# Understanding the role of Key Encapsulation Mechanisms in Cryptographic Migrations

## Towards Cryptographic-Agility in IoT Systems Based on End-to-End Encryption Approach

**Alexandr Silonosov**

Department of Computer Science
Blekinge Institute of Technology

Licentiate Dissertation Series no. 2025:04

# Understanding the role of Key Encapsulation Mechanisms in Cryptographic Migrations.

Towards Cryptographic-Agility in IoT Systems Based on End-to-End Encryption Approach.

**Alexandr Silonosov**

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2025:04

# Understanding the role of Key Encapsulation Mechanisms in Cryptographic Migrations.

## Towards Cryptographic-Agility in IoT Systems Based on End-to-End Encryption Approach.

**Alexandr Silonosov**

Licentiate Dissertation in Computer Science

Department of Computer Science
Blekinge Institute of Technology
SWEDEN

Copyright pp Alexandr Silonosov
Paper 1 © Association for Computing Machinery, USA
Paper 2 © AIRCC Publishing Corporation
Paper 3 © Elsevier Publishing
Paper 4 © IEEE
Paper 5 © by the Authors (Manuscript unpublished)

Blekinge Institute of Technology
Department of Computer Science

Blekinge Institute of Technology Licentiate Dissertation 2025:04
ISBN 978-91-7295-498-4
ISSN 1650-2140
urn:nbn:se:bth-27713

Printed in Sweden by Media-Tryck, Lund University, Lund 2025

Media-Tryck is a Nordic Swan Ecolabel
certified provider of printed material.
Read more about our environmental
work at www.mediatryck.lu.se

MADE IN SWEDEN

*Dedication*

*5A 6D 46 74 61 57 78 35 49 48 4E 6A 61 57 56 75 59 32 55 67 63 48 4A 76 5A 6D 56 7A 63 32 6C 76 62 69 42 77 59 58 4E 7A 61 57 39 75*

# Abstract

The increasing data-security regulation and cyber-threats requires IT vendors to use new cryptographic tools or refactor live systems to encrypt existing data. One step in that direction is integrating an appropriate security protocol and cryptographic software library during the system design phase. However, it is not sufficient when it happens to migrate existing data to encrypted form in the live system on-the-fly, re-encrypt data to a different encryption standard, or have the data of the same origin but encrypted with different standards.

This thesis explores the new emergent area of cryptographic agility, which focuses on various challenges while adopting cryptographic migrations in live systems. We proposed End-to-End Encryption (E2EE) design for telemetry data security in two different applications: maritime surveillance and drone-management. We aimed to understand the role of Key Encapsulation Mechanism (KEM) cryptographic primitive in the data-security domain. The notion of crypto-agility constitutes a context-sensitive, activity-based perspective on data security. In this thesis, we aim at both understanding and exploring practical possibilities of this notion. We employ a mixed-methods approach to achieve our aim: Experimentation, Literature Review and Survey.

We have studied and applied quantum-safe KEM cryptographic primitives to simulate practical cryptographic migration in live IoT systems. We have shown the importance of KEM security properties and the performance of KEM primitives for telemetry data confidentiality. We proposed new crypto-agility values and trade-offs as decision making support tool for consumers of cryptographic technologies. Furthermore, we have employed systematization of knowledge to structure how different types of contributions developed various KEM notions, its influence on the standardization process, and presence in cryptographic software libraries over the last 40 years. The proposed approaches have been shown to be capable of explaining the role of KEM in cryptographic migrations and underlying properties of crypto-agility. This can facilitate domain experts in narrowing down the scope of analysis while achieving sufficiency for cryptographic migrations in live IoT systems based on end-to-end encryption protocols.

# Acknowledgements

I want to express my deep gratitude towards my lead supervisor, Lawrence Henesey , who have endorsed me with trust and gave me the opportunity to pursue this research in the context of EU-funded research projects: Connect2SmallPorts, DigiTechPort2030 and DecoInter. Especially, I would like to thank my supervision team Emiliano Casalicchio and Anton Borg for support and feedback, and express grateful appreciation to my colleague Oleksy Baranovsky for valuable discussions which enriched my scope of work.

# List of Papers

This thesis consists of five papers. The formatting of the published papers included in this thesis have been changed to achieve a consistent style in the thesis.

## Paper I

©2024 ACM. A. Silonosov and L. Henesey, "Telemetry data sharing based on Attribute-Based Encryption schemes for cloud-based Drone Management system" in Proceedings of the 19th International Conference on Availability, Reliability and Security, in ARES '24. New York, NY, USA: Association for Computing Machinery, Jul. 2024, pp. 1–8. doi: 10.1145/3664476.3670909.

## Paper II

©2024 AIRCC. A. Silonosov and L. Henesey, "Crypto-agility performance analysis for AIS data sharing confidentiality based on attribute-based encryption" presented at the 14th International Conference on Computer Science and Information Technology (CSIT 2024), Copenhagen Sept 21-22, 2024, AIRCC Publishing Corporation, 2024, pp. 193–212. doi: 10.5121/csit.2024.141714. urn:nbn:se:bth-26989

## Paper III

©2024 Elsevier. A. Silonosov and L. Henesey, "The role of key encapsulation mechanism in end-to-end encryption agility - a case for telemetry data confidentiality" presented at the 14th International Conference on Industry Sciences and Computer Science Innovation (ISCSI 2024), Porto Oct 29, 2024, Elsevier Publishing, 2024. urn:nbn:se:bth-27705

## Paper IV

©2024 IEEE. A. Silonosov and L. Henesey, “Towards cryptographic agility manifesto in end-to-end encryption systems: a position paper from the perspective of crypto-consumers,” in 2024 IEEE Conference on Dependable, Autonomic and Secure Computing (DASC), Nov. 2024, pp. 65–72. doi: 10.1109/DASC64200.2024.00015.

## Paper V

A. Silonosov, E. Casalicchio, L. Henesey, “SoK: Evolution of KEM role in cryptographic migrations for IoT systems based on E2EE approach” . Manuscript (preprint). urn:nbn:se:bth-27704

## Other Publications

Other research contributions that are related to this thesis but not included:

## Report I

A. Silonosov and L. Henesey. ”SECMAR Project: Enhancing Maritime Cybersecurity with Drones and Drone Management in the South Baltic Region”. 2023. Presented on 22nd Conference on Computer Applications and Information Technology in the Maritime Industries. COMPIT23. May 23, Drübeck, Germany, 2023.

## Poster I

©2024 ACM. A. Silonosov, O. Baranovskyi, and L. Henesey. ”Poster: Towards cryptographic agility in end-to-end encryption systems for computer generated telemetry data.”. Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications. p. 144-144. 2024. doi: 10.1145/3638550.3643624

# Author's contribution to the papers

According to the taxonomy of contributor roles, proposed by Larivière et al. [**Larivière**], we report the the division of scientific labor as following:

## Paper I

A. Silonosov: Conceptualization (equal), Data curation (lead), Investigation (lead), Methodology (lead), Visualization (lead), Writing - Original Draft Preparation (lead), Writing - Review & Editing (equal).

L. Henesey: Conceptualization (equal), Project administration (supporting), Writing - Review  Editing (equal).

## Paper II

A. Silonosov: Conceptualization (equal), Data curation (lead), Investigation (lead), Methodology (lead), Visualization (lead), Writing - Original Draft Preparation (lead), Writing - Review & Editing (equal).

L. Henesey: Conceptualization (equal), Project administration (lead), Writing - Review & Editing (equal).

## Paper III

A. Silonosov: Conceptualization (lead), Data curation (lead), Project administration (lead), Investigation (lead), Methodology (lead), Visualization (lead), Writing - Original Draft Preparation (lead), Writing - Review & Editing (equal).

L. Henesey: Conceptualization (supporting), Project administration (supporting), Writing - Review & Editing (equal).

## Paper IV

A. Silonosov: Conceptualization (lead), Data curation (lead), Project administration (lead), Investigation (lead), Methodology (lead), Visualization (lead), Writing - Original Draft Preparation (lead), Writing - Review & Editing (equal).

L. Henesey: Writing - Project administration (supporting), Review & Editing (supporting).

## Paper V

A. Silonosov: Conceptualization (lead), Data curation (lead), Project administration (equal), Investigation (lead), Methodology (lead), Visualization (lead), Writing - Original Draft Preparation (lead), Writing - Review  Editing (equal).

E. Casalicchio: Project administration (equal), Investigation (supporting), Methodology (supporting), Writing - Review & Editing (equal).

L. Henesey: Project administration (supporting), Methodology (supporting), Writing - Review & Editing (equal).

# Abbreviations

| | |
|---|---|
| AKE | Authenticated Key Exchange. |
| API | Application Programming Interface. |
| CA | Cryptographic-Agility. |
| CVE | Common Vulnerability Exposure. |
| DH | Diffie-Helman key-exchange algorithm. |
| E2EE | End-to-End Encryption. |
| ECC | Elliptic-Curves Cryptography. |
| HPKE | Hybrid Public Key Encryption. |
| IoT | Internet-of-Things. |
| KEM | Key Encapsulation Mechanism. |
| MLS | Messaging Layer Security. |
| PKC | Public Key Cryptography. |
| PKE | Public Key Encryption. |
| PQC | Post-Quantum Cryptography. |
| RSA | RSA cryptosystem. |
| TLS | Transport Layer Security. |

# Table of Contents

**Abstract** i

**Acknowledgements** iii

**List of Papers** v

**Abbreviations** ix

**Chapter 1 Introduction** 1

1.1 Overview . . . 1
1.2 Background . . . 3
1.3 Related Work . . . 6
1.4 Research Gaps and Questions . . . 8

**Chapter 2 Research Methodology and Results** 11

2.1 Research Methods . . . 11
2.2 Results . . . 13
2.3 Validity Threats . . . 17

**Chapter 3 Conclusion and Future Work.** 19

**Bibliography** 21

**Paper I Telemetry data sharing based on Attribute-Based Encryption schemes for cloud-based Drone Management system.** 33

1 Introduction . . . 33
1.1 Background . . . 34
2 Related work . . . 35
2.1 Attribute-Based Encryption . . . 35
2.2 ABE applications . . . 37
2.3 Data security compliance . . . 38
2.4 End-to-end encryption protocols . . . 39
3 ABE experiment for telemetry data protection . . . 40
3.1 Telemetry data components . . . 40
4 Results . . . 44
5 Discussion . . . 47
6 Conclusions and future research . . . 47
References . . . 48

**Paper II** **Crypto-agility performance analysis for AIS data sharing confidentiality based on attribute-based encryption.** **53**
1 Introduction . . . 53
1.1 Background . . . 55
2 Related work . . . 56
2.1 Attribute-Based Encryption . . . 56
2.2 ABE applications . . . 57
2.3 ABE code libraries . . . 59
2.4 End-to-end encryption protocols . . . 60
3 ABE experiment for AIS data confidentiality . . . 61
3.1 E2EE architecture for AIS data . . . 62
3.2 AIS data components . . . 63
3.3 Fine-grained access control for AIS data confidentiality 64
4 Results . . . 66
5 Discussion . . . 68
6 Conclusions and future research . . . 68
References . . . 69

**Paper III** **The role of key encapsulation mechanism in end-to-end encryption agility - a case for telemetry data confidentiality.** **75**
1 Introduction . . . 75
1.1 Background . . . 76
2 Related work . . . 77
2.1 Key Encapsulation Mechanism . . . 77
3 Key encapsulation experiment for telemetry data protection . 78
3.1 E2EE architecture for telemetry data . . . 79
3.2 Telemetry data model . . . 80
3.3 Fine-grained access control and forward secrecy for telemetry data . . . 81
4 Results . . . 83
5 Discussion . . . 84
6 Conclusions and future research . . . 85
References . . . 85

**Paper IV** **Towards cryptographic agility manifesto in end-to-end encryption systems: a position paper from the perspective of crypto-consumers.** **91**
1 Introduction . . . 91
2 Related work . . . 94
3 Present state of practice . . . 95
3.1 Cyber-security community and cyber-threats . . . 96
3.2 End-to-end encryption approach . . . 97
4 Cryptographic-agility Manifesto . . . 99
5 Conclusions and discussion . . . 102
References . . . 102

**Paper V** **SoK: Evolution of KEM role in cryptographic migrations for IoT systems based on E2EE approach** **109**
1 Introduction . . . . . 110
2 Related work . . . . . 112
3 Background . . . . . 114
3.1 Key Encapsulation Mechanism . . . . . 114
3.2 End-to-end encryption in IoT . . . . . 118
3.3 Challenges for cryptographic consumers . . . . . 119
4 Research Methodology . . . . . 121
4.1 Data sources and search strategy for RQ1 and RQ2 . . 121
4.2 Data sources and search strategy for RQ3 . . . . . 123
5 KEM state of the art . . . . . 124
6 Cryptographic software . . . . . 129
7 Conclusions and discussion . . . . . 134
8 Acknowledgements . . . . . 137
References . . . . . 137

# 1 Introduction

## 1.1 Overview

Applied cryptography serves as a bridge between a variety of theoretical cryptographic constructions and their practical implementations in real-life software systems. The term Key Encapsulation, hereafter KEM, is one example of how developments in theoretical Cryptology going through the standardization efforts settled as a new essential element in cryptographic software. KEM is one of the main cryptographic tools offered by Public Key Cryptography (PKC), and due to its versatile security properties, it can be used to implement data encryption and key agreement in data security protocols [1], [2], [3]. E2EE [4], [5] is a common approach employed to ensure the confidentiality, integrity and other security properties of user-to-user communication protocols such as in secure messaging [6], [7], in case when all data exchange occurs through the untrusted cloud server. The recent development of quantum computing poses a great threat [8], [9] to all security protocols such as Transport Layer Security (TLS), which rely on conventional PKC algorithms such as Diffie-Helman key-exchange algorithm (DH), RSA cryptosystem (RSA), and Elliptic-Curves Cryptography (ECC), invented more than 30 years ago. Standardization and regulation authorities around the world [10] initiated the transition process to quantum-safe data security standards [11] [12], known as Post-Quantum Cryptography (PQC) and studies reports that it would require massive resources and time [13] due to: 1) differences between existing crypto-algorithms and PQC, in terms of key and ciphertext size 2) uncertainty with the actual security levels of PQC systems under a conventional, i.e. non-quantum computing cryptanalytical attack. 3) reliability of proven cryptosystems, which security-levels attested by years of research i.e. having a persistent, long-term trade-off between two generations of cryptography (quantum-resistant and conventional, such as ECC, RSA).

Cryptographic-Agility (CA) hereafter crypto-agility or CA, has emerged into a research frontier as a satellite area of PQC phenomenon. The term was initially introduced by Bryan Sullivan [14] as the whitepaper presented at the Black Hat USA conference, proposing new approaches of dynamically adapting live IT systems to new cryptographic standards, as cyber-threats constantly evolve. There are a few definitions of CA in literature, and one thing they all have in common is a ability of live IT systems to dinamically adapt to new cryptographic technology [15] , [16]. For example an IT-system is considered crypto-agile, if it has a built-in capability to

switch from one encryption algorithm to another without replacing the surrounding infrastructure while maintaining its stability towards other systems [3]. Migration agility [17] refers to the ability to automatically switch between different cryptographic levels of security, per user or cloud tenant, depending on defined criteria, which requires the use of cryptographic metadata at the level of application data. Such an ability means that IT system should be able to store and process existing user data encrypted with the different schemes, or be able re-encrypt (i.e. migrate) existing data with new encryption scheme.

Although the efficiency of KEM is well proved in literature and specified in standardization, also exist alternative approaches for encapsulation which have their own terms, for example *integrated encryption*, and it is also present in standardization. As a result, cryptographic software varies for the implementation and terminology for the Public Key Encryption (PKE), which appear to be competing approaches and can make choosing difficult. On the other side the national governance authorities tend to increase the use of data encryption in various industries, which leads to intensive use of cryptography. Thus, the importance of crypto-agility will increase [18] over time, thanks to the values it adds to the IT systems, such as fast response to cyber threats, integrated threat intelligence and parametrization for data security.

An emergent hypothesis states that engineering of IT systems with intensive use of cryptography will benefit from crypto-agility values that can be formulated in terms of the general cyber-security domain. Considering those observations, we can well argue that the Key Encapsulation Mechanisms (KEM), a successor of PKE, play a broader role in context of entire cybersecurity domain, crossing different perspectives, including governance, operations, and engineering of cybersecurity controls, and its critical role in securing data and enabling secure communication, supporting cryptographic operations, as an essential part of cryptographic software Application Programming Interface (API). Such an interplay between governance via standards, operations via implementation, and engineering via cryptographic constructions suggest more research on CA to better cope with it's everyday challenges [13], [19], [20].

This thesis focuses on KEM for data security based on E2EE approach for distributed systems such as IoT. The aim is to improve the state of-the-art in this area: understanding, development and verification of cryto-agility for cryptographic migrations. The objective is to explore the role of KEM in the context of Cryptology, cryptographic software, cryptographic migrations and a broader perspective of cryptographic agility. We formulate the main goal in respect to the theme of cryto-agility as Understanding, to establishing a state of the art and state of practice in three facets of data-security domain: engineering, governance and operations.

The main contribution of this thesis is the study of three different KEM algorithms, development scenarios of cryptographic migrations in two different applications, and performing four-axis systematization of knowledge in the focus area.

The thesis contains five papers. In preliminary study [21], we provided a cyber-

security analysis of a cloud-based drone management system, performed static analyses for appliance software, identified the use of cryptography and main cyber threats for the data models, and proposed security controls by using STRIDE methodology. In Paper I, we proposed a telemetry data confidentiality model for the drone management systems. We study E2EE approach and cryptographic migration scenario between two KEM algorithms. In Paper II, we validated cryptographic migration scenario for cloud based system in the maritime surveillance domain. We study, the efficiency of quantum-safe KEM algorithm. In Paper III we generalized our approach for telemetry data confidentiality IoT systems. We study, the efficiency of cryptographic migration from RSA encryption algorithm to Crystals-Kyber. In Paper IV we study crypto-agility research frontier in context of three type of challenges for cryptographic users and proposed crypto-agility manifesto that might provide decision-support for organizations allocating resources towards transition to quantum-safe cryptography, assisting with examples of core values and trade-offs. Paper V, is a continuation of paper IV to systematize the knowledge about KEM in four dimensions: state of literature survey, state of the art strictly in cryptology science, state of practice across cryptographic software libraries, and state of standardization.

## 1.2 Background

The technology of PKC includes three main tools: key agreement, encryption, and digital signature. Well-known cryptosystems, such as DH, RSA, ECC are the first implementations of PKC and contain algorithms to implement one or all tools as shown in Table 1.1. To archive certain properties, some cryptosystems provides only a single tool. For example CRYSTALS-Kyber [22] provides only encryption also known as 'asymmetric encryption' or PKE in literature.

**Table 1.1:** Diffirent public key cryptosystems and available capabilities .

| Cryptosystem (year) | Key-exchange | Signature | Encryption |
|---|---|---|---|
| Diffie-Helman (1976) | + | - | - |
| RSA (1978) | + | + | + |
| ECC Elgamal (1985) | + | + | + |
| DSA/DSS (1991) | - | + | - |
| CRYSTALS-Kyber (2018) | - | - | + |

Within ten years, after the invention of the first cryptosystems, that contains all three tools, the malleability property was discovered [23] that if exist, it is not secure enough to use PKE to protect general-purpose data, as it was shown in specific type of attack well known as chosen plain-text attack and ciphertext attack (CPA, CCA). For example, ElGamal encryption [24] is unconditionally malleable, i.e. if given a ciphertext C1, C2 of some message M, an attacker can easily construct a valid ciphertext C1, 2*C2 of message 2*M [25], whereas M is a very large integer value.

If cryptosystem is a resistant to adaptive chosen ciphertext attack, it has become commonly known in cryptology under the abbreviations CCA, CCA1, CCA2 or A-CCA2 secure. This notion in cryptography was introduced by Galil and Naor [26], [27] and implies important property of cryptosysten, ciphertext non-malleability [23].

To overcome the weaknesses of PKE in practical use cases, a lot of effort in cryptology research was spent to propose new encryption *schemes* and *constructions* where a combination of algorithms have been used to encrypt arbitrary-length data. New construction archive properties of CCA with certain proofs and defined adversary model. Asymmetric encryption algorithm coupled with symmetric encryption evolved into a Hybrid Public Key Encryption (HPKE) scheme that is resistant to very powerful adversary model. Such a combination required to separate the different types of encryption in the protocol in terms of taxonomy, to highlight that assymetric encryption is used to protect technical parameters, and symmetric encryption is used to encrypt general-purpose data of any size [28], [29]. The term encapsulation has become established in scientific literature since 2001 [30], and since 2006 in standardization documentation and in open source cryptographic software. Since this modularity shift, the term key KEM is used in the context of security protocols or cryptographic implementations, and term PKE is used in the context of the Cryptology field. These changes supported the development of the modularity paradigm in cryptography.

The variety of *schemes* and *constructions* can also be classified by the underlying mathematical model that forms the basis of KEM and PKE. A variety of such models continued to grow for the last 20 years, and quantum threats led to an increased number of models and their variations. And since Applied Cryptography propagates many of them to be implemented in open-source cryptographic software it may be a challenge for the users of such software to switch between alternatives in case of unpredicted cyber threat. It is normal practice if a cryptographic software library implements concurrent cryptosystems having the same tools, i.e., asymmetric encryption, but have different properties due to different mathematic foundations. As shown in Table 1.2. first invented cryptosystems contain three algorithms for asymmetric encryption implementation and a key pair for each participant, including two keys, public and secret keys. The other cryptosystems contain additional algorithms and a different number of keys per participant, which make them incompatible with each other in terms of cryptographic migration.

In case-study [21], we studied a Drone Management IoT system. Data-security analysis reveal the main assets in the system and the use of cryptography in software assets. We applied STRIDE risk assessment framework to identify the possible gaps in data security and proposed data encryption for security controls as mitigation. Studies by Shelley [37] and Yaacoub et al. [38] also confirm the gap of using cryptography as the main security measure against 50% of cyber-attacks.

