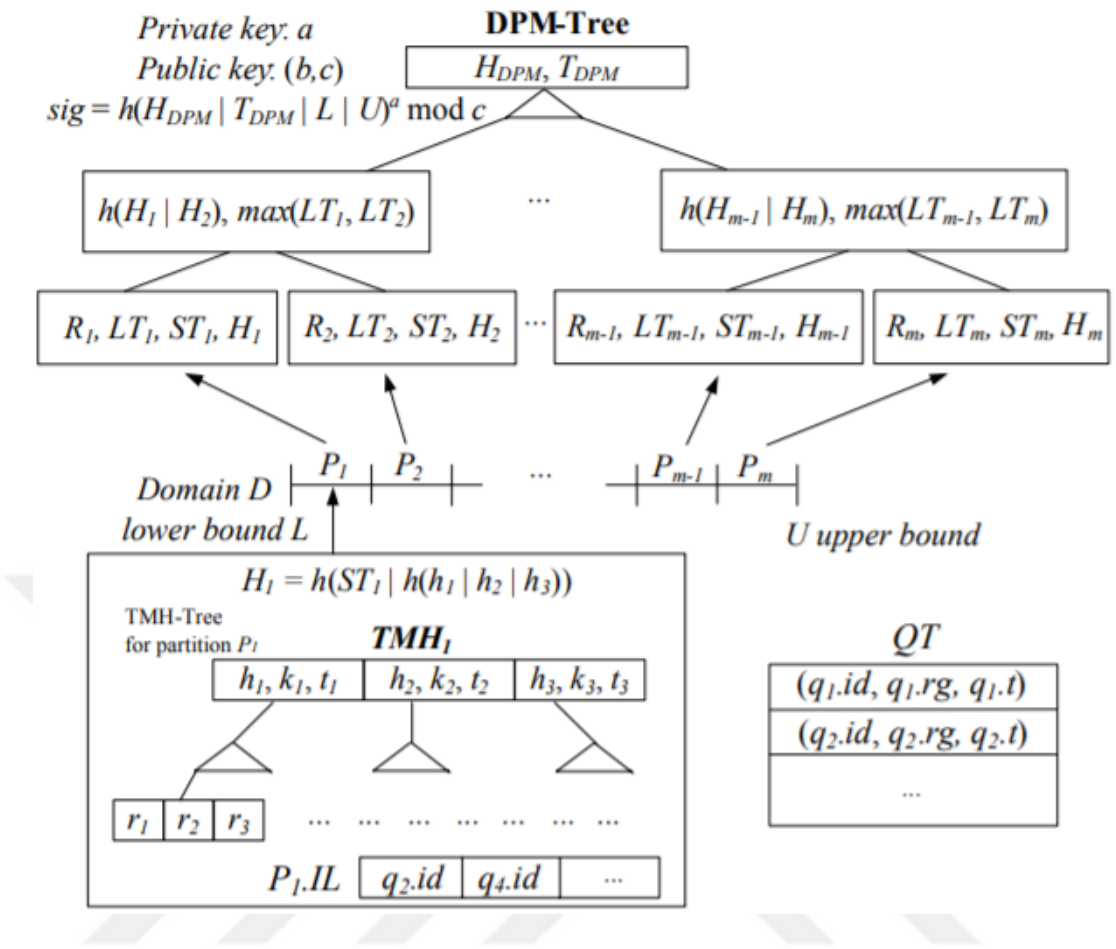


Figure 2.8. TMH-tree and DPM-tree for indexing (Papadopoulos et al. 2007)

SP collects both the primary key and the search key tuple to index it using Dynamic Merkle Hash-Tree (DMH-Tree). DO and SP maintain identical DMH-Trees, while DO computes a hash value obtained by the root of the tree. This hash value is signed with RSA to produce a signature and result is stored in SP. DMH also provides flash updates due to its similarities to B+ tree insertion and deletion operations. SP also provides boundary records to provide completeness by traversing the tree to locate lower and upper boundary values. Furthermore, RangeDMH algorithm is applied to complete VO. The client verifies the source of its results by keeping tree-structure information through Temporal Merkle Hash-Tree (TMH-tree) and Domain Partition Merkle-Tree(DPM). Along with the VO, the algorithm uses the resulting information to compute DO signature. This is called the reference solution, REF of CADS (see Figure 2.8).

VO generation with REF is necessary to provide temporal correctness. There is also timestamp component provided with VO. Despite the fact that there are temporal correctness and completeness, false transmission still occurs for data that have not affected by updates yet.

Indexing scheme in CADS is provided through indexing key tuples of domain par-

titions in Temporal Merkle Hash-Tree (TMH-Tree) and indexing partitions in Domain Partition Merkle-Tree (DPM-Tree). These indexing schemes support multiple updates simultaneously.

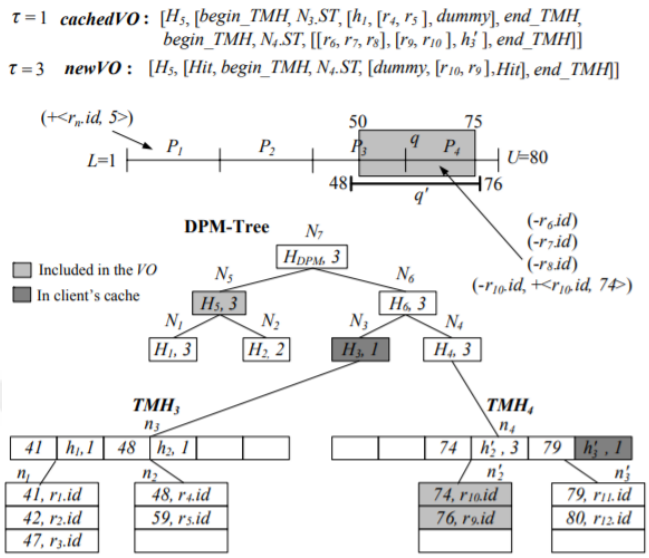


Figure 2.9. Query monitoring in CADS (Papadopoulos et al. 2007)

Queries in CADS are monitored through an algorithm using the virtual caching mechanism (VCM) provided in the work of Papadopoulos et al. (2007). SP receives the update list from DO and creates a set of affected partitions and affected queries. The client merges the updated results sent by SP using CombineVO (see Figure 2.9).

CADS is evaluated for single and multiple owners and the proposed mechanism performed well. The increase of DOs increased the efficiency of the system. CADS is extended by Papadopoulos et al. (2010) with additional features. This work aimed to provide security and granularity for relational streams. Optimal indexing granularity is achieved through an algorithm called Bestm which designed to operate over changes in the data distribution. Also A-CADS (Adaptive CADS), a variable-length partitioning structure is developed to minimize the false transmissions and empty partitions. It also provides dynamicity for structures relative to data distribution change in order to achieve high performance.

2.2.3. Lightweight Authentication of Linear Algebraic Queries on Data Streams

Stream outsourcing is another approach for authenticating the streams. Papadopoulos et al. (2013), focused on trust mechanisms in unreliable channels. However, this works assumes DOs as trust clients. As in previous work, signatures are used to authenticate the streams. Servers process queries over unions of streams by clients and produces result correctness to clients.

A simple architecture developed where input streams are treated as linear algebraic queries. This scheme provides authentication using sum and dot product operations on dynamic vectors and products over dynamic matrices where matrices are generated by different sources. These operations are used for group and join queries, data aggregation, passive participation, eliminating duplicate streams, and event processing.

The security goal of the trust is provided through integrity with signatures and freshness through counters. The work also aims to achieve a standardized tool for error-checking to provide reliable file transfer.

The system settings are consist of DO outsourcing data streams to third-party servers, processing a set of machines that generate and observe the streams and clients submitting a continuous query. Machines in the system maintain a small summary of the stream and compute a signature to send to the server. Servers process the query and send result and proof of correctness to the client to verify. This scheme is in the form of building blocks, that allows observation of performance over three of the operations of linear algebraic queries. Authors, build the scheme on strong cryptographic assumptions. The exchange of authentication elements can be viewed in Figure 2.10

A stream authentication protocol defined with a set of five algorithms: KeyGen, Update, Sign, Combine, and Verify. Former three are probabilistic algorithms while the remaining latter are deterministic algorithms. KeyGen algorithm generates a tuple of a secure key and public information with given security parameter. Update algorithm takes an index, secret key, a summary of the stream, and incoming data as input to generate an updated summary of the stream at the given index. Sign algorithm takes the index, secret key, a summary of the stream, and an epoch as a counter for input to produce a signature. Combine algorithm takes the union of signatures, public information, and streams at given epoch as input to produce proof of result correctness for the given epoch. Verify algorithm takes the secret key, proof of result correctness for the given epoch, result at a given epoch, and epoch to output a verification output: Yes or no.

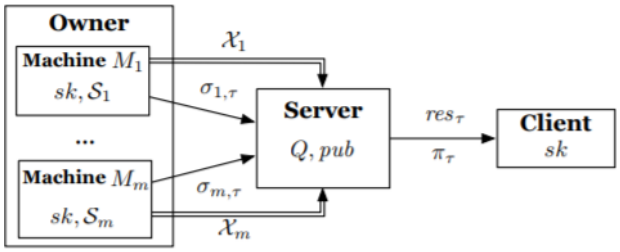


Figure 2.10. Lightweight authentication flow (Papadopoulos et al. 2013)

This lightweight scheme is performed over; Dynamic Vector Sum Authentication (DVS), Dynamic Matrix Product Authentication (DMP), and finally Dynamic Dot Product Authentication (DDP). The schemes evaluated for the group by query, join queries,

data aggregation, similarity measures, event co-occurrence. All of the authentications showed satisfying performance in the result of experiments.

2.2.4. RBAC Inspired Access Control Model for Data Stream Management Systems

Role Based Access Control (RBAC) Models are introduced by Sandhu et al. (1996). This model abstracts the role of assigning model in organizations into information systems. It is inspired by the early multi-user computer systems. Since organizations do not change their activities frequently, the roles of system actors are more dynamic. It is more efficient to separate the actors into responsibilities and qualifications in the system and make their role transitions easier. Roles also essential focus on authority which is the main concern in access control models.

Components of RBAC are defined as users, roles, permissions, and sessions. There are a certain set of relations defined for the model. The relations can be listed as; many permissions could be assigned to many roles and many users can be assigned to many roles. There should be a mapping for each session to a single user and another for each session to a set of roles and a set of permissions.

Lindner and Meier (2006) proposed an RBAC based Data Stream Management System (DSMS) security architecture. This model does not only provide an efficient access control mechanism but also improves system performance. This model is integrated into the Borealis Engine as shown in Figure 2.11. Moreover, it provides object-level and data-level security.

It is important to model the security upon user interaction in the system while concerning the asynchronous nature of the DSMSs. The authors emphasize three major threats. Firstly, introducing new information into the system without proper security measures may cause the disclosure of data and internal information. Secondly, modification of existing data without proper measures in terms of integrity and confidentiality can cause the system not to operate well as well as affect the environment. And lastly, denial of service attacks (DoS) should be concerned with proper measures for availability and intrusion detection.

The framework defines three main tasks. The first task is issuing the roles to users properly, thus every activity in the system can be associated with a user. The second task is for distributing permissions properly to users, therefore they can only access to objects they are allowed to. The third task is to guarantee the confidentiality and integrity of the data in DSMS. In this framework, Session Manager and Authenticator are associated with the first task. Authorizer, User Abstraction Layer, and SecFilter can manage the second task. Lastly, Encrypted Transport component can perform the third task.

The general RBAC model for DSMS is shown in Figure 2.12. Encrypted Transport component is located at the outer layer of the architecture to receive requests. It directly communicates with the Control Channel which is connected to the Session Manager. Session Manager, Authenticator, Authorizer, and User Abstraction Layer are in the same level

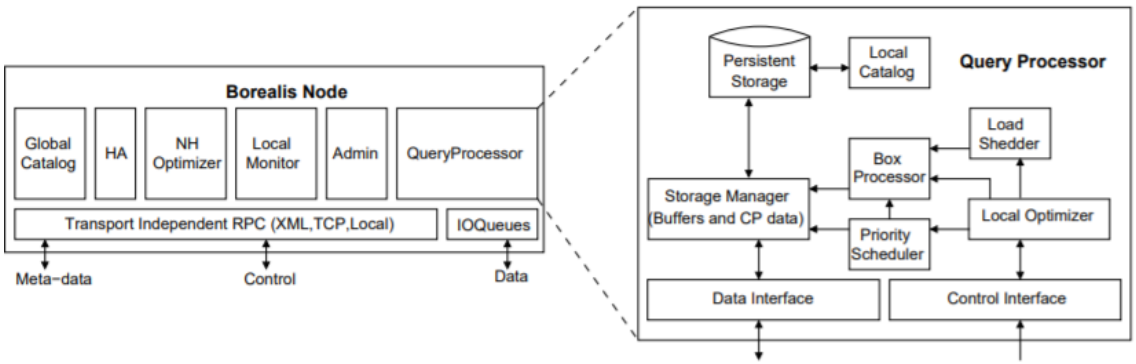


Figure 2.11. The Borealis Stream Engine architecture (Abadi et al. 2005)

along with system catalog, admin, and quality of service components. Query Processor takes input from Encrypted Transport through I/O Input Channel. It processes the incoming queues in the Queue Manager and Operator Executer using the Optimizer, Scheduler, and Monitor. It sends the results firstly to SecFilter, then I/O Output Channel. The I/O Output Channel sends it to the Encrypted Transport to encrypt the data and send the results in the output stream.

OxRBAC Model is abstracted into user, role, object, and permission entities (Figure 2.13). Users can have sessions with roles. Roles have “has” relation with object and permissions. Users can have ownership over objects. There are security classes defined over Borealis: Session Managers can have sessions, Authenticators can have roles, and Session can have roles. Session Manager and Authenticator can use SecAdmin, while Authorizer can use both SecAdmin and SecFilter. DataPath and AuroraNode use SecFilter.

Object level security is provided through defined system objects: Schema, Stream, Query, and System. There are basic access permissions such as view catalog, view object, add, set query status, subscribe, read the tuple, etc.

In the data level, security can guarantee a user who is able to get aggregated value with only sufficient credential. This mechanism is provided with the aggregate-read-write mechanism.

The mechanism is analyzed against several security threats on Borealis and resulted in efficient results. However, DoS attacks can be avoided by more policies to limit user access.

2.2.5. Security Punctuation Framework

There exists a variety of stream-based approaches. One of them is proposed by Nehme et al. (2008) where the credentials are streamed along with the streamed data. This approach, embeds security constraints under the name security punctuations, into the data st-

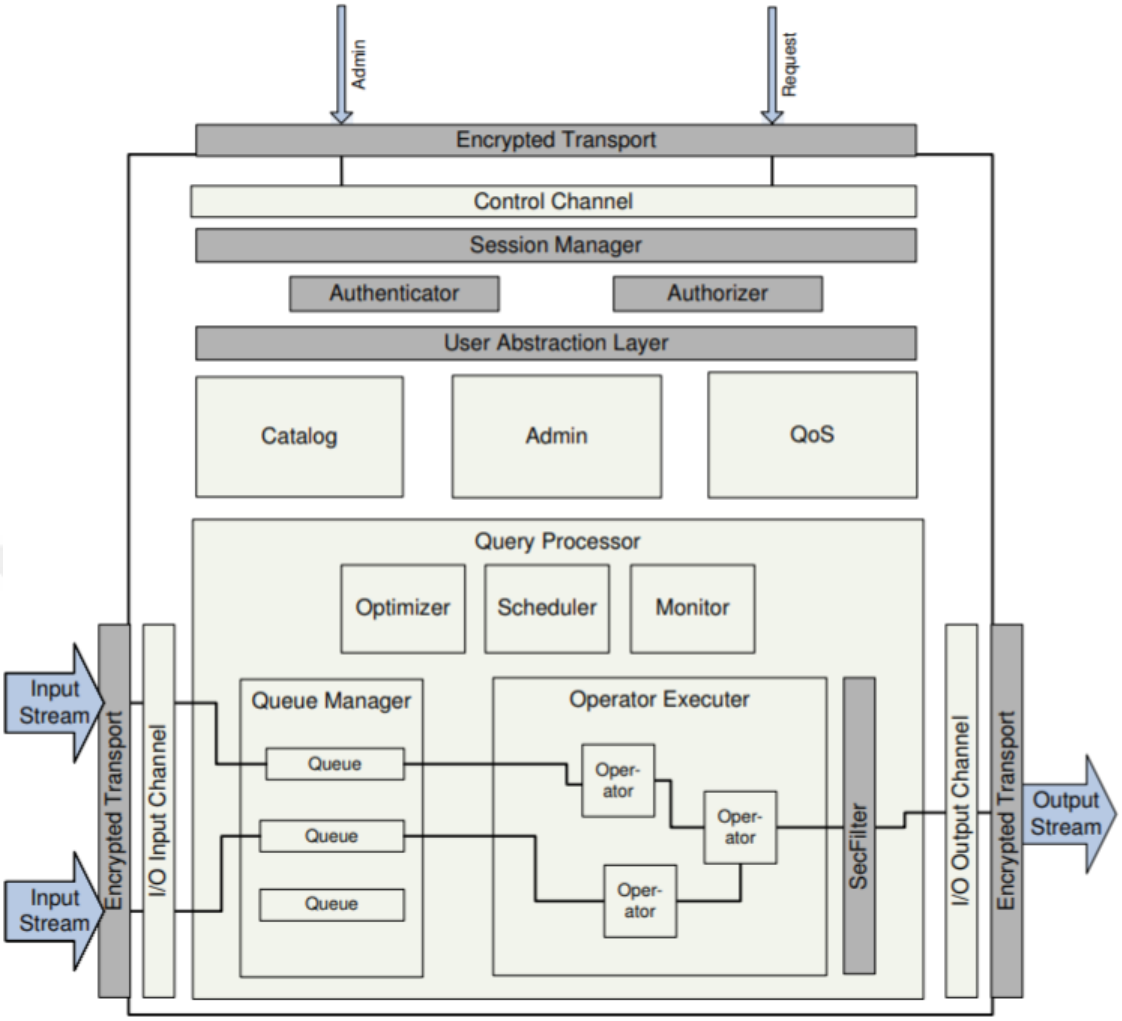


Figure 2.12. The RBAC Model for DSMS structure (Lindner and Meier 2006)

ream as a meta-data to provide flexibility and dynamicity on security measures. Security punctuation framework is not an access control model but an enforcement. Its abstraction is given in the Figure 2.14.

Security punctuation framework focuses on security-aware query operations aside from enforcement which acts pervasive against context-aware security threats or privacy over real-time services those process sensitive information. Authors describe existing approaches such as store-and-probe, tuple-embedded, and punctuation-based. Nehme et al. (2008) emphasized the punctuation-based approach. This approach provides faster processing of data as credentials which are verified during streaming. Also, those punctuations could be shared by multiple tuples which reduce redundancy and minimize memory usage.

The system defines two types of users: Data providers (DPs) and query streamers

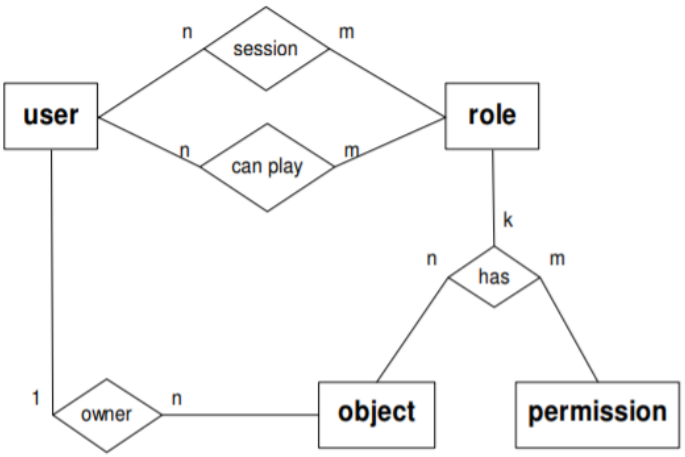


Figure 2.13. 0xRBAC Security flow chart (Lindner and Meier 2006)

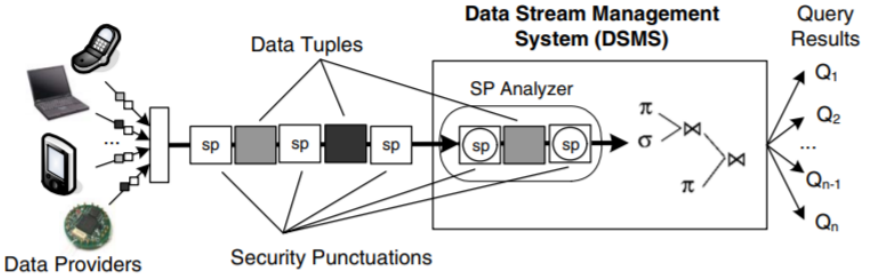


Figure 2.14. The diagram of secure punctuation system (Nehme et al. 2008)

(QSs). Each continuous query submitted by QS inherits the same credentials QS has. The query processing engine controls the credentials, verify QS, and discards the data that is not requested or not authorized query in the communication channel which it has access to. System labels entities as objects and subjects. Objects are the DOs of the framework while Subjects are the clients those request access to the object. The subjects retrieve their rights when they sign into DSMS.

The streaming model in this framework describes a security punctuation analyzer for combining the similar punctuations to reduce the memory usage and allow the server to define their own policies. New policies can be defined by the organizations. Although they cannot override the existing policies, they can add new constraints.

Security punctuations are meta-data streamed along with the message as described earlier. Their meta-data consists of blocks of data; Data Description Part (DDP), Security Restriction Part (SRP), Sign, Immutable, and Timestamp as shown in Figure 2.15. DDP is a collection of access control policy specified for the target object. SRP describes the

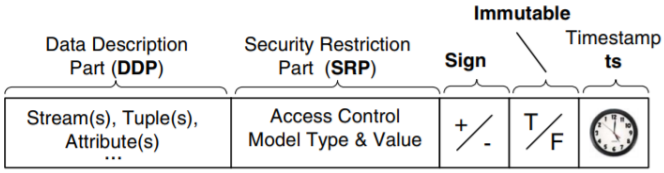


Figure 2.15. Security punctuation data (Nehme et al. 2008)

access control model and subjects that are authorized by the policy on the target object. Sign information is a specification of authorization; it is either positive or negative. An Immutable block indicates whether the secure punctuation data can be combined with other policies. And Finally, Timestamp indicates whether the time policy is active.

Security-aware query processing is proposed with approaches such as pre-filtering, post-filtering, and intermediate filtering. The first approach is achieved by each query are pre-loaded with access control while the second is achieved by filtering the rights after the query process. Although there is computational overhead, the latter approach performs better. The last approach is applied by discarding the data that no query can have access. The algebra is split into: Security Shield, projection, selection, join, duplicate elimination, and group-by.

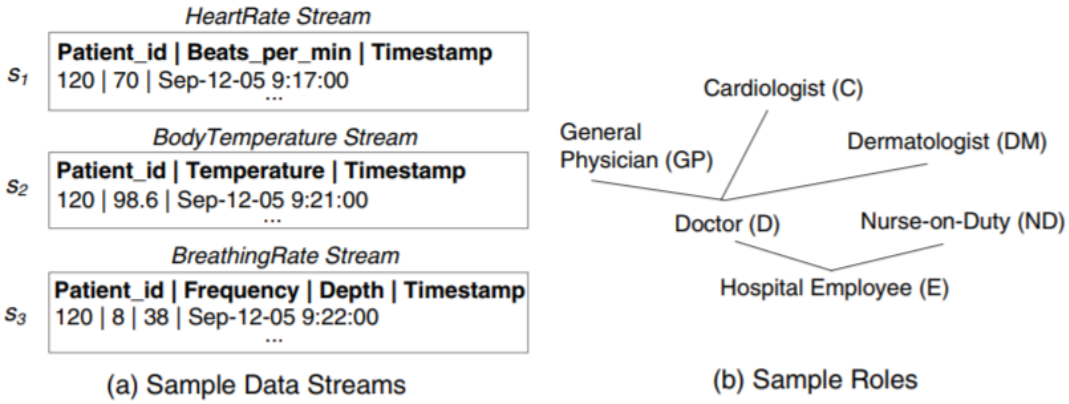


Figure 2.16. An application example (Nehme et al. 2008)

After experimenting with the framework, it is shown that the security punctuation is effective and there are minimal overheads as there quite similar to input streams in behavior.

2.2.6. Publicly Verifiable Grouped Aggregation on Outsourced Data Streams

Previous security architectures have not implemented any proper security measure for query result verification of untrusted clients. However, there may be competition among clients on outsourcing data streams. In the presence of untrusted clients, DO cannot trans-

mit its secret information either. This requires public verifiability. Nath and Venkatesan (2013) find a solution where a grouped-aggregation is provided.

Grouped-aggregation is when each of the data streams is associated with one or more groups. The authors designed the system in a way that the aggregation is maintained in an incremental fashion. A client can query the current aggregated data for groups. There is a histogram structure to maintain a data stream sum for each group. A mechanism called, Digest for Streaming Histograms(DiSH) is developed where it enforces the correctness of results from the server (Figure 2.17). This enables verification for clients without releasing sensitive information. In the presence of a large number of groups, it is infeasible for clients to verify all the groups in terms of overhead. The subset group queries are introduced as an extension to DiSH in order to improve the performance against a large number of groups. Nonetheless, implementation of this on limited-memory owners is not possible. The DiSH is evaluated with separate and synthetic datasets. Their solution provides reasonable overhead.

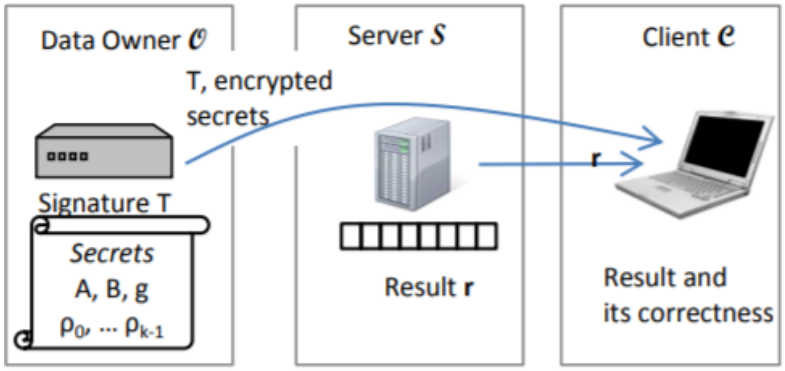


Figure 2.17. Dish Architecture (Nath and Venkatesan 2013)

The system communication model is illustrated in Figure 2.18. The system refers to DOs as delegators and clients as verifiers. DO has limited-memory to store all aggregations from groups. Clients also have limited-memory and cannot store all aggregations but it can perform verification of dynamic stream. The verification process is incremented with discrete timer ticks while the resulting stream is transmitted. Query partitions the stream and maintains the groups. As server and clients are unreliable they are still trusted by DO. Clients verify only when the correctness of results from all queries is verified.

There are solutions for synchronization provided. Query-ahead solution requires the client to plan ahead. Before receiving the results it sends requests to both DO and server to send their data within a given time. Buffering solution utilizes a sliding window mechanism to keep a certain amount of tuples within a given time.

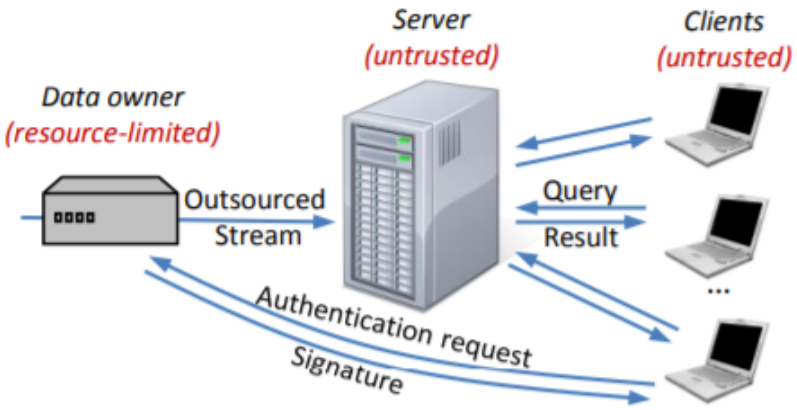


Figure 2.18. Stream aggregation (Nath and Venkatesan 2013)

Queries on a subset of groups are provided by considering two variants of queries; dynamic subset queries and static subset queries. Although dynamic subset queries have better performance for limited memory, static subset queries are more feasible. Deterministic and probabilistic signature functions are considered for dynamic subset queries in terms of efficiency. As for static subset queries, an overlapping subset of groups and concurrent queries are provided as solutions.

Experiments of this framework are provided for both real and synthetic data. The solution proved to be practical and effective.

2.2.7. ACStream: Tagging Stream Data for Rich Real-Time DSMS

Nehme et al. (2008) defined a security punctuation mechanism provided an efficient access control mechanism for data stream management systems. Nonetheless, there is a need for more sophisticated data streams with a new meta-data format. Cao et al. (2009) design a new streaming-tag format, called tick-tags are introduced (see Figure 2.19). This allows users to exploit the access control mechanism to provide more information-rich query results. However, the data streams attached with more information raise the efficiency concerns. Authors bring answers to these concerns through a scalable Stream Tag Framework (STF). Here tick-tags are attached to the data streams and transmitted along with them. Those streams have more privilege in the network.

Recent technological evolution brought the requirement for omnipresence for the devices we have in our daily lives. The expectations for a higher quality of living started to influence the smart devices surrounding us. Thus, many systems require to provide highly sophisticated profiles of actors in the systems, efficient and easy ways to reach data, and interactions between actors. These can be achieved through fine-grained tagging to eliminate overhead during real-time processing.

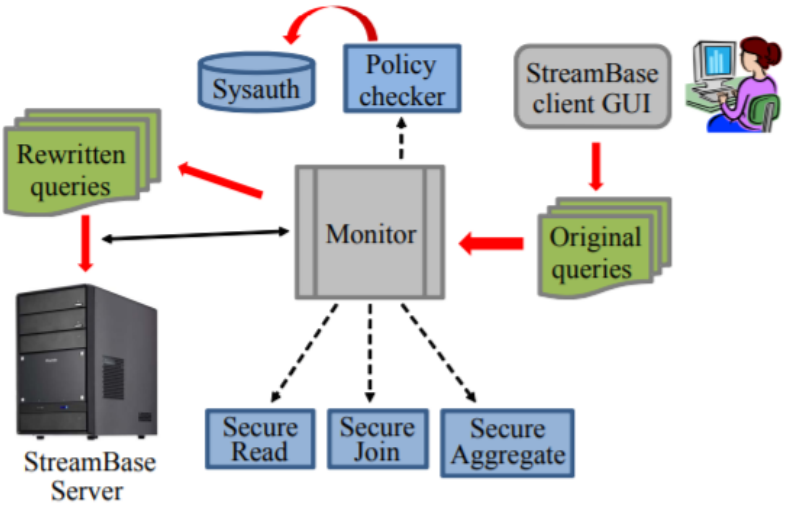


Figure 2.19. Stream aggregation (Nath and Venkatesan 2013)

Some of the earlier approaches include; tables, extended data tuples, and streaming XML data. Those approaches are prone to overhead and infeasible to implement due to computational limitations. Here, streaming tags are introduced. In addition to reducing those previous concerns, stream tags are dynamic since they are interleaved with streaming data. They also allow users to opt out of using them when they are not necessary.

Stream Tagging Framework aims dynamicity and personalization of tags respect to users and streamed data. Furthermore, continuous and ad-hoc query processing is developed. STF has four components, i.e. tag model, Tag Query Language (TAG-QL), tag-oriented query processing, and tag-aware query processing. Users submit their data tuples with tags in TAG-QL at client-side, and those queries are streamed together with tick-tags into DSMS. STF processes these queries in tag-oriented and tag-aware processing units and outputs them into tick-tag streams, tuple streams, and enriched tuple streams.

Data tuples in STF are consist of stream identifier, tuple identifier, taggable streaming object, and timestamp. An object can have multiple tags. Tagger users can create any tags with any content and attach them to stream objects. Tick-tags provide an efficient folksonomy compared to traditional tags. They are ephemeral, interleaved, accessible sequentially, streamed at high rates, and processed continuously. Furthermore, there is no limit to tag size.

The physical implementation of tick-tags is composed of tagger identifier, applicability information that provides a description to where the tags are applied to, the content of the tag, tag type, qualitative description of tag, lifespan of the tag, tag mode, and timestamp of tag generation. STF defines five tag types; objective, subjective, physical, acronym, and junk. Objective type of tag is a description of a state irrelevant to any actors of the system. Subjective tags imply personalized information. Physical tags depict the physical state of a subject. Acronym gives a personalized name for the state. And finally,

junk tags are meaningless. The sign attribute of tick-tags provides reputation points in order to decide their applicability of information. Mode attribute defines whether the user continues his earlier tag or overwrites it.

Tick-tags are generated manually by users. Tagger operators evaluate the streaming tags continuously and produce tick-tags to be interleaved with output streams. There are two tag-based query processing; explicit tag-oriented algebra and implicit tag-aware operations using TAG-QL. The first processing deals with simple query operations using algebra. They locate the values of tags and their associates through operations such as; select, join, and aggregation. The second processing deals with the correct propagation of tags through the query pipeline. Queries are processed with operations such as; filtering, join, and projection.

The framework is tested through various performance parameters. The results show that it is scalable, efficient in terms of output rate, processing time, memory usage, and computational overhead. The tag-aware feature does not decline the performance exceedingly.

2.3. Physically Unclonable Functions

Physical unclonable functions are first introduced by Pappu et al. (2002) as an alternative cryptographic key generation method based on intrinsic features of the physical elements. They are hardware counterparts of mathematical one-way functions as physical features are observable, yet hard to reproduce. Initially, they were referred to as physical one-way functions.

The early applications of PUF mechanisms were based on silicon or coating variations during the manufacturing process of IC. However, these features are hard to observe. However, Lim et al. (2007) proposed the Arbiter PUF which relies upon a series of switches and an arbiter circuit to identify the difference between two identical delay paths. Arbiter PUF is also the starting point of the challenge-response pair (CRP) structure.

A basic structure of PUFs is illustrated by Maiti et al. (2013) as shown in Figure 2.20. A group of challenge bits is processed in an array of PUF to produce the responses to each challenge. They are evaluated and compared in a higher-level environment and the results are used during the authentication requests. Basically, a legitimate node has a set of CRPs to authenticate itself.

In the contemporary PUF mechanisms, the CRPs are determined by unpredictable, unclonable, and unique statistical variations between the logic gates and IC circuits. There is a wide range of PUFs proposed to this day to answer the requirements of a variety of application contexts.

They are suitable for IoT infrastructures in terms of low overhead, simple hardware-based computations, and less complex cryptographic operations. However, there is external noise to discard during comparison operations. The designers should be wary of those and proposed solutions in order to extract the purest form of CRPs.

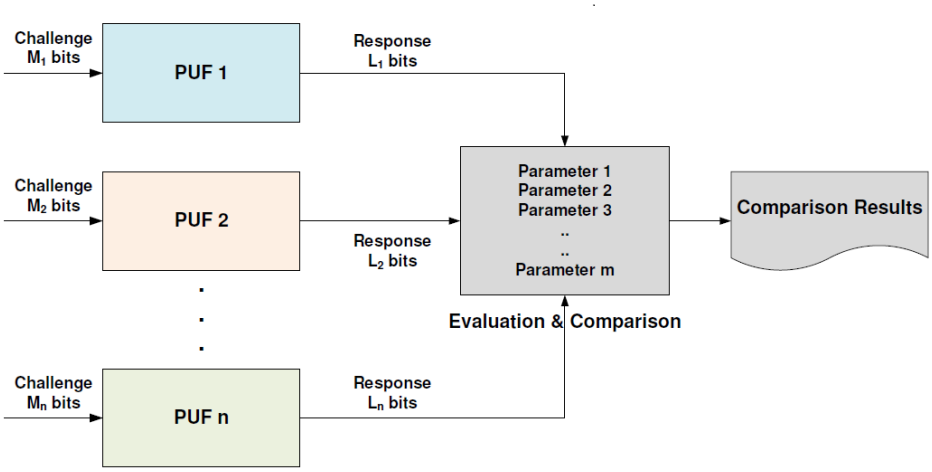


Figure 2.20. Basic structure of CRP model (Maiti et al. 2013)

2.3.1. Arbiter PUF

Lim et al. (2005) designed the very early application of PUF on authentication mechanisms: Arbiter PUF. At the time their work is proposed, the invasive and non-invasive physical tampering methods were newly emerging to extract critical information from the integrated circuits(ICs). There are existing tamper-sensing applications that were proposed previously, but they were not effective during the power cuts. Consequently, the need for pseudorandom functions has been increased to eliminate the demand for key storage.

There have been several physical applications for unlinkable and irreversible functions, nonetheless, until the development of PUF, those were not providing enough Challenge-Response Pair (CRP) space. This work stresses the statistical delay variations on wires and transistors those originate from manufacturing process using arbiter-based implementation.

There cannot exist two of the same silicon circuit. Hence, each of the circuits gives different responses to challenges. However, the noise has been the most significant feature of this implementation. The environmental variation, meta-stability, and aging can affect reliability by means of noise. The Arbiter-PUF uses an arbiter component to create a differential structure and compares two identical delay paths. The arbiter component is placed at the end of the delay paths(see Figure 2.21). The difference is derived from the configuration of the top and bottom paths. There is a maximum delay variation in manufacturing. Therefore, if the response delay is greater than this variation the responses are biased. The symmetry of the delay paths affects the inter-chip variation positively. This way, the nominal delay between the two paths will be smaller. This also reduces the risk of an attacker to intercept the internal components, since the circuit will be damaged.

Arbiter-PUF originally uses D-latch for an arbiter to introduce a skew factor. However, the authors point out that symmetrically implementable latches will improve performance.

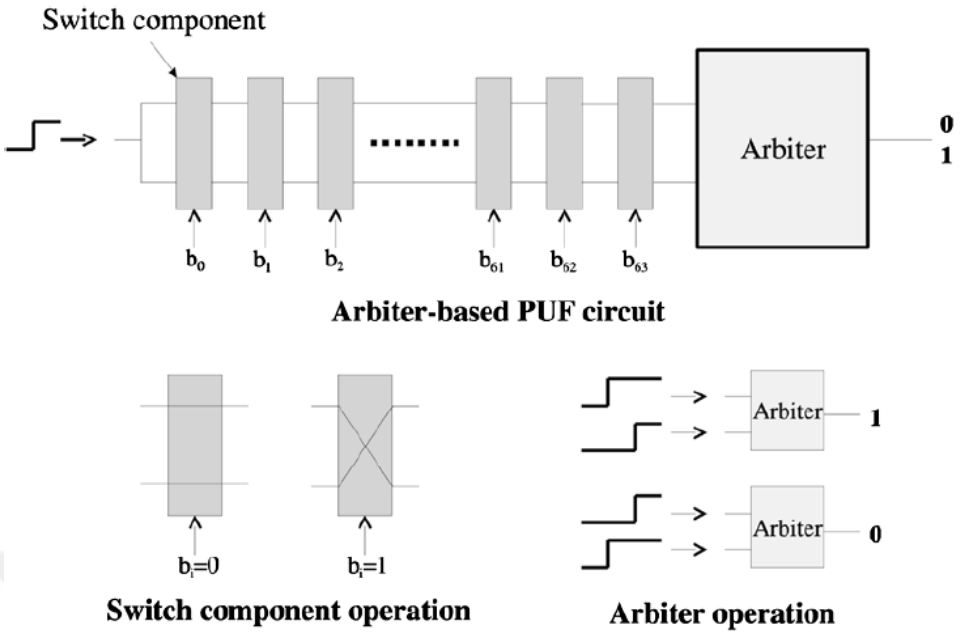


Figure 2.21. The Arbiter-PUF architecture (Lim et al. 2005)

The reference response is precisely measured in the experiments and a relative delay measurement reduced the environmental variation. The differentiation between these two measurements operates in a single cycle, unlike the early proposals. Arbiter system also provides a solution to meta-stability and thermal noise.

One of the challenges of an arbiter is virtual counterfeits in the form of software to imitate the original circuit to predict correct CRPs. This can be averted greatly by adding a nonlinearity factor to increase the negligibility of generating valid CRP. Feed-forward version of this arbiter-based scheme provides this nonlinearity through memory operations in the intermediate stages to determine the final challenge as illustrated in Figure 2.22.

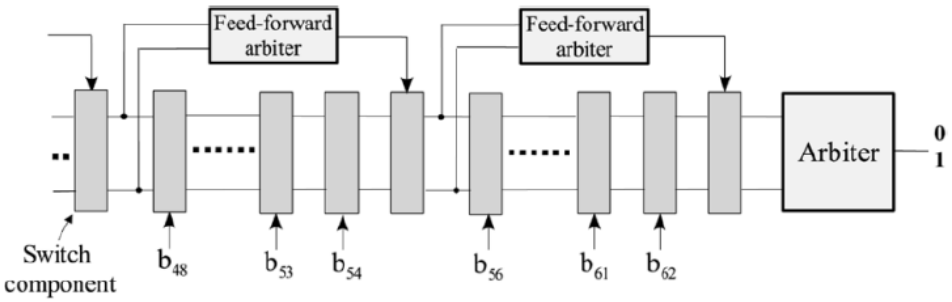


Figure 2.22. The Feed-forward arbiters by Lim et al. (2005)

The arbiter-based PUF is evaluated through certain characteristics such as inter-chip variation, environmental noise, and measurement noise. It is determined that inter-chip variation across the wafers is negligibly different than a single wafer. Notwithstanding, environmental variations are significant in PUF circuits. Electromigration and hot-carriers cause decline on physical components of ICs. One-month period of aging did not show a compelling decline, nonetheless for commercial use longer periods should be observed. There should be a better range of testing and error correction implementations for the reliability of this proposal.

During the modeling of arbiter-based PUF, it is important to know the average delay difference variation that is relative to the delay path and process variation. These values provide knowledge on the correctness of the given model.

