

Continuous Authentication and Lightweight Implementation of Elliptic-Curve Cryptography for the Internet of Things

Lu Zhou

A DISSERTATION

SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

IN COMPUTER SCIENCE AND ENGINEERING

Graduate Department of Computer and Information Systems
The University of Aizu

2019



Copyright by Lu Zhou

All Rights Reserved






The thesis titled

Continuous Authentication and Lightweight Implementation of Elliptic-Curve Cryptography for the Internet of Things

by

Lu Zhou

is reviewed and approved by:

Chief referee			
<i>Professor</i>	Chunhua Su	Su Chunhua	
<i>Professor</i>	Anh T. Pham	Anh T. PHAM	
<i>Professor</i>	Akihito Nakamura		
<i>Professor</i>	Junbo Wang	Junbo Wang	

The University of Aizu

2019

Abstract

The Internet of Things (IoT) is a revolutionary innovation that seamlessly integrating an enormous number of smart objects within the Internet. It is imperceptibly affecting the way of human life and work. The Internet of Things is considered as the next generation network technology and will reach a total market capitalization of trillion-dollars. However, the Internet of Things is facing critical security and privacy problems which will hamper its large scale deployment. Traditional Internet security and privacy approaches cannot be directly applied to the Internet of Things due to the limited computing power and restrained resources of IoT devices.

Over the past decade, the research of security and privacy for the Internet of Things have made many achievements in both theory and practice. However, the Internet of Things is still in the infant stage as there are many security and privacy problems remain to be solved. This dissertation focuses on researching secure continuous authentication schemes for the emerging cloud-based IoT architecture and the endpoint devices security in IoT. Precisely, first, we presented a continuous authentication scheme using the bio-feature as the identity token for the cloud-based IoT network. Then, we presented a lightweight implementation of NIST P-256 and SM2 ECC, which is newly published by Chinese Commercial Cryptography Administration Office, and standardized at ISO in 2017, on a typical resource constrained IoT device, 8-bit AVR processors. The endpoint devices form a core part of the IoT network and cryptographic algorithm is the basic mechanism for secure endpoint devices. Finally, we provided a systematic exposition and proposed some research perspectives for security and privacy of IoT endpoint devices.

The main works of this dissertation are as follows.

Continuous user authentication with biological feature for cloud-based IoT

Continuous authentication is a crucial aspect for the cloud-based IoT security. We proposed a new prototype for continuous user authentication for the IoT network where a user's biological features such as the brainwave, retrieved by wearable devices, were adopted as the raw data. Machine learning-based techniques were leveraged to derive user's bio-features as authentication tokens in the system to support continual entity verification in the background without the user's notice. Implementation and evaluation demonstrate that the proposed continuous authentication scheme is efficient and achieves

high verification accuracy.

Implementation of NIST P-256 and SM2 ECC on 8-bit AVR processors

Cryptographic primitives are the cornerstone of security protocols for the IoT network. Optimize the implementation of cryptographic primitives is especially crucial for the resource constraint IoT devices. The elliptic curve cryptography (ECC) is one of the most important approaches to instantiate asymmetric encryption and signature schemes and has been extensively exploited to protect the security of the IoT network. We proposed a lightweight implementation of NIST P-256 and SM2 ECC, which is a newly proposed set of public key cryptographic algorithms based on elliptic curves published by Chinese Commercial Cryptography Administration Office, and was standardized at ISO in 2017, on 8-bit AVR processors. The comparison between SM2 and NIST was conducted to make better efficiency and security intuition to the IoT network.

Systematic exposition and research perspective for security and privacy of IoT

Endpoint devices form a core part of the IoT network and cryptographic algorithms is a core mechanism for secure endpoints. We provided a high-level introduction to IoT endpoint security requirements followed by a discussion on cryptographic algorithm implementation. We focused on an overview of efficient cryptography for IoT endpoints and system privacy issues. We first introduced existing cryptographic mechanisms for IoT, and discussed efficient algorithm implementation. Then we discussed key management approaches, positives, negatives and challenges to resolve, linking to the endpoint device security section with regards to realistic device needs/capabilities. Finally, We demonstrated a high level discussion on crypto for system level data security and privacy, including a discussion on mechanisms to ensure IoT adheres to privacy standards/legal compliance.

要旨

モノのインターネット (Internet of Things, IoT) は、インターネットに接続して通信機能を持つ膨大な数のモノ達をスマートに統合する革新的な技術である。この技術は人間の生活や仕事の方式に多大な影響を及ぼすつつある。モノのインターネットは次世代のネットワーク技術と見なされ、時価総額は1兆ドルに達するでしょう。しかし、モノのインターネットは、その大規模な展開を妨げる重大なセキュリティとプライバシーの問題に直面している。従来のインターネットセキュリティとプライバシーを守るためのプロトコルは、計算能力とメモリなどのリソースが限られるIoTデバイスに直接適用することはできない。

過去10年間で、IoTのセキュリティとプライバシーの研究は、理論と実践の両方で多くの成果を上げてきた。しかし、多くのセキュリティとプライバシーの問題が解決されていないため、IoTはまだ段階にあります。本博士論文は、新しいクラウドベースのIoTアーキテクチャとIoT端末向けの安全かつ継続的な認証方式の研究に焦点を当てる。具体的には、クラウドベースのIoTネットワークのIDトークンとして生体特徴を使用した連続認証方式を提案した。それから、NIST P-256およびSM2 ECCを典型的なリソース制約のあるIoTデバイス8ビットAVRプロセッサにおける軽量実装検証を行った。特にSM2 ECCは、中国の商業用暗号管理局によって新たに承認されたもので、2017年にISOで標準化された。IoT端末はIoTネットワークの中核部分であり、暗号化アルゴリズムはセキュア端末デバイスの基本的なメカニズムである。最後に、我々はモノの向けのセキュリティ研究に体系的なサーベイを提供し、IoT端末のセキュリティとプライバシーのために若干の研究展望を提案しました。

本博士論文の主な貢献は以下の通りである。

クラウドベースのIoTのための生物学的機能を備えた継続的なユーザー認証

継続的認証は、クラウドベースのIoTセキュリティにとって非常に重要な技術要素である。特にウェアラブルデバイスによって収集された脳波などのユーザの生体特徴が認証用のデータとして採用されるIoTネットワークのための継続的ユーザ認証のための新しいプロトタイプを提案した。ユーザの操作なしに

バックグラウンドで継続的な認証を実現するために、システム内の認証トークンとしてユーザーの生体特徴を抽出と識別するために、機械学習を利用した。我らの実装と評価は、提案した連続認証方式が効率的でありそして高いユーザー認証精度を達成することを示した。

8ビットAVRプロセッサにおけるNIST P-256とSM2 ECCの軽量化実装検証

暗号プリミティブは、IoTネットワークのセキュリティプロトコルの基本技術要素である。暗号化プリミティブの実装を最適化することは、リソース制約のあるIoTデバイスにとって特に重要である。楕円曲線暗号 (ECC) は、公開鍵暗号と電子署名方式を構築するための最も重要な技術の一つであり、IoTネットワークのセキュリティを保護するために広く利用されている。本博士論文は楕円曲線暗号の良く使用されるNIST P-256とSM2 ECCの軽量実装を提案した。さらにIoTネットワークに対する効率性とセキュリティの直感を向上させるため、IoT端末向けの8ビットAVRプロセッサで実装検証も行い、SM2とNISTの性能上の比較を実施した。

IoTのセキュリティとプライバシーの体系的なサーベイと研究展望

IoT端末はIoTネットワークの中核部分を構成し、暗号化アルゴリズムはIoT端末のセキュリティの一番重要なメカニズムである。IoT端末セキュリティ要件の概要を説明した後、暗号化アルゴリズムの実装についてもサーベイした。著者はIoT端末の効率的な暗号化の実装手法とおよびシステムのプライバシー問題の概要に焦点を当てた。まずIoTの既存の暗号化メカニズムを紹介し、効率的なアルゴリズムの実装について説明した。次に、IoTがプライバシー基準/法令順守を確実に実行するためのメカニズムに関する議論を行い、システムレベルのデータセキュリティとプライバシーに関する議論も行った。

Contents

Abstract	ii
1 Introduction	1
1.1 The State of the Art of IoT Security and Privacy	3
1.1.1 Continuous Authentication Schemes for Cloud-based IoT	3
1.1.2 Lightweight Implementation of ECC in IoT	3
1.1.3 Endpoint Security and Privacy for IoT	4
1.2 Motivation of the Dissertation	4
1.3 Contributions of the Dissertation	5
1.4 Structure of the Dissertation	6
2 Continuous Authentication with Brainwave Bio-feature for IoT	8
2.1 Continuous Authentication	8
2.2 Wearable Devices	12
2.3 Continuous Authentication with Brainwave Scheme	15
2.4 Implementation and Evaluation	16
2.5 Chapter Conclusion	22
3 Implementation of NIST P-256 and SM2 ECC on 8-bit AVR Processors	24
3.1 ECC Implementation in IoT Devices Background	24
3.1.1 Elliptic Curve Cryptography	25
3.1.2 NIST Versus SM2	27
3.1.3 8-bit AVR Processors	29
3.1.4 Previous Implementations on 8-bit AVR Processors	30
3.2 Efficient Implementation of NIST P-256 and SM2	31

Contents	vii
3.2.1 Finite Field Arithmetic	31
3.2.2 Elliptic Curve Group Arithmetic	39
3.3 Implementation and Evaluation	42
3.4 Chapter Conclusion	43
4 Endpoint Security and Privacy for IoT: Systematic Exposition	45
4.1 Security Requirements of IoT Endpoint Devices	46
4.2 Cryptographic Solutions for Endpoint Devices	48
4.3 Asymmetric Cryptographic Algorithms for IoT Endpoint Devices	49
4.4 Device Key Management for IoT	54
4.4.1 Secure Key Generation	55
4.4.2 Secure Key Storage and Retrieval	58
4.5 Privacy for IoT	60
4.5.1 Data Confidentiality, Integrity and Authenticity	62
4.5.2 Privacy Protection on Cloud Servers with Big Data Analysis	62
4.5.3 Privacy Management on Endpoint Devices	63
4.6 Chapter Conclusion	65
5 Conclusions and Future Research	66
5.1 Conclusions	66
5.2 Future Research	67
Acknowledgements	68
Bibliography	69

List of Figures

2.1	Continuous Authentication.	10
2.2	The State of The Art of Continuous Authentication.	10
2.3	Schematic Flow of Proposed Authentication Scheme.	15
2.4	BR8 PLUS Headset Used to Monitor Brainwaves During Experiments. . .	16
2.5	Functionality of Each Part of The Brain.	17
2.6	Test Photos Used in Experiments.	18
2.7	Training Data Set.	18
2.8	Diagram of Improved System.	21
3.1	The execution flow of computing 4-word multiplication using the Hybrid Method and Reverse Product Scanning Method respectively	36
4.1	Lightweight Key Generation for IoT devices	56
4.2	Matrix-based Oblivious Random Access Mechanism	59
4.3	Back-End and Human Interface of IoT	60

List of Tables

2.1	Symbols and their meanings	9
2.2	Authentication Accuracy With Alpha, Beta, Gamma, Delta, Theta Signals.	19
2.3	Authentication Accuracy With Data Purification and NB.	19
2.4	Authentication Accuracy With Data Purification and SVM-GRBF.	20
2.5	Authentication Accuracy With Data Filtration and NB.	21
2.6	Authentication Accuracy With Data Filtration and SVM-GRBF.	22
3.1	Comparison of Multi-precision Multiplication/Squaring Implementations on 8-bit AVR Processors.	37
3.2	Cycle counts of Finite Field Arithmetic Software for AVR ATmega128 Microcontrollers.	43
3.3	Cycle Counts and Code Size of Scalar Multiplication Software for AVR ATmega Microcontrollers.	44
4.1	IoT consortium and OpenFog Security Objectives and Recommendations	46
4.2	the execution time of existing ECC-based implementations for IoT end- point devices	53

List of Abbreviation

SVM	Support Vector Machine
CSP	Cyber-Physical Systems
PLC	Programmable Logic Controller
IC	Internet Consortium
AES	Advanced Encryption Standard
IFP	Integer Factorization Problem
	concatenation function
DH	Diffie-Hellman
ECC	Elliptic Curve Cryptography
PPT	Probabilistic Polynomial Time
ECDH	Elliptic-Curve Diffie-Hellman
ECDLP	Elliptic-Curve Discrete Logarithm Problem
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDLP	Elliptic-Curve Discrete Logarithm Problem

Chapter 1

Introduction

The Internet of Things (IoT) paradigm is one of the most thrilling innovations of information technology. It is predicted that there will be 50 billion devices connected to the internet by 2020 [1]. With the rapid growth and universality of information and communication technologies of IoT, numerous ubiquitous applications have found an increasingly wide deployment in diverse daily-operated services to probe for more business opportunities or higher individual benefit. For example, a smart home consisting of smart IoT-devices may provide tailored and on-demand entertainment services to accomplish better satisfaction for individuals. Another example is individuals gradually changing their purchasing styles from classic credit cards to new approaches such as online payments via wearable devices.

However, the development of the Internet of Things is affected by the numbers of possible security issues which grow exponentially in the past decades. The Internet of Things is more vulnerable to attacks due to the lack of security measures and open deployment environment [2]. There have been many IoT attacks [3], which have led a lost of billion dollars.

Authentication security and endpoint security are the basis of the security for IoT systems. However, the IoT systems are an attractive target for attacks due to its constrained resources. Securing authentication and endpoint devices are made more difficult by a variety of device hardware and system restrictions including limited device energy, memory and processing resources, communication latencies, message size and real-time operation [4]. Implementing traditional security techniques might fail, as the time the

device dedicates to execute these techniques will delay the handling of its core function, which could be unacceptable in time-critical contexts. Efficient security solutions for the IoT (and other technology contexts) that are capable of securing systems regardless of limitations on power consumption, processing capacity and memory footprint are of a high priority if we are to satisfy the security expectations users and operators have for applications. Security solutions for endpoint devices also applies to other growing technologies such as edge and fog computing. The fog computing paradigm aims to enable real-time analysis and faster actuation of sensor data by moving computation, control and storage closer to the network edge in an IoT network [5].

Recently, with the widely deployed wearable device and smart phone, people are facing the problem of massive IoT data maintenance. The cloud-based IoT architecture, in which the IoT data was outsourced and maintained by the cloud was proposed to deal with the huge data in the IoT. This alleviates the problem of limited computing power in the Internet of Things to some extent. However, secure data retrieval and robust access control becomes a new security problem for the cloud-based IoT architecture. Authentication for data retrieval and access control is crucial for the cloud-based IoT architecture.

On the other hand, cryptographic primitives are the cornerstone of security protocols in the Internet of Things. The elliptic curve cryptography is one of the most important approaches to instantiate asymmetric encryption and signature protocols, which has been extensively exploited to protect the security of cyber-physical systems. With the advent of the Internet of Things (IoT), the elliptic curve cryptography primitives have been widely deployed for its security. Implementing elliptic curve cryptography in resource constraint IoT devices is especially crucial for IoT security.

Though, there have been many works studied the security and privacy problems of IoT, the is rare work that presents a high-level exposition of the IoT endpoint security. However, endpoint devices form a core part of the IoT architecture. Securing endpoint devices is made more difficult by a variety of device hardware and system restrictions including limited device energy, memory and processing resources, communication latencies, message size and real-time operation [4]. Although there is a mature set of algorithms available, challenges remain in terms of efficient cryptographic algorithm implementation, sharing similarities with efficient implementation of functions for digital

signal processing, in the context of various constraints associated with the IoT. Endpoints are largely heterogeneous, with a wide range of overarching applications and resources, and therefore need efficient implementation approaches with regards logical processing, memory required and execution time.

1.1 The State of the Art of IoT Security and Privacy

1.1.1 Continuous Authentication Schemes for Cloud-based IoT

Recent years have seen a series of studies investigating continuous authentication (also called continuous authentication) with various bio-authentication tokens, such as electrocardiography (ECG) [6–8], iris & eye movement [9, 10], face [11, 12], ear [13], voice [14, 15], finger nail [16], handwritten signature [17], keystroke dynamics & on-screen movement [18–28], text analysis [29,30], brainwave [31] and gait gesture [32,33]. Unfortunately, most existing continuous authentication schemes require complex devices, such as those used for the extraction of bio-data, which make them unsuitable for most end-users.

1.1.2 Lightweight Implementation of ECC in IoT

With the advent of the Internet of Things, a great deal of constrained devices (such as 8-bit microcontrollers) are now widely used in cyber-space for pervasive applications like wireless sensor networks (WSN) or RFID tags. Compared with traditional cable networks, security in these sensor networks has been an ever-increasing challenge mainly due to their widespread acceptance and wireless nature. The restriction of computation capability, storage space and even energy consumption for such constrained devices also increases the difficulty in deploying cryptographic schemes. But even so, a lot of cryptographic techniques have been investigated to ensure the security of such devices, of which the elliptic curve cryptography is paid much attention owing to its lightness and security [34–42]. The ECC now is included in current standards like those of ISO, ANSI, IETF, BSI as well as NIST, and has become a common approach to instantiate asymmetric encryption and signature schemes.

The SM2 [43], a set of public key cryptographic algorithms based on elliptic curves released by Chinese Commercial Cryptography Administration Office in December 2010, was standardized at ISO in 2017. It is the first standard of the digital signature algorithm which has been used in electronic authentication service system in China. The SM2 has won the eyes of fields from both academy and industry, based on which many implementations and applications have come to being [44]. However, few research works on implementing SM2 for constrained devices have been conducted.

1.1.3 Endpoint Security and Privacy for IoT

IoT endpoint devices are an attractive target for attacks, and therefore it is critical that we protect the large-scale and often unmonitored deployment of devices [45]. Securing endpoint devices is made more difficult by a variety of device hardware and system restrictions including limited device energy, memory and processing resources, communication latencies, message size and real-time operation [4]. Implementing traditional security techniques might fail, as the time the device dedicates to execute these techniques will delay the handling of its core function, which could be unacceptable in time-critical contexts. Efficient security solutions for the IoT (and other technology contexts) that are capable of securing systems regardless of limitations on power consumption, processing capacity and memory footprint are of a high priority if we are to satisfy the security expectations users and operators have for applications. Security solutions for endpoint devices also applies to other growing technologies in applications such as edge and fog computing. The fog computing paradigm aims to enable real-time analysis and faster actuation of sensor data by moving computation, control and storage closer to the network edge in an IoT network [5].

1.2 Motivation of the Dissertation

The Internet of Things has been dominating the Information Technology industries, since it can significantly extend the edge of the Internet. The Internet of Things has been widely used in many aspects including the Internet of Industry, the Internet of Vehicles and so on. The security problems is one of the main obstacles to the development of

the Internet of Things. Confidentiality, integrity, and authenticity are the basic security requirements of IoT. Cryptography is the basic cornerstone to achieve these requirements. Though, there have been many cryptographic protocols designed for IoT security, there are still many challenges in cryptographic protocols in IoT.

Firstly, with the development of wearable devices and smart phone, massive data are generated in the IoT. The IoT architecture is changing to the cloud-based architecture from the traditional model. There is no practical authentication protocols for the cloud base architecture. Continuous authentication is an enhanced requirement for IoT security. Achieving continuous authentication using the biology token for IoT is an emerging technique and interesting problem.

Secondly, as cryptographic protocols are the basic component in the security of IoT, how to securely and efficiently implement lightweight cryptographic protocols in the resource constrained IoT devices is critical for the IoT security.

Finally, since there is rare work that gives a high-level view of the endpoint security in IoT, we aim to provide the first systematic exposition of the cryptography implementation of IoT and present some research perspectives for security and privacy in IoT. This will help the fresh researchers and engineers of IoT to get started quickly.

1.3 Contributions of the Dissertation

In this dissertation we proposed a new authentication schemes in the cloud-based IoT architecture. Further, we investigated the lightweight implementations of cryptographic primitives on resource constraint devices. Finally, we provided a high-level introduction to IoT endpoint security requirements followed by a discussion on cryptographic algorithm implementation. The contribution of the dissertation is list as follows:

- We introduced a novel and efficient continuous authentication protocol for cloud-based IoT architecture. We refines the proposed continuous authentication protocol which using the brain wave as the identity token. We integrated the bio-feature authentication protocol with the machine learning technique to improve the accuracy of the continuous authentication protocol.
- We, for the first time, proposed an efficient, secure and compact implementation of

scalar multiplication on a 256-bit elliptic curve recommended by the SM2, as well as a comparison implementation of scalar multiplication on the same bit-length elliptic curve recommended by NIST. We re-designed the existent techniques to fit the low-end IoT platform, namely 8-bit AVR processors.

- We provided a high-level introduction to IoT endpoint security requirements followed by a discussion on cryptographic algorithm implementation. And then, we discussed some system-wide design considerations for data security and privacy in current and emerging system designs.