The E2EE approach assumes client-server architecture, and clients encrypt data on the application level before sending it to server. Data is decrypted only inside the client application, on the intended recipients' hosts. Data on the server is always

**Table 1.2:** Comparison of Encryption capabilities in different public key cryptosystems by number of algorithms and number of keys per participant.

| Cryptosystem | Mathematic | Other algorithms in addition to Generate Key, Encrypt, Decrypt | Keys |
|---|---|---|---|
| RSA [31] | Integer fact. | + | 2 |
| ECC Elgamal [32] | Discrete log. | + | 2 |
| Identity-Based Encryption (IBE/ABE) [33] | Pairings. | Generate Params | 1..n |
| Proxy Re-encryption [34], [35] | Pairings. | Delegate, Re-encrypt | 2..n |
| Threshold encryption [36] | Pairings, Lattices | Partial Decrypt, Combine | 2..n |
| Homomorphic encryption | Lattices. | Generate Params, Add, Mult, Refresh, Check | 3..n |
| CRYSTALS-Kyber [22] | Lattices. | + | 2 |

encrypted — not just in transit and at rest — because the server never has access to decryption keys. The server does not have access to the cryptographic keys necessary to read or invisibly alter the message [39]. One of the earliest messenger applications to implement E2EE was ”Silent Circle,” founded in 2012 by Phil Zimmermann, Jon Callas, and Mike Janke. WhatsApp Messenger, for example, introduced E2EE in 2014, by using the Signal protocol library. Since then, E2EE application and adaptations have been part of the research frontier in Applied cryptography [6], [40], [7]. Cohn-Gordon et al. [41] described the evolution of Signal protocol's main components, which are the Double Ratchet algorithm, prekeys, and a triple Elliptic-curve Diffie–Hellman (X3DH) key agreement, and proved that under a set of assumptions and threat model, several standard security properties are definitely satisfied among all declared properties such as confidentiality, integrity, authentication, participant consistency, destination validation, forward secrecy, post-compromise security, causality preservation, message unlinkability, message repudiation, participation repudiation, and synchronicity.

Standardization authority NIST, USA, designed the time scale for cryptographic algorithms and keys referred to as *security lifetime* and matrix security strength. For security lifetime data owner needs to define the length of the required confidentiality period for the data in years (referred to as 'security life of the data'), mapped to an absolute point in time in the 2031 year, after which the algorithms with low-security strength will be prohibited to use. The matrix 'security strength' sorts algorithms by the maximum possible key length. These can be used as a tool that calculates for how long and why the '*security lifetime*' of the algorithm should be longer than the crypto period of a key, with a binding to a constant date when the measurements will expire. However, at the same time, NIST assumes that the security strength provided by an algorithm or key may be reduced or lost completely, for example, the algorithm or key length used may no longer offer adequate security because of improvements in computational capability or cryptanalysis. Therefore, the cryptoperiod may be unexpectedly reduced to zero at some point in time. To conclude, to limit the risk of compromise, the length of the associated lifetime will decrease in the future. In 2013, MITRE organization started to document common tactics, techniques, and

procedures used by cyber-criminals to penetrate enterprise networks [1]. This involves detecting, recognizing, and naming all the tiny parts of a cyber-attack. Later, this was developed into an Adversarial Tactics, Techniques, and Common Knowledge (ATTCK) framework that hierarchically maps over 230 attack techniques into 14 tactics. Nowadays, all cyber-attacks are organized as a consequence of stages and are considered successful if all stages are executed successfully. ATT&CK can be used to identify the attack pattern and thus understand which defense mechanism to use for the next stage, depending on the detected attack procedure. According to ATT&CK, preventing an entire cyber-attack is enough to stop one stage. ATT&CK can be used with another objective to estimate if a particular defense strategy will work in a specific attack scenario and calculate a gap in security controls. In ATT&CK, Techniques can be understood as entry points in systems where weaknesses are actively searched or attackers can perform misuse. ATT&CK includes a few hacking techniques related to weaknesses in encryption, and two techniques use encryption to commit harm to data.

## 1.3 Related Work

In this thesis, we work with the three distinct facets of the data-security domain [42], [43] and we first propose to name them as governance (G), operations (O), and cryptography engineering (E), where (E) is the source of data security technology [44]. Ott et al. [17] proposed eighth modalities which represents the basis of CA area and they serves as evidences of all three perspectives. KEMs have been studied and applied in these perspectives such as standardization documentation, cryptographic software implementations and definition in cryptology science. We structure the related work across these three perspectives since each one has specific aims, criteria and taxonomy for KEM and E2EE technologies.

Security research on KEM in facet (E) holds an excessive body of works at Cryptology tier-1 venues, mainly supported by International Association for Cryptologic Research (IACR), and venues that covers general aspects of cyber-security with the research topics such as *secure authentication and key exchange in IoT-enabled application*, *quantum secure authentication and user privacy in IoT*, or *enhancing authentication for Smart Grid Communications*, *usability and security of cryptographic software API*. To survey KEM we decided to target venues reported in Table 1.3. Since this thesis is primarily concerned with the use of cryptographic software, we give only a brief introduction into KEM in Cryptology research field and survey constructions and schemes that was included into cryptographic software or standardizations. The scoping criteria for selected ten venues is motivated by the objective to analyze KEM contribution only in *engineering* facet, out of 1280 existing papers in total, indexed in Scopus, mentioning terms *kem* or *incapsulation*.

[1] https://mitre.org

**Table 1.3:** Tier-1 venues with research publications in the field of Cryptology by umbrella organozation and number of occurrence if conference.

| Venue | Org. | Occurence |
|---|---|---|
| Crypto | IACR | 45 |
| Eurocrypt | IACR | 44 |
| Asiacrypt | IACR | 32 |
| Public Key Cryptography | IACR | 22 |
| Theory of Cryptography | IACR | 21 |
| Journal of Cryptology | IACR | 2 |
| Computer and Communications Security | ACM | 32 |
| Symposium on Security and Privacy | IEEE | 46 |
| USENIX Security Symposium | USENIX | 34 |
| Conference on Post-Quantum Cryptography, PQCrypto | IACR | 16 |

The term KEM was first introduced by Shoup [30] , followed by work of Bellare et al. [29] and Cramer et al. [45] which proposed the use of PKE in combination with hash function and symmetric encryption algorithm (cipher) to mitigate chosen ciphertext attacks. It took time to understand that it is not secure enough to use the PKC Encryption by itself, and finally it evolved into a Hybrid Publick Key Encryption (HPKE) scheme that is resistant to strong security attacks Adaptive Choosen Ciphertext Attack (A-CCA2). Key encapsulation mechanism is the main component of HPKE (RFC-9180, 2024). Today, KEM has been well studied, defined in various data security standards (FIPS-203), and applied in different communication protocols [46].

There exists a considerable number of studies that provide a comprehensive, and systematized knowledge (SoK) of the interplay between state-of-the-art (E) and state-of-practice (0) for CA, due to its emergent nature. Qualitative analysis of existing research of KEMa and PKE security properties [47], [48], [49], [50], including PQC KEM [2], [51], [52], or role of PKE in E2EE content moderation[39]. Research on empirical analysis of open-source cryptographic software implementations [53], [54] including the state of verified implementations [55] conclude still exist multiple challenges in the transition process, and certain limits still needs to be studied or interoperability issues related to hardware such as embedded systems. To our knowledge exist also very limited number of case-studies for real-world cryptographic transitions at enterprise-level [19] to demonstrate the necessity to increase research the operational perspective of data-security. Interview studies [44] , [56] that review the cryptographic ecosystem and traverse the flow from the scientific proposal up to the product with cryptography from an operational perspective, i.e., implementations. According to the identified elements and roles in cryptography ecosystem on Figure 1.1 the interview study by Fischer et al. [44] discovered four types of engineering tasks namely, theory, design, application, analysis; The interview confirmed the role of Applied Cryptography across multiple perspectives, but also demonstrates obvious gaps such as Product Designers should still use or comply with Cryptographic Standards while using Secure Implementations provided by Applied Cryptographers who created Standards, assuming that just Implementations is not yet capable to assure certain aspects of data security; at the same time neither of Applied Cryptographers

or Cryptanalists involved into verification of data-security in Product; But Product Security Analyst is not using Standards to verify if Secure Implementations integrated into product is still comply to standards; The Interview also identified a vague role of Industry Stakeholders between Product and Standard, which probably means increasing operational facet that still needs to be identified, as for example to align PQC threat factor and general cyber-security threat actors in the ecosystem.

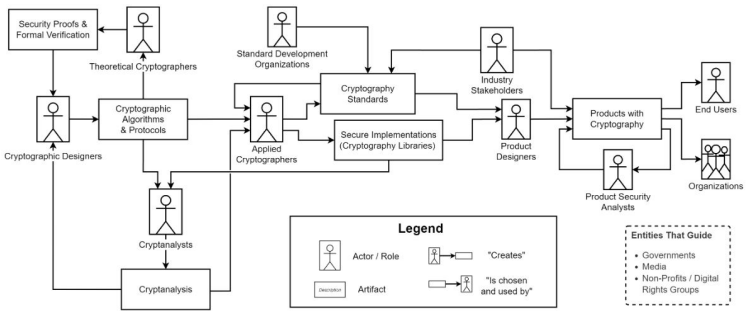


**Figure 1.1:** Cryptography ecosystem and the path of bringing cryptography from paper to product. Picture from paper [44] (c) USENIX, 2024

Ding et al. [57] proposed Application Layer Encryption integration into eHealth Framework Platform. Authors of RFC-9420 [58] proposed Messaging Layer Security (MLS) protocol to be integrated into higher abstraction layers of software architecture in particular as E2EE approach of instant messaging. The protocol defines how cryptographic primitives and constructions such as Authenticated Key Exchange (AKE) and key ratcheting should be integrated with the authentication and data delivery layers of the application.

## 1.4 Research Gaps and Questions .

This thesis focuses on studying and developing approaches for cryptographic migrations in E2EE systems, which includes the problems of CA. Such problems are often facet-specific or domain-specific. Understanding cryptographic technology and modeling the cryptographic migrations is the core of CA in the case of Internet-of-Things (IoT) system based on E2EE approach. One of the challenges in the CA problems relates to the operational facet of Applied cryptography. Existing IoT systems as a consumers of cryptographic software needs to respond on evolving cyber-threats, improve compliance and adjust technological stack to periodically changing cryptographic software. Applied cryptography is vital for addressing evolving data-security threats by proposing data-security tools , but significant research gaps remain at the intersection of applied cryptography and cryptographic consumers. These gaps arise from the differing posture, and nature of the two facets, underscoring the need for

a justified approach to robust and adaptable solutions designed for intensive use of cryptography.

In the data-security domain, robust frameworks like ISO/IEC standards and FIPS provide guidance on cryptographic implementations, yet these standards often need to be first adopted by cryptographic users, in operational facet, to account for the specific needs of IoT. For example IoT environments, which are highly resource-constrained, demand lightweight cryptographic solutions and tailored compliance mechanisms. In addition software development technology stack contains multiple programming languages, operating systems and platforms. The lack of alignment between cryptographic software and IoT-specific requirements creates a gap in operational and governance perspectives, particularly when transitioning to post-quantum cryptography. The need for dynamic updates to cryptographic algorithms and protocols in response to emerging threats highlights operational gaps. While software industry has developed tools for secure software updates and cryptographic metadata management, IoT systems often lack the solutions to implement such changes seamlessly. Data security protocol usually solidifies the data model in IoT system, making it challenging to ensure real-time data re-encryption with new algorithm on a big scale. This gap creates vulnerabilities when outdated cryptographic algorithms remain in use due to operational constraints. From an engineering perspective, there is a mismatch between the cryptographic frameworks used in general cybersecurity applications and those needed for IoT cloud systems. Cloud platforms like AWS, Azure, and Google Cloud rely on advanced key management systems (e.g., CloudHSM and KMS) to support scalable encryption. However, IoT systems require lightweight end-to-end encryption (E2EE) protocols or application–layer encryption that integrate seamlessly with these cloud services. The absence of engineering solutions that balance cryptographic agility with the performance constraints of IoT devices is another research gap. Both domains struggle with managing cryptographic artifacts like keys, algorithms, as application level metadata. In IoT cloud systems, the dynamic nature of devices and connections requires efficient metadata frameworks to track and adapt cryptographic configurations. While the data-secyrity domain has approaches for this purpose, they often lack the scalability and simplicity needed for IoT environments, or not represented at all in cryptographic software API, leaving a gap in cryptographic artifact management that hinders agility.

Combining all three facets of data-secyrity in a single framework for informed cryptographic migration for real-life IoT systems is difficult [19]. To tackle the identified gaps, we define the main research questions addressed by this thesis as follows:

- RQ1. What are the challenges that influence the nature of cryptographic code refactoring in cryptography-intensive system and how do they influence?

  This question investigates the role of cryptography in data-security of cloud-based IoT system, that we initially review in a case study [21], Then we ad-

dressed following research question in Papers IV and V.

- RQ2. How do the challenges discovered in addressing RQ1 drive the evolution of KEM approaches and standards?

  The reason for studying this question is that we need first to understand the taxonomy and differences between PKE, Key Exchange (Establishment) vs KEM.

  This research question is addressed in Papers I, II and V.

- RQ3. What are the structural and security differences between traditional PKE, KEM and PQC KEM, and how do these differences influence software architecture, performance and resilience to cryptographic attacks?

  The reason for studying this question is to identify core properties of PKE, PQC KEM and compare different KEM according to the criteria. This research question is addressed in Papers II, III.

- RQ4. What are the key challenges and considerations in adopting cryptographic migrations from traditional PKE to the next generation of KEM and it's approaches in E2EE systems?

  The reason for studying this question is that we need to identify the modalities of CA and the role of KEM in each one. Based on the study conducted in Paper III, we have proposed generalized approach for cryptographic migration in IoT system. This research question concerns both paper IV and V.

# 2 Research Methodology and Results

## 2.1 Research Methods

In this thesis, we have designed and performed three experiments and employed a Systematic Literature Review (SLR) method to design multi-perspective Systematization of Knowledge (SoK) study. Experiment method was used in papers I, II, III to study the performance of the cryptographic migration based on diffirent KEMs in two diffirent IoT systems and then data-security in generalized abstract IoT system.

The input for Experiment represents the data and system requiring cryptographic migration. This could involve encrypted files, databases, and access model in IoT system. The central process of cryptographic migration, refers to the act of transitioning data securely from one cryptographic state to another. The output is the result of this process, which could be measured in terms of performance and security properties. The experiment is influenced by two sets of factors. Controllable factors include variables that can be adjusted during the experiment, such as the choice of cryptographic library, the type of KEM, and security parameters like encryption key length. These factors can be configured to observe their impact on the migration process. Uncontrollable factors are external or inherent constraints that affect the process but cannot be directly adjusted. Examples include the data model, which refers to the structure and format of the input data, and the security model, which pertains to the underlying security protocols or framework in the system. The experimental process involves defining objectives, configuring controllable factors to simulate different scenarios, for example computational capacity, and assessing the impact of uncontrollable factors. The output is measured using metrics such as migration time, post-migration data model adjustments, and security levels. The results are analyzed to identify optimal configurations for cryptographic migration and evaluate the system's resilience under various constraints. This approach enables a systematic study of cryptographic migration processes.

Systematic literature reviews, used to aggregate the experiences gained from a range of different studies in a specific domain with the main objective of analyzing their scientific contribution, while having the aim to answer a specific research question [59].

The Systematization of Knowledge (SoK) is a recognized research methodology, particularly prevalent in computer science and cryptology. SoK papers aim to evaluate, systematize, and contextualize existing knowledge within a specific domain, providing comprehensive analyses that guide future research directions. The IEEE Symposium on Security and Privacy has been a pioneer in promoting SoK papers since 2010 [60]. According to Journal of Systems Research SoK offer in-depth evaluations of current research landscapes, empirical analysis of open-source software, identify gaps, and suggest areas for research directions [1]. Exist evidence [61], [56], [40], [7] on how open-source cryptographic software influence data-security technology in a quasi standardization way, highlighting the nature of industry-driven innovations, by practitioners. In this sense, SoK method can be used not only to extract empirical evaluation of novel techniques but also evidence such as: code repository properties, cyber-threat statistics Common Vulnerability Exposure (CVE) [2] and software defects identification [62], [63], [64], [65].

Exist also the body of research of cryptographic software that study security properties as inherited codified requirements [66], [67], [68], [69], including vulnerabilities exposure, cryptographic API complexity and miss-use[70], [71], [72], [20], [73], [74]. In such studies the Survey is the most often used research method and latest studies in this area has been published also in Applied Cryptography venues. Since the discipline of Empirical Software Engineering also focus on software source code, using metrics, such as complexity, quality and defects [75] [76], arise conditions for interdisciplinary area. And this explains our choice for knowledge systematization. In our work the primary goal of SoK is to convert recognized and tacit knowledge to recognized and codified knowledge [77] in the field of Cryptology science. Guo et al. [78] propose a novel classification of contributions in Cryptology, which explains the nature of incremental development in both applied and theoretical cryptography. According to Guo's view, exist two philosophic approaches in cryptographic research 1) research can be centered around the problem, so then it should be capable to scope on a variety of technical routes that lead to a solution or 2) it can be focused on the specific route, clearly identify its value, advantages and propose a novelty on this route, such as distinct details of the implementation of the specific solution(s). Guo propose to classify contributions into three major categories which are then divided into six routes, each of which contains several improvement areas (from 2 to 6). By using this classification Guo identified 40 technical routes. In Paper V, we employed this classification to define our SLR protocol, review and systematize major contributions related to PKE and KEM development in Cryptology. We mainly focused on contributions in categories of constructions and definitions, according to Guo. Then we summarized the existing evidence of the state of practice for cryptographic software and regulation authorities related to KEM technology.

[1] https://www.jsys.org/type_SoK/

[2] https://www.cve.org/CVERecord/SearchResults?query=cryptography

Developing new approaches to design efficient cryptographic migrations must be accompanied by reliable evaluation methodologies. The number of CPU cycles required to perform a cryptographic operation on a given CPU is frequently used in Cryptology to compare alternative approaches. In computer science the time in milliseconds on a given CPU and cryptographic software is used for evaluation. The later evaluation method is more practical in our case since we are mainly focusing on contexts where CPU performance is allocated by cloud platform in terms of Single Thread Performance rating [79], [80], [81]. In the first case the Cryptology community it is common to use the SUPERCOP (System for Unified Performance Evaluation Related to Cryptographic Operations and Primitives) toolkit [3] to conduct both the experimentation and evaluate the results, since SUPERCOP is the part of eBATS (ECRYPT Benchmarking of Asymmetric Systems) [4] project , a publicly available and maintained performance results of public-key systems benchmark tests. eBATS contains results for benchmarking KEMs on many systems and implemenations, and results contains CPU cycles (RDTSC instruction for Intel and AMD, or RDPMC since 2021) per each cryptographic operation and sizes of key material and ciphertext. Since the cycle-based measurements may depend on CPU-core run-time load balancing, power optimization, heterogeneous cores, even subject to interpretation ”CPU clock cycles” vs ”CPU cycles”, SUPERCOP toolkit mitigation measures do not guarantee full discrimination between events caused by different processes in the life system. As the mitigation the toolkit itself recommend to reproduce ideal experimentation environment i.e. idle system, which property of is not included into results.

## 2.2 Results

In this section, we provide an overview of the studies conducted in scope of this thesis. The derivation of research objectives from the gaps and the decomposition into RQs and Papers are visualized in Figure 2.1.

In Paper I we studied the application of ABE for telemetry data encryption. First we surveyed the existing studies of ABE applications in real-world systems, separated by domains: healthcare and logistics. And state-of-practice for ABE in standardization. We proposed secure access to telemetry data in untrusted cloud environment, where access is enforced cryptographically. The validity of the results is investigated by comparing efficiency between two KEM algorithms RSA and ABE.

In Paper II we studied PQC KEM algorithm Kyber. We proposed AIS data confidentiality approach based on end-to-end encryption. We identified cryptographic metadata for OpenABE library, which highlights the need of necessary abstraction layer to access metadata in the ciphertext. Opposite to it, Kyber KEM metadata in-

[3] https://bench.cr.yp.to/supercop.html

[4] https://bench.cr.yp.to/results-kem.html

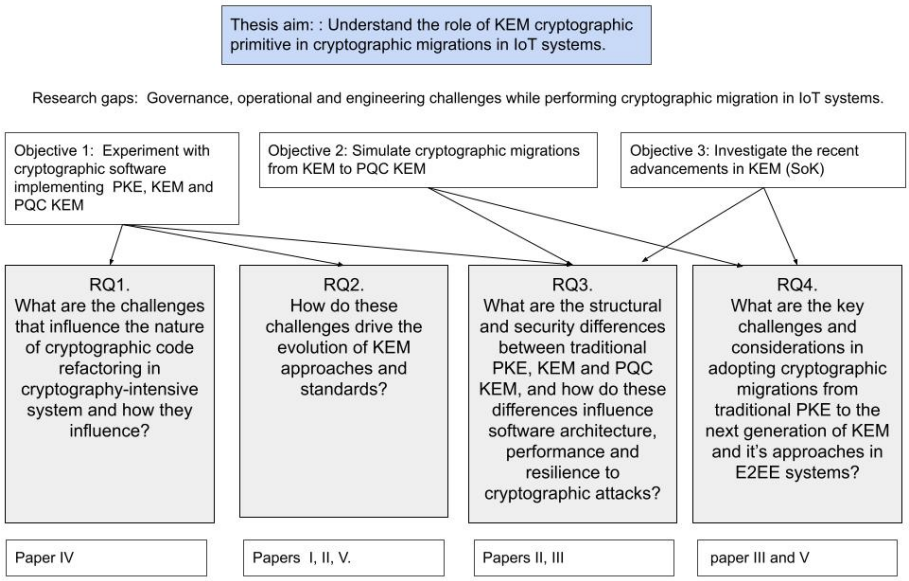


**Figure 2.1:** Mapping thesis contributions with the aim, gaps, objectives and research questions to papers.

cludes only public key bytes assuming the metadata information will be coded directly by the application abstraction layers. Both libraries include integrated symmetric encryption abstraction (SYM) that allows the encryption of arbitrary data of any length alongside an encapsulated session key. Kyber KEM was implemented in a stateful model (ST) to maintain the session key across multiple crypto operations. Our study provides strong evidence supporting the feasibility and efficiency between two types of KEM of the proposed approach for cryptographic agility in E2EE system.