The authentication capability is implemented through two error probability variables: the false alarm rate (FAR) and the false detection rate (FDR). FAR indicates that the authentication is failed and FDR is the wrong authentication that does not belong to the one that is validating herself. The identification probability(IP) indicates the probability of successful authentication with a given noise probability.

Finally, arbiter-PUF is proved to be applicable to the given physical characteristics of integrated circuits. It is stated that the evaluation properties aforementioned are promising for future technologies.

2.3.2. Ring Oscillator PUF

Suh and Devadas (2007) proposed an alternative approach to the arbiter-PUFs to get an advantage over the ease of implementation and entropy evaluation as well as the reliability. Nonetheless, this scheme is slower and consumes more energy.

The main concept of Ring Oscillator PUF (RO-PUF) is to generate secret keys by making use of the manufacturing-related fluctuations on oscillator frequencies. Ring oscillators are simple delay loops compared to the slightly more complex design of arbiter-design. They also do not need to be placed symmetrically. The main operation is to generate a fixed oscillator sequence pairs and compare their frequencies to generate the random bits.

The illustration of the RO-PUF circuit is shown in Figure 2.23. The entropy value in RO-PUF scheme is the number of independent bits that are produced from pair comparisons. It will be way less than the number of output bits since there are correlated outputs. As the associative law of Boolean Algebra, when the comparison of oscillator A and B yields the same result as the oscillator B and C, the oscillator A and C will yield the result from those operations. Also to prevent any correlated results, it is the best approach to use each oscillator once. So the number of bits in the key will be the same as the number of oscillators in the PUF circuits.

The errors are referred to as bit-flips in RO-PUFs. When choosing the pairs it is important to choose base frequencies. It is the best practice to pick oscillators that have far

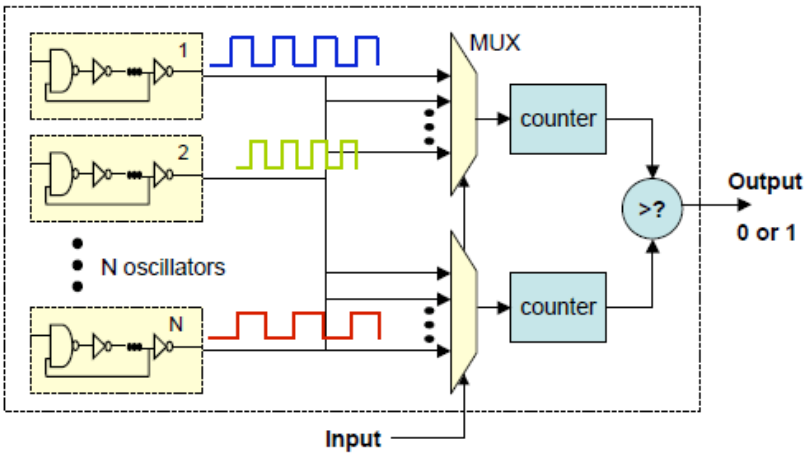


Figure 2.23. The RO-PUF Circuit (Suh and Devadas 2007)

base frequencies. Even during the environment related fluctuations, the flip will not occur. The authors perform 1-out-of-k masking scheme, however, there are other effective masking methods to reduce the bit-flips. For further bit-flips, error correcting codes can be effective.

Although it is important for the PUF circuit to generate exponentially-many challenge-response pairs(CRPs), the ring oscillator PUF can only generate a small number. The additional delay paths are similar to the ones which are described in the Arbiter PUF can help to increase the number of CRPs or FPGAs can be configured in a way to produce more unique bits from different parts of the circuit. CRPs should not be reused to avoid the man-in-the-middle attacks. These pairs are stored in the database of the trusted party.

The main authentication scheme can be seen in Figure 2.24. An legitimate device with records of CRPs receives a challenge and returns response. In the untrusted environment there is a mixture of legitimiate and illegitimate devices. The illegitimate devices do not contain the correct CRPs. Therefore, they will not be accepted.

The cryptographic keys are generated in two steps; initialization and regeneration (Figure 2.25). The initialization step consists of the PUF circuit and error correction syndrome. It is important to note that all the information requires non-volatile memory at this step are public. In the re-generate step, the data is generated by the PUF. The authentication takes place by using the bit-vector to select the correct pairs with the error correction by the stored syndrome. The mask data might reveal some information about the place of the ring oscillators since the pairs are selected according to their base frequency differences. But it does not leak any data about the output bits. Also, the error correcting code is hashed to be used as a cryptographic key. The security properties of this scheme might aid to physically strong processors.

Suh and Devadas evaluated RO-PUF by their inter-chip and intra-chip variations. It is

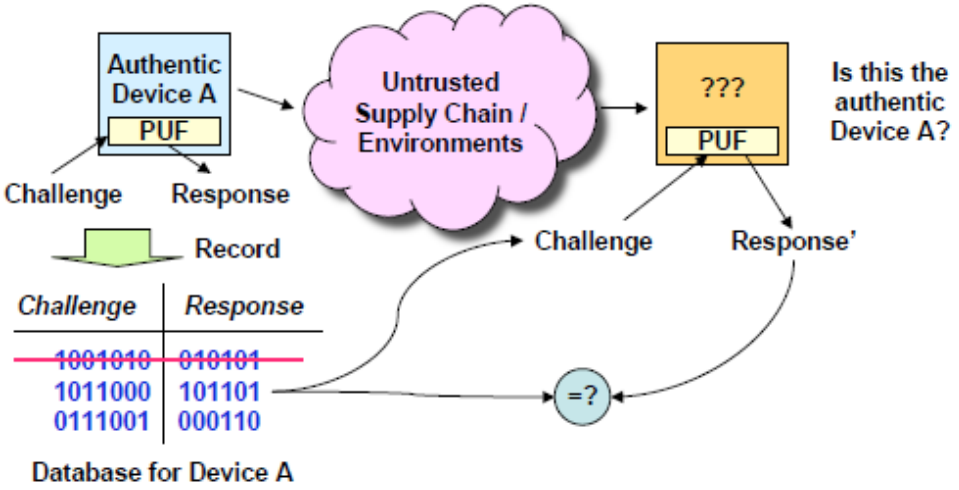


Figure 2.24. The authentication in RO-PUF (Suh and Devadas 2007)

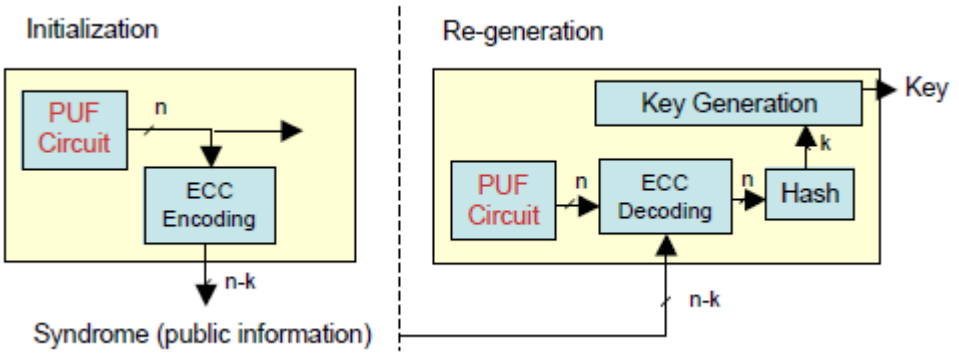


Figure 2.25. The Cryptographic key generation (Suh and Devadas 2007)

cryptographically measured by the means of uniqueness for security purposes and reproducibility for reliability purposes. The inter-chip variation is determined by the difference of output bits between two oscillators. The expected average variation is 50%, while the experiments result in 46.15%. The intra-chip variation is based on the environmental factors which determined by the number of output bits changed during the re-generate step with and without external noise. The expected variation is 0%, while the experiments result in 0.48%. The experiments give satisfactory results for the scheme.

2.3.3. SRAM PUF

In recent business practices, it became standard to include third-party intellectual property(IP) services into IT products. This brings two advantages. Firstly, they bring modularity to system designs and make them flexible. Secondly, these require a set of additional licenses. Therefore, IP services provide a higher income. However, these modular systems should provide insurance of protection for private data of users against security threats such as cloning, gray market production, etc. Numerous PUF schemes aim to pro-

vide solutions for three of the security services defined in IP protection chain; hardware IP authentication, hardware platform authentication, and complete design confidentiality.

Guajardo et al. (2007) refer to their Encrypt-then-MAC approach to the generic composition paradigms described by Bellare and Namprempre (2000) to support their solution in terms of confidentiality. However, the integrity of the scheme relies on redundancy functions. The security and authentication primitives are provided during enrollment and online phases.

An SRAM cell has a structure built of 6 transistors with two cross-coupled inverters as shown in Figure 2.26. However, the intrinsic fluctuations in SRAM creates unstable variations. Nonetheless, during the power-up they do not receive any external signals.

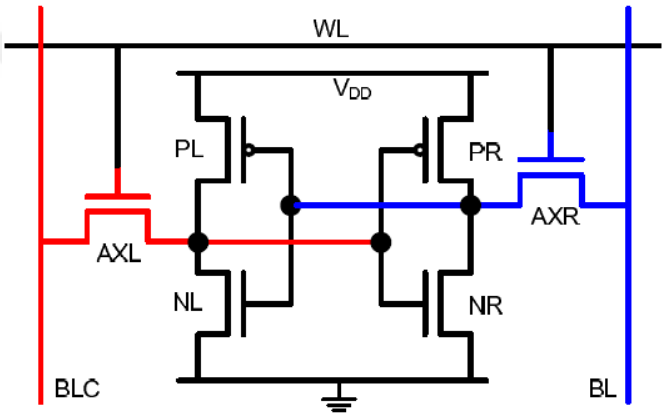


Figure 2.26. The structure of an SRAM PUF cell (Vijayakumar et al. 2017)

The SRAM provides simplified PUF scheme in terms of communication, assumptions, and cryptographic complexity. It is public-key encryption intrinsic to FPGAs and its implementation costs do not affect the overall system. This work cooperates fuzzy extractor and helper data elements into the design.

After the key generation, CRPs are destroyed via blown fuses after the enrollment phase and the resulting response is only available to the FPGA where the circuit is located. Helper data is generated during the enrollment phase. It is used when the key is reconstructed with the help of information reconciliation and privacy amplification. Information reconciliation operation is error correction code algorithm for decoding the response value. Privacy amplification is also called the randomness extraction that reconstructs the final value of the key using an appropriate hash function from the response value.

The work provides two separate protocols as partial and total privacy protocols. In partial privacy, the third-party provider has access to the IP block while the latter does not allow access to the third-party provider.

The key extraction protocol can be explained in several steps. Firstly, the encrypted

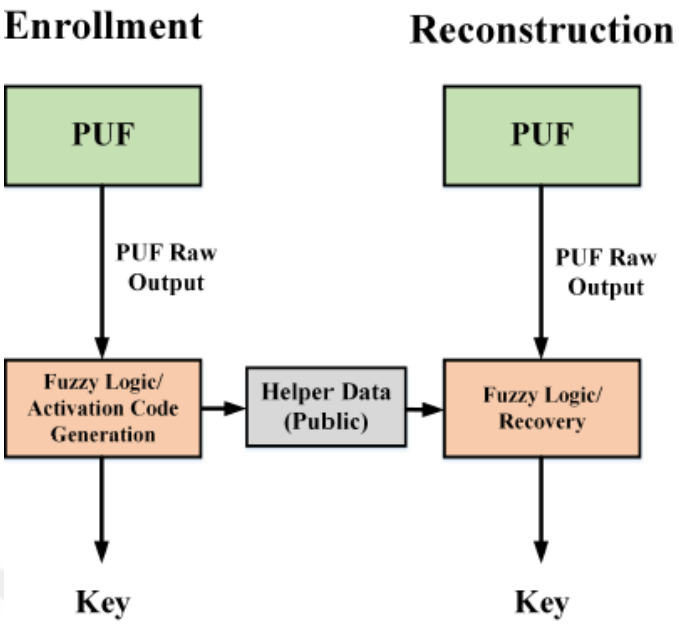


Figure 2.27. The Enrollment in SRAM (Vijayakumar et al. 2017)

bitstream is loaded to SRAM. Then, the PUF circuit is challenged with a chosen challenge from the challenge space. PUF response is measured while helper data is retrieved from the non-volatile memory. A fuzzy extractor is implemented to extract the key by using the generated response and the helper data as its inputs. Then, the bitstream is decrypted and FPGA is configured. This scheme provides hardware confidentiality and integrity. The entrollment phase is illustrated in Figure 2.27.

The software privacy is implemented by the encryption of software data with PUF generated CRPs. The response data from the CRPs are used for encryption. The challenge data from the CRPs are appended to the encrypted message and the resulting value is applied to Message authentication code (MAC) in order to provide integrity. SRAM also prevents a possible system bottleneck. Confidentiality is proposed in the form of a protocol for system developers of IP providers. The protocol consists of one decryption, one MAC operation, and two additional hash functions. It is designed within an honest-but-curious setting. The additional hash functions do not require hardware resources as they use the built-in AES-based hash.

SRAM scheme in this work is intrinsic to FPGAs as it is described above. The authors assume a security module that is based on the behaviors of different SRAM cells during start-up of FPGAs. The environmental noise is the least at this phase as they do not receive any other signals externally. Therefore, the random behaviors of cells provide unique information for key generation during the manufacturing process.

The protocol presented with the SRAM hardware is analyzed in terms of stability within a certain time period, stability towards temperature deviations, stability during aging,

and randomness of the key. Firstly, the experiments on time period variations have shown less than 4% of Hamming distance since the start-up. Secondly, temperature deviations are tested from -20°C to 80°C and resulted in a 12% Hamming distance at a maximum fraction. Thirdly, aging within 10 minutes periods are resulted in 4.5% of fractional Hamming distance. Finally, the randomness test succeeds at 49.97% of mean and 0.3% of standard deviation with a normal distribution of inter-class fractions of Hamming distance.

The SRAM protocol is analyzed in terms of costs in 128-bit keys. The secrecy rate is measured using Context-Tree Weighting Method (Willems et al. 1995) and resulted in an average secrecy rate of 0.76 bit per SRAM bit. This means, in order to derive an N-bit key at least [1.32N] bits are required from memory. The error correction performance is analyzed by the number of bits of information required. BCH Error correction code is used in this implementation. The probability of error bits is assumed to be 0.06. This scheme requires 1023 bits of SRAM for at least 102 bits of errors to generate 278 bits of information.

The overall SRAM provides good cryptographic assumptions. However, further implementations are required to test the practicality of fuzzy extractor. The hash functions impose heavy computation for larger key sizes. The key generation process solely relies on the architecture of FPGA and the noise assumptions are relying on the power-up operation at manufacturing. This raises some concerns in terms of application and scalability.

2.3.4. Butterfly PUF

It is understandable that Modular IP has an economic advantage. However, IP could be leaked to unauthorized parties and the license could be overused. Although the key is generated on volatile memory bitstream is stored in the non-volatile memory. There are ways to encrypt the bitstream, but the previously proposed methods require keys to be stored on external memory or require a continuous power supply. Hence, the earlier solutions bring high costs for production. SRAM-based PUF circuits rely on positive feedback supported states.

Intrinsic SRAM PUFs also come with the disadvantage of limited applications. Uninitialized SRAM memory is not supported by all FPGAs. Supporting FPGAs are set to a known state during startup. There are many types of FPGAs that do not allow this state change.

This method also relies on the idea of cross-coupled circuits of SRAM cells. However, Butterfly PUFs (Kumar et al. 2008) implement separate structures in FPGA matrix that act like SRAM cells. The cross-coupled circuits include inverters that have an unstable operating point and two stable operating points. The initial state is an unstable state. The state changes to higher or lower voltage values during small changes in the delays and component characteristics. These circuits require symmetry to get a higher entropy in the generated bits.

The additional cross-coupled logic circuit within the FPGA is not a straightforward method so the combinational loops do not exist. Nevertheless, these can be implemented

by simulating the loops with latch components. The latches are illustrated in Figure 2.28.

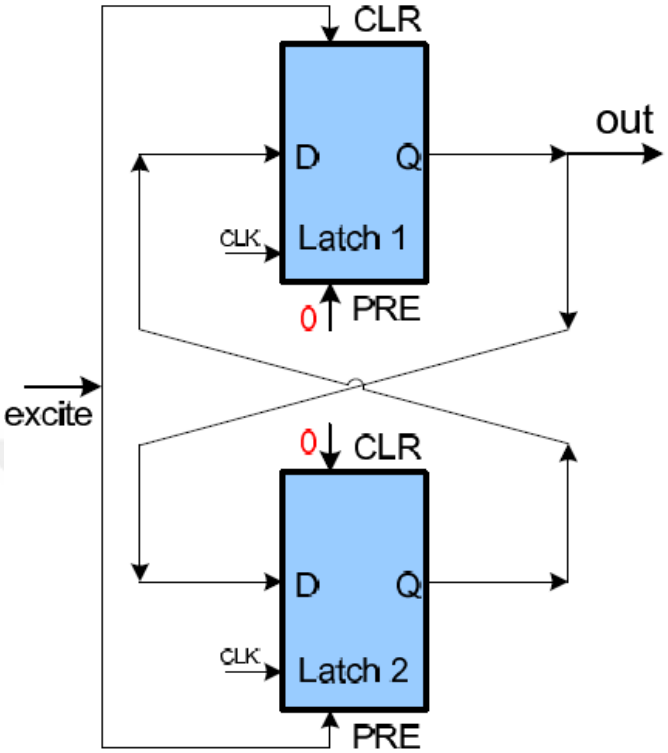


Figure 2.28. The Cross-coupled latches (Kumar et al. 2008)

The latches are fed with high input voltage on clocks when the circuits are initialized to set on unstable operating points. Latch locations, routing of the wires, temperature variations can trigger unique secret values. These latches are not visible and they are hard to find in the circuit. The attacker cannot acquire the secret by observing the circuit.

The experiments show that in the temperature of $20°C$, the within-class fractional Hamming distance results in 6% and the between-class Hamming distance results in a mean close to 50%. The experiments are carried out between -$20°C$ and $80°C$ temperature range, 50MHz, and 120 MHz frequency range, and FPGA core voltages. These changes do not affect the distribution of unique values. The 128-bit key requires 1500 Butterfly PUF cells with an error bit rate of 10^{-6}.

2.3.5. Loop PUF

Previously presented schemes(RO-PUF, Butterfly PUF, Arbiter PUF, SRAM PUF, etc) provide a lightweight implementation of strong privacy with the property of robustness. However, they are vulnerable to invasive and semi-invasive electromagnetic attacks due to their placement.

Cherif et al. (2012) proposed the Loop PUF(LPUF) and is based on a single ring

oscillator. Multiple variations are compared in a sequential fashion. The structure of LPUF is flexible and easy to implement. There is no distinct routing limitation. The LPUF is illustrated in Figure 2.29. Each loop is composed of two inverters and a multiplexer. A sequence of these loops constructs a delay chain and the multiplexers of each loop is controlled by "control words". These control words are defined as the Challenge of this PUF architecture. The number of challenge bits is also the number of delay chains which are N bits. The resulting frequency or delay measurement at the output of serially placed delay chains is fed to the controller unit. The controller unit returns the N-bit Response key.

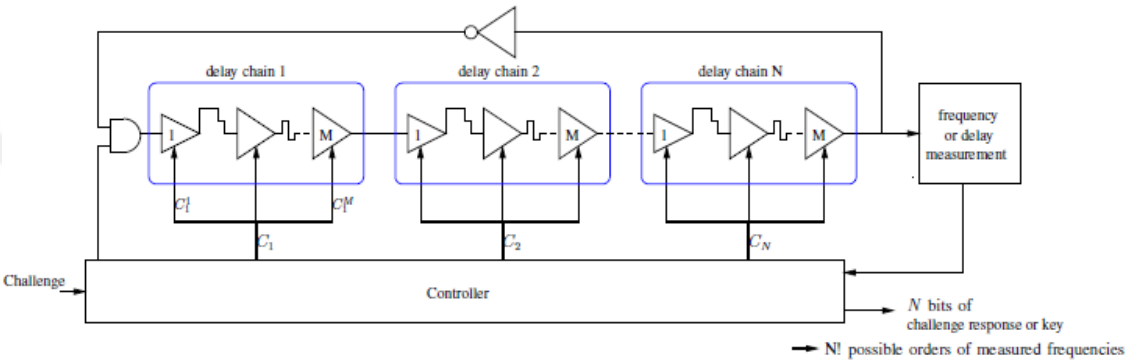


Figure 2.29. The Loop PUF structure (Cherif et al. 2012)

The resulting key of LPUF is directly controlled by the frequency variations. The controller generates a combination of control words to perform a pairwise operation. The delay differences of each loop are calculated among these pairs. For each iteration of the control word, a maximum Hamming distance value is calculated. The ideal number of challenges is decided by the given equation:

$$\forall j \in [1, M] \prod_{i=1}^{N} C_i^j = 0$$

A control example with *N= 3* words is shown in Figure 2.30. In order to produce a 64-bit response with three rotations, 26 challenges are required. This method as previously indicated provides a good level of flexibility. The PUF ID is also reliable with a given measurement time that is properly high and an Elliptic Curve Cryptography (ECC) is implemented. Inter-chip variation with 15-bit ID produces an average 7.51% fractional Hamming distance. The standard deviation is 60.8 kHz with the measurement time window of 250microseconds at $20°C$ and at the nominal power supply.

LPUF is evaluated by means of randomness, uniqueness, and reliability. The results are compared to Arbiter PUF. Randomness in LPUF is statistically perfect since the response completely depends on the delay difference. The intra-class variations are referred

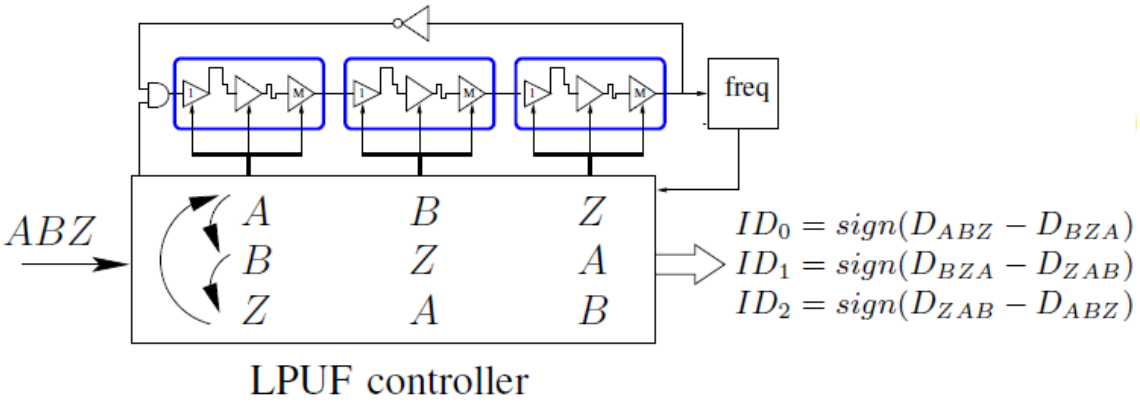


Figure 2.30. An example of LPUF control (Cherif et al. 2012)

to as uniqueness and it shows 95% of fractional Hamming distance. Reliability is dependent on the steadiness of the structure which results in 98.7% of fractional Hamming distance.

In order to improve the speed and reliability, one can use ECC supported by fuzzy extractors. However one needs to choose between the ECC complexity or LPUF latency in terms of performance. The robustness is evaluated in the presence of electromagnetic attacks. Authors propose the use of transformation through non-linear functions. Substitution boxes similar to the AES block cipher algorithm can be utilized. However, Substitution Box of AES is complex for the RFID applications. PUF schemes require low-cost and lightweight methods. Also, side-channel attacks can be thwarted with the use of true random number generators(TRNGs).

2.3.6. TERO PUF

There have been many silicon-based PUF schemes. Those schemes rely on the ring-oscillators and cross-coupled methods have shown the best statistical results to this day. RO-PUF has the advantage of being scalable. However, the locking phenomenon arises from the slight dependent structure of ring oscillators. Locking phenomenon delivers the risk of fault injection and other electromagnetic attacks. This risk also requires the frequency values to be obfuscated.

The locking phenomenon can be avoided with oscillatory metastability components. Observation of oscillatory metastability improves the entropy in the PUF results through the extraction of statistical values of oscillation parameters. The Transient Effect Ring Oscillator, TERO-PUF (Bossuet et al. 2013) relies on an oscillatory mechanism to extract both PUF at the manufacturing process and true random number generators(TRNGs) with a single set of hardware.

Each of the TERO-PUF loop circuits is implemented with SR flip-flops which consist of two AND gates and inverters (see Figure 2.31). These circuits are based on positive feedback and RC time constant. Oscillations should continue forever in an ideal sym-

metrical setting. However, the physical conditions of the components bring some slight asymmetry and this asymmetry stops the oscillation in a relatively short amount of time. The SR flip-flop loop remains in oscillatory metastability state with AND gates and inverters.

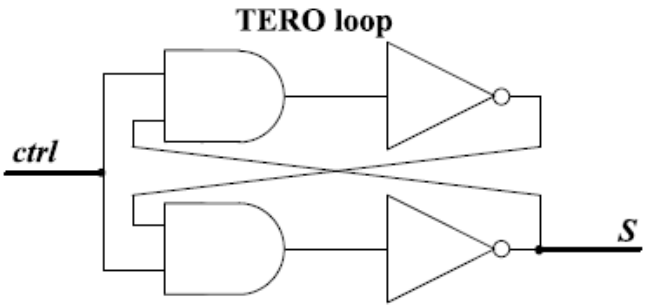


Figure 2.31. A TERO Loop Circuit (Bossuet et al. 2013)

This scheme is tested with nine Altera Cyclone II EP2C20 FPGAs. Each FPGA contains 1172 TERO loops. The observed oscillations measured with 8-bit counters are used as the source of entropy because independent values affect the normal distribution. However, the final state of the output signal of SR flip-flop is not reliable. The last significant bits(LSB) of the 8-bit counters are not stable so they can only be applied on TRNG while the most significant bits(MSB) are stable. The MSBs are used as PUF response.

The entire TERO-PUF architecture has a different structure which allows the noise to be reduced and PUF characterization to be unique. The architecture is illustrated in Figure 2.32. The set of SR flip-flops are fed to 8-bit counters and using 26-bit accumulator and 18-bit shift register the mean of oscillations are calculated. Each loop is differentiated by pairs and their result provide a random bit.

The resulting bits of the TERO-PUF architecture is analyzed by the means of bias, intra-device variation, and inter-device variation. The output bits are selected according to these statistical properties. Their bias and inter-device variation should be close to 50% and intra-device variation should be close to 0%. However, there will be a slight fraction of Hamming distance in intra-device variation. This fraction requires additional measurements for error correction.

The results of the experiments show that TERO-PUF has a Hamming distance of 1.7% in intra-device variation and a Hamming distance of 48% in inter-device variations for 126 bits of PUF ID. The resulting number of bits shows the reliability performance of this scheme.

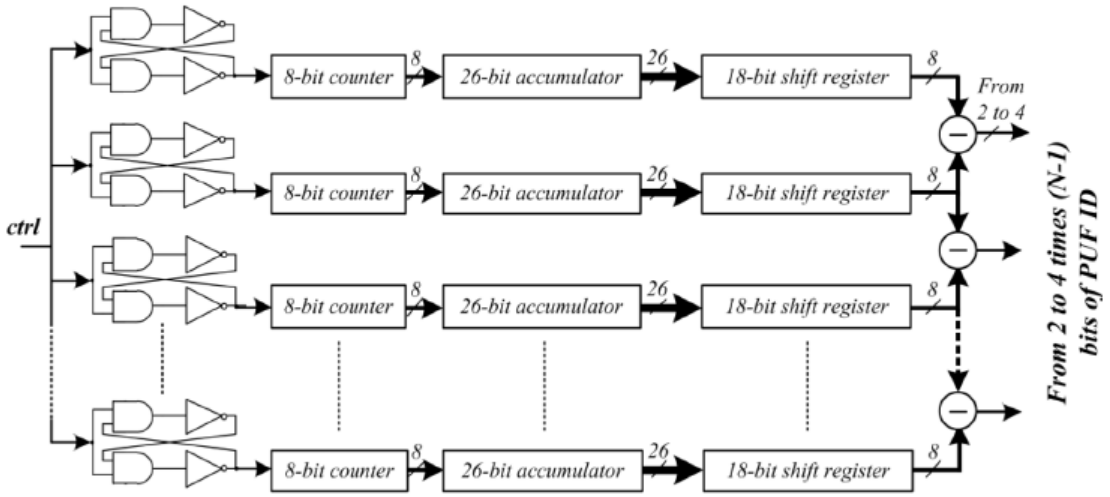


Figure 2.32. A TERO-PUF Architecture (Bossuet et al. 2013)

2.3.7. A Lightweight Mutual Authentication Protocol Based on PUF

Xu et al. (2018) designed a solution against the counterfeit goods that cause financial and security threats. RFID tags are generally prone to forgery, mutual verification, data anonymity, steal, replay, clone, backtracking, and desynchronization attacks.

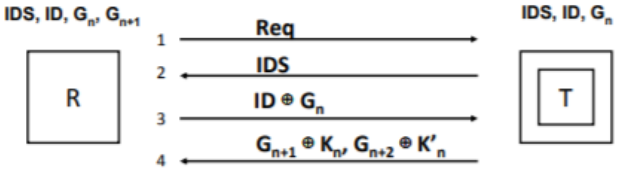


Figure 2.33. The Kulseng's communication algorithm (Kulseng et al. 2010)

Since RFID technology has a lot of positive features in IoT environments, it is important to search for efficient methods that protect the systems against the aforementioned attacks.

This work implements PUF based random number generation with additional temporal and updated secret keys.

Many of the previous works rely on a secure key that is used between the client and a central unit through one-way hash functions. However, most of those works are not efficient with the limited resources of the IoT frameworks as many of the hash functions require intensive computations.

Kulseng et al. (2010) proposed a mutual verification protocol provides the most effective authentication through PUF mechanism. It is a low-cost solution, nonetheless, it still indicates some security vulnerabilities. Kulseng's algorithm is given in Figure 2.33.

In the proposed protocol, RFID stores five main security values: The secret value of the tag generated by PUF through (P_n), the index of the tag in the database ID (IDS), the false ID of tag (FID), the shared secret key that is used between the tag and the reader (K_n) and the greeting number of each round (G_n). After each verification, these values are updated. Tags store a set of secret key values from each of the last update and the most recent update. The authentication process is shown in Figure 2.34. The verification of the

secret values operates in three phases. Firstly, the tag recognition takes place. The reader sends a search request to the tag. The tag generates a random number and transmits it to the reader with the FID value. The receiver checks the authenticity of the FID value in its database.

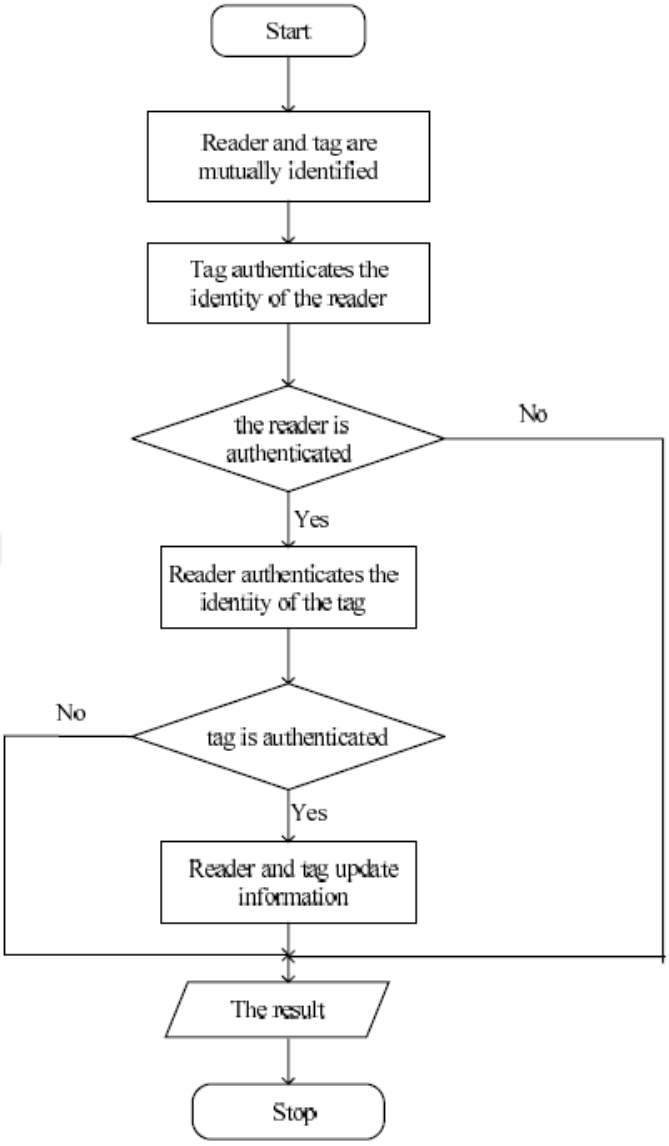


Figure 2.34. The tag verification process (Xu et al. 2018)

Secondly, mutual verification is performed. This phase authenticates both the tag and the reader. The reader generates a random number and transmits it to the tag with its FID and their shared key with the tag. It produces a string of data containing its own set of secret values. Then the reader compares the final values it computes and receives from the tag.

Lastly, the sets of secret values are updated. The update takes place either the host database is updated or the host database indicates that the tag should update its new FID value. In these cases, the FID values of the tag are updated. The update operation takes place after tag verification.

The values are stored in the open source HBase technology which is a distributed and

column-oriented database system. HBase efficiently stores millions of columns and raws and can store multiple versions of all data.

All the attacks specified earlier are covered in this protocol. The update mechanism provides freshness property. This also means, even if the adversary acquires a set of secret key through these attacks, it will not be able to use these keys on another set of operations. Furthermore, as the PUF relies on the biometrics of the chipset it is running on at a given time, it is infeasible to generate the same value.

The experiments take place in ultrahigh frequency reader with air interface protocol and autopilot technique of Speedway Revolution. The experiment is analyzed in time and space complexities. Although the protocol is more complex than Kulseng's algorithm, it provides higher security with reduced computational overhead.

2.3.8. PUF-based Reliable Biometric Access Control for IoT

Karimian et al.(2018) proposed an IoT security backbone based on biometric keys. However, the authors know how nontrivial is to store, process, and manage detailed large natural biometric data continuously they implement a system to provide less computationally intensive operations.

Biometric Access Control by Karimian et al. also reduces the risk of compromised biometric information. The core idea is implementing hardware obfuscation supported with PUF nature.

While obfuscation provides a non-intensive unobvious data processing, PUF provides non-linkability and non-invertibility through one-way hash functions.

One of the biggest concerns of raw biometric data is noise. A noise aware system allows more reliable keys. The general biometric authentication systems deal with five major concerns; sensing, feature extraction, template storage, matcher, and decision making. Here also different types of failures in the systems are considered.

Karimian et al. (2018) presented Biometric Locking by Obfuscation, Physically Unclonable Keys, and Reconfigurability (BLOcKeR) which provides hardware personalization with reconfigurable nature. This nature is obtained through the utilization of FPGA. PUF is also considered to be biometric for hardware. BLOcKeR process is illustrated in Figure 2.35.

The hardware obfuscation is designed to operate in three steps. Logical encryption, Logical permutation, and locking through Finite State Machine. Bitstream obfuscation is used because of low overhead and applicability on FPGA boards. It is also more flexible and scalable compared to other obfuscation methods. Strong PUF is used. Even at an instance of an adversary to have access to PUF for a long period, it is still infeasible to guess Challenge-Response Pair.

The enrollment process consists of hardware enrollment where a strong PUF model is

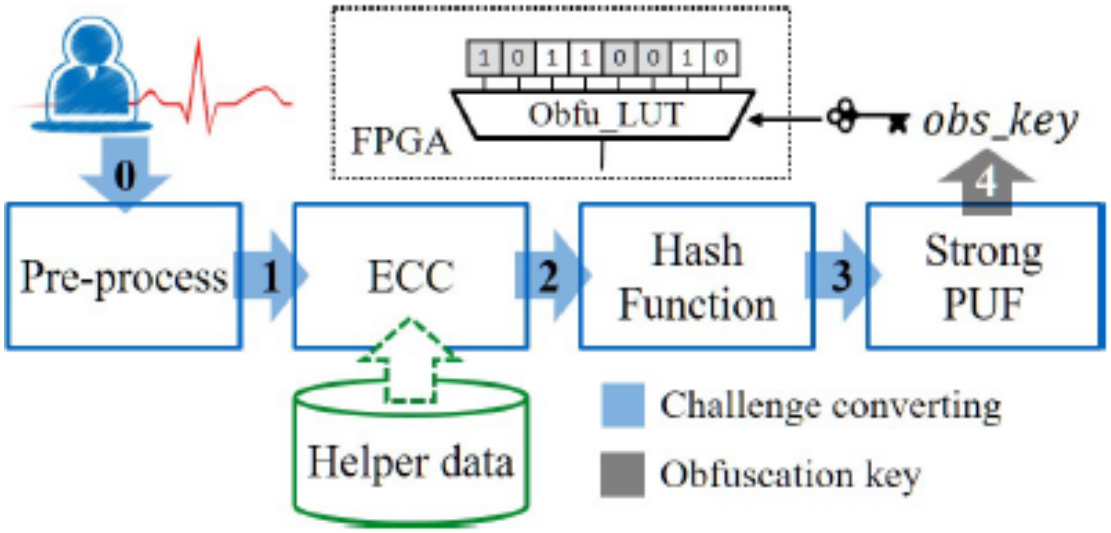


Figure 2.35. The process flow of BLOcKeR (Karimian et al. 2018)

built and its firmware is discarded, ownership claim where an obfuscation key is generated through preprocessing algorithm and transmitted to the designer through a secure channel, and firmware customization where the biometric key is generated. Series of attack analyses tested for non-invasive, semi-invasive, and invasive attacks.

There are several distinctive steps in IOMBA. Firstly, data preprocessing eliminates the noise and normalize the feature elements in the standard normal distribution before feature extraction. Secondly, each feature is quantized at variable lengths to calculate margins. Thirdly, key enrollment and helper data generations are followed. Helper data keeps indices for reliable features, data lengths, and parameters to normalize the quantized features. Lastly, there is a key generation.

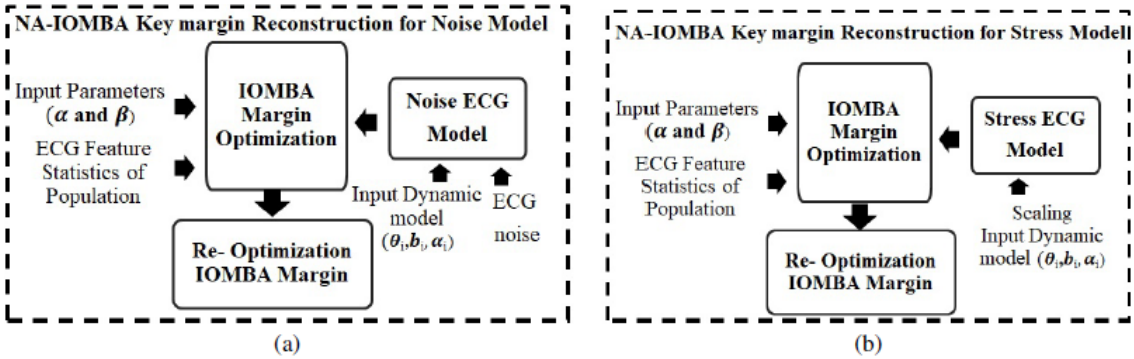


Figure 2.36. The NA-IOMBA margin reconstruction (Karimian et al. 2018)

As IOMBA system is impractical on IoT grounds, noise aware version is considered. It

has positive various impacts on key lengths, key reliability, cost, and time. Karimian et al. considers three main case studies. Firstly, a comparison of different modalities of signals such as ECG, PPG, iris, and fingerprint. Under reliability criteria, ECG brings the best performance. Fingerprint has the best entropy and PPG works efficiently at the smallest key lengths. However, ECG and PPG have the lowest costs. Secondly, the denoising of overhead reduction is considered under three steps. ECG synthetic model generation from the real ECG database, noise modeling for ECG from different sources such as muscle artifacts, baseline wander for body movement, and electrode movement due to poor contact to the sensor, and lastly FPGA implementation of IOMBA. Lastly, the reduction of enrollment times is performed. The NA-IOMBA margin reconstruction key algorithm for noisy ECG and stressed ECG is illustrated in Figure 2.36.

2.4. Lightweight Cryptography

The lightweight cryptography became prominent due to the development of wireless sensor networks and wireless body area networks. This field defines the general metrics to utilize encryption mechanisms in resource-constrained. The lightweight encryption algorithms are separated mainly into block ciphers, hash functions, more complex high-performance systems, stream ciphers, and dedicated low-resource devices.

Despite the fact block ciphers are the most popular and effective cipher, stream ciphers have been widely used in RFID authentication systems. There have been ongoing projects including eSTREAM(2008) to support the developments of stream ciphers and providing standards for them. They are known for their general compact structure and show relatively higher performance with smaller hardware implementations compared to widely known block ciphers. Nevertheless, block ciphers can be useful for their inherent ease of implementation and expressive nature. Block ciphers can be flexible and they can be converted to stream cipher mode with the correct mode of operations. When block ciphers are designed with consideration of better resource constraints they can be useful in RFID technologies as well. One of the main drawbacks of block ciphers is their structure of repetition and the attackers seek for this repetition. Repetition should be designed well enough to provide the avalanche effect.