1.4 Structure of the Dissertation

The rest of the dissertation is organized as follows:

- Chapter 2 proposed a continuous authentication scheme with bio-features for the IoT, in which the brainwave bio-feature is sampled as the secure token for the authentication scheme.
- Chapter 3 introduced a lightweight implementation of NIST P-256 and SM2 elliptic curve cryptography on a resource constraint device, 8-bit AVR.
- Chapter 4 provided a systematic exposition of the implementation of cryptographic algorithms in IoT and presented some research perspectives of security and privacy implementation for the IoT.
- Chapter 5 concludes this dissertation and offers discussions about the points for future work.

Publications

The following papers have been published in peer reviewed journals.

Referred Journals

1. **Lu Zhou**, Chunhua Su, Wayne Chiu, Kuo-Hui Yeh, “You Think, Therefore You Are: Transparent Authentication System with Brainwave-oriented Bio-features for IoT Networks”, *IEEE Transactions on Emerging Topics in Computing*. No.99 (2017): 1-1.
2. **Lu Zhou**, Chunhua Su, Zhi Hu, Sokjoon Lee, Hwajeong Seo, “Lightweight Implementations of NIST P-256 and SM2 ECC on 8-bit Resource-Constraint Embedded Device”, *ACM Transactions on Embedded Computing Systems*, Vol.18 (2019): 23:1-23:13.
3. **Lu Zhou**, Kuo-Hui Yeh, Gerhard Hancke, Zhe Liu, Chunhua Su, “Security and Privacy for the Industrial Internet of Things: An overview of approaches to safeguarding endpoints”, *IEEE Signal Processing Magazine*, Vol.35, No.5 (2018): 76-87.

Chapter 2

Continuous Authentication with Brainwave Bio-feature for IoT

The universal Internet connectivity of such smart objects has brought about a new era of ubiquitous application development for the Internet of Things. Meanwhile, security has become critically important. Academia and industry have dedicated great efforts to the design of continuous authentication for multi-modal networks in the past decade. Multi-form authentication bio-tokens have been introduced for continuous entity identification and verification. With the rapid growth and universality of wearable devices, in this article we target continuous authentication for IoT-based environment with users possessing wearable related smart objects. In this chapter, we provide a comprehensive review of continuous authentication in recent years and introduced critical characteristics of new biometrics. Further, we present a wearable brainwave bio-feature extractor constructed via sensors. The prototype is adopted to retrieve user's brainwave bio-data as the raw data in the proposed authentication system. Finally, we apply machine learning-based techniques to derive user's brainwave bio-features as authentication tokens in the system to support continual entity verification in the background.

2.1 Continuous Authentication

In traditional authentication, authentication tokens are usually presented as the forms of what we know (e.g., user name and password), what we have (e.g., smartcard or key-

Table 2.1: Symbols and their meanings

Schemes	Bio-Features	Performance		
		FAR	FRR	EER
scheme [31]	Low-frequency brainwaves	NA	NA	0.03%-0.4%
scheme [9]	Iris	NA	NA	0%
scheme [13]	Shape of ear, tragus	0%	00.15%	NA
scheme [11]	Face image	NA	0.7%-13.7%	NA
scheme [46]	Voice	0.01%	15%	NA
scheme [8]	Electrocardiography(ECG)	NA	NA	2.007%
scheme [17]	Handwritten signature, secure keys derived from passwords	NA	NA	3.4%
scheme [21]	Keystroke and mouse usage behaviors	0.1%	5.7%	NA
scheme [23]	Keystroke behavior from frequency spectrograms	NA	NA	4.1%
scheme [29]	Inputted-text (lexical, syntactic, n-gram analysis)	NA	NA	9.98%-21.45%
scheme [32]	Gait gesture	NA	NA	0%-10.8%
scheme [7]	Electrocardiography (ECG)	1.57%	0.39%	7.89%-10.10%
scheme [28]	Click-draw based graphical password	NA	NA	NA

token), and what we are (e.g., biometrics derived from physiological signals or user behaviors), respectively. Traditional authentication protects the security only at the login point and cannot guarantee the security from login to logout during system operation. Therefore, the continuous authentication mechanism is indispensable to fill in the gap that may be vulnerable. In general, a continuous authentication is represented as that in Figure 2.1 in which a biometric-based verification mechanism is continually performed in the background to support the robustness of the system operation processes. The authentication token during verification process is based on a unique pattern derived from physiological signals or behavior events. Significant efforts have been dedicated to this interesting and important area. In the following, we present the state of the art of continuous authentication. Figure 2.2 summarizes the existing continuous authentication system based on ten classifications according to their adopted bio-features, such as brainwave [31], iris & eye movement [9], ear [13], face [11], voice [46], electrocardiography (ECG) [8] [7], handwritten signature [17], keystroke dynamics & on-screen movement [21,23] [28], text analysis [29] and gait gesture [32]. In addition, we present a comparative figure, i.e. Table 2.1, for all of the investigated studies in terms of the adopted algorithms for classifier extraction, the adopted bio-features and the performance of each method. Note that the general metrics for the performance evaluation of continuous authentication are as follows:

- False Acceptance Rate (FAR): Rate of invalid users identified as legitimate users.



Figure 2.1: Continuous Authentication.

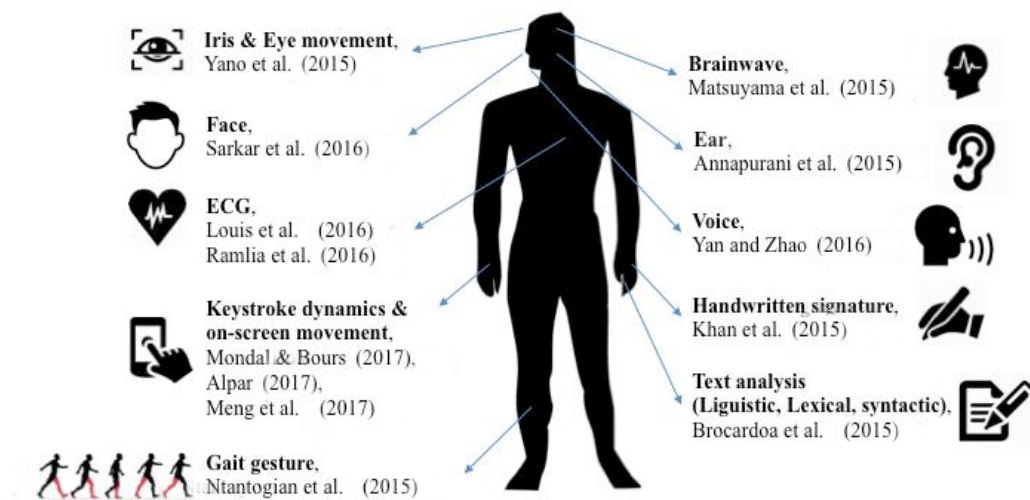


Figure 2.2: The State of The Art of Continuous Authentication.

- False Rejection Rate (FRR): Rate of legitimate users identified as invalid users.
- Equal Error Rate (EER): A common value is referred to as EER when FAR equals to FRR.

In 2015, Matsuyama et al. [31] proposed an authentication based on low-frequency brain signals in which the oxyhemoglobin changes in the brain are measured through near-infrared spectroscopy. The authors presented two experiments, i.e. resting and keyboard typing, for evaluating the performance of their system. They demonstrated that an equal error rate (EER) of less than 1% can be achieved with the support of principal component analysis and a support vector machine (SVM). In the same year, Yano et al. [9] extracted the dynamic features from the pupillary light reflex, and combined it with the static features from the iris pattern to construct a multimodal biometric authentication system. The

authors then examined the authentication accuracy in terms of EER, and showed a perfect performance, i.e. 0% EER, could be achieved via their experiments on a pupillometric database created from 90 volunteers. Next, Annapurani et al. [13] proposed an authentication method with two biometric features, i.e. the human ear's tragus and the shape of the ear, derived via an automatic edge detection method. The authors exploited the Hamming distance technique and the Euclidean distance method to calculate the differences between the comparison target and the maintained biometric features. Their scheme enjoyed higher accuracy on entity verification than traditional authentication schemes, and the authors also concluded the tragus is better than ear shape when it comes to serving as a robust authentication token. In 2016, Ramlia et al. [8] developed a wearable bracelet for user's ECG data acquisition. In addition, the authors adopted wavelet transform as the feature extraction method and SVM as a classification algorithm to construct a biometric authentication system. The demonstrated experiment results showed that a rate of 2.0069% of EER performance can be achieved. Then, Khan et al. [17] presented a two-factor authentication mechanism adopting biometric features (e.g., user's handwritten signatures) and cryptographic based factors (e.g., keys derived from user passwords). The performance evaluation was then performed via three publicly available signature datasets demonstrate security claims, i.e. (1) a template-level security for handwritten signatures can be provided and (2) a reasonable EER ratio can be guaranteed.

Recently, Mondal and Bours [21] presented a continuous authentication system based on the patterns of user's keystrokes and mouse usage behaviors. The regression models adopted for data classification are the Artificial Neural Network, Counter-Propagation Artificial Neural Network and SVM. In the experiment, 50 out of 53 testers could pass the examination of the proposed system (5.7% of FRR) and 0.1% of the impostors (FAR) were undetected. Next, Alpar [23] proposed an authentication system with a keystroke-based classifier in a frequency domain. In contrast to other studies with data in a time domain, the author used the Fourier transformation with the optimized window size to generate the spectrograms of the retrieved data. Next, the outputted spectrograms were exploited to conduct the classifier via the Gauss-Newton based Neural Network algorithm. A result of 4.1% of EER is obtained with the scenario of 60 real attempts made by legitimate users and 60 fraud attacks from 12 malicious users. After that, Ntantogian et al [32]

demonstrated a two-factor authentication system with gait features. To eliminate the noise and distortions caused by a user's silhouette, the authors conducted a weighted sum-based method to further evaluate the bio-factor of gait features extracted from the user. In terms of the attacker's ability, three types of imposters were simulated in the experiments. The authors concluded that the proposed authentication system can achieve, at best, 0% of EER against general imposters. However, if an imposter possesses a valid token stolen from a genuine user, the imposter will have a probability of 10.8% of breaking through the proposed system. Next, Louis et al. [7] proposed a scheme, called one-dimensional multi-resolution local binary patterns (1DMRLBP), to derive ECG-based biometrics for continuous authentication. The system adopts the sequential sampling technique and 1 DMRLBP feature extraction method to attain better authentication accuracy, such that a 0.39% false rejection rate (FRR) and 1.57% false acceptance rate (FAR) can be guaranteed.

2.2 Wearable Devices

Nowadays, our lives are deeply impacted by IoT-based applications, which has led us to look ahead to future possibilities in this field. Advances in information communication technologies on smart objects have transformed network connections from a rarity to a ubiquitous infrastructure, changing computing patterns from single-and-sited to scalable-and-networked in the process. Versatile smart objects, such as specific-purpose sensors and intelligent wearable devices, are universally deployed and seamlessly integrated into our daily life to support tailored IoT-based application services for individuals in an on-demand and real-time manner. In IoT-based applications, smart objects are capable of interconnection, storing data and receiving user commands to accomplish the tasks users request and desire. In addition, heterogeneous communication architectures may be formed when various types of smart objects and relevant communication techniques, such as Radio Frequency (RF), Bluetooth Low Energy (BLE), Zeebe, LoRa and WiFi, are adopted in IoT-oriented environments.

Wearable devices are without doubt one of the most promising paradigms for ubiquitous computing in the IoT. Good examples include fitness bands (activity trackers), run-

ning watches and wearable glasses, all of which are considered "things" that are capable of connectivity to the Internet for enabling such "things" to exchange data without human intervention. CCS Insight indicated in 2016 that the future of wearable devices will see 411 million smart wearable devices, worth a staggering \$34 billion, sold in 2020 [47]. It is predicted that 97 million pieces of eyewear, 9 million hearables, 164 million wristbands, 25 million wearable cameras, 110 million watches and 4 million tokens, clip-ons and jewelry will be included in device sales in 2020. As wearable technologies become more popular, it is necessary to take stock of efficiency, benefits and the security risks of wearable -based network models. We summarize our observations below.

Wearable devices collect and store an unprecedented volume of personal data about users' daily lives. Without any built-in data protection mechanisms, such as PINs, passwords, fingerprinting or encryption, everyone is able to access the contents stored on a wearable device once it is lost or stolen. Meanwhile, invasion of privacy concerns may also be raised, since the stored individuals' data is sensitive. When a wearable device operates via synchronizing to a companion object (e.g., smartphones or smart watches), it may offer vulnerable points of entry for attackers to steal stored data (or launch malicious behaviors). For example, *Hello Barbie*, an IoT-oriented commercial product for children, revealed a potential privacy threat whereby an attacker is able to spy on everything in the house via camera and voice-interaction functionalities provided by the product itself and the paired smartphone. Based on this example, we can see that wearable devices give attackers a possible way to capture sensitive data in terms of video and audio formats. It is important that any privacy-aware functionality from wearable devices (or their companion objects) should be accompanied with policy control mechanisms and, in addition, regular checks are strongly suggested to ensure enforcement of those policies. Furthermore, BLE channels between wearable devices and paired smartphones provide possible vulnerability attackers can exploit. Since most wearables are without input monitors (or devices), complicated security protection schemes, such as crypto-based authentication or misbehavior analysis, are difficult to implement on them. However, because wearable devices always involve sensitive individual data, from fitness records to health statuses, the lack of a strong security mechanism will result in potential security and privacy threats.

In the history of security, crypto-based authentication has been thoroughly investi-

gated to support the security robustness of various kinds of communication networks. In recent years, the Internet of Things, regarded as the next generation of network paradigms, has attracted a heightened level of research interest. Based on our observations, IoT-based crypto-based authentication can be categorized into three classes, i.e. for specific purposes, for underlying system architecture, and for communication technique adoption. In the first category, authentication schemes are proposed to support specific system requirements or practical scenarios. Authentication schemes compatible with mobile healthcare systems in hospitals are one such example. In this kind of scheme, the practicability and feasibility of meeting the requirements of specific scenarios are the main focus during the design of authentication protocols. Next, the second category demonstrates authentication schemes designed for specific system architecture (e.g., client-server-server based or client-server based architecture). Examples of this kind are studies which are designed for wireless sensors networks regarded as client-server-server based system architecture. The third category includes authentication schemes implemented by specific IoT-oriented communication techniques, such as RFID, BLE or MQTT, where the characteristics of adopted communication techniques are seamlessly integrated into the authentication schemes.

In another thread of investigation, recent years have seen the research community studying continuous authentication (also called active authentication) due to the inability of crypto-based authentications to fulfill the requirement of real-time and continuous entity verification without user intervention. Every crypto-based authentication needs a user's secrets, such as a password or smart card, as authentication tokens to support user identification and verification. Hence, continuous authentication has emerged to support continuous verifications of logged-in users and eliminate the gap between login-based authentication and full-processes authentication. Versatile bio-authentication tokens have been introduced for continuous authentication. Nevertheless, most of these tokens require specialized and complicated instruments, such as bio-feature extractors, with the result that previously developed continuous authentication schemes have not been suitable for end users. Thanks to the wearable technique, however, the challenge has been overcome. Since the wearables industry currently focuses primarily on fitness and wellness application development, bio-data retrieval has evolved to become one of the basic functionalities

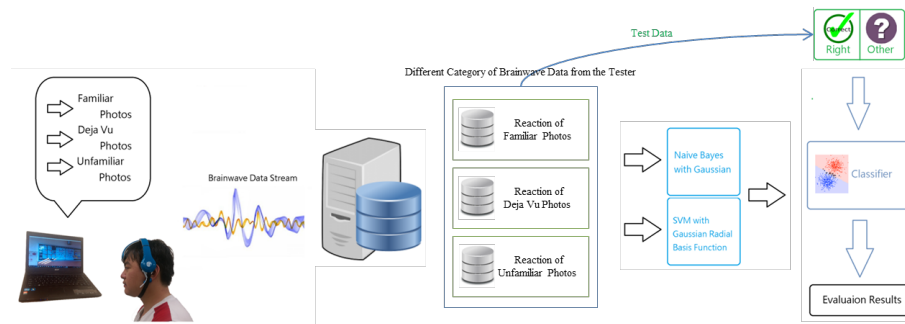


Figure 2.3: Schematic Flow of Proposed Authentication Scheme.

of virtually every wearable device. As a result, continuous authentication can formally come into play at the end user side since it's possible for bio-authentication tokens to be made available via the wearable devices.

2.3 Continuous Authentication with Brainwave Scheme

The architecture of continuous authentication with brainwave scheme is shown in Figure 2.3. Raw brainwave data is retrieved using the commercial brainwave headset, BR8 PLUS, shown in Figure 2.4, in which eight sensors are embedded and exploited to sense the brainwaves of the test-taker. In Figure 2.4 and Figure 2.5, we see channels Ch1, Ch2 and Ch3, located on the frontal lobe, which are used to retrieve the status of brainwaves in terms of working memory, mental workload, meditation attention, emotion, fatigue and appreciation, while channels Ch4 and Ch5 are located on the central sulcus and are used to collect data about motor control and human learning efficiency. Moreover, channel Ch6 will be exploited when it's desired to test long-term memory, and channels Ch7 and Ch8 are for visual control. Raw data from a human subject is separated into a training set and a testing set for the establishment and evaluation of classifiers. The support vector machine (SVM) with Gaussian radial based function (GRBF) and Na?ve Bayesian (NB) were used in the extraction of bio-features. Classifiers based on bio-features are conducted as the authentication tokens in the proposed continuous authentication system. Newly arriving brainwave data (testing data) is then examined via the classifier for entity identification and verification.

As we see in Figure 2.5, in a human-brain, multiple types of functional perceptions will be triggered by different stimuli, which are usually based on visual or auditory events

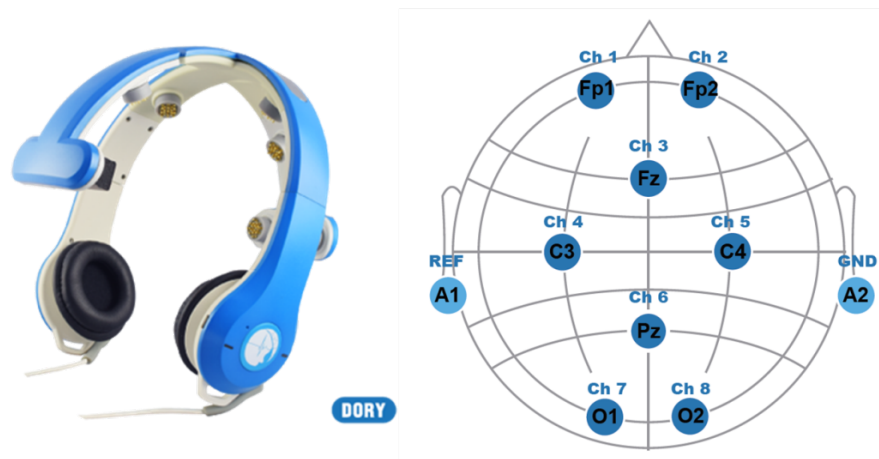


Figure 2.4: BR8 PLUS Headset Used to Monitor Brainwaves During Experiments.

because of their universally common use in human perception testing. To probe a possible new way of developing brainwave-based authentication, we adopt the user's perception of his/her long-term memory to construct the bio-features as the authentication token in our system. Long-term memory is typically used to support straight-face tests, in which a tester is examined through a series of events, presented in the form of pictures. For example, in a murder testing, three categories of pictures, i.e. definitely-unknown pictures, definitely-known pictures and the murder-related pictures, are shown to a suspect in a straight-face test. Only an innocent person can pass the straight-face test since it is extremely difficult for a person to control his or her brainwaves using will alone. To investigate the feasibility of adopting new brainwave features for the purpose of authentication, we would like to retrieve the user perceptions associated with stimuli on the parietal lobe and use this to represent the user's perception when his/her long-term memory is stimulated. Essentially, the proposed system retrieves human reactions stimulated by familiar photos, and unfamiliar photos, and unfamiliar photos, in the proposed authentication system.