In Paper III, we proposed a generalized approach for telemetry data confidentiality based on PQC KEM E2EE.

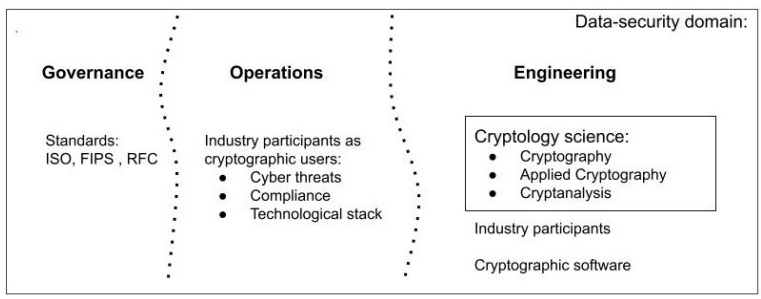


**Figure 2.2:** Relationship between science, technology and regulators in three facets of data-security domain.

In Paper IV we initiated a study of CA by using systematization of knowledge in literature and practice. Our generalized assumption formulated as main CA values and trade-offs as crypto-agility manifest.

In Paper V we continue study of paper IV, systematized the SoTA for KEM , PKE and SoPA in open-source cryptographic software. Our systematization work identified there distinct perspectives (facets) for data-security domain, and we propose to call it respectively as Governance (G), Operations (O), and Cryptography engineering (E). The relationship between these perspectives can be illustrated in Figure 2.2. The study identified the objectives of each facet of KEM technology by using industry artifacts such as state-level regulation, cryptographic software, industry-specific standardization, cyber-threats and scientific literature.

With the help of our results we can answer the research questions of this thesis as follows.

*RQ1. What are the challenges that influence the nature of cryptographic code refactoring in cryptography-intensive system and how do they influence?*

Based on the results of Paper IV, we identified three types of challenges: regulatory, community-driven, and those arising from cyber-threats. All challenges may significantly evolve during the system's lifetime, requiring the adoption of data-security protocols, which may require existing data to be encrypted or re-encrypted with different algorithms. National regulations may evolve to require the use of encryption for new data types or introduce the mandatory use of stronger cryptographic primitives. The Cybersecurity community identifies vulnerabilities and weaknesses in live systems, and the development of software fixes may lead to a redesign of security protocol and changing cryptographic suppliers or libraries. The community is represented by ethical hackers, scientific researchers, and professionals. Cyber-crime incidents or cyber-attacks may lead to the same consequences but require immediate fixes.

*RQ2. How do the challenges discovered in addressing RQ1 drive the evolution of KEM approaches and standards?*

The answer for this question is two-fold. First, the discovery of the malleability property of ciphertext in 1991 led to the separation of the encapsulation tool from the encryption tool and the introduction of new property for asymmetric encryption algorithms known as indistinguishability for chosen-ciphertext attacks. Second, regulatory compliance, since 2017, started using these terms and in 2023 introduced a new generation of encapsulation tools into standards. New tools are resistant to quantum computers, and specifications also propose how to implement combinations of conventional algorithms with new generation KEM to ensure resistance to both threats : quantum and non-quantum.

*RQ3. What are the structural and security differences between traditional PKE, KEM and PQC KEM, and how do these differences influence software architecture, performance and resilience to cryptographic attacks?*

In papers I, II, III we identified that the given classes of asymmetric encryption differ in terms of mathematical problem hardness used to construct the main calculation, which in Cryptology is referred to as a *trapdoor function*. The difference in security is determined by the presence of indistinguishability properties between chosen-plaintext attacks and chosen-ciphertext attacks. Currently, there is no algorithm for quantum computers that can efficiently solve *trapdoor functions* that being used in a new generation of KEM, and therefore, they are considered quantum safe.

The majority of cryptographic software libraries, with time, accumulated a set of algorithms for asymmetric encryption and encapsulation and contain a new set of abstraction layers to use a combined approach but still lack parametrization capabilities. The analysis of cryptographic software also identified that their architecture does not include metadata or the ability to determine what method was used to encrypt the data. A new generation of KEM is 50 times more efficient than conventional KEM for encryption time but less efficient for communication costs: the sizes of ciphertext and encryption keys.

*RQ4. What are the key challenges and considerations in adopting cryptographic migrations from traditional PKE to the next generation of KEM and it's approaches in E2EE systems?*

The results in Paper V indicate that changing cryptographic software library for live IT system is challenging task due to the identified constraints:

- Compliance with the data-security standard.
- Cryptographic backend dependencies
- Available bindings to different programming languages and technological stacks.

The following considerations influence the cryptographic migration for the data in live IT system based on E2EE approach is:

- Efficiency of KEM algorithms is different for computations and communication costs.
- The approach based on Hybrid KEM, i.e., KEM Combiners, creates a necessity for cryptographic inventory on a high layer of software abstractions; stored as metadata for cryptograhic parametrization. The X-Wing KEM demonstrated how metadata can contribute to support redundancy strategy.

- Security properties such as forward secrecy require formal verification for each data element and the use of additional tools within the process of software engineering process. An alternative could be the integration of Application Layer Encryption or additional high-level security protocol represented by separate software library.

## 2.3 Validity Threats

In our studies, there are some issues that can threaten the validity of the results and they should be considered before applying the proposed techniques in IoT systems and performing the validation. The E2EE designs proposed in Paper I, II, III could be biased due to misconceptions about the desired access control model and security properties of cryptographic protection. To overcome this risk, this chapter identifies the potential validity threats in the proposed concepts. We report threats to validity in taxonomy by Cook and Campbell [82].

Construct validity refers to the extent to which the results of a study reflect the theory or the concept behind [82]. According to Wohlin et al. [83] ”Construct validity concerns generalizing the results of the experiment to the concept or theory behind the experiment”. The experimentation results in terms of the performance, can be affected by the potential faults that may happen in the implementation either because of CPU virtualization peculiarities or lack of parameter tuning. For decreasing the effect of CPU virtualization, the validity of the implemented experiments are tested on both physical and virtual CPU, while cheking consistency with ingegrated CPU performance indicators multiple times. In terms of cryptographic artifacts measurements the validity threats are low since the results can be validated with eBATS database, described in Chapter 1.5.

The other validity threat, which can occur while performing the evaluation and validation, targets both the internal and external validities. Internal validity ensures that the observed relationship between the treatment and the outcome is due to a causal relationship and is not because of an uncontrolled factor, so that experimental design represent true population, and results are not biased with measurement errors. External validity refers to the ability of generalizing the results of the study to other domains, times or places [82]. In such situation generalizing the results of the treatments to the whole E2EE approach is realistic if evaluated on multiple data models for different access scenarios, ensuring a realistic representation of the studied population in both the experiment and the validation. In paper III the proposal was validated in a more generalized setting.

In this thesis, we also may choose to discuss the reliability aspect of our results, i.e., the extent to which a study and its results can be replicated by other researchers [84]. To maximize the replicability of our experiments in papers I, II, and III, we used well-known, open-source, and maintained cryptographic software libraries available

as repositories so that the exact version of the software can be easily instantiated for execution. During knowledge systematization in Paper IV and V, we analyzed qualitative data gathered through internet search and literature search. To mitigate the potential threats to reliability due to researchers' bias the three first authors ensured that inferences contained in the resulting data were appropriately discussed and agreed upon.

# 3 Conclusion and Future Work.

In IT systems based on cloud computing, we observe a tendency to increase the volume of data processing due to new data sources, for example, telemetry data. At the same time, this leads to an increased number of parties and stakeholders that participate in the data processing, which includes systems vendor employees, customers, and the cloud computing supply chain. The variety of stakeholders increases the cyber-attack surface, and cyber-threat intelligence demonstrates that data and AI systems, which are also cloud-based, can unexpectedly become part of a cyber threat with large-scale consequences. To tackle these challenges, governments of different countries are introducing new regulatory compliance and standards in the domain of data security and data sovereignty, which, in particular, demands increased use of cryptography in IT systems. These phenomena served as the basis for our review of how the intensive use of cryptography and encrypted data poses various challenges for IT vendors in existing systems.

Each chapter contributes to the notion of CA identifying the role of KEM in cryptographic migrations through multiple facets of data security: Governance, Operations, and Engineering. We can summarize the contributions in the following folds:

- We have proposed novel examples of cryptographic migration experimentation for new types of data in IoT systems.
- We have demonstrated the interplay between three facets of data security by applying specific KEM standards in three cryptographic software libraries in our experiments.
- We have proposed a cryptographic-agility manifesto as a hypothesis for the emergent area of cryptographic agility, by identifying the values and trade-offs of cryptographic consumers.
- We have explored the notion of cryptographic agility employing the SoK research method through a novel mapping structure. item

The specific role of KEM in each perspective constitutes the emergent area of CA. To bridge identified gaps, more research is needed, and possible directions are as follows.

One direction is to design scalable frameworks for real-time cryptographic updates and algorithm transitions for concrete IoT system, considering the heterogeneous nature of its technological stack. For this work, we are considering designing and performing new experiments for digital-twin IoT, which will include new cryptographic software implementing lightweight E2EE protocols that also integrate with cloud-based key management services. Working with interoperable metadata management systems for cryptographic artifacts integrated with Application Layer Encryption of digital-twin system will provide additional metrics for efficiency measurements for evaluation and improve generalizability of results.

The SoK results also identified a new class of cryptographic software libraries based on high-assurance principles and memory-safe programming languages such as Rust and F* referred to in literature as *formally verified* implementations and *computer aided cryptography* [55]. This work will also contribute to understanding how *formally verified* principles influence cryptographic agility. One possible direction towards agility in *formally verified* cryptography is the experimentation with HACL [85] software, which will expand the notion of cryptographic agility with new modality. Another direction would be to focus on interplay between governance and operational facets with the objective to review what parts of *computer aided cryptography* can be reflected in compliance aligned with cryptographic agility. This will improve the efficiency of operational facet in terms of cryptographic software updates, metadata management, and dynamic reconfiguration of data security protocol in live IoT system.

# Bibliography


[1] N. Bindel, J. Brendel, M. Fischlin, B. Goncalves, and D. Stebila. *Hybrid Key Encapsulation Mechanisms and Authenticated Key Exchange*. Publication info: Published elsewhere. Major revision. 10th International Workshop on Post-Quantum Cryptography (PQCrypto 2019). 2018. URL: https://eprint.iacr.org/2018/903 (visited on 11/19/2024).

[2] M. Harmalkar, K. Jain, and P. Krishnan. "A Survey of Post Quantum Key Encapsulation Mechanism". In: *2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*. Jan. 2024, pp. 141–149. DOI: 10.1109/ICMCSI61536.2024.00028. URL: https://ieeexplore.ieee.org/abstract/document/10493946 (visited on 01/01/2025).

[3] K. G. Paterson. "The Cyber Security Body of Knowledge v1.1.0, 2021". In: Section: Applied Cryptography. University of Bristol, 2021. URL: https://www.cybok.org/.

[4] B. Hale and C. Komlo. *On End-to-End Encryption*. Publication info: Preprint. MINOR revision. 2022. URL: https://eprint.iacr.org/2022/449 (visited on 04/02/2024).

[5] S. Scheffler and J. Mayer. "SoK: Content Moderation for End-to-End Encryption". en. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (Apr. 2023), pp. 403–429. ISSN: 2299-0984. DOI: 10.56553/popets-2023-0060. URL: https://petsymposium.org/popets/2023/popets-2023-0060.php (visited on 10/10/2023).

[6] K. G. Paterson, M. Scarlata, and K. T. Truong. "Three Lessons From Threema: Analysis of a Secure Messenger". en. In: 2023, pp. 1289–1306. ISBN: 978-1-939133-37-3. (Visited on 03/14/2025).

[7] T. von Arx and K. G. Paterson. "On the cryptographic fragility of the Telegram ecosystem". In: *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 2023, pp. 328–341.

[8] C. Näther, D. Herzinger, S.-L. Gazdag, J.-P. Steghöfer, S. Daum, and D. Loebenberger. *Migrating Software Systems towards Post-Quantum Cryptography – A Systematic Literature Review*. arXiv:2404.12854 [cs]. June 2024.

DOI: 10.48550/arXiv.2404.12854. URL: http://arxiv.org/abs/2404.12854 (visited on 12/20/2024).

[9] C. Gidney and M. Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". en-GB. In: *Quantum* 5 (Apr. 2021). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 433. DOI: 10.22331/q-2021-04-15-433. URL: https://quantum-journal.org/papers/q-2021-04-15-433/ (visited on 01/21/2025).

[10] "Post Quantum Government Initiatives by Country and Region". In: *gsma.com* (Oct. 2024). URL: https://www.gsma.com/newsroom/post-quantum-government-initiatives-by-country-and-region/.

[11] J. Haney, M. Theofanos, Y. Acar, and S. S. Prettyman. *Organizational views of NIST cryptographic standards and testing and validation programs*. en. Tech. rep. NIST IR 8241. Gaithersburg, MD: National Institute of Standards and Technology, Dec. 2018, NIST IR 8241. DOI: 10.6028/NIST.IR.8241. URL: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8241.pdf (visited on 10/12/2023).

[12] D.-T. Dam, T.-H. Tran, V.-P. Hoang, C.-K. Pham, and T.-T. Hoang. "A Survey of Post-Quantum Cryptography: Start of a New Race". en. In: *Cryptography* 7.3 (Sept. 2023). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 40. ISSN: 2410-387X. DOI: 10.3390/cryptography7030040. URL: https://www.mdpi.com/2410-387X/7/3/40 (visited on 01/07/2025).

[13] D. J. Bernstein and A. H. Tanja Lange. *Post-Quantum Cryptography - Integration study | ENISA*. en. Nov. 2022. URL: https://www.enisa.europa.eu/publications/post-quantum-cryptography-integration-study (visited on 01/21/2025).

[14] B. Sullivan. "Cryptographic agility: Defending against the sneakers scenario". In: *MSDN Magazine* (2009).

[15] A. Wiesmaier, N. Alnahawi, T. Grasmeyer, J. Geißler, A. Zeier, P. Bauspieß, and A. Heinemann. *On PQC Migration and Crypto-Agility*. arXiv:2106.09599 [cs]. June 2021. DOI: 10.48550/arXiv.2106.09599. URL: http://arxiv.org/abs/2106.09599 (visited on 01/07/2025).

[16] N. Alnahawi, N. Schmitt, A. Wiesmaier, A. Heinemann, and T. Grasmeyer. *On the State of Crypto-Agility*. Publication info: Published elsewhere. 18. Deutscher IT-Sicherheitskongress. 2023. URL: https://eprint.iacr.org/2023/487 (visited on 08/25/2023).

[17] D. Ott, C. Peikert, and o. w. participants other workshop. *Identifying Research Challenges in Post Quantum Cryptography Migration and Cryptographic Agility*. arXiv:1909.07353. Sept. 2019. DOI: 10.48550/arXiv.1909.07353. URL: http://arxiv.org/abs/1909.07353 (visited on 11/20/2024).

[18] D. Ott, K. Paterson, and D. Moreau. “Where Is the Research on Cryptographic Transition and Agility?” en. In: *Communications of the ACM* 66.4 (Apr. 2023), pp. 29–32. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3567825. URL: https://dl.acm.org/doi/10.1145/3567825 (visited on 05/09/2023).

[19] D. Sikeridis, D. Ott, S. Huntley, S. Sharma, V. K. Dhanasekar, M. Bansal, A. Kumar, A. U. N, D. Beveridge, and S. Veeraswamy. *ELCA: Introducing Enterprise-level Cryptographic Agility for a Post-Quantum Era*. Publication info: Preprint. 2023. URL: https://eprint.iacr.org/2023/1539 (visited on 10/12/2023).

[20] J. Blessing, M. A. Specter, and D. J. Weitzner. “Cryptography in the Wild: An Empirical Analysis of Vulnerabilities in Cryptographic Libraries”. In: *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*. ASIA CCS ’24. New York, NY, USA: Association for Computing Machinery, July 2024, pp. 605–620. ISBN: 9798400704826. DOI: 10.1145/3634737.3657012. URL: https://dl.acm.org/doi/10.1145/3634737.3657012 (visited on 12/27/2024).

[21] A. Silonosov. *SECMAR Project: Enhancing Maritime Cybersecurity with Drones and Drone Management in the South Baltic Region*. May 2023.

[22] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. “CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM”. In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. London: IEEE, Apr. 2018, pp. 353–367. ISBN: 978-1-5386-4228-3. DOI: 10.1109/EuroSP.2018.00032. URL: https://ieeexplore.ieee.org/document/8406610/ (visited on 07/19/2024).

[23] D. Dolev, C. Dwork, and M. Naor. “Non-malleable cryptography”. In: *Proceedings of the twenty-third annual ACM symposium on Theory of Computing*. STOC ’91. New York, NY, USA: Association for Computing Machinery, Jan. 1991, pp. 542–552. ISBN: 978-0-89791-397-3. DOI: 10.1145/103418.103474. URL: https://dl.acm.org/doi/10.1145/103418.103474 (visited on 01/17/2025).

[24] T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (July 1985). Conference Name: IEEE Transactions on Information Theory, pp. 469–472. ISSN: 1557-9654. DOI: 10.1109/TIT.1985.1057074. URL: https://ieeexplore.ieee.org/document/1057074 (visited on 01/19/2025).

[25] M. Abdalla, M. Bellare, and P. Rogaway. *DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem*. Publication info: Published elsewhere. Appeared in the THEORY OF CRYPTOGRAPHY LIBRARY and has been included in the ePrint Archive. 1999. URL: https://eprint.iacr.org/1999/007 (visited on 01/13/2025).

[26] Z. Galil, S. Haber, and M. Yung. “Symmetric Public-Key Encryption”. en. In: *Advances in Cryptology — CRYPTO ’85 Proceedings*. Ed. by H. C. Williams. Berlin, Heidelberg: Springer, 1986, pp. 128–137. ISBN: 978-3-540-39799-1. DOI: 10.1007/3-540-39799-X_12.

[27] M. Naor and M. Yung. “Public-key cryptosystems provably secure against chosen ciphertext attacks”. en. In: *Proceedings of the twenty-second annual ACM symposium on Theory of computing - STOC ’90*. Baltimore, Maryland, United States: ACM Press, 1990, pp. 427–437. ISBN: 978-0-89791-361-4. DOI: 10.1145/100216.100273. URL: http://portal.acm.org/citation.cfm?doid=100216.100273 (visited on 01/17/2025).

[28] M. Bellare and P. Rogaway. “Minimizing the use of random oracles in authenticated encryption schemes”. en. In: *Information and Communications Security*. Ed. by Y. Han, T. Okamoto, and S. Qing. Berlin, Heidelberg: Springer, 1997, pp. 1–16. ISBN: 978-3-540-69628-5. DOI: 10.1007/BFb0028457.

[29] M. Bellare and P. Rogaway. “Optimal asymmetric encryption–How to encrypt with RSA”. In: *Eurocrypt*. Vol. 94. 1994, pp. 92–111.

[30] V. Shoup. *A Proposal for an ISO Standard for Public Key Encryption*. Publication info: Published elsewhere. Unknown where it was published. 2001. URL: https://eprint.iacr.org/2001/112 (visited on 02/20/2024).

[31] R. L. Rivest, A. Shamir, and L. Adleman. “A method for obtaining digital signatures and public-key cryptosystems”. In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342. URL: https://dl.acm.org/doi/10.1145/359340.359342 (visited on 12/16/2024).

[32] N. Koblitz. “Elliptic curve cryptosystems”. en. In: *Mathematics of Computation* 48.177 (1987), pp. 203–209. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/S0025-5718-1987-0866109-5. URL: https://www.ams.org/mcom/1987-48-177/S0025-5718-1987-0866109-5/ (visited on 01/19/2025).

[33] D. Boneh and M. Franklin. *Identity Based Encryption From the Weil Pairing*. Publication info: Published elsewhere. Unknown where it was published. 2001. URL: https://eprint.iacr.org/2001/090 (visited on 12/12/2023).

[34] M. Blaze, G. Bleumer, and M. Strauss. “Divertible protocols and atomic proxy cryptography”. en. In: *Advances in Cryptology — EUROCRYPT’98*. Ed. by K. Nyberg. Berlin, Heidelberg: Springer, 1998, pp. 127–144. ISBN: 978-3-540-69795-4. DOI: 10.1007/BFb0054122.

[35] B. Wall and P. Walsh. “Cryptographically Enforced Orthogonal Access Control at Scale”. In: *Proceedings of the 6th International Workshop on Security in Cloud Computing*. SCC ’18. New York, NY, USA: Association for Computing Machinery, May 2018, pp. 57–65. ISBN: 978-1-4503-5759-3. DOI: 10.1145/3201595.3201602. URL: https://dl.acm.org/doi/10.1145/3201595.3201602 (visited on 08/29/2023).

[36] D. Boneh, X. Boyen, and S. Halevi. “Chosen Ciphertext Secure Public Key Threshold Encryption Without Random Oracles”. en. In: *Topics in Cryptology – CT-RSA 2006*. Ed. by D. Pointcheval. Berlin, Heidelberg: Springer, 2006, pp. 226–243. ISBN: 978-3-540-32648-9. DOI: 10.1007/11605805_15.