The common structures in lightweight ciphers are key and block sizes, linear and non-linear operation, and key generation. The most critical issues are power and energy consumptions since the smallest elements in the environment require lightweight cryptographic operations. These elements can harvest energy from external sources. Furthermore, in wireless body area networks (WBANs), the frequent replacement of batteries or recharging them could be infeasible. Hence, it is also important to decide on the level of security requirements in order to balance privacy and power consumption. The security levels are commonly modeled over the potential breaches induced by known attacks and the sensitivity of the information. The designer must define the sensitivity of information and correctness of exchanged data. Although WSNs and WBANs are mentioned in the same context, in this section the designated ciphers are not suitable for both networks.

The lightweight design space is finite yet dynamic. Therefore, the classification should be scalable and well generalized as it is indicated in Bossuet et al. (2013). They mainly

separate the systems into three categories such as high-performance systems, the general purpose processor systems, and low-resource devices.

A high-performance system priorities throughput, flexibility, and security with strong importance. Such systems regard costs from power and design components below these three requirements. There are customized CPUs integrated with a crypto ALU. They optimize the logical operations during encryption with the aid of instruction set architecture (ISA). The crypto processors which are similar to DSP modules are integrated as an extension to the main system. There are also crypto co-processors which provide cryptographic operations which are triggered by the main processor. However, they induce communication overhead. Additionally, there are crypto arrays which introduce parallelism and multi-core crypto processors for simultaneous high encryption schemes.

General purpose systems do not have dedicated resources or components for cryptographic operations. They run machine-independent platforms which provide ease of implementation and analysis for speed of cryptographic operations.

The low-resource devices split into software and hardware platforms. Software-oriented implementations utilize small low-cost processors. They can be machine-dependent or -independent. Software implementations aim to minimize memory usage, implementation speed, and energy efficiency. The hardware-oriented platforms are either full-custom, ASIC, or FPGA. These provide ease of implementation and flexibility. FPGAs also provide low development cost, agility, maintainability, and flexibility. These hardware implementations intent to optimize speed and low power dissipation. The cycle count and memory usage are not a big concern in hardware implementations.

There is no systematic or uniform platform for hardware implementations. It is also indicated that there is no perfectly accurate security metric existing for both. The software implementations are mostly based on coding techniques. On the other hand, the hardware implementations are heavily based on logic gates, physical architecture, and high-level and physical supporting tools. Firstly, during the design process, the requirement specification is done regarding security goals and computational constraints. For the software-oriented implementations with the aid of development tools, simulators, designated coding style, code partition, and appropriate data structures, the designed algorithm is implemented. The hardware-oriented implementations require appropriate physical platforms, technology node, logic library, and a defined architecture.

The following sections are detailed descriptions about lightweight stream and block ciphers. Their early properties are suitable for IoT systems. And the last section presents a brief summary on other categories based on Singh et al. (2017) survey.

2.4.1. The MICKEY Stream Cipher Family

The MICKEY is a stream cipher family (Babbage and Dodd 2008) that is designed to conform to 'Profile 2 ' in EUROCRYPT 'Call for Stream Cipher Primitives' in 2005, which is aiming to show stream ciphers can significantly perform well in resource-constrained physical environments under this category. The name is an abbreviation for

Mutual Irregular Clocking KEYstream generator.

The scheme is published in two key sizes; 80- and 128-bits. It is inspired by the applications based on Fibonacci-clocked jumping LFSR from the work of Jansen(2004). However, it is implemented with Galois-stepping LFSR. The system is based on two registers R and S. R is clocked with Galois-stepping LFSR linearly and S is clocked with random initial values and stepped with a non-linear sequence. The register R provides statistical properties and a good choice of period for jumping. On the other hand, the register S would provide security primitives against adversaries those exploit the linearity of R.

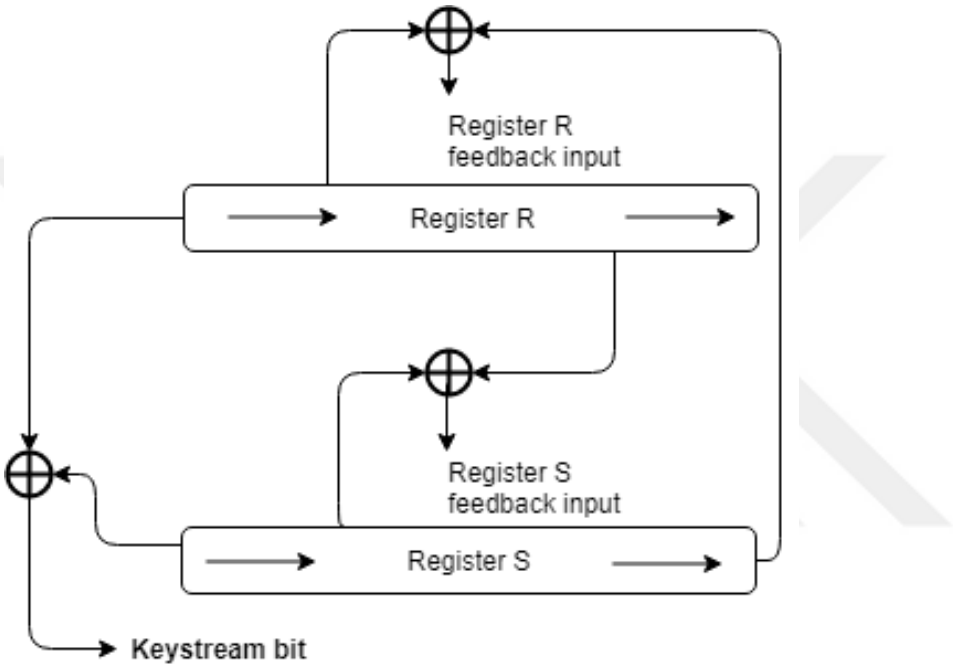


Figure 2.37. The control diagram of MICKEY (Babbage and Dodd 2008)

The two of the shift registers are applied to the exclusive-OR (XOR) operation to get their control feedback data in the result as shown in the Figure 2.37. The values of the registers are also XOR'ed in a way to produce keystream value. The keystreams are at $2^{|K|/2}$-bits at a maximum where K denotes the key size. E.g. for 80-bits key size, the maximum length of a keystream sequence is 2^{40}-bits. Each register runs in 100 stages. The register R has a set of feedback tap positions. The last stage is XOR'ed with the input bit and fed as feedback bit. The register S has a sequence of control and feedback bits for each binary state at every stage. These values are retrieved from the table in the Figure 2.38. This table provides non-linearity to the register S.

Variable clocking provides 'jump' for the register R. As it is declared earlier, the authors adopted the jumping technique on Fibonacci-clocked LFSRs of Jansen(2004) into Galois-clocked LFSR. The variable clocking caused security vulnerabilities in the past as in LILI-128 (Dawson et al. 2000). However, jump property prevents these by forming the

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$COMP0_i$		0	0	0	1	1	0	0	0	1	0	1	1	1	1	0	1	0	0	1
$COMP1_i$		1	0	1	1	0	0	1	0	1	1	1	1	0	0	1	0	1	0	0
$FB0_i$	1	1	1	1	0	1	0	1	1	1	1	1	1	1	1	0	0	1	0	1
$FB1_i$	1	1	1	0	1	1	1	0	0	0	0	1	1	1	0	1	0	0	1	1

i	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
$COMP0_i$	0	1	0	1	0	1	0	1	0	1	1	0	1	0	0	1	0	0	0	0
$COMP1_i$	0	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0	0	1	1	0
$FB0_i$	1	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	1
$FB1_i$	0	0	0	1	0	0	1	1	0	0	1	0	1	1	0	0	0	1	1	0

i	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
$COMP0_i$	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	1
$COMP1_i$	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0
$FB0_i$	1	1	0	0	1	0	0	1	0	1	0	1	0	0	1	0	1	1	1	1
$FB1_i$	0	0	0	0	1	1	0	1	1	0	0	0	1	0	0	0	1	0	0	1

i	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
$COMP0_i$	1	1	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	0	1
$COMP1_i$	0	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0
$FB0_i$	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0
$FB1_i$	0	0	1	0	1	1	0	1	0	1	0	0	1	0	1	0	0	0	1	1

i	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
$COMP0_i$	0	1	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	1	1	
$COMP1_i$	0	1	0	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	0	
$FB0_i$	0	0	1	1	0	1	1	1	0	0	1	1	1	0	0	1	1	0	0	0
$FB1_i$	1	1	0	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0	1

Figure 2.38. The S register table of MICKEY (Babbage and Dodd 2008)

register R as an "engine" in the system that ensures in a keystream sequence that there is no repeat.

The clock control bits are derived in a way in order to make guess-and-determine and divide-and-conquer attacks infeasible. The clocking function of the register S is non-linear and irreducible. The sequence in the register S is designed to avoid any strong affine relations. This allows for any linear inputs affecting the keystream to originate from the register R. The register S design provides local randomness in case of when the initial state is uniform.

Hong and Kim (2005) provided cryptanalysis against the earlier version of MICKEY. They cover the vulnerabilities under three sections; time-memory-data(TMD) trade-off, state entropy loss and keystream convergence, and the existence of weak keys. Firstly, the register sizes have been changed from the key size to 1.25 times of the key size. The TMD attacks can be seen effective, although precomputation time is often disregarded. These attacks are not better than exhaustive key search attacks. The authors provide an additional caution against Biryukov-Shamir TMD (Biryukov and Shamir 2000) attacks by

increasing the state size significantly. It is pointed out by Hong and Kim that some states have multiple or no preimages after clocking which brings concerns about collisions. However, since the control bits are derived from the internal states, not the preimages, the entropy loss in that area is not a concern. Finally, weak keys are very small in numbers. Therefore, they could be easily avoided. It is more feasible for an adversary to try those keys than devising an attack. The authors point out that the scheme is not practical for side channel attacks. However, they regard these attacks are not applicable in applications of stream ciphers physically.

The cipher was meant to run on low-resource hardware. Algorithm efficiency was not the concern of authors. Nonetheless, it shows good implementation and operational performance. MICKEY does not support pipelining due to its variable clocking structure.

2.4.2. CLEFIA

The block ciphers are designed to reduce the cost of implementation during the encryption of large plaintext. It has been previously denoted that the cryptographers mainly aim to reuse the components in the design. Reuse of the components usually happens during the intermediate steps of encryption and decryption. However, since the attackers know this fact, they focus on statistical and algebraic attacks to exploit these intermediate operations. The recent techniques in block cipher emphasize on the importance of immunity towards these attacks. Techniques with high performance for RFID technologies with limited resources have been widely discussed. However, these designs should be implemented with a guarantee regarding the intrinsic disadvantages. FOX and HIGHT techniques are developed with the consideration of costs of implementation and operation.

Shirai et al. (2007) proposed the CLEFIA in order to provide immunity against known attacks for block ciphers and flexibility of implementation. The performance of this method is evaluated in software and hardware levels. It defines a mechanism for both encryption and decryption with a key scheduling technique for round key generation. The design is considered with three key sizes; 128-, 192-, and 256-bit.

The defined architecture uses a Type-2 generalized Feistel network with 4 or 8 branches and a given minimum number of rounds. In order to decrypt the ciphertext, the inverse of the function is achieved simply by changing the directions of round keys.

Encryption and decryption operations in CLEFIA are defined as data processing part. The operations are obtained by adding the whitening keys and round keys in the generalized Feistel network(GFN) functions. The number of rounds is defined as 18, 22, and 26 for 128-, 192-, and 256-bit keys respectively (see Figure 2.39).

Two different F-functions are defined based on two separate S-boxes. Their difference is important in order to create algebraic immunity. These F-functions are processed in the intermediate steps of the generalized Feistel network functions. They use S-boxes to achieve linearity. The input is split into four parts and each part is reassigned to their respective values from S-boxes. The final value to be used in GFNs is obtained from the dot product of the concatenation of parts resulted from S-boxes with 4x4 Hadamard-type

Step 1. $T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow X_0 \mid X_1 \mid X_2 \mid X_3$
Step 2. For $i = 0$ to $r - 1$ do the following:
Step 2.1 $T_1 \leftarrow T_1 \oplus F_0(RK_{2i}, T_0), \quad T_3 \leftarrow T_3 \oplus F_1(RK_{2i+1}, T_2)$
Step 2.2 $T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow T_1 \mid T_2 \mid T_3 \mid T_0$
Step 3. $Y_0 \mid Y_1 \mid Y_2 \mid Y_3 \leftarrow T_3 \mid T_0 \mid T_1 \mid T_2$

a) Defining $GFN_{4,r}$

Step 1. $T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow P_0 \mid (P_1 \oplus WK_0) \mid P_2 \mid (P_3 \oplus WK_1)$
Step 2. $T_0 \mid T_1 \mid T_2 \mid T_3 \leftarrow GFN_{4,r}(RK_0, \ldots, RK_{2r-1}, T_0, T_1, T_2, T_3)$
Step 3. $C_0 \mid C_1 \mid C_2 \mid C_3 \leftarrow T_0 \mid (T_1 \oplus WK_2) \mid T_2 \mid (T_3 \oplus WK_3)$

b) Data Processing

$X_{(128)} \mapsto Y_{(128)}$
$Y = X[7-63] \mid X[121-127] \mid X[0-6] \mid X[64-120]$

c) DoubleSwap Function

(Generating L from K)
Step 1. $L \leftarrow GFN_{4,12}(CON_0^{(128)}, \ldots, CON_{23}^{(128)}, K_0, \ldots, K_3)$

(Expanding K and L)
Step 2. $WK_0|WK_1|WK_2|WK_3 \leftarrow K$
Step 3. For $i = 0$ to 8 do the following:
$T \leftarrow L \oplus (CON_{24+4i}^{(128)} \mid CON_{24+4i+1}^{(128)} \mid CON_{24+4i+2}^{(128)} \mid CON_{24+4i+3}^{(128)})$
$L \leftarrow \Sigma(L)$
if i is odd: $T \leftarrow T \oplus K$
$RK_{4i}|RK_{4i+1}|RK_{4i+2}|RK_{4i+3} \leftarrow T$

d) 128-bit Key Scheduling with DoubleSwap

Figure 2.39. The main operations in the CLEFIA (Shirai et al. 2007)

matrices. Two of the F-Functions have a different Hadamard matrix (see Figure 2.40a and Figure 2.40b).

Key scheduling part generates whitening and round keys for the data processing part. The DoubleSwap (Figure 2.39c) function takes the key and splits them into four parts. The output is the rearrangement of these four parts. The intermediate keys are acquired by applying the initial key and Constant Values as round keys. The number of constant values is 60, 84, and 92 for 128-, 192-, and 256-bit keys respectively. They are acquired by using the base of the natural logarithm, e and the circle ratio, pi as constants and

their rotation. The final multiplication is performed in a GF(2^{16}) primitive polynomial. After the generation of the intermediate key, it is expanded along with the initial key. The DoubleSwap function is used during the iteration of eight parts of the initial key. In the expansion, the rotating keys and whitening keys are generated (see Figure 2.39d).

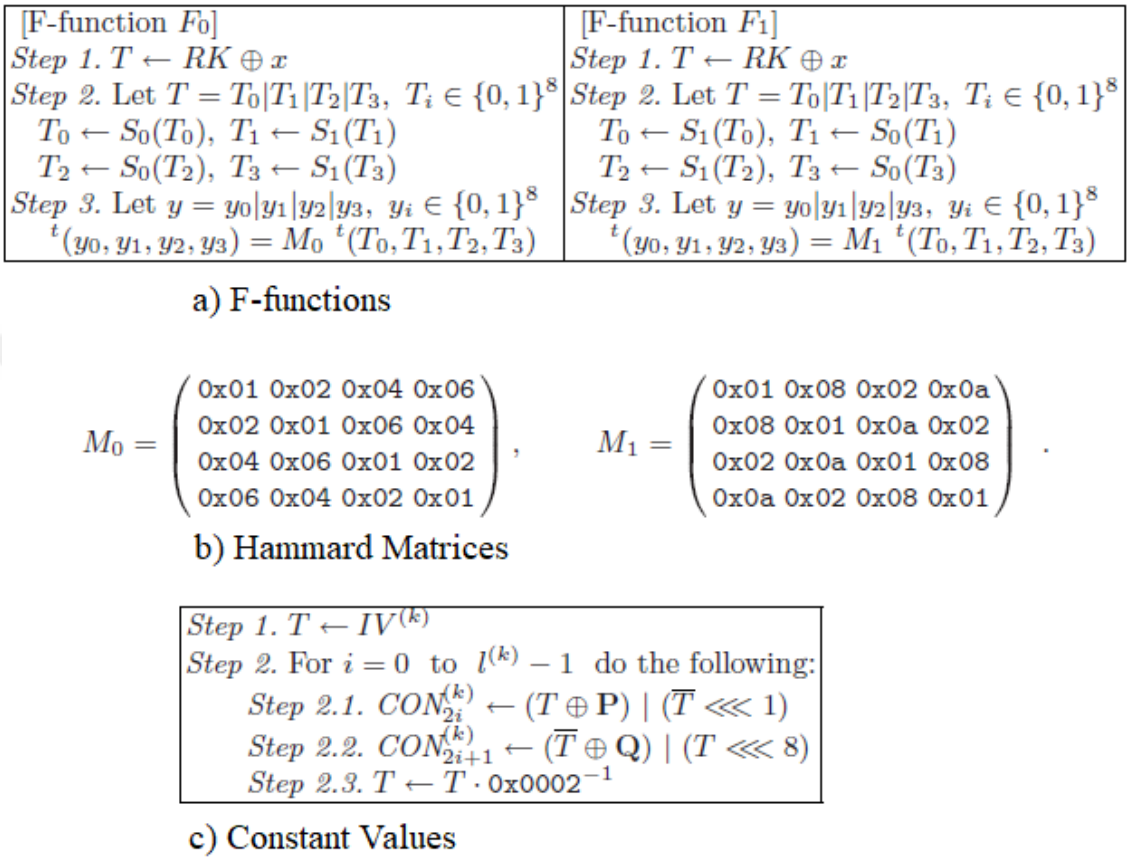


Figure 2.40. Important components in the CLEFIA (Shirai et al. 2007)

CLEFIA employs Diffusion Switching Mechanism(DSM) in order to prevent difference cancellations with the neighborhood rounds which provides immunity against statistical attacks. DSM is applied to CLEFIA with more than three rounds. Hence, it provides a higher number of active S-boxes against differential and linear attacks.

Shirai et al. note several additional design aspects to increase efficiency. Small sizes of F-functions and elimination of their inverse functions increase the quality of GFNs. Reducing the number of rounds is important to improve DSM performance. Small S-boxes could be used in order to decrease memory usage. Matrices should contain elements with low hamming weights. Key schedules could be improved by using the same structures with the data processing part, small key registers, and small footprints.

The security analysis is performed against mainly Differential, Linear, Impossible Differential, Saturation, Algebraic, and Relative key attacks. These attacks could be rather

infeasible with small improvements. The performance has experimented with Athlon 64 (AMD64) 4000+ 2.4Ghz processor running on Windows XP 64-bit edition. Single-block implementation resulted in 12.9 cycles/byte in encryption and 13.3 cycles/byte in decryption. Key setup only requires 217 cycles for 128-bit. 192- and 256-bit key size tests are made. Authors note that with the CTR mode the scheme is suitable for double-block implementation in parallel. They show close performance results against AES implementation in single-block.

2.4.3. PRESENT

The winning Rijndael algorithm of AES was a successful block cipher with considerations of low-cost implementation. However, the considerations were towards software efficiency. The implementation required 3600 gate equivalents (GE). Later Tiny Encryption Algorithm(TEA) and its variant XTEA were developed with 2100GE and 2000GE. Other low-cost block cipher variants have been mCrypton, HIGHT, and SEA according to the authors. SEA has been the closest one to the PRESENT (Bogdanov et al. 2007).

The architecture of the block cipher proposed by Bogdanov et al. consists of exclusive-or(XOR) operations and diffusion layers. The overall algorithm can be seen in Figure 2.41. At each layer iteration, a round-key is updated and diffused with the input block. The diffusion layer defines the substitution-permutation network(SP-network) structure of this scheme (see Figure 2.42). Here, round-keys are mainly used for key whitening. Substitution box(S-box) of the diffusion layer provides non-linearity before applying the linear permutation. It is important to design sound S-box. The key schedule here constructed in terms of 80-bits key while it is also evaluated with 128-bits key. The keys are user entered and stored in key registers. The round-keys are produced through a rotation, application of S-box, and the round_counter component.

Nonlinear S-box is the main source of avalanche change and it turns the solution into NP-Hard problem for possible adversaries. Nonetheless, the S-box needs to conform to nonlinearity properties in order to be secure. Furthermore, the S-box in this scheme is considered in 4-bit to 4-bit size. Furthermore, this S-box size is reasonable, since the higher and lower values would risk the compactness and security. S-box defined in this work can resist differential and linear attacks.

PRESENT is implemented with the intention of security, efficient hardware implementation, and simplicity. PRESENT is mainly based on encryption and decryption operations, though the authors declare that considering encryption-only PRESENT with subkeys generated on-the-fly would make it ultra-lightweight solution.

The work is evaluated in terms of differential, linear, integral, bottleneck, truncated differential analysis, and algebraic attacks. When PRESENT is implemented with reasonable key size, a number of rounds, and S-box size its security and efficiency show good results. Experiments use 32 clock cycles to encrypt 64-bits of plaintext with 80-bit keys with 16 S-boxes, 1.8 Volt core voltage, $25°C$ temperature, and $0.18\mu m$, and 1570GE.

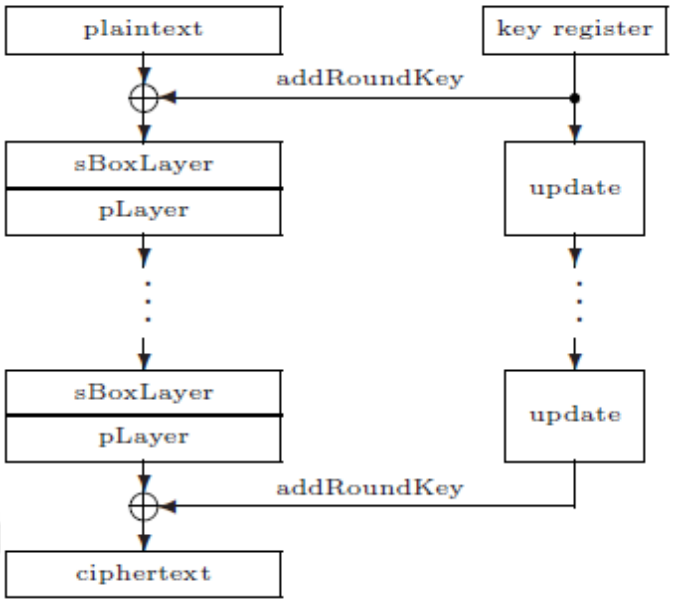


Figure 2.41. General Structure of the PRESENT (Bogdanov et al. 2007)

2.4.4. HC-128

eSTREAM is a project that aims to identify stream ciphers that could have an advantage over block ciphers. As it is indicated earlier, they are compared to the Rijndael algorithm which is the finalist of the AES project. The submissions required to fall into at least one profile out of two. One of the profiles required exceptionally high throughput in software applications and the other required an exceptionally low cost in hardware applications.

Hongjun (2008) proposed the HC-128 in order to provide high throughput for software applications by providing the use of only strong keys during the encryption process. It is a free open source project and the simplified 128-bit version of HC-256.

The HC-128 is composed of two secret tables containing 512 elements with 32-bit size. During 1024 steps all of the elements in the tables are updated. Those updates are supported by non-linear feedback functions. Compared to HC-256, HC-128 aims implementation over new generation superscalar microprocessors with its three separate operations with little dependency. Therefore, two of its independent operations are suitable for parallel computations.

HC-128 focuses on making exhaustive key search and distinguishing attacks as infeasible as possible. It provides extremely high lower bounds on outputs required for the adversaries to be successful.

According to Hongjun, the 128-bit initialization vector is required to generate keyst-

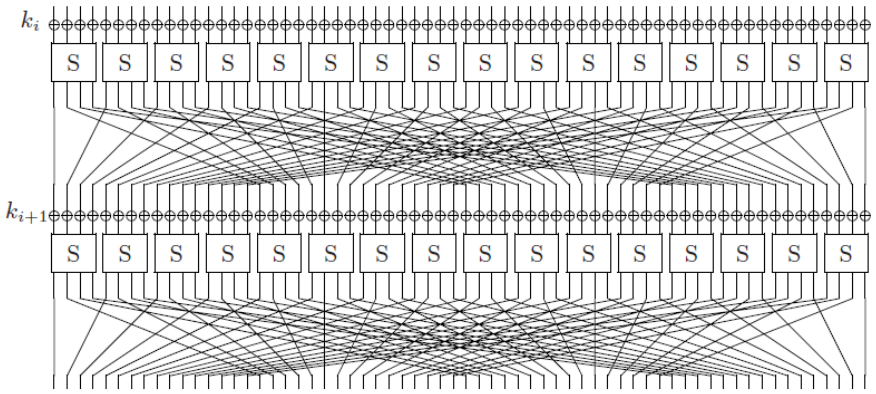


Figure 2.42. The SP-network in PRESENT (Bogdanov et al. 2007)

reams with the length of 2^{64}. Two tables are defined to provide non-linearity masks with S-boxes in output and feedback functions. They reduce the information leakage that makes other attacks than brute force search infeasible. These masks are implemented with the shift, exclusive-or, and adding operations.

HC-128 has 32778-bits of states which makes keystream periods very large. Therefore, in order to distinguish the period the attacker needs more than 2^{256} outputs. Furthermore, although output functions from non-linear masks leak partial information it is nearly impossible to recover the entire secret.

During the expansion, any modification on a bit results in a completely different state in the tables. Those are unlinkable changes. S-boxes provide a very low probability of collision given their high entropy and random inputs. Although, guessing the least significant bit seems to have a quite high probability, given the non-linear feedback function recovering the remaining bits cannot be distinguished at the same level as the least significant bit.

Hongjun (2008) provides several optimizations to the proposed scheme. Loop unrolling is used in the code. This method reduces the number of modulo operations with no performance deterioration. During experiments, GCC compiler is used. GCC provides an additional three optimization options. Therefore the result is 3.05 cycles/byte in Pentium M 1.6 GHz, 32 KB Level 1 cache, and 2MB Level 2 cache for encryption. Intel C++ Compiler 9.1 in Windows XP (Service Pack 2) results in 3.3 cycles/byte and Microsoft Visual C++ 6.0 in Windows XP(Service Pack 2) results in 3.6 cycles/byte.

The key setup requires 27300 clock cycles. This operation takes more cycles in order to provide protection from related-key/IV attacks. This implementation is not suitable for applications where the key needs to be updated frequently.

2.4.5. The Rabbit Stream Cipher

Boesgaard et al. (2003) proposed the Rabbit, a comprehensive stream cipher. Later in their work, they provided a detailed analysis against possible vulnerabilities and introduced a newer version against threats that are denoted in the analysis.

The design of Rabbit was based on the chaotic maps(Boesgaard et al. 2003). Chaotic maps are secret key cryptosystems that iterate one-dimensional map functions that generate chaotic behavior. An illustration of chaos-based block encryption is shown in Figure 2.43 by Jakimoski and Kocarev (2001). Chaotic systems are sensitive to parameters and initial points and provide randomness. They are advantageous against statistical attacks while can be efficient with simple implementations. However, Biham (1991) showed that the encryption of Habutsu et al. (1991) was weak against chosen-plaintext attacks. However, this notion is expanded with more complex applications. These applications have been software and hardware level, while the latter has been divided into analog and digital encryption schemes. These applications were analyzed in terms of distinguishability and observed in terms of one-time pad performance.

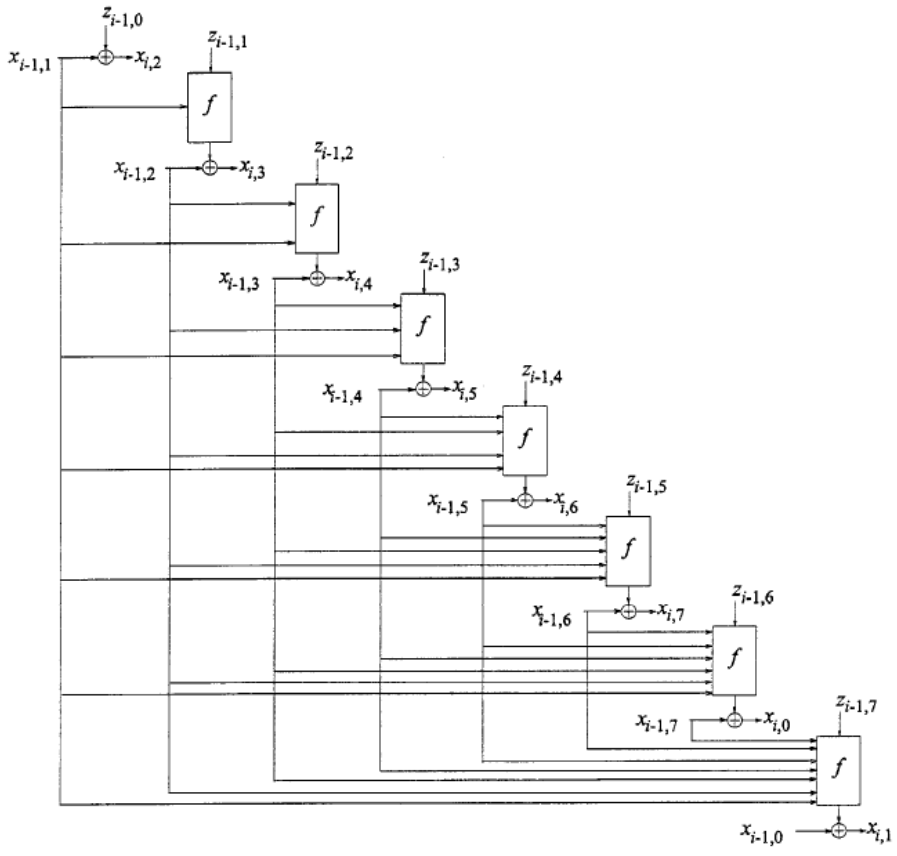


Figure 2.43. The chaos-based block cipher (Jakimoski and Kocarev 2001)

Rabbit (Boesgaard et al. 2008) exploits the pseudo-random structures of chaotic maps and extracts the cryptographic features from them. It is based on a 128-bit secret key and

additional 64-bits of initialization vector(IV) from the recent version. The scheme is based on key scheduling and symmetrical encryption/decryption operation.

The key scheduling is a three steps operation; key expansion, system iteration, and counter modification. During the key scheduling, 128-bits secret key is expanded. This operation consists of iterations where internal state bits are generated. At every iteration, a 128-bit pseudo-random output is generated with these state bits. These internal state data are 513 bits in total and split into state variables, counter values, and a carry bit.

State and counter variables have the size of 32-bit and there are eight of each variable in the internal state. These variables are derived from the key during key scheduling operation. Each one of the state variables is modified with a one-on-one corresponding non-linear function which also depends on the key and initial counter values.

IV setup operation has two stages; IV addition in order to modify the counter values and system iteration for avalanche effect. The internal state is indicated as the master state when key scheduling operation is terminated and the copy of it is modified during the IV setup. Using exclusive-OR operations, 64-bits of IV and 256-bits of counter state values are modified through four iterations. Counters keep the period length of state variables with a lower bound value.

The next-state function is an exclusive operation of Rabbit stream cipher. The algorithm modifies the next state at each operation with XOR and left-shift operations which results in a coupled rotating system among 8 internal states and counter states as shown in the Figure 2.44. $x_{j,i}$ are state variables and $c_{j,i}$ are counter variables. Counter values are updated with adding and modulo operation with a given carry bit and a set of constants. After each iteration/rotation by XOR'ing the internal bits and counter values, 32-bit words of pseudo-random values are generated. The encryption and decryption operations XOR these pseudo-random words with the plaintext and encrypted text respectively.

Boesgaard et al. (2008) aims to encrypt up to 2^{64} blocks of plaintext. This size keeps the encrypted text indistinguishable and makes brute force attacks over 2^{128} keys infeasible. Key setup operation depends on the non-linear map, hence, even if the adversary can obtain the counter bits, it will be hard to recover the entire secret key. However, the size of the map will raise questions on key collisions. Nevertheless, the iterations produce random values that are not affected directly by counter values and only one collision expected in the 256-bits keyspace, so collisions on counters are not a concern.

Related-key attacks threaten the Rabbit due to the symmetric operations. However, the correlation between the next-state and key setup functions can be thwarted by preventing symmetry among the set of constants used in the next-state function. This way, the weak keys will be eliminated.

In Rabbit stream cipher scheme, it is important to keep a proper lower bound on periods. There are several attacks analyzed on partial guessing: Guess-and-Verify and Guess-and-Determine. The first attack can be possible when output bits could be predicted from

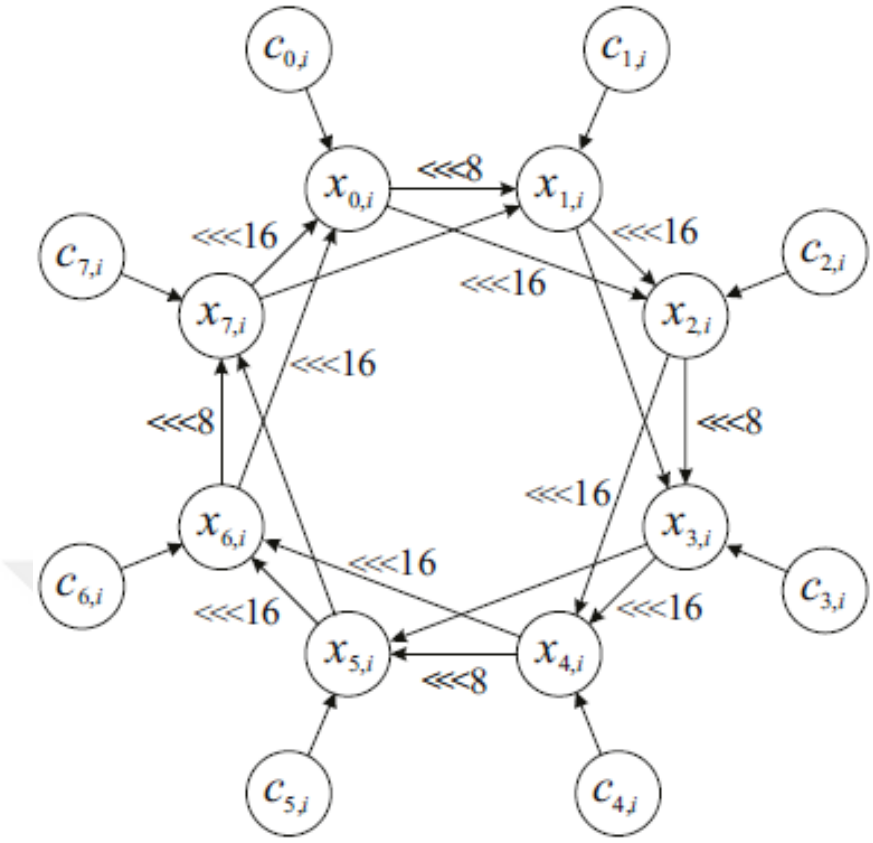


Figure 2.44. The next-state function of Rabbit (Boesgaard et al. 2003)

partial knowledge on intermediate state bits. The adversary will guess a part of the state and predict output to verify whether it was correct. The second attack is when the adversary can guess some unknown variables in the scheme, then deduce the secret. These attacks are more costly in complexity than exhaustive key search attack.

Boesgaard et al. analyze known algebraic attack and declare that those are not effective on the Rabbit scheme. Resulting text and g-functions are in Algebraic Normal Form(ANF). They depend on a large number of monomials that composes a multivariate polynomial. Their degrees are well distributed to keep the secret properties in non-linear form. Adversaries cannot obtain any statistical information that is not random. The state variables and the key are formed from over-defined equation systems. Therefore, a non-linear system generating these values are in a high degree and not sparse.

The Rabbit stream cipher provides an analysis in correlation attacks such as Linear approximation and second-order approximation. Walsh-Hadamard Transformation(WHT) is used for linear approximation and a distinguishing attack is infeasible when using less than 2^{64} blocks of output with a corresponding correlation coefficient of $2^{-57.8}$ when the transform is applied between the inputs and output of the next-state function. In second-order approximations, it is shown that the truncated ANFs of g-functions after the second

iteration gives correct approximations. When two neighbor bits are XOR'ed, some properties are dissolved. However, such an attack is arduous to perform. The analysis of differential attacks gives similar results as the correlation attacks.

The Rabbit stream cipher is implemented on Pentium III, Pentium 4, ARM7TDMI, and MIPS 4Kc. Pentium III performs in 3.7 cycle/bytes for encryption with code side of 440-bits and memory requirements of 40 Bytes. Key scheduling performs in 278 cycle/bytes for encryption with code side of 617-bits and memory requirements of 3 6Bytes. Finally, IV setup operation performs in 253 cycle/bytes for encryption with code side of 720-bits and memory requirements of 44Bytes. The scheme is also analyzed by its hardware performance. FPGA (Xilinx Spartan 3 or Altera Cyclone II) performs Rabbit on a 2-pipeline design with 6 multiplier units and gives decryption performance of 8.9 Gbit/s and 5.4 Gbit/s for each pipeline. If the multipliers increase, the throughputs will be significantly higher.

2.4.6. SOSEMANUK

SOSEMANUK is proposed by Berbain et al. (2008) as a synchronous symmetric software-based stream cipher. It is implemented based on two other cipher designs. One is SNOW 2.0 (Ekdahl and Johansson 2002) and the other is SERPENT (Biham et al. 1998) which is a block cipher. It promises 128-bit security with variable length for key between 128 and 256 bits. SOSEMANUK avoids some architectural vulnerabilities of both ciphers. Furthermore, Berbain et al. give a comparison of the performance differences.

SNOW 2.0 provides a good feedback structure on LFSR mechanisms while SERPENT provides linear transformations as shown in Figure 2.45. The diagram is in finite state machine (FSM) from. S is the Substitution box. α is a root of a primitive polynomial of degree 4 over $\mathbb{F}_2^8$. The LFSR is loaded according to the chosen primitive polynomial. Both SNOW 2.0 and SOSEMANUK are synchronous stream ciphers. However, SOSEMANUK reduces the internal state sizes and provides a direct data mapping on memory during the initialization vector(IV) injection. SERPENT encryption method proceeds in 32 rounds. However, the encryption of SOSEMANUK is based on its single round version, which is Serpent1. However, during keystream generation, SOSEMANUK uses 24 rounds version of it, which is Serpent24.

This architecture has mainly three operations; encryption, key scheduling, and IV injection. As declared above, the key scheduling part of the Serpent24 is used in the key scheduling. The mechanism processes the secret key and outputs 25 words of 32-bit quartets of subkeys (128-bit subkeys). The IV injection uses these 25 quartets with the initial IV starts the internal state of the cipher with the use of round operations of Serpent24. SOSEMANUK uses 12th, 18th, and 24th round outputs of subkeys for cipher. The little-endian convention is used with these quartets.

The encryption mechanism of SOSEMANUK consists of an LFSR, two registers, the linear transformation function *Trans*, *Serpent1* mechanism, intermediate functions, and supporting logic operations (Figure 2.46). LFSR contains 10 elements which are based on SNOW 2.0 implementation. Entire mechanism forms a finite state machine(FSM) with

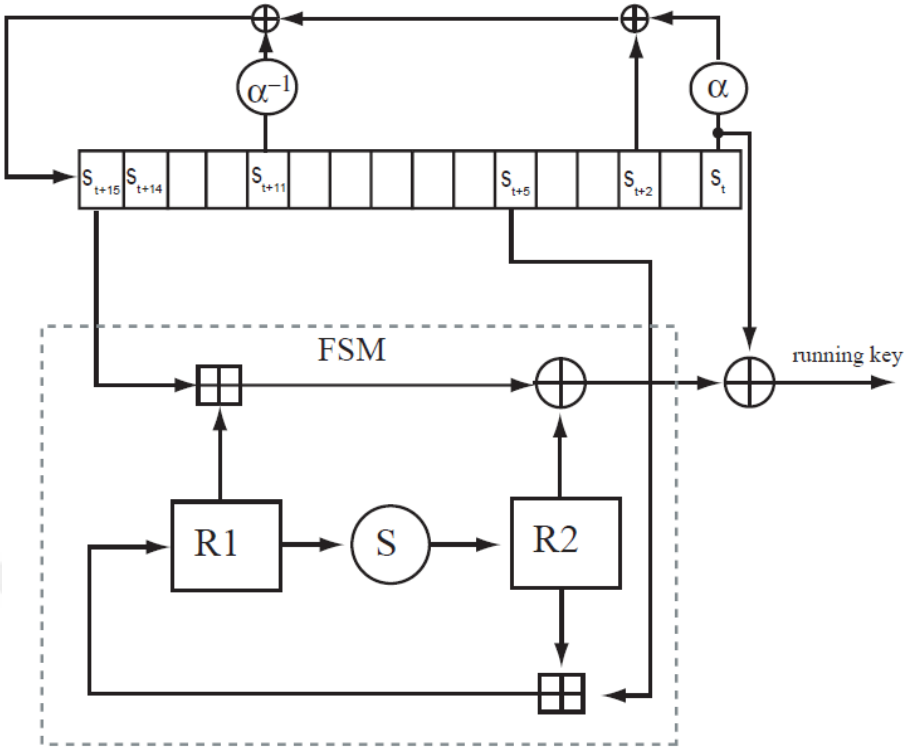


Figure 2.45. The diagram of SNOW 2.0 registers (Ekdahl and Johansson 2002)

64 bits of memory and two registers. At the end of each step, the FSM produces a 32-bit word. Two registers are initially loaded with 1st and 3rd words from the subkeys of the 18th round which are produced in the IV injection mechanism. The rest of the outputs initializes the LFSR blocks. The first four of the words are the quartet from the 24th round while the last four rounds are the quartet from the 12th round. The remaining blocks are filled with the 2nd and 4th words from the quartets of the 18th round.