2.4 Implementation and Evaluation

We describe the experiment used to retrieve user perceptions from the parietal lobe. We constructed three classes of pictures: user-familiar, Deja vu and user-unfamiliar (See Figure 2.6 for examples). The pictures were randomly chosen and displayed on the screen

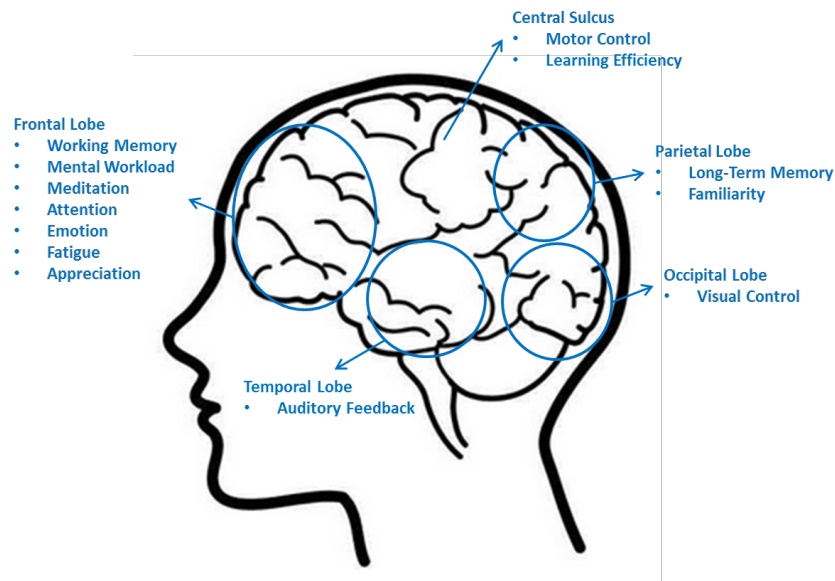


Figure 2.5: Functionality of Each Part of The Brain.

during the experiment. The pictures numbered 67 and the time of duration for each displayed picture was one-and-one-half seconds. In addition, one second was inserted between two pictures displayed on the screen. After collecting the data stimulated by the testing photos, we transformed the brainwave raw data into the alpha, beta, gamma, delta and theta signals via Fourier Transform Technique. The outputted results, i.e. alpha, beta, gamma, delta and theta values, are then used as the input sequence of our proposed authentication system. With SVM-GRBF and NB algorithms, classifiers are conducted as authentication tokens in our system. An entity identification and authentication process is then performed continuously based on these authentication tokens (i.e. classifiers). The performance of our authentication system is evaluated (see Table 2.2) and we see that a maximum 95.29% authentication accuracy can be achieved via the classifier conducted by SVM-GRBF. That is, at best 4.71% of FRR is guaranteed under the input data, i.e. alpha, beta, gamma, delta and theta.

Despite satisfactory authentication accuracy (i.e., FRR), the proposed system still underperformed with regard to FAR. To pursue better performance efficiency, we are thus inspired to develop a data purification process to eliminate noise which is useless for the classifier construction. During brainwave retrieval, we observed similarities in the composition of input data sequences from different users. For example, parts of the input data sequences from two distinct users may be similar and the similar parts of data sequence



Figure 2.6: Test Photos Used in Experiments.

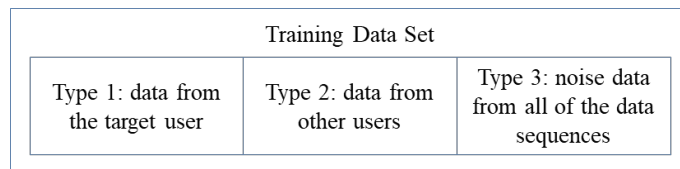


Figure 2.7: Training Data Set.

will be accepted by both of these two distinct users' classifiers during the entity verification. This phenomenon diminishes the system performance in terms of FAR. In other words, similar portions of a data sequence may be incorporated within the classifiers of different users during entity verification, thereby hindering performance of FAR. Hence, we devised a data purification process aimed at eliminating environmental interference and other forms of noise that could interfere with the construction of classifiers. These efforts were based on the assumption that much of the noise common among users is misidentified as specific to a particular user. Removal of this noise should make it easier to extract uniquely distinguishable features for the formulation of classifiers. In our experiment, a data purification procedure is constructed to possibly eliminate the target user's input brainwave data that is similar to other users' during the classifier establishment stage. The noise is regarded as the target user's brainwave data that is similar to other users'. The data purification phase begins with the collection of noise data from all data sequences (i.e., brainwaves) derived by all users to be separated into three categories as a new training set (Figure 2.7). Type I refers to data sequences from the target user that are highly distinct from those of other users. Type II refers to data sequences from non-target users. Type III refers to noise data common to the outputs of all users. As

mentioned above, the noise will have similar composition, and thus we collect the noise data from all of the user's input data sequences at the first stage. After the noise is identified, the Type I and II data sets will then be identified. Note that the identification and removal of noisy data (Type III) greatly facilitates the identification of Type I and Type II data sets. The next step involves the establishment of classifiers based on the three data set categories. In the process, the input noise data will engage with noise from the data sequences input by the target user and other users. It is believed that the constructed classifier will thus be more distinguishable after the noise has been removed.

Table 2.2: Authentication Accuracy With Alpha, Beta, Gamma, Delta, Theta Signals.

Tester	SVM	NB
YZW	98.60%	86.30%
KGT	95.60%	21.55%
WSF	91.50%	90.70%
HYC	98.05%	75.65%
LILY	86.14%	50.83%
LSY	97.85 %	97.30%
YIJ	99.55%	97.85%
YUN	95.05 %	70.80%
Average	95.29%	73.87%

Table 2.3 and Table 2.4 lists the authentication accuracy results of the proposed scheme with data purification respectively performed using the SVM-GRBF and NB techniques. The NB algorithm performed poorly in classifier construction, resulting in authentication

Table 2.3: Authentication Accuracy With Data Purification and NB.

		TESTING DATA							
		YZW	KGT	WSF	HYC	LILY	LSY	YIJ	YUN
CLASSIFIER	YZW	67.3	41.5	39.7	31.85	19.4	29.5	48.55	29.3
	KGT	2.15	15.45	5.7	18.45	0.4	5.65	35.15	31.75
	WSF	47.52	21.21	77.29	16.16	12.76	11.76	28.26	14.46
	HYC	29.6	18.65	9.45	43.15	1.4	45.45	57.8	47.65
	LILY	66.03	56.3	22.76	50.43	50.73	28.11	33.82	27.36
	LSY	19.01	4.4	3.9	25.31	0.05	9.73	34.92	25.01
	YIJ	44.72	18.31	15.21	39.62	1.55	49.32	57.33	39.22
	YUN	26	48.55	8.75	46.7	9.25	22.1	44.4	52.4

Table 2.4: Authentication Accuracy With Data Purification and SVM-GRBF.

		TESTING DATA							
		YZW	KGT	WSF	HYC	LILY	LSY	YIJ	YUN
CLASSIFIER	YZW	72.6	40.6	38.45	44.85	29.05	32.9	49.1	44.6
	KGT	4.5	49.5	4.85	33.95	1.2	18.7	32.15	27.3
	WSF	56.18	41.02	83.94	23.71	10.01	8.25	24.46	23.96
	HYC	33.2	16.45	15.1	69.85	9.45	55.05	55.7	61.2
	LILY	61.69	6.20	20.76	40.02	81.04	25.61	32.12	38.92
	LSY	17.71	6.30	7.10	60.33	3.052	62.48	33.32	48.68
	YIJ	14.56	16.91	19.41	64.08	14.56	51.63	59.83	60.83
	YUN	13.76	29.85	10.35	56.6	32	32.7	48.35	79.3

accuracy far below the results in the previous experiment (Table 2.2). Nevertheless, a useful result emerges, in that we find that the authentication accuracies of some users, i.e. WSF, LILY, LSY and YUN, are comparatively better than the cases of inputting their testing data into the system with other users' classifiers. For the example of LILY, when we input LILY's testing data, we achieve 50.73% authentication accuracy when using LILY's classifier. However, inputting LILY's testing data with YZW's classifier reduces authentication accuracy to just 19.4%. Obviously, the authentication accuracy of LILY is absolutely not good (50.73%), but it is comparatively better than other cases, 19.4%, 0.4%, 12.76%, 1.4%, 0.05%, 1.55% and 9.25%, in which LILY's data is tested with the classifiers from other users. It is obvious that the property of comparatively-better emerged in our experiment results can be applied in our authentication system to pursue better system performance in terms of FAR. That is, if we have an authentication accuracy rate which is higher than all of the other cases involving the target user's testing data and the other users' classifiers, the system will identify that the currently examined user is not an imposter. The example of LILY in Table 2.3 is one such successful case.

From each column of Table 2.4, fortunately, we see that the property of comparatively-better emerges in all of the cases of user verification. All of the authentication accuracies are, comparatively, the highest. We then integrate this property with our proposed continuous authentication system. As shown in Figure 2.8, the proposed system was improved by implementing a two-stage design to deal with FRR and FAR, respectively. The first stage uses one dataset from the target user and one dataset from the other users as training

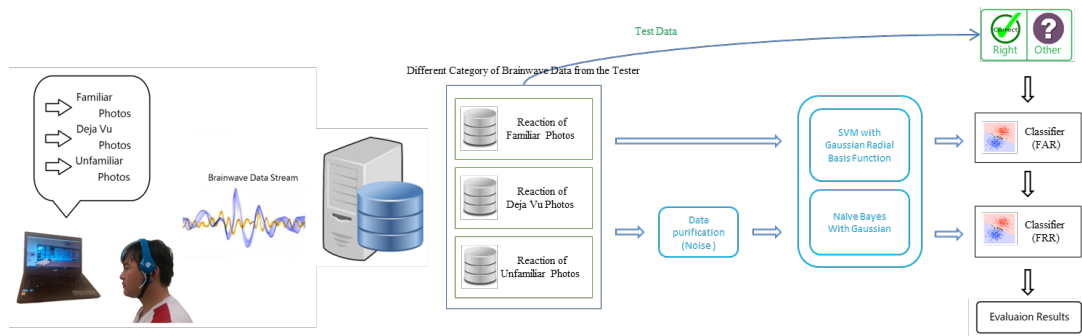


Figure 2.8: Diagram of Improved System.

Table 2.5: Authentication Accuracy With Data Filtration and NB.

		TESTING DATA						
		YZW	KGT	WSF	HYC	LILY	LSY	YIJ
CLASSIFIER	YZW	75.1	3.9	44.1	7.5	35.9	3.9	58.2
	KGT	11.3	99	15.1	50.6	0	72.4	15.4
	WSF	73.9	21.6	92	33.6	0.2	21.6	39.4
	HYC	15.3	27.8	5.6	95.8	7.5	42.3	88.1
	LILY	7.2	0.6	2.6	3.6	94.2	2.7	11.8
	LSY	22.3	48.5	12.4	50.1	8.5	81.3	33.32
	YIJ	37.96	9.9	23.5	57.8	22	17.4	94.8

data for the formulation of classifiers. An appropriate threshold value (e.g., 85% or 90%) enables user identification of high accuracy. In the second stage, we implement the examination of the property of comparatively-better into a user verification procedure. That is, if the authentication accuracy is the comparatively highest one, the examination on the second stage is passed. The improved SVM-GRBF system in Figure 2.8 achieved perfect authentication accuracy, i.e. 100%, in our experiments.

To explore more valuable information for improving the system performance, we further present a data filtration scheme to enhance the distinguish ability of the constructed classifier as much as we possibly can. The idea of the data filtration scheme is also to remove the non-distinguishable data common to the target user and the other users. In the proposed scheme, we use the originally constructed classifiers to identify and eliminate the non-distinguishable data for each pair of the target user and the other users. For instance, assuming there are four system users, i.e. target user A, user B, user C and user D, we first adopt the users A and B's classifiers to identify the data which belong to the

Table 2.6: Authentication Accuracy With Data Filtration and SVM-GRBF.

		TESTING DATA						
		YZW	KGT	WSF	HYC	LILY	LSY	YIJ
CLASSIFIER	YZW	83.4	2.5	28.8	4.6	33.8	1.7	46.9
	KGT	6.2	98.6	4	17.1	0	91.8	15.4
	WSF	47.9	19.4	86.5	28.7	0.7	25.6	28.3
	HYC	5.9	24.7	14.6	91.6	2.3	66.2	43.8
	LILY	14.4	0.3	6.5	2.4	91.8	2.5	8.1
	LSY	7.6	58.3	11	41.1	8.4	84.5	27.4
	YIJ	45.1	14.3	21.2	39.7	11.1	26.2	93.9

users A and B simultaneously. After the non-distinguishable data is identified, this data must be removed due to its non-distinguishability. The above process is a stage of the data filtration process for the user pair of A and B. Next, two more stages of the data filtration process are performed for the user pairs of “A and C” and “A and D”, respectively. Once all of the stages are finished, it is believed that the remaining data of the target user A is more distinguishable than before. The classifier establishment will then be invoked to construct a new classifier of the target user A for the purpose of authentication. Tables 2.5 and 2.6 present the experiment results for our proposed system (SVM-GRBF and NB) with our data filtration scheme. The authentication accuracies are better than the cases without data purification (e.g., Tables 2.3 and 2.4). We thus can conclude that the proposed data filtration scheme is useful for enhancing the authentication accuracy of the proposed system.

2.5 Chapter Conclusion

In this chapter, we envision a ubiquitous network consisting of wearable devices and IoT-based sensors. In the quest for the support of wearable equipment with brainwave retrieval functionality, we developed a continuous authentication system using brainwaves as bio-features for IoT-based networks. System performance was further enhanced by incorporating a data purification process and a data filtration scheme to facilitate the differentiation of classifiers. Our data purification process and data filtration scheme remove undifferentiated data common to the target user as well as other users. Experiment results

demonstrate the efficacy of the proposed authentication system in verifying the identity of device users with a high degree of authentication accuracy.

Chapter 3

Implementation of NIST P-256 and SM2 ECC on 8-bit AVR Processors

In this chapter, we proposed an efficient, secure and compact implementation of scalar multiplication on a 256-bit elliptic curve recommended by the SM2, which is a set of public key cryptographic algorithms based on elliptic curves published by Chinese Commercial Cryptography Administration Office and standardized at ISO in 2017. We also conducted a comparison implementation of scalar multiplication on the same bit-length elliptic curve recommended by NIST. We re-design some existent techniques to fit the low-end IoT platform, namely 8-bit AVR processors, and our implementations evaluated on the desired platform show that the SM2 algorithms have competitive efficiency and security with NIST, which would work well to secure the IoT world.

3.1 ECC Implementation in IoT Devices Background

With the advent of the Internet of Things (a.k.a IoT), a great deal of constrained devices (such as 8-bit microcontrollers) are now widely used in cyber-space for pervasive applications like wireless sensor networks (a.k.a. WSN) or RFID tags. Compared with traditional cable networks, security in these sensor networks has been an ever-increasing challenge mainly due to their widespread acceptance and wireless nature. The restriction of computation capability, storage space and even energy consumption for such constrained devices also increases the difficulty in deploying cryptographic schemes. But

even so, a lot of cryptographic techniques have been investigated to ensure the security of such devices, of which the elliptic curve cryptography (a.k.a ECC) is paid much attention owing to its lightness and security [34–42]. The ECC now is included in current standards like those of ISO, ANSI, IETF, BSI as well as NIST, and has become a common approach to instantiate asymmetric encryption and signature schemes.

Recently, the SM2 [43], a set of public key cryptographic algorithms based on elliptic curves released by Chinese Commercial Cryptography Administration Office in December 2010, was standardized at ISO in 2017. It is the first standard of the digital signature algorithm which has been used in electronic authentication service system in China. The SM2 is published by Standardization Administration of the People’s Republic of China (No. GB/T 32918. 2016) as 5 parts, covering: General, Digital Signature Algorithm, Key Exchange, Public Key Encryption Algorithm, and Parameter Definition [48]. Specifically, it is composed of three distinct algorithms: an elliptical curve digital signature algorithm (“SM2DSA”), a key exchange protocol (“SM2KEP”), and a public key encryption algorithm (“SM2PKE”).

The SM2 has won the eyes of fields from both academy and industry, based on which many implementations and applications have come to being [44]. However, few research works on implementing SM2 for constrained devices have been conducted. To bridge the gap between theory and practice, in this chapter we implemented the scalar multiplications of both SM2 and NIST recommended 256-bit elliptic curves on low-end 8-bit AVR processors, where optimized finite field arithmetic and elliptic curve group arithmetic are adopted for the highest performance. Unfortunately, there haven’t any work of implementing the 256-bit SM2 curve on 8-bit AVR processors, which achieved the new speed-record on the desired platform.

3.1.1 Elliptic Curve Cryptography

The elliptic curve cryptography (ECC), a public key cryptography instantiation based on the algebraic structure of elliptic curves over finite fields [49], has been extensively exploited to secure cyber-physical systems. Since its birth in 1985, ECC has drawn much attention from many cryptography practitioners mainly due to its smaller key size, reducing storage, and transmission requirements. Compared with an RSA-based system with

a large modulus (e.g., a 3072-bit RSA public key), the group of an elliptic curve could provide the same level of security with a much smaller modulus (e.g., a 256-bit elliptic curve public key).

Let \mathbb{F}_p be a finite field with the characteristic $p > 3$. The elliptic curve with short Weierstrass form is described as

$$E/\mathbb{F}_p : y^2 = x^3 + ax + b,$$

where $a, b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \neq 0$. Current standards like those of ISO, SM2 and NIST describe elliptic curves in short Weierstrass form, since such curve model has been extensively studied.

The main operation in ECC is scalar multiplication. Techniques for accelerating the computation of such operation have been proposed, one of which is adopting special curve model with faster elliptic curve group arithmetic. Specific elliptic curves like Montgomery curves [50] and twisted Edwards curves [51]) have been taken into consideration as curve models for the next NIST ECC generation standards (e.g. Curve25519 [52]).

The Montgomery curve model of elliptic curve was first proposed by Montgomery [50], and now has been widely adopted for evaluating efficient and regular scalar multiplication. The Montgomery curve can be given by the following equation

$$E_{M,A,B}/\mathbb{F}_p : By^2 = x^3 + Ax^2 + x,$$

where $A, B \in \mathbb{F}_p$ and $B(A^2 - 4) \neq 0$. The group arithmetic on $E_{M,A,B}$ relies only on x-coordinate and can be implemented in a fashion named as ‘‘Montgomery ladder’’, which is regular and thus side-channel attack resistant.

Another model which also provides more efficient group arithmetic is called Twisted Edwards model, which was introduced by Bernstein et al. [51] in 2008. Twisted Edwards curve can be given by

$$E_{T,a,d}/\mathbb{F}_p : ax^2 + y^2 = 1 + dx^2y^2,$$

where $a, d \in \mathbb{F}_p, ad(a - d) \neq 0$. It is well known that a twisted Edwards curve $E_{T,a,d}$

is birationally equivalent to some elliptic curve in Montgomery $E_{M,A,B}$ with $A = 2(a + d)/(a - d), B = 4/(a - d)$.

Though Montgomery curves or twisted Edwards curves own several advantages and have been fully considered for implementing elliptic curve cryptosystems, the main disadvantage is that they concentrated on the case $\#E(\mathbb{F}_p) \equiv 0 \pmod{4}$. Note that most standard elliptic curves (i.e., NIST curves and SM2 curves) have prime group orders and only admit Weierstrass model, therefore, efficient and secure implementation of ECC based on Weierstrass curves should be further exploited.

3.1.2 NIST Versus SM2

NIST P-256. The NIST in the FIPS 186-2 standard [53] recommended 15 elliptic curves of varying security levels for U.S. federal government use. Among them the curve P-256 is given by $E/\mathbb{F}_p : y^2 = x^3 + ax + b$ with prime group order n , while the corresponding parameters are stated as

```

p := FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFF,
a := FFFFFFFF 00000001 00000000 00000000 00000000 FFFFFFFF FFFFFFFF FFFFFFFFC,
b := 5AC635D8 AA3A93E7 B3EBBD55 769886BC 651D06B0 CC53B0F6 3BCE3C3E 27D2604B,
n := FFFFFFFF 00000000 FFFFFFFF FFFFFFFF BCE6FAAD A7179E84 F3B9CAC2 FC632551.

```

SM2. The SM2 [43] is a set of public key cryptographic algorithms based on elliptic curves. These algorithms and recommended parameters are published by Chinese Commercial Cryptography Administration Office for the use of electronic authentication service system. The SM2 includes generation algorithms to provide desired elliptic curves for ECC applications. Specially, it recommended a 256-bit elliptic curve $E/\mathbb{F}_p : y^2 = x^3 + ax + b$ with prime group order n , of which the parameters are given by

```

p := FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFF,
a := FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 00000000 FFFFFFFF FFFFFFFC,
b := 28E9FA9E 9D9F5E34 4D5A9E4B CF6509A7 F39789F5 15AB8F92 DDBCBD41 4D940E93,
n := FFFFFFFE FFFFFFFF FFFFFFFF FFFFFFFF 7203DF6B 21C6052B 53BBF409 39D54123.

```

Both the NIST and SM2 standards choose elliptic curves for the sake of security and efficiency:

Security Issues. The security issues of ECC usually include ECDLP security, complex-multiplication (a.k.a CM) field discriminant, twist security, regular (or constant time) scalar multiplication and so on. For NIST P256 and SM2 curves, we can conclude that

- Both curves have prime group orders of 256-bit, and the corresponding ECDLPs are at the security level of around 128-bit. (Note that the best general algorithm for solving ECDLP with group size r has computational complexity $O(\sqrt{r})$)
- The CM field discriminants of both curves are greater than 2^{100} , which satisfy the requirement of SafeCurves proposed by Bernstein and Lange [54]. Though there is no clear evidence that small CM discriminants would influence the security of elliptic curves, cryptographers suggested choosing large CM discriminants for elliptic curve generation, in order to avoid potential threats.
- The order of the quadratic twist of NIST P256 curve has a large prime divisor with about 241-bit, while that of SM2 curve provides the largest prime divisor with about 148-bit. This implies that the twist security of NIST P256 curve is much better. Nevertheless, it could not corrupt the security in practice if the x -coordinate only arithmetic is not applied to the ECC implementations.
- Both curves are presented in short Weierstrass form, which means that they could adopt the same regular (or constant time) scalar multiplication algorithms.