[37] A. Shelley. “Addressing Security Concerns with Chinese Drones and DJI Products”. en. In: (). URL: https://www.academia.edu/43142234/Addressing_Security_Concerns_with_Chinese_Drones_and_DJI_Products (visited on 01/30/2025).

[38] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab. “Security analysis of drones systems: Attacks, limitations, and recommendations”. In: *Internet of Things* 11 (Sept. 2020), p. 100218. ISSN: 2542-6605. DOI: 10.1016/j.iot.2020.100218. URL: https://www.sciencedirect.com/science/article/pii/S2542660519302112 (visited on 01/30/2025).

[39] S. Scheffler and J. Mayer. “SoK: Content Moderation for End-to-End Encryption”. en. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (Apr. 2023), pp. 403–429. ISSN: 2299-0984. DOI: 10.56553/popets-2023-0060. URL: https://petsymposium.org/popets/2023/popets-2023-0060.php (visited on 10/30/2023).

[40] M. Backendal, M. Haller, and K. G. Paterson. “MEGA: malleable encryption goes awry”. In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 146–163.

[41] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila. *A Formal Security Analysis of the Signal Messaging Protocol*. Publication info: Published elsewhere. Major revision. IEEE EuroS&P 2017. 2016. URL: https://eprint.iacr.org/2016/1013 (visited on 09/19/2023).

[42] W. Wang, S. M. Sadjadi, and N. Rishe. “A Survey of Major Cybersecurity Compliance Frameworks”. In: *2024 IEEE 10th Conference on Big Data Security on Cloud (BigDataSecurity)*. May 2024, pp. 23–34. DOI: 10.1109/BigDataSecurity62737.2024.00013. URL: https://ieeexplore.ieee.org/abstract/document/10565236 (visited on 01/24/2025).

[43] M. Parmar and A. Miles. “Cyber Security Frameworks (CSFs): An Assessment Between the NIST CSF v2.0 and EU Standards”. In: *2024 Security for Space Systems (3S)*. May 2024, pp. 1–7. DOI: 10.23919/3S60530.2024.10592293. URL: https://ieeexplore.ieee.org/document/10592293 (visited on 01/24/2025).

[44] K. Fischer, I. Trummová, P. Gajland, Y. Acar, S. Fahl, and M. A. Sasse. *On The Challenges of Bringing Cryptography from Papers to Products: Results from an Interview Study with Experts (Extended Version)*. eng. Publisher: Zenodo. Oct. 2023. DOI: 10.5281/zenodo.8404611. URL: https://zenodo.org/records/8404611 (visited on 06/08/2024).

[45] R. Cramer and V. Shoup. “A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack”. en. In: *Advances in Cryptology — CRYPTO ’98*. Ed. by H. Krawczyk. Berlin, Heidelberg: Springer, 1998, pp. 13–25. ISBN: 978-3-540-68462-6. DOI: 10.1007/BFb0055717.

[46] P. Schwabe, D. Stebila, and T. Wiggers. “Post-Quantum TLS without Handshake Signatures”. In: 2020, pp. 1461–1480. DOI: 10.1145/3372297.3423350.

[47] N. Alnahawi, J. Müller, J. Oupický, and A. Wiesmaier. *SoK: Post-Quantum TLS Handshake*. Publication info: A major revision of an IACR publication in CIC 2024. 2023. URL: https://eprint.iacr.org/2023/1873 (visited on 12/17/2024).

[48] J.-A. Septien-Hernandez, M. Arellano-Vazquez, M. A. Contreras-Cruz, and J.-P. Ramirez-Paredes. “A Comparative Study of Post-Quantum Cryptosystems for Internet-of-Things Applications”. en. In: *Sensors* 22.2 (Jan. 2022). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 489. ISSN: 1424-8220. DOI: 10.3390/s22020489. URL: https://www.mdpi.com/1424-8220/22/2/489 (visited on 01/03/2025).

[49] C. Brunetta, H. Heum, and M. Stam. *SoK: Public Key Encryption with Openings*. Publication info: Preprint. 2023. URL: https://eprint.iacr.org/2023/1337 (visited on 12/18/2024).

[50] L. Medley, A. F. Loe, and E. A. Quaglia. “SoK: Delay-Based Cryptography”. English. In: IEEE Computer Society, July 2023, pp. 169–183. ISBN: 9798350321920. DOI: 10.1109/CSF57540.2023.00028. URL: https://www.computer.org/csdl/proceedings-article/csf/2023/219200a426/1On91gWwHsI (visited on 12/18/2024).

[51] J. Duman, D. Hartmann, E. Kiltz, S. Kunzweiler, J. Lehmann, and D. Riepel. *Group Action Key Encapsulation and Non-Interactive Key Exchange in the QROM*. Publication info: A major revision of an IACR publication in

ASIACRYPT 2022. 2022. URL: https://eprint.iacr.org/2022/1230 (visited on 11/21/2024).

[52] P. Ravi, J. Howe, A. Chattopadhyay, and S. Bhasin. "Lattice-based Key-sharing Schemes: A Survey". In: *ACM Comput. Surv.* 54.1 (Jan. 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3422178. URL: https://doi.org/10.1145/3422178.

[53] M. Pursche, N. Puch, S. N. Peters, and M. P. Heinl. *SoK: The Engineer's Guide to Post-Quantum Cryptography for Embedded Devices*. Publication info: Preprint. 2024. URL: https://eprint.iacr.org/2024/1345 (visited on 11/19/2024).

[54] J. Howe, T. Prest, and D. Apon. *SoK: How (not) to Design and Implement Post-Quantum Cryptography*. Publication info: Published elsewhere. Major revision. CT-RSA 2021. 2021. URL: https://eprint.iacr.org/2021/462 (visited on 08/25/2023).

[55] M. Barbosa, G. Barthe, K. Bhargavan, B. Blanchet, C. Cremers, K. Liao, and B. Parno. *SoK: Computer-Aided Cryptography*. Publication info: Published elsewhere. Oakland 2021. 2019. URL: https://eprint.iacr.org/2019/1393 (visited on 04/18/2024).

[56] N. Huaman, J. Suray, J. H. Klemmer, M. Fourné, S. Amft, I. Trummová, Y. Acar, and S. Fahl. ""You have to read 50 different {RFCs} that contradict each other": An Interview Study on the Experiences of Implementing Cryptographic Standards". en. In: 2024, pp. 7249–7266. ISBN: 978-1-939133-44-1. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/huaman (visited on 09/27/2024).

[57] Y. Ding and K. Klein. "Model-Driven Application-Level Encryption for the Privacy of E-health Data". In: *2010 International Conference on Availability, Reliability and Security*. Feb. 2010, pp. 341–346. DOI: 10.1109/ARES.2010.91. URL: https://ieeexplore.ieee.org/abstract/document/5438071 (visited on 04/08/2024).

[58] R. Barnes, B. Beurdouche, R. Robert, J. Millican, E. Omara, and K. Cohn-Gordon. *RFC 9420: The Messaging Layer Security (MLS) Protocol*. USA: RFC Editor, June 2023.

[59] D. Budgen and P. Brereton. "Performing systematic literature reviews in software engineering". In: *Proceedings of the 28th international conference on Software engineering*. ICSE '06. New York, NY, USA: Association for Computing Machinery, May 2006, pp. 1051–1052. ISBN: 978-1-59593-375-1. DOI: 10.1145/1134285.1134500. URL: https://dl.acm.org/doi/10.1145/1134285.1134500 (visited on 02/03/2025).

[60] I. S. o. Security and S. p. Privacy. *Systematizing Systematization of Knowledge*. en. 2010. URL: https://oaklandsok.github.io/ (visited on 01/23/2010).

[61] K. Ermoshina and F. Musiani. ""Standardising by running code": the Signal protocol and de facto standardisation in end-to-end encrypted messaging". In: *Internet Histories* 3.3-4 (Oct. 2019). Publisher: Routledge _eprint: https://doi.org/10.1080/24701475.2019.1654697, pp. 343–363. ISSN: 2470-1475. DOI: 10.1080/24701475.2019.1654697. URL: https://doi.org/10.1080/24701475.2019.1654697 (visited on 09/18/2023).

[62] T. Walshe and A. Simpson. "An Empirical Study of Bug Bounty Programs". English. In: 2020, pp. 35–44. ISBN: 978-1-72816-280-5. DOI: 10.1109/IBF50092.2020.9034828.

[63] K. Sridhar and M. Ng. "Hacking for good: Leveraging HackerOne data to develop an economic model of Bug Bounties". In: *Journal of Cybersecurity* 7 (Feb. 2021). DOI: 10.1093/cybsec/tyab007.

[64] S. Atefi, A. Sivagnanam, A. Ayman, J. Grossklags, and A. Laszka. "The Benefits of Vulnerability Discovery and Bug Bounty Programs: Case Studies of Chromium and Firefox". In: *Proceedings of the ACM Web Conference 2023*. WWW '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 2209–2219. ISBN: 978-1-4503-9416-1. DOI: 10.1145/3543507.3583352. URL: https://dl.acm.org/doi/10.1145/3543507.3583352 (visited on 10/12/2023).

[65] A. Laszka, M. Zhao, A. Malbari, and J. Grossklags. "The Rules of Engagement for Bug Bounty Programs". In: *Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26 – March 2, 2018, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, Feb. 2018, pp. 138–159. ISBN: 978-3-662-58386-9. DOI: 10.1007/978-3-662-58387-6_8. URL: https://doi.org/10.1007/978-3-662-58387-6_8 (visited on 04/04/2024).

[66] A. Mosteiro-Sanchez, M. Barceló, J. Astorga, and A. Urbieta. *All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of CP-ABE Libraries*. Sept. 2022.

[67] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta. *Too Many Options: A Survey of ABE Libraries for Developers*. arXiv:2209.12742 [cs]. Sept. 2022. URL: http://arxiv.org/abs/2209.12742 (visited on 02/09/2024).

[68] S. Ho, J. Protzenko, A. Bichhawat, and K. Bhargavan. “Noise: A Library of Verified High-Performance Secure Channel Protocol Implementations”. In: *2022 IEEE Symposium on Security and Privacy (SP)*. ISSN: 2375-1207. May 2022, pp. 107–124. DOI: 10.1109/SP46214.2022.9833621. URL: https://ieeexplore.ieee.org/document/9833621 (visited on 11/27/2024).

[69] J. Protzenko, B. Parno, A. Fromherz, C. Hawblitzel, M. Polubelova, K. Bhargavan, B. Beurdouche, J. Choi, A. Delignat-Lavaud, C. Fournet, N. Kulatova, T. Ramananandro, A. Rastogi, N. Swamy, C. Wintersteiger, and S. Zanella-Beguelin. *EverCrypt: A Fast, Verified, Cross-Platform Cryptographic Provider*. Publication info: Published elsewhere. Proceedings of the IEEE Symposium on Security and Privacy, 2020. 2019. URL: https://eprint.iacr.org/2019/757 (visited on 04/18/2024).

[70] I. Muslukhov, Y. Boshmaf, and K. Beznosov. “Source attribution of cryptographic API misuse in android applications”. English. In: 2018, pp. 133–146. ISBN: 978-1-4503-5576-6. DOI: 10.1145/3196494.3196538.

[71] J. Lee and D. S. Wallach. “Removing Secrets from Android’s TLS”. en. In: *Proceedings 2018 Network and Distributed System Security Symposium*. San Diego, CA: Internet Society, 2018. ISBN: 978-1-891562-49-5. DOI: 10.14722/ndss.2018.23365. URL: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_01B-3_Lee_paper.pdf (visited on 01/02/2024).

[72] D. Lazar, H. Chen, X. Wang, and N. Zeldovich. “Why does cryptographic software fail? a case study and open problems”. In: *Proceedings of 5th Asia-Pacific Workshop on Systems*. APSys ’14. New York, NY, USA: Association for Computing Machinery, June 2014, pp. 1–7. ISBN: 978-1-4503-3024-4. DOI: 10.1145/2637166.2637237. URL: https://dl.acm.org/doi/10.1145/2637166.2637237 (visited on 12/27/2024).

[73] J. Blessing, M. A. Specter, and D. J. Weitzner. *You Really Shouldn’t Roll Your Own Crypto: An Empirical Study of Vulnerabilities in Cryptographic Libraries*. arXiv:2107.04940 [cs]. July 2021. DOI: 10.48550/arXiv.2107.04940. URL: http://arxiv.org/abs/2107.04940 (visited on 01/20/2024).

[74] S. Rahaman, Y. Xiao, S. Afrose, F. Shaon, K. Tian, M. Frantz, M. Kantarcioglu, and D. Yao. “Cryptoguard: High precision detection of cryptographic vulnerabilities in massive-sized Java projects”. English. In: ISSN: 1543-7221. 2019, pp. 2455–2472. ISBN: 978-1-4503-6747-9. DOI: 10.1145/3319535.3345659.

[75] D. Méndez Fernández, W. Böhm, A. Vogelsang, J. Mund, M. Broy, M. Kuhrmann, and T. Weyer. “Artefacts in software engineering: a fundamental positioning”. en. In: *Software & Systems Modeling* 18.5 (Oct. 2019), pp. 2777–2786.

ISSN: 1619-1374. DOI: 10.1007/s10270-019-00714-3. URL: https://doi.org/10.1007/s10270-019-00714-3 (visited on 02/04/2025).

[76] B. Turhan, T. Menzies, A. Bener, and J. Di Stefano. “On the relative value of cross-company and within-company data for defect prediction”. In: *Empirical Software Engineering* 14.5 (2009), pp. 540–578. DOI: 10.1007/s10664-008-9103-7.

[77] T. Tomiyama, Y. Umeda, M. Ishii, M. Yoshioka, and T. Kiriyama. “Knowledge systematization for a knowledge intensive engineering framework”. en. In: *Knowledge Intensive CAD*. ISSN: 1868-422X. Springer, Boston, MA, 1996, pp. 33–52. ISBN: 978-0-387-34930-5. DOI: 10.1007/978-0-387-34930-5_4. URL: https://link.springer.com/chapter/10.1007/978-0-387-34930-5_4 (visited on 02/03/2025).

[78] F. Guo, W. Susilo, X. Chen, P. Jiang, J. Lai, and Z. Zhao. *SOK: Research Motivations of Public-Key Cryptography*. Publication info: Preprint. 2023. URL: https://eprint.iacr.org/2023/715 (visited on 11/01/2024).

[79] A. Abraham and J. Yang. “A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services”. en. In: *Proceedings of the 2023 International Conference on Advances in Computing Research (ACR’23)*. Ed. by K. Daimi and A. Al Sadoon. Cham: Springer Nature Switzerland, 2023, pp. 408–422. ISBN: 978-3-031-33743-7. DOI: 10.1007/978-3-031-33743-7_33.

[80] B. Posey, C. Gropp, B. Wilson, B. McGeachie, S. Padhi, A. Herzog, and A. Apon. “Addressing the Challenges of Executing a Massive Computational Cluster in the Cloud”. In: *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. May 2018, pp. 253–262. DOI: 10.1109/CCGRID.2018.00040. URL: https://ieeexplore.ieee.org/abstract/document/8411029 (visited on 09/22/2024).

[81] X. Ding and J. Shan. “Diagnosing Virtualization Overhead for Multi-threaded Computation on Multicore Platforms”. In: *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*. Nov. 2015, pp. 226–233. DOI: 10.1109/CloudCom.2015.102. URL: https://ieeexplore.ieee.org/abstract/document/7396161 (visited on 09/22/2024).

[82] W. R. Shadish. *Experimental and quasi-experimental designs for generalized causal inference*. sv. (Visited on 01/23/2025).

[83] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. en. Berlin, Heidelberg: Springer, 2024. ISBN: 978-3-662-69305-6 978-3-662-69306-3. DOI: 10.1007/978-3-

662-69306-3. URL: https://link.springer.com/10.1007/978-3-662-69306-3 (visited on 02/05/2025).

[84] P. Runeson and M. Höst. “Guidelines for conducting and reporting case study research in software engineering”. en. In: *Empirical Software Engineering* 14.2 (Apr. 2009), pp. 131–164. ISSN: 1573-7616. DOI: 10.1007/s10664-008-9102-8. URL: https://doi.org/10.1007/s10664-008-9102-8 (visited on 02/06/2025).

[85] J.-K. Zinzindohoué, K. Bhargavan, J. Protzenko, and B. Beurdouche. “HACL*: A Verified Modern Cryptographic Library”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 1789–1806. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3134043. URL: https://dl.acm.org/doi/10.1145/3133956.3134043 (visited on 02/11/2025).

# Paper I

# Telemetry data sharing based on Attribute-Based Encryption schemes for cloud-based Drone Management system.

*Alexandr Silonosov, Lawrence Henesey*
*In: in Proceedings of the 19th International Conference on Availability, Reliability and Security, in ARES □24. pp. 1□8., New York, NY, USA: Association for Computing Machinery, Jul. 2024,*

Abstract

The research presented in the paper evaluates practices of Attribute-Based Encryption, leading to a proposed end-to-end encryption strategy for a cloud-based drone management system. Though extensively used for efficiently gathering and sharing video surveilance data, these systems also collect telemetry information with sensitive data. This paper presents a study addressing the current state of knowledge, methodologies, and challenges associated with supporting cryptographic agility for End-to-End Encryption (E2EE) for telemetry data confidentiality.

To enhance cryptographic agility performance, a new metric has been introduced for cryptographic library analysis that improves the methodology by considering Attribute-Based Encryption (ABE) with a conventional key-encapsulation mechanism in OpenSSL. A comprehensive series of experiments are undertaken to simulate cryptographic agility within the proposed system, showcasing the practical applicability of the proposed approach in measuring cryptographic agility performance.

## 1 Introduction

In this paper, we study the feasibility and advantages of implementing end-to-end encryption (E2EE) approach [1] in drone management system for telemetry data confidentiality, particularly in addressing real-world data confidentiality challenges, which are essential for protecting cyber-security in cloud-based systems. Further investigation into the benefits of modern cryptographic tools like Attribute-Based Encryption

(ABE) compared to traditional cryptography such as Elliptic-Curve Cryptography (ECC) can contribute to identifying the most sustainable approach for ensuring telemetry data confidentiality in cloud-based systems.

Applied cryptography has increased complexity in real-world applications [2–4]. Regulatory compliance such as FIPS, NIS2, GDPR, HIPPA, PCI-DSS [5] increases efforts to advocate a paradigm shift for industry participants for the trending need: maximize the use of data in encrypted form.

Compliance with this standarts typically involves implementing appropriate data safeguards, and E2EE is frequently recognized as an effective means of achieving data confidentiality and integrity [6, 7] which means intensive use of cryptography due to complexities of key management, security criteria establishment [8, 9]. A new research frontier has emerged in science that helps evaluate the efficiency of solving mentioned tasks at a scale referred to as cryptographic-agility and there is a knowledge gap in this area [10].

This paper first introduces the problem of telemetry data confidentiality captured from various IoT devices and software. Our case- study shows what type of confidential information is captured by telemetry in Drone-management system. State of practice shows cases when telemetry data was a valuable target of cyber-criminals therefore it is feasible to investigate confidentiality and protection controls for such data. Our contribution in this study is two-fold. First we introduce security properties and Application Level Encryption concept for telemetry data sharing confidentiality, second we contribute to crypto-agility research frontier by providing experiment results for cryptographic migration at scale for telemetry data encryption in cloud-based architecture.

The paper is structured as follows: The background section provides information related to case-study of cloud-based drone management system and confidentiality problems arising from telemetry data logs, challenges related to it along with a proposal for comparing two crypto-libraries for telemetry data confidentiality. Section 2 encompasses a review of related work. Section 3 outlines the method and experiments. Section 4 discusses the results, while Section 6 concludes the paper with remarks and provides pointers for future research.

## 1.1 Background

Software Vendors and IoT manufacturers want to improve user experience with their products and therefore collect various product related performance data such as error rates, resources use, accuracy of operations, delays in functioning, internal state of integrated components [11, 12].

Figure 1 represents the architecture of Drone Management system studied during SECMAR research project [13]. Drone Appliance (A) device collects telemetry audit data from Drone (D) and stores it as plain text data in memory. Each data element

contains seven text elements including payload. Case-study demonstrated that payload strings contains confidential information such as drone coordinates, flight path, and authentication information to access server (FM). Cyber threat analysis identified certain weaknesses in the system and potential ways for attackers to access assets represented in Fig.1. During the case study, we also identified that telemetry data could be accessed by employees of the Drone Management (FM) vendor and stored in their computers for an unlimited time. The problem has two aspects, if telemetry data is not protected then 1) a successful cyber attack on employee's computer may lead to an information leakage of user on-premises (P) location and user authentication information for server (FM) 2) by successfully attacking (P) attackers may retrieve user authentication information for server (FM) to steal user drone data including aerial photos and locations of (P). Given that the payload data may be sensitive, it is feasible to investigate how encrypting it may influence the performance of application (A) and cloud architecture (FM) . Waters et al. [14] first proposed a solution for the problem by using Identify Based Encryption scheme, later Goyal et al. [15] proposed an application of an Attribute-Based Encryption scheme to encrypt audit logs of system activity. However, neither study provided detailed performance evaluations and comparisons with traditional public key encryption schemes. In April 2024 cyber-security incident with Cisco DUO cloud-based multi-factor authentication service also revealed how cyber-criminals can exploit access to unencrypted telemetry logs in supply chain attacks [1].

# 2 Related work

Exist body of research on Attribute-based encryption (ABE) applications in healthcare, logistics and supply chain management. In the following subsections, we represent ABE and mention some similarities with conventional public key encryption (PKE) systems, then we dive into ABE application across different sectors, then the state of practice in data security compliance and End-to-End Encryption (E2EE).