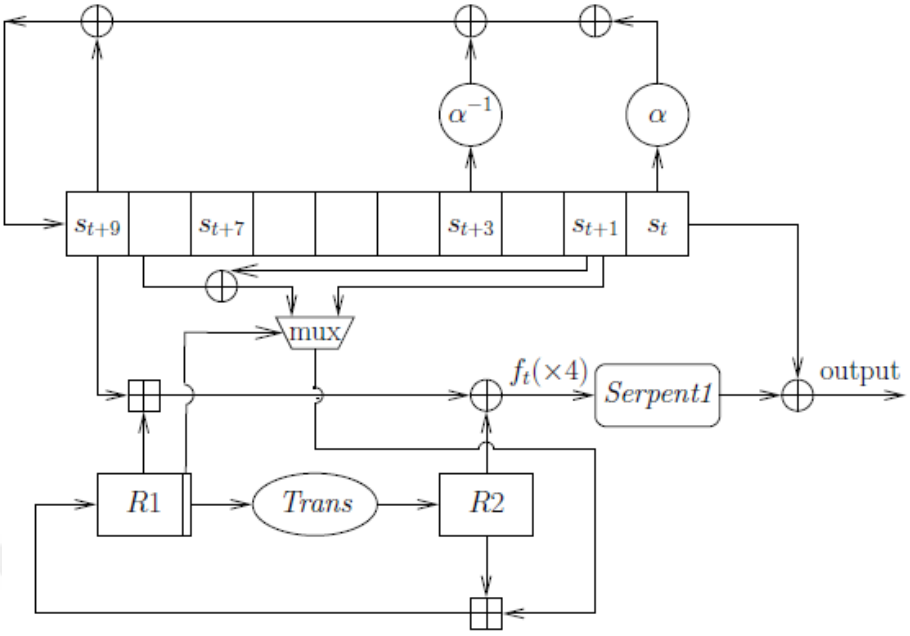


Figure 2.46. The SOSEMANUK based on SNOW 2.0 (Berbain et al. 2008)

During the iteration of encryption operation in SOSEMANUK registers and the internal functions update the 64-bit memory and both of the registers. While the steps proceed, the first register receives the exclusive-OR'ed (XOR) value of the second register and the multiplex(mux) operation of the 2nd block and XOR of the 9th block with 2nd. The mux result is selected by the initial value in the least significant bit of the first register. The second register receives the transformation function value of the first register. The transformation function multiplies the register value with the hexadecimal expression of the first ten decimals of the pi number. Then, it shifts the result to left by 7 times. The memory is updated via the internal function. This function sums the 10th block in the LFSR with the first register value. Then, it XOR's the result with the contents of the second register.

Serpent1 receives the internal function result in quartets and the final outputs are computed by XOR'ing the result of Serpent1 with the first four words of the LFSR. Finally, the encrypted text streamed at each step. The Figure **??** shows a diagram of these operations for four rounds.

Since it is less expensive to update the IV than the secret key, it was a good design choice to separate the key schedule and the IV injection. When the injection function is indistinguishable as in ideal Pseudo-Random Permutations(PRP), the IV setup cryptographically acts like a block cipher. Block cipher is advantageous in order to prevent any static data.

The authors point out that, the LFSR must never be physically shifted. This affects the

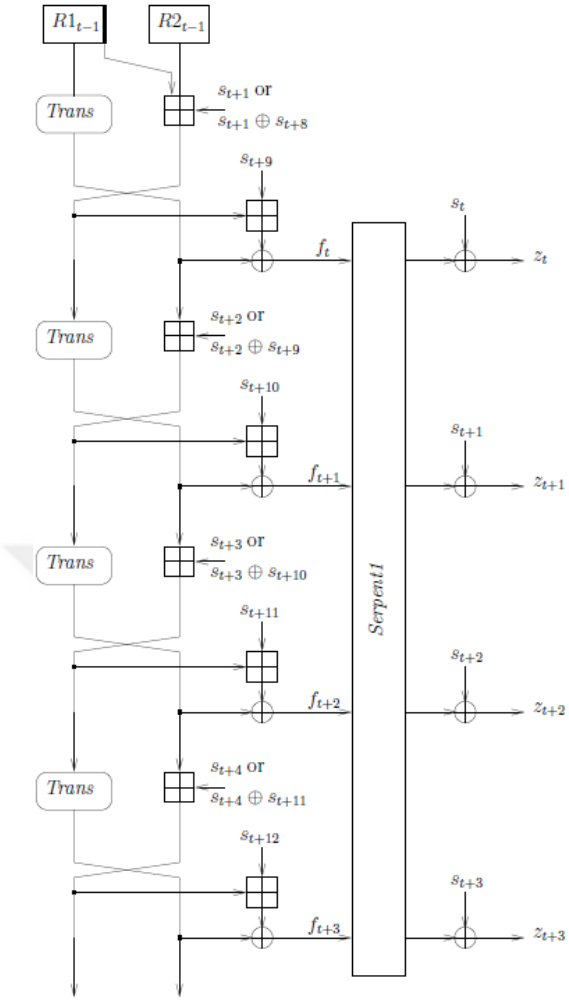


Figure 2.47. Output transformation of SOSEMANUK (Berbain et al. 2008)

timing, the implementation should be synchronous. Ten elements size is chosen to conform to modern processors in order to prevent guess-and-determine attacks. The feedback polynomial has been chosen as primitive and as sparse as possible similar to SNOW 2.0.

In the FSM, the authors chose transformation function to prevent static data to reduce the cache pressure. The rotation function removes the linear properties in the system. The multiplexing gate increases the complexity against correlation and algebraic attacks. During each step, the block locations in the LFSR is selected in order to have a good distance and to be coprime with LFSR length. Output transformation is mainly designed to avoid the algebraic and correlation attacks by providing non-linear mixing operations. Finally, the S-box in the Serpent1 operation is designed for efficiency.

SOSEMANUK is analyzed against, time-memory-data tradeoff, guess and determine, correlation, distinguishing and algebraic attacks. The scheme was implemented with C compiler. The architecture required 5 KB of code space, 4KB of static data, showed 900

cycles on key setup and 480 cycles performance on a Pentium 4 microprocessor. It has shown that, SOSEMANUK one of the best performing algorithms as a stream cipher. Due to its both software and hardware performance and cryptographic properties.

2.4.7. Grain

Linear feedback shift registers(LFSRs) are commonly used in this lightweight ciphers topic due to their statistical properties and simple hardware implementations. This improves their operations on resource-constrained environments such as RFID tags. LFSRs are mainly either bit- or word-oriented. Although bit-oriented LFSRs have simpler hardware implementation word-oriented LFSRs increase the throughput.

Grain (Hell et al. 2007) scheme uses bit-oriented LFSR implementation which allows vendors to implement the speed with additional hardware components. The architecture is built in order to process keystream and plaintext separately and synchronously.

The Grain architecture has two main components which are LFSR and non-linear feedback shift register(NFSR) as shown in the Figure 2.48. LFSR provides statistical properties while NFSR provides nonlinearity for the key and initialization vector(IV). The final keystream is produced through a balanced filter function that receives four LFSR bits and a single NFSR bit as a set of inputs. The final keystream is the output from the exclusive-OR'ed (XOR) seven bits of NFSR which XOR'ed with the output of the filter function.

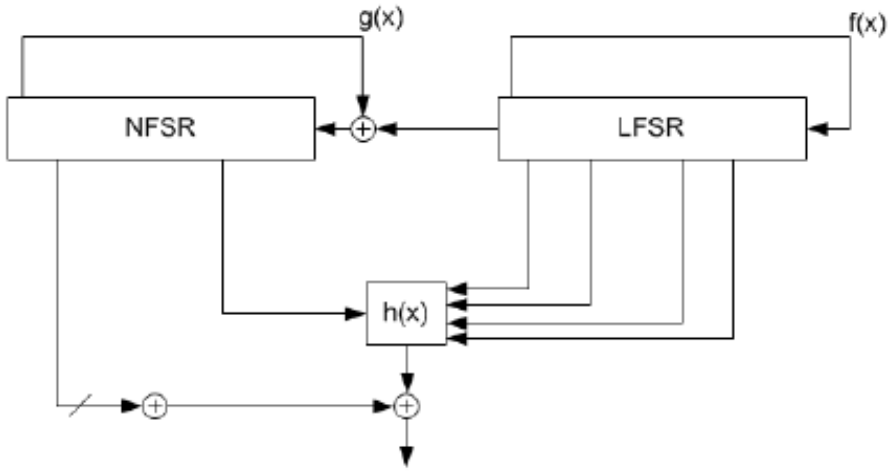


Figure 2.48. The Grain architecture (Hell et al. 2007)

Each of the shift registers has update functions. Initially, LFSR only receives its own output value. However, after the step of the keystream, it receives the XOR result of both its shifted values and the keystream. NFSR on the other hand initially receives the

XOR of its shifted value and LFSR. After the keystream initialization, it also receives the keystream XOR'ed with its early inputs.

The key initialization requires a key and an initialization vector(IV) which are 80- and 64-bit in size respectively (see Figure 2.49). During the initialization process, NFSR is loaded with the key values and LFSR is loaded with the IV at its first 64 bits locations. The remaining bits of LFSR is loaded with ones in order to prevent all zero state.

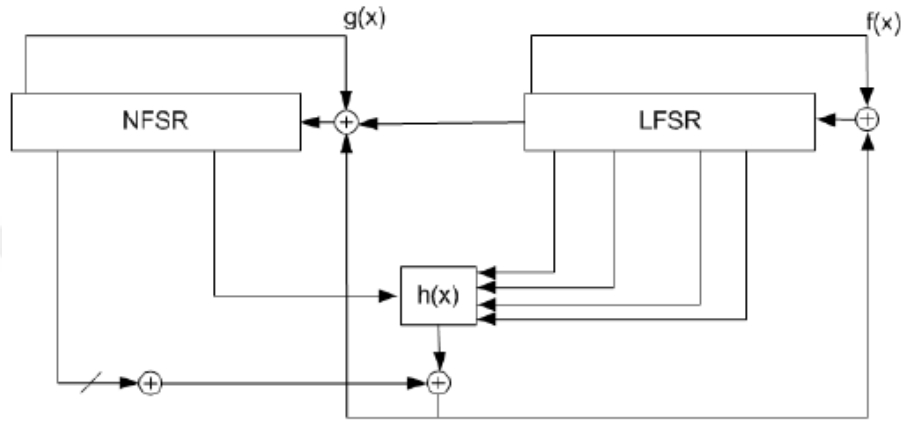


Figure 2.49. The key initialization (Hell et al. 2007)

The main goal of key initialization of Grain is to scramble the contents of the shift register in order to hide the secret value. However, the number of clocks is a tradeoff between throughput and security. Although the reinitialization of IV provides efficient security, it also induces a possible bottleneck. In order to provide a speed-up factor to vendors, the shift registers should implement multiple bits per clock (see Figure 2.50). The hardware specifications for speed factors are provided in the work. The authors declared that 160 steps of clocking are proper for this cipher, therefore it would be suitable for the factor to be divisible by 160.

There are no software efficiency tests provided in Hell et al. (2007) work. The main focus was on the efficiency of hardware implementation. They conducted their tests on FPGA architectures. They mainly used the ALTERA MAX 3000A family, EPM3256. However, for reference they also tested their work on ALTERA MAX II and ALTERA Cyclone. MAX II and Cyclone have shown better throughput performance almost 5 times of the MAX 3000A. The details of their throughputs are shown in the figure. The Grain scheme is also analyzed against E0 which is used for Bluetooth technologies and A5/1 which is used for GSM.

The cryptanalysis against the Grain architecture has shown that correlation, algebraic, Time/Memory/Data tradeoff, and Chosen-IV attacks are not efficient at 80-bit key sizes, 64-IV, and after 80 key initialization steps. 160 steps provide a good avalanche effect on

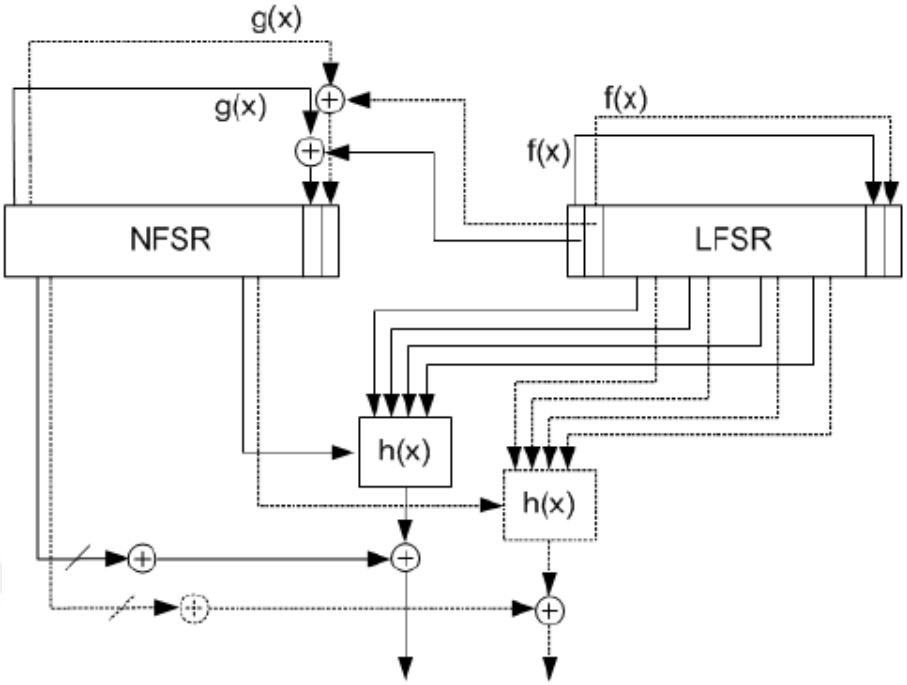


Figure 2.50. A modification to speed-up the Grain cipher (Hell et al. 2007)

the keystream. However, bit induced fault attacks have been a big threat to stream ciphers. The attacker can apply certain bit-flips to induce a fault in order to retrieve partial information about the secret key. It is important to protect the five inputs of the filter function. However, during such attacks, it is very difficult to determine the induced positions. Furthermore, the non-linearity of NFSR does not expose any relation with its input on filter function against other inputs those originate from LFSR. Fault injection attacks are not feasible in this context.

2.4.8. The Salsa20 Stream Cipher Family

Bernstein proposed a family of stream ciphers called Salsa20 in eSTREAM project that aims for high speed at wide variety of applications with 256-bit security despite eSTREAM call for submissions requested 128-bit security. Bernstein provides different security levels with effective implementation as previously indicated schemes. However, Salsa20 have a static key unlike those while the number of rounds change e.g. Salsa20/20 has 20 rounds, Salsa20/8 has 8 rounds. Despite the fact the Salsa20 was designed before the release of Intel Core-2 architecture it is able to take advantage of the technology. The cryptanalysis parameters are provided over Core 2 performance.

Salsa20 is a stream cipher based on operations of addition, exclusive-OR(XOR), and rotation. It performs like a block cipher. However, these operations are applied on an array of expanded 256-bit key and a 64-bit nonce. The array is in the form of a 4x4 matrix and it is made of 16 32-bit words. Four of the constant values are set in diagonal locations of the array. The rest of the locations are filled with the first four word of the key, two words

of the nonce, two words of the block counter, and finally the remaining words of the key in the given order. In the case when the key size is chosen as 128-bit zero-padded into 10 Bytes, the constant values change.

For the given number of rounds, Salsa20 sums diagonal words and above-diagonal words, rotate them to left by 7 bits, and finally, XORs the result into the below-diagonal words. Then sums diagonal words with below-diagonal words, rotate them to left by 9 bits, and finally, XORs the result into the below-below-diagonal words. It continues in the same direction for each column and rotate left by 13 bits. Then, it modifies the diagonal words and rotate them to left by 18 bits. Finally, transposes the result. This modification is for one round. The implementation can eliminate the transposes and switch between rows and columns at each round. The final result is a 64-byte output block. During the column rounds, there is no relation between different column words and during the row rounds, there is no relation between different row words. This eliminates the need for diffusion operation. The number of rotations are chosen to spread every low-weight change although their values do not make much difference. The rotation over sum is chosen because of the three-operand operation(LEA) existing on the x86 microprocessor architecture.

The encryption is done by XORing the output block with plaintext and the message is decrypted by XORing the output block with ciphertext. The order of access to the output stream is not important. Furthermore, parallel encryption is possible. There are no additional preprocess operations nor there is any need for a feedback from the plaintext or the ciphertext. This model is encrypted as in modern block ciphers with a mode of operation, counter mode. As it is explained during the generation of the output block, the block counter is implemented into the 4x4 array with the key, the nonce, and the constant values. Bernstein did not choose CBC as in AES. By following some arguments against CBC about the disadvantage of having to add extra rounds and delay, he decided on the counter mode.

In contempt of eSTREAM requesting 128-bit security, Bernstein insists on 256-bit key with the option of 128-bit with zero padding. He further explains why the 128-bit security is vulnerable despite the overestimated cost analysis of bulk prices of hardware components, ignoring the performance of simulatenous attacks and the possibility of early guessing.

Bernstein, in 2005 after proposing this scheme has offered $1000 for whomever found the most interesting cryptanalysis over Salsa20. Crowley (2005) on Salsa20/5 and presented his attack the following year has won the prize. His attack works by finding input difference after 3 rounds and works 2 rounds backwards, and finally, was able to guess 160 relevant key bits. Fischer et al (2006) reported an attack on Salsa20/6 with input difference after 4 rounds, continue to work 2 rounds backwards, and then, retrieve 160 relevant key bits. Tsunoo et al(2007) reported an attack on Sala20/7 finding an input difference after 4 rounds, work 3 rounds backwards, and retrieve 171 relevant key bits. Finally, Aumasson et al (2008) reports finding an input difference after 4 rounds, work 4 rounds backwards, and retrieve 228 extremely relevant key bits on Salsa20/8.

2.4.9. TRIVIUM

Cannière and Preneel(2008) propose TRIVIUM in eSTREAM as a stream cipher that exploits the understandability and improved efficiency of the block ciphers. While they aim to restore the efficiency in stream ciphers which was their strong feature previously, they also observe the common attacks against the stream ciphers which are mainly distinguishing and guess-and-determine.

The block ciphers can be separated into three layers which are input and output, substitution and diffusion and masking as shown in Figure 2.51. Substitution layer provides the nonlinearity properties while diffusion layer forces characteristics to prevent correlations. Finally, the masking layer provides the homogeneity. During the construction of the block ciphers, the designers first concern about the linearity of the keystreams by calculating the boundaries on the linearity of the cipher. The adversaries must not be able to distinguish the secret by observing the linear correlations. They also forward and backward trace the path where they find the propagated correlations using the piling-up lemma principle (Matsui 1993).

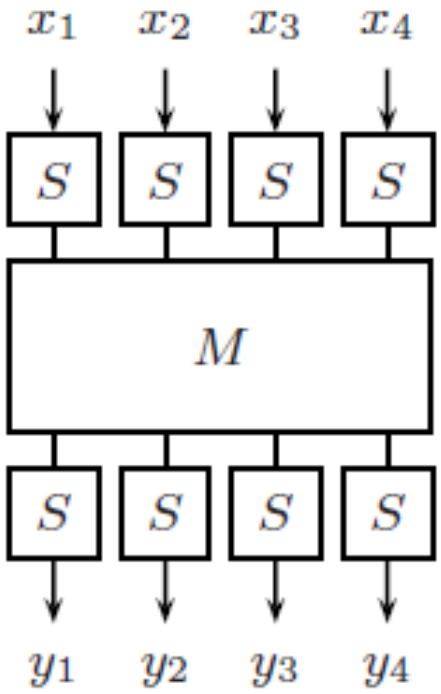


Figure 2.51. Three layers of standard block ciphers (Canière and Preneel 2008)

TRIVIUM uses a word-oriented approach on stream ciphers to be applied to substitution boxes(S-boxes). The cipher is based on a circular implementation of 4th-order linear filters(see Figure 2.52). These filters simplify the stream implementation of the three-layered architecture used in block ciphers as shown in Figure 2.53. The linear filter implementation can be represented in feedforward and feedback polynomials. The authors

designed the polynomials in a way the correlated propagated correlations to be minimal.

TRIVIUM consists of a two-rounds keystream generator and a three-rounds stream cipher. The basic idea is using an output feedback (OFB) mode of operation passes the initial value through the layers of substitution, diffusion, and masking operations within multiple rounds. The construction discards the need for round-keys with the use of linear filters.

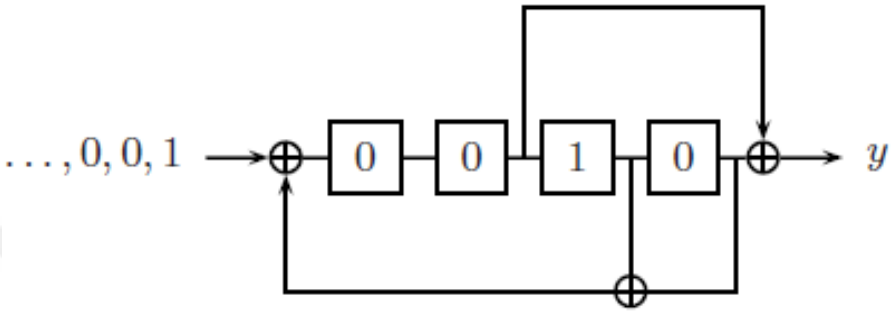


Figure 2.52. The 4^{th} order linear filter (Canière and Preneel 2008)

The design evolved into a bit-oriented form from a word-oriented one for a more compact design and to prevent the clustering phenomenon in applications. This means the S-boxes are reduced to 1x1 bit in size which makes them linear. They are replaced with a stream of biased random bits. However, the construction now relies on these bits and generating them is another problem.

The cipher implementation of TRIVIUM requires an additional round compared to keystream generation. The cipher has three sub-generator polynomials to solving the previous problem induced by the generation of the biased random bits. The designers also implement additional AND-gates receiving state bits from the text to interleave these sub-generator functions with the resulting ciphertext.

TRIVIUM requires sizes of 80 bits for the key, 80 bits for the initial value, and 288 bits for internal states. The internal state is generated by the two-round construction and is an iterative algorithm which modifies the 3 bits of the state and produces 1 bit for the output keystream.

The authors discuss the security parameters in the light of possible attacks such as correlation, period, guess-and-determine, algebraic, and resynchronization. Their results show that the linear distinguishing under correlation attacks requires 2^{144} time complexity and 2^{144} bits should be correlated in the key stream. A straightforward guess-and-determine attack requires 195 correlated bits in the key stream to be solved within 2^{195} time. Khazei(2006) presented an attack under 2^{135} complexity of time that required 288 bits to be correlated in keystream. Maximov and Biryukov(2007) reported another attack that required 2^{90} time complexity while 2^{61} of the bits needed to be correlated in the

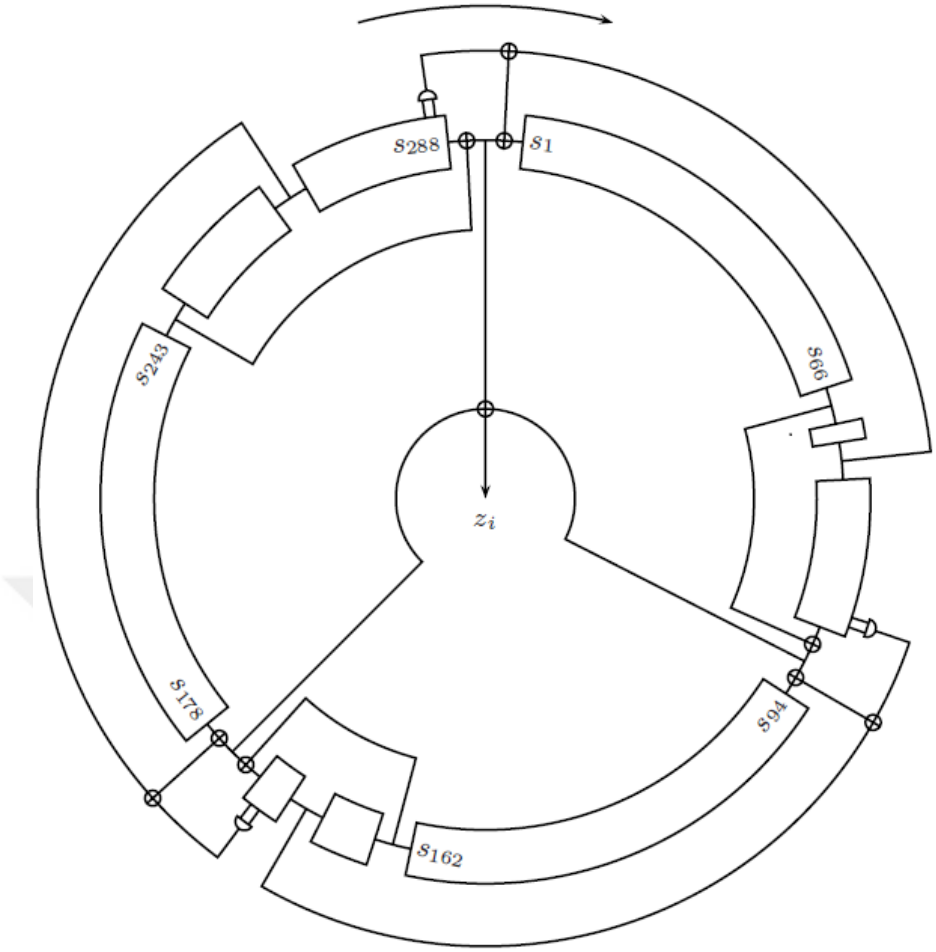


Figure 2.53. TRIVIUM structure (Canière and Preneel 2008)

keystream. Raddum presents an algebraic attack that requires 2^{164} time complexity and 288 correlated keystream bits. These analyses show that the security for an 80-bit key is provided.

There are researchers who ran hardware tests on this design to show it is suitable for low-power implementations and for high throughput against AES implementation. TRIVIUM is not intended for software efficiency, however, software applications implemented on 1700 MHz PENTIUM M with C language compiling resulted in 5.3 cycles/byte for keystream generation speed and cost of 51 cycles for key setup while 774 cycles for IV setups.

2.4.10. Evaluation of IoT Applicable Lightweight Cryptography

The lightweight block ciphers are analyzed in terms of small block sizes, small key sizes, simple rounds, and simple key schedules. Leander et al. (2007) proposed DESL(DES lightweight) which has the smallest key size which is 54-bit. It is an improved version of DES with two times of rounds with less computational requirements. Hummingbird and Hummingbird2(Mohd et al. 2015) has the smallest block sizes of 16-bit while they

have 256-bit key size. RC5(Rivest 1994) has the most flexible number of rounds varying from 1 to 255. However, Hummingbird block cipher family performs better with 4 rounds. The next high performing block cipher is AES with the variable number of rounds such as 10, 12, and 14. The block cipher structures are mainly, Substitution-permutation networks, Feistel networks, and GFS. TEA(Wheeler and Needham 1994) and XTEA(Yu et al. 2011) have the simplest key schedule.

The important primitives related to IoT in lightweight hash functions are small output and message sizes. PHOTON(Guo et al. 2011), Quark(Aumasson et al. 2013), SPONGENT(Bogdanov et al. 2011), and Lesamnta-LW(Hirose et al. 2010) have been analyzed under these two primitives. Also Bogdanov et al.(2008) presented a hash function that is dedicated to RFID tags.

High-performance systems are divided into Customized CPUs, Crypto co-processors, Crypto arrays, and Crypto multicores. Tillich and Großschädl(2006) presented the Instruction Set Architecture(ISA) for cryptographic instruction sets. It is not very applicable in practice, but it provides new research topics. The co-processors are designated to operate separately from the main CPU in order to handle the overhead information and execution performance. Hodjat and Verbauwhede (2004) designed a co-processor to operate with DES and AES algorithms.

Crypto arrays are designed to provide processing elements for parallel executions for algorithmic tasks. However, they additionally require an architecture to move the data between ALU and memory. The multi-core systems are high-performance encryption systems those are able to run parallel computations. Grand et al. (2011) designed MCCP with 8 cores. The core is implemented on FPGAs and analyzed with AES algorithm.

James and Kumar (2016) proposed a way to implement AES that they could be applicable in RFID tags. Li et al. (2016) implemented QTL which is a lightweight variant of Feistel network. Karakoç et al. (2015) suggested AKF which is also a Feistel network variant with key altering features. They focus on the major attacks those aim at vulnerabilities at absence of key scheduling in order to thwart them. Bansod et al. (2015) proposed another lightweight block cipher algorithm which uses S-boxes of PRESENT scheme. They present an improved memory usage with their hybrid design. Biswas et al. (2015) presented an algorithm that is based on elliptical curve cryptography(ECC) for a simple authentication mechanism. Guo et al. (2015) emphasized on the sensitive PHI(Patient Health Information) and built a high-end secure platform which enforces roles and rules on the entities in the platform. The user-friendly NCRYPT is designed by Verma et al. (2014) for Android mobile platform for an efficient encryption of selected files. Peng et al. (2016) provide an 8-round highly lightweight block cipher scheme for Underwater Acoustic Networks(UAN) that enables the security at energy-constrained environments. The communication protocol relies on a chaotic oscillation model. They present high-level solutions against exhaustive key search attacks as well as conventional attacks widely performed against IoT frameworks.

There are lightweight encryption schemes aimed for cloud computation. A collabora-

tion with attribute-based encryption(ABE)(Naruse et al. 2015) is designed by Huang et al. (2016). with a fine-grain access control. A Proxy Re-Encryption (PRE) is proposed by Liang et al. (2015) which focuses on cloud data sharing security. They also collaborate ABE with their Ciphertext-Policy system while aiming to push most of the high computational operations into cloud system from low-resource devices. Fugkeaw et al. (2016) published the hybrid VL-PRE scheme which has three phase key generation, re-encryption key update, and re-encryption key renewal operations in highly-constrained environments. Baharon et al. (2015) defined a lightweight encryption scheme, the Lightweight Homomorphic Encryption(LHE). They aim to reduce the computation density in key generation and encryption scheme. Zegers et al.(2015) proposed a lightweight cryptographic handshake protocol which aims at mobile cloud data sharing.

There are also specific protocols designed for IoT standards. Yao et al. (2015) proposed an ABE based encryption using ECC and Diffie-Hellman protocol. It focuses on central attribute authority. Yang et al. (2016) presented a lightweight security for health information based on Electronic Health Records[EHRs] in order to find an efficient dictionary search in the system while performing in cross-domain PHI. Sahraoui and Bilami (2015) suggested a HIP-based end-to-end security scheme using Diffie-Hellman protocol and DTLS for flexible key management and cost reduction to be used in smart home applications. Baskar et al.(2016) designed a secure environment based on ALTERA DE1 system, Blowfish and XTEA encryption, and chaotic maps. It provides high security and performance. Finally, ERNEST (Ernest W, 2017) is a lightweight cryptography scheme proposed for the Internet of Everything. They suggest new generation lightweight encryption techniques.

3. MATERIAL METHOD

Roman et al.(2011) studied the high-level security approaches on the Internet of Things in the context of threats the constrained environments are vulnerable to due to their heterogeneous and interconnected nature. They point out that the biggest challenge in the IoT vision is preventing the growth of ingenious malicious attacks and provide solutions to limit their activities.

Traditional lightweight cryptography and security protocols are not enough by themselves to protect the IoT environments. The approaches should be structured with multiple mechanisms in order to provide the same level of security for all humans and things in the environment. It is important to design security frameworks concerning about well-known and novel threats while providing extensive data analysis and reports. Aside from privacy, the security definition in these frameworks should include safety, economy, legal properties, and up-to-dateness.

The environments that encompass IoT and its variants are newly referred to as the super-connected world. Along with the complexities and manyfold threats in this world, it is important to have a holistic approach to the newly developed backbones. The recent attacks such as NotPetya and WannaCry and many data breaches alarm the necessity of adaptive security backbones those are autonomous and require less human intervention. In the light of these new concerns, the questions about centralization arisen. Many of the contemporary technologies that are related or unrelated to IoT, aim for decentralized approaches those have less dependency on a central entity but relying on a common trust mechanism.

IEEE 802.15.4 standards require secure communication channels with proper encryption and key management schemes employed, in environments where all the devices are easily accessed physically. Existing suitable cryptographic schemes are explained and analyzed in the early sections. Many of the devices are not easily replaced or removed. Therefore, autonomous rekeying operations are required on top of the cryptographic schemes under the concerns of optimal battery life drainage. In order to reach the maximum interconnectivity, it is important to follow the standards while inducing autonomy.

In order to avoid dystopic scenarios such as big brother, three features are important. Firstly, privacy by design allows users to control their own privacy levels through dynamic consent tools. Secondly, transparency for users who may be concerned about their sensitive data used in the system. And lastly, data management is a big issue in order to designate an entity that is responsible for privacy information.

The authors declare principles for identity mechanisms. Every object should have a distinct unique identifier. They should be able to acquire multiple identities including their core identity. An object should be able to authorize itself and authenticate its owner. Multiple identities are important since having disposable identities or pseudonyms allow them to create anonymity. Granularity is also a concern in this context to distribute the credentials among humans and things in the environment.

It is pointed out that governance and trust reduce the liability among the individuals and optimize the operations. They also reduce the uncertainty of actions and provide appropriate levels of service for entities according to their credentials. They also enable to control their activities in the framework. However, a proper balance is required in order to have stability, fairness, and to avoid excessive monitoring.

Fault tolerance decides the threshold of protection against malicious attacks and how entities will treat them. It can be applied within three phases. Firstly, securing everything in the framework is essential. Then, the elements must be allowed to monitor their network and service states. Finally, with the incorporation of security mechanisms, they should be able to defend themselves. Recovery can also be investigated under fault tolerance context.

There are several standards for IoT technologies in development. ISO/IEC 14443 propose an architecture for contactless proximity cards through information flow protection. IEC 62591, WirelessHART, is a protocol for industrial WSNs which consists of encryption, authentication, and key management mechanisms. GS1 keys form an identification system and ucode provides a hardware-agnostic identifier.

The Internet Engineering Task Force(IETF) provides a set of standards for cryptographic privacy and protocols. 6LowPAN and ROLL provide IP connectivity. The CoRE is a lightweight RESTful web service architectures while the CoAP is a generic Web protocol definition. However, there are impractical constraints in these standards.

Roman et al. declares that single-sign-on(SSO) mechanisms are useful for practical authentication mechanism. Creating virtual profiles for users on logical nodes to verify ownership through digital shadowing is also a good approach.

There are several modern approaches with adaptive features on privacy protection. Although, the limiting access scenarios are very common some approaches may not provide sufficient anonymity to the system actors. Preference matching and location-dependent query applications can provide the desired level of anonymity. There is also the concept of privacy coach where users can authenticate themselves with loyalty cards and download their privacy policies. The card readers can verify user consent on information sharing. The coach acts like a middle secure entity that provides a layer of anonymity and services with personal information for users.

3.1. IPv6 and Low-power Wireless Personal Area Networks

In the 1990s IPv6 was developed in order to make the Internet Protocol scalable against continuous and fast growth of the Internet. The existing web protocols like HTTP are computationally complex for IoT devices. IPv6 and Low-power Wireless Personal Area Networks (6LoWPAN) defines how IPv6 is adapted into low data rates, energy sources, and small sizes in RFID-based networks according to Mulligan and Group (2007). In the network stack, IPv6 is operated in the network layer while there is a 6LoWPAN Adaptation Layer between the network layer and 802.15.4 Media Access Control layer. The main design requirements are low data packet and routing overhead, memory, and

computation. Also, the implementation of sleeping nodes in order to extend the battery life specifications are important. 6LoWPAN mainly supports mesh-topologies. They have several modes of operation such as flooding, hierarchical routing, geographic routing, and self-organizing coordinate routing. Here, "routing" refers to the path formation, path computation, and packet forwarding within the IP layer. Route operations are provided with 6LoWPAN Router or forwarder device and 6LoWPAN Border Router between the devices (Kim et al. 2012). 6LoWPAN Border Routers are indicated as 6BR in this thesis.

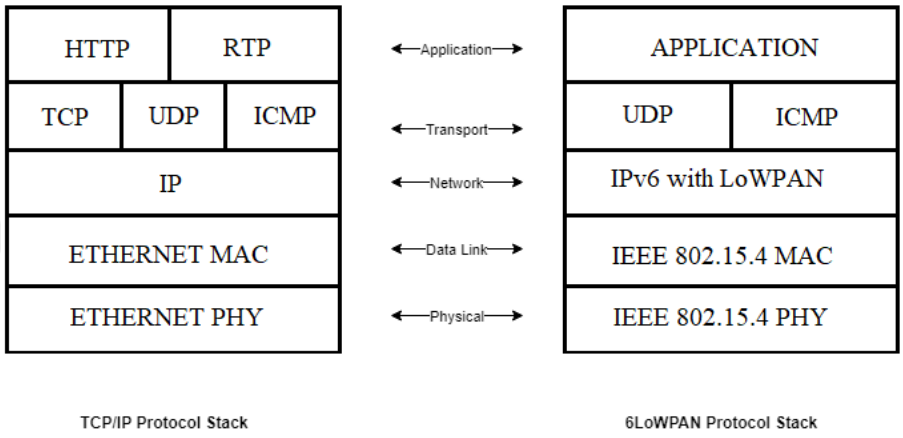


Figure 3.54. Standard TCP/IP protocol stack vs 6LoWPAN protocol stack

The differences between the TCP/IP protocol stack and 6LoWPAN protocol stack are given in Figure 3.54. 6LoWPAN stack does a separate application layer than HTTP and RTP. It only supports UDP and ICMP at the transport layer. Therefore, 6LoWPAN and TCP/IP systems have to translate packets of others in order to communicate. Moreover, 6LoWPAN supports IEEE 802.15.4 MAC and physical layer. At Network layer they have a different set of a protocol which can be converted during communication.

The general structures of 6LoWPAN packets are shown in Figure 3.55. The physical layer has 5 octets of synchronization header consists of 4 octets of the preamble, 1 octet of the start of frame delimiter (SFD) and 1 octet of the physical header. The Physical Service Data Unit (PSDU) is related to the MAC layer. The MAC header (MHR) has two octets of frame control data, 1 octet of sequence number, an addressing field including destination identity, destination address, source identity, and source address, variable (from 0 to 102 octets) length of frame payload, MAC Service Data Unit (MSDU), and 2 octets of MAC footer. MSDU is related to the network layer where IPv6 protocol is defined. It has a variable size of IPv6 network header and data payload.

3.1.1. Attacks Against 6LoWPAN

i) **Fragmentation Attack**

The adaptation layer of 6LoWPAN provides header compression and packet fragmentation for IoT frameworks. The fragments of IPv6 packets include information about

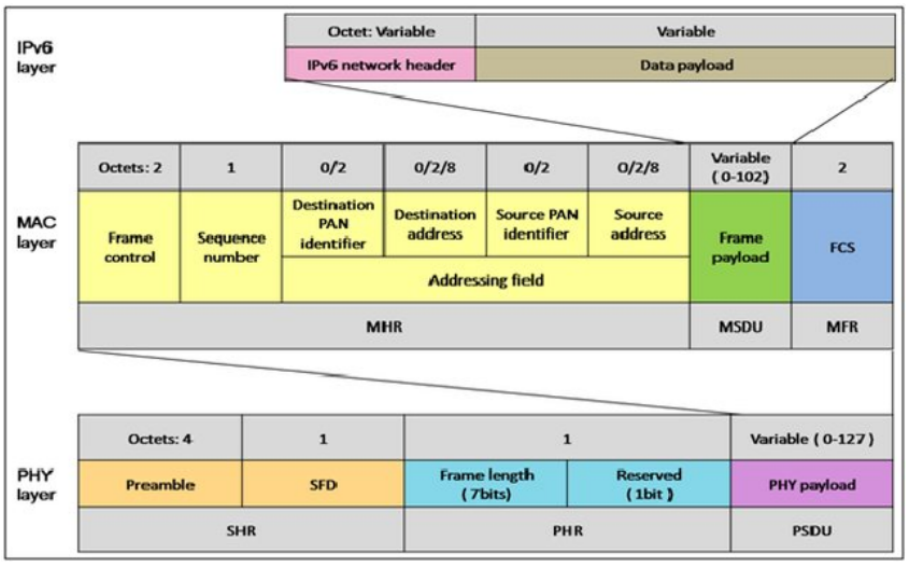


Figure 3.55. IPv6 packet format (Chan et al. 2011)

re-assembling so the fragments are able to arrive asynchronously. An adversary can intervene during the fragment transmission and replaces the legitimate fragments with its own illegitimate fragments at the chain. The absence of authentication of fragments gives way to many spoofing attacks. Split mechanisms and content chaining can prevent this threat (Hummen et al. 2013).

ii) **Authentication Attack**

There is no existing authentication mechanism for nodes during the network joining process. These attacks are possible with the absence of node authorization, data filtering, distribution of legitimate node list, and node entrance detection (Oliveira et al. 2013). The client application can access the 6BR through TCP communication to utilize the management operations, authorized node list, and pending node list. Sensor nodes and 6BR are connected through UDP. Sensors have their agent applications and an authorized node list dedicated for each node.

iii) **Confidentiality Attack**

Eavesdropping, the man in the middle, spoofing kinds of attacks can be categorized under confidentiality attacks against 6LoWPAN. IPsec provides End-to-End secure communication for WSNs and the Internet. The authentication header and encapsulation security payload can provide a level of security against such attacks.

iv) **Internet-based Attack**

The various attacks received from adversaries through the Internet. The 6LoWPAN is originally unprotected against Internet communication. The installation of a gateway on 6BRs can prevent such attacks.

3.2. RPL

RPL(IPv6 Routing Protocol for Low-power and Lossy-networks) defines communication protocols between the communications. It supports point-to-point, multipoint-to-point, point-to-multipoint communication in lossy networks complying with the 6LoWPAN standards.