Efficiency Issues We also conclude some commonalities shared by both curves in their efficient implementation.

- For efficient finite field arithmetic, both curves adopt some generalized Mersenne prime [55] as the base field characteristic ($2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ for NIST P-256 and $2^{256} - 2^{224} - 2^{96} + 2^{64} - 1$ for SM2), which allow us to execute a fast modular reduction.
- Both curves use the curve shape $y^2 = x^3 - 3x + b$, which could provide very efficient elliptic curve group arithmetic. Note that the corresponding curve in projective coordinates is $Y^2 = X^3 - 3XZ^4 + bZ^6$, which could induce the expression $3X^2 - 3Z^4 = 3(X + Z^2)(X - Z^2)$ in the group arithmetic proposed by [56], thus the cost of doubling operation in Jacobian coordinates is $4M + 4S + 9A$. As claimed in [56], these are the fastest formulae in short Weierstrass model.

3.1.3 8-bit AVR Processors

The AVR is a family of microcontrollers developed by Atmel Corporation for twenty more years, and now has been an excellent choice for embedded systems. The low-end 8-bit AVR processors are popular in Internet of Things (IoT) environments due to their nice properties, such as low-price, high-performance, and low energy consumption. These devices are based on the most code-efficient architecture for C and assembly programming.

There is a rich instruction set for the typical 8-bit AVR core, accompanied with 8-bit 32 general purpose registers [57]. Such 8-bit registers are directly connected to the Arithmetic Logic Unit (ALU), while two independent registers are accessed in one single instruction which can be executed in one or two clock cycles¹. The AVR family instruction set consists of 133 powerful instructions, including of arithmetic and logic instructions (e.g. addition without carry ADD, addition with carry ADC), branch instructions (e.g. relative jump RJMP), data transfer instructions (e.g. copy register MOV), bit shift instructions (e.g. logical shift left LSL) and Microprogrammed Control Unit (MCU) Control Instructions (e.g. no operation NOP), whereby most of them can be executed within one single clock cycle. In order to access to the target memory address, two adjacent registers (e.g. (R27, R26), (R29, R28), (R31, R30)) are used, which requires two clock cycles. Though there are several families of AVR microcontrollers different in peripheral

¹8-bit wise multiplication MUL requires 2 clock cycles.

units and memory sizes, they still share the same basic instruction sets.

The AVR pipeline consists of two stages, one for fetching the instruction into memory and the other stage is for execution process. The AVR processor supplies three kinds of memory—data memory, program memory and internal memory. The data memory is an internal SRAM which is expandable with external memory, the program memory is the one in system programmable flash memory, while the internal memory is called EEPROM which is not directly addressable but offers an interface via special function registers requiring 3 clock cycles.

Our implementation employed the ATmega128 as the targeted platform, which is a derivative of the AVR family based on a modern highly structured RISC design. As its name implies, the ATmega128 equips 128 kB of programmable flash memory, as well as 4 kB SRAM, 4 kB EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. Furthermore, there are many development and compile tools available for the Atmel AVR family. The Atmel corporation offers an free AVR Studio development environment, which includes an AVR compiler, an assembler, and a graphical simulator under visual studio environments.

3.1.4 Previous Implementations on 8-bit AVR Processors

Due to the advantage in its security and efficiency, ECC has been implemented on the 8-bit AVR platform by plenty of cryptography practitioners. As far as we know, the first ECC software implementation on 8-bit processors was proposed by Woodbury et al. in 2000 [34]. Their work is based on the 8051-compatible micro-controller, which was developed by Intel in the 1980s and remain for use in embedded systems. Their implementation relied on the arithmetic of finite field $\mathbb{F}_{(2^8-17)^{17}}$, and the timings for fixed (or random) base scalar multiplication on desired platform is $23.4 \cdot 10^6$ (or $100 \cdot 10^6$, respectively) clock cycles. Gura et al. in CHES 2004 [35] reported the first highly efficient ECC software based on 8-bit AVR processor (ATmega128), where they exploited the large register file of the processor to process four bytes of the operands for each iteration of the inner loop in the multiple-precision multiplication. Compared with a conventional byte-wise multiplication, such optimization (a.k.a hybrid method) significantly reduces the number of load/store instructions. Therefore, it costs only $6.48 \cdot 10^6$ clock cycles for

evaluating a full scalar multiplication over a 160-bit finite field.

Afterward, a lot of research literatures have been devoted to accelerate the ECC implementation on such processors, and most of them focused on improving the so-called hybrid method for multiplication or exploring some variants which would be more efficient [36,58–60]. The most notable progress is reported in CHES 2011, where Hutter and Wenger introduced the operand-caching method [61]. Such method caches the operands in the general purpose registers to reduce the number of memory access. Seo et al. also proposed an advanced consecutive operand caching method in WISA 2012 [62]. In 2015, Hutter and Schwabe in [63] carefully revisited the previous work and made further improvement for the implementation of Karatsuba multiplication on ATmega128 processors, which achieved record-setting performance on such platform.

However, the above lightweight ECC implementations did not take the security into consideration, and thus were basically vulnerable to side-channel attacks, in particular *Simple Power Analysis (SPA)* [64]. Several papers have shown that SPA attacks on unprotected (or insufficiently protected) ECC implementations on embedded devices would pose real-world security threats [65, 66]. The SPA attacks usually exploit conditional statements, as they may induce key-related information leakage. When it comes to ECC implementation, the conditional statements come from conditional subtractions in finite field arithmetic [67–69]), or irregularities in the execution pattern of scalar multiplication algorithms. Therefore, secure finite field arithmetic and regular group arithmetic are both needed.

3.2 Efficient Implementation of NIST P-256 and SM2

3.2.1 Finite Field Arithmetic

The finite field we adopted here is the prime field which is denoted as \mathbb{F}_p whereas p is a prime number. The finite field \mathbb{F}_p consists of a finite integer set $\{0, 1, \dots, p-1\}$, two operations performed modulo p , modular addition (+) and modular multiplication (\cdot). The finite integer set \mathbb{F} together with the modular addition (+) operation forms an abelian group with identity 0, and the $\mathbb{F} \setminus \{0\}$ together with the modular multiplication (\cdot) operation also forms an abelian group with identity 1. The distributive law holds for the

integer set \mathbb{F} and two operations: $(A + B) \cdot C = A \cdot C + B \cdot C$ for all $A, B, C \in \mathbb{F}$. By using two basic operations above, we can derive modular subtraction and modular division as $A - B = A + (-B)$ and $A/B = A \cdot B^{-1}$ respectively whereas $-B$ is a unique element in \mathbb{F} satisfying $B + (-B) = 0$ and B^{-1} is a unique element in \mathbb{F} satisfying $B \cdot B^{-1} = 1$.

Generalized Mersenne primes were first introduced by Solinas in 1999 [55], and have been widely used as finite field characteristics in ECC for faster modular reduction. The generalized form of such primes is $p = 2^k - c_1 2^{k-1} - \dots - c_i 2^{k-i} - \dots - c_k$, where all c_i s satisfy $c_i \in \{1, -1, 0\}$.

The NIST recommended five of these primes for use in ECC cryptosystems, one of which is the P-256 ($P_{256} = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$). The SM2 also adopts a similar prime P_{sm2} as $2^{256} - 2^{224} - 2^{96} + 2^{64} - 1$. The reduction operation with a generalized Mersenne prime can also be done by using some congruence relation, which is similar to that of real Mersenne primes. Both NIST and SM2 adopt such generalized Mersenne primes as the characteristics for the desired finite fields. This allows us to execute a fast reduction.

In the following we let M, S, I and A denote the multiplication, squaring, inversion and addition over finite field \mathbb{F}_p , respectively. In our test the main field arithmetic functions are implemented in an unrolled way, which can help us to further reduce the computation cost for the loop controlling. For example, when we do the addition operation one CP instruction (one clock cycle) is required for controlling the number of iterations, and one RJMP instruction (two clock cycles) for jumping to the next iteration. In this case, the cost for controlling loop of each iteration requires 3 clock cycles. Assuming a 256-bit wise operand addition/subtraction case, the operand is loaded 8-bit by 8-bit from memory into registers. Thus, at least 96 (32×3) clock cycles for controlling the loop have to be paid if the addition is implemented in a 8-bit wise looped way. As we see, unrolled approach requires larger memory than looped way, but this ensures higher performance. Similarly, we implement the other field operations in the same way.

Constant-time Addition and Subtraction. Finite field addition or subtraction operation requires the final subtraction or addition with target modulus to fit the results into the target field, respectively. If we perform the conditional final subtraction or addition statement, the execution timing becomes varied depending on the program routines. Since the

program routines have high correlations with secret values, the attacker may get the secret information from conditional execution of final subtraction for reduction [70].

Algorithm 1 Previous Constant-time Reduction Modular Addition

Require: Two s -word operand $A = (a_{s-1}, \dots, a_1, a_0)$ and $B = (b_{s-1}, \dots, b_1, b_0)$ in the range of $[0, sw - 1]$ and s -word modulus $p = (p_{s-1}, \dots, p_1, p_0)$;

Ensure: Incomplete reduction result $C = (A + B) \bmod 2^{sw}$;

```

1:  $(\epsilon, c_0) \leftarrow a_0 + b_0$ 
2: for  $i$  from 1 to  $s - 1$  by 1 do
3:    $(\epsilon, c_i) \leftarrow a_i + b_i + \epsilon$ 
4: end for
5:  $c_s = \epsilon$ 
6:  $mask \leftarrow -c_s \bmod 2^w$  ( $w$  is the length of the word)
7:  $(\epsilon, c_0) \leftarrow c_0 - (p_0 \& mask)$ 
8: for  $i$  from 1 to  $s - 1$  by 1 do
9:    $(\epsilon, c_i) \leftarrow c_i - (p_i \& mask) - \epsilon$ 
10: end for
11:  $c_s = 0$ 
12: return  $C = (c_{s-1}, \dots, c_1, c_0)$ 

```

To avoid the conditional statements, constant-time reduction method for modular addition was suggested in Algorithm 1 [40]. The method used the conditional reduction (i.e. a multi-precision subtraction) of field arithmetic with the mask. After executing the first part of modular addition (i.e. $A + B$), it first generates the 2's complement of c_s , and then follows the value ($mask$). When the carry bit ϵ in Step 3 is 1, the $mask$ is set to 0xFF. Otherwise, the value is set to 0x00. Afterward, the masked modulo is subtracted without the comparison, as it performs $(p_i \& mask)$. That is to say, when the case is $c_s = 1$, it can do the reduction with the actual modulus p , while for $c_s = 0$ it performs no operation since the mask is set to zero. This constant-time reduction makes the execution time independent from the operands, which can reduce side-channel leakage. Moreover, this method doesn't require to perform a comparison between C with p , which significantly saves the execution time coming from memory access and comparison arithmetic operations.

The modular subtraction is implemented in a similar way as the described modular addition operation. The only difference is that the modular subtraction performs constant-time final addition with modulus p instead of final subtraction for modular addition.

Based on previous approach, we further reduce the mask operation in Steps 7 and 9 of Algorithm 1 by using the features of target curves. From the hexadecimal representations

of P_{256} and P_{sm2} in Section 2.1, we can see that the modulo P_{256} consists of only three 8-bit wise patterns, including 0xFF, 0x00, 0x01, while those for P_{sm2} are 0xFF, 0xFE, 0x00. Since we do not need to perform the mask operation with 0x00 pattern, the number of 8-bit wise mask operation could be reduced.

Multiplication and Squaring. In this part, we first review the previous standard algorithms for multiplication and squaring and thereafter, we introduce our implementation of multi-precision multiplication and squaring on 8-bit AVR processors. The following techniques aims at computing $C = A \cdot B$ whereas $A = (a_{s-1}, \dots, a_1, a_0)$, $B = (b_{s-1}, \dots, b_1, b_0)$, and $C = (c_{2s-1}, \dots, c_1, c_0)$.

Operand Scanning Method. This is a basic and elementary method to implement multi-precision multiplication and squaring. Operand scanning method consists of an inner loop and a outer loop. The outer loops iterates every word of the multiplier A , denoted as a_i , and the inner loop traverses every word of the multiplier B , denoted as b_j , and multiplies each word b_j by a_i . Each inner loop iteration produces a 2-word product which will be added to the intermediate result (u, v) . This Multiply-Accumulate (MAC) process performs operation in the form of $(u, v) \leftarrow a_i \cdot b_j + c_{i+j} + d$ whereby (u, v) is a 2-word intermediate result and a_i, b_j, c_{i+j}, d are single-precision words. After the MAC process, v will be set as the $(i + j)$ -th word of the product C and u will participate into the next inner loop iteration. According to [71], in order to compute a multiplication of two s -word elements using operand scanning method, it requires s^2 **mul**, $4s^2$ **add** (or **adc**), $2s^2 + s$ **ld** (load), and $s^2 + s$ **st** (store) instructions.

Product Scanning Method. Another elementary method to implement multi-precision multiplication and squaring is the product scanning method. Similar with operand scanning method, product scanning method also deploys two loops of which the outer loop traverses through $2s$ words of the product and the inner loop generates one word of the product. As for $C = A \cdot B$, the i -th word of C is computed by the MAC process $(t, u, v) \leftarrow (t, u, v) + a_j \cdot b_{i-j}$ whereby (t, u, v) is a 3-word intermediate result. After the MAC process, v will be set as the i -th word of the product C and t and u will participate into the next inner loop iteration. According to [71], in order to compute a multiplication of two s -word elements using product scanning method, it requires s^2 **mul**, $3s^2$ **add** (or **adc**), $2s^2$ **ld**, and $2s$ **st** instructions. The operand scanning method is comparatively simple

to implement in a high-level programming language, but when using the assembly language, it is less efficient than the product scanning method since it requires more memory access (**ld**, **st**) and addition (**add**) instructions.

Hybrid Method. The hybrid method, proposed in [35], aims to combine the operand scanning method and product scanning method together to optimize the number of registers and the number of memory accesses. It reduces the number of **ld** instructions using several extra registers to store the operands from memory. And due to the fact that these operands are multiplied in several multiplications, this technique leads to efficiency improvement since these operand are only loaded once. The technique in hybrid method is to employ the operand scanning method as the inner algorithm and adopt the product scanning method as the outer algorithm. In the inner algorithm, the hybrid method loads $d \leq 2$ words of the multiplicand from memory for just once and then multiplies it to d words of the multiplier as the operand scanning method does. Thereafter, the inner algorithm saves the cumulative sum in $2d + 1$ registers. The number of the **ld** instructions in this process is reduced to $2\lceil s^2/d \rceil$. However, this speed up improvement has to increase the number of **add** or **mov** instructions.

Operand Caching Method. The operand caching method also aims to reduce the number of **ld** instructions but it exploits the sophisticated cache instead of using extra registers in the hybrid method. It is stated that the number of **ld** instructions is reduced to $2\lceil s^2/e \rceil$ ($e > d$ and e denotes the row size) and this technique achieves a performance improvement by a factor of 10% compared to the hybrid method in [60]. The operand caching method adopts the execution flow of the product scanning method but separates the calculation into e rows. This method assumes that the registers file cannot hold all the operands in multiplication and in this case, the product scanning method requires at least $2s^2$ **ld** instructions. And this situation makes it possible to utilize the cache to minimize the **ld** instructions. This goal is achieved by reordering the execution of the row sections and reusing the operands that have been loaded before to compute the next multiplication. According to [60], the overall memory access operations (**ld** and **st** instructions) number is $3s^2/e + s$.

Reverse Product Scanning (RPS). The RPS technique, proposed in [72], is a similar technique like the hybrid method but it optimizes the hybrid method by utilizing the byte-

products to catch carries and minimize the overuse of extra "carry-catcher" registers. This is achieved by computing the byte-product from the most-significant word to the least-significant word of the product in the inner algorithm. As shown in 3.1, the RPS technique reduces the number of the "carry-catcher" registers from 6 down to 2 and minimizes the number of **adc** instructions. The left figure of 3.1 is taken from [60] and the right one is taken from [72]. As claimed in [72], the implementation of RPS multiplication and RPS squaring not only is faster than the operand caching method [61], but also has smaller code size.

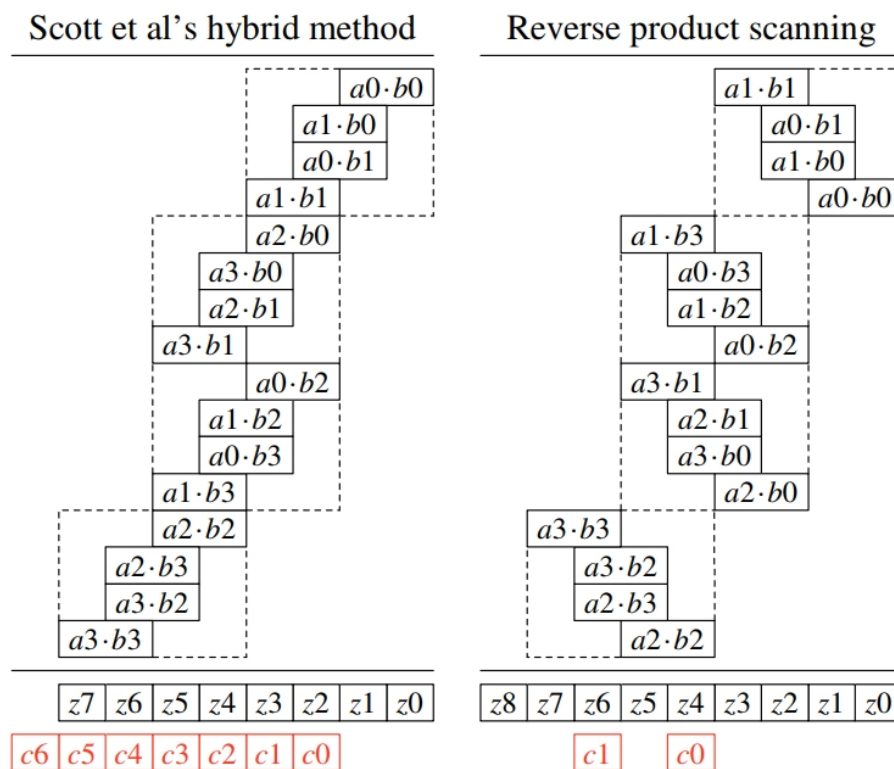


Figure 3.1: The execution flow of computing 4-word multiplication using the Hybrid Method and Reverse Product Scanning Method respectively

Karatsuba Algorithm. Karatsuba algorithm adopts the divide-and-conquer method to reduce the complexity to sub-quadratic computational complexity. As for the multiplication of two s -word integers, Karatsuba algorithm performs as follow. Let $k = \lceil s/2 \rceil$ and divide A, B into two parts A_h, A_l, B_h, B_l where each part contains k word of A, B respec-

tively and $A = A_h 2^{8k} + A_l, B = B_h 2^{8k} + B_l$. Then

$$\begin{aligned} A \cdot B &= (A_h 2^{8k} + A_l) \cdot (B_h 2^{8k} + B_l) \\ &= A_h \cdot B_h 2^{8s} + (A_h \cdot B_l + A_l \cdot B_h) 2^{8k} + A_l \cdot B_l \\ &= A_h \cdot B_h 2^{8s} + ((A_h + A_l) \cdot (B_h + B_l) - A_h \cdot B_h - A_l \cdot B_l) 2^{8k} + A_l \cdot B_l. \end{aligned}$$

According to the above equation, we can replace a s -word multiplication by computing three $(s/2)$ -word multiplications and a couple of additions. However, it should be noticed that when computing the $(s/2)$ -word multiplication $(A_h + A_l) \cdot (B_h + B_l)$, $A_h + A_l$ and $B_h + B_l$ may produce carry bits that may increase the computational cost. Therefore, a more efficient Karatsuba algorithm, subtractive Karatsuba, is proposed in [73]. The subtractive Karatsuba algorithm replaces $(A_h + A_l) \cdot (B_h + B_l)$ to avoid producing carry bits by computing the multiplication of two absolute difference $|A_l - A_h|$ and $|B_l - B_h|$ and one conditional negation of $|A_l - A_h| \cdot |B_l - B_h|$.

Our Implementation. We first list several previous work as Table 3.1 on performing such operations and remark which one was the speed record.

Table 3.1: Comparison of Multi-precision Multiplication/Squaring Implementations on 8-bit AVR Processors.