## 2.1 Attribute-Based Encryption

Key-Policy ABE[15] (2006) and Ciphertext-Policy ABE [16] (2008) are public key cryptosystem, a successor of Fuzzy Identity Based Encryption (IBE) [17], that allows to encrypt a message by using a combination of public keys instead of a single public key in conventional PKE systems like RSA. However in ABE such a key combination referred as attributes because it is normal text strings and ABE authority use them to generate an access structure cryptographically embedded into each secret key and as a result the key has a flexible decryption capability, i.e. can decrypt any message if

[1] https://www.securityweek.com/cisco-duo-says-hack-at-telephony-supplier-exposed-mfa-sms-logs/

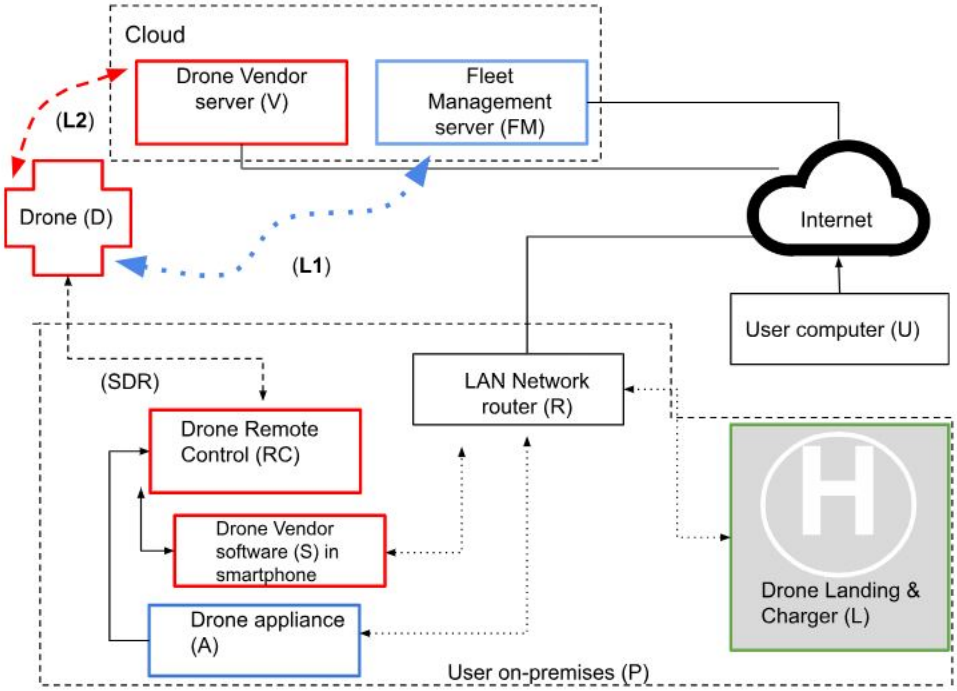


**Figure 1:** Architecture of cloud-based Drone Management system from study [13]

.

there is a match between embedded access structures in a key and ciphertext.

As a result in ABE different secret keys can be used to decrypt the same ciphertext. Another unique feature is that ABE is capable to generate new secret key that will decrypt any existing ciphertext. It means, it is possible to create ciphertext without existing secret key for decryption. This feature does not exists in conventional PKE with strongly bounded public and secret key that start coexist at the same time. The role of recipient in ABE is not defined so strictly as in conventional PKE, because during encryption sender specifies access structure containing a combination of attributes. This can be interpreted as when you send a message to multiple recipients but without a knowledge of actual users that will be able to decrypt, or imagine a set of recipients, each has its own secret key but they all can read the same ciphertext.

Exist body of research for similar functionality implemented in conventional ECC known as Key Encapsulation Mechanism (KEM) to multiple parties proposed by [18] in 2005.

## 2.2 ABE applications

A range of attribute-based encryption (ABE) schemes have been proposed for healthcare data sharing, each with its own strengths and weaknesses. Mhatre et al. [19] provided a comprehensive survey of 13 ABE schemes, highlighting the need for a multi-authority ABE model in Health Information Exchange. A sample Attribute-Based Access Control (ABAC) type of fine-grained access control where Electronic Health Record attributes are used to create access policies by attributes such as Patient Name, Doctor name, illness name, date. More recent SLR by Imam et al. [20] reviews ABE applications in healthcare with the aim of retrospective tracing the development of ABE approaches, including enhancement over conventional cryptosystems since 2005 including metrics mostly related to ABE such as access structure, revocation type and authority type. The review identified 7 domains for EHR applications of ABE: which are ‘‘CPABE”, ‘‘KPABE”, ‘‘Hybrid”, ‘‘Multiauthority-ABE”, ‘‘Searchable Encryption-ABE”, ‘‘Blockchain / Decentralized” ‘‘Hierarchical ABE”. The authors identified 4 major applications related to healthcare : ‘‘Wireless Sensor Networks (WSN)”, ‘‘Personal Health Record (PHR)”, ‘‘Electronic Health Record (HER)” and ‘‘Internet of Things (IoT)” and mentions 2 real-world implementations of ABE, [21], [22] and none of the business implementation as a state of practice and readiness indicator. The review also contains ”Features” and ”Enhancements” metrics which are listed as open-ended keywords such as ”proxy re-encryption”, ” surpasses similar schemes in functionality and security”, ”reduce complexity of key management”, ”Policy management” , Reduces storage complexity”, ”Policy hiding”, ”Dynamic ABE paradigm”, ”Efficient key generation”.

Zhao et al. [23] , [24] introduces an ABE scheme with non-monotonic access structures and fine-grained attribute revocation, enhancing data protection and access control in Mobile-healthcare cloud computing system. These studies collectively underscore the importance of ABE in healthcare data sharing and the need for further research to optimize its performance. In more recent work Zhao et al. [25] summarized that for some ABE schemes the size of public parameters may take up to 3 kilobytes per attribute, cipertext size can take up to 1.3 kilobytes per attribute. Encryption and Decryption time up to 5 ms - 9 ms per attribute. Mentioned papers compare schemes on Revocation type (user, atribute), Non-monotonic access structure, communication costs in terms of ciper text size and computation performance in seconds for key generation, single encryption and decryption operation. Also research outlines architecture of healthcare data sharing system without discussing how many ABE encryption keys needed per patient, per each electornic health record (EHR) or other entity in system, this represent an open challenge or limitation since all presented ABE schemes requires that plaintext message, i.e. EHR data, should belong to Gt group meaning that it should be an elliptic curve point, however no discussion about mapping a message to elliptic curve point is provided. Despite exist ABE schemes that does not require plaintext message belongs to Gt, they uses XOR as encryption

function, which make it practical to encrypt only high-entropy data i.e. encryption keys. Also it is hard to map encryption performance to the most common privacy phases of EHC [26]. Private Key distribution choices and threat model are also out of scope of work in ABE for healthcare.

Revocable Attribute-Base Scheme is when attribute should be revoced from encrypted data. MA-ABE allows multiple entities to issue private and public keys. MA-ABE can be used to cooperate various entities in one system without a central authority

In logistics information system privacy protection based on ABE was proposed in [27], [28]. Gao et al. [28] adopted ciphertext-policy attribute-based encryption (CP-ABE) to encrypt segmented logistics information in different access policies with defined threat model, data flow and security properties as goals described in various attack scenarios. [29]

**Table 1:** Comparison of studies of ABE applications separated by domains: healthcare, logistics. Criteria list in columns, CP-computational performance in ms time, CC- Communication costs in kilobytes, TM - structured threat and adversary model, DF - data model and flow, AM - data access model, SP - security properties, KD - key distribution

| Ref | CP | CC | - | TM | DF | AM | SP | KD |
|---|---|---|---|---|---|---|---|---|
| [23] | + | F | | | | | | |
| [24] | F | F | | | | | | |
| [25] | + | + | | | | | | |
| [28] | + | | | + | + | + | + | |
| [27] | + | | | - | + | | | |
| [29] | + | + | + | + | + | | + | + |

## 2.3 Data security compliance

For our work we need to mentions existing body of data security compliance related to ABE. Since 2016 NIST USA has initiated standardization processes for new types of cryptography such as Post-Quantum Cryptography (PQC), Lightweight Cryptography, pairing-based scheme cryptography [2] and Multi-Party Threshold Cryptography [3], where ABE and IBE are listed in Subcategiry 2 as threshold-friendly encryption and key generator schemes.

in 2018 European Telecommunications Standards Institute (ETSI) published two technical specifications (ETSI TS 103-458, 103-532) of ABE applications and one technical report about quantum safe IBE [4]. Specification 103-458 describe the recommended ABE scheme for 4 use cases: IoT devices, WLAN, cloud and mobile services. Each use cases includes high-level security requirements with sub-use-cases with stakeholder lists, scenarios of access and data flow. Specification 103-532 describe in detail four ABE schemes CP-WATERS-KEM [16], KP-GPSW-KEM [15],

[2] https://csrc.nist.gov/Projects/pairing-based-cryptography

[3] https://csrc.nist.gov/Projects/threshold-cryptography

[4] https://www.etsi.org/

KP-FAME-KEM and CP-FAME-KEM [30], referred as key encapsulation mechanism (KEM) in ciphertext-policy (CP) or key-policy (KP) mode. Attribute-revocation and secret key revocation is recommended by storing a list of revoked attributes and associating expiration time intervals as attribute for each secret key.

A Survey of ABE software libraries by Mosteiro-Sanchez et al. [31] shows that all tree libraries designed for production (OpenABE, Rabe, Charm) contain separate KEM abstraction layer and data encryption layers. Indeed, according to ABE schemes construction [16], [15] the encrypted message should belong to group Gt (bilinear map of two multiplicative cyclic groups by elliptic curves). Since it is hard to map real-world data to be the element of algebra group, it is feasible to use message in ABE with a purpose of a symmetric encryption key, higliting the fact that this value is good enough (have a good entropy) to be used as is as encryption key for symmetric encryption algorithm like AES. The code review presented in in table 2, of implementation of all libraries, designed for production and research, demonstrates that fact. All libraries uses ABE as KEM and conventional cryptographic tools for arbitrary-length data encryption. And even more interesting fact, that all libraries always uses random symmetric encryption key for ABE encryption assuming such simplified session key management due to inherited property of ABE: it is possible to produce an unlimited number of decryption keys for the same data block.

## 2.4 End-to-end encryption protocols

End-to-end encryption (E2EE) approach assumes client-server architure with integrated encryption scheme where the “ends” of the communication (a “sender” and one or more “recipients”) can send messages to each other via an abstract central channel and where the server does not have access to the cryptographic keys necessary to read or invisibly alter the message [32] .

Figure 1 represents the proposed End-to-End Encryption approach for telemetry data in Drone Management system. For proposed E2EE architecture, appliance (A) employ Application Level Encryption (ALE) [33] to encrypt telemetry data. As shown, the Appliance (A) transfers the encrypted data in the gray ALE area to the server, and later, ALE-encrypted data arrives at cloud storage in always-encrypted form. In this case, the cloud also uses encryption in-transit and encryption at-Rest principles, but encryption key management for ALE belongs only to the ’ends’ i.e. appliance (A) and vendor engineering team (U).

According to the E2EE protocol design, only the ’ends’ owns the encryption and decryption keys. According to our proposal, requests and approvals to access data are controlled only by the Vendor representative (FM), who manages the encryption key lifecycle and storage. Encryption key ownership by the client is an important requirement that distinguishes E2EE from other approaches. Before E2EE adoption, data protection architectures in cloud systems assume that the server always has access

to decryption keys to process the data in plain, decrypted form.

Research of E2EE applications shows the increased complexity of internal key management, where each ”end” holds a particular set of own key material, including all recipients [34], [35], [36]. For example, Signal protocol is literally a key factory, producing new encryption keys with each chat message [9]. This creates an overhead on the application layer to organize additional key management. State of practice shows that code libraries exist to implement the ALE approach, for example LogSentinel [5], Cloaked Search [6] and AWS Crypto Tools [7].

To conclude, E2EE approach is cryptography-intensive since they contains significant amount of cryptographic code in ALE layer apart from the underlying cryptographic library. Depending on the mentioned challenge the software variability issues [37] also relate to ALE as software abstraction layer.

# 3 ABE experiment for telemetry data protection

The review of the E2EE technology state of practice demonstrated that it is highly interconnected with the application layer, which assumes a specific model for telemetry data. In this section we describe the proposed data protection architecture for telemetry logs and experiment. We assume the Vendor would like to be able to allow legitimate access to a subset of all audit log entries to the respective engineering department or even individual engineering according to some criteria. We present new criteria for the construction of secure telemetry logs, specifically to allow the secure access capabilities by means of decryption of log entries of a narrow scope. For example, if an engineering team dedicated to image recognition needed to investigate records generated only by image recognition code assets, or suppose a first tier- technical support department need to access Module name field from all of records generated by a specific customer ID or specific date. For such access to be considered secure, it must be impossible for an adversary to learn the content of telemetry records. We admit that the adversary can be an insider in the sense that it may have access to User on-premises assets (see Figure 1 or may have some legitimate access given to it by the cloud infrastructure provider. We would like to ensure that, assuming he does not compromise the Vendor engineering team itself, it cannot view the contents of any telemetry log record.

## 3.1 Telemetry data components

According to case study [13], telemetry each log L consists of a series of records, R0, R1, . . ., Rn. Each record Ri contains seven text fields presented Table 2. In our

[5] https://logsentinel.com/why-is-searchable-encryption-so-important

[6] https://ironcorelabs.com/products/cloaked-search/

[7] https://docs.aws.amazon.com/aws-crypto-tools/

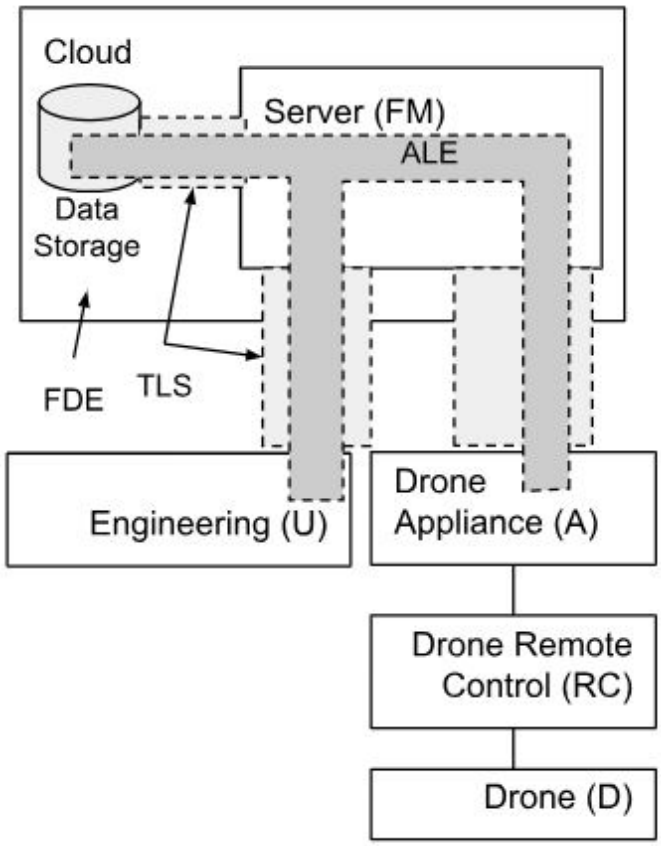


**Figure 2:** Logical confidentiality perimeter (green) between data sourses D1, D2 and Customer in proposed E2EE system.

proposal, encrypted record R contains the following fields:

1. $C, C', C_{10}...C_n$ Encapsulation of symmetric key SK with ABE or RSA scheme for n={10, 20, 40} access attributes or public keys, respectively.

2. $E_{SK1}(F_i)$, the symmetric encryption of the telemetry fields (i=1..7) to be logged under a key SK.

For our experiment we reviewed 5 open-source ABE libraries listed in Table 2[8] [9] [10] labeled as Maintained in work [31]. The performance analysis, maturity in terms on maintenance status, number of supported ABE schemes of these libraries was

[8] github.com/zeutro/OpenABE
[9] github.com/Fraunhofer-AISEC/rabe
[10] github.com/JHUISI/charm

done by Mosteiro-Sanchez et al. [31]. Here we introduce new comparison criteria for ABE librarues, Bin - library can be precompiled as binary code (C++/Rust/Go languages), KEM - only key encapsulation functionality is provided, meaning that a message from Gt field should be provided for input, SYM - conventional symmetric encryption abstraction is present that allows to encrypt arbitrary data of any length, SK - session key control, AM - interface/serialization to access attributes from ciphertext metadata. ST - Stateful model to maintain the session state across multiple crypto operations. All libraries, except Charm, compiled to native binary code of target Platform like ARM, X64, x86 and it makes computation performance comparison between the libraries more reliable when the same abstration interface is used, KEM or KEM + SYM. Rabe and GoFE provide the only abstration layer that include ABE-specific functionality, which is KEM, and data encryption implemented by conventional symmetric encryption (SYM), in literature this approach is referred as hybrid mode. The purpose of KEM is to generate random session key and encrypt it with ABE, later it is used in SYM layer to encrypt arbitrary size data by conventional symmetric encryption scheme, but some libraries has SYM layer integrated with KEM and thus not provide control over session key, i.e. they do not designed to re-use the same session key over a few consecutive encryption operations, which is possible in OpenAPE, Charm and CiFEr. OpenABE is the only library that has two abstraction interfaces for KEM and SYM and they are separated so it can be used in hybrid mode or KEM-only mode that allows re-using session key over multiple encryption operations (ST) with any symmetric encryption scheme. Also OpenABE has ciphertext metadata interface (AM) that includes both access policy and SYM information. .

**Table 2:** Comparison of ABE code libraries. Math - Pairings code library dependency.

| Ref | Math | Bin | KEM | SYM | SK | AM | ST |
|---|---|---|---|---|---|---|---|
| OpenABE | RELIC | + | +/- | AEAD | + | + | + |
| Rabe | Rabe-bn | + | - | AES-GCM | - | + | - |
| Charm | RELIC | - | + | - | + | N/A | - |
| GoFE | Bn-256 | + | - | AES-CBC | - | - | - |
| CiFEr | Miracl | + | + | - | + | + | - |

The absence of SYM layer in Charm and CiFEr assume the only KEM use case. For example, the possibility to control session key (rfc-8446. 4.6.1 Session Ticket Message) is a property of TLS protocol and the key lifespan can be up to 7 days. OpenABE, RABE AND GoFE, through the interface design provide clear assumptions that new random session key should be used per each encryption operation.

Our experiment relies on the different KEM schemes assuming the data model and certain cryptographic API abstrations play a certain role in cryptographic agility performance in terms of cryptographic migration capacity. Comparing the performance of different KEM schemes illustrates how practical aspects of the cryptographic API

design, such as the interfaces and metadata control can affect cryptographic agility in terms of dependence on mathematic library or code library and data structure. For the experiment, we simulate multiparty KEM with OpenABE and OpenSSL (EVP-SealInit) code libraries for the first part of encrypted record R. The symmetric encryption for second part of telemetry data record R is out of scope in this experimnet. Key Encapsulation approaches to multiple Parties also exist in conventional cryptosystems [18] and it is feasible to compare it with to ABE scheme.

**Table 3:** Telemetry data record fields

| Id | Description | Value example |
|---|---|---|
| F1 | Date | 01-01-2024 |
| F2 | Time | 12:00:00 245 |
| F3 | Event type | Info, Debug, Error |
| F4 | Process session id | System context number |
| F5 | Module name | Software asset name |
| F6 | Message | Performance indicator value |
| F7 | Payload | Other specific value |

On the figure 1 we provide schematical representation of ends (U and D) in E2EE setup business User (U) and Drone (D). In our experiment we consider Drone (D) model without user provided encryption key management and therefore E2EE encryption start in Drone Appliance (A). Initially, Drone equipment generates plain data, unencrypted and encrypted by the (A) before upload to Cloud. For telemetry data we propose implementation of Application Level Encryption (ALE) on side (U) and (A).

The ABE architecture for our proposed E2EE system, consists of the following three entities:

- Vendor: is a Key generation center (KGC) of ABE setup and primarily responsible for generating ABE public parameters and distributing public and secret keys for the engineering teams. Engineering teams encode public keys to software code for telemetry encryption.
- Cloud: The cloud is responsible for storing the encrypted telemetry data and public parameters.
- engineering team can access encrypted telemetry data stored in the cloud and can decrypt it if and only if its attribute set satisfies the access policy in secret key.

Telemetry dataset described in table 2 contains 7 fields. According to proposed architecture fields F2..F6 are employed to define access policy for ABE key encapsulation for fine-grained access control. Engineers with different level of access level will be able to decrypt log records data only if field value match specific pattern, for example logs for a specific date (F2), logs from a specific software module (F5). In addition

access policy can be increased to include customer identification and drone model identification.

Simulation running flow:

- Setup CPU single thread performance rating to R=1000, 2000, 4000.
- Encapsulate 256 bit session key SK with CP-ABE [38]), KP-ABE with Open-ABE library for I =10, 20, 40 access attributes representing field values.
- Encapsulate 256 bit session key SK with EVP-SealInit, OpenSSL for I =10, 20, 40 RSA 2048 public keys.

. All tests were performed on the virtual host with Ubuntu OS with three configurations of vCPU with single thread performance rating [11] defined from 1000 to 4000, thus simulating three different CPUs starting from low-performance CPU up to Intel Core i9-12900F.

# 4 Results

Figures 3,4 , 5 represent the comparison of key encapsulation performance between ABE and traditional RSA public key cryptography, to delegate access structure based on 10, 20, 40 access attributes in ABE or public keys in RSA respectively. Figure 4 represents performance results for RSA in a separate graph to see its timings more precisely due to faster speed over ABE. All experimental evaluations were performed on three different vCPU configurations defined by single thread performance rating, complemented by 16GB RAM. The implementation of our simulation was realized using the x64 binaries compiled in Ubuntu 22 system with default Make file coming for each code library. Results in table 3 shows that KEM in OpenABE produces composite ciphertext material (Comm. Cost - communication cost) with associated data including access policy, this highlight the requirement of necessary abstraction layer to access metadata (AM) in the ciphertext. Opposite to it, OpenSSL KEM metadata includes only public key bytes assuming the metadata information will be coded directly by the higher code abstraction layers. Both libraries includes integrated symmetric encryption abstraction (SYM) that allows to encrypt arbitrary data of any length alongside encapsulated session key. ( SK - session key control) OpenSSL KEM implemented in stateful model (ST) to maintain the session key across multiple crypto operations. Communication cost repesent the required storage size for ciphertext including metadata per each log record.