RPL involves with external access and transport control. It also implements a method to distribute data over a dynamic network. It assembles the topologies of independent routers and can group devices under a subnet with an alias. RPL consists of operations such as IPv6 Neighbor Discovery(ND), Prefix Information Option (PIO), and Route Information Option(RIO) (Winter et al. 2012). It forms the DODAG (Destination Oriented Directed Acyclic Graph) with one root which is also known as the sink device. Every node should be able to determine the direction of packets they receive. They know all of their descendents and uses the route towards its parents or the DODAG tree root.

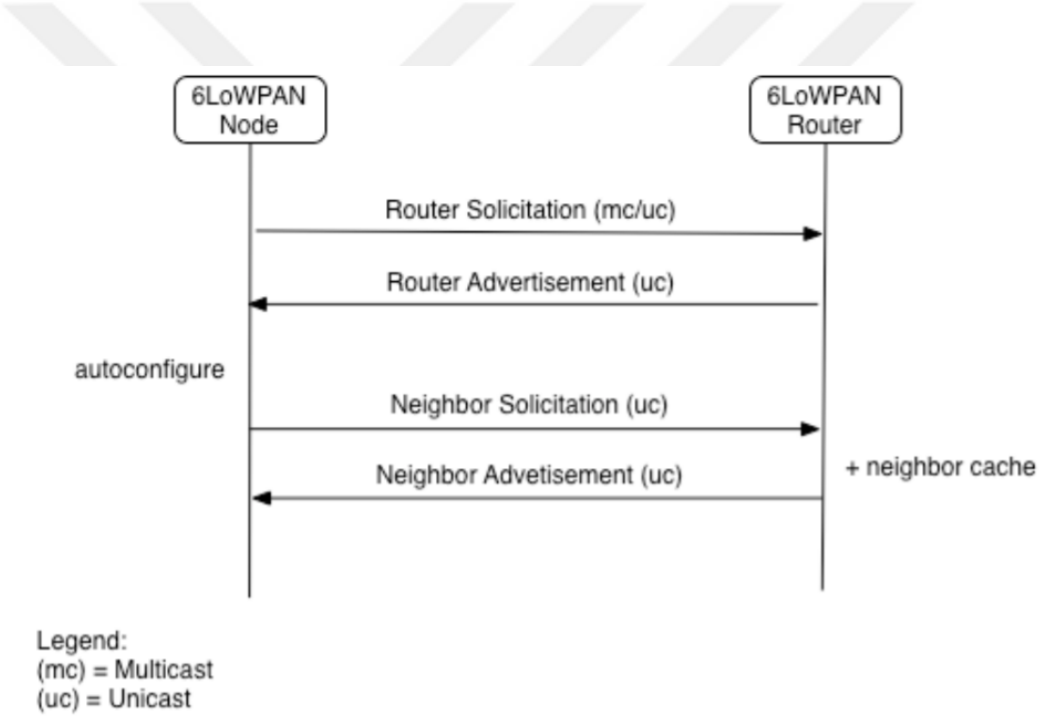


Figure 3.56. RPL Neighbor Discovery Protocol (Vasseur et al. 2011)

In Figure 3.56 ND protocol is illustrated. First, IPv6 node sends a router solicitation message to the IPv6 router. The router replies with advertisement message. After receiving the advertisement the requesting node automatically configure itself and sends a neighbor solicitation message to the router. The IPv6 includes the information on its cache and replies with an advertisement message. Most of the messages between the replying router and the requesting node is unicast. The initiating node proceeds with this protocol with all available neighbor nodes.

There are five settings of RPL routing as shown in the Figure 3.57. The first diagram

is a sample wireless network. The diagram b illustrates a multipoint-to-point, c shows a point-to-multipoint, d indicates the point-to-point with storing mode, and finally e illustrates the point-to-point without the storing mode communication flow. In the first illustration, a wireless network shown. Secondly, a multipoint to single point is illustrated. The third figure is the reverse; single point to multipoint communication. The last two are point-to-point communication with two modes; storing and non-storing.

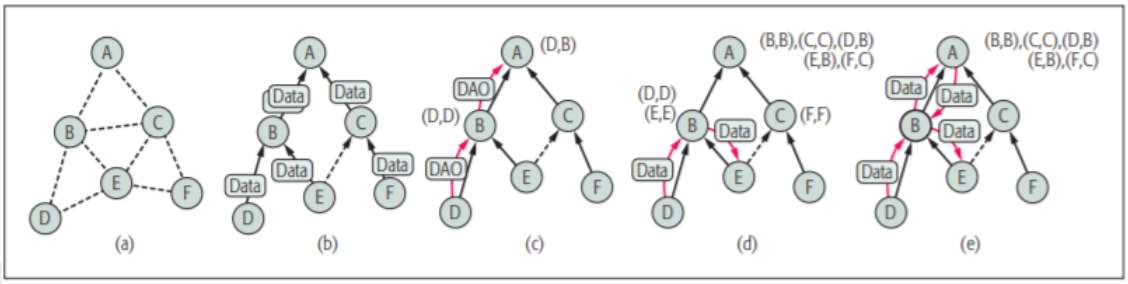


Figure 3.57. Routing in RPL (Iova et al. 2016)

3.2.1. Attacks Against RPL

The attacks against RPL protocol are Selective Forwarding, Sinkhole, Sybil, Hello Flooding, Wormhole, Clone ID, Blackhole, Denial of Service, and Spoofing attacks.

i) **Selective Forwarding Attack**

In Selective forwarding attacks, a malicious node selectively forwards or drop packets from a node(s) in process of being corrupted to another node. The worst-case scenario is where the adversary does not forward any packet. It is an attack that is prone to be part of a more complex attack for a more powerful threat. Dynamicity of the paths and encryption in order to prevent the adversary to able to study the traffic.

ii) **Sinkhole Attack**

Here, an adversary aims to obtain sensitive information in the network by compromising a node to retrieve all the relevant information of traffic. The compromised node acts as a sink device and gains the trust of other elements in the network. Using this breach, the adversary can launch further attacks to obtain the desired information.

iii) **Sybil Attack**

The adversary in Sybil attack impersonates several legitimate nodes using different identities and able to introduce anomalies into the network at its own accord. The attack could be launched during direct and indirect communication. The identities are either fabricated or stolen. The impersonating nodes can act simultaneously. Temporal verifications can provide aid against these attacks.

iv) **Hello Flooding Attack**

Hello messages are required packets transmitted over the network by newly int-

roduced nodes. They announce themselves nearby to the nodes in close proximity. However, malicious nodes can flood the network with those messages without proper delay, thus creating a breach from disruption. It is enough for the attacker to send re-broadcast the previously send hello packets. It does not need to initiate message transmission. This attack can damage both the topology and information traffic.

v) **Wormhole Attack**

In this attack (Pandey and Tripathi 2010) the adversary places two rogue nodes in a strategically good location of a network enough to disrupt the packages and creates a tunnel from these two nodes. One of those rogue nodes is called origin-end and the other called destination-end. The data packets are transmitted and continuously replayed from origin rogue node to the destination rogue node. This attack is observed by Mahajan et al. (2008) early in Mobile Ad-Hoc Networks(MANETs). The adversary created an illusion of being connected directly to the MANET despite the nodes being illegitimate. The attacks can be created separately in order to disrupt the network and its topology. However, there are settings where the overlay tunnels pass through legitimate colluder nodes.

vi) **Clone ID Attack**

This attack is also called the replica attack, where the adversary copies the legitimate sensors or other entities and deploys them in the network (Conti et al. 2014). The adversary needs to acquire the secret values of them. This type of attack can be combined with a disrupting attack such as wormhole to create a breach in topology. Moreover, with a false package injection, one can collect sensitive data. Temporal pseudonyms and/or geographical information can be considered as countermeasures against clone attacks with communication delays and location spoofing are considered.

vii) **Blackhole Attack**

Blackhole is another attack performed on MANET and has been a threat to 6LoWPAN. When the adversary can spoof its identity in the network as a legitimate destination node, it can forge reply packets and disrupt the communications between the original source and destination nodes (Kurosawa et al. 2006). Without proper authorization on replies, the adversary can drop all the packets during the communication. This attack can be carried out combined with selective forwarding attack. In IoT, an attacker can send DIO messages to carry out Blackhole mechanism.

viii) **Denial of Service Attack**

Denial of Service attacks(DoS) is performed when the adversary wants to disrupt the network and make it available for a certain period of time or for good by flooding a particular server with packets (Mallikarjunan et al. 2016). It is one of the most well-known attacks on internet servers and wireless sensor networks. On IoT, this attack is performed by flooding one of the prominent nodes with IPv6 UDP packets. There are many intrusion detection systems (IDS) mechanisms proposed to detect and create jamming operations against malicious sources. Those attacks can aim to spoof the routes in IoT or replay routing information in order to create loops (routing DoS attack). Also the existence of Constraint Application Layer Protocol (CoAP) creates vulnerabilities on the application layer for attacks such as SYN message flood, protocol processing, proxy and caching, IP

spoofing, cross-protocol (application layer DoS attack).

ix) **Spoofing Attacks: Rank Attack**

The Rank attacks are designated for RPL protocols and focused particularly on the rank properties in RPL operations (Le et al. 2016). As the rank property provides effective topology formation, loop detection, and prevention of communication overheads. The attacks affect the QoS properties of the network. If the adversary is successful to claim a child node or introduce a reliable property to the network, it can change the traffic flow in the root direction.

x) **Spoofing Attacks: Version Attack**

These attacks are also referred to as version number attacks. It introduces a new formation of DODAG tree with a high version number to the network and this un-optimized version would cause disruption. The network QoS can also be degraded with this attack. It aims to drain the IoT resources.

xi) **Spoofing Attacks: Local Repair Attack**

There are two ways of performing this attack (Le et al. 2016). Firstly, similar to the rank attack where the rank is increased excessively and broadcasted to neighbor nodes, then the new rank information causes the nodes to seek for a new parent node. Secondly, changing the DODAG tree identity on the node in order to change the new network it belongs to. Then the child nodes of this parent require local repair operation to get a new parent. The nodes are able to verify whether they need to update their parent. At each attempt of local repair, the topology requires to be updated. Using local repair attack, the adversary will be able to drain the resources of IoT.

xii) **Spoofing Attacks: Neighbor Attack**

This is an RPL designated attack as the other spoofing attacks described here. Here the adversary forwards the replications of DODAG information object (DIO) messages to the neighbor without any modification. This causes the neighbors to assume that there is a new neighbor. Moreover, if they receive a good rank from the malicious node, they will modify their preferred parent to this node. This changes the DODAG topology. In the end, this attack will disrupt the route into a false one and use energy resources to the point of draining them completely (Le et al. 2016).

xiii) **Spoofing Attacks: DODAG Information Solicitation(DIS) Attack**

Here, neighbor nodes receive continuous DIS messages from malicious nodes. The first way to perform this attack is to broadcast the DIS message and the neighbors will need to reset their DIS timers and need to update the route information. The second way is to unicast DIS messages to a list of neighbor nodes. All the neighbors who received DIS messages will generate DIO messages. This will cause a communication overhead in the network and cause resource consumption.

3.3. Fragmentation in IPv6

In 6LoWPAN there exists a fragmentation mechanism which handles the fragmented frames by buffering, forwarding, and processing. The fragments of a standard IPv6 packet

include split packet parts and a fragment header (see Figure 3.58). The fragmentation of a packet is carried out when the original packet exceeds the remaining link layer data size. The fragment headers are in a fixed size. The fragment headers consist of dispatch value, datagram size, datagram tag, and offset. Only the first piece of the fragment does not have the offset value as it includes the packet header information. The offset is important as it indicates the order of the fragments to be able to reassemble them. All the fragments except the first one can be buffered or forwarded in any order. However, the first fragments include the packet information for early look-up operations.

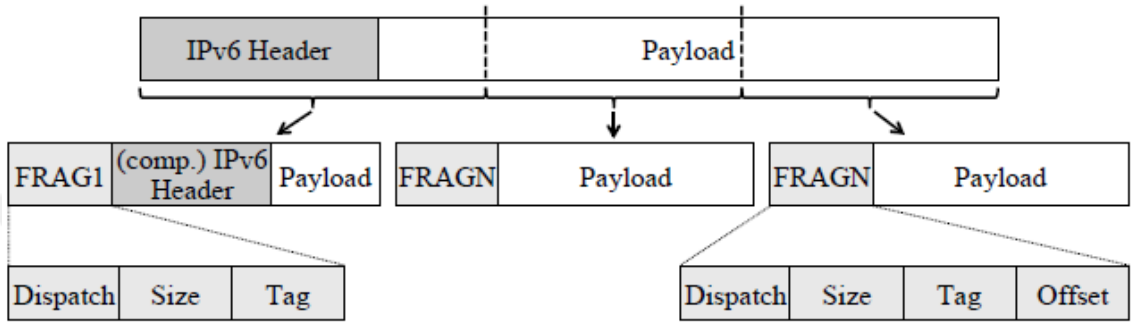


Figure 3.58. The partitioning of an IPv6 packet into fragments

The fragment forwarding mechanisms are mainly three. Firstly, the mesh-under routing where mesh routing head is appended to the front of all fragments. End-to-end link layer addresses are included in the header in order to use them during the link layer routing by the transmitting node. However, this does not support anything particular for fragment structure. Secondly, in the route-over routing, the packets are reassembled before routing. Nonetheless, the packets will need to be fragmented again during the transmissions between the layer. This mechanism is quite inefficient since the fragmentation takes place twice. Lastly, enhanced route-over routing, it takes the fragmentation headers into account. It works in a similar fashion with route-over, except a decision-making mechanism instead of reassembling the packets it checks the fragment pieces and transmits the fragments of the same packet at the same time.

Despite the comprehensive fragmentation support in 6LoWPANs, there is no authentication support in the layers; therefore, it is likely that the internal nodes will not notice the misconfigured or malicious fragments. There needs to be an additional timeout mechanism for incomplete, inconsistent and forged fragments.

3.4. High Level Approaches

3.4.1. The Hydra

Akram and Hoffman (2008) declared ten laws for context-aware and secure designs smart space and body area network architectures under the field of Ambient Environments and Ubiquitous Computing. The Hydra architecture is a hybrid framework composed of early security mechanisms and it is called the Hydra Identity Manager(HIM).

Hydra Middleware is composed of three main elements such as device elements, service elements, and user elements. Each element is separated into three layers; semantic, service, and network layer. Identity-manager is part of the semantic layer of user elements, along with orchestration- and context-manager. Service level consists of event-, service-, ontology-, crypto-, policy-, and profile-manager. Finally, there is only a network-manager in the network layer. Akram and Hoffman(2008) focused mainly on the identity manager of Hydra Middleware. The goal of the identity-manager is to provide efficient identity processing during the activity life cycle of users and things in the environments where Hydra architecture is supported.

Hydra architecture is based on the ten laws in order to provide consistent security, confidentiality, and integrity. Those laws are:

i) User awareness and control
ii) Supervised and limited access to sensitive data
iii) Non-repudiation
iv) Provision of directional identity topologies
v) Universal Identity Bus
vi) Definition of identity strength
vii) Separating the manager layer from the application layer
viii) The usability of the mechanisms for identity selection and their security.
ix) Coherency of integration between contexts
x) Scalability

Hydra Identity Manager (HIM) is composed of existing standards. It relies on the WS-* family for web compatibility. It works compliant with WS-Security, -Trust, -Policy, and -Federation. HIM takes advantage of Security Assertion Markup Language(SAML) which is built upon XML-mechanisms such as assertions, protocol, bindings, and profile. However, SAML alone is prone to phishing attacks for its open redirection to the authentication web site which is against the first law of identity. OpenID is utilized for its reliable community-driven platform for message exchanging with omnipresent identity structure. Aside from the similar phishing threat as in SAML, the identity is reused and the privacy will be compromised once it is captured. Windows CardSpace is developed by Microsoft in order to provide an access control mechanism between users and the Relying Party(RP). It is also integrated with SAML due to its compatibility with the WS-* family. Nonetheless, it is not complying with the fifth law of identity because it is non-comprehensive and only can communicate with WS-* standards. It also does not inspect the confidentiality of RP as long as the user accepts it. This problem violates the third law of identity.

HIM encompasses two main structures; HIM interface and Hydra Federation Engine. The federation engine has the Identity Provider (IdP) of SAML and OpenID in order to provide token service, RP of SAML and OpenID Consumer structures for consumer service, and Identity Selector of Windows CardSpace and Custom InfoCard structures for selector interfaces. The authentication of this engine is provided by OpenID and Windows CardSpace systems.

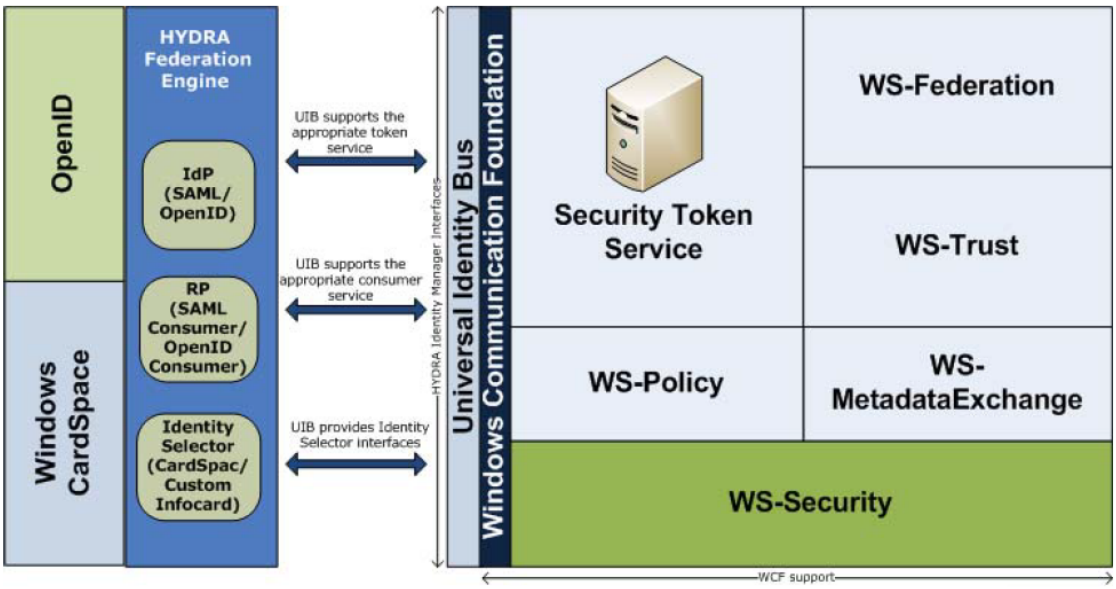


Figure 3.59. The HIM backbone (Akram and Hoffmann 2008)

HIM interface has an outer layer of Universal Identity Bus(UIB) in order to communicate with the Hydra Federation Engine in order to utilize its services and interfaces. The rest of the interface is composed of Windows Communication Foundation(WCF) structures including Security Token Service, WS-Federation, WS-Trust, WS-Policy, WS-MetaDataExchange and finally the WS-Security.

Laws of HIM are all provided by this identity framework. The first and the ninth laws are supported by mechanisms of Windows CardSpace and custom InfoCard. SAML helps the structure to comply with the second and fourth laws. Law three complies with the support of WS-Security specifications. UIB provides support for the fifth law of HIM. "Identity" namespace of HIM provides the strength of identity in order to satisfy the sixth law. Hydra Federation Engine separates the manager layer from the application (law 7) while providing scalability (law 10). Finally, the eighth law depends on the developers choice of custom InfoCard and the nature of the application.

3.4.2. The Identinet and Digital Shadow

Sarma and Girão(2009) proposed a secure IoT framework called Identinet and a digital shadow inspired conceptual structure. Originally some of the EU projects such as Daidalos (Aguiar et al. 2007) and SWIFT (Lopez et al. 2009) proposed solutions for heterogeneous and highly-constrained wireless sensor networks. The main concern in these proposals and related surveys is how to identify the entities in order to communicate them. Bunge (1974) declares the importance of a mechanism that defines the entities as true to their nature and role in the environment. In this work, it is elaborated that Data should denote the entity and designate an identity construct, while the identity should reference the entity.

Sarma and Girão note that, in order to create a structure to satisfy the expectations from the network, the designers should find an optimal consensus on the needs of stakeholders. The stakeholders can be identified as users who are interested in the service quality, providers who maintain the infrastructure and deliver services in order to do a profitable business, and finally the society and legal framework which deal with the privacy issues. It is important to address the needs and issues of all stakeholders. Sarma and Girao also point out that it is important to find solutions for possible side-effects during the analysis of trade-offs.

Identinet treats each person or thing as an end-point and denoted by a virtual identity (see Figure 3.60). Some of the devices in the structure can be intermediate points to join some of the end-points. Those interims are also identified in a similar way. It aims to provide a future global village in order to make recognition, integrity, and scaling easier by using these virtual identities. This interconnectivity allows ease at filtering and prioritizing the service consumers while treating them anonymously. Identinet also considers privacy and maintenance of the framework.

Digital Shadow is the virtual identity that has a presence in the framework in order to communicate and consume services. It also defines the functions regarding this identity. The digital shadow is the semantic representation of the physical nodes that build the framework. Access control and network maintenance are imposed on virtual identity. This identity includes information in order to define the entity. Thus, personalized privacy with required subscriptions, services, pseudonyms, preferences are defined.

As an example, passport or ID cards, operator contracts, service EULA, Drivers license, etc. can have legal representation of these real-world identifiers. In order to provide interoperability, social networks, service provider accounts, preferences, and attributes along with this legal representation can be translated into a digital representation. Through a series of filterings, these digital representations are formed into virtual identities which are part of the digital shadow.

Virtual identity allows the user or object to influence the behavior of the networks. On the other hand, it also helps with cross-layer and cross-domain integration. However, it is important to give limitations and control for influence and integrity. Scoping the information, general policies for service utilization, assessing the privacy levels are crucial.

The identity data may be originated from location, devices, services, and network before getting filtered and formed into separate virtual identities as shown in the Figure 3.61. Those can be used at several levels of digital shadow such as Session and Transport, Mobility, Terminal Capabilities, Data Storage and Filters, Network Access, and User Input and Output.

The proposed construction allows us to build seamless, consistent, and ubiquitous architectures. It is important to keep access control mechanism and authentication volatile and strong in the cryptographic context. Also, the problem of scope appears relative to

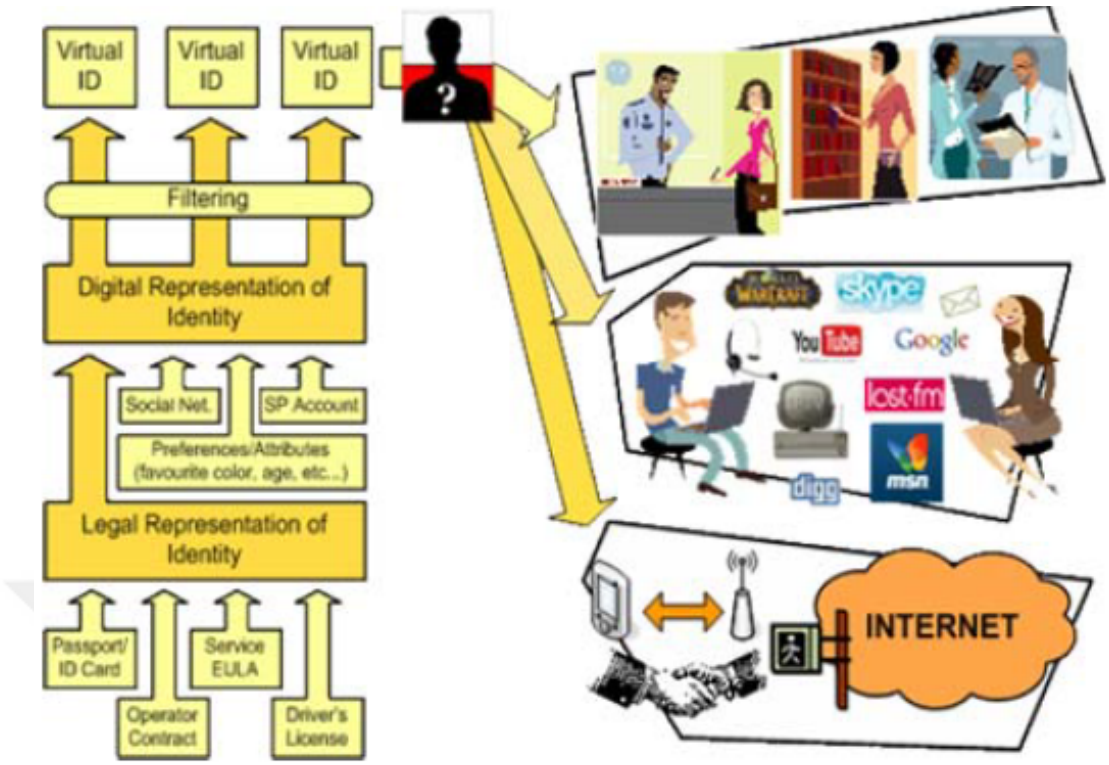


Figure 3.60. The User side of Identinet (Sarma and Girão 2009)

the application of nature and purpose in order to keep the computational load at optimal levels. The SWIFT architecture considers the distributed behavior for the propagation of data through the definition of separate domains which also allows hierarchical construction.

The deployment in SWIFT is separated into terminal node, certification authority for a root of trust, identity provider, enterprise, service providers, ID aggregators, and the visited network. The Identity provider is comprised of attribute server in order to store the virtual ID attributes and management, discovery server to look-up existing IDs and service information, and identity lifecycle manager for ID information management. Enterprise elements include attribute server, application server, identity aggregator for hierarchical deployment support, and federation manager for communication across domains. Service providers can be separated into two types. The first type includes data gateway for communication and transformation functions across different domains, attribute server, and federation manager. The second type of service provider consists of an application server and session admission controller. Session control functions are categorized under the session admission controller element. ID aggregators are comprised of a federation manager, data gateway, authentication server, identity lifecycle manager, discovery server, identity aggregator, session admission controller, and charging manager for accounting and billing functions. Session control functions are generalized under the name of session admission controller. Visited network entities include an access point for the logical entry point for the terminal in the network, federation manager, and network access router for

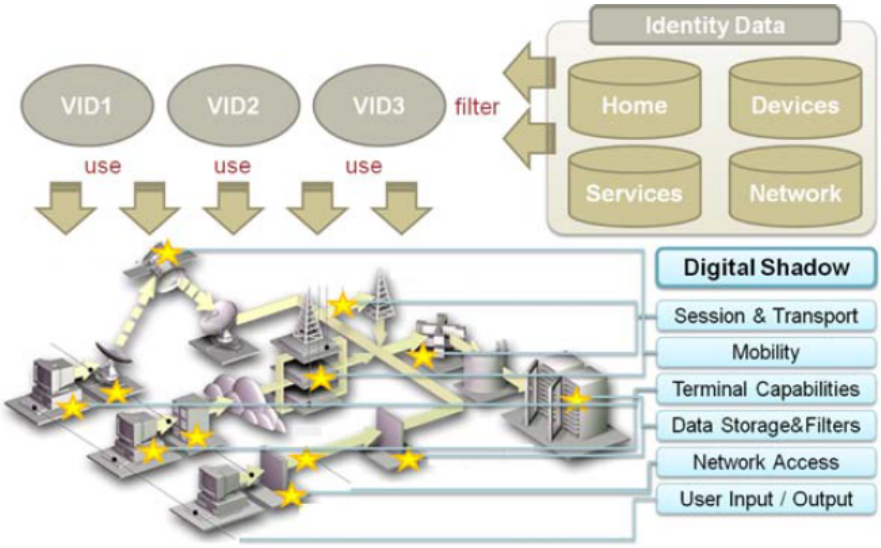


Figure 3.61. The Digital Shadow structure (Sarma and Girão 2009)

higher level functions for access control management.

The access control infrastructure is hierarchical and supported by several types of points. Policy decision point and policy enforcement points are cascaded and they allow cross-domain policy management. The policy administration point resolves the attributes of digital shadows designated for a given virtual identity. All of the entities admitted into the network are policy information points. User is introduced to the network through enforcement and decision points in gateway and directed towards operators, then services. Their service requests are accepted according to the administration and decision points associated with the service policy database into the IdM platform. The administration and decision points associated with the legal regulations database is applied to IdM platform. User sessions are managed with the data retrieved from the administration point associated with the user policy database. IdM platform is integrated with the authentication server, with a static attribute server, and context evaluator and contextual attribute server (see Figure 3.62).

3.4.3. A Holistic Approach with High Granularity and Context-Awareness

A holistic approach for RFID systems is proposed by Rekleitis et al. (2010) which focuses on the modularity, fine granularity and context-awareness in security and privacy policies. The protocol is built upon the back-end/reader and tag entities regarding the tag and reader authentication, secret delegation, and cryptographic operations for verification and secret message passing. Chien (2007) provides a classification for RFID authentication protocols such as full-fledged, simple, lightweight, and ultra-lightweight. Although ultralightweight protocols are desired the most, there are many proposals in the literature with successful analysis proved against them. The simplest implementations require the most meticulous countermeasures while being prone to a wide range of attacks.

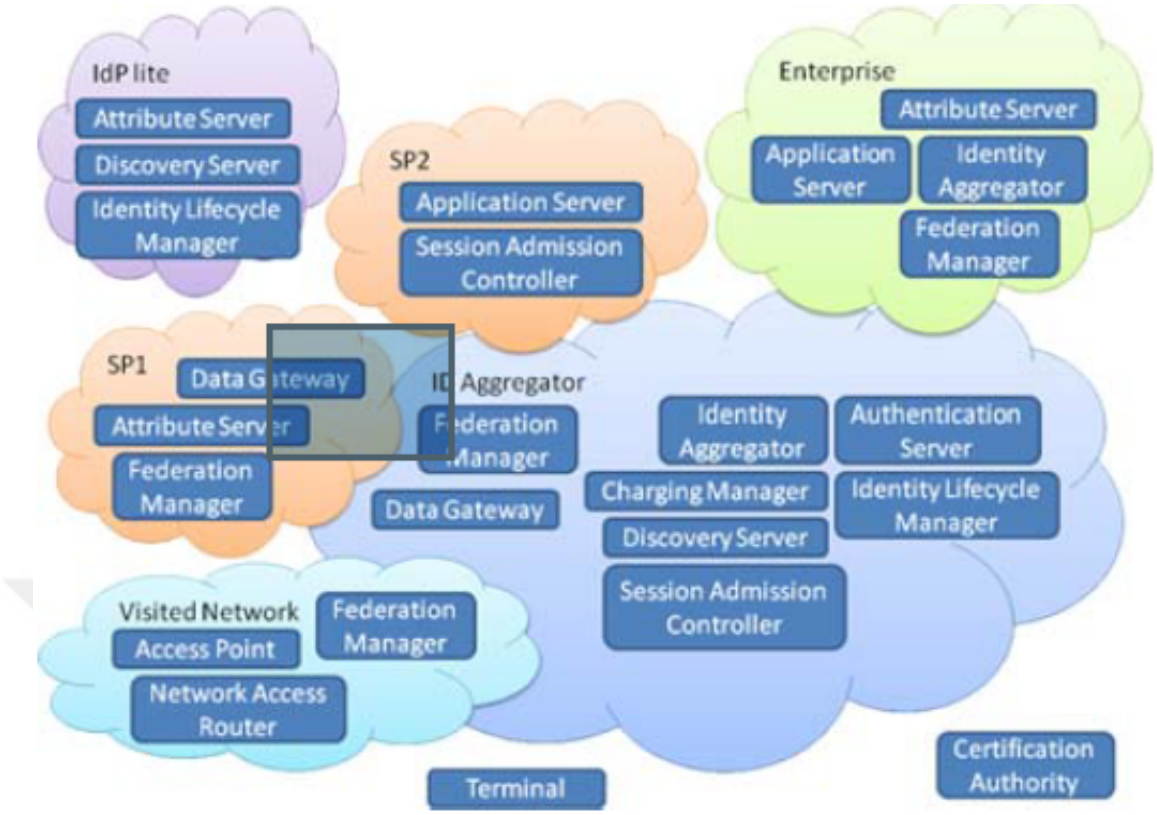


Figure 3.62. Identity management structure (Sarma and Girão 2009)

Rekleitis et al. propose important security requirements. Firstly, an attacker should not be able to authenticate an illegitimate tag masquerading as a legitimate one to a reader or an illegitimate reader masquerading as a legitimate one to a tag. Secondly, there must be countermeasures against DoS attacks to block the communications during information exchange sessions. These attacks are also regarded as desynchronization attacks. Thirdly, tag anonymity should be preserved through strict policies against forward and backward tracing attacks at a given time during the message exchange sessions.

The set of tag management operations for secure communication can be listed as tag authentication, revocable access delegation, ownership transfer, and permanent and terminal tag invalidation. This set of operations provide flexibility for access control management. It is important to build the tag operations considering the four steps RFID lifecycle which is the creation of tag and initialization of identity, attachment to an object and becoming an entity in the network, operation with authorization for access, and finally death and recycling.

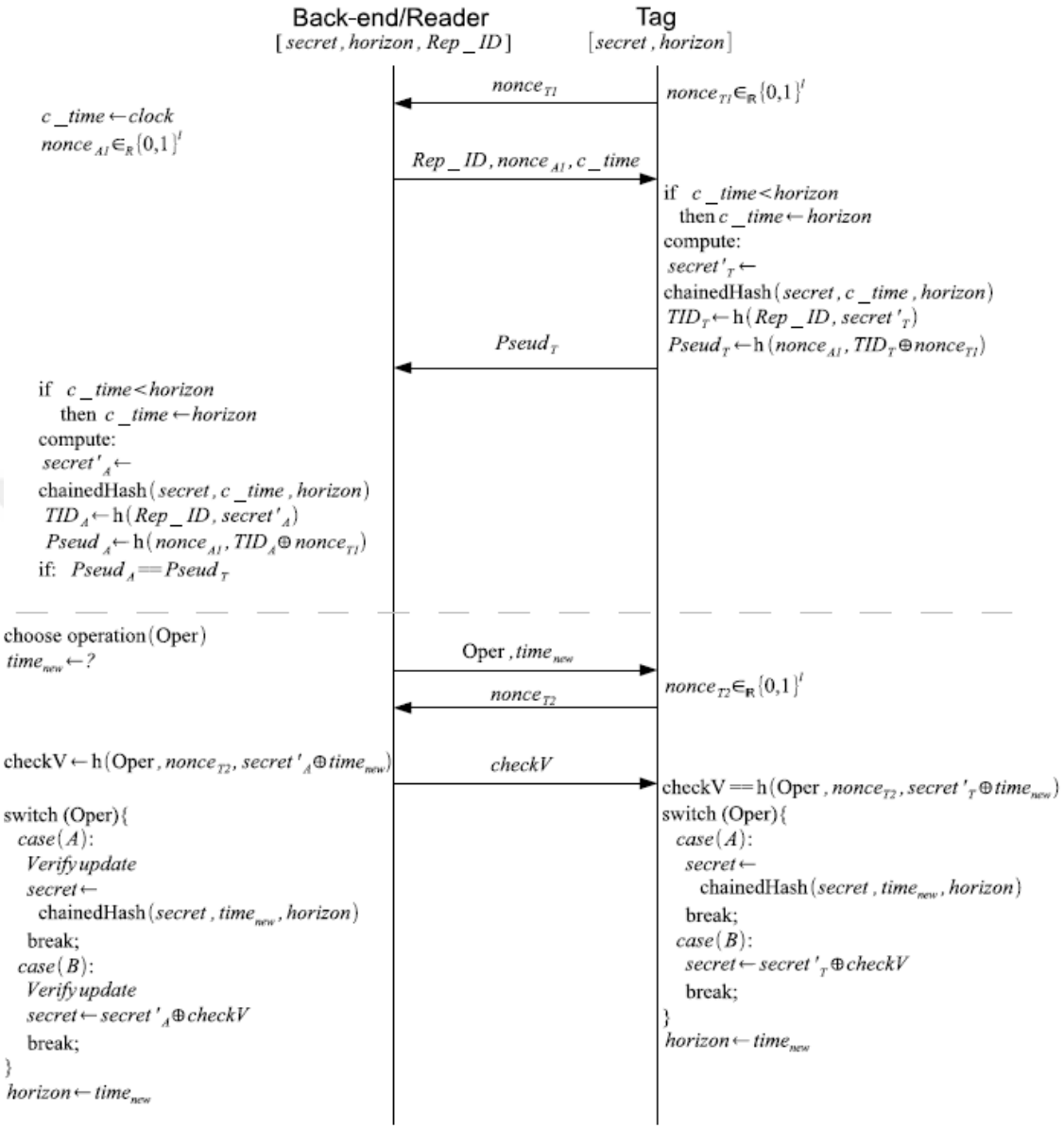


Figure 3.63. Tag Query protocol (Rekleitis 2010)

The approach of Rekleitis et al. is based on the steps of a simple protocol as shown in Figure 3.63. Firstly, at the initial layer, the tag will request access from the back-end according to the attribute that its representing entity holds. Then, a second layer will verify the authorization of the tag in order to reply to access deny or continue the operations. At that point, according to the existence of the tag entry, a relevant message is returned regarding its defined trust levels. Eventually, a policy will define the access levels for an authorized tag. Finally, additional security measures can be handled according to the defined policies for continuous access and limitations on the received information.

There are several crucial problems that the policies should address. Firstly, the efficiency of the operations according to the IoT framework. Well-defined and -expressed policies that are not necessarily technical such as XML or similar mark-up languages. Next, the complexity of access control that maintains the high-dynamic and interconnected nature of the RFID network. Also, information disclosure is mandatory among the participating entities. Finally, interoperability shall be cultivated according to the heterogeneous nature of IoT infrastructures. Additionally, fine granularity must be maintained through less general policies but well-defined groups without loss of computational efficiency.

Rekleitis et al. imposed that temporal pseudonyms are essential for anonymous tag delegations. In order to be able to keep temporal updates efficient, the use of one-way hash functions and lightweight pseudo-random number generator give good computational results. The protocol only requires a single secret value and a temporal value (private key) to produce "horizon" (public key).

The proposed protocol comprised of two phases which are tag authentication and tag data update. In the first phase, the authentication is initialized with tag generating a random secret and presenting it to the back-end. Back-end replies with its identity, a random secret, and the operation time. Then, tag checks whether the time that back-end sent is later than its horizon value. If it is true, it computes a secret key that with a chained hash function designated from the temporal values (horizon and the received back-end time) and its secret value. Next, it computes an identifier with this secret key and back-end identity as input to a hash function. The resulting identifier is used to produce a pseudonym along with the nonce received from back-end and its XOR with the initial nonce produced by the tag. Finally, in the tag authentication phase, the back-end computes the identifier with the hash function using using its self identifier and the stored time-dependent secret. The pseudonym is computed as it was computed by the tag with the resulting values. Finally, pseudonyms are compared. In order to prevent DoS/desynchronization attacks, this final step is computed twice when the authentication was unsuccessful.

The tag data update, the second phase, starts with back-end choosing an operation and forwarding the new horizon value. The tag generates a new random secret and presents it to the back-end. Back-end computes a checksum operation with a hash function using the operation indicator, the recent tag secret, and xor of its secret and the new horizon value as inputs. The result is forwarded to the tag. Tag updates the value by checking the checksum value with the correspondent values it has. Tag updates it secret value with chained hash and sets the current temporal value to the horizon. Back-end stores both the new and old values. These two phases provide all the tag management operations that were aforementioned.

The architecture is analyzed against the types of adversaries and attacks. The types of adversaries that are considered are weak (passive and active) and corruptive (forward, destructive, and strong). This type of construction can be strong against such adversary models such as tag and reader impersonation, desynchronization attack prevention, anonymity, backward and forward untraceability.

3.4.4. The Privacy Coach

Broenink et al. (2010) proposed the Privacy Coach, a privacy policy architecture with a focus on customer anonymity and prevent the clandestine inventorying with an abstract approach. It is a mobile phone application that provides a layer of privacy protection during RFID communications, mainly Near Field Communication (NFC) technology. It requires an internet connection in order to outsource the database.

The authors argue the impracticality of the Users License Agreement. Many of the users ignore the contents of the agreement and accept them without reading it thoroughly. This way, the user has no assurance for the quality of the service he/she is receiving or the protection of his/her personal data. They also indicate that it is important for users to build, program, and modify their own IoT network. They should also be able to assess the privacy policies for the entities in their network as flexible as possible. They must be able to personalize their access or create groups for common applications or entities which require specific policies. This application provides an agency layer over their personalized IoT networks through matching policies according to their privacy preferences.

Ubiquitous Communicator (UC) of Japan and The RFID Guardian proposed by Rieback et al. (2006) are the main inspirations of this architecture. UC is developed by Ken Sakamura for users to have access on tags on objects and landmarks in cities. It has an uncommon approach to RFID technology. However, it provides a variety of efficient applications such as information about important landmarks in cities, animals in the zoos, arts in exhibitions, mobile payment for shopping, and local directions and timetables for public transportation. Everything has a 128-bit unique code and there is an open U-code database for information and transaction queries. On the other hand, the RFID Guardian is a very simple approach allowing only the devices with predefined policy to read the tags and jamming the rest of the readers. It is similar to a gateway from readers to tags. However, the tags are open to readers without any proper privacy policy.