Implementation	Multiplication	Squaring
Gura et al. [35]	Quadratic	N/A
Scott and Szczechowiak [60]	Quadratic	Quadratic
Uhsadel et al. [58]	Quadratic	N/A
Kargl et al. [74]	Quadratic	N/A
Liu et al. [75]	Quadratic	Quadratic
Zhang and Großschädl [59]	Quadratic	N/A
Hutter and Wenger [61]	Quadratic	N/A
Seo and Kim [62]	Quadratic	N/A
Lee et al. [76]	N/A	Quadratic
Seo et al. [77]	N/A	Quadratic
Hutter and Schwabe [63]	Sub-quadratic(Speed-record)	Sub-quadratic(Speed-record)

As described in above table, the fastest approach for both operations is the Karatsuba algorithm in [63], which optimizes the number of operations in sub-quadratic computational complexity. This promotes us to implement the multi-precision multiplication and squaring using Karatsuba algorithm. There are two ways to implement Karatsuba algo-

rithm. 1. Combine the Karatsuba algorithm with the above multi-precision multiplication techniques. 2. Recursively adopted the Karatsuba algorithm until all the multiplications are just single-precision multiplications. In order to minimize the code size, we do not use the recursive Karatsuba algorithm, instead, we combine the Karatsuba algorithm with the RPS technique for 256-bit multiplication, which is a memory-efficient approach to implementation multiplication. The memory-efficient approach exploits only 128-bit wise multiplication and calls them three times to complete the whole 256-bit wise multiplication. For this reason, the code size of multiplication and squaring operations becomes almost one third. Interestingly, the execution timing of memory-efficient method is similar to speed-optimized approach. In our test, the memory-efficient 256-bit multiplication for requires 5,109 clock cycles and 3,526 bytes memory, while the 256-bit squaring requires 3,420 clock cycles and 2,346 bytes, respectively. These are most optimal results on 8-bit AVR processors.

Reduction of NIST P-256 and SM2 Prime Fields. The result of a 256-bit wise multiplication (or squaring) operation is a 512-bit integer, which must be reduced modulo P_{256} or P_{sm2} to get a 256-bit residue for NIST P-256 and SM2, respectively. We implemented the reduction on NIST P-256 and SM2 prime fields by using the fast reduction method. Such method consists of only cheap addition and subtraction operations rather than the expansive division operation, which saves the execution timing. In particular, the addition is performed in 32-bit wise from the least significant word to the most significant word. Afterward, the remaining subtraction routine is performed. Among operations for the modular reduction, the most expensive one is a memory access routine. In order to reduce the number of memory access, part of intermediate results are cached in the registers and can be directly used.

Fermat's Little Theorem based Inversion. The most common way to perform the modular inversion is the Extended Euclidean Algorithm, which is very efficient but not constant-time. In order to resist against timing attack, we choose the constant-time solution, i.e., Fermat's Little Theorem based inversion. For an element $z \in \mathbb{F}_p$, the inversion z^{-1} can be evaluated via $z^{-1} \equiv z^{p-2} \pmod{p}$. We can use the "addition chain" to minimize the number of multiplications in the iteration of inversion computation, and thus the constant-time inversion in $\mathbb{F}_{P_{256}}$ can be computed at a cost of 255S + 12M by following

Algorithm 2, while the inversion in $\mathbb{F}_{P_{sm2}}$ can be computed at a cost of 255S + 15M by Algorithm 3.

Algorithm 2 Fermat-based inversion for NIST P-256

Require: Integer z satisfying $1 \leq z \leq p - 1$.

Ensure: Inverse $t = z^{p-2} \bmod p = z^{-1} \bmod p$.

1: $z_2 \leftarrow z^2$	{ cost: 1S }
2: $z_3 \leftarrow z_2 \cdot z$	{ cost: 1M }
3: $z_{12} \leftarrow z_3^{2^2}$	{ cost: 2S }
4: $z_{15} \leftarrow z_{12} \cdot z_3$	{ cost: 1M }
5: $t_0 \leftarrow z_{15}^{2^4} \cdot z_{15}$	{ cost: 4S+1M }
6: $t_1 \leftarrow t_0^{2^8} \cdot t_0$	{ cost: 8S+1M }
7: $t_2 \leftarrow t_1^{2^{16}} \cdot t_1$	{ cost: 16S+1M }
8: $t_3 \leftarrow (t_2^{2^{32}} \cdot z)^{2^{96}}$	{ cost: 128S+1M }
9: $t_4 \leftarrow ((t_3^{2^{32}} \cdot t_2)^{2^{32}} \cdot t_2)^{2^{16}} \cdot t_1$	{ cost: 80S+3M }
10: $t \leftarrow ((t_4^{2^8} \cdot t_0)^{2^4} \cdot z_{15})^{2^4} \cdot z_{12} \cdot z$	{ cost: 16S+2M }
11: return t	

Algorithm 3 Fermat-based inversion for SM2

Require: Integer z satisfying $1 \leq z \leq p - 1$.

Ensure: Inverse $t = z^{p-2} \bmod p = z^{-1} \bmod p$.

1: $z_2 \leftarrow z^2$	{ cost: 1S }
2: $z_3 \leftarrow z_2 \cdot z$	{ cost: 1M }
3: $z_{15} \leftarrow z_3^{2^2} \cdot z_3$	{ cost: 2S+1M }
4: $t_0 \leftarrow z_{15}^{2^1} \cdot z$	{ cost: 1S+1M }
5: $t_1 \leftarrow t_0^{2^5} \cdot t_0$	{ cost: 5S+1M }
6: $t_2 \leftarrow t_1^{2^5} \cdot t_0$	{ cost: 5S+1M }
7: $t_3 \leftarrow t_2^{2^{15}} \cdot t_2$	{ cost: 15S+1M }
8: $t_4 \leftarrow t_3^{2^1} \cdot z$	{ cost: 1S+1M }
9: $t_5 \leftarrow (((t_4^{2^{32}} \cdot t_4)^{2^{31}} \cdot t_4)^{2^{31}} \cdot t_4)^{2^{31}} \cdot t_4$	{ cost: 125S+4M }
10: $t_6 \leftarrow (t_5^{2^4} \cdot z_{15})^{2^{32}}$	{ cost: 36S+1M }
11: $t_7 \leftarrow (t_6^{2^{31}} \cdot t_4)^{2^{31}} \cdot t_4$	{ cost: 62S+2M }
12: $t \leftarrow t_7^{2^2} \cdot z$	{ cost: 2S+1M }
13: return t	

3.2.2 Elliptic Curve Group Arithmetic

Curve Models, Coordinates and SPA Resistance. Optimizations for elliptic curve group arithmetic are usually based on the special form of curve equation. Most previous

work concentrates on Montgomery curves [50] or twisted Edwards curves [51]. Since the NIST P-256 and SM2 256 curves both have prime group orders, which do not have Montgomery form or twisted Edwards form. Therefore, we have to use the general-form (a.k.a. short Weierstrass form) $y^2 = x^3 + ax + b$ to represent them.

It is preferable to use the Jacobian coordinates (or extended ones) for faster point doubling and addition formulae. In this case, the projective equation of desired curves is $Y^2 = X^3 + aXZ^4 + bZ^6$, where the projective point (X, Y, Z) corresponds to the affine point $(X/Z^2, Y/Z^3)$. Moreover, since both curves have the coefficient $a = -3$, which could induce the expression $3X^2 + aZ^4 = 3(X + Z^2)(X - Z^2)$ in the group arithmetic proposed by [56], thus the cost of doubling a point $P_1 = (X_1, Y_1, Z_1)$ to $P_3 = 2P_1 = (X_3, Y_3, Z_3)$ in Jacobian coordinates is roughly $4M + 4S + 9A$ by the following procedure

$$\begin{aligned} A &= 3(X_1 + Z_1^2) \cdot (X_1 - Z_1^2), \quad B = 2Y_1, \quad C = B^2, \quad D = C \cdot X_1, \\ X_3 &= A^2 - 2D, \quad Y_3 = (D - Y_3) \cdot A - C^2/2, \quad Z_3 = B \cdot Z_1, \end{aligned}$$

while the cost of addition in mixed Jacobian-affine coordinates is $8M + 3S + 7A$. As claimed in [56], these are the fastest formulae for both curves in traditional Weierstrass model.

Traditional ECC algorithms implemented on embedded systems might be vulnerable to side channel attacks. To withstand such security threats, the scalar multiplication algorithm should be regular, or in other words, running in a fashion independent of the input scalar. There are several ways to prevent simple side channel attacks, which includes adding some dummy operations [78], using some model of elliptic curves which admit unified group arithmetic formulae (such as twisted Edwards model in [51]), and using some already-regular scalar multiplication algorithms (such as Montgomery ladder algorithm in [50]). There are plenty of literatures which focus on efficient and regular computation of scalar multiplication. However, killing two birds with one stone (i.e., achieving the best performance as well as high security in ECC implementation) is not an easy job in practice.

Co-Z Jacobian arithmetic on Weierstrass Curves. For implementing scalar multiplication on general-form elliptic curves defined over finite field with large prime character-

istic, we take both the efficiency and security into consideration and complete the current state-of-the-art. First of all, we review some optimization of the point addition proposed by Meloni [79], who evaluates co-Z Jacobian arithmetic with X and Y coordinates only.

Consider two input points sharing the same Z-coordinate as $P_1 = (X_1, Y_1, Z)$ and $P_2 = (X_2, Y_2, Z)$, the so-called co-Z addition of P_1 and P_2 ($P_1 \neq P_2$) is defined as $P_3 = P_1 + P_2 = (X_3, Y_3, Z_3)$, which could be evaluated through the following formula

$$A = (X_2 - X_1)^2, B = X_1 \cdot A, C = X_2 \cdot A, D = (Y_2 - Y_1)^2, E = Y_1(C - B),$$

$$X_3 = D(B + C), Y_3 = (Y_2 - Y_1)(B - X_3) - E, Z_3 = Z(X_2 - X_1).$$

This co-Z addition is very efficient, and it can be computed at the cost of $6M+2S+7A$. Moreover, we can update the coordinates of P_1 for free (i.e., an updated representation of P_1 would share the same Z-coordinate as $P_1 + P_2$), since $B = X_1(X_2 - X_1)^2 = x_1Z_3^2$ and $E = Y_1(X_2 - X_1)^3 = y_1Z_3^3$, where $(x_1, y_1) = (X_1/Z^2, Y_1/Z^3)$ denotes the affine coordinates of P_1 . Thus $P_1 \equiv (B : E : Z_3)$. It was shown that the conjugate $P_1 - P_2$ could also share the same Z-coordinate as $P_1 + P_2$, only bringing a small additional cost. Indeed, we have $P_1 - P_2 = (X_4, Y_4, Z_3)$, where

$$F = (Y_1 + Y_2)^2, X_4 = F - (B + C), Y_4 = (Y_1 + Y_2)(X_4 - B) - E.$$

“Somewhat” Montgomery Ladder on Weierstrass Curves. Based on the above group arithmetic formulae, Matthieu Rivain [80] proposed “somewhat” Montgomery ladder scalar multiplication algorithms for general-form (i.e., short Weierstrass) elliptic curves. Such algorithms are slightly different from traditional Montgomery ladder [50], but still fast and regular. Though these algorithms usually cost more than the traditional Montgomery ladder, they do not require the desired curves to be Montgomery type and would be applicable for more general cases.

Following the method of Rivain, we denote XYCZ-ADD (costs $4M+2S+7A$) as the function which takes the (X, Y) -coordinates of two co-Z points P_1, P_2 as input and computes the (X, Y) -coordinates of $P_1 + P_2$ as well as the update (X, Y) -coordinates of P_1 (i.e. P_1 and $P_1 + P_2$ share the same Z-coordinate). We also let XYCZ-ADDC (costs $5M+3S+11A$) be the function which computes the (X, Y) -coordinates of $P_1 + P_2$ and its co-Z conjugate

$P_1 - P_2$. Moreover, we denote `XYCZ-IDBL` (costs $2M + 4S + 10A$) as the function which processes the (X, Y) -only initial doubling with co- Z update, and `FinalInvZ` (costs $1I + 4M + 1A$) as the function which computes the inverse λ of the final Z -coordinate. Then based on the Algorithm 9 in [80], we proposed the Montgomery ladder with (X, Y) -only co- Z addition as Algorithm 4. Note that this algorithm always performs `XYCZ-ADDC` and `XYCZ-ADD` in each iteration, which means that it is regular. Moreover, we used an index (c) in the algorithm in order to ensure the implementation without conditional statements.

Algorithm 4 Montgomery ladder with (X, Y) -only co- Z addition

Require: $P \in E(\mathbb{F}_q), k = (k_{n-1}, \dots, k_1, k_0)_2 \in \mathbb{N}$ with $k_{n-1} = 1$.

Ensure: $Q = [k]P$.

```

1:  $T \leftarrow (T[1], T[2], T[3], T[4])$ 
2:  $T \leftarrow \text{XYCZ-IDBL}(P)$ 
3: for  $i$  from  $n - 2$  to  $1$  by  $-1$  do
4:    $c_1 \leftarrow k_{1+i}$ 
5:    $c \leftarrow k_i$ 
6:    $c_0 \leftarrow (c_1 + c) \bmod 2$ 
7:    $T \leftarrow \text{XYCZ-ADDC}([T[1 + 2 \cdot c_0], T[2 + 2 \cdot c_0], T[3 - 2 \cdot c_0], T[4 - 2 \cdot c_0]])$ 
8:    $T \leftarrow \text{XYCZ-ADD}(T)$ 
9: end for
10:  $c_1 \leftarrow k_1$ 
11:  $c \leftarrow k_0$ 
12:  $c_0 \leftarrow (c_1 + c) \bmod 2$ 
13:  $T \leftarrow \text{XYCZ-ADDC}([T[1 + 2 \cdot c_0], T[2 + 2 \cdot c_0], T[3 - 2 \cdot c_0], T[4 - 2 \cdot c_0]])$ 
14:  $\lambda \leftarrow \text{FinalInvZ}([T[3 - 2 \cdot c], T[4 - 2 \cdot c]], [T[1 + 2 \cdot c], T[2 + 2 \cdot c]], P, c)$ 
15:  $T \leftarrow \text{XYCZ-ADD}(T)$ 
16: return  $[T[2 \cdot c + 1] \cdot \lambda^2, T[2 \cdot c + 2] \cdot \lambda^3]$ 

```

Rivain in [80] also proposed some other scalar multiplication algorithms, but essentially they could be viewed as variants of the above one, where the “M-S” strategy [81] is utilized. According to the S/M and the A/M cost ratios in our test (details can be referred to next section), Algorithm 4 is more efficient than other variants when implemented on the desired platform.

3.3 Implementation and Evaluation

Our implementations are based on the 8-bit AVR ATmega128 processor, while the results are validated using *Magma* (a famous and well-supported software package for

computational algebra). The comparison results of finite field arithmetic for both NIST P-256 and SM2 curves are given in Table 3.2. Since both curves have similar modulus, the finite field arithmetic routines are also performed in a similar routine. For this reason, the finite field arithmetic operations in $\mathbb{F}_{P_{256}}$ achieved very similar performance to those in $\mathbb{F}_{P_{sm2}}$.

Table 3.2: Cycle counts of Finite Field Arithmetic Software for AVR ATmega128 Micro-controllers.

Curve	ADD	SUB	MUL	SQR	INV
NIST P-256	418	424	6,609	4,919	1,347,009
SM2	422	423	6,491	4,797	1,326,246

In Table 3.3, the comparison results of scalar multiplication software on 8-bit AVR processors are given. We listed the state-of-art implementations for NIST curves. In terms of binary field ECC implementation, the work of Aranha et al. [38] on NIST K-233 curve shows the highest performance among them by using the endomorphism of Koblitz curve. For the case of prime field ECC implementation, we target the NIST P-256 curve, which provides sufficient security margin than NIST P-224 curve and the required clock cycle of proposed implementation is only 25,384,494 clock cycles. This is faster than previous works [39] by 27.3%. The proposed implementation achieved the highest performance for the 128-bit security level NIST curve. Furthermore, the code size is also smaller than previous works [39] by 13.2%. To the best of our knowledge, this is the first work which considers the implementation of SM2 curve on 8-bit AVR ATmega microcontrollers. For this reason, we only list our results on the Table 3.3. As we observed in Table 3.2, the timings for finite field operations are almost similar so NIST P-256 and thus the SM2 curve shows very similar performance.

3.4 Chapter Conclusion

In this chapter, we re-designed some existent techniques to fit 8-bit resource-constraint platform, and achieved the new speed record for NIST P-256 and SM2 curves on 8-bit AVR processors.

- For finite field arithmetic, we adopt the following techniques: (1) the optimized

Table 3.3: Cycle Counts and Code Size of Scalar Multiplication Software for AVR AT-mega Microcontrollers.

Implementation	Curve	Clock cycles	Code size (bytes)
Aranha et al. [38]	NIST K-233	5,382,144	38,600
Aranha et al. [38]	NIST B-233	13,934,592	34,600
Gura et al. [35]	NIST P-224	17,520,000	4,812
Wenger et al. [39]	NIST P-256	34,930,000	16,112
Proposed implementation	NIST P-256	25,384,494	13,980
Proposed implementation	SM2	24,925,764	13,820

masked operand technique is exploited for modular addition and subtraction to reduce the number of mask computations and latency; (2) the multi-precision multiplication and squaring operations are based on memory-efficient subtractive Karatsuba technique to achieve the sub-quadratic complexity; (3) the results are efficiently reduced through fast reduction techniques such as optimized addition/subtraction routine and cached intermediate results; (4) when it comes to the finite field inversion, we used Fermat's little theorem and addition-chain method to ensure fast and constant time solution.

- For elliptic curve group arithmetic, we used co- Z representation as well as the Montgomery ladder algorithm. On one hand, this techniques preserve high efficiency in implementing scalar multiplication, as we can see in Table 3.2. On the other hand, our implementation is carried in a regular fashion, and thus implies the security against simple power analysis or timing attack.

Together with above techniques, finally, we achieved the fastest NIST P-256 and SM2 curves implementations on 8-bit AVR processors. The implementation of NIST P-256 is faster in timings and smaller in code size than previous works by 27.3% and 13.2% respectively. The implementation of SM2 curve shows similar performance of NIST P-256 since both curves have similar modulus. As far as we know, the work in this chapter is the first implementation of SM2 curve on 8-bit AVR processors.

Chapter 4

Endpoint Security and Privacy for IoT: Systematic Exposition

In this chapter, we provided an overview of efficient cryptography for IoT endpoints and system privacy issues. Networked endpoint devices, like programmable logic controllers (PLCs), are an area of particular concern. These devices are vulnerable to physical tampering, a typical deployment often leaves devices unattended, and a target for remote, logical attacks as they offer a stepping stone to access the wider system. In addition, the size of IoT deployments requires security mechanisms to be scalable. The first point of this article is the endpoint device security by introducing existing cryptographic mechanisms for IoT, and discussing efficient algorithm implementation (resources/execution time). Traditional signal processing and IoT cryptographic algorithm implementation has similarities, given the challenges of optimizing the underlying mathematical operations of cryptography for resource efficiency and speed. The second IoT issue considered is privacy management. Privacy could relate to device level (directly linked to specific user) or system level data (information about a specific user is inferred from multiple sources). In some IoT application scenarios, the perception of security by people interacting with the system could be crucial to system acceptance and deployment. In addition, we demonstrate a high level discussion on crypto for system level data security and privacy (data integrity only or also confidentiality, including a discussion on mechanisms to ensure IoT adheres to privacy standards/legal compliance (for IoT overlap with consumers, such as smart meters/home)).

Table 4.1: IoT consortium and OpenFog Security Objectives and Recommendations

Functions	Security Objectives		Security Recommendations
	IoT consortium	Open Fog	
Physical	X	X	Tamper resistance, evidence, detection and response
Trust	X	X	Hardware root of trust(HW-RoT), secure or verified boot, remote attestation, secure boot processes
Identity	X	X	Credentials, immutable identifier with attestation
Access Control	X	X	Authentication (cryptographic) and authorization
Integrity Protection	X	X	Secure boot, run-time integrity checking and introspection
Data Protection	X		Data confidentiality and integrity (cryptographic)
Monitoring and analysis	X		Detect anomaly events
Configuration & Management	X		Signed Software (cryptographic)
Cryptographic techniques	X	X	Symmetric encryption, message authentication, hash
	X	X	Asymmetric encryption – Integer and Elliptic Curve Secure key generation and storage
Isolation techniques	X		Trusted execution environment/hypervisor

4.1 Security Requirements of IoT Endpoint Devices

The value attached to security requirements are often subjective and application-specific. As such, instead of promoting our own opinions, we use as a basis two well-known specifications of detailed security requirements for endpoint devices within the IoT. The IoT consortium has developed a general security architecture, which is to be used in conjunction with the reference framework [4]. Six inter-operational building blocks, organized within three layers, form the functional basis of the security framework. The top layer comprises four foundations, namely: endpoint protection, communication and connectivity protection, security monitoring and analysis and security configuration management. A similar reference architecture has been developed by the OpenFog Consor-

tium. The security pillar defined in the architecture specifically discusses the following important security attributes required of a fog device: privacy, anonymity, integrity, trust, attestation, verification and measurement.

The general security specifications developed by the IoT consortium and the OpenFog Consortium largely overlap, as shown in Table 4.1. Whereas the IoT consortium provides more general guidelines as to what security services should be included and the objectives that they should meet, the OpenFog Consortium provides more specific recommendations as to the technical mechanisms that could be used in order to provide a sub-set of these services. Looking at the two architectures in combination provides a good indication of what is expected of a secure endpoint device within the IoT. What can be seen from both specifications is that cryptographic mechanisms are required and play an important role in several security functions, such as access control (authentication), device configuration and management, and data protection.