The results demonstrated that ABE key encapsulation takes 100 times more time for each telemetry record encryption compared to the traditional RSA-based scheme in CPU with a rating of 1000 and 50 times more on faster CPU with a rating of 4000.

[11] www.cpubenchmark.net/singleThread.html

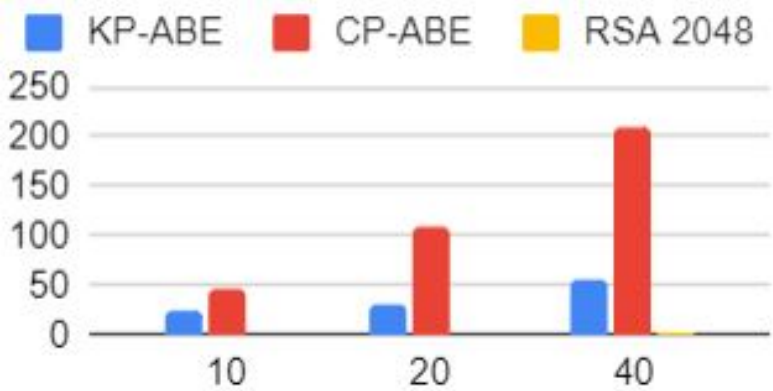


**Figure 3:** ABE key encapsulation performance in ms. on vCPU with R=1000 single thread performance rating, for 10, 20, 40 access attributes.

.

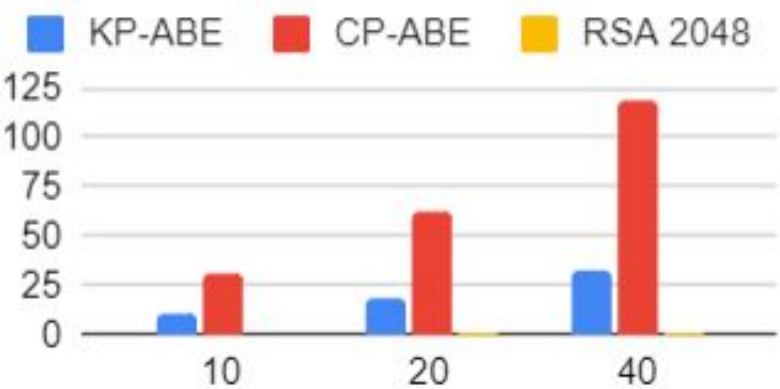


**Figure 4:** ABE key encapsulation performance in ms. on vCPU with R=2000 single thread performance rating, for 10, 20, 40 access attributes.

.

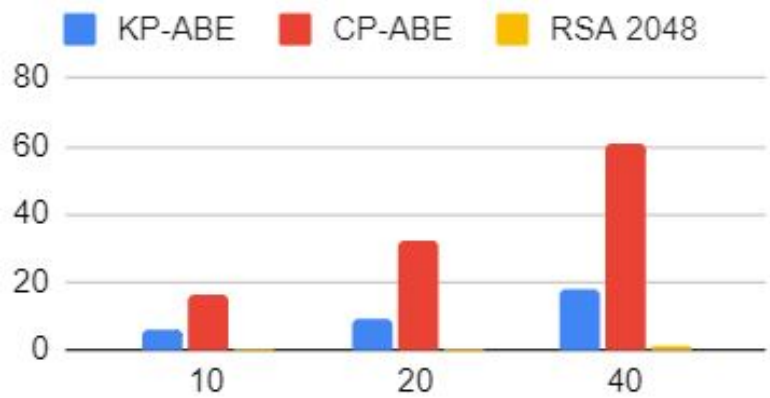


**Figure 5:** ABE key encapsulation performance in ms. on vCPU with R=4000 single thread performance rating, for 10, 20, 40 access attributes.

.

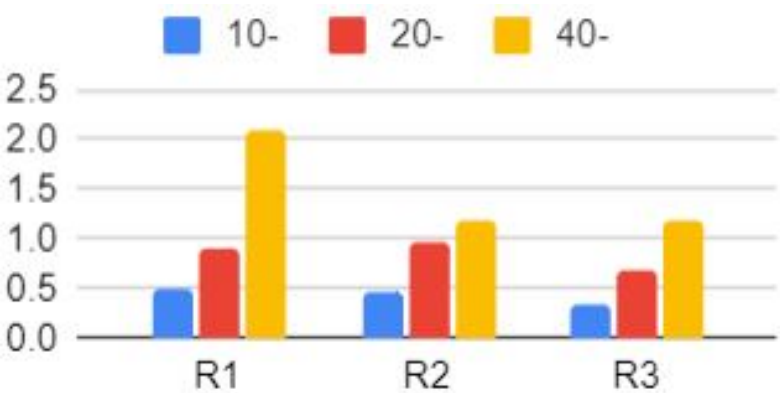


**Figure 6:** OpenSSL (RSA 2048) multi-recipient key encapsulation performance in ms. on vCPU with R1=1000, R2=2000, R3=4000 single thread performance rating, for 10, 20, 40 public keys.

.

**Table 4:** Comparison of KEM criteria and performance in OpenABE and OpenSSL code libraries.

| Criteria | OpenABE | OpenSSL |
|---|---|---|
| SYM | AES | Any |
| Mathematics | RELIC BN P254 | RSA 2048 |
| SK | + | + |
| AM | + | - |
| ST | +/- | + |
| Communication cost in bytes | CP-ABE: 80+140*n<br>KP-ABE: 100+55*n | 256 * n |
| Key size, in bits. | 2080 | 2048 |

However KP-ABE is the most effective in terms of the small size of the encapsulated key size (communication costs). The overhead on the size of the encrypted log Record, according to Table 3, estimates 650 bytes as the minimum for 10 access attributes for the key encapsulation part only (KP-ABE), not taking into account the second part with record fields in plain text encrypted with the symmetric key. This means that in practice, for an average log record of size 190 characters, the size of the encrypted log record increases to 650 + 190 = 840 bytes.

# 5 Discussion

In the proposed approach for telemetry data protection, asymmetric KEM provides a security advantage since each deployed software only stores public parameters, there are no secret keys in Drone Appliance for an attacker to steal. Compromising a server does not allow the attacker to decrypt historical telemetry data. Compromising an engineering workstation allows the attacker to access only a subset of telemetry data and only stored locally. A drawback of this approach is the performance overhead. We note that this approach is also easy to modify to allow 1) separating the ability to access (i.e., decrypt) only certain fields from the log Record and 2) to decrease performance overhead. To do this multiple Symmetric Keys needs to be introduced per each record. To minimize the overhead on computation and communication costs, Symmetric Key(s) can be reused across multiple log records.

# 6 Conclusions and future research

This paper introduces telemetry data confidentiality approach based on end-to-end encryption and explores various challenges, with a central emphasis application of ABE cryptosystem in IoT infrastructure. An examination of the existing ABE applications was performed and identified the necessity of further research for public-key cryptography applications for telemetry data confidential sharing. Our study provides strong evidence supporting the feasibility and efficiency of the proposed approach for cryptographic agility for telemetry data. This research paves the way for

future developments in secure cloud-based telemetry data management, offering a scalable solution that can adapt to the evolving landscape of cybersecurity threats and the continuous advancement of end-to-end encryption adoptions for new applications.

# References


[1] B. Hale and C. Komlo. *On End-to-End Encryption*. Publication info: Preprint. MINOR revision. 2022. URL: https://eprint.iacr.org/2022/449 (visited on 04/02/2024).

[2] J. Haney, M. Theofanos, Y. Acar, and S. S. Prettyman. *Organizational views of NIST cryptographic standards and testing and validation programs*. en. Tech. rep. NIST IR 8241. Gaithersburg, MD: National Institute of Standards and Technology, Dec. 2018, NIST IR 8241. DOI: 10.6028/NIST.IR.8241. URL: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8241.pdf (visited on 10/12/2023).

[3] L. Nurgalieva, A. Frik, and G. Doherty. "A Narrative Review of Factors Affecting the Implementation of Privacy and Security Practices in Software Development". In: *ACM Computing Surveys* 55.14s (July 2023), 320:1–320:27. ISSN: 0360-0300. DOI: 10.1145/3589951. URL: https://dl.acm.org/doi/10.1145/3589951 (visited on 10/12/2023).

[4] P. L. Gorski, L. L. Iacono, D. Wermke, C. Stransky, S. Möller, Y. Acar, and S. Fahl. "Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic {API} Misuse". en. In: 2018, pp. 265–281. ISBN: 978-1-939133-10-6. URL: https://www.usenix.org/conference/soups2018/presentation/gorski (visited on 10/12/2023).

[5] S. W. A. Hamdani, H. Abbas, A. R. Janjua, W. B. Shahid, M. F. Amjad, J. Malik, M. H. Murtaza, M. Atiquzzaman, and A. W. Khan. "Cybersecurity Standards in the Context of Operating System: Practical Aspects, Analysis, and Comparisons". In: *ACM Computing Surveys* 54.3 (May 2021), 57:1–57:36. ISSN: 0360-0300. DOI: 10.1145/3442480. URL: https://dl.acm.org/doi/10.1145/3442480 (visited on 04/12/2024).

[6] N. Cecchinato, A. Toma, C. Drioli, G. Oliva, G. Sechi, and G. L. Foresti. "A Secure Real-time Multimedia Streaming through Robust and Lightweight AES Encryption in UAV Networks for Operational Scenarios in Military Domain". In: *Procedia Computer Science*. 2022 International Conference on Military Communication and Information Systems (ICMCIS) 205 (Jan. 2022), pp. 50–57. ISSN: 1877-0509. DOI: 10.1016/j.procs.2022.09.006. URL: https://www.sciencedirect.com/science/article/pii/S1877050922008717 (visited on 04/12/2024).

[7] C. Esposito, M. Ficco, A. Castiglione, F. Palmieri, and A. De Santis. “Distributed Group Key Management for Event Notification Confidentiality Among Sensors”. In: *IEEE Transactions on Dependable and Secure Computing* 17.3 (May 2020). Conference Name: IEEE Transactions on Dependable and Secure Computing, pp. 566–580. ISSN: 1941-0018. DOI: 10.1109/TDSC.2018.2799227. URL: https://ieeexplore.ieee.org/document/8274911 (visited on 04/12/2024).

[8] X. Huang and R. Chen. “A Survey of Key Management Service in Cloud”. In: *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. ISSN: 2327-0594. Nov. 2018, pp. 916–919. DOI: 10.1109/ICSESS.2018.8663805. URL: https://ieeexplore.ieee.org/document/8663805 (visited on 04/11/2024).

[9] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila. *A Formal Security Analysis of the Signal Messaging Protocol*. Publication info: Published elsewhere. Major revision. IEEE EuroS&P 2017. 2016. URL: https://eprint.iacr.org/2016/1013 (visited on 09/19/2023).

[10] D. Ott, K. Paterson, and D. Moreau. “Where Is the Research on Cryptographic Transition and Agility?” en. In: *Communications of the ACM* 66.4 (Apr. 2023), pp. 29–32. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3567825. URL: https://dl.acm.org/doi/10.1145/3567825 (visited on 05/09/2023).

[11] C. Bird, T. Menzies, and T. Zimmermann. *The Art and Science of Analyzing Software Data*. English. The Art and Science of Analyzing Software Data. Pages: 648. 2015. ISBN: 978-0-12-411543-9. DOI: 10.1016/C2012-0-07289-4.

[12] B. Ding, J. Kulkarni, and S. Yekhanin. “Collecting Telemetry Data Privately”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/253614bbac999b38b5b60cae531c4969-Abstract.html (visited on 04/12/2024).

[13] A. Silonosov. *SECMAR Project: Enhancing Maritime Cybersecurity with Drones and Drone Management in the South Baltic Region*. May 2023.

[14] B. Waters, D. Balfanz, G. Durfee, and D. Smetters. “Building an Encrypted and Searchable Audit Log”. In: (Dec. 2003).

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters. *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data*. Publication info: Published elsewhere. Extended abstract to appear in ACM CCS 2006. This is the full version. 2006. URL: https://eprint.iacr.org/2006/309 (visited on 10/31/2023).

[16] B. Waters. *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*. Publication info: Published elsewhere. Unknown where it was published. 2008. URL: https://eprint.iacr.org/2008/290 (visited on 02/18/2024).

[17] A. Sahai and B. Waters. *Fuzzy Identity Based Encryption*. Publication info: Published elsewhere. Unknown where it was published. 2004. URL: https://eprint.iacr.org/2004/086 (visited on 02/22/2024).

[18] N. P. Smart. "Efficient Key Encapsulation to Multiple Parties". en. In: *Security in Communication Networks*. Ed. by C. Blundo and S. Cimato. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 208–219. ISBN: 978-3-540-30598-9. DOI: 10.1007/978-3-540-30598-9_15.

[19] S. Mhatre, A. V. Nimkar, and S. N. Dhage. "Comparative study on attribute-based encryption for health records in cloud storage". In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (May 2017). Conference Name: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) ISBN: 9781509037049 Place: Bangalore Publisher: IEEE, pp. 647–652. DOI: 10.1109/RTEICT.2017.8256677.

[20] R. Imam, K. Kumar, S. M. Raza, R. Sadaf, F. Anwer, N. Fatima, M. Nadeem, M. Abbas, and O. Rahman. "A systematic literature review of attribute based encryption in health services". In: *Journal of King Saud University - Computer and Information Sciences* 34.9 (Oct. 2022), pp. 6743–6774. ISSN: 1319-1578. DOI: 10.1016/j.jksuci.2022.06.018. URL: https://www.sciencedirect.com/science/article/pii/S1319157822002269 (visited on 02/15/2024).

[21] C. Wang, X. Liu, and W. Li. *Implementing a Personal Health Record Cloud Platform Using Ciphertext-Policy Attribute-Based Encryption*. Journal Abbreviation: Proceedings of the 2012 4th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2012 Pages: 14 Publication Title: Proceedings of the 2012 4th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2012. Sept. 2012. ISBN: 978-1-4673-2279-9. DOI: 10.1109/iNCoS.2012.65.

[22] S. Porwal and S. Mittal. "Implementation of Ciphertext Policy-Attribute Based Encryption (CP-ABE) for fine grained access control of university data". en. In: *2017 Tenth International Conference on Contemporary Computing (IC3)*. Noida: IEEE, Aug. 2017, pp. 1–7. ISBN: 978-1-5386-3077-8. DOI: 10.1109/IC3.2017.8284289.

[23] X. Zhao and Q. Su. “Revocable Attribute-Base Scheme with Enhanced Security and Privacy for Healthcare Data Sharing”. en. In: *2023 IEEE 14th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*. Beijing, China: IEEE, Nov. 2023, pp. 1–8. ISBN: 9798350371024. DOI: 10.1109/PAAP60200.2023.10391571. URL: https://ieeexplore.ieee.org/document/10391571/ (visited on 02/07/2024).

[24] Y. Zhao, P. Fan, H. Cai, Z. Qin, and H. Xiong. “Attribute-based Encryption with Non-Monotonic Access Structures Supporting Fine-Grained Attribute Revocation in M-healthcare”. In: *Int. J. Netw. Secur.* (2017). (Visited on 02/12/2024).

[25] J. Zhao, K. Zhang, J. Gong, and H. Qian. “Lavida: Large-Universe, Verifiable, and Dynamic Fine-Grained Access Control for E-Health Cloud”. In: *IEEE Transactions on Information Forensics and Security* 19 (2024). Conference Name: IEEE Transactions on Information Forensics and Security, pp. 2732–2745. ISSN: 1556-6021. DOI: 10.1109/TIFS.2024.3350925. URL: https://ieeexplore.ieee.org/document/10382677 (visited on 02/12/2024).

[26] A. V, A. P. Nirmala, B. K, A. Christi, and N. A. “A Review on Cloud Cryptography Techniques to Improve Security in E-health Systems”. In: *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. Mar. 2022, pp. 100–104. DOI: 10.1109/ICCMC53470.2022.9753999. URL: https://ieeexplore.ieee.org/document/9753999 (visited on 02/12/2024).

[27] S. Hong, H. Pan, Y. Fang, J. Ma, X. Qi, and Y. Hu. “A Logistics Privacy Protection Scheme Based on Ciphertext Policy Attribute-Based Key Encapsulation”. In: *2022 International Conference on Blockchain Technology and Information Security (ICBCTIS)*. July 2022, pp. 218–224. DOI: 10.1109/ICBCTIS55569.2022.00057. URL: https://ieeexplore.ieee.org/document/9845073 (visited on 02/12/2024).

[28] Q. Gao, J. Zhang, J. Ma, C. Yang, J. Guo, and Y. Miao. “LIP-PA: A Logistics Information Privacy Protection Scheme with Position and Attribute-Based Access Control on Mobile Devices”. en. In: *Wireless Communications and Mobile Computing* 2018 (July 2018). Publisher: Hindawi, e9436120. ISSN: 1530-8669. DOI: 10.1155/2018/9436120. URL: https://www.hindawi.com/journals/wcmc/2018/9436120/ (visited on 02/12/2024).

[29] T. Li, R. Zhang, and Y. Zhang. “PriExpress: Privacy-preserving express delivery with fine-grained attribute-based access control”. In: *2016 IEEE Conference on Communications and Network Security (CNS)*. Oct. 2016, pp. 333–341. DOI: 10.1109/CNS.2016.7860501. URL: https://ieeexplore.ieee.org/document/7860501 (visited on 02/15/2024).

[30] S. Agrawal and M. Chase. *FAME: Fast Attribute-based Message Encryption*. Publication info: Published elsewhere. ACM CCS 2017. 2017. URL: https://eprint.iacr.org/2017/807 (visited on 02/27/2024).

[31] A. Mosteiro-Sanchez, M. Barceló, J. Astorga, and A. Urbieta. *All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of CP-ABE Libraries*. Sept. 2022.

[32] S. Scheffler and J. Mayer. "SoK: Content Moderation for End-to-End Encryption". en. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (Apr. 2023), pp. 403–429. ISSN: 2299-0984. DOI: 10.56553/popets-2023-0060. URL: https://petsymposium.org/popets/2023/popets-2023-0060.php (visited on 10/30/2023).

[33] Y. Ding and K. Klein. "Model-Driven Application-Level Encryption for the Privacy of E-health Data". In: *2010 International Conference on Availability, Reliability and Security*. Feb. 2010, pp. 341–346. DOI: 10.1109/ARES.2010.91. URL: https://ieeexplore.ieee.org/abstract/document/5438071 (visited on 04/08/2024).

[34] K. G. Paterson, M. Scarlata, and K. T. Truong. "Three Lessons From Threema: Analysis of a Secure Messenger". en. In: 2023, pp. 1289–1306. ISBN: 978-1-939133-37-3. (Visited on 03/14/2025).

[35] M. Backendal, M. Haller, and K. G. Paterson. "MEGA: malleable encryption goes awry". In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 146–163.

[36] T. von Arx and K. G. Paterson. "On the cryptographic fragility of the Telegram ecosystem". In: *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 2023, pp. 328–341.

[37] M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch. "COVAMOF: A Framework for Modeling Variability in Software Product Families". English. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3154 (2004). ISBN: 9783540229186, pp. 197–213. ISSN: 0302-9743. DOI: 10.1007/978-3-540-28630-1_12.

[38] J. Bethencourt, A. Sahai, and B. Waters. "Ciphertext-Policy Attribute-Based Encryption". en. In: *2007 IEEE Symposium on Security and Privacy (SP '07)*. Berkeley, CA: IEEE, May 2007, pp. 321–334. ISBN: 978-0-7695-2848-9. DOI: 10.1109/SP.2007.11.

# Paper II

# Crypto-agility performance analysis for AIS data sharing confidentiality based on attribute-based encryption.

*Alexandr Silonosov, Lawrence Henesey*
*In: 14th International Conference on Computer Science and Information Technology (CCSIT 2024), Copenhagen Sept 21-22, 2024, AIRCC Publishing Corporation, 2024, pp. 193□212.*

Abstract

The research presented in the paper evaluates practices of Attribute-Based Encryption as a key encapsulation mechanism and proposes end-to-end encryption architecture for a cloud-based ship tracking system confidentiality. Though extensively used for efficiently gathering and sharing maritime data, these systems draw information from Automated Identification Systems, ports, and vessels, which can lead to cyber-security vulnerabilities. This paper presents a study addressing the current state of knowledge, methodologies, and challenges associated with supporting cryptographic agility for End-to-End Encryption (E2EE) for AIS data. To study cryptographic agility performance, a new metric has been introduced for cryptographic library analysis that improves the methodology by comparing Attribute-Based Encryption (ABE) with state of the art CRYSTALS-Kyber key encapsulation mechanism (KEM) that belongs to Post-Quantum Cryptography (PQC). A comprehensive series of experiments are undertaken to simulate large-scale cryptographic migration within the proposed system, showcasing the practical applicability of the proposed approach in measuring cryptographic agility performance.

## 1 Introduction

The Automatic Identification System (AIS) is a vessel navigation safety communications system; its main aim is to improve maritime traffic awareness and safety [1]. IMO standards for AIS technology do not include message confidentiality, integrity, or authentication of participating parties and elements. Therefore, it is vulnerable to

various cyber-threats [2], [3]. To enhance the security of the AIS [4], [5] proposed the use of Public Key Infrastructure (PKI) and applying symmetric key cryptography to protect and authenticate AIS communications between two AIS transceivers. Goudosis et al. [6], [7] proposed the use of PKI based on identity-based encryption (IBE) for authentication and encryption of AIS messages between multiple parties. Both approaches mainly focus on protecting AIS data confidentiality in transit, but they can also be used to protect AIS data at rest with some limitations since it is encrypted. Proposals [8], [9] include a technique using the TESLA authentication protocol ( RFC-4082 standard) to ensure AIS transceivers authentication and message integrity in transit only. Commercial AIS products SAAB R6 exist that provide encrypted AIS communication between trusted devices in conventional AIS networks. These findings, which identified the limitations of the above-mentioned approaches for AIS communication security, motivate the work presented in this paper, which extends the AIS confidentiality to data encryption at rest.