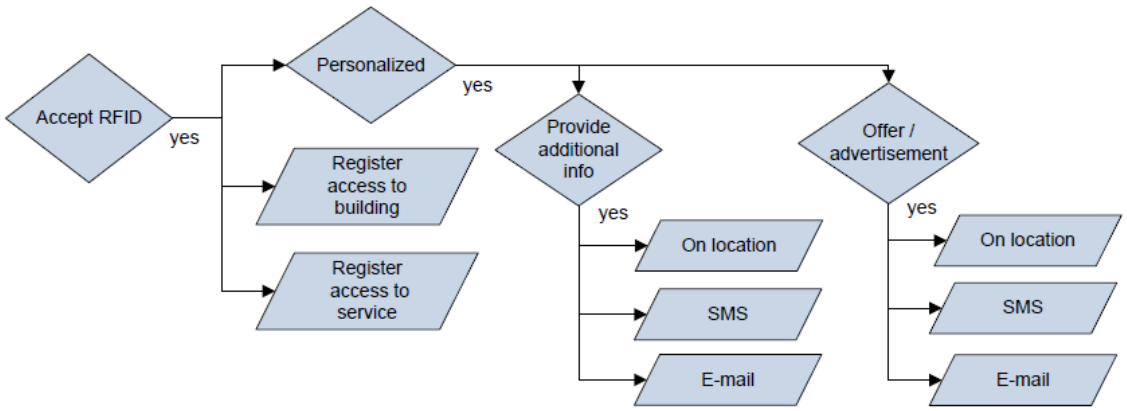


Figure 3.64. The control flow for ID badges (Broenink et al. 2010)

The Privacy Coach stores the tag policies in a remote server and allows it to communi-

cate with the applications installed on the consumer mobile phones. The mobile application stores the policies defined by the consumer. In order to communicate with the RFID tags, users make a request to the tag policy database server with the tag number and their consumer identifier. There are mainly two profiles for privacy; ID badges for consumers and RFID tags for the objects as shown in Figure 3.64 and Figure 3.65. After authenticating the tag with the database server, the consumer ID badges have three operations. Register access to a service, register access to a building, or personalized information about general updates and offers from an object in close proximity, SMS, or e-mail. The RFID tags can be used by supply chains, local administration offices, etc. One should be able to validate the RFID information with the tag policy database. The tags can provide price information, inventory control, and additional general information. The consumer profiling could be done anonymously or personalized through a simple agreement. The users allow their profiles to be used openly can access further information about the product and get notifications on promotions from nearby relevant RFID objects, SMS, and e-mail.

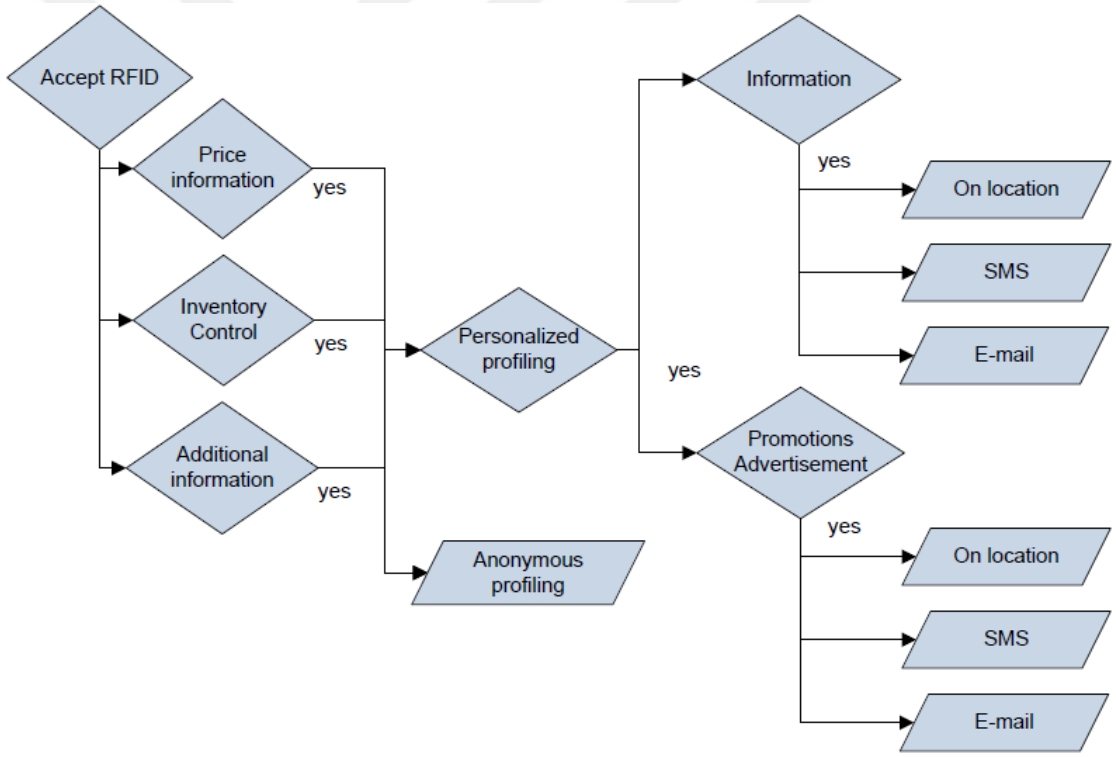


Figure 3.65. The control flow for RFID tags (Broenink et al. 2010)

The mobile clients of Privacy Coach should be able to configure their own profile, read objects through an RFID tag and assess their policies. They should be able to see the profiles and products they can view by simply getting near to the tag. The matching of profiles, tags, and policies should be conducted without loss of security, availability, and integrity.

However, this approach only gives a skeleton of a possible architecture without any further concerns about security. Access control and authentication mechanisms should be cryptographically supported by one-way hash functions and handshake protocols on all entities: tag policy database server, mobile profile, ID badges, and RFID tags. Also, the designers should be able to assess the workloads on more computationally strong devices.

3.4.5. Fingerprinting and Profiling

In order to provide well-defined high-level security and privacy approaches Radomirovìc (2010) proposed dense Internet of Things architecture based on several assumptions and contemporary Internet of Things standards. Their approach is based on the similarities between the evolution of IoT and operating systems. It is noted that although most of the current security attacks are threats in the long run with a more complex IoT network topology. Those attacks will be more effective and the future architecture designers will no longer be able to leave them without proper countermeasures. The phenomenon is similar to the honeymoon effect (Clark et al. 2010) in software design.

The dense IoT is a network to observe the capabilities of a possible adversary model. It is based on two main assumptions; the connectivity in this network is high-degree and every item is ubiquitous. It is important that each item have communication capability independent of their purpose. Although Radomirovìc aims for a general-purpose architecture, he also designates his work in a three-step deployment where the construction starts with a specific purpose environment. This allows him to observe the problems in an organized fashion, therefore it will able to provide an adequate solution. This approach avoids the honeymoon effect as well.

Here, security elements are defined as a fingerprint which is an identifying characteristic of an entity in the system. These fingerprints can be dynamic and change with time and provide more identifying characteristics. Moreover, an entity can have several of them to create a profile. Nonetheless, the operations consisting of observation of the characteristics, analyzing the dynamics, and constructing a more elaborate definition for an entity by exploiting the analysis results are generalized under the name of profiling. The characterizations can be physical measures of devices as in implementations of PUFs and lightweight random number generators with the use of LFSR as explained in early sections. They could also be a speech pattern, specific fingerprint bacteria to a specific person, internet browsing history, or other parameters that can be obtained from the IoT users. Fingerprinting and profiling are passive adversary models. In fact, most of the security models today are acquired from the analysis of adversaries over time.

The constant arms race between adversarial models and security architecture improvements is also observable in operating systems. The operating systems constantly update malware detection software or firewalls. As they advance towards the end-user needs, the scanning and filtering mechanisms for malicious intruders are improved. Similar viruses can intrude our daily lives through physical or software-based interactions. Thus, the Trojan horse eventually becomes an entity materialized in the physical environment as in Homer's Iliad. Radomirovìc points out that, there should also be additional designated measures for the IoT systems due to their physical structure and location aspects.

Here, it is noted that every two nodes in the IoT network should be able to communicate with each other despite the contextual difference. There should be "commonplace" functionalities for each node and those should be applied ubiquitously. The fact that the node is either a passive RFID tag or a reader with high-computational capabilities should not affect these features. The implementation of such features provides the dense Internet of Things model. However, due to the computational and energy limitations, there should be a limitation to connections. The topologies should be designed with this issue in mind. Consequently, the network will run efficiently in asynchronous form.

The fingerprinting and profiling operations aforementioned can be applied with the use of "hardware tokens". They should be unclonable and their impersonation should be ineffective. Those hardware tokens should be physically protected in order to prevent fraud devices to create system breaches through jam signals. The privacy measures should be renovated actively due to countermeasure against the erosion of security against adaptive social engineering attacks.

The idea proposed here can be extended with more adaptive approaches with formally defined secure authentication mechanisms, modeling against more sophisticated attacks, and protocols considering the physical attributes of the nodes and the IoT network infrastructure.

3.4.6. Network Admission Control

Oliveira et al. (2013) proposed an access control mechanisms that are based on 6LoWPAN Neighbor Discovery protocols where each node has a control list including the rules on its own. The main design pattern is based on administrative approval in order to prevent foreign third party entities to access the network. The architecture has risen from the common phenomenon in wireless sensor networks, where it is important for each entity to be able to organize, configure, optimize, and heal itself. Additionally, they must be able to manage their fault-tolerance. However, in order to provide these capabilities properly in 6LoWPAN, the designer must consider the workload and robustness.

As shown in the Figure 3.66, the system architecture is mainly built upon three entities; those are sensor nodes within 802.15.4 standards while supporting RPL and 6LoWPAN Neighbor Discovery protocols, 6BR, and a management station for remote administration approval. Each sensor node has an authorized node list and an agent application. Those modules communicate with 6BR through UDP communication. 6BR is a gateway that has a management server application, a list of nodes that are authorized within the network, and a list of pending nodes. The management stations those can communicate by IPv6 and Internet protocols use Management Client application to communicate with the Management Server application within the 6BR through TCP communication.

Oliveira et al. (2013) defined four steps to node administration. Those steps are the detection of node presence when it enters the network, the authorization of the node, authorized node list propagation, and data filtering. The Neighbor Discovery protocol detects the node presence. When a node is detected, an administrator manages its access control mechanism and authorizes it through communication between the Management Client and

Management Server applications. Management Server propagates the authorized node list by messages through UDP. Data filtering is carried out by using the authorized node lists. The pending list is used for nodes that are waiting to be authorized.

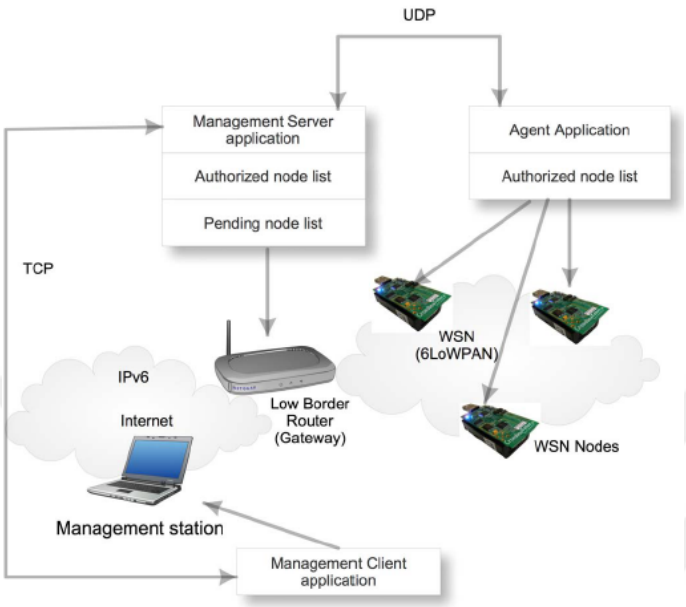


Figure 3.66. Oliveira et al. (2013) communication mechanism

During the node present detection, 6BR initialized a node and associates link-local address. The node to be authorized sends a Router Solicitation message to 6BR to be replied with a Router Advertisement message by it. Router Advertisement message includes self-configuration information. When the IPv6 address is configured, the node sends a Neighbor Solicitation message with address registration option to 6BR. 6BR registers the authorized node.

The node list propagation carried out on a UDP connection between 6BR (Management Server application) and node (Agent application). A complete list is broadcasted when a new node requests for authorization. The Agent application of the pending node responds to Management Server with an ACK message. 6BR also handles the error messages.

The data filtering operation (see Figure 3.67) is initiated with the arrival of a new packet. Node detection checks whether it is RPL or data control message. If it is a data control message the packet is forwarded and the filtering is complete. If it is an RPL message, the system checks whether the source is already authorized. If the source is not authorized then the packet is discarded. The authorization of the destination address is checked. If it is the same as the source address the packet is processed. Otherwise, the authorization of the destination node is controlled. The packet is forwarded if the destination is authorized. If there is no authorization for the destination node, the packet is discarded.

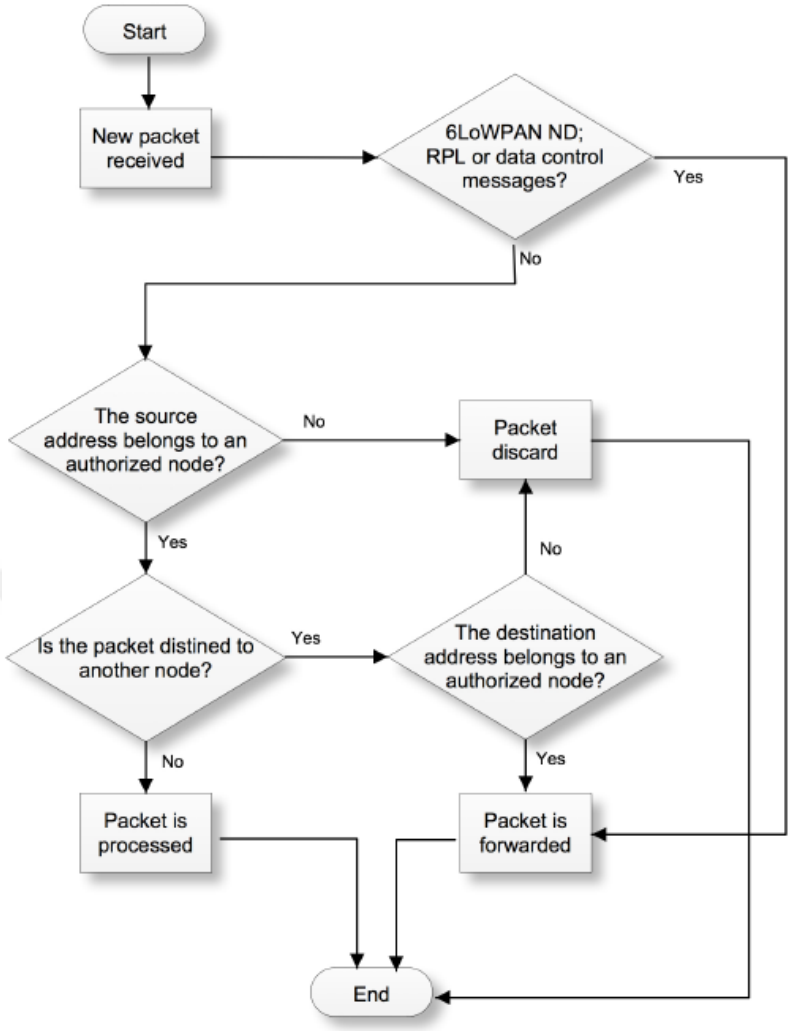


Figure 3.67. The flow diagram of packet processing (Oliveira et al. 2013)

The testbed was processed with three wireless sensor nodes, a low 6BR as a gateway, a management client through Ethernet. The application was developed on Java IDE NetBeans and performed on Ubuntu 10.0.4. The access control mechanism requires additional security measures as described in early architectures. Also, the access control mechanism is overly manual and packet processing delay affects the communication performance.

3.5. Intrusion Detection Systems

The intrusion detection systems(IDS) are classified into two main categories: Host-based and Network based. Host-based IDS monitor the network behavior on running applications in order to find a malicious activity (Wagner and Soto, 2002). They check the internal operations such as system calls in the operating systems. It is inspired by the human immune system. During the application process have regular activities a system can be modeled to use as a reference. It traces the system calls during the system is not under the attack and proceeds with the learning phase. There are subtraces produced from the regular system states and variables. Outside the learning phase, when the IDS finds a subt-

race that is not stored in the system, the source of the subtrace is considered anomaly. As the number of anomalies increases the system alarms and requires attention for a malicious intrusion. These IDS are good at determening the damage on a network, while they're not good for real-time intrusions. They check the malicious adversaries from inside.

The Network-based IDS analyses the event within the network such as traffic volume, IP addresses, service ports, etc. The retrieval of packet headers for anomaly and the acquisition of network status information provide properties to analyze the traffic. The packet includes a lot of information about regular activities. The analysis of all this data provides a communication pattern in the regular system which seldom deviates. Network-based IDS are good with real-time intrusion detection. However, as opposed to host-based IDS, they are not good at determining the harm the network has received. Network-based IDS are good for external attacks.

The intrusion patterns are separated into misuse intrusion and anomaly intrusion. In misuse intrusion, the IDS controls the pattern of intrusion. The source data contains signatures like description which are comparable to the signatures in the system. The IDS using misuse pattern are mainly role-based solutions. In anomaly intrusion, the IDS observes the communication traffic behaviors. These IDS systems keep profiles of traffic for regular uninfected operations. Therefore, the anomaly pattern can be compared.

There are several methods followed in order to build an appropriate IDS. These are as follows:

i) **Statistical Analysis:** These techniques can be applied to wormhole attacks. They observe the delays between the hops to find the links belong to the wormhole origin and destination nodes. IDS collects the information from multi-path routing. However, mainly these mechanisms only detect the adversary. There is no further prevention or healing mechanism. Statistical analysis can be coupled with other methods in order to acquire these feature (Song et al. 2005).

ii) **Evolutionary Algorithms:** This method evaluates rules in the network and creates a routing path for optimal safe communication. Using genetic algorithms it can differentiate anomalies such as intrusion and errors. The most common evaluation parameter in the mechanism is the IP address.

iii) **Protocol Verification:** Protocol verification method relies on the established protocol standards in order to conform to commercial demands. The idea is simple, once the node does not follow a given standard. However, an adversary familiar with the protocol can easily impersonate a legitimate node.

iv) **Rule-based:** Here, the IDS following anomaly intrusion pattern can build a model of a finite state machine with signatures and analysis transitions. This model is based on the non-malicious communication pattern. All the packets inside the network can be tested in this finite state machine model. Eswari and Vanitha (Eswari and Vanitha 2013) proposed the rule-based IDS method with three steps of the process. Firstly, the origin of

incoming messages is verified. Then, rules are applied to the communication pattern. And lastly, the routing attack is detected by validating the source of packets.

v) **Artificial Neural Network:** Artificial Neural Networks (ANNs) have shown good evaluation performance for pattern recognition researches. Their feedback propagation techniques can give good analysis results and decisions for unseen events. The intrusion pattern can be supported by ANNs using these properties. False-positive and true-negative results can be used as feedback data to improve the network through an activation function. The gradient descent will give more elaborate intrusion patterns. Supervised and unsupervised learning techniques can be applied. Both of the methods have input and output pairs. However, during the training phase supervised learning requires a classification variable which is manually applied. On the other hand, the unsupervised learning technique finds patterns in the existing data.

3.5.1. DoS-based IDS

Kasinathan et al. (2013) proposed a semi-decentralized misused-based IoT security backbone. Their proposal aimed to detect DoS attacks and evaluated by jamming DoS attacks. It was deployed in ebbits, a European project focuses on standardized end-to-end mainstream enterprise applications for efficient heterogeneous and fully interoperable environments. The architecture was built upon the idea of preventing the attacks against the Quality of Service(QoS) of the network.

The DoS IDS architecture is an extended version of existing ebbits infrastructure (see Figure 3.68). It is separated into two layers; Internet of Things layer and Physical World layer. The network manager is part of the IoT layer and the extensions of DoS IDS with nodes are part of the Physical World layer. Physical World Adaptation Layer (PWAL) including the 6BR stands between these two and acts as a gate between the network manager and IDS. Network Manager is separated into Opportunistic Manager, Network Management, and Security Manager units. Network Management unit deals with network performance information such as parameters, analysis, traffic states, physical states, and collisions. It keeps the availability of information flow between the Opportunistic Manager and Security Manager.

Opportunistic Manager in the Network Manager is an optimization unit of ebbits framework. It provides good communication performance by dealing with delay inconsistencies in real-time using Frequency Agility mechanism. Security Manager of ebbits communicates directly with Network Management unit and Frequency Agility of Opportunistic Manager unit. Kasinathan et al. (2013) implemented the DoS Protection Management mechanism into this unit.

The DoS Protection Management receives the alerts directly from the IDS from the Physical World layer. It is a verification mechanism for an existing attack through analysis of IDS results. Part of the IDS operation is here; therefore, the architecture is partially centralized.

PWAL communicates directly with the Network Manager and indirectly through the

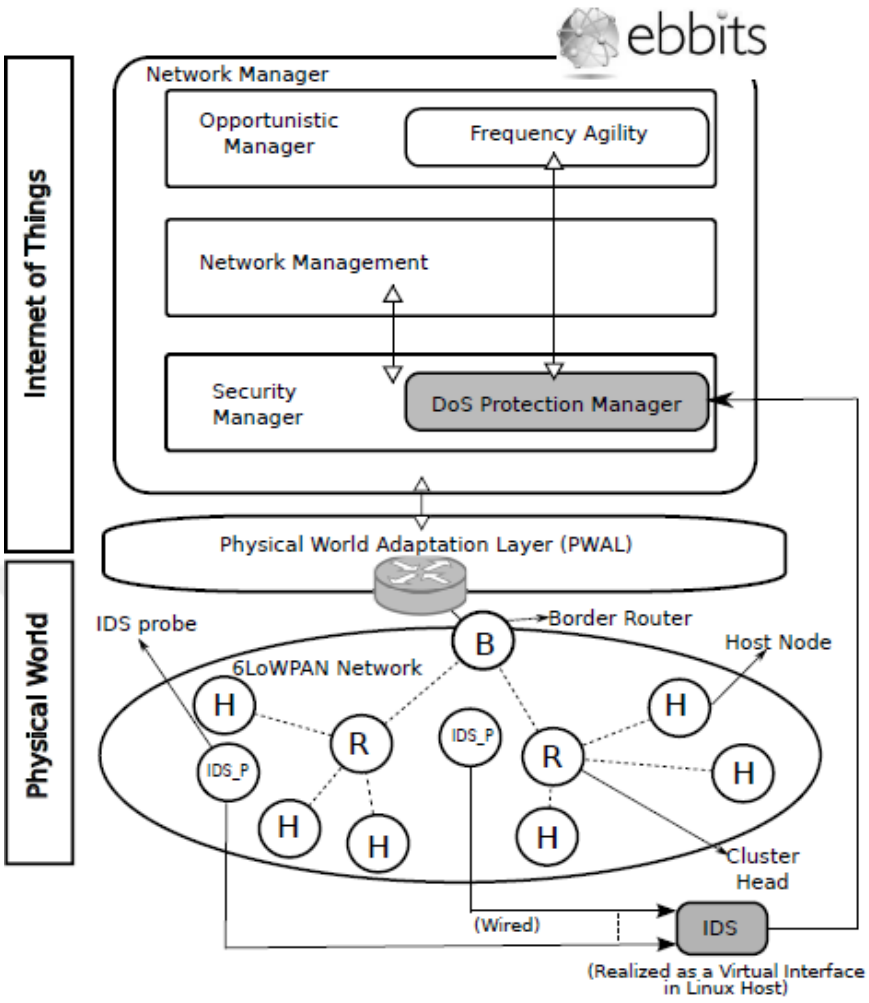


Figure 3.68. DoS detection backbone (Kasinathan et al. 2013)

6BR. 6BR is connected to the cluster nodes which group some of the host nodes. Within the area of 6BR, there are IDS probes that collect network information from the sensors and other smart things. The IDS probes are wired directly to the network-based IDS(NIDS) mechanism which is implemented on Virtual Interface in a Linux host device. This NIDS detects misused intrusion patterns. The more the network is complex, the more it requires IDS probes for more detailed information. Each probe is able to read all the information on the communication channel. NIDS is directly connected to the DoS Protection Manager to feed it information.

The main mechanism follows the Jamming attack threat. The jamming attack is a type of DoS attack where there is an interference of radio signals either continuously or with recurring periods of episodes. The operate at link-layer during the running time of the network and can create packet collisions. In the end, the target network receives damage and cannot operate efficiently if it is not completely disabled. Once the IDS raise a signal for intrusion, DoS Protection Manager checks the level of interference and compare it with the information from Frequency Agility in the Opportunistic Manager. Then, the

ratio of packets that are not transmitted is controlled. The inconsistencies between the real-time state and recent update in the network management indicate the possibility of an attack.

The IDS is called Suricata with main components of the information capture device, decoder with the application, network, adaptation, datalink, and physical layers. Applies predefined rules of a signature set and the decoder results as inputs to a detection engine. The detection engine results alert. The detection engine is comprised of an event generation mechanism and custom modules. Suricata decodes the captured information according to IEEE 802.15.4 (layer 2) and 6LoWPAN standards.

IDS probes are part of a Penetration Testing System. They are sniffer/injector devices communicates with the virtual interface through the USB port and receives Metasploit information. Tosca packet handler is used. The number of Metasploit IDS probes is directly proportional to the increase of true positive alerts from the same number of intruders.

3.5.2. SVELTE

There are several standards existing for IoT security such as IPsec, DTLS, and IEEE 802.15.4 link-layer security. There are existing mechanisms, those support the message security standards, cover the cryptographic encryption schemes and authentication algorithms which are lightweight and reliable in IoT framework; however, many types of attacks exist in the literature where the communication channels can be invaded due to the physical and topological structure.

Raza et al. (2013)designed an intrusion detection system, SVELTE that mainly focuses on sinkhole and selective forwarding attacks. The main goal is based on several assumptions defined in IoT frameworks. 6BR is always available, message security is existent and there is a global addressing method. In addition to the common features of "things" in IoT definition, they are assumed to communicate over lossy channels with IoT-based protocols; CoAP, RPL, and 6LoWPAN.

The backbone of the system is given in the Figure 3.69. Each node has a network stack which consists of IDS-modules defined by SVELTE such as packet loss, firewall alarm, and the dedicated 6Mapper component. Those nodes can directly communicate with each other through paths. There is a child-parent relationship. 6BRs are dedicated to path maintenance. A border-router, 6BR, has additionally 6Mapper, spoofing and alteration detection, node availability control, graph validity control, and end-to-end pack loss detection mechanisms. Also, a mini-firewall is implemented.

SVELTE architecture is a hybrid pattern matching taking advantage of both misuse and anomaly patterns. They avoid the MAC and IP address spoofing, the authors follow a node ignoring approach. The IDS systems follow either blacklisting to prevent the malicious nodes or whitelisting to only allow the legitimate nodes.

Raza et al. (2013) built the architecture in a way that the setup is semi-decentralized. IDS modules are both in the nodes and 6BR. The centralization of the approach is ma-

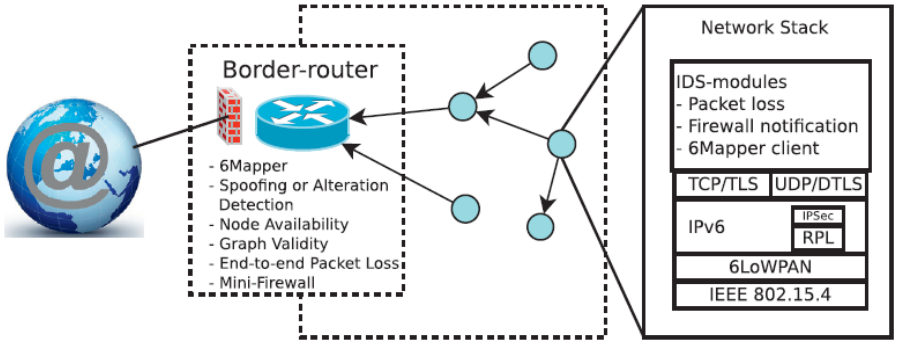


Figure 3.69. SVELTE architecture (Raza et al. 2013)

inly provided by 6LoWPAN Mapper (6Mapper). 6Mapper provides a topology for RPL network from the analyzed information. Intrusion detection component is also a centralized structure. And finally, the mini-firewall, where the nodes receive filtered data from its mechanism, allows nodes to carry less data through-out the communication in order to preserve resources.

The packet format of the mapper is given in the Figure 3.70. The package includes the origin node identity, RPL instance ID, DODAG tree ID, version number, timestamp value, parent node identity, and neighbor information. Neighbors part of the packet has existence check, neighbor identity, rank, neighbor identity, rank,... Timestamp value provides the freshness property of the communication time in order to prevent the processing of outdated packets. The entire packet is 5 bytes in total.

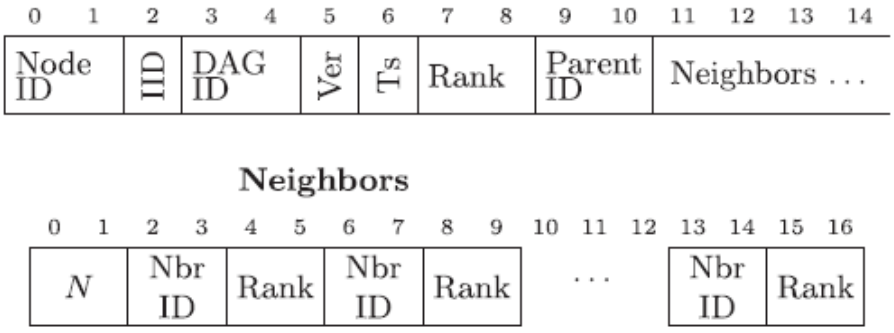


Figure 3.70. The packet format of SVELTE mapper (Raza et al. 2013)

The main purpose of the 6Mapper is to form RPL DODAG tree for 6BR and update the neighboring and parent nodes at each node. If the 6BR is the same as the DODAG root node, there is no point in adding the destination IP in the header. Also, timestamp might be unnecessary due to the use of CoAP with its acknowledgment messages. Moreover, an adversary should not be able to distinguish 6Mapper packets from the other packets in the communication channel. This can be achieved by encrypting the messages and not

revealing information at headers. However, the adversary can compromise the network by analyzing the behavioral and signature patterns similar to methods in IDS with more resources.

The intrusion detection system is focused on sinkhole and selective forwarding attacks. Raza et al. (2013) exploit the faulty rank information and the difference between the two reported ranks in order to distinguish the consistencies in the network. All nodes are visited for faulty nodes and agreements for consistency. There is a given faulty threshold value to determine the final consistency level. The nodes are not immediately removed from the whitelist in the case they raise the inconsistency. If their effect on the network is continuous after several times, then the nodes are removed.

The intrusion detection also concerns with the node availability. The DODAG root indicates a starting point for the available nodes. However, this does not only cancels out the compromised nodes but also the temporarily unavailable nodes. The collection of previous RPL messages helps to filter out those unavailable nodes.

It is important to identify whether the topology is changed by an illegitimate source. These activities indicate the sinkhole attacks. SVELTE can analyze those attacks by detecting incoherency from child-parent rank value differences. However, it is likely that intrusion detection can raise false positive alarms. Therefore, a threshold should be decided for continuous incoherencies. There are also end-to-end packet loss and Sybil attack countermeasures. These give the IDS dynamicity of packet authentication.

The third feature of centralization is distributed mini-firewall. A third party firewall will not be able to adapt to the SVELTE structure and distinguish legitimate nodes. It is a simple filtering mechanism for both external and internal(impersonation or node compromising) attackers. The destination host has local and global filters to analyze the active communicating nodes. Each node can have its own filter and modify it according to the activities. The global filter is a set of all external hosts to be filtered, while the local filter is a set of mapping of external nodes to the set of local nodes. Each active node can contribute to reporting by blaming an external node. If the filtered nodes at eventually surpass the reporting threshold during the iterations, those nodes will be removed from mapping and end up filtered.

SVELTE is implemented on the Contiki operating system. 6Mapper is implemented natively using a serial socket of Cooja. SVELTE shows good performance against sinkhole and selective forwarding attacks in lossy networks. The energy overhead is negligible for small networks; however, it gives severe results in large and complex topologies. 6Mapper consumes more energy than mini-firewall and packet loss correction, almost three times.

3.5.3. VeRA

RPL protocol has no intrinsic countermeasures against version attacks. Dvir et al. (2011) proposed a protection IDS scheme in order to prevent such attacks called Version Number and Rank Authentication(VeRA). The architecture is built against an adversarial

model. The model impersonates a DODAG root and modifies either the version number or the rank number. Although the adversary can move more towards the root by increasing the rank value, it is more effective to decrease it to allow more nodes to connect to root through it.

VeRA aims to prevent compromised nodes which send illegitimate DIO messages which masquerade DODAG tree root or manipulating the traffic and eavesdrop by acting like a legitimate intermediate node forwarding messages towards the root. The protocol mainly consists of initialization and version number update phases.

During the initialization phase (see Figure 3.71), first, the DODAG tree root generates a random number in order to create a hash chain using the root of the version number. For each value of the version number hash chain, a random number is generated. These random numbers are used to create a Rank value with a given maximum rank value. A rank value is generated with the root value and objective function. A signature value is produced from the difference between the resulting rank and the sender rank. Then the root calculates a signature value and checks the freshness of rank value. If the timer value is expired requests a new value from neighbors with DIO messages. The first iteration of the rank is applied to a one-way hash function and MAC value is produced. The root sends the MAC value, initial Version Numer, root version number, and IP to the neighbor nodes. Upon receiving the signature DIO message from the root, the node verifies the signature. If it fails the packet is dropped. Otherwise, it forwards the DIO to its neighbors, and so on.

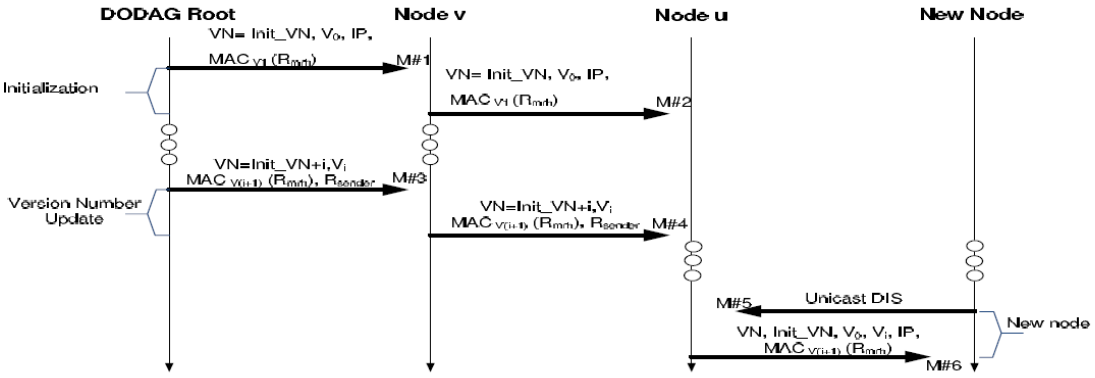


Figure 3.71. The diagram of security protocol (Dvir et al. 2011)

During the version update process as shown in Figure 3.72, root gets notified about the modification. Once the alarm is raised, it calculates the rank with its root value. Calculates the sender rank value from the one-way hash function and calculates the maximum rank value from the hash chain. Finally, the root forwards the MAC of calculated hash and sender rank to its neighbor as a DIO packet. The neighbor nodes upon receiving the DIO messages verifies the received value with its counter hash function. If it is a valid version number, the node produces the control rank value by calculating a hash value from the

sender rank value. The chain is as long as the hop distance between the verifying node and the root. MAC values are calculated and there must be a monotonical increase in the rank value at each hop away from the root(each iteration). The deviation is verified and MAC and calculated Rank values are propagated towards the neighbors.

A new node unicasts a DIS message towards root through its nearest neighbor. Neighbors reply with verification message. All the local rank and version numbers are stored at their nodes.

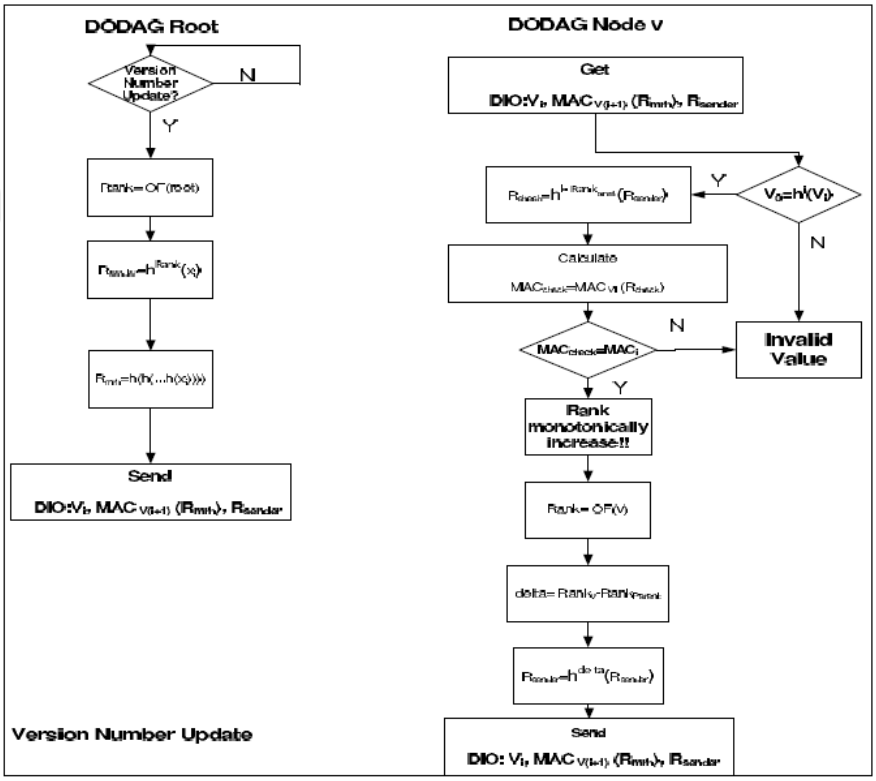


Figure 3.72. Version number update flow diagram (Dvir et al. 2011)

The building blocks of VeRA are Secure Hash Algorithm (SHA) (Eastlake and Jones) for a hash function, Keyed-hashing for Message Authentication (HMAC) for MAC function (Krawczyk et al. 1997), and RSA (Rivest et al. 1978) or Elliptic Curve DSA for signature operations. The operations are mainly divided into Version Number Authentication(VNA) and Rank Authentication(RA). During VNA on a root, an elliptic curve DSA (ECC) for once and SHA hash function for all connected nodes are applied. And while the intermediate phase, an ECC for version hash chain and O(n^2) of SHA hash functions are calculated.

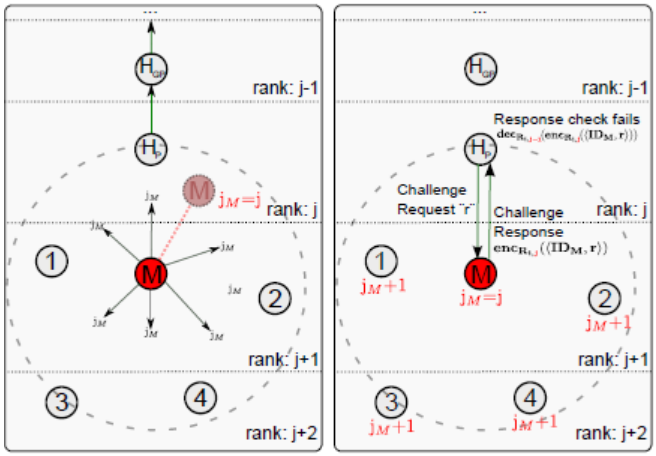
At RA, for each version update, a MAC and chain length of SHA operations are applied on a root. It is similar to the intermediate node operations. However, the SHA operations maximum rank hash values of the length at each iteration are excluded.

The simulations give similar RA performance results for root and intermediate nodes. VNA simulation has a big performance difference due to the number of iterations. RA overall requires more time for processing.

3.5.4. TRAIL

The VeRA architecture mainly concentrates on rank attacks however forgery attacks with rank order and jamming attacks are still a great threat. The hash chain can be forged by two-backward without a proper encryption method. VeRA is also vulnerable to replay attacks that target the rank numbers. It requires cryptographic primitives forming credentials that are independent of the sender.

Perrey et al. (2015) propose an improved version of VeRA against attacks above and a form of IDS that is called TRAIL. It is shown that the rank of spoofing attacks can attract neighboring nodes in order to prepare for a sinkhole attack. Also, in a replay attack, the malicious node probes the parent node to attract the neighboring nodes and create an upstream.

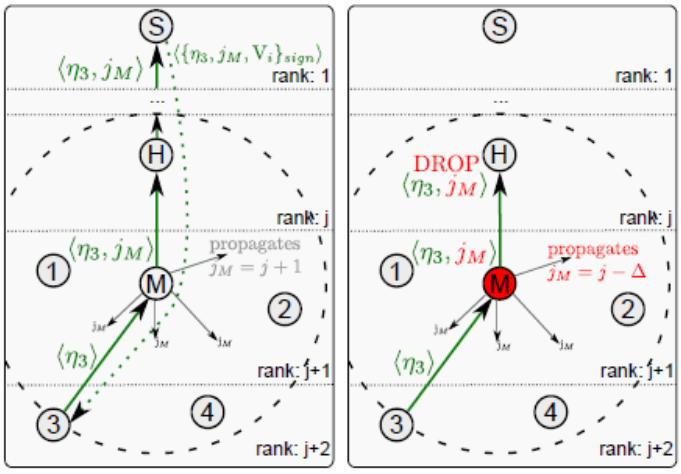


(a) Replaying rank announcement (b) Failing the challenge-response

Figure 3.73. Rank anouncement by the attacker *M*. (Perrey et al. 2015)

VeRA++ is proposed to improve the vulnerabilities of the inspired architecture, VeRA. It replaces the MAC calculation steps with AES encryption scheme. After generating the version number and rank hash chains VeRA++ node computes the encryption scheme using encryption. Also, the architecture broadcasts the DIO message. The underlying encryption scheme, one-way hash function, and ECC for signature are secure. The only way to calculate the entire hash chain is forging the signature. If the signature is generated and preserved in a secure environment, then it is negligible to forge it.