Any cryptographic solution will come at a cost, either in terms of additional device resources or system processing delay. IoT devices are often highly resource-constrained, in comparison to traditional devices, and are required to operate at low power for months or years after their initial deployment. Although performance may improve with the use of new generation IoT processors, some implementations of cryptography are unsuitable for use on legacy devices. For example, with software cryptographic algorithm implementations, large increases in memory occupation, execution time and power consumption can be observed, particularly with older generation devices. Similarly, adding cryptographic mechanisms has the potential to hog resources and introduce delays, and as such that device becomes unable to operate in the real-time, mission-critical manner required. With all cryptographic mechanisms, an appropriate implementation is needed to ensure devices can provide security services while maintaining system functionality and ensuring that endpoint devices remain at a realistic cost point. Efficient cryptographic implementation in terms of execution time, resource cost, and energy consumption is therefore an important technical challenge that needs to be addressed for IoT endpoint devices.

4.2 Cryptographic Solutions for Endpoint Devices

Most IoT endpoint devices are equipped with embedded processors, which have limited computation resources and memory footprints. Endpoint devices are often deployed in critical areas, meaning there is not only a need to communicate and authenticate with the control center, but also between the devices themselves. On the other hand, an attacker may be able to access such devices and perform various kinds of physical attacks, e.g. side-channel cryptanalysis. The integration of side-channel-resistant cryptographic solutions to secure the communication and computation inside the devices is a nontrivial task, due to the resource constraints, and particularly the limited energy, of endpoint devices.

Symmetric key cryptography refers to algorithms where the same secret cryptographic key is used for both the process of encryption and decryption. There are three categories of symmetric ciphers, depending on the concrete functions: *block cipher*, *stream cipher* and *hash function*. The idea behind a block cipher is to first partition the plain text into relatively larger blocks (e.g., 128-bit for AES-128) and further encode each of the blocks separately. A stream cipher, in contrast, is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream), for example RC4. A hash function is any function that can be used to map data of arbitrary size to data of fixed size, for example, SHA3. The latter does not require a secret key, although it can be combined with a key to build symmetric cryptographic algorithms, such as HMAC.

From an algorithm standpoint, more than 20 lightweight ciphers have been designed and used in some devices since the 1990s. For example, the lightweight ciphers *A5/1*, *A5/2*, and *ORYX* designed in the 1990s have been used in cell phones; the ciphers *Hitag2* (designed in 2012) and *Megamos* (designed in 2013) have been adopted in car keys.

From an implementation perspective, most of the lightweight ciphers have been designed with special implementation properties, for example, *Hight*, *Clefia*, *DESXL*, and *Present*. An implementation of *Hight* requires approximately the same chip size as the Advanced Encryption Standard (AES) algorithm (3,048 versus 3,400 gate equivalents, or GEs) but the former is much faster. Most of the lightweight symmetric ciphers have the core structure of ARX-based and bitsliced-S-Box-based designs, and simple key schedules, thus requiring less memory footprint while achieving fast execution time.

MCUs are increasingly equipped with encryption hardware accelerators for standardized symmetric encryption and hash algorithms, such as AES, 3DES, SHA, and TRNG. However, asymmetric cryptography, often proposed for device and message authentication and key exchange, is still very expensive when it comes to directly integrating it with IoT endpoint devices. The following subsection will therefore mainly focus on lightweight implementation of elliptic curve cryptographic algorithms for IoT endpoint devices.

4.3 Asymmetric Cryptographic Algorithms for IoT Endpoint Devices

Asymmetric cryptography, also commonly referred to as public-key cryptography, offers scalable solutions for key exchange and digital signatures, which are important in large IoT networks. Key exchange can be seen as a method to securely establish the secrecy key via a public channel. The Diffie-Hellman (DH) key exchange, first published by Whitfield Diffie and Martin Hellman, is one of the earliest practical examples implemented in the field of cryptography. The security of DH key exchange is based on the hardness of the discrete logarithm problem. RSA, first published by Rivest, Shamir and Adleman, is based on the hardness of the Integer Factorization Problem (IFP) and allows for encryption and digital signatures.

The key point of any software implementation of a public-key cryptographic scheme for endpoint devices is to find a suitable compromise between the following four requirements: (1) short execution time, (2) high flexibility and scalability (i.e. support of curves providing different levels of security), (3) low memory (i.e. RAM) footprint, and (4) some basic protection against passive implementation attacks. Energy is, in general, the most precious resource of a battery-powered endpoint device. Compared to RSA and Diffie-Hellman public-key cryptography, Elliptic Curve Cryptography (ECC) is a lightweight public-key cryptography which was used in cryptography field by Neal Koblitz and Victor Miller in the 1980s. Its security is based on the intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP), which allows one to use much smaller groups (compared to its classical counterpart, RSA based on IFP). For example, it is generally

accepted that ECC instantiated with a 160-bit elliptic-curve group provides about the same level of security as the RSA signature scheme using a 1024-bit modulus. Moreover, ECC has short-sized public/private key pairs and occupies less memory footprint. These features make ECC more suitable to be used in the scenario of the IoT.

ECC can be used to implement key-exchange and digital signatures more efficiently than classical DH and RSA. Elliptic-curve Diffie-Hellman (ECDH) is an anonymous key agreement protocol that allows two parties, each having an elliptic-curve public-private key pair, to establish a shared secret over an insecure channel. Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography. ECDSA can be used to provide entity and data-origin authentication, integrity protection, and non-repudiation services, which makes it an essential tool for enabling secure communication. Common security protocols like TLS, often used in device-to-backend or gateway-to-backend communication within the IoT, rely on these security algorithms to authenticate the server to the client (and optionally, the client to the server) and to securely exchange the public keys needed for the establishment of an ephemeral shared secret.

Most of the current elliptic curve standards (e.g., NIST curve, IEEE P1363 curve) have adopted the form of a Weierstrass curve, and all of these standards rely on the fact that the elliptic-curve discrete-logarithm problem (ECDLP) is difficult. However, the security of real world ECC on IoT devices does not only mean the security of ECDLP, but also the security of concrete implementations. For example, the widely adopted NIST curve P-256 is not considered to be a safe curve and fails to provide the features of complete point addition formulas and indistinguishability from uniform random strings. In the past ten years, researchers have paid a lot of attention to evaluating new models of elliptic curves. Some examples of well-studied curve models are the Montgomery model [50] and twisted Edwards curve [51]. On the other hand, more than 15 years have passed since the standard curves were developed, and the cryptography community now has a better understanding of the security of elliptic curve cryptography and practical implementation issues. The current state-of-the-art has advanced. In research and other standards venues, newer variants of cryptographic schemes have been proposed which pursue better performance and/or simpler and more secure implementations. For example, MoTE-ECC is a

novel approach for the implementation of ephemeral ECDH key exchange that exploits the birational equivalence of Montgomery and twisted Edwards curves. By taking the individual computational advantages of the Montgomery form and twisted Edwards form into account, MoTE-ECC reaches higher performance (and better energy efficiency) and is also secure against basic side-channel attacks (e.g., timing attacks and simple analysis attacks). The Edwards-Curve Digital Signature Algorithm (EdDSA) is a state-of-the-art signature scheme using elliptic curves in (twisted) Edwards form that was developed with the intention of achieving both high performance (especially in software) and high security [82, 83]. A variant of the EdDSA is specified in RFC 8032 [84] and will be one of the signature algorithms supported in the next version of the TLS protocol, i.e. TLS 1.3.

From an arithmetic point of view, ephemeral ECDH key exchange between two sensor nodes requires each node to perform two scalar multiplications: one fixed point scalar multiplication to generate an ephemeral key pair and another random point scalar multiplication to obtain the shared secret. ECDSA requires one fixed point scalar multiplication $k \cdot P$ to perform signature signing, while the verification process is relatively computation intensive, requiring a double scalar multiplication with a form of $k \cdot P + l \cdot Q$, where k, l are positive integers called *scalar* and P, Q are points on an elliptic curve E over a finite field F_p . Thus, efficient implementation of scalar multiplication is critical for cryptographic schemes. As shown in Figure 1, an ECC implementation can be implemented in four layers: cryptographic protocols (e.g., ECDH, ECDSA), scalar multiplication, group arithmetic (e.g., point doubling, point addition) and field arithmetic (e.g., multiplication, addition). In the past 15 years, a great deal of research work has been done to improve the performance of elliptic curve operations on 8 or 16-bit microcontrollers, making ECC more attractive for resource-constrained environments. Most of the work improved the performance of scalar multiplication either by proposing the performance of field arithmetic (e.g., field multiplication) or choosing the special family of the underlying fields or elliptic curve models.

The first research line is to propose new variants of multi-precision arithmetic and focus on improving the standardized elliptic curve. The first really efficient ECC software for an 8-bit microcontroller was introduced by Gura et al. [35] in CHES 2004. Gura et al. introduced the first optimized multi-precision multiplication for small embedded devices

called hybrid multiplication. Hybrid multiplication combines the advantages of both the operand scanning method and product scanning method and was the first multi-precision platform-specified arithmetic that carefully optimized the number of addition with carry and memory-access instructions. Based on this classic method, Gura et al. reported an execution time of only $6.48 \cdot 10^6$ clock cycles for a full scalar multiplication over a 160-bit SECG-compliant prime field on IoT endpoint devices. In the 14 years since the publication of Gura et al.'s seminal paper, a large body of research has been devoted to further reducing the execution time of ECC on IoT devices. The majority of this research focused on advancing the multi-precision arithmetic operation or devising more efficient variants of it. For example, Lederer et al. [37] in WISTP2009 improved Gura et al.'s work to further reduce the number of addition with carry instructions by reorganizing the byte-multiplication in the inner loop and then implementing ECDH key exchange using a NIST P-192 curve. Their implementation requires an execution time of $12.33 \cdot 10^6$ cycles for a random base point scalar multiplication and $5.20 \cdot 10^6$ cycles when the base point is fixed and known a priori. Besides implementation on 8-bit endpoint devices, another platform which frequently sees endpoint devices used is the MSP430 series of microcontrollers produced by Texas Instruments. On such 16-bit platforms, Wang et al. reported one of the first ECC software implementations on some Weierstrass curve defined over a 160-bit prime field, of which the execution time is 25.0 (resp. 28.1) million cycles for a fixed-base (resp. variable-base) scalar multiplication. Some well-known libraries on endpoint devices are TinyECC, WM-ECC and Nano-ECC, all of which are highly scalable and configurable, and support Weierstrass curves defined over 128, 160, and 192-bit prime fields.

Another research line is to employ a special family of prime fields or elliptic curves to further reduce the energy consumption of elliptic curve key exchange and signature. One classic ECC software implementation for an endpoint device equipped with an 8-bit microcontroller was reported by Woodbury et al. in 2000. The authors chose a special family of fields called *Optimal Extension Field (OEF)*, which refers to a finite field consisting of p^m elements where p is a *pseudo-Mersenne prime* (i.e. a prime of the form $p = 2^k - c$) and m is chosen such that an irreducible binomial $x(t) = t^m - \omega$ exists over $\text{GF}(p)$. The specific OEF is $\text{GF}((2^8 - 17)^{17})$, which allows the arithmetic operations, especially the multipli-

Table 4.2: the execution time of existing ECC-based implementations for IoT endpoint devices

Method	clock cycles	security lever	platform
Gura et al. [35]	$6.48 \cdot 10^6$	80 bit	8-bit AVR
Lederer et al. [37]	$5.20 \cdot 10^6$	96 bit	8-bit AVR
Liu et al. [85]	$8.59 \cdot 10^6$	128 bit	8-bit AVR
	$6.10 \cdot 10^6$	128 bit	16-bit MSP430
Liu et al. [86]	$1.8 \cdot 10^6$	102 bit	/
Dull et al. [35]	$7.0 \cdot 10^6$	128 bit	8-bit AVR
	$4.5 \cdot 10^6$	128 bit	16-bit MSP430
	$1.8 \cdot 10^6$	128 bit	32-bit ARM
Liu et al. [88]	$2.9 \cdot 10^6$	128 bit	8-bit AVR
	$1.8 \cdot 10^6$	128-bit	16-bit MSP430
	$0.23 \cdot 10^6$	128-bit	32-bit ARM
Liu et al. [89]	$1.6 \cdot 10^6$	128 bit	32-bit ARM

cation and inversion, to be executed efficiently with small devices. Their implementation requires an execution time of roughly $100 \cdot 10^6$ clock cycles for random point scalar multiplication. Liu et al. in INSCRYPT2013 adopted *Optimal Prime Fields (OPF)* and studied the suitability of OPFs for a lightweight implementation of ECC with a view towards high performance and security. They proposed a performance-optimized implementation using a Montgomery curve and a security-optimized implementation using a GLV curve on an 8-bit IoT platform. Later in [85], Liu et al. presented the design of a scalable, regular, and highly-optimized ECC library using a MoTE curve for both MICAz and Tmote Sky IoT endpoint devices, which supports both widely-used key exchange and signature schemes. Their parameterized implementation of elliptic curve group arithmetic supports pseudo-Mersenne prime fields at different security levels with two optimized-specific designs: the high-speed version (HS) and the memory-efficient (ME) version. Some other well-known fast and secure ECC implementations on endpoint devices include FPGA implementation of signature verification operation [86], NaCl library, Curve25519 implementation [87] the recently proposed FourQ [88] and MEECC (Memory-Efficient ECC) [89] libraries. We summary the execution time of existing implementations in Table 4.2.

4.4 Device Key Management for IoT

IoT devices are commonly used and facilitate the application of various wireless communication technologies for small devices with low cost hardware and software interfaces. At the same time, secure communication and related applications rely on the security of key management inside the IoT devices and their supporting environment. An attacker always wants to compromise a device and get the secret key via communication interception, side-channel analysis or reverse engineering. If the attacker can manage to reveal the device key, the time needed is substantially reduced. In such an event, if a mechanism existed to deactivate this compromised key, the potential risk from the attack could be mitigated. Under such a malicious IoT application environment, we need to establish secure device key management technology to defend our IoT-enabled applications.

The key inside of IoT devices have their individual life-time cycle, and the key management is to manage various key life-time cycles for many IoT devices. The life-time cycle includes the random key bit generation, key distribution among devices, key storage and key update/revocation. IoT devices' secure key management are quite challenging and extraordinarily difficult to implement when there are a large number of unattended devices. There are some challenging issues we have to clarify, as follows.

- The devices are produced by different external manufacturers, it is necessary for them to be provisioned with cryptographic keys, and those keys must be protected once provisioned. Different key sizes provide different security levels;
- IoT devices can be more easily hacked compared to conventional computing devices, such as a PC, or tamper-proof devices, such as a smart card, so the update mechanism should be robust and capable of providing key recovery functions; and
- IoT devices are resource-restrained, for this reason, it is difficult to employ conventional cryptography-based key management schemes directly on IoT devices.

Cryptography is one of fundamental primitives in IoT secure key management. Cryptographic techniques are applied after the keying material is agreed upon in advance in communicating IoT devices. As such a main task of key management protocols for IoT, the key management mechanism should be centralized, decentralized or distributed for

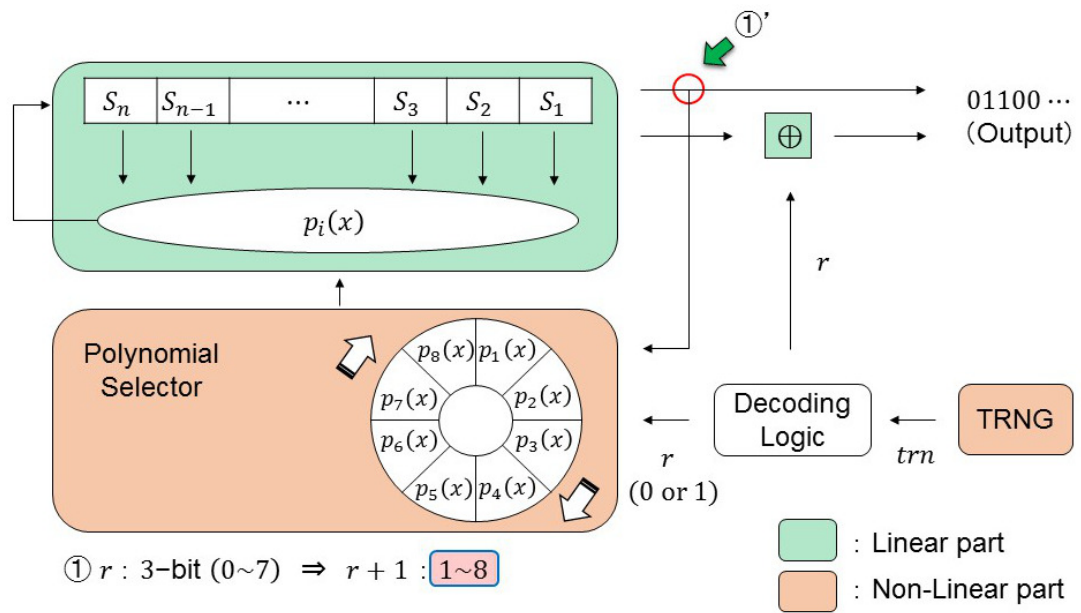
IoT applications. Centralized solutions are based on a centralized implementation called a key distribution center (KDC) which produces and distributes the keys to all IoT devices. Decentralized solutions operate on a network partitioned into a fixed number of small groups where each group has a managing device. The functionality of the KDC is to share the keys between the group managing devices that are usually organized in a hierarchical structure. For, distributed solutions, nodes collaborate to ensure key management operations such as key generation, distribution, renewal and revocation.

4.4.1 Secure Key Generation

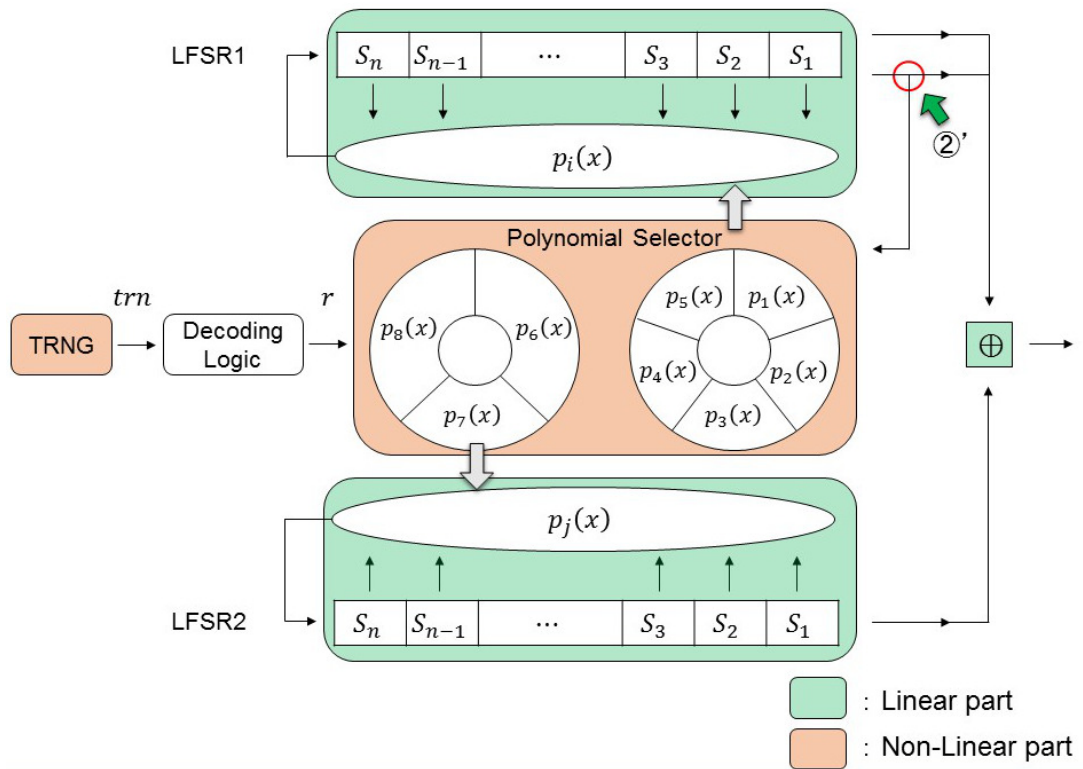
The secret keys in many IoT devices have to be pre-installed. However, this kind of method is vulnerable to adversary who can perform reversed engineering hardware or software to obtain the secret keys, so it is preferable the keys to be updated by the devices themselves. In such cases, for the general purpose of lightweight key generation for IoT, we should design mechanisms that satisfy the following properties:

- **Low resource consumption:** The resource consumption of both hardware and software for generating the pseudo-random key bits must be low due to the limited power available to IoT devices.
- **Low Memory Requirement:** The data stored in IoT devices should be kept as small as possible since the memory in IoT devices are normally extremely constrained.