According to the International Maritime Organization (IMO), the publication of AIS data transmitted by ships on the World Wide Web or elsewhere could be detrimental to the safety and security of ships and port facilities. Currently, historical AIS data is publicly available (for a fee through a subscription license) by commercial providers such as MarineTraffic, Fleetmon, VesselFinder and Spire. For example, Spire captures AIS communication by using satellites, processes it almost in real-time, and then makes it available as a historical AIS data source in the cloud. Other logistics and maritime industry participants highly depend on such data to support their vessel fleet surveillance, logistics management systems, and port management systems. Research related to ship tracking systems based on AIS indicates [10] highlights the requirements to manage significant amounts of tracking data and various personnel roles to access it.

Despite the benefits of AIS, there are concerns about the publication of historical AIS data, as it could pose risks to the safety and security of ships and port facilities, as highlighted by the IMO. Research emphasizes the challenges of managing significant amounts of tracking data and various personnel roles to access it. According to research [11] open access to navigation data and the availability of commodity software-defined radio transponders, these exploits might be carried out inexpensively and leverage entirely open information sources; therefore, data confidentiality for historical AIS data remains an unexplored area of research.

In this paper, we study the feasibility of extending PKI-based approaches for historical AIS data confidentiality by implementing an end-to-end encryption (E2EE) approach with KEM based on public key cryptography. Our study addresses real-world AIS data confidentiality challenges for vessel traffic surveillance systems in a cloud-based environment. Further we introduce the experimentation approach of comparing different key encapsulation mechanisms (KEM) for AIS data confidentiality in real-world dataset.

Our contribution to this study is two-fold. First, we introduce forward secrecy

with Attribute-Based Encryption (ABE) for AIS data confidentiality. Second, we contribute to the crypto-agility research frontier [12], [13] by providing experimental results of cryptographic migration performance from ABE to Kyber KEM for AIS data protection.

The paper is structured as follows: The background section provides information related to the case study of cloud-based AIS data management systems and problems arising from plain-text AIS logs, along with a proposal for comparing two types of key encapsulation mechanisms. Section 2 encompasses a review of related work. Section 3 outlines the method and experiments. Section 4 represents the experiments results, while Sections 5 and 6 discuss on the paper results and provide pointers for future research.

## 1.1 Background

Satellite AIS (S-AIS) is a method for collecting AIS data transmissions generated by vessels. It specifically allows capturing AIS data transmissions from beyond the reach of land-based receiving AIS stations. According to the state of the industry [14], satellite captures raw AIS data packets and then transfers them to the Owner data storage almost in real-time. The data storage is located in the Cloud as presented in Figure 1 also receiving AIS data from coastal AIS stations. The organization that owns the system provides instant access to the AIS data for business users who integrate the data into their business systems for ship fleet monitoring. To our knowledge, Cloud uses various safeguards to protect the data at-rest but the recent cyber-attacks show that it is not sufficient. If attackers penetrate cloud providers, they may gain access to all data in cloud storage. One of the possible solutions is to implement encryption for data in motion by using an end-to-end encryption approach. This approach allows the protection of data from cyber threats to cloud providers and compromised data owners. The key challenge is understanding the benefits of various Public Key Encryption tools that influence the architecture of end-to-end encryption systems in terms of performance, key distribution, and security properties and maintaining their usefulness for the Owner and Users. This paper presents a design for end-to-end encryption architecture for AIS data. We propose to include encryption ”ends” all types of AIS data capture equipment, including S-AIS. The other ”end” is a designated trusted party, the User, which is capable of decrypting only a subset of historical AIS data according to the access policy defined by the system owner. We also introduce the notion of a specific surveillance decryption key that allows the User to perform decryption for ship traffic information by using, for example, only vessel ID, type of vessels, or geographic region. We also designed an experiment to compare two types of KEM by a few metrics to understand the feasibility of ABE for low-performance embedded systems in comparison to state of the art, quantum-resistant cryptography as Kyber.

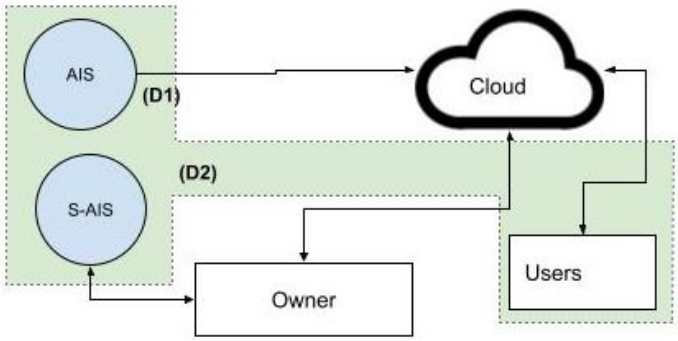


**Figure 1:** Logical confidentiality perimeter (green) between data sources D1, D2, and Users in proposed E2EE architecture.

# 2 Related work

According to surveys [15] and [16], the most well-studied applications of Attribute-Based Encryption (ABE) include healthcare, logistics, and generic IoT use cases. In the following subsections, we represent ABE real-life use cases and mention some similarities with conventional public key encryption (PKE) systems. Then we dive into ABE application across different sectors, then the state of practice in data security compliance related to ABE and End-to-End Encryption (E2EE).

## 2.1 Attribute-Based Encryption

Key-Policy ABE[17] (2006) and Ciphertext-Policy ABE [18] (2008) are successors of Fuzzy Identity Based Encryption (IBE) [19], that allows to encrypt a message by using a combination of multiple public keys instead of a single public key in traditional PKE systems like RSA. In ABE, such a key combination is referred to as an attribute because it is a normal text string, and ABE authority uses them to generate an access structure cryptographically embedded into each secret key. As a result, the key has a flexible decryption capability and can decrypt any message if there is a match between embedded access structures in a secret key and ciphertext.

As a result, in ABE, different secret keys can be used to decrypt the same ciphertext, which is not possible in conventional PKE system like RSA. ABE allows a set of recipients, each having its own secret key, to decrypt the same ciphertext as outlined by our previous study [12]. Similar functionality implemented in traditional PKE known as Key Encapsulation Mechanism (KEM) to multiple parties proposed by [20] in 2005.

A Survey of ABE software libraries by Mosteiro-Sanchez et al. [21] discovered tree libraries designed for production systems, which are OpenABE, Rabe, Charm. According to our review it contains separate abstraction layers used as KEM and

data encryption layers. Indeed, according to ABE schemes construction [18], [17] the encrypted message should belong to group Gt which is bilinear map of two multiplicative cyclic groups by elliptic curves. Since it is hard to map real-world data to be the element of Gt, it is only feasible to use a message in ABE with the purpose of a symmetric encryption key, highlighting the fact that this value is good enough (have a good entropy) to be used as is as an encryption key for symmetric encryption algorithm like AES. The review of ABE libraries source code presented in table 2, of implementation of all libraries designed for production and research, demonstrates that fact. All libraries use ABE as KEM and conventional cryptographic tools for arbitrary-length data encryption. An even more interesting fact is that all libraries always use random symmetric encryption keys for ABE encryption, assuming such simplified session key management due to the inherited property of ABE: it is possible to produce an unlimited number of decryption keys for the same data block.

## 2.2 ABE applications

A range of attribute-based encryption (ABE) schemes have been proposed for healthcare data sharing, each with its own strengths and weaknesses. Mhatre et al. [22] provided a comprehensive survey of 13 ABE schemes, highlighting the need for a multi-authority ABE model in Health Information Exchange. A sample Attribute-Based Access Control (ABAC) type of fine-grained access control where Electronic Health Record attributes are used to create access policies by attributes such as Patient Name, Doctor name, illness name, and date. A more recent survey by Imam et al. [23] reviews ABE applications in healthcare with the aim of retrospective tracing the development of ABE approaches, including enhancement over conventional cryptosystems since 2005, including metrics mostly related to ABE such as access structure, revocation type, and authority type. The review identified 7 domains for EHR applications of ABE: which are ‘‘CPABE”, ‘‘KPABE”, ‘‘Hybrid”, ‘‘Multiauthority-ABE”, ‘‘Searchable EncryptionABE”, ‘‘Blockchain / Decentralized” ‘‘Hierarchical ABE”. The authors identified 4 major applications related to healthcare : ‘‘Wireless Sensor Networks (WSN)”, ‘‘Personal Health Record (PHR)”, ‘‘Electronic Health Record (HER)” and ‘‘Internet of Things (IoT)” and mentions 2 real-world implementations of ABE, [24], [25] and none of the business implementation as a state of practice and readiness indicator. The review also contains ”Features” and ”Enhancements” metrics which are listed as open-ended keywords such as ”proxy re-encryption”, ” surpasses similar schemes in functionality and security”, ”reduce the complexity of key management”, ”Policy management” , Reduces storage complexity”, ”Policy hiding”, ”Dynamic ABE paradigm”, ”Efficient key generation”.

Zhao et al. [26] , [27] introduces an ABE scheme with non-monotonic access structures and fine-grained attribute revocation, enhancing data protection and access control in Mobile-healthcare cloud computing system. These studies collectively

underscore the importance of ABE in healthcare data sharing and the need for further research to optimize its performance. In more recent work Zhao et al. [28] summarized that for some ABE schemes the size of public parameters may take up to 3 kilobytes per attribute, and cipertext size can take up to 1.3 kilobytes per attribute. Encryption and Decryption time up to 5 ms - 9 ms per attribute. Mentioned papers compare schemes on Revocation type (user, attribute), Non-monotonic access structure, and communication costs in terms of cipertext size and computation performance in seconds for key generation, single encryption, and decryption operation. Also, research outlines the architecture of a healthcare data sharing system without discussing how many ABE encryption keys are needed per patient, per each electronic health record (EHR) or other entity in the system. This represents an open challenge or limitation since all presented ABE schemes requires that plaintext message, i.e. EHR data, should belong to Gt group meaning that it should be an elliptic curve point, however no discussion about mapping a message to elliptic curve point is provided. Despite existing ABE schemes that do not require plaintext messages belonging to Gt, they use XOR as an encryption function, which makes it practical to encrypt only high-entropy data, i.e., encryption keys. Also, it is hard to map encryption performance to the most common privacy phases of EHC [29]. Private Key distribution choices and threat models are also out of the scope of work in ABE for healthcare.

In the logistics industry, confidentiality protection based on ABE was used for real-life systems in [30], [31]. Gao et al. [31] adopted ciphertext-policy attribute-based encryption (CP-ABE) to encrypt segmented logistics information in different access policies with defined threat model, data flow and security properties as goals described in various attack scenarios. [32]

| Ref | CP | CC | TM | DF | AM | SP | KD |
|---|---|---|---|---|---|---|---|
| [26] | + | F | | | | | |
| [27] | F | F | | | | | |
| [28] | + | + | | | | | |
| [31] | + | | + | + | + | + | |
| [30] | + | | | + | | | |
| [32] | + | + | + | + | | + | + |

**Table 1:** Comparison of studies of real-world ABE applications separated by healthcare (41,42,40) and logistics (10,14,17)

Table 1 represents a comparison of 6 studies of real-world ABE applications in healthcare and logistics domains, and we identified that computational performance (CP) and communication costs (CC) are regular metrics measured in ABE proposals as time and number of bytes. However, some studies measure ABE performance in the amount of total number of bilinear pairings and exponentiations in the Gt field (F) for each operation of key generation, encryption, and decryption. The threat model (TM) and security properties (SP) are presented in two papers, assuming the presented application relies on the security properties of the underlying ABE scheme and its

implementation. Only two studies have presented the detailed access model (AM) and key distribution (KD). All studies for healthcare do not include detailed data flow (DF), access model, and key distribution scenario because all proposals use a single ABE encryption/decryption operation per each patient record as a solid data object.

## 2.3 ABE code libraries

For our experiment we reviewed two open-source ABE libraries listed in Table 2[1] [2]. The performance analysis, maturity in terms on maintenance status, number of supported ABE schemes of these libraries was done by Mosteiro-Sanchez et al. [21]. Here we introduce new comparison criteria for ABE libraries, Bin - library can be compiled as binary code (C++/Rust/Go languages), KEM - only key encapsulation functionality is provided, meaning that a message from Gt field should be provided for input, SYM - conventional symmetric encryption abstraction is present that allows to encrypt arbitrary data of any length, SK - session key control, AM - interface/serialization to access attributes from ciphertext metadata. ST - Stateful model to maintain the session state across multiple crypto operations. OpenABE written in C and compiled to native binary code of target Platform like ARM, X64, and x86, making computation performance comparison between the libraries more reliable when the same abstraction interface is used, KEM or KEM + SYM. Therefore we scoped out Charm. The purpose of KEM is to generate a random session key and encrypt it with ABE, later it is used in SYM layer to encrypt arbitrary size data by conventional symmetric encryption scheme, but some libraries has SYM layer integrated with KEM and thus not provide control over session key, i.e. they do not designed to re-use the same session key over a few consecutive encryption operations, which is possible in OpenAPE and Charm. OpenABE is the only library that has two abstraction interfaces for KEM and SYM and they are separated so it can be used in hybrid mode or KEM-only mode that allows re-using session key over multiple encryption operations (ST) with any symmetric encryption scheme. Also OpenABE has ciphertext metadata interface (AM) that includes both access policy and SYM information.

| Ref | Bin | KEM | SYM | SK | AM | ST |
|---|---|---|---|---|---|---|
| OpenABE | + | +/- | AEAD | + | + | + |
| Charm | - | + | - | + | N/A | - |

**Table 2:** Comparison of ABE code libraries based on RELIC code library.

The absence of a SYM layer in Charm assumes the only KEM use case. OpenABE, through the interface design, provides explicit assumptions that a new random

[1] github.com/zeutro/OpenABE
[2] github.com/JHUISI/charm

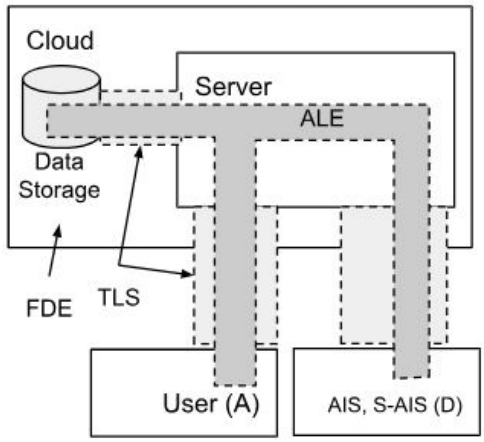


**Figure 2:** Data encryption flow (gray color) in E2EE architecture.

session key should be used per each encryption operation, and we aim to change this assumption in our study.

Comparing the performance of different KEM schemes illustrates how practical aspects of the cryptographic API design, such as the interfaces and metadata control can affect cryptographic agility in terms of dependence on mathematic library or code library and data structure. Key Encapsulation approaches to multiple Parties also exist in conventional cryptosystems [20] and it is feasible to compare it with ABE scheme. For our experiment, we simulate multiparty KEM implemented in OpenABE code library and state of art CRYSTALS-Kyber KEM to compare it by different metrics.

## 2.4 End-to-end encryption protocols

The End-to-end encryption (E2EE) approach assumes client-server architecture with an integrated encryption scheme where the “ends” of the communication (a “sender” and one or more “recipients”) can send messages to each other via an abstract central channel and where the server does not have access to the cryptographic keys necessary to read or invisibly alter the message [33] .

Figure 2 represents the data flow through transport-layer security (TLS) and data storage with full-disk encryption (FDE) in E2EE architecture. As shown, AIS device (D) uses Application Level Encryption (ALE) [34] to transfer encrypted data to the server, and later cloud storage process data in always-encrypted form. In this case, the cloud also uses encryption in-transit and encryption at-rest principles, but encryption key management for ALE belongs only to the ’ends’, i.e., device (D) and User (A).

According to the E2EE protocol design, only the ’ends’ owns the encryption and decryption keys. According to our proposal, requests and approvals to access data are controlled only by the Vendor representative (FM), who manages the encryption

key lifecycle and storage. Encryption key ownership by the client is an important requirement that distinguishes E2EE from other approaches. Before E2EE adoption, data protection architectures in cloud systems assume that the server always has access to decryption keys to process the data in plain, decrypted form.

State of practice search shows that exist commercial code libraries that implements the ALE approach, for example, LogSentinel, Cloaked Search , and AWS Crypto Tools.

To conclude, the E2EE approach is cryptography-intensive since it includes a significant amount of cryptographic code and key management logic in the ALE layer apart from the underlying cryptographic library. Depending on the mentioned challenge, the software variability issues [35] also relate to ALE as a software abstraction layer.

# 3 ABE experiment for AIS data confidentiality

In this section, we describe the proposed data protection architecture for AIS data and experiments.

In general ABE encryption scheme can be represented as four procedures: Setup, Encrypt, KeyGen, and Decrypt.

**Setup**$\rightarrow PK, MK$. Generates the public parameters PK (i.e. public key) and a master key MK. This procedure is performed once by the authority as part of the system initialization. Master Key MK should be kept secret and available only for the Authority.

**Encrypt(*PK, M, A*)**$\rightarrow CT$**.** The encryption procedure takes as input the public key PK, a message M, which should be a random member of a specific algebraic group, and an access structure A, an array of strings identifying the message's recipients. The algorithm will encrypt M and produce a ciphertext CT, an array of elements, and its size depends on the size of A.

**KeyGen(*MK, S*)**$\rightarrow SK$**.** The key generation algorithm takes the master key MK and a set of attributes S as inputs, identifying the message's recipient decryption capacity. It outputs a private key SK.

**Decrypt(*PK, CT, SK*)**$\rightarrow M$**.** The decryption algorithm takes as input the public parameters PK, a ciphertext CT, and a private key SK. If the key SK satisfies the access structure A of a ciphertext then the algorithm will decrypt the ciphertext and return a message M.

According to the description, in ABE, access structure $A$ represents the message recipients, i.e., an array of public keys of recipients, but in ABE, it may be a plain text string and ABE authority uses it to generate an access structure cryptographically embedded into each secret key and as a result the key has a flexible decryption capability, i.e. can decrypt any message if there is a match between embedded access structures in a key and ciphertext.

As a result, in ABE, different secret keys can be used to decrypt the same ciphertext. Another unique feature is that ABE can generate a new secret key that will decrypt any existing ciphertext. It means it is possible to create ciphertext without an existing secret key for decryption. This feature does not exist in conventional PKE with strongly bounded public and secret keys that start to coexist simultaneously. The recipienthe t's role in ABE is not defined so strictly as in traditional PKE,because ,during encryption , thesender specifies aan ccess structure containing a combination of attributes. This can be interpreted as when you send a message to multiple recipients but without a knowledge of actual users that will be able to decrypt or imagine a set of recipients; each has its secret key, but they all can read the same ciphertext. Key Encapsulation approaches to multiple recipients also exist in traditional cryptosystems [20] and comparing it with the ABE scheme is feasible to understand cryptographic migration capacity [13], [36]. As a second independent variable for experiment we choose CRYSTAL-Kyber [37] key encapsulation library as NIST KEM finalist for quantum-resistant cryptography [3] .

To compare with ABE scheme, here we provide main procedures of KEM based on NIST standard called ML-KEM: KeyGen, Encaps and Decaps.

**KeyGen()**$\rightarrow EK, DK$**.** The key generation algorithm takes no input and generates an encapsulation key EK and decapsulation key DK. These keys serve as public and private keys in traditional PKE.

**Encaps(*EK, M*)**$\rightarrow CT$**.** The encapsulation (i.e. encryption) procedure takes as input the encapsulation key EK identifying the message's recipients, a message M. The algorithm will encrypt M and produce a ciphertext CT of constant size .

**Decaps(*CT, DK*)** $\rightarrow M$ The decapsulation algorithm takes as input the ciphertext CT and a decapsulation key DK. The algorithm will decapsulate (i.e. decrypt) the ciphertext and return a message M.

As we see, Authority does not exist in NIST standard ML-KEM, key-pairs can be generated by any member of the system and secret key DK is capable to decrypt any message in past and future.

In the following section, we describe the proposed E2EE architecture for AIS data and experiment protocol.

## 3.1 E2EE architecture for AIS data

We assume the system Owner would like to be able to allow legitimate access to a subset of all AIS events to the respective Users according to surveillance criteria. We present new criteria for the construction of secure AIS, specifically to allow secure access capabilities by means of decryption of AIS entries of a narrow scope. For example, if the User is needed to track AIS data generated only by a subset of ship ID (IMO, MMSI numbers), or suppose a surveillance capacity is needed for all vessels

[3] https://csrc.nist.gov/pubs/fips/203/ipd

but only within a defined geographical area. The list of such criteria is provided in Table 3, where 13 items are marked to form the access criteria as ABE attribute. For such access to be considered secure, it must be impossible for an adversary to learn the content of AIS records in the cloud or in transit. We admit that the adversary may have some legitimate access given to it by the cloud infrastructure provider. We would like to ensure that, assuming he does not compromise the AIS data sources or the Owner engineering team itself, it cannot view the contents of any AIS log records.

## 3.2 AIS data components

AIS data consists of a series of records, R0, R1, . . ., Rn. Each record Ri contains seven text fields presented Table 3. In our proposal, encrypted record R contains the following fields:

1. $C, C', C_{10}...C_n$ Encapsulation data of symmetric key SK with ABE for n={10, 20, 40} access attributes, respectively.