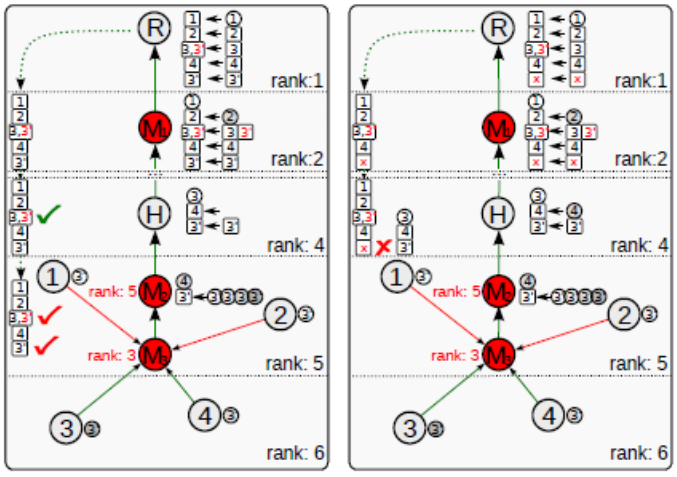
The authors of the TRAIL proposed a scheme against rank replay attacks using the rank hierarchy of RPL. Forwarding nodes have a challenge-response operation pair. In the rank replay a malicious node multicast a rank and try to modify the topology. The attacker



(a) Successful Rank Validation (b) Failed Rank Validation

Figure 3.74. The rank validation attempts (Perrey et al. 2015)

M propagates the rank j_M which is decreased falsely and creates a sinkhole. After getting acknowledged by neighboring nodes and a parent node, it can create an upstream towards the parent node. After its parent receives the intrusion it sends a challenge to the malicious node with a nonce value. The challenge will receive a correct response if the challenger's parent has a relation with the malicious node.



(a) Detection of Duplicate Nonces (b) Detection of Removed Nonces

Figure 3.75. The duplicate node detection (Perrey et al. 2015)

The malicious node cannot calculate the challenge because of pre-image resistance in the one-way hash function that is forming the hash chain. Also, all the nodes in this system can only know its parents and children. However, they cannot deduce the properties of their grandparent or grandchildren. On the other hand, the challenge generating RPL node

should be at the same level of hierarchy as the adversary and at the maximum communication distance to it. This implicates self-organization of nodes and anomaly detection feature. Nonetheless, this scheme is vulnerable to out-of-band challenges.

TRAIL is a more generic and structured approach on IDSs which is aiming to detect and prevent the consistencies. It provides a path validation method by detecting the irregular sub-DODAG trees. The root of the trees can start a local repair or change their paths. TRAIL does not support encryption chains, but it focuses on routing direction towards the root.

During the rank advertisement a node with children, one of the children nodes receives the message and sends a random nonce value to its parent (see Figure 3.73). However, the attacker must also form a relation with the grandparent for a successful result. The parent node which receives the nonce value forwards it with the rank number of its own as a test message to its parent node. At each iteration, the receiver of rank number and nonce value verifies that the rank number it received is bigger and sending node rank number is between of its own and rank number in the message. When a malicious node is present, during one of the iterations an inconsistency arises. It is notified by the receiving node signing it with its nonce. When the rank and version numbers are not valid, the node stops the propagation of the message. The child nodes may choose another upstream for propagation. The test messages are passed hierarchically in a recursive fashion.

The recursive operation of TRAIL raises overhead in messages and signature verification. Therefore, this implementation on its own is not scalable. In order to make it scalable, Bloom filters (Bloom 1970) are applied. It is a space-efficient random data structure. These filters keep the nonces of a grandparent in a group and later validated together then the array is multicasted to children town the DODAG tree.

Table 3.1. Message overhead with *k* number of children and *h* heights

Network Configuration			Message Overhead [Bytes]		
k	h	# Nodes	# Msg. per node	Average Size	Max. Size
2	3	15	2	3.5	10.5
	4	31	2	7.5	22.5
	5	63	2	15.5	46.5
4	3	85	2	12.6	63
	4	341	2	51	255
	5	1365	2	204.6	1023

TRAIL considers that the multiple malicious nodes are either collaborating at limited levels or not collaborating at all (see Figure 3.75). The squares are the array elements and the circles are nodes. Attacker M_1 copies the η values with a false rank. Other nodes detect the duplications, while M_1 removes the duplicate ηs. When the honest nodes detect ηs they drop their messages. Therefore, duplicate detection fails. A malicious node may

not include its children in the test array it forwards. They may rearrange the nonces at wrong positions. It may also attempt to avoid submitting its nonce value in order to avoid the attestation hierarchy.

Message overhead values are as given in the Table 3.1. Overheads increase significantly as the number of children increases. Height of the tree also affects the message overheads. TRAIL shows good timing performance with 17 nodes attack. Initialization phase is considered a reasonable level despite the communication variations due to the wireless ad-hoc infrastructure.

3.5.5. Event-based IDS with Frequency Agility manager

Kasinathan et al. (2014) described a secure backbone against jamming and flooding DoS attacks. For the less resource-constrained networks, a more complex wireless medium could be used to store and transmit the information of adversarial threats. Also, the system must be monitored in real-time for such mechanisms to be effective. The service available in this context is crucial. This framework adapts to the ebbits system and uses penetration test probes of Scapy instead of Metasploit as in SVELTE.

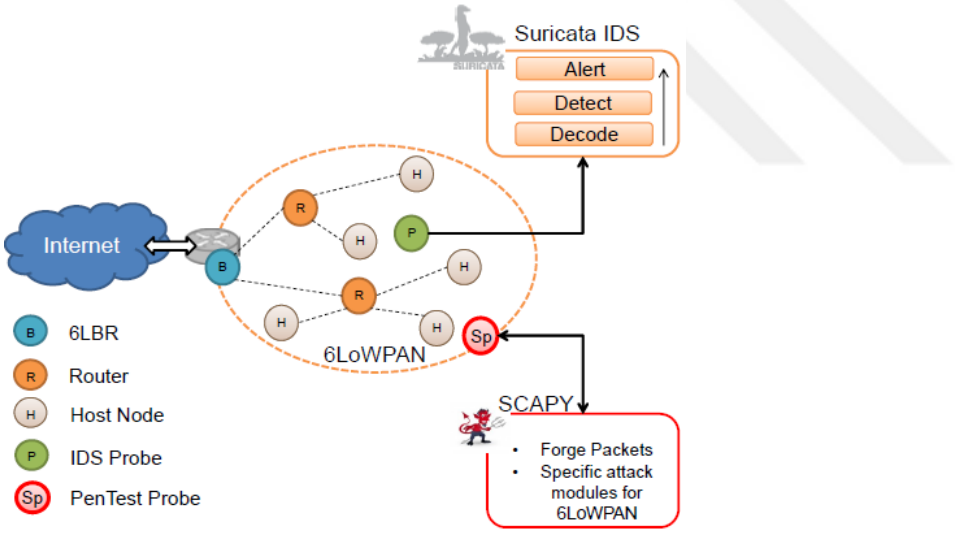


Figure 3.76. The DoS protection architecture (Kasinathan et al. 2014)

The DoS Protection architecture is simple as shown in the Figure 3.76. There is a 6BR between 6LoWPAN and Internet communication as a gateway. 6BRs are connected to the hosts and hosts are connected to the DODAG root devices. Suricata IDS probes are used as a sniffing mechanism in the network. They can alert possible threats, detects them, and decode them. They are in direct communication with the IDS network. Suricate probes can simulate the attacks by monitoring and injecting packets.

Scapy was chosen instead of Metasploit which was used in SVELTE architecture. It is able to simulate more complex attacks and more efficient in packet forgery. These systems

offer light-weight support for flexibility.

The IDS system in this section, corporates with a security incident and event management system(SIEM) and Frequency Agility(FA) manager. FA provides real-time interference detection within the network. When a system surpassed a given threshold of interference, FA changes the upstream of communication. FA and IDS provide a safeguard system.

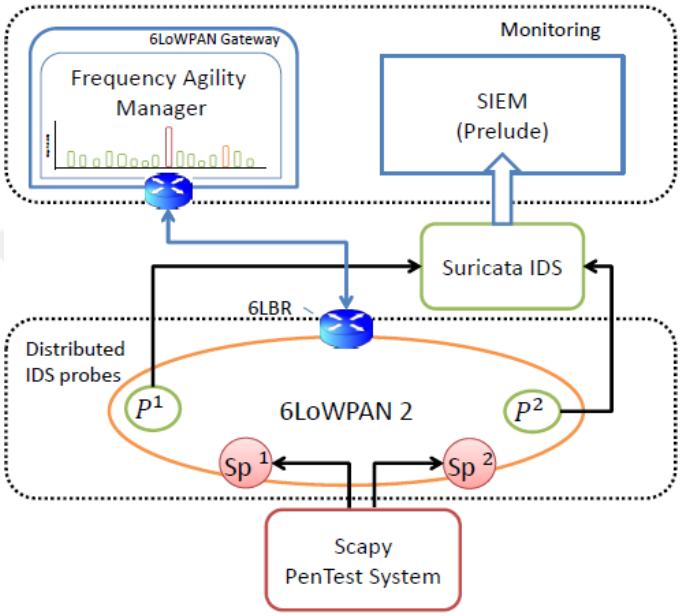


Figure 3.77. The IDS framework (Kasinathan et al. 2014)

The entire IDS framework can be explained in four main layers (see Figure 3.77). Firstly, the monitoring layer which consists of FA as a 6LoWPAN gateway that communicates with 6BR of physical devices and the SIEM module that analyzes the information from Suricata IDS. The second layer is Suricata IDS which receives network monitoring information from its probes placed among the physical devices. Thirdly, the physical layer where DODAG trees are present with Suricata probes to monitor them and enclosed by the penetration test probes. The last layer is the penetration test system, Scapy.

The system was evaluated with a graphical user interface (GUI). The outage probability is affected by the given threshold interference. The framework provides scalability and system availability to 6LoWPAN communications.

3.5.6. CEP-based IDS

DoS attacks, aside from the usual definition, can be explained as a large volume of message requests those sent frequently to disrupt the network. The early IDS frameworks are not mainly designed to monitor the system in real-time for large volumes of messages. They do not also find patterns in the streams of events.

Cugola and Magara (2012) designed Complex Event Processing (CEP) to detect an unusual event activity in a real-time system and filter them. It is easy to implement for users to determine events in a wireless sensor networks. Jun and Chi designed a CEP-based IDS system in order to take advantage of these properties. They proposed the architecture in terms of data collection from physical devices, forming events from the collected data, and filtering of anomaly events from the legitimate ones.

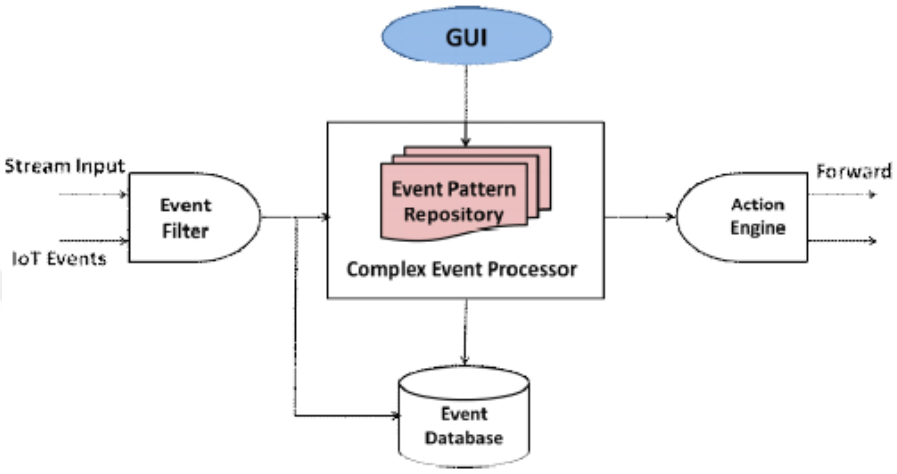


Figure 3.78. CEP-based IDS architeture (Chen and Chen 2014)

Chen and Chen (2014) designed a pattern for the CEP-based IDS called Event Processing Model (EPM) in Rule Pattern Repository. It is inspired by SQL and consists of four main elements. Firstly, the event operators which are SQL-like query operators such as select, from, where, order, group, etc. those are applied to event streams. Secondly, the view element is used for filtering and joining of events. There are two types of views; built-view for EPM clause and self-defined view for window clause. Thirdly, the event pattern which is compared for event identification. Lastly, integration of relational database. Event Processing Engine can be used with policy rules of IDS which are set in advance.

The architecture of CEP-based IDS has four important modules. These are data filtering, event modeling, event analysis, and security response. The data filtering from raw data before building a proper model is important to get better quality of performance. Stream inputs and IoT events enter this event filter. The filtered events are submitted to the event database. In the CEP there exists an Event Pattern Repository which gets inputs from a graphical user interface (GUI). The results of the CEP is submitted to both the event database and an action engine.

The Event filter monitors the network behavior and collects traffic information. CEP processes the security events with the core module of Event Pattern Repository. Action Engine deals with the events that not fit the defined Event Processing Model. Finally, Event Database stores the logs of processed events and results of CEP (see Figure 3.78).

The CEP-based IDS is evaluated in terms of real-time settings. It is compared against traditional IDS with different data scales such as 200, 400, 800. Their performance deviations are similar to different scales in CPU utilization and memory consumption. However, CEP-based system results in the half processing time of traditional IDS. The CPU utilization is slightly better in traditional IDS schemes; however, CEP-based IDS results in better memory consumption and processing time.

3.5.7. RIDES

IP-based Ubiquitous Sensor Network (IP-USN) is designed to converge the IoT infrastructures with IP protocols, so they can have a common communication structure (Choe et al. 2008). Amin et al. (2009) exploits Robust Intrusion Detection System(RIDES) which is an IP-USN environment to model a dynamic and hybrid IDS.

The communication structure of IP-USN consists of an IP Host, IP-USN gateway, and Sensor node (see Figure 3.79). The IP Host communicates with the IP-USN gateway through the Ethernet physical layer which is separated from the 6LoWPAN adaptation layer and IEEE 802.15.4 physical layer. In order to communicate with the sensor node, the packet is processed in the IP/IPv6 network layer and the result is returned to the 6LoWPAN adaption layer and leaves the gateway through the IEEE 802.15.4 physical layer. After exiting the gateway it enters the sensor node communication stack from its physical layer and continues to move towards higher layers. Sensor node communication stack is comprised of physical, adaptation, network, transport, and application layer. The application layer is the highest level layer. Adaptation layers are where the packets are partitioned and reassembled.

RIDES framework supports both anomaly and signature-based intrusion detection mechanisms. Anomaly based attacks are not strong for small attack ranges with few packets. Also, IoT systems are intrinsically resource constraint. Therefore, signature computation for the entire protocol is not feasible. RIDES is consisting of a gateway (also known as a sink) and IP-based sensor nodes. Gateways include a database for policies and signatures. They also facilitate an analyzer that works with anomaly detection principles. IP-based sensor nodes have Bloom filters, Signature-code generator(SCG) to support Bloom filters, and a Network Anomaly Detector(NAD) mechanism.

The Anomaly-based analyzer has a mechanism to support the storage of network states and a scoring system for those states which is called Contamination Score(CS). CS values range between 0 to 100. These scores act as threshold for the infection from an intrusion. Gateway analyses those states and initiates a response against the adversary. The mechanism analyses values from a packet sniffer. Every packet is checked in the NAD module while SCG module is initiated when the packets are directed to the node itself.

Bloom filters are convenient for filtering of the anomaly in events or signatures, but the workload on a single filter will damage the efficiency of the network. Although, an approach where the separate machines utilizing different Bloom filters with a distributed approach is existing, they do little for the efficiency by themselves. The workload of hash calculation and pattern matching of an entire payload is heavy on the IoT systems. Amin

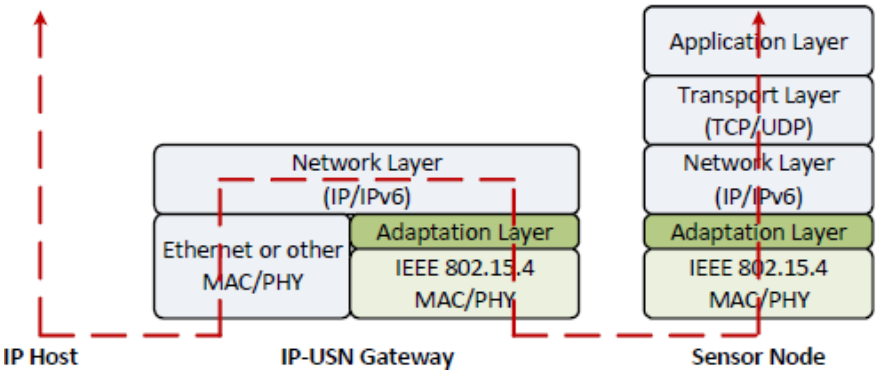


Figure 3.79. IP-USN (Chen and Chen 2014)

et al. (2009 describe the Signature-code concept to detect the identifiers in real-time and represent the signatures in very small sizes in terms of a few bytes.

At the beginning of signature matching, Bloom filters receive signatures with different lengths and results in arrays to signatures and signature-codes. Bloom filters verify the matches and packets processing is stopped. Then, the gateway is alarmed by receiving the designated signature-code in order to verify the match. Signature-codes gives the location of the entire signature in the database with the policies appointed to them. The signature-code is generated by concatenating two consecutive hash functions which are not equal to each other, as the signature is the input of those functions. Gateway queries the database with this code to extract the full signature. In the same row, the rules dedicated to the signature are accessible. The signature set used in this architecture is based on Snort.

The performance metrics are based on the array size of Bloom filters, the number of signatures used by the adversary, and the number of hash functions. The hash functions should be collision resistant. Bloom filter uses 1.44log(1/fpr) bits for space for each signature injection attack, fpr is the false positive rate. As the value of fpr increases, the storage required for sensor nodes is decreased severely. 10000 signatures can be added without any collision. Without SCG, energy consumption is too high compared to when there is SCG. The energy consumption rates are not significantly affected by the distance between the sensor nodes, unlike non-SCG implementations.

4. RESULTS AND DISCUSSION

4.1. Analysis of WSNs

The mechanisms described in this topic have a variety of features. Therefore, it is not possible to make a fair comparison among themselves. Moreover, despite the fact they consider resource constrained techniques, they are not designed particularly for 6LoWPAN standards. Although in the last decade there are many mechanisms developed for 6LoWPANs, it is important to revisit to build new efficient approaches for better solutions in the future. The mechanisms are summarized in table 4.2. Here ZigBee has options to

Table 4.2. The Summary of WSN Mechanisms

	Network Structure	Keys	Key Generation	Key Establishment	Integrity	Redundancy
ZigBee	Centralized and Decentralized	Network key, link key, master key	SKKE for link key	CBKE	AES-CCM* with MIC	DSSS
SPINS	Semi Distributed	Master key, pairwise key, broadcast one-way function key chain	Keys are derived from pseudo-random function for symmetric encryption	Routing beacons	SNEP: DES with CBC-MAC, µTESLA: key chain MAC based on delay disclosure	Watchdog Behavior
TinySec	Link Layer Security defined for both approaches	Utilizes the TinyOS properties	Utilizes the TinyOS properties	Utilizes the TinyOS properties	Skipjack with CBC-MAC	-
LEAP	Semi Distributed	Individual key, group key, cluster key, pairwise key	TimerC, RandomLFSR, RC5	A recursive process for spanning tree, one-way key chain	RC5 with CBC-MAC	-

work centralized or decentralized. SPINS and LEAP work as semi-distributed with trust centers. TinySec is an extension for TinyOS to support the frame security at link layer.

ZigBee utilized three types of keys such as network, link, and master. On the other hand, SPINS has three types of keys which are master key, pairwise, and broadcast key chain in a similar fashion. LEAP has four types of keys. The first key is for identifying, the second one is for communicating with the nodes in the same group, the third one is for communicating with the nodes in the cluster, and finally, the fourth key is for direct communication to exchange sensitive information.

Symmetric-key key exchange protocol is used to generate link key for nodes to communicate in ZigBee. SPINS has a pseudo-random function for deriving a key for master. The rest of its keys are derived from the master key. LEAP uses TimerC and RandomLSR components to generate keys.

SPINS has a unique approach on the key establishment which is routing beacons. LEAP utilizes a recursive process of spanning tree for establishing the generating the keys throughout the network. It also uses the one-way key chain. ZigBee uses certificate-based key exchange protocol.

Integrity in the protocols mostly based on CBC-MAC operations. Only ZigBee has a slightly different approach which is also derived from CBC-MAC. It applies AES-CCM* for encryption with message integrity code. SNEP of SPINS uses DES, TinySec uses

Skipjack and LEAP uses RC5 encryptions. Additionally, SPINS uses the one-way key chain as a message authentication code scheme in $\mu TESLA$.

ZigBee and SPINS additionally apply redundancy in their mechanisms. While ZigBee uses a physical signal error coding method, SPINS uses an anomaly detection for error checking. However, IDS systems are able to provide both misconfiguration and malicious anomalies.

4.2. Analysis of Data Stream Management Systems

The data stream management systems have many common features with IoT networks. IoT also deals with dynamic data processing in streams, efficient and lossy systems. However, DSMSs are not concerned with hardware properties as in WSNs.

A brief summary of the mechanisms explained in table 4.3. FT-RC4 provides an extension to Nile data stream engine with cryptographic components and an access control manager. The streaming is also partitioned into cycles. On the other hand, CADS outsource database through a semi-distributed approach with separate service providers. A virtual caching mechanism is used for query monitoring. The lightweight linear algebraic approach separates the system into building blocks and outsources the DOs similar to CADS. Also has small summaries included for streams in order to provide efficient queries. RBAC has a distributed approach based on roles given to entities. It also supports encrypted transportation. The secure punctuation mechanism is not a full access control mechanism but enforcement of meta-data to enable existing access control mechanisms to work efficiently. It also allows query processing and optimization. Publicly Verifiable Grouped Aggregation on Outsourced Data Streams support outsourcing as well and public verification; hence, its name indicates.

The security mechanisms proposed are diverse in the given access control mechanisms. The FT-RC4 uses the common stream cipher, RC4. The CADS uses RSA-based Temporal Merkle Hash-Trees and Domain Partition Merkle Hash-Trees mechanisms. The Linear Algebraic approach, on the other hand, supports minimal lightweight matrix summation and production with Hash-based MAC and SHA1 digital signature. RBAC has object-level and data-level security during query processing. Publicly verifiable grouped aggregation queries support lightweight query security along with DiSH structure which is supported by probabilistic signature functions.

All of the mechanisms are evaluated in separate environments. The FT-RC4 is tested in the Nile, the CADS is tested on an Intel Pentium 4, 3GHz CPU with 2GBs RAM using Crypto++ environment, the linear algebraic approach is tested on an Intel Core i7, 2.6 GHz with 4GBs of RAM using GMP and OpenSSL environments, RBAC is tested in the Borealis stream engine on a Fedora Core2 AMD Athlon XP 1800 CPU with 1GB of RAM, SP is tested in the CAPE stream engine on an Intel Pentium 4 CPU with 1GB of RAM, and the publicly verifiable approach is tested on Intel Core 2, 2.5 GHz CPU with 4GBs of RAM with GNU C++ and NTL library.

The FT-RC4 has the advantages of high transmission rates by running in a small me-

Table 4.3. The Summary of DSMS Access Control mechanisms

	Methods	Security Mechanisms	Experiment	Advantages	Disadvantages
FT-RC4	Extending Nile, Cryptographic components, Access Control manager, Streaming is separated into cycles	RC4	Nile	Provides efficiency in high transmission rates, runs in smal memory, requires low computation power	More processing time, data loss increases with the cycle size
CADS	Database outsourcing, semi distributed system with multiple SPs, VCM	TMH and DMH using RSA	P4, 3GHz CPU with 2GBs of RAM, Crypto++	Fast update, temporal correctness, redundancy	False transmissions
Linear Algebraic	Building blocks, data owner outsourcing, small summaries of streams	Linear Algebraic Queries such as DVS, DMP, and DDP and HMAC with SHA1 algorithm	Intel Core i7, 2.6 GHz with 4GBs of RAM, GMP and OpenSSL	Correctness, Freshness, Observation of performance	Applied on only specific query types and vulnerable to DoS attacks
RBAC	Role and permission distribution, encrypted transport	Lightweight object-level and data-level security using query operations	Boreals Engine, Fedora Core2 AMD Athlon XP 1800 with 1GB of RAM	Efficiently seperated actions and actors	Applied on only specific query types and vulnerable to DoS attacks
SP	Not an access control mechanism but an enforcement, meta-data for flexibility, query processing and optimization framework	No cryptographic security, data blocks such as DDP, SRP, Sign, Immutable, and Timestamp	CAPE, Intel Pentium IV CPU with 1GB of RAM, Windows XP	Flexibility and Dynamicity, minimal overheads	Requires additional semantic security components
PVGAQ	Outsourced Stream Aggregation, Publicly Verifiable	Lightweight query operations with proposed DiSH mechanism with probabilistic signature functions	GNU C++, NTL library, Intel Core 2, 2.5 GHz CPU with 4GBs RAM	Practical and efficient, solutions for both static and dynamic subset queries	Cryptographic operations are costly

mory with low computational power. Its disadvantages are requiring more processing time, increase in the data loss proportional with cycle size. The CADS is advantageous on fast updates, temporal correctness, and redundancy. However, it has high rates of false transmission. The linear algebraic approach has correctness, freshness, and performance monitoring features. Nevertheless, it is only applicable to a specific set of queries. Moreover, it is vulnerable to DoS attacks, which are the most common in IoT.

The RBAC system has the advantage of efficient distributed operations. However, they are also applied to specific types of queries and vulnerable to DoS attacks. SP provides flexibility and dynamicity with minimal overheads to the applied system. However, the applied system requires an interpretation mechanism with cryptographic components for

a proper access control framework. Lastly, publicly verifiable grouped aggregation query approach is practical and efficient for both static and dynamic queries. Despite, it requires lightweight cryptographic operations while the proposed ones are not suitable for IoT environments.

4.3. Analysis of PUF Mechanisms

Maiti et al. (2012) defined three measurement dimensions for PUF mechanisms: Device, space, and time. Many of the parameters used in context are redundant or overlapping. The most distinctive parameters are uniqueness, reliability, uniformity (intra-chip and inter-chip variation), and efficiency. There are additional measurements such as bit-aliasing, probability of misidentification, and diffuseness.

Uniformity and diffuseness are evaluated in terms of spatial dimension. Uniqueness, bit-aliasing, and the probability of error are part of the device dimension. Finally, reliability and steadiness belong to the temporal dimension.

In table 4.4, popular PUF mechanisms are summarized according to early descriptions. The many of recent mechanisms are either fine-tuned versions with additional hardware components or abstract ones.

Many of the mechanisms evaluated in table 4.4 are diverse in their mechanisms for bit production. The most are transistor based and exploit the delay or voltage differences. TERO PUF is the most effective in these terms. However, there is computational overhead because on the last bits give the best performance, while the next two bits are barely useful. The remaining bits are incompetent.

The final two methods are practical applications of PUF for IoT infrastructures. However, despite the variety of existing PUF mechanisms, not many of them are integrated due to the manufacturing reasons.

4.4. Analysis of Lightweight Cryptography

In this section, the analysis is based on the findings of Mohd et al. (2015). Their work is noteworthy and complimentary to analysis made in this section. It is important to test the ciphers based on hardware on the equivalent environment to extract fair analysis. The analysis carried statically and dynamically. Dynamic analysis requires a test-bench environment in order to probe the testing inputs to enable the timing and power measurements during cryptographic computations. Zhang et al. (2013) proposed an approach for dynamic metric extraction by assuming default signal activities at the input and output pins, obtaining the activities from the internal and input/output signals from behavioral simulation, and finally obtaining the signal results from gate-level simulations. There are also different approaches to extract metrics when this method is not applicable (e.g design area). However, these induce inaccuracy to the performance results due to irrelevant features.

One of the common questions is if it is important to analyze encryption and decryption

Table 4.4. The Summary of PUF mechanisms

	Approach	Average Inter-chip Variation	Average Intra-chip Variation	Average Uniqueness	Error	Efficiency
Arbiter PUF	Switch (use of latches) and arbiter circuits are used. Additional feed-forward cascade arbiter circuits are added for uniformity	38%	14.40%			190 tests, 24.5µs from 490 CRPs
RO PUF	Ring oscillator implemented with inverter mosfet circuit. Delay loops, multiplexer, counters, and comparison logic	46.15%	0.08%	47.31%	10%	512 oscillations 4.8ns
SRAM	SRAM cell circuits with six cross-coupled inverters, access transistors, data bit-lines based on word-line signal. Additional fuzzy logic	49.97%	3.57%	49.2%	12%	1023 bits to generate 278 bits of data
Butterfly PUF	Cross-coupled latches, symmetrical setting	49.8%	6%	49%	10^{-6}	1500 Butterfly cells to generate 128 bits of data
Loop PUF	Single loop of cascaded delay chains. The number of the chains are equal to the challenge/response size	≈ 50%	7.5%	≈ 50%	10^{-4}	250µs required for 24 loops to generate 360 CRP
TERO PUF	N number SR flipflops cascaded with 8-bit counter, 26-bit accumulator, and 18-bit shift register, in order to produce N bit data.	48%	First two bits are 39% and the variation is gradually decreased for the rest of the bits until 1.7%. Last 2 bits can be used.	≈ 50%	NA	Effective for producing 126 bits at the same time.

together. It is true that the combination of both operations gives an insight about the performance of all scheme. However, most of the encryption and decryption schemes are almost the same. There are many researches focus on encryption only analysis for the lack of mode of operations in decryption.

Software metrics are mainly code size in bytes, RAM size in bytes, *Cycles/block*, *Cycles/byte*, *Energy/block*, *Code size x Cycle count / Block size* as introduced by Eisenbarth et al. (2007), Efficiency over storage and power, and throughput in unit of encrypted Kbps. Throughput can be expressed as block size x frequency / the number of cycles to encrypt a single block. It is important to avoid analysis on different platforms. The synthethic metrics combine several primitive metrics which induce inaccuracy. These metrics are not applicable for highly constrained environments due to their dependencies.

Mohd et al. (2015) define the hardware metrics by gate equivalent(GE) area, *area/bit*,

throughput, performance efficiency which is *throughput/area*, figure-of-metric(*FoM*), power and energy, *energy/bit*, *energy x area/bit*, and *area x energy/bit*. The area depends on the type and implementation. Although GE is the common unit, ASIC architecture uses μm^2, Altera expresses it in logic elements(*LE*s), and Xilinx describes the area in configurable logic blocks(*CLB*s). Although the representing elements are different, each unit represents a certain logic unit in the circuit. It does not indicate anything related to computational performance but the vendor price. *Area/bit* ratio gives the cost for each block size. Throughput is considered a metric in order to show the performance results with changes in clock frequency when the technology is faster for the given design area. This metric is dependent on many other metrics relative to the reference point of analysis. *FoM* metric analysis is the performance over power change. However, this metric is not reliable with the leakage in standard cell libraries and wire loads during operations. *Energy/bit* unit analyzes the energy over a given block size and the results give the energy efficiency over cost. Energy*Area/bits merges two critical properties which are indicated earlier.

Mohd et al. (2015) defined a general metric consisting of common metrics regarding their dependencies and relevancies to the level of security and resource constraints. It is defined in 5-dimensional space with variables such as design area, time to encrypt one block, energy, number of cycles per encryption of a block, and block size. They assigned specific coefficients for each metric for the level of their influence in the analysis results. This generalization of metrics allows some facts and optimum design choices about implementations (e.g appropriate rounds for block ciphers). However, the recent semiconductor developments show that area is not as significant as it used to be. Energy consumption encapsulates the main issues in hardware-oriented cipher designs.

It is aforementioned that the software-oriented platforms are implemented either machine-dependent or -independent. Machine-dependent implementations are applied to small devices with cheap micro-controllers. RC5 and RC6 which implemented on 16-bit RISC architecture have small code size but poor key properties. MISTY1 has shown good performance by means of CPU cycle counts and storage requirements, however, it still has security weaknesses. For energy-efficiency AES is the best performing cipher despite the design area cost. During the analysis of top performing recent ciphers on Atmel, 8-bit AVR micro-controller, it is seen that AES, Noekeon, and TEA show the best performance. The results also prove that energy performance is correlated with the cycle count. Idea, Hight, TEA, and AES show good energy/block and throughput results. Idea, PRESENT, TEA, Sea, and AES have optimal code sizes. Cazorla et al. (2013), reported a performance analysis on MP430, a 16-bit microcontroller used with the sensor node WSN430 and implemented ciphers in C programming languages. TEA, xTEA, DESXL, Noekeon, Klein, and AES shows the best performance respect to cycle count and cycles/byte. TEA, xTEA, Twine, and LED show the highest code-size results. xTEA, Lblock, TEA, and MIBS use RAM the most efficiently. TEA and XTEA ciphers are the most memory efficient ciphers for machine-dependent software-oriented platforms. AES with 128-bit block size performs the best for machine-independent implementation. There are also variants of AES perform well according to the application context with a smaller size.

The hardware-oriented platforms can be analyzed under ASIC and FPGA architectu-

res. ASIC architecture is based on 65-nm low-power CMOS technology. During its analysis, one can reach several conclusions. Firstly, the combinatorial logic should be small. Second, there are times when a single cipher round should split into several sub-rounds as they do not directly affect the delays. Adding unrolling rounds at the beginning increases the throughput, but they decrease the throughput when they are added later. Energy/bit encryption is carried out with the highest performance by Noekeon, PRESENT, and Katan in the given order. Adding too many rounds decreases energy efficiency. Parallelism and pipelining do not contribute to algorithm efficiency. Katan and Klein have the least area and area/bit values. Katan, Klein-serial, PRESENT, and Klein-parallel show the lowest power dissipation. On the static power analysis Klein, Prince, mCrypton, and PRESENT show the lowest energy/bit evaluation. As for the dynamic power analysis Klein-parallel, mCrypton, Prince-folded, and PRESENT perform the best in respective order. Among the key scheduling ciphers, LED has the lowest number of GEs while PRINTCipher has the lowest number among the ciphers with hardwired keys. mCrypton shows the best throughput/GEs. In general, the smallest design area by means of GEs is Ktantan, PRESENT, and Katan.

The evaluation of performance is more difficult in FPGAs than in ASIC implementations. Most of the implementations are designed for different FPGA devices. The implementations that require the smallest design area are Hight and PRESENT. The best throughput results are given by xTEA, Hight, Piccolo, PRESENT, and Khudra. PRESENT also shows the best slices/bit and Mbps/slices evaluation results.

It is possible to give a general summary of the general metrics for software- and hardware-oriented implementations (Mohd et al. 2015). In software-oriented metrics, we can split them into throughput, code-size, energy efficiency, and RAM usage.

i) Throughput: Idea, Hight, MISTY1, Noekeon, DESXL
ii) Code-size: Hight, SEA, PRESENT, Noekeon, Idea, Katan
iii) Energy efficiency: MISTY1, DESXL, TEA, xTEA, Klein, Noekeon
iv) RAM usage: Lblock, SEA, Klein, PRESENT, Katan

The hardware performance results can be separated into throughput, design area, power, and energy.

i) Throughput: Present, Piccolo, Khudra, Armadillo, Hummingbird2
ii) Area: xTEA, Khudra, Klein, Hight, PRESENT, Katan, Ktantan
iii) Power: mCrypton, Prince, LED, Iceberg, Klein, PRESENT
iv) Energy: Hight, AES, Katan, PRESENT, Noekeon, Prince, mCrypton, Klein

Analysis of the lightweight cipher implementations should not be limited to given general metric. Since the technology is evolving rapidly, some of the metrics will not be significant. It is important to be vigilant towards several issues during the cipher design process such as the design and role of the performance model, preventing hardware trojans by modeling them, creating a cryptographically sound security metric, and efficient algorithmic implementations.

In order to build a comprehensive performance model, one should consider semantic, structural, and resource constraints. It is important to decide on a cryptographically se-

cure structure such as Feistel, SPN, etc. Regarding the given constraints, one should find common grounds with security, complexity, memory, and energy parameters. Specific to the block ciphers, it is important to balance the number of rounds and the complexity of the combinatorial logic of rounds. Parallelism should be considered for higher throughput while running these rounds. The design should include optimal cipher libraries.

It is indicated earlier, there is a paradoxical absence of security metric. However, there are several cryptographical structures such as AES rounds, PRESENT S-boxes, etc. to use as a reference model. There should be levels of securities defined regarding the sensitivity of the information.

4.5. A proposal of multilayered IoT Framework

Here, we propose a skeleton of an IoT architecture on 6LoWPAN with an anomaly-based IDS system. This architecture is scalable with several degrees of semi-decentralization structure. The main entities in this architecture are users, devices, and outsources. Their activities are based on hard-coded and adjustable rules. There is also a grouping option for different application-contexts to give better analysis and adaptive results. The following sections will give a brief description of these properties. The general structure can be seen in Figure 4.80. The "H" circles indicate the host devices in the cluster (i.e. level 1, level 2, and level 3). "$6BR_{i,j}$" circles indicate the level 2 gateway devices, 6LoWPAN border routers. $GM_{i,j}$ circles indicate the group managers.

4.5.1. Devices

The 6LoWPAN network in this backbone is split into clusters and groups as in LEAP (Chen et al. 2006). Clusters are for devices in close proximity while groups define application contexts. While cluster managers deal with locations, the group managers orchestrate common applications. This way highly-constrained devices can have a limit to keep the communication keys of their neighbors. These devices can communicate without any problem within their clusters. However, they receive routing support from high-level devices those act like gateways (6BRs) in order to communicate out of their cluster but within their application-context group. The gateways of groups have more complex analysis, detection, and administrative tools.

The devices in the framework can be either passive or active with IPv6 communication support. There are three levels of devices. The level 1 devices are the passive devices can be RFID tags or cards. They have the standard 6LoWPAN protocol stack with IEEE 802.15.4 physical and MAC layers, IPv6 adaptation layer, UDP and ICMP supporting transport layer, and finally the application layer. They have their identity keys and direct communication (link) keys. Each of these devices can authenticate themselves with simple processes in the application layers. A TERO PUF (Bossuet et al. 2013) mechanism for a pseudo-random number generator for a key generation provides good statistical properties. These devices can keep the list of point-to-point messaging keys. They can advertise themselves to those who have their link keys. They require to change their master keys periodically or when they are triggered by a threat. The entities which keep their master

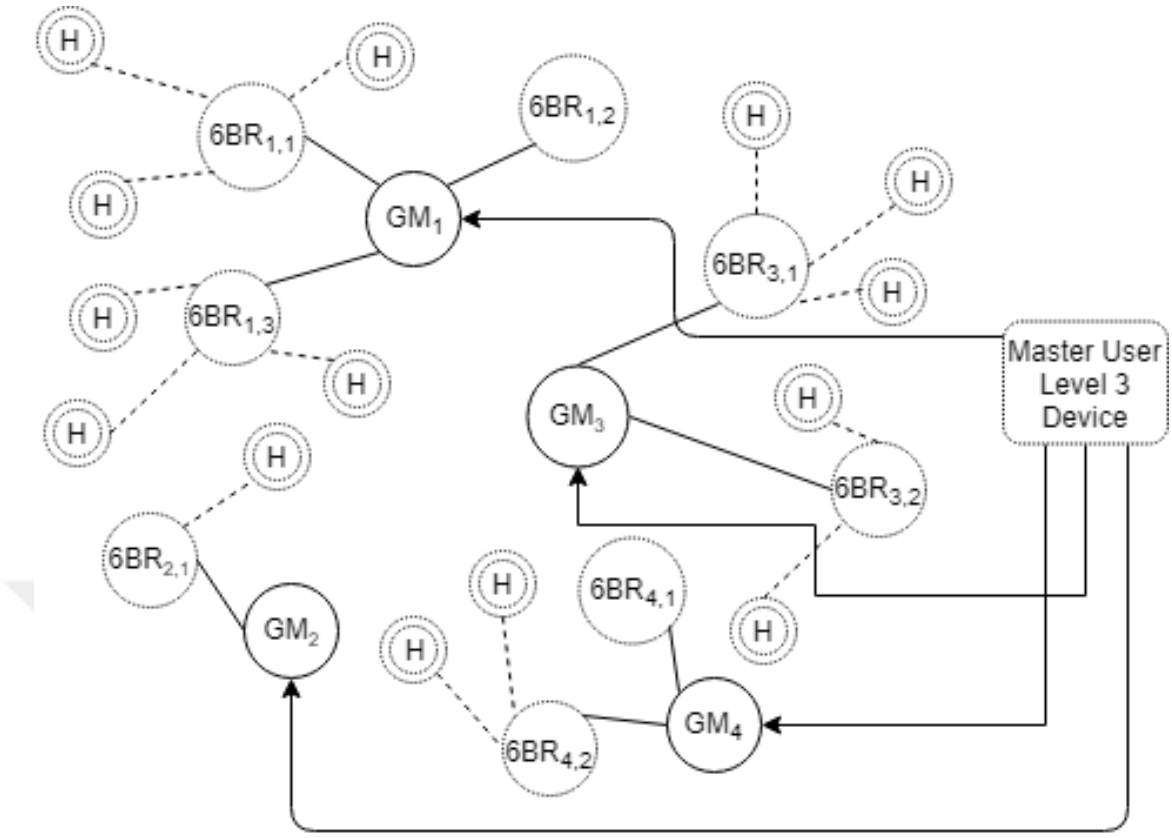


Figure 4.80. The Multilayered IoT structure

keys can initiate their key scheduling. Those entities store counters along with the master keys in order to update at each key schedule. This provides freshness property. This idea is inspired by SPINS (Perrig et al. 2002).

The level 2 devices are 6BR gateway devices with additional computational power, higher energy capacity, and larger data storages than level 1 devices. Naturally, they are active devices such as routers. They have an IP-USN Gateway protocol stack (Chen and Chen 2014) with the additional application and transport layers. The lower layers of stacks are separated into Ethernet and IEEE 802.15.4 layers. Ethernet layer consists of MAC and physical layers. On the other hand the IEEE 802.15.4 consists of IPv6 MAC and physical layers. Unlike the level 1 devices, the network layer allows level 2 devices to communicate on TCP as well. At the application layer, they can interpret TCP/IP requests from users, level 2 and higher devices, and external sources. 6BR supports the RPL topology. They keep the topology with other 6BRs as explained in previous sections (see Figure 3.57).