Many existing solutions are based on lightweight cryptography and utilize Linear Feedback Shift Register (LFSR) designs to keep the cost low. To give an overview of these designs, we discuss as examples two proposals for key generation in lightweight IoT devices. The first proposal is based on modified multiple LFSRs pseudo-random number generators. The basic idea is to make a random choice from eight 16-bit LFSRs. It is inspired by Sugei's J3Gen scheme, where the feedback polynomials are implemented as a wheel which rotates depending on the bit value given by the TRNG module. If the truly random bit is a logical 0, the wheel rotates one position; that is, it selects the next feedback polynomial. Conversely, if the truly random bit is a logical 1, then the wheel rotates two positions; that is, the Polynomial Selector jumps one feedback polynomial and selects the next one. The first proposal modifies this, as shown in Fig 4.1(a).



(a) Proposal One



(b) Proposal Two

Figure 4.1: Lightweight Key Generation for IoT devices

In the second case which is shown in Fig.4.1(b), the randomized key bits are loaded into two independent register respectively, and the randomization is executing in this two registers. Our proposal is inspired by a well-know light-weight stream cipher, KATAN-

TAN [90], the random key bits are derived from LFSR while doing randomization. For each round, some bits are taken from the LFSR and input to the mixing process, or two nonlinear Boolean functions can be used. Our simple construction is modified by adding internal random bit instead of using the computational costly non-linear functions. The Boolean functions used in our construction will output random bits which is loaded to the least significant bits of the registers, after the internal key bits are shifted. This should be done in an invertible manner. To ensure sufficient randomization when generating the key bits, the devices should wait for several rounds of the LFSR processing to be executed, after that the devices can obtain key bits with higher security.

In many IoT applications, the input seed of internal PRNG state is loaded once and fixed inside the device, that is a vulnerability allowing adversary to obtain it to predict the key bits. To improve the security in such cases, our proposal provide an efficient randomized approach which make the inout seed not stored in memory to avoid the attack. We also construct new internal operations of XOR-Expression for the irreducible polynomials used in our PRNG.

As a routine in implementing security primitive for IoT devices, many solutions uses the XOR operator whose implementation is cheap, usually using some LFSR only. The degree of an XOR-Expression depends on the number of distinctly named variables in an expression. We can observer that the sum of three irreducible polynomial expression $x \oplus y \oplus z$ has a degree of three, but the sum remains linear which require non-linear processing to make it more secure. The purpose of our proposal is to increase the degree for randomizing the internal state of key bits without increasing hardware and software resource consumption during the implementation. For some distinct bitwise variables inside the IoT devices, we can select our customized reduced polynomial form if the irreducible is expressed as the minimum degree which still make security analysis simple.

We divide our chosen LFSR into two parts to lower the cost of implementation. At the same time, we attain a stronger security condition by doubling the internal state of LFSRs with two eight-bit LFSRs combined together, allowing key bits generation to get the full randomness from a sixteen-bit LFSRs.

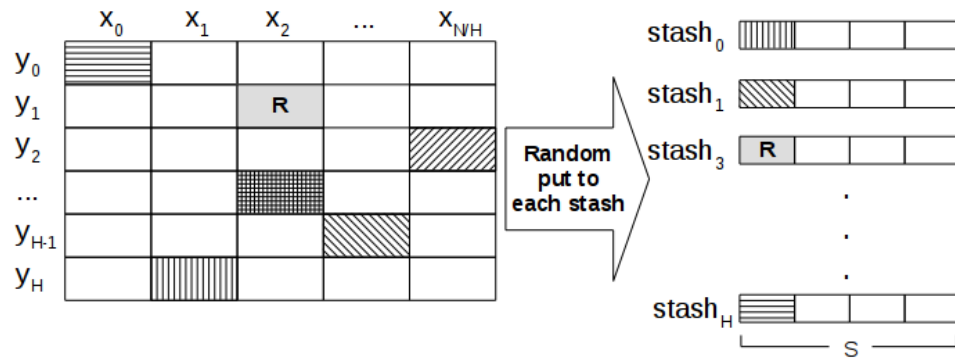
The reasons for our irreducible polynomial assignment are to achieve efficient hardware implementation by choosing polynomials with several coefficients in common, with

the common coefficients x_{16}, x_{11}, x_6, x_5 and x_0 shared by two irreducible polynomials. Our methods simplifies the hardware construction with fewer gates. Furthermore, our methods make selections of these feedback polynomials more flexible and without potential key bit leakage. We can employ a dynamic key server also generates a key for an identity pattern without storing that key using our methods. Access to a key works the same way as with a static key server, except the key is generated again for subsequent retrieval. A dynamic key server depends on a functional derivation per IoT identity for a key, if the same identity is presented multiple times, the same key bits will be XORed with random bits.

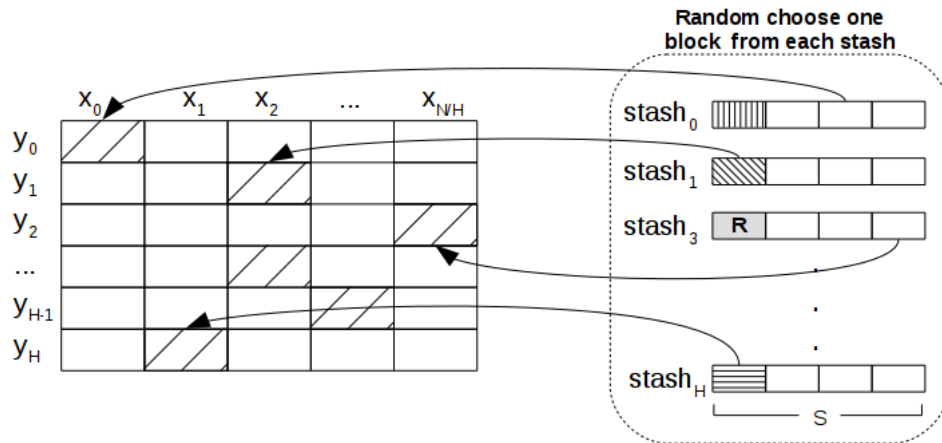
4.4.2 Secure Key Storage and Retrieval

Many IoT devices are not tamper resistant and do not have access to trusted hardware modules. To protect the keys in such IoT devices, it is very important to hide the memory access pattern in the devices themselves because adversaries can observe the memory read and write operations and get the key via side-channel attack or Trojan virus. One possible approach is whenever the IoT device reads or make update for the key in memory, we make all the key bits access pattern randomized when communicating with cloud servers or other IoT devices. This can be accomplished with Oblivious Random Access Mechanism (ORAM) schemes, for example, as shown in our method in 4.2. The accessed location of one key bit can be important information required by the IoT device, because during encryption and decryption process, the key bits are the most accessed compared to normal data access.

For this reason, the uploaded data blocks and the memory locations which contain the secret key bits should be different as was previously downloaded. Whenever the IoT device wants to access (read or write) key bits data, the address (x_i, y_j) is obtained from the position map. The device then reads H blocks from the server, one block from each row in the matrix. The devices will choose the memory units or data block via columns and rows of the matrix such as when the row is y_j , then the column is x_i . Otherwise, the data block locations are chosen uniformly at a random manner from the set of memory units accessed by the previous operation, and the columns are chosen in uniformly random manner for the remaining rows.



(a) Key Bits Read Operation



(b) Key Bits Write Operation

Figure 4.2: Matrix-based Oblivious Random Access Mechanism

The purpose of choose columns more randomly (in addition to the block which includes the data of interest) is ensuring the adversary cannot track the key bits access pattern which contains location of key bits storage in the memory. In our scheme, we also make some key bits locations remain the same as the previous memory access. In such cases, the adversary cannot distinguish whether previous and latter access is different or not. That is, if we don't choose some addresses from the previous memory locations which access the same key bits, then accessing two different key bits would result in two difficult memory locations, which allowing the adversary to identify the access patterns.

Compared to other existing ORAM schemes, we improve the security by making re-encryption using AES after each time an IoT memory unite is accessed, After a memory unit with data block is downloaded, it will be decrypted in local IoT devices. For next data access, the devices use a new key. Therefore, the adversary cannot identify that the uploaded data is the same as that which was previously accessed. In M-ORAM,

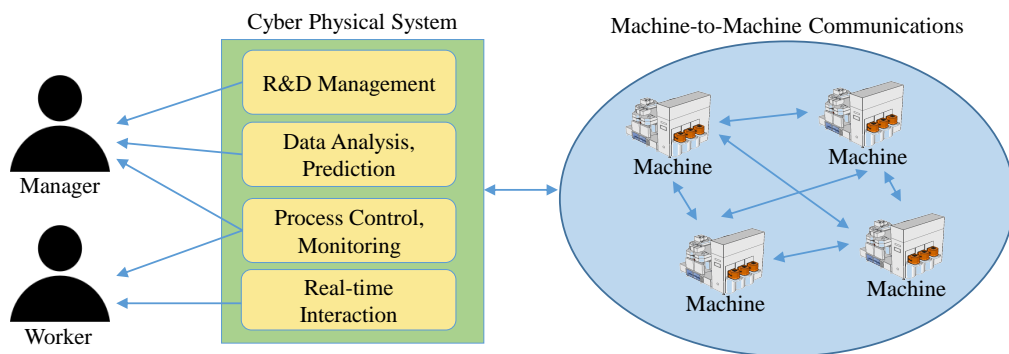


Figure 4.3: Back-End and Human Interface of IoT

we can apply any encryption schemes, where the data block and its ID are encrypted using a key generated from a pseudo-random function (PRF). Importantly, the PRF takes dataID (unique to each memory unit), a common secret key for all memory units and a counter which is associated with each memory unit as input, which reduces the resource occupation when making the key storage and access in IoT devices.

4.5 Privacy for IoT

Rapid advancement in wireless communications and pervasive computing abilities of smart objects have brought about a new era of application development, from control systems to critical IoT infrastructure, providing intelligence and optimization of processes with regard to resource-utilization. With an increasing number of IoT objects being equipped with technology to provide identification, computation and communication capabilities during operation processes, we also need to consider system-wide security and privacy issues. A secure endpoint should be part of an overall security approach that ensures data security for any interactions among endpoint devices (or smart things) and the backend cyber-physical system (CPS), as shown in 4.3. The privacy implications of system data, especially data originating from customers, should also be considered with respect to processing, storage and access by system operators. When considering data security and privacy in IoT, there are three levels of sensitive (or private) data and information involved.

1. **User Level:** This level involves access control to allow authorized persons to access appropriate-level data stored on cloud clusters (or objects). Examples of such data include real-time monitoring data and meaningful analyzed information. Useful protection techniques include ID-card-based login mechanisms and biometric authentication. In addition, a log corresponding to each access activity must be maintained for audit.
2. **Machine Level:** At this level, data is stored and transmitted among multiple objects (and gateways) in an IoT system. Snooping on the network for the purpose of probing organization-oriented private data, such as identification and access history, is highly possible when data is transmitted in an unprotected way across networks. It is suggested to implement secure machine-to-machine (M2M) communications, device management, and automatic firmware updating to maintain data confidentiality and system robustness.
3. **System Level:** An CPS, consisting of physical and software components, acts as a computing platform which monitors and controls physical processes. The data processing in an CPS is critical for IoT. This level involves data collected from machines and analyzed by the CPS itself. Without appropriate security mechanisms, organizational privacy leakage is unavoidable. Enhanced efforts on security architecture for CPS are strongly suggested.

While the IoT promises new opportunities for innovative service applications and business models through effective use of next generation mobile devices, it brings with it many challenges with respect to CPS, such as Slammer worm, Stuxnet and DUQU, as well as with regard to the end-point (e.g., device and user), such as individual (or organizational) privacy concerns, social engineering, man-in-the-middle attack, denial of service attack, reverse engineering, malware, and side-channel attacks [45]. As a result, in terms of the enhancement of security and privacy for IoT, significant efforts have been dedicated to eliminating these potential vulnerabilities and threats. In the following section we will discuss possible solutions, based on the aspects of data confidentiality, integrity and authenticity, privacy protection on cloud servers with big data analysis, and privacy management on end-objects, respectively.

4.5.1 Data Confidentiality, Integrity and Authenticity

Cryptography techniques, such as encryption, hash functions and digital signatures, are an important area when it comes to ensuring data confidentiality, integrity and authenticity. However, computing resource limitations and the heterogeneity of IoT objects give rise to critical new challenges, making it inevitable to re-engineer traditional security mechanisms or even create new solutions to fit the specific requirements of the IoT. First of all, authenticated encryption is one of the most promising techniques to secure IoT endpoint devices, as it is able to provide both confidentiality and authenticity of data, while achieving high efficiency of computation and end-to-end communication. Recently, a competition called CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) was launched to search for a new authenticated encryption algorithm which can offer advantages over AES-GCM and is suitable for widespread use. So far, among the candidates in Round 3 of CAESAR, the computation efficiency of Deoxys, which adopts tweakable block and linear transformations, has shown itself to be suitably efficient for implementation on IoT applications. At the same time, the security density Deoxys provides is acceptable. In addition, another candidate, called *CLOC* & *SILC*, is secure against partial nonce misuse and can provide an acceptable security level as well. In particular, excellent performance, i.e. computation efficiency and memory utilization, can be achieved with small size data, and thus *CLOC* & *SILC* is suitable for machine-to-machine authentication. What's more, with the rapid growth and universality of wearable devices, it is feasible to implement a continuous authentication scheme for an IoT-based environment with users possessing wearables. New types of continuous authentication mechanisms, e.g. Brainwaves [91], have been realized to support continuous (or real-time) entity verification in the background without the need for direct input from the user. This shifts the retrieval of physical signals and bio-factors for entity verification and authentication closer to the consumer end.

4.5.2 Privacy Protection on Cloud Servers with Big Data Analysis

The use of predictive analytics to make useful decisions about individuals may have negative impacts. An illustrative example would be a case where, due to automated de-

cision making, a company promoted baby-related products to an expectant mother before she announced to her family she was pregnant. Similar situations may arise with regard to sensitive personal information about one's sexual orientation or health status. Moreover, it is increasingly difficult for organizations to anonymize data and simultaneously use the anonymized data for individual identification. Hence, it is critical to protect individuals' privacy during data processing, and the following tenets are recommended as a baseline for privacy protection:

- Data must be processed fairly and used for specified and lawful purposes;
- Unauthorized or unlawful processing of data must be efficiently detected and dealt with;
- Accountability should be guaranteed;
- The consent obtained for data processing should be freely given;
- Data must not be exploited without an adequate level of protection;
- Data must be adequate, relevant and must not be excessive in relation to the purpose for which it is processed; and
- Data processed for any purpose must not be kept for longer than is necessary for that purpose.

4.5.3 Privacy Management on Endpoint Devices

Data privacy on IoT objects requires an effective access control scheme to govern access to data stored inside these objects. It is recommended to extend traditional access control approaches to fine-grained context-based access control systems in which IoT objects can be dynamically controlled in terms of acquiring data based on context. In addition, implementing secure machine-to-machine (M2M) communications among IoT objects for data confidentiality is suggested. On the other hand, heterogeneous communication architectures are common in IoT-oriented environments because various types of smart objects and relevant communication techniques, such as Radio Frequency (RF), Bluetooth Low Energy (BLE), Zigbee, LoRa and WiFi, are adopted. It is necessary to

consider the robustness of the privacy protection schemes of IoT-based communication techniques, such as the random address technique of BLE and anonymous communication. Moreover, it is highly recommended to adopt standards for privacy protection in the IoT. As far as the European Union (EU) is concerned, the future evolution of EU laws and directives regarding privacy and personal data protection will see a move towards a Privacy by Design (PbD) legal framework [92, 93], where seven major processes are recommended:

1. Proactive and Preventative: anticipating and preventing privacy-invasive events before they happen.
2. Privacy as the Default: ensuring the maximum degree of data protection and privacy preservation in the IoT.
3. Privacy Embedded into Design: privacy protection must be an essential component of the core IoT system.
4. Full Functionality: preserving privacy must be accomplished without making any non-relevant tradeoffs with security.
5. End-to-End Security: all data relevant to the IoT must be securely collected, retained, and destroyed at the end of the process, which represents the concept of secure lifecycle management of information.
6. Visibility and Transparency: the user should know who possesses his/her data, what data have been collected and processed, and for what purposes.
7. Respect for User Privacy: offer users strong privacy defaults and appropriate notices with user-friendly options.

In addition, a complete process consisting of identification, preservation, collection, processing, review, analysis and production for the management of electronically-stored data and information is required to support auditing throughout the data life cycle. Finally, wearable devices undeniably represent one of the most promising paradigms in terms of ubiquitous computing in IoT-enabled scenarios. Good examples include fitness bands

(i.e. activity trackers), running watches and wearable glasses which are capable of connectivity to the Internet so as to enable the exchange of data without human intervention. In IoT scenarios, individuals may be embedded with their own wearables during working periods. Therefore, it is necessary to take stock of the efficiency, attendant benefits and security risks of the so-called “wear your own device” (WYOD) scenario(s) and to implement a WYOD model for the purpose of management.

4.6 Chapter Conclusion

There are several initiatives for specifying security specifications and requirements for IoT endpoints. One of the core security mechanisms required for secure endpoints is cryptographic algorithms. Although there is a mature set of algorithms available, challenges remain in terms of efficient algorithm implementation and associated key management in the context of the various constraints associated with the IoT. We presented a brief discussion on symmetric and asymmetric cryptographic algorithms. With the former increasingly being integrated using efficient cryptographic co-processors, future research challenges lie more with the latter, which is still often implemented in device software, where it must compete for resources with other system processes. In this regard, Elliptic Curve Cryptography (ECC) is a promising approach to providing both scalable key exchange and digital signature mechanisms in large IoT systems, and we provide an overview of current implementation approaches. Key management is also challenging on devices with limited resources and little-to-no trusted hardware. New methods for allowing devices to generate keys, in addition to storing and accessing them securely, are needed, and we provide examples of common lightweight approaches to LFSR-based key generation and oblivious random access mechanisms. Finally, we concluded with a system-wide overview of data security and privacy issues that need to be considered in the IoT, including future security issues related to big-data analysis and storage and data legal frameworks.

Chapter 5

Conclusions and Future Research

5.1 Conclusions

This research focused on solving the security authentication issues in the emerging cloud-based IoT architecture and the endpoint security and privacy especially the implementing of security and privacy cryptographic algorithms in the IoT.

We proposed a novel and efficient authentication protocol for cloud-based IoT architecture as authentication schemes are crucial for the cloud-based IoT applications. We researched an enhanced authentication scheme, continuous authentication and proposed a continuous authentication system with bio-feature for IoT. In the quest for the support of wearable equipment with brainwave retrieval functionality, we developed a concrete continuous authentication system using brainwaves as bio-features for IoT-based networks.

Since cryptographic primitives are crucial for the Internet of Things network. Implementation of cryptography is resource constraint IoT devices are the basis of the IoT security protocols. We proposed an efficient, secure and compact implementation of scalar multiplication on a 256-bit elliptic curve recommended by the SM2, as well as a comparison implementation of scalar multiplication on the same bit-length elliptic curve recommended by NIST. We re-designed the existent techniques to fit the low-end IoT platform, namely 8-bit AVR processors. The implementations evaluated on the desired platform show that the SM2 algorithms have competitive efficiency and security with NIST, which would work well to secure the IoT world.

Finally, we provided a high-level security explosion for the endpoint in IoT as end-

point is the cornerstone of security mechanism. Following that, we discussed some system-wide design considerations for data security and privacy in current and emerging system designs.

5.2 Future Research

With the development of IoT, besides the researches of this dissertation, there are many interesting works need to be issued in the future including the followings.

- How to combine cryptographic solutions with the current continuous authentication in resource-constrained IoT devices to maintain usability, security and privacy.
- With the development of the quantum technology, lightweight implementation of post-quantum cryptographic primitives in resource constraint IoT devices is an important problem to be addressed.
- Leveraging the emerging blockchain technique to realize the security and privacy of IoT. The blockchain technology has been viewed as a promising method for the IoT security. Though, there have been some pioneered work using blockchain for the IoT security, it is still lack of the unified architecture of combining the blockchain and IoT.

Acknowledgements

This doctoral thesis is not only the result of my own two-year Ph.D studies but also the guidance, support and help from many people around me. I am very appreciated for all of them.

The thesis would not have been possible without the support, supervision and advice of my supervisors Prof. Chunhua Su. I would like to express my sincere gratitude for the academic guidance and encouragement he gave me in the past two years. From time to time, his valuable insights and suggestions guided me to understand these problems in a better way and many times guided our discussions towards good solutions.

I am deeply grateful to Prof. Pham, Prof. Nakamura and Prof. Wang for being doctoral committee members and taking time to review my thesis and making precious comments to help me improve the quality of my dissertation.

Many thanks go to the administrative staffs at SAD of the University of Aizu: Miyuki Yamahira, Hitomi Hashimoto, Yuka Mori, Ayami Ueno and other staffs who help me a lot in the campus life during my stay in Japan.

I would like to appreciate effective collaboration and fruitful discussions with my external co-authors: Kuo-Hui Yeh, Gerhard Hancke, Hwajeong Seo, Xiong Li, Zhi Hu, Wayne Chiu, etc.

I would like to thank Prof. Liming Fang and Dr. Chunpeng Ge who really gave me lots of suggestions and guidance on my research during the period of my study abroad at College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics.

The deepest appreciation goes to my husband for his love, patience and understanding. Lastly, I would like to thanks my parents for their unconditional love and support.

Bibliography

- [1] R. H. Weber, “Internet of things—new security and privacy challenges,” *Computer law & security review*, vol. 26, no. 1, pp. 23–30, 2010.
- [2] A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou, and A. Bouabdallah, “A systemic approach for iot security,” in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*. IEEE, 2013, pp. 351–355.
- [3] E. Ronen and A. Shamir, “Extended functionality attacks on iot devices: The case of smart lights,” in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 3–12.
- [4] “Industrial Internet Consortium: Industrial Internet of Things Volume G4: Security Framework,” http://www.iiconsortium.org/pdf/IIC_PUB_G4_V1.00_PB-3.pdf, 2016, accessed on 4 Feb. 2018.
- [5] B. Tang, Z. Chen, G. Hefferman, S. Pei, T. Wei, H. He, and Q. Yang, “Incorporating intelligence in fog computing for big data analysis in smart cities,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2140–2150, Oct 2017.
- [6] Y. N. Singh, “Human recognition using fisher’s discriminant analysis of heartbeat interval features and ecg morphology,” *Neurocomputing*, vol. 167, pp. 322–335, 2015.
- [7] W. Louis, M. Komeili, and D. Hatzinakos, “Continuous authentication using one-dimensional multi-resolution local binary patterns (1dmrlbp) in ecg biometrics,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2818–2832, 2016.

- [8] D. Ramli, M. Hooi, and K. Chee, "Development of heartbeat detection kit for biometric authentication system," *Procedia Computer Science*, vol. 96, pp. 305–314, 2016.
- [9] V. Yano, A. Zimmer, and L. L. Ling, "Extraction and application of dynamic pupilometry features for biometric authentication," *Measurement*, vol. 63, pp. 41–48, 2015.
- [10] C. Galdi, M. Nappi, D. Riccio, and H. Wechsler, "Eye movement analysis for human authentication: a critical survey," *Pattern Recognition Letters*, vol. 84, pp. 272–283, 2016.
- [11] S. Sarkar, V. M. Patel, and R. Chellappa, "Deep feature-based face detection on mobile devices," *arXiv preprint arXiv:1602.04868*, 2016.
- [12] P. Samangouei, V. M. Patel, and R. Chellappa, "Attribute-based continuous user authentication on mobile devices," in *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*. IEEE, 2015, pp. 1–8.
- [13] K. Annapurani, M. Sadiq, and C. Malathy, "Fusion of shape of the ear and tragus—a unique feature extraction method for ear authentication system," *Expert Systems with Applications*, vol. 42, no. 1, pp. 649–656, 2015.
- [14] H. Crawford, K. Renaud, and T. Storer, "A framework for continuous, transparent mobile device authentication," *Computers & Security*, vol. 39, pp. 127–136, 2013.
- [15] M. Baloul, E. Cherrier, and C. Rosenberger, "Challenge-based speaker recognition for mobile authentication," in *Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG-Proceedings of the International Conference of the*. IEEE, 2012, pp. 1–7.
- [16] A. Kumar, S. Garg, and M. Hanmandlu, "Biometric authentication using finger nail plates," *Expert systems with applications*, vol. 41, no. 2, pp. 373–386, 2014.
- [17] S. H. Khan, M. A. Akbar, F. Shahzad, M. Farooq, and Z. Khan, "Secure biometric template generation for multi-factor authentication," *Pattern Recognition*, vol. 48, no. 2, pp. 458–472, 2015.

- [18] H. Saevanee, N. Clarke, S. Furnell, and V. Biscione, "Continuous user authentication using multi-modal biometrics," *Computers & Security*, vol. 53, pp. 234–246, 2015.
- [19] Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smart-phone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, 2016.
- [20] L. Lu and Y. Liu, "Safeguard: User reauthentication on smartphones via behavioral biometrics," *IEEE Transactions on Computational Social Systems*, vol. 2, no. 3, pp. 53–64, 2015.
- [21] S. Mondal and P. Bours, "A study on continuous authentication using a combination of keystroke and mouse biometrics," *Neurocomputing*, vol. 230, pp. 1–22, 2017.
- [22] J. Nader, A. Alsadoon, P. Prasad, A. Singh, and A. Elchouemi, "Designing touch-based hybrid authentication method for smartphones," *Procedia Computer Science*, vol. 70, pp. 198–204, 2015.
- [23] O. Alpar, "Frequency spectrograms for biometric keystroke authentication using neural network based classifier," *Knowledge-Based Systems*, vol. 116, pp. 163–171, 2017.
- [24] L. Zhou, Y. Kang, D. Zhang, and J. Lai, "Harmonized authentication based on thumbstroke dynamics on touch screen mobile phones," *Decision Support Systems*, vol. 92, pp. 14–24, 2016.
- [25] C.-L. Liu, C.-J. Tsai, T.-Y. Chang, W.-J. Tsai, and P.-K. Zhong, "Implementing multiple biometric features for a recall-based graphical keystroke dynamics authentication system on a smart phone," *Journal of Network and Computer Applications*, vol. 53, pp. 128–139, 2015.
- [26] O. Alpar, "Intelligent biometric pattern password authentication systems for touch-screens," *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6286–6294, 2015.

- [27] A. Alsultan, K. Warwick, and H. Wei, “Non-conventional keystroke dynamics for user authentication,” *Pattern Recognition Letters*, vol. 89, pp. 53–59, 2017.
- [28] W. Meng, W. Li, L.-F. Kwok, and K.-K. R. Choo, “Towards enhancing click-draw based graphical passwords using multi-touch behaviours on smartphones,” *Computers & security*, vol. 65, pp. 213–229, 2017.
- [29] M. L. Brocardo, I. Traore, and I. Woungang, “Authorship verification of e-mail and tweet messages applied for continuous authentication,” *Journal of Computer and System Sciences*, vol. 81, no. 8, pp. 1429–1440, 2015.
- [30] L. Fridman, A. Stolerman, S. Acharya, P. Brennan, P. Juola, R. Greenstadt, and M. Kam, “Multi-modal decision fusion for continuous authentication,” *Computers & Electrical Engineering*, vol. 41, pp. 142–156, 2015.
- [31] Y. Matsuyama, M. Shozawa, and R. Yokote, “Brain signal’s low-frequency fits the continuous authentication,” *Neurocomputing*, vol. 164, pp. 137–143, 2015.
- [32] C. Ntantogian, S. Malliaros, and C. Xenakis, “Gaithashing: a two-factor authentication scheme based on gait features,” *Computers & Security*, vol. 52, pp. 17–32, 2015.
- [33] S. Choi, I.-H. Youn, R. LeMay, S. Burns, and J.-H. Youn, “Biometric gait recognition based on wireless acceleration sensor using k-nearest neighbor classification,” in *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014, pp. 1091–1095.
- [34] A. D. Woodbury, D. V. Bailey, and C. Paar, “Elliptic curve cryptography on smart cards without coprocessors,” in *Smart Card Research and Advanced Applications*, ser. International Federation for Information Processing, J. Domingo-Ferrer, D. Chan, and A. Watson, Eds., vol. 180. Kluwer Academic Publishers, 2000, pp. 71–92.
- [35] N. Gura, A. Patel, A. S. Wander, H. Eberle, and S. Chang Shantz, “Comparing elliptic curve cryptography and RSA on 8-bit CPUs,” in *Cryptographic Hardware*

- and Embedded Systems — CHES 2004*, ser. Lecture Notes in Computer Science, M. Joye and J.-J. Quisquater, Eds., vol. 3156. Springer Verlag, 2004, pp. 119–132.
- [36] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, and R. Dahab, “NanoECC: Testing the limits of elliptic curve cryptography in sensor networks,” in *Wireless Sensor Networks — EWSN 2008*, ser. Lecture Notes in Computer Science, R. Verdone, Ed., vol. 4913. Springer Verlag, 2008, pp. 305–320.
- [37] C. Lederer, R. Mader, M. Koschuch, J. Großschädl, A. Szekely, and S. Tillich, “Energy-efficient implementation of ECDH key exchange for wireless sensor networks,” in *Information Security Theory and Practice — WISTP 2009*, ser. Lecture Notes in Computer Science, O. Markowitch, A. Bilas, J.-H. Hoepman, C. J. Mitchell, and J.-J. Quisquater, Eds., vol. 5746. Springer Verlag, 2009, pp. 112–127.
- [38] D. F. Aranha, R. Dahab, J. López, and L. B. Oliveira, “Efficient implementation of elliptic curve cryptography in wireless sensors.” *Adv. in Math. of Comm.*, vol. 4, no. 2, pp. 169–187, 2010.
- [39] E. Wenger, T. Unterluggauer, and M. Werner, “8/16/32 shades of elliptic curve cryptography on embedded processors,” in *International Conference on Cryptology in India*. Springer, 2013, pp. 244–261.
- [40] Z. Liu, H. Seo, J. Großschädl, and H. Kim, “Efficient implementation of NIST-compliant elliptic curve cryptography for 8-bit AVR-based sensor nodes,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1385–1397, 2016.
- [41] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, “On emerging family of elliptic curves to secure internet of things: Ecc comes of age,” *IEEE Transactions on Dependable & Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.
- [42] Z. Liu, J. Großschädl, Z. Hu, K. Jarvinen, H. Wang, and I. Verbauwhede, “Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things,” *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 773–785, 2017.

- [43] C. E. Administration, “Sm2 elliptic curve public key algorithms,” December, 2010.
- [44] Z. Wang and Z. Zhang, “Overview on public key cryptographic algorithm sm2 based on elliptic curves (in chinese),” *Journal of Information Security Research*, vol. 2, no. 11, pp. 972–982, 2016.
- [45] A. R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and privacy challenges in industrial internet of things,” in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.
- [46] Z. Yan and S. Zhao, “A usable authentication system based on personal voice challenge,” in *Advanced Cloud and Big Data (CBD), 2016 International Conference on*. IEEE, 2016, pp. 194–199.
- [47] P. Lamkin, “Wearable tech market to be worth \$34 billion by 2020,” <https://www.forbes.com/sites/paullamkin/2016/02/17/wearable-tech-market-to-be-worth-34-billion-by-2020/>, [Accessed on 13th August, 2017].
- [48] G. 32918, “Information security technology–public key cryptographic algorithm sm2 based on elliptic curves,” 2016.
- [49] N. Koblitz, “The state of elliptic curve cryptography,” *Designs Codes & Cryptography*, vol. 19, no. 2-3, pp. 173–193, 2000.
- [50] P. L. Montgomery, “Speeding the Pollard and elliptic curve methods of factorization,” *Mathematics of Computation*, vol. 48, no. 177, pp. 243–264, Jan. 1987.
- [51] D. J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, “Twisted Edwards curves,” in *Progress in Cryptology — AFRICACRYPT 2008*, ser. Lecture Notes in Computer Science, S. Vaudenay, Ed., vol. 5023. Springer Verlag, 2008, pp. 389–405.
- [52] D. J. Bernstein, “Curve25519: New Diffie-Hellman speed records,” in *Public Key Cryptography — PKC 2006*, ser. Lecture Notes in Computer Science, M. Yung, Y. Dodis, A. Kiayias, and T. Malkin, Eds., vol. 3958. Springer Verlag, 2006, pp. 207–228.

- [53] F. 186-2, “Digital signature standard (dss),” *Federal Information Processing Standards Publication 186-2*, National Institute of Standards and Technology, 2000.
- [54] D. J. Bernstein and T. Lange, “Safecurves: Choosing safe curves for elliptic curve cryptography,” accessed 1 May 2018, available for download at <https://safecurves.cr.yt.to>.
- [55] J. A. Solinas, “Generalized Mersenne numbers,” Centre for Applied Cryptographic Research (CACR), University of Waterloo, Waterloo, Canada, Tech. Rep. CORR-99-39, 1999.
- [56] D. R. Hankerson, A. J. Menezes, and S. A. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer Verlag, 2004.
- [57] A. Corporation, “Avr instruction set manual,” November, 2016, available for download at <http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-0856-AVR-Instruction-Set-Manual.pdf>.
- [58] L. Uhsadel, A. Poschmann, and C. Paar, “Enabling full-size public-key algorithms on 8-bit sensor nodes,” in *Security and Privacy in Ad-hoc and Sensor Networks — SASN 2007*, ser. Lecture Notes in Computer Science, F. Stajano, C. Meadows, S. Capkun, and T. Moore, Eds., vol. 4572. Springer Verlag, 2007, pp. 73–86.
- [59] Y. Zhang and J. Großschädl, “Efficient prime-field arithmetic for elliptic curve cryptography on wireless sensor nodes,” in *Proceedings of the 1st International Conference on Computer Science and Network Technology (ICCSNT 2011)*, vol. 1. IEEE, 2011, pp. 459–466.
- [60] M. Scott and P. Szczechowiak, “Optimizing multiprecision multiplication for public key cryptography,” Cryptology ePrint Archive, Report 2007/299, 2007, available for download at <http://eprint.iacr.org>.
- [61] M. Hutter and E. Wenger, “Fast multi-precision multiplication for public-key cryptography on embedded microprocessors,” in *Cryptographic Hardware and Embedded Systems — CHES 2011*, ser. Lecture Notes in Computer Science, B. Preneel and T. Takagi, Eds., vol. 6917. Springer Verlag, 2011, pp. 459–474.

- [62] H. Seo and H. Kim, “Multi-precision multiplication for public-key cryptography on embedded microprocessors,” in *Information Security Applications — WISA 2012*, ser. Lecture Notes in Computer Science, D. H. Lee and M. Yung, Eds., vol. 7690. Springer Verlag, 2012, pp. 55–67.
- [63] M. Hutter and P. Schwabe, “Multiprecision multiplication on AVR revisited,” *Journal of Cryptographic Engineering*, vol. 5, no. 3, pp. 201–214, 2015.
- [64] E. Oswald, “Enhancing simple power-analysis attacks on elliptic curve cryptosystems,” in *Cryptographic Hardware and Embedded Systems — CHES 2002*, ser. Lecture Notes in Computer Science, B. S. Kaliski, Ç. K. Koç, and C. Paar, Eds., vol. 2523. Springer Verlag, 2002, pp. 82–97.
- [65] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer Verlag, 2007.
- [66] G. de Meulenaer and F.-X. Standaert, “Stealthy compromise of wireless sensor nodes with power analysis attacks,” in *Mobile Lightweight Wireless Systems — MOBILIGHT 2010*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, P. Chatzimisios, C. V. Verikoukis, I. Santamaría, M. Laddomada, and O. Hoffmann, Eds., vol. 45. Springer Verlag, 2010, pp. 229–242.
- [67] C. D. Walter, “Simple power analysis of unified code for ECC double and add,” in *Cryptographic Hardware and Embedded Systems — CHES 2004*, ser. Lecture Notes in Computer Science, M. Joye and J.-J. Quisquater, Eds., vol. 3156. Springer Verlag, 2004, pp. 191–204.
- [68] Y. Sakai and K. Sakurai, “Simple power analysis on fast modular reduction with NIST recommended elliptic curves,” in *Information and Communications Security — ICICS 2005*, ser. Lecture Notes in Computer Science, S. Qing, W. Mao, J. Lopez, and G. Wang, Eds., vol. 3783. Springer Verlag, 2005, pp. 169–180.
- [69] D. Stebila and N. Thériault, “Unified point addition formulæ and side-channel attacks,” in *Cryptographic Hardware and Embedded Systems — CHES 2006*, ser.

- Lecture Notes in Computer Science, L. Goubin and M. Matsui, Eds., vol. 4249. Springer Verlag, 2006, pp. 354–368.
- [70] C. D. Walter and S. Thompson, “Distinguishing exponent digits by observing modular subtractions,” in *Topics in Cryptology — CT-RSA 2001*, ser. Lecture Notes in Computer Science, D. Naccache, Ed., vol. 2020. Springer Verlag, 2001, pp. 192–207.
- [71] J. Großschädl, R. M. Avanzi, E. Savaş, and S. Tillich, “Energy-efficient software implementation of long integer modular arithmetic,” in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2005, pp. 75–90.
- [72] Z. Liu, H. Seo, J. Großschädl, and H. Kim, “Reverse product-scanning multiplication and squaring on 8-bit AVR processors,” in *Information and Communications Security — ICICS 2014*, ser. Lecture Notes in Computer Science, L. C.-K. Hui, S. Qing, E. Shi, and S.-M. Yiu, Eds., vol. 8958. Springer Verlag, 2015, pp. 158–175.
- [73] M. Hutter and P. Schwabe, “Multiprecision multiplication on AVR revisited,” Cryptology ePrint Archive, Report 2014/592, 2014, available for download at <http://eprint.iacr.org/>.
- [74] A. Kargl, S. Pyka, and H. Seuschek, “Fast arithmetic on ATmega128 for elliptic curve cryptography,” Cryptology ePrint Archive, Report 2008/442, 2008, available for download at <http://eprint.iacr.org>.
- [75] Z. Liu, J. Großschädl, and I. Kizhvatov, “Efficient and side-channel resistant RSA implementation for 8-bit AVR microcontrollers,” in *1st International Workshop on the Security of the Internet of Things (SECIOT 2010)*, Tokyo, Japan, 2010.
- [76] Y. Lee, I.-H. Kim, and Y. Park, “Improved multi-precision squaring for low-end RISC microcontrollers,” *Journal of Systems and Software*, vol. 86, no. 1, pp. 60–71, 2013.

- [77] H. Seo, Z. Liu, J. Choi, and H. Kim, “Multi-precision squaring for public-key cryptography on embedded microprocessors,” in *International Conference on Cryptology in India*. Springer, 2013, pp. 227–243.
- [78] J.-S. Coron, “Resistance against differential power analysis for elliptic curve cryptosystems,” in *Cryptographic Hardware and Embedded Systems*, ser. Lecture Notes in Computer Science, Ç. K. Koç and C. Paar, Eds., vol. 1717. Springer Berlin Heidelberg, 1999, pp. 292–302.
- [79] N. Meloni, “New point addition formulae for ecc applications,” in *Arithmetic of Finite Fields*, ser. Lecture Notes in Computer Science, C. Carlet and B. Sunar, Eds., vol. 4547. Springer Berlin Heidelberg, 2007, pp. 189–201.
- [80] M. Rivain, “Fast and regular algorithms for scalar multiplication over elliptic curves,” *Iacr Cryptology Eprint Archive*, no. 2011, 2011.
- [81] P. Longa and A. Miri, “New composite operations and precomputation scheme for elliptic curve cryptosystems over prime fields,” in *Public Key Cryptography – PKC 2008*, ser. Lecture Notes in Computer Science, R. Cramer, Ed., vol. 4939. Springer Berlin Heidelberg, 2008, pp. 229–247.
- [82] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, “High-speed high-security signatures,” in *Cryptographic Hardware and Embedded Systems — CHES 2011*, ser. Lecture Notes in Computer Science, B. Preneel and T. Takagi, Eds., vol. 6917. Springer Verlag, 2011, pp. 124–142.
- [83] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, “High-speed high-security signatures,” *Journal of Cryptographic Engineering*, vol. 2, no. 2, pp. 77–89, Sep. 2012.
- [84] S. Josefsson and I. Liusvaara, “Edwards-Curve Digital Signature Algorithm (EdDSA),” Internet Research Task Force, Crypto Forum Research Group, RFC 8032, Jan. 2017.

- [85] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: Ecc comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.
- [86] Z. Liu, J. Großschädl, Z. Hu, K. Järvinen, H. Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 773–785, 2017.
- [87] M. Düll, B. Haase, G. Hinterwälder, M. Hutter, C. Paar, A. H. Sánchez, and P. Schwabe, "High-speed Curve25519 on 8-bit, 16-bit and 32-bit microcontrollers," *Designs, Codes and Cryptography*, vol. 77, no. 2–3, pp. 493–514, Dec. 2015.
- [88] Z. Liu, P. Longa, G. C. C. F. Pereira, O. Reparaz, and H. Seo, "Fourq on embedded devices with strong countermeasures against side-channel attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, 2017, pp. 665–686.
- [89] Z. Liu, H. Seo, A. Castiglione, K.-K. R. Choo, and H. Kim, "Memory-efficient implementation of elliptic curve cryptography for the internet-of-things," *IEEE Transactions on Dependable and Secure Computing*, 2018.
- [90] C. De Canniffire and M. Dunkelman, "A family of small and efficient hardware-oriented block ciphers," in *Cryptographic Hardware and Embedded Systems - CHES 2009*, ser. Lecture Notes in Computer Science, C. Clavier and K. Gaj, Eds. Springer Berlin Heidelberg, 2009, vol. 5747, pp. 272–288. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04138-9_20
- [91] L. Zhou, C. Su, W. Chiu, and K. H. Yeh, "You think, therefore you are: Transparent authentication system with brainwave-oriented bio-features for iot networks," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–1, 2017.
- [92] A. Cavoukian, "Privacy by design - the 7 foundational principles," https://www.iab.org/wp-content/IAB-uploads/2011/03/fred_carter.pdf, accessed 4 Feb. 2018.

- [93] N. Foukia, D. Billard, and E. Solana, “Pisces: A framework for privacy by design in iot,” in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, Dec 2016, pp. 706–713.