The level 3 devices are the generalized category for all devices with capable processors within the 6LoWPAN. These can be computers, mobile phones, and servers. They are mainly group managers. They can analyze the network according to the application needs. They can communicate with both 6LoWPAN protocols and TCP. They are equipped with ebbits similar to DoS-based IDS (Kasinathan et al. 2013). Ebbits facilitates a

network manager. Have three separate managers of opportunistic, network, and security. The additional frequency agility module for interferences in the network flow and a DoS manager to deal with anomaly entities within the framework as in the work of Kasinathan et al. (2013) suits our architecture. However, the frequency agility manager here is not only concerned with malfunctioned or malicious devices. It is also concerned with user behavior. The DoS protection manager is also adaptable to this context. However, a level 2 or level 3 device should capture network state periodically and reports back to the group managers.

4.5.2. Users

The users in this architecture are entities with possession of level 3 devices and issued identities. The network owner is the user with the highest priority. It has the unique identity that is issued once the user starts the network (e.g. the provider company can assign a unique subscription identity for the user and only can be assigned and updated with the decision of the provider). The authentication of the users is started with advertising its identity to his/her initial level 3 device. Level 3 device gives the highest priority to the master user.

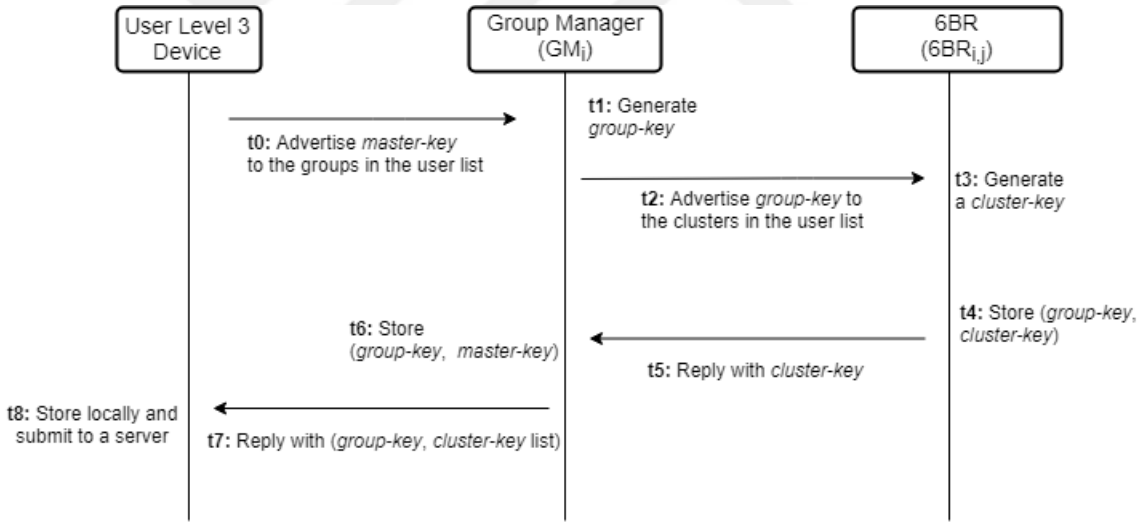


Figure 4.81. The user authentication protocol

The users have a master key, group key, cluster key, levels, assigned roles, and adjustable score variable. A user, other than the master user, can advertise himself with his level 3 device (e.g. mobile application, web interface, etc.) by applying his master key. His level 3 device adjusts his application-context to apply for a group to the group manager level 3 device. The group manager also receives the application permits, generates a group key and associates it with a message from the user device. Then stores this information. Next, the group manager propagates the user's rights to the accessible level 2 devices, the 6BRs, the user is allowed to access. The 6BR generates a cluster key for and associates it with his group key and the devices he is allowed to access. Then stores this information in its

storage. Replies the group manager with the cluster key is generated. After receiving all the replies for its requests, the group manager back-propagates the group key is generated with the array of cluster keys. The personal level 3 device of the user stores all this information in its storage, then submits all information regarding the user on a server in a json or a markup language-based document. The process is illustrated in Figure 4.81.

There are five levels as seen in Table **??**. The first level is an anonymous user with a new card (e.g. guest card). They start with the lowest score. It can only authenticate itself. The scores are adjusted in a crowd-sourced fashion by positive and negative feedbacks of the higher level users, level 2, and level 3 devices. The adjustments are not applied immediately. The score is finalized with an increment or a decrement by values received the participating entities. The reputation of a participating entity affects the significance of their voting entry.

Table 4.5. User levels

User Level	Self-authenticate	Able to vote	Device administration	User administration
1	YES	NO	NO	None
2	YES	YES	NO	None
3	YES	YES	YES	None
4	YES	YES	YES	Level 1 & Level 2
5	YES	YES	YES	Level 1, 2, 3, and 4

The level 2 users are differentiated from level 1 users by being able to participate during the voting session. On the other hand, level 3 users are the ones with devices or cluster rights. Other than the voting participation they can set the rules of access on their devices. The level 4 users can manage the rights of level 1 and level 2 users in addition to the level 3 user rights. Level 5 users are the highest administration users. They are allowed to administrate the level 3 and level 4 users. The master users are the only users who are able to manage these users. However, the anomalies can appear so they are also can be voted during the voting sessions.

The users can vote ubiquitously. However, the evaluations are finalized at periodical times. The level 3 devices also keep the device anomalies in check. In case of malicious device entered the system they can be discarded with the minimum damage.

4.5.3. External Sources

Users do not always have corporation level networks. Some will like to own these devices for small room automation. In order to be able to achieve good performance,

they will need to store the information and analyze them with their existing devices and storage. Integration can be done with an appropriate application to communicate with the 6LoWPAN network we propose. The information should be stored via an application specific document. This document can be in a json format, a common markup language, or a dedicated language. The communication protocol should be defined in an efficient way. The communication protocol should be concerned with large data such as video and audio.

4.5.4. Communication

The DoS manager that is located in the ebbits network manager deals with the anomalies is as described in Kasinathan et al. (2013). It receives alerts from the IDS modules in the 6BR devices, has one itself to report to an administrative level user or a parent device. They both monitor the user score changes and packet droppings. The IDS components keep the packet exchange rates, interferences, user and other device reports. They have patterns for anomaly attacks and those patterns are adjustable by administrative users or level 3 devices.

The internal communications in this network are done with IPv6 packets. The data payload of the IPv6 packets can be in fragmented form for very large packets such as videos, audios, etc.

It is computationally intensive to encrypt and decrypt every packet. Therefore, in this framework, the communications are started with encrypted Hello DIO messages and timestamps during the communication. The receivers of messages with a given delay threshold can check the timestamp with their own. This way it is possible to notice an anomaly and report to a parent node. The encryption is also applied during the key establishment and propagation. The PRESENT (Bogdanov et al. 2007) is suitable in terms of efficiency, energy consumption, code size, and throughput.

The patterns of communication behaviors can be message rates, delays for communication desynchronization, legitimate and verified topology changes, registered external addresses, blacklisting and whitelisting of external addresses, and voting verification periods. These patterns are stored in the level 3 devices.

4.6. Future Work

The proposed framework is designed for scalability and interoperability. However, the proper implementation of this work has not been done to evaluate in a real environment. Outside of the proposed design, communication with external entities should be described. The information about the topology, device, user, keys, permissions should be stored in a normalized database.

The patterns of user and device behavior can be iteratively generated for possible adversary models. The patterns can be adjusted using genetic algorithms and feedback propagation neural networks.

5. CONCLUSIONS

The Internet of Things infrastructures is rapidly growing since the last decade. This paper declares that it is important to review the timeline of the development in terms of security that lead to the construction of IoT standards today. Many of the ongoing projects are tightly knitted to each other to see the big picture. Nonetheless, their growth can be limited. We do not declare that the growth is not satisfying but for more efficient and secure platforms the existing proposal should be reviewed.

The homogeneous structure of IoT allows us to create an unlimited number of solutions. Therefore, treating the frameworks in building block model and make them flexible to adapt to old and new solutions.

This paper has mainly covered the wireless sensor networks, access control mechanisms for data stream engines, physically unclonable functions (PUFs), lightweight cryptographic constructions, high-level and hybrid constructions, and intrusion detection systems.

Since the starting point of IoT systems is wireless sensor networks it is important to review them first. Their constructions start from way back when the RFID systems came to the information technology market. They also deal with mildly constrained environments. There have been solutions widely ranging from extension mechanisms implemented on early systems to standalone decentralized infrastructures.

The data stream management systems deal with continuous communications as in WSNs and IoT environments. The protocols for access control mechanisms are not fully applicable to highly-constrained IoT systems. However, they can inspire some semantic backbone mechanisms. The role-based access control mechanisms, publicly verifiable grouped aggregation system, meta-data tagging on streams, and security punctuations are adaptable in order to collect information for network analysis on small passive devices.

The physically unclonable functions (PUF) are designed for devices with low computational power. They aim to provide uniformly distributed generation of numbers which are close to random by exploiting the physical differences of devices from manufacturing.

The standard cryptographic schemes require storage of large keys and intensive computation. The growth of block ciphers brought solutions to problems of existing stream ciphers. Eventually, they took over the cryptography literature by replacing the stream ciphers. Moreover, stream ciphers started to cease to provide effective solutions. However, the communication standards in wireless sensor networks brought back the importance of stream ciphers. Therefore, new stream ciphers which inspired by contemporary block ciphers were proposed.

The standardization of the IoT frameworks brought de facto protocols for communication similar to TCP/IP. The standards are mainly based on 6LoWPAN and RPL as explained in the early sections along with a brief classification of attacks on both systems.

Later, the fragmentation standards for large data packets described.

In this work, we explored the existing hybrid systems and their application in later sections. The intrusion detection systems have a separate classification, as most of the recent work and cryptanalysis are based on IDS frameworks. They are classified into different approaches and patterns. Anomaly and misuse intrusion are the most common patterns. The approaches take handle these patterns from different parameters. Some also use artificial intelligence methods to detect security threats.

The contribution of this thesis is mainly an analysis of previous methods and a proposal of a multilayered framework. The proposed architecture is required to have physical implementation for an evaluation. It can be improved with the implementation of more elaborate analysis tools.

6. REFERENCES

Abadi D.J., Ahmad Y., Balazinska M., Cherniack M., Hwang J., Lindner W., Maskey A.S., Rasin E., Ryvkina E., Tatbul N., Xing Y., and Zdonik S., 2005. The design of the borealis stream processing engine, CIDR, pp. 277–289, January, Asilomar, California, USA.

Amin S.O., Siddiqui M.S., Hong C.S., and Choe J., 2009. A Novel Coding Scheme to Implement Signature based IDS in IP based Sensor Networks, 2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops, pp. 269-274, New York, New York, USA.

Aguiar R.L., Sarma A., Bijwaard D., Marchetti L., Pacyna P., and Pacyna R., 2007, Pervasiveness in a competitive multi-operator environment:the daidalos project. *IEEE Communications Magazine*, IEEE, 45(10): 22-26.

Akram H. and Hoffmann, 2008. Supports for Identity Management in Ambient Environments - The Hydra Approach -, The Third International Conference on Systems and Networks Communications, pp. 371-377, October, Sliema, Malta.

Alcaide A., Palomar E., Montero-Castillo J., and Ribagorda A., 2013. Anonymous authentication for privacy-preserving IoT target-driven applications *Computers and Security*, 37: 20-23.

Ali M., ElTabakh M., and Nita-Rotaru C., 2005. FT-RC4: A Robust Security Mechanism for Data Stream Systems, Department of Computer Science Technical Report TR-05-024, paper 1638, November, Purdue University.

Alrababah D. and Alshammari E. 2017. A Survey: Authentication Protocols for Wireless Sensor Network in the Internet of Things; Keys and Attacks, IInternational Conference on New Trends in Computing Sciences (ICTCS), pp. 270-276, Amman.

Aumasson J.P., Fischer S., Khazaei S., Meier W., and Rechberger C., 2008. New features of Latin dances: analysis of Salsa, ChaCha, and Rumba, 15th International Workshop, FSE 2008, pp. 470-488, February, Lausanne, Switzerland.

Aumasson J.P., Henzen L., Meier W., and Naya-Plasencia M., 2013. Quark: a lightweight hash. *J Crypto*, Springer-Verlag, 26(2): 470-488

Babbage S. and Dodd M., 2008, The MICKEY Stream Ciphers, New Stream Cipher Designs, LNCS 4986, pp. 191-209, September, Vienna, Austria.

Baharon M.R., Shi Q., and Llewellyn-Jones D., 2015. A new lightweight homomorphic encryption scheme for mobile cloud computing, 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. pp 618–625, October, Liverpool, UK.

Bansod G., Raval N., and Pisharoty N., 2015, Implementation of a new lightweight encryption design for embedded security. *IEEE Trans Inf Forens Sec* , IEEE, 10(1): 142-151.

Baskar C., Balasubramaniyan C., and Manivannan D., 2016. Establishment of light weight cryptography for resource constraint environment using FPGA. *Procedia Computer Science*, Elsevier, 78: 165–171.

Belare M. and Namprempre C., 2000. Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm, Okamoto T. (eds) Advances in Cryptology — ASIACRYPT 2000, the 6th International Conference on the Theory and Application of Cryptology and Information Security, vol 1976, pp. 531-545, December, Kyoto, Japan.

Berbain C., Billet O., Canteaut A., Courtois N., Gilbert H., Goubin L., Gouget A., Granboulan L., Lauradoux C., Minier M., Pornin T., and Sibert H., 2008, Sosemanuk, a fast software-oriented stream cipher. New Stream Cipher Designs, LNCS 4986, pp. 98-118, September, Vienna, Austria.

Bernstein D.J., 2008, The Salsa20 family of stream ciphers, New Stream Cipher Designs, LNCS 4986, pp. 84-97, September, Vienna, Austria.

Biham E., 1991. Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT'91, the 10th annual international conference on Theory and application of cryptographic techniques, pp. 532-534, April, Brighton, UK.

Biham E., Anderson R., and Knudsen L., 1998. Serpent: A New Block Cipher Proposal,5th International Workshop, FSE' 98, pp. 222-238, March, Paris, France.

Biryukov A. and Shamir A., 2000. Cryptanalytic time/memory/data tradeoffs for stream ciphers, Asiacrypt 2000 - the Sixth Annual ASIACRYPT Conference, pp. 1-13, December, Kyoto, Japan.

Biswas K., Muthukkumarasamy V., and Singh K., 2015, An encryption scheme using chaotic map and genetic operations for wireless sensor networks. *IEEE Sensors J*, IEEE, 15(5): 2801-2809.

Bloom B.H., 1970. Space/Time Trade-offs in Hash Coding with Allowable Errors, *Communications*, ACM, 13(7): 422-426.

Boesgaard M., Vesterager M., Pedersen T., Christiansen J., and Scavenius O., 2003, Rabbit: A New High-Performance Stream Cipher. *Johansson T. (eds) Fast Software Encryption. FSE 2003. Lecture Notes in Computer Science*, Springer, 2887: 307–329.

Boesgaard M., Vesterager M., and Zenner E., 2008. The Rabbit Stream Cipher, New Stream Cipher Designs, LNCS 4986, pp. 69–83, September, Vienna, Austria.

Bogdanov A. Knudsen L.R., Leander G., Paar C., Poschmann A., Robshaw M.J.B, Seurin Y., and Vikkelsoe C., 2007. PRESENT: An ultra-lightweight blockcipher, Cryptographic Hardware and Embedded Systems - CHES 2007, pp. 450–466, September, Vienna, Austria.

Bogdanov A., Knežević M., Leander G., Toz D., Varıcı K., and Verbauwhede I., 2011. SPONGENT: a lightweight hash function, International Workshop on Cryptographic Hardware and Embedded Systems CHES 2011, pp. 312–325, September, Nara, Japan.

Bossuet L., Grand M., Gaspar L., Fischer V., and Gogniat G., 2013. Architectures of flexibles ymmetric key crypto engines — a survey: From hardware coprocessor to multicrypto-processor system on chip. *ACMComputSurv2013*, ACM, 45(4) Article no. 41, 41: 1–32.

Bossuet L., Ngo X.T., Cherif Z., and Fischer V. 2014. A PUF Based on a Transient Effect Ring Oscillator and Insensitive to Locking Phenomenon. *IEEE Transactions on Emerging Topics in Computing*, 2(1): 30-36.

Boyle D. and Newe T., 2007. A Survey of Authentication Mechanisms, Authentication for Ad-Hoc Wireless Sensor Networks, SAS 2007 IEEE Sensors Applications Symposium, pp. 1-6, San Diego, California, USA.

Broenink G., Hoepman J.H., Hof C.v., Kranenburg R.v., Smits D., and Wisman T., 2010. The Privacy Coach: Supporting customer privacy in the Internet of Things, Michahelles F. (Ed.) What can the Internet of Things Do for the Citizen? (CIOT), pp. 72-81, Nijmegen, Netherlands.

Bunge M., 1974. On reference in relation to denotation and designation in "Sense and Reference", *Treatise on basic philosophy*, Semantics I, 1: 33-82.

Cannière C.D. and Preneel B., 2008. TRIVIUM, New Stream Cipher Designs, LNCS 4986, pp. 244–266, September, Vienna, Austria.

Cao J., Carminati B., Ferrari E., and Tan K.L., 2009. ACStream: Enforcing Access Control over Data Streams, ICDE, pp. 1495–1498, April, Shangai, China.

Cazorla M., Marquet K., and Minier M., 2013. Survey and benchmark of lightweight block ciphers for wireless sensor networks, 2013 International Conference on Security and Cryptography (SECRYPT), pp. 1-6, July, Reykjavik, Iceland, Iceland.

Chan C.W., Ee K.G., Ng C.K., Hashim F., and Noordin N.K., 2011. Development of 6LoWPAN Adaptation Layer with Fragmentation and Reassembly Mechanisms by Using Qualnet Simulator, International Conference on Informatics Engineering & Information Science (ICIEIS2011), pp. 199-212, November, Kuala Lumpur, Malaysia.

Chen Y., Leong H.V., Xu M., Cao J., Chan K.C.C., and Chan A.T.S, 2006. In-Network Data Processing for Wireless Sensor Networks, MDM '06 Proceedings of the 7th International Conference on Mobile Data Management, pp. 26, May, Nara, Japan.

Chen J. and Chen C., 2014. Design of Complex Event-Processing IDS in Internet of Things, 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, pp. 226-229, January, Zhangjiajie, China.

Cherif Z., Danger J., Guilley S., and Bossuet L., 2012. An Easy-to-Design PUF Based on a Single Oscillator: The Loop PUF, 15th Euromicro Conference on Digital System Design, pp. 156-162, September, İzmir, Turkey.

Chien H.C., 2007. Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity, *IEEE Transactions on Dependable and Secure Computing*, IEEE, 4: 337-340.

Choe Y.H., Kelly T., and Adolph M., 2008, Ubiquitous Sensor Networks, ITU-T Technology Watch Report, report no: 4, ITU, Geneva, Switzerland.

Conti M., Pietro R.D., and Spognardi A., 2014, Clone wars: Distributed detection of clone attacks in mobile WSNs. *Journal of Computer and System Sciences*, Elsevier, 80(3): 654-669.

Clark S., Frei S., Blaze M., and Smith J., 2010. Familiarity breeds contempt: The honeymoon effect and the role of legacy code in zero-day vulnerabilities, 26th Annual Computer Security Applications Conference (ACSAC 2010), pp. 251-260, December, Austin, Texas, USA.

Crowley P., 2005. Truncated differential cryptanalysis of five rounds of Salsa20, IACR Cryptology ePrint Archive 2005, report no: 2005/073, IACR Archive.

Cugola G. and Maragara A., 2012, Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.*, ACM, 44(3): 15:1–15:62.

Dawson E., Clark A, Golić J., Millan A., Penna L., and Simpson L., 2000. The LILI-128 Keystream Generator, the First Open NESSIE Workshop, pp. 61-63, November, Leuven, Belgium.

Devasena C.L., 2016, IPv6 Low Power Wireless Personal Area Network (6LoWPAN) for Networking Internet of Things (IoT) – Analyzing its Suitability for IoT *Indian Journal of Science and Technology*, Indian Society for Education and Environment, 9(30): DOI: 10.17485/ijst/2016/v9i30/98730, August 2016.

Douceur J., 2002. The Sybil Attack, First Interntional, Workshop on Peer-to-Peer Systems (IPTPS'02), pp. 251-260, October, Heidelberg, Berlin, Germany.

Eastlake D. and Jones P., 2001.Secure Hash Algorithm 1, *RFC 2001*, RFC, 3174: 1-22.

Eisenbarth T., Kumar S., Paar C., Poschmann A., and Uhsadel L., 2007, A survey of lightweight cryptography implementations. *Des Test Comput*, IEEE, 24(6): 522–533.

Dvir A., Holczer T., and Buttyan L., 2011. VeRA - Version Number and Rank Authentication in RPL, Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, pp. 709-714, October, Valencia, Spain.

Ekdahl P. and Johansson T., 2002, 9th Annual International Workshop, SAC 2002, pp. 47-61, August, St. John's, Newfoundland, Canada.

Ernest W., 2017, Light primitives and new technologies are driving the next generation of lightweight cryptography, http://semiengineering.com/lightweight-cryptography-for-the-ioe, [last access date: 01.02.2017].

Eswari T. and Vanitha V., 2013. A novel rule based intrusion detection framework for Wireless Sensor Networks, the International Conference on Information Communication and Embedded Systems (ICICES '13), pp. 1019–1022, February, Chennai, India.

Ferraiolo D.F., Gilbert D.M., and Lynch N., 1993. An Examination of Federal and Commercial Access Control Policy Needs, 16th NIST-NSA National Computer Security Conference, pp. 20-23, Baltimore, Maryland, USA.

Fischer S., Meier W., Berbain C., Biasse J.F., and Robshaw M.J.B., 2006. Nonrandomness in eSTREAM Candidates Salsa20 and TSC-4, 7th International Conference on Cryptology in India, pp. 2-16, December, Kolkata, India.

Fugkeaw S., and Sato H., 2016. Improved lightweight proxy re encryption for flexible and scalable mobile revocation management in cloud computing, 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), pp. 894–899, June-July, San Francisco, California, USA.f

Gislason D. 2008. Zigbee Wireless Networking, Newnes, pp. 42, October, Burlington, Massachusetts, USA

Grand M. Bossuet L., Le Gal B., Gogniat G., and Dallet D., 2011. Design and implementation of a multi-core crypto-processor for software defined radios, Reconfigurable Computing: Architectures, Tools and Applications - 7th International Symposium, ARC 2011, pp. 29–40, March, Belfast, UK.

Guajardo J, Kumar S.S., Schrijen G.J., and Tuyls P., 2007. FPGA Intrinsic PUFs and Their Use for IP Protection, Paillier P., Verbauwhede I. (eds) Cryptographic Hardware and Embedded Systems - CHES 2007, pp. 63-80, September, Vienna, Austria.

Guo J., Peyrin T., and Poschmann A., 2011. The PHOTON family of lightweight hash functions, Proceeding of Annual Cryptology Conference CRYPTO 2011, pp. 222–239, May, Santa Barbara, CA, USA.

Guo P., Wang J., Ji S., Geng X.H., and Xiong N.N., 2015, A lightweight encryption scheme combined with trust management for privacypreserving in body sensor networks. *J. of Medical Systems*, Springer, 39(12): 190–198.

Habutsu T, Nishio T., Sasase I. and Mori S., 1991. Workshop on the Theory and Application of Cryptographic Techniques, April, Brighton, UK.

Hammad A., Mokbel M.F., Ali M.H., Aref W.G., Catlin A.C., Elmagarmid A.K., Eltabakh M., Elfeky M.G., Ghanem T., Gwadera R., Ilyas I.F., Marzouk M., and Xiong X., 2004. Nile: A Query Processing Engine for Data Streams, 20th International Conference on Data Engineering, pp. 851, April, Boston, Massachusetts, USA.

Hell M., Johansson T., and Meier W., 2007, Grain - a stream cipher for constrained environments. *International Journal of Wireless and Mobile Computing*, Inderscience Publishers, 2(1): 86-93

Hill J., Szewczyk R., Woo A., Hollar S., Culler D., and Pister K., 2000. System architecture directions for networked sensors, ACM ASPLOS IX, pp. 93–104, November, Cambridge, Massachusetts, USA.

Hirose S., Ideguchi K., Kuwakado H., Owada T., Preneel B., and Yoshida H., 2010. A lightweight 256-bit hash function for hardware and low-end devices: lesamnta-LW, 13th International Conference ICISC 2010, pp. 151–168, December, Seoul, Korea.

Hong J. and Kim W., 2005. MD-Tradeoff and State Entropy Loss Considerations of Streamcipher MICKEY, Indocrypt 2005 LNCS 3797, pp. 169-182, August, Bangalore, India.

Hodjat A. and Verbauwhede I., 2004. High-throughput programmable crypto coprocessor*IEEE Micro*, IEEE, 24(3): 34–45.

Hongjun W., 2008. The Stream Cipher HC-128, New Stream Cipher Designs, LNCS 4986, pp. 39–47, September, Vienna, Austria.

Huang Q., Yang Y., and Shen M., 2016. Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing. *Fut Gen Comput Sys*, Elsevier, 72: 239–249.

Hummen R., Hiller J., Wirtz H., Henze M., Shafagh H., and Wehrle K., 2013. 6LoWPAN fragmentation attacks and mitigation mechanisms, the sixth ACM conference on Security and privacy in wireless and mobile networks, pp. 55-66, April, Budapest, Hungary.

IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), in IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006) , Sept. 5 2011, pp. 229

, Iova O., Picco P., Istomin T., and Kiraly C., 2016. RPL: The Routing Standard for the Internet of Things... Or Is It? *IEEE Communications Magazine*, IEEE, 55(12): 16-22.

James M. and Kumar D.S., 2016, An Implementation of Modified Lightweight Advanced Encryption Standard in FPGA. *Procedia Technology*, Elsevier, 25(2016): 582-589.

Jansen C. J. A., 2004. Streamcipher Design: Make your LFSRs jump!, the ECRYPT SASC (State of the Art in Stream Ciphers) workshop, pp. pp. 94–108, October, Bruges, Belgium.

Jakimoski G. and Kocarev L., 2001, Chaos and cryptography: block encryption ciphers based on chaotic maps. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, IEEE, 48(2): 163-169.

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

Karakoç F., Demirci H., and Harmancı A.E., 2015, AKF: A key alternating Feistel scheme for lightweight cipher designs. *Information Processing Letters*, Elsevier, 115(2): 359-367.

Karimian N., Guo Z., Tehranipoor F., Woodard D., Tehranipoor M., and Forte D. 2018. Secure and Reliable Biometric Access Control for Resource-Constrained Systems and IoT, arXiv preprint arXiv:1803.09710[last access date: 26.03.2018].

Karlof C., Sastry D., and Wagner D., 2004. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks, SenSys'04 - Proceedings of the Second International Conference on Embedded Networked Sensor Systems, pp. 162-175, November, Baltimore, Maryland, USA.

Kasinathan P., Pastrone C., Spirito M.A., and Vinkovits M., 2013. Denial-of-Service detection in 6LoWPAN based Internet of Things, 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 600-607, October, Lyon, France.

Kasinathan P., Khaleel H., Costamagna G., and Pastrone C., 2014. DEMO: An IDS framework for internet of things empowered by 6LoWPAN, the 2013 ACM SIGSAC conference on Computer & communications security (CCS Í3), pp. 1337-1340, November, Berlin, Germany.

Khazaei S., 2006. Posted on the eSTREAM Forum (2006), not publicly accessible but referenced in Canière and Preneel (2008), http://www.ecrypt.eu.org/stream/phorum/read.php?1,448.

Kim E., Kaspar D., Gomez C., 2012, Problem Statement and Requirements for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing. *RFC 2012*, RFC, 6606: 1 -32.

, Krawczyk H., Bellare M., and Canetti R., 1997. HMAC: Keyed-Hashing for Message Authentication, *RFC 1997*, RFC 2104: 1-11.

Kulseng L., Yu Z., Wei Y., and Guan U., 2010. Lightweight mutual authentication and ownership transfer for RFID systems, the IEEE Conference on INFOCOM, pp. 251–255, March, Piscataway, New Jersey, USA.

Kumar S.S., Guajardo J., Mae R., Schrije G.J., and Tuyls P., 2008. Extended abstract: The butterfly PUF protecting IP on every FPGA, 2008 IEEE International Workshop on Hardware-Oriented Security and Trust, pp. 67-70, June , Anaheim, California, USA.

Kurosawa S., Nakayama H., Kato N., Jamalipour A., and Nemoto Y., 2006, Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method. *International Journal of Network Security*, National Chung Hsing University, 5(3): 338-346.

Le A., Loo J., Luo Y., and Lasebae A., 2011, Specification-based IDS for securing RPL from topology attacks, 2011 IFIP Wireless Days (WD), pp. 1-3, October, Niagara Falls, Ontario, Canada.

Le A., Loo J., Chai K.K., and Aiash M., 2016, A Specification-Based IDS for Detecting Attacks on RPL-Based Network Topology *Information*, MDPI, 7(2): 25.

Leander G., Paar C., Poschmann A., and Schramm K, 2007. New lightweight DES variants, International Workshop on Fast Software Encryption, pp. 196–210, March, Luxembourg, Luxembourg.

Li L., Liu B., and Wang H., 2016, QTL: a new ultra-lightweight block cipher. *Microprocessors and Microsystems*, Elsevier, 45(A): 45-55.

Liang K., Au M.H., Liu J.K., Susilo W., Wong D.S., Yang G., and Yang A., 2015. A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. *Fut Gen Comput Sys*, Elsevier, 52: 95–108.

Lim D., Lee J.W., Gassend B., Suh G.E., van Dijk M., and Devadas S. 2005. Extracting secret keys from integrated circuits *IEEE Transactions on Very Large Scale Integration (VLSI)*, 13 (10): 1200-1205.

Linder W. and Meier J., 2006. Securing the Borealis Data Stream Engine, 10th International Database Engineering and Applications Symposium, pp. 137–147, December, Delhi, India.

Lopez G., Canovas O., Gomez-Skarmeta A.F., and Girao J., 2009. A SWIFT Take on Identity Management. *Computer*, IEEE, 42(5): 58-65.

Mallikarjunan K., Muthupriya K., and Shalinie S.M., 2016. A survey of distributed denial of service attack, 2016 10th International Conference on Intelligent Systems and Control (ISCO), pp. 1-6, January, Coimbatore, India.

MahajanV., Natu M., and Spognardi A., 2008. Analysis of Wormhole Intrusion Attacks in MANETs, MILCOM 2008 - 2008 IEEE Military Communications Conference, pp. 1-7, November, San Diego, California, USA.

Maiti A., Gunreddy V., and Schaumont P., 2012, A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions. Athanas P., Pnevmatikatos D., Sklavos N. (eds) Embedded Systems Design with FPGAs. Springer, New York, NY.

Matsui M., 1993. Linear Cryptanalysis Method for DES Cipher, Workshop on the Theory and Application of Cryptographic Technique EUROCRYPT '93, pp. 386-397, May, Lofthus, Norway.

Maximov A. and Biryukov A. 2007. Two trivial attacks on Trivium, the 14th international conference on Selected areas in cryptography, pp. 36-55, August, Ottawa, Canada.

Miorandi D., Scari S., Pellegrini F.D., and Chlamtac I. 2012. Internet of things: Vision, applications and research challenges *Ad Hoc Networks*, 10 (7): 1497-1516.

Missbach M., Staerk T., Gardiner C., McCloud J., Madl R., Tempes M., and Anderson G. 2015. SAP on the Cloud. Springer, 139

Mohd B.J., Hayajneh T., and Vasilakos A.V., 2015. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues *Journal of Network and Computer Applications*, Elsevier, 58: 73-93.

Mulligan G.. and Group L.W. 2007. The 6LoWPAN architecture, EmNets '07 Proceedings of the 4th workshop on Embedded networked sensors, pp. 78-82, June, Cork, Ireland.

Nabeel M., Zage J., Kerr S., Bertino E., Athula Kulatunga N., Sudheera Navaratne U., and Duren M. 2012. Cryptographic key management for smart power grids, CERIAS Tech. Report.

Naruse T., Mohri M., and Shiraishi Y., 2015. Provably secure attributebased encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating. *Human-centric Comput Inf Sci*, Springer, 5(1): 8–25.

Nath S. and Venkatesan R. 2013. Publicly Verifiable Grouped Aggregation Queries on Outsourced Data Streams, IEEE 29th International Conference on Data Engineering (ICDE), pp. 517-528, April, Brisbane, QLD, Australia.

Nehme R., Rundesteiner E., and Bertino E., 2008. A security punctuation framework for enforcing access control on streaming data, the 24th International Conference on Data Engineering, pp. 406–415, April, Cancun, Mexico.

Oliveira L.M.L., Rodrigues J. J. P. C., Neto C., and Sousa A.F.de., 2013. Network Admission Control Solution for 6LoWPAN Networks, Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 472-477, July, Taichung, Taiwan.

Pandey A. and Tripathi R.C., 2010. A Survey on Wireless Sensor Networks Security. *International Journal of Computer Applications (0975 – 8887)*, Foundation of Computer Science, 3(2): 43-49.

Papadopoulos S., Yang Y., and Papadias D., 2007. Cads: continuous authentication on data streams, the 33rd International Conference on Very Large Data Bases (VLDB), pp. 135–146, September, Vienna, Austria.

Papadopoulos S., Yang Y., and Papadias D., 2010. Continuous authentication on relational data streams *Very Large Data Bases (VLDB) journal*, 19(1): 161-180.

Papadopoulos S., Cormode G., Deligiannakis A., and Garofalakis M., 2013. Lightweight authentication of linear algebraic queries on data streams, the 2013 ACM SIGMOD International Conference on Management of Data, pp. 881–892, June, New York, USA.

Peng C., Du X., Li K., and Li M., 2016. An ultra-lightweight encryption scheme in underwater acoustic networks. *Journal of Sensors*, Hidawi, 2016: 1–10.

Perrey H., Landsmann M., Ugus O., Wählisch M., and Schmidt T.C., 2015. TRAIL: Topology Authentication in RPL, EWSN '16 Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks, pp. 59-64, Fubruary, Graz, Austria.

Perrig A., Canetti R., Tygar J.D., and Song D., 2000. Efficient authentication and signing of multicast streams over lossy channels, IEEE Symposium on Security and Privacy, pp. 56, May, Berkeley, California, USA.

Perrig A., Szewczyk R., Wen V., Culler D., and Tygar J., 2002. SPINS: security protocols for sensor networks. *Wireless Networks*, 8(5): 521-546.

Pfitzmann, A. and Hansen M. 2009. A Terminology for Talking About Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management., TU Dresden and ULD Kiel, available at: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf

Radomirović Saša, 2010. Towards a Model for Security and Privacy in the Internet of Things, First International Workshop on Security of the Internet of Things, November, Tokyo, Japan.

Raza S., Wallgren L., and Voigt T., 2013. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netoworks*, Elsevier, 11(2013): 2661–2674.

Rekleitis E., Rizomiliotis P., and Gritzalis S., 2010. A holistic approach to RFID security and privacy, 1st International workshop on the security of the Internet of Things. SecIoT'10, Network Information and Computer Security Laboratory, 2010; www.nics.uma.es/seciot10/files/pdf/rekleitis_seciot10_paper.pdf.

Rieback M.R., Gaydadjiev G.N., Crispo B., Hofman R.F.H., and Tannenbaum A.S., 2006. A Platform for RFID Security and Privacy Admini-stration, LISA '06, pp. 89-102, December, Washington DC., USA.

Rivest R.L., Shamir A., and Adleman L., 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, ACM, 21(2): 120-126.

Rivest R.L., 1994. The RC5 encryption algorithm, International Workshop on Fast Software Encryption 1994, pp. 86–96, December, Leuven, Belgium.

Roman R., Najera P., and Lopez J., 2011. Securing the Internet of Things, *Computer*, IEEE, 44(9): 51-58.

Sahraoui S. and Bilami A., 2015. Efficient HIP-based approach to ensure lightweight end-to-end security in the internet of things. *Computer Networks*, Elsevier, 91: 26–45.

Sandhu R.S., Coyne E.J., Feinstein H.L., and Youman C.E., 1996. Role-based access control models *Computer*, 29(2): 34-47.

Sarma A.C. and Girão J., 2009, Identities in the Future Internet of Things, *Wireless Personal Communication*, Springer, 49(3): 353–363.

Shirai T., Shibutani K., Akishita T., Moriai S., and Iwata T., 2007. The 128-Bit Blockcipher CLEFIA (Extended Abstract), Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 181–195. Springer, Luxembourg, Luxembourg.

Sicari S., Rizzardi A., Grieco L.A., and Coen-Porisini A. 2015. Security, privacy, and trust in Internet of Things: The road ahead. *Computer Networks*, 76: 146-164.

Singh S., Sharma P.K., and Moon S.Y., 2017. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions *Journal of Ambient Intelligence and Humanized Computing*, 10(54): 1-18. 1.25cmSnort Snort, the de facto standard for intrusion detection/prevention, http://www.snort.org/, [last access date: 22.02.2009]

Song N., Qian L., and Li X., 2005. Wormhole attacks detection in wireless ad hoc networks: a statistical analysis approach, 9th IEEE International Parallel and Distributed Processing Symposium, pp. 8, October, Denver, Colorodo, USA.

Suh G.E. and Devadas S., 2007. Physical Unclonable Functions for Device Authentication and Secret Key Generation, 44th ACM/IEEE Design Automation Conference, pp. 9-14, June, San Diego, California, USA.

Tillich S. and Großschädl J., 2006. Instruction set extensions for efficient AES implementation on 32-bit processors, 8th International Workshop CHES 2006, pp. 270-284, October, Yokohama, Japan.

Tsunoo Y., Saito T., Kubo H., Suzaki T., and Nakashima H., 2007. Differential cryptanalysis of Salsa20/8, Workshop Record of SASC 2007: The State of the Art of Stream Ciphers, report no: 2007/010, SASC.

Vasseur J.P., Agarwal N., Hui J., Shelby Z., Bertrand P., and Chauvenet C., 2011. RPL: The IP routing protocol designed for low power and lossy networks, Internet Protocol for Smart Objects (IPSO) Alliance, White Paper, April 2011.

Verma S, Pal S.K., and Muttoo S.K., 2014. A new tool for lightweight encryption on android, Proceeding of Advance Computing Conference (IACC), 2014 IEEE International, pp. 306–311, March, Gurgaon, India.

Vijayakumar A., Patil V.C., and Kundu S., 2017. On Improving Reliability of SRAM-Based Physically Unclonable Functions. *Journal of Low Power Electronics and Applications* , 7(1): 1-15.

Wagner D. and Soto P., 2002, Mimicry Attacks on Host Based Intrusion Detection Systems, the 9th ACM conference on Computer and communications security - CCS'02, pp. 255-264, November, Washington DC, USA.

Waschke M. 2017. Personal Cybersecurity: How to Avoid and Recover from Cybercrime, Apress, pp. 61, January, Bellingham, Washington, USA.

Wheeler D.J. and Needham R.M., 1994. TEA, a tiny encryption algorithm, International Workshop on Fast Software Encryption 1994, pp. 363-366, December, Leuven, Belgium.

Willems F., Shtarkov Y.M., and Tjalkens J. 1995. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory* , 41(3): 653 - 664.

Winter T., Thubert P., Brandt A., Hui J., Kelsey R., Pister K., Struik R., Vasseur J.P., and Alexander R., 2012. RPL: IPv6 routing protocol for low-power and lossy networks. *IETF 2012*, RFC, 6550.

Xu T., Wendt J.B., and Potkonjak M. 2014. Security of IoT Systems: Design Challenges and Opportunities, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 417-423, November, San Jose, California, USA.

Xu H., Ding J., Li P., Zhu F., and Wang R. 2018. A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function. *Sensors*, 18(3): 1-20, 760.

Yang Y., Zheng X., and Tang C., 2016. Lightweight distributed secure data management system for health internet of things. *Journal of Network and Computer Applications*, Elsevier, 89: 26-37.

Yao X., Chen Z., and Tian Y., 2015. A lightweight attribute-based encryption scheme for the Internet of Things. *Fut Gen Comp Sys*, Elsevier, 49: 104–112.

Yasmin R., Ritter E., and Wang G., 2010. An authentication framework for Wireless Sensor Networks using identity-based signatures, 10th IEEE International Conference on Computer and Information Technology, pp. 882-889, June, Bradford, UK.

Yu J, Khan G., and Yuan F., 2011. Xtea encryption based novel RFID security protocol, 24th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 58–62, May, Niagara Falls, ON, Canada.

Zegers W., Chang S.Y., Park Y. and Gao J., 2015, A lightweight encryption and secure protocol for smartphone cloud, 2015 IEEE Symposium on Service-Oriented System Engineering, pp. 259–266, March-April, San Francisco Bay, California, USA.

Zhang X., Heys Howard M., and Li C., 2013. FPGA implementation and energy cost analysis of two lightweight involutional block ciphers targeted to wireless sensor networks. *Mob Netw Appl 2013*, Springer, 18(2): 222-234.

Zhu S., Setia S., and Jajodia S. 2006. LEAP: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(4): 500-528.

CURRICULUM VITAE

Manolya ATALAY
manolyaatalay@akdeniz.edu.tr

EDUCATION DETAILS

Graduate 2016-2019	Akdeniz Üniverversi Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı, Antalya
Undergraduate 2007-2012	Dokuz Eylül Üniversitesi Mühendislik Fakültesi, Bilgisayar Mühendisliği Anabilim Dalı, İzmir

PROFESSIONAL AND ADMINISTRATIVE DUTIES

Research Assistant 2017 - Devam Ediyor	Akdeniz Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Anabilim Dalı, Antalya