2. $E_{SK1(F_i)}$, the symmetric encryption of the AIS fields to be logged under a key SK.

| Information type | ABE Attr | Cloud | Owner | User |
|---|---|---|---|---|
| Static data (AIS) | | | | |
| MMSI | + | + | | + |
| Name | - | - | | + |
| IMO | + | + | - | + |
| Callsign | + | + | - | + |
| VesselType | + | - | - | + |
| Dynamic data (AIS) | | | | |
| Timestamp | + | - | | + |
| Latitude | + | - | - | + |
| Longitude | + | - | - | + |
| Speed | - | - | - | + |
| Nav. status | - | - | - | + |
| Heading | - | - | - | + |
| Draught | - | - | - | + |
| Time ETA | - | - | - | + |
| Destination | + | - | - | + |
| Vessel particulars | | | | |
| Time Voyage | - | - | - | + |
| Country Flag | + | - | | + |
| Vessel Characteristics | + | - | - | + |
| Vessel Ownership | + | - | - | + |
| Vessel performance | - | - | - | + |
| AIS capture particulars | | | | |
| Sattelite ID | + | + | + | - |
| Sattelite region | + | + | + | - |

**Table 3:** AIS record fields as surveillance criteria

## 3.3 Fine-grained access control for AIS data confidentiality

In ABE, the access policy is a set of strings combined with logical operators. One example of an access policy is given below:

A = (”MMSI=0xFEAFE46AB” or ”date=01/12/2022” or ”VesselType=4174” or (”Latitude=54.5” and ”Longitude=13.0”) or ”Sattelite ID =0x72FF”)

It defines access control with forward secrecy security property since the system Owner can generate a secret key SK that will decrypt only records dated to ”01/12/2022” or about specific type of vessel .

Our experiment relies on the different KEM schemes assuming the data model and certain cryptographic API abstrations play a certain role in cryptographic agility performance in terms of cryptographic migration capacity. Comparing the performance of different KEM schemes illustrates how practical aspects of the cryptographic API design, such as the interfaces and metadata control can affect cryptographic migration ability in terms of dependence on mathematic library or code library and data structure. For the experiments, we simulate cryptographic migration from OpenABE to CRYSTAL-Kyber multiparty KEM (indcpa-enc). Key Encapsulation approaches to multiple Parties also exist in conventional cryptosystems [20] and thus can be reviewed as an alternative to ABE. Table 2 represents the real-world ship tracking database we used during the experiment.

| Description | Value |
|---|---|
| Tracking period: | 484 days |
| Total AIS Events | 1 330 169 |
| Events per hour avg. | 114 |
| Resolution | 10 Min |
| Data Size | 350 MB |
| Traffic area | 2000 $km^2$ |
| Static AIS fields | 5 |
| Dynamic AIS fields | 9 |
| Vessel Particulars | 5 |

**Table 4:** Ship tracking dataset description

In figure 1, we provide a schematical representation of AIS data sources (D1, D2), confidentiality perimeter for E2EE setup, and two types of actors: Owner and Customer. Both actors are composite and may include multiple sub-groups, but for the experiment, we limit to highlighting how the E2EE principle is applied to the Customer actor. Devices D1 and D2 capture plain AIS data and encrypt it by embedded systems before uploading to the Cloud. The owner’s task is to generate ABE keys and parameters and transfer them to devices D1 and D2 embedded systems.

According to the latest AIS review [1], vessel tracking use cases include requests by Vessel ID, country code, destination port, or specific traffic area.

The E2EE architecture for our proposed system consists of the following three entities:

- Owner: is a key generation authority (KGA) of ABE setup primarily responsible for generating ABE public parameters (PK) and master key. KGA publishes PK to software configuration of AIS embedded system of D1 and D2. KGA generates private keys (SK) for the Customers include surveillance criteria for each case of ship tracking scenario, for example based on vessel MMSI code, or coordinate area.

- Cloud: The cloud is responsible for storing the encrypted AIS data.

- Customers can access encrypted AIS stored in the cloud and can decrypt the encrypted AIS data if and only if its attribute set satisfies the access policy in the secret key (SK), i.e. AIS data matches the surveillance criteria.

AIS dataset described in table 2 contains 5 AIS fields, such as vessel MMSI, Name, IMO, and VesselType which are constant, and 9 dynamic fields, such as Latitude, Longitude, Speed, Speed, Navigation status, Heading, Draught , Destination port. According to E2EE design under simulation fields from Table 3 representing vessel identification number, coordinate, country, ownership information, etc, are employed to define access policy in ABE to create a ship tracking system with fine-grained access control. According to a dataset in Table 2, a real-world AIS system captures 114 AIS events per hour for a 2000 $km^2$ area. Users with different levels of access levels will be able to decrypt AIS data only if the AIS field value matches a specific pattern, for example, vessels of a specific type or owner, vessels only in specific traffic areas.

Simulation running flow:

- Setup CPU single thread performance rating to R=1000, 2000, 4000.

- Encapsulate 256-bit session key SK with CP-ABE, KP-ABE with OpenABE library for I =10, 20, 40 access attributes representing AIS field values. According to table 2 and AIS specification [7], we set the maximum length of the attribute to 30 bits.

- Encapsulate 256-bit session key SK with Kyber key encapsulation library (indcpa enc) for I =10, 20, 40 encapsulation keys, referred as public key.

All tests were performed on the virtual host with Ubuntu OS with three configurations of virtual CPU with single-thread performance rating [4] defined from 1000 to

[4] www.cpubenchmark.net/singleThread.html

4000, thus simulating three different CPUs starting from low-performance CPU up to Intel Core i9-12900F.

# 4 Results

Figures 3,4 , 5 represent the comparison of key encapsulation performance between ABE and PQC Kyber KEM mechanisms, to delegate access structure based on 10, 20, 40 access attributes in ABE or public keys in Kyber respectively. Figure 4 represents performance results for Kyber in a separate graph to see its timings more precisely due to faster speed over ABE. All experimental evaluations were performed on three different vCPU configurations defined by single thread performance rating, complemented by 16GB RAM. The implementation of our simulation was realized using the x64 binaries compiled in the Ubuntu 22 system with the default Make file coming for each code library. Results in table 3 shows that KEM in OpenABE produces composite ciphertext material (Comm. Cost - communication cost) with associated data including access policy, this highlight the requirement of necessary abstraction layer to access metadata (AM) in the ciphertext. Opposite to it, Kyber KEM metadata includes only public key bytes assuming the metadata information will be coded directly by the higher code abstraction layers. Both libraries include integrated symmetric encryption abstraction (SYM) that allows the encryption of arbitrary data of any length alongside an encapsulated session key. ( SK - session essential control) Kyber KEM was implemented in a stateful model (ST) to maintain the session key across multiple crypto operations. Communication cost represents the required storage size for ciphertext, including metadata per each log record.

**Table 5:** Comparison of KEM criteria and performance in OpenABE and Kyber code libraries.

| Criteria | OpenABE | Kyber |
|---|---|---|
| SYM | AES | Any |
| Mathematic | RELIC BN P254 | LWE Lattices 1024 |
| SK | + | + |
| AM | + | - |
| ST | +/- | + |
| Communication cost in bytes | CP-ABE: 80+140*n KP-ABE: 100+55*n | 1568 * n |
| Key size, in bits. | 2080 | 512 |

The results demonstrated that ABE key encapsulation takes 100 times more time for each AIS record compared to the Kyber mechanism in CPU with a rating of 1000 and 50 times more on faster CPU with a rating of 4000. However KP-ABE is the most effective in terms of the small size of the encapsulated key size (communication costs). The overhead on the size of the encrypted AIS Record, according to Table 3, estimates 650 bytes as the minimum for ten access attributes for the key encapsulation part only (KP-ABE), not taking into account the second part with record fields in plain

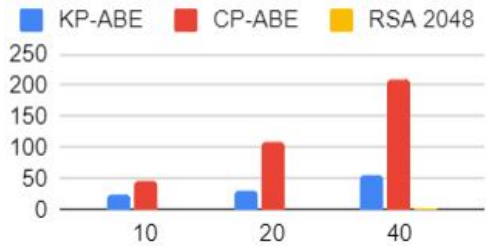


**Figure 3:** ABE key encapsulation performance in ms. on vCPU with R=1000 single thread performance rating, for 10, 20, 40 access attributes.

.

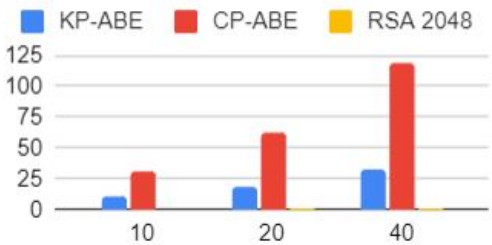


**Figure 4:** ABE key encapsulation performance in ms. on vCPU with R=2000 single thread performance rating, for 10, 20, 40 access attributes.

.

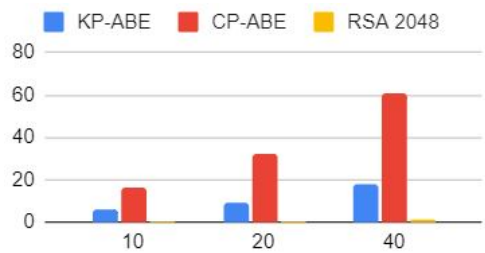


**Figure 5:** ABE key encapsulation performance in ms. on vCPU with R=4000 single thread performance rating, for 10, 20, 40 access attributes.

.

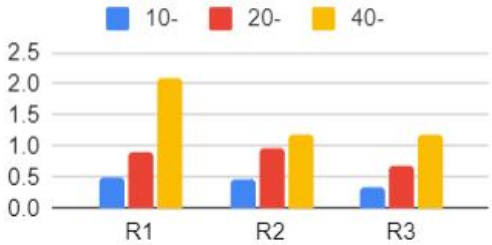


**Figure 6:** Kyber-1024 multi-recipient key encapsulation performance in ms. on vCPU with R1=1000, R2=2000, R3=4000 single thread performance rating, for 10, 20, 40 public keys.

.

text encrypted with the symmetric key. This means that in practice, for an average AIS record of size 30 bytes, the size of the encrypted log record increases to 650 + 30 = 680 bytes.

# 5 Discussion

In the proposed approach for AIS data confidentiality, asymmetric KEM provides a security advantage since each deployed AIS device only stores public key parameters, and there are no secret keys in the AIS or S-AIS embedded system for an attacker to steal. Compromising a server does not allow the attacker to decrypt historical IAS data. A drawback of this approach is the performance overhead. We note that this approach is also straightforward and can be modified to decrease performance overhead. Multiple Symmetric Keys need to be introduced for each AIS event to do this. Symmetric Key(s) can be reused in various AIS records to minimize the overhead on computation and communication costs.

# 6 Conclusions and future research

This paper introduces a AIS data confidentiality approach based on end-to-end encryption. It explores various challenges, with a central emphasis on applying the KEM approach to satellites of other coastal AIS capture devices with embedded computation systems. An examination of the existing methods for AIS data confidentiality was performed and identified the necessity of further research for public-key cryptography applications in the domain. Our study provides strong evidence supporting the feasibility and efficiency of the proposed approach for cryptographic agility in E2EE systems. This research paves the way for future developments in secure system design, offering a scalable solution that can adapt to the evolving landscape of cybersecurity threats and the continuous advancement of cryptographic standards.

# References


[1] T. Emmens, C. Amrit, A. Abdi, and M. Ghosh. “The promises and perils of Automatic Identification System data”. In: *Expert Systems with Applications* 178 (Sept. 2021), p. 114975. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2021.114975. URL: https://www.sciencedirect.com/science/article/pii/S0957417421004164 (visited on 03/06/2024).

[2] S. Levy, E. Gudes, and D. Hendler. “A Survey of Security Challenges in Automatic Identification System (AIS) Protocol”. en. In: *Cyber Security, Cryptology, and Machine Learning*. Ed. by S. Dolev, E. Gudes, and P. Paillier. Cham: Springer Nature Switzerland, 2023, pp. 411–423. ISBN: 978-3-031-34671-2. DOI: 10.1007/978-3-031-34671-2_29.

[3] O. Soner, G. Kayisoglu, P. Bolat, and K. Tam. “Risk sensitivity analysis of AIS cyber security through maritime cyber regulatory frameworks”. In: *Applied Ocean Research* 142 (Jan. 2024), p. 103855. ISSN: 0141-1187. DOI: 10.1016/j.apor.2023.103855. URL: https://www.sciencedirect.com/science/article/pii/S0141118723003966 (visited on 05/01/2024).

[4] A. Aziz, P. Tedeschi, S. Sciancalepore, and R. Pietro. *SecureAIS - Securing Pairwise Vessels Communications*. June 2020. DOI: 10.1109/CNS48642.2020.9162320.

[5] G. Wimpenny, J. Šafář, A. Grant, and M. Bransby. “Securing the Automatic Identification System (AIS): Using public key cryptography to prevent spoofing whilst retaining backwards compatibility”. en. In: *The Journal of Navigation* 75.2 (Mar. 2022). Publisher: Cambridge University Press, pp. 333–345. ISSN: 0373-4633, 1469-7785. DOI: 10.1017/S0373463321000837. (Visited on 02/14/2024).

[6] A. Goudosis and S. Katsikas. “Secure AIS with Identity-Based Authentication and Encryption”. en. In: *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation* 14.2 (2020), pp. 287–298. ISSN: 2083-6473. DOI: 10.12716/1001.14.02.03. URL: http://www.transnav.eu/Article_Secure_AIS_with_Identity-Based_Goudosis,54,1003.html (visited on 11/01/2023).

[7] A. Goudosis and S. Katsikas. “Secure Automatic Identification System (SecAIS): Proof-of-Concept Implementation”. en. In: *Journal of Marine Science and Engineering* 10.6 (June 2022). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, p. 805. ISSN: 2077-1312. DOI: 10.3390/jmse10060805. URL: https://www.mdpi.com/2077-1312/10/6/805 (visited on 11/01/2023).

[8] R. E. Litts, D. C. Popescu, and O. Popescu. “Authentication Protocol for Enhanced Security of the Automatic Identification System”. In: *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. May 2021, pp. 1–6. DOI: 10.1109/BlackSeaCom52164.2021.9527840. URL: https://ieeexplore.ieee.org/document/9527840 (visited on 05/01/2024).

[9] S. Sciancalepore, P. Tedeschi, A. Aziz, and R. Pietro. “Auth-AIS: Secure, Flexible, and Backward-Compatible Authentication of Vessels AIS Broadcasts”. In: *IEEE Transactions on Dependable and Secure Computing* PP (Mar. 2021), pp. 1–1. DOI: 10.1109/TDSC.2021.3069428.

[10] W. R. Cairns. “AIS and Long Range Identification & Tracking”. en. In: *The Journal of Navigation* 58.2 (May 2005). Publisher: Cambridge University Press, pp. 181–189. ISSN: 1469-7785, 0373-4633. DOI: 10.1017/S0373463305003267. URL: https://www.cambridge.org/core/journals/journal-of-navigation/article/ais-and-long-range-identification-tracking/62FB1C2B05A023D9F794F21AA639A83D (visited on 02/14/2024).

[11] G. Longo, M. Martelli, E. Russo, A. Merlo, and R. Zaccone. “Adversarial waypoint injection attacks on Maritime Autonomous Surface Ships (MASS) collision avoidance systems”. In: *Journal of Marine Engineering & Technology* 0.0 (2023), pp. 1–12. ISSN: 2046-4177. DOI: 10.1080/20464177.2023.2298521. (Visited on 02/11/2024).

[12] A. Silonosov and L. Henesey. “Telemetry data sharing based on Attribute-Based Encryption schemes for cloud-based Drone Management system”. In: *Proceedings of the 19th International Conference on Availability, Reliability and Security*. ARES ’24. New York, NY, USA: Association for Computing Machinery, July 2024, pp. 1–8. ISBN: 9798400717185. DOI: 10.1145/3664476.3670909. URL: https://dl.acm.org/doi/10.1145/3664476.3670909 (visited on 08/25/2024).

[13] A. Silonosov, O. Baranovskyi, and L. Henesey. “Poster: Towards cryptographic agility in end-to-end encryption systems for computer generated telemetry data.” In: *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*. HOTMOBILE ’24. New York, NY, USA: Association for Computing Machinery, Feb. 2024, p. 144. ISBN: 9798400704970. DOI: 10.1145/3638550.3643624. URL: https://dl.acm.org/doi/10.1145/3638550.3643624 (visited on 04/08/2024).

[14] R. Challamel, T. Calmettes, and C. N. Gigot. “A European hybrid high performance Satellite-AIS system”. In: *2012 6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC)*. ISSN: 2326-5949. Sept. 2012, pp. 246–252. DOI:

10.1109/ASMS-SPSC.2012.6333084. URL: https://ieeexplore.ieee.org/abstract/document/6333084?casa_token=xtCzunGAZpgAAAAA:ZN6fGlKlMb_hPqrrrYbYejnb8Gl4cTaA6Nrj2_BxWMMFgUHjqOletzgt-1J502aem5a-GjRlMBM (visited on 05/14/2024).

[15] M. Rasori, M. L. Manna, P. Perazzo, and G. Dini. "A Survey on Attribute-Based Encryption Schemes Suitable for the Internet of Things". In: *IEEE Internet of Things Journal* 9.11 (June 2022). Conference Name: IEEE Internet of Things Journal, pp. 8269–8290. ISSN: 2327-4662. DOI: 10.1109/JIOT.2022.3154039. URL: https://ieeexplore.ieee.org/abstract/document/9721417 (visited on 05/04/2024).

[16] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng. "Attribute-based Encryption for Cloud Computing Access Control: A Survey". In: *ACM Computing Surveys* 53.4 (Aug. 2020), 83:1–83:41. ISSN: 0360-0300. DOI: 10.1145/3398036. URL: https://dl.acm.org/doi/10.1145/3398036 (visited on 05/04/2024).

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters. *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data*. Publication info: Published elsewhere. Extended abstract to appear in ACM CCS 2006. This is the full version. 2006. URL: https://eprint.iacr.org/2006/309 (visited on 10/31/2023).

[18] B. Waters. *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*. Publication info: Published elsewhere. Unknown where it was published. 2008. URL: https://eprint.iacr.org/2008/290 (visited on 02/18/2024).

[19] A. Sahai and B. Waters. *Fuzzy Identity Based Encryption*. Publication info: Published elsewhere. Unknown where it was published. 2004. URL: https://eprint.iacr.org/2004/086 (visited on 02/22/2024).

[20] N. P. Smart. "Efficient Key Encapsulation to Multiple Parties". en. In: *Security in Communication Networks*. Ed. by C. Blundo and S. Cimato. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 208–219. ISBN: 978-3-540-30598-9. DOI: 10.1007/978-3-540-30598-9_15.

[21] A. Mosteiro-Sanchez, M. Barceló, J. Astorga, and A. Urbieta. *All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of CP-ABE Libraries*. Sept. 2022.

[22] S. Mhatre, A. V. Nimkar, and S. N. Dhage. "Comparative study on attribute-based encryption for health records in cloud storage". In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (May 2017). Conference Name: 2017

2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) ISBN: 9781509037049 Place: Bangalore Publisher: IEEE, pp. 647–652. DOI: 10.1109/RTEICT.2017.8256677.

[23] R. Imam, K. Kumar, S. M. Raza, R. Sadaf, F. Anwer, N. Fatima, M. Nadeem, M. Abbas, and O. Rahman. "A systematic literature review of attribute based encryption in health services". In: *Journal of King Saud University - Computer and Information Sciences* 34.9 (Oct. 2022), pp. 6743–6774. ISSN: 1319-1578. DOI: 10.1016/j.jksuci.2022.06.018. URL: https://www.sciencedirect.com/science/article/pii/S1319157822002269 (visited on 02/15/2024).

[24] C. Wang, X. Liu, and W. Li. *Implementing a Personal Health Record Cloud Platform Using Ciphertext-Policy Attribute-Based Encryption*. Journal Abbreviation: Proceedings of the 2012 4th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2012 Pages: 14 Publication Title: Proceedings of the 2012 4th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2012. Sept. 2012. ISBN: 978-1-4673-2279-9. DOI: 10.1109/iNCoS.2012.65.

[25] S. Porwal and S. Mittal. "Implementation of Ciphertext Policy-Attribute Based Encryption (CP-ABE) for fine grained access control of university data". en. In: *2017 Tenth International Conference on Contemporary Computing (IC3)*. Noida: IEEE, Aug. 2017, pp. 1–7. ISBN: 978-1-5386-3077-8. DOI: 10.1109/IC3.2017.8284289.

[26] X. Zhao and Q. Su. "Revocable Attribute-Base Scheme with Enhanced Security and Privacy for Healthcare Data Sharing". en. In: *2023 IEEE 14th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*. Beijing, China: IEEE, Nov. 2023, pp. 1–8. ISBN: 9798350371024. DOI: 10.1109/PAAP60200.2023.10391571. URL: https://ieeexplore.ieee.org/document/10391571/ (visited on 02/07/2024).

[27] Y. Zhao, P. Fan, H. Cai, Z. Qin, and H. Xiong. "Attribute-based Encryption with Non-Monotonic Access Structures Supporting Fine-Grained Attribute Revocation in M-healthcare". In: *Int. J. Netw. Secur.* (2017). (Visited on 02/12/2024).

[28] J. Zhao, K. Zhang, J. Gong, and H. Qian. "Lavida: Large-Universe, Verifiable, and Dynamic Fine-Grained Access Control for E-Health Cloud". In: *IEEE Transactions on Information Forensics and Security* 19 (2024). Conference Name: IEEE Transactions on Information Forensics and Security, pp. 2732–2745. ISSN: 1556-6021. DOI: 10.1109/TIFS.2024.3350925. URL: https://ieeexplore.ieee.org/document/10382677 (visited on 02/12/2024).

[29] A. V, A. P. Nirmala, B. K, A. Christi, and N. A. “A Review on Cloud Cryptography Techniques to Improve Security in E-health Systems”. In: *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. Mar. 2022, pp. 100–104. DOI: 10.1109/ICCMC53470.2022.9753999. URL: https://ieeexplore.ieee.org/document/9753999 (visited on 02/12/2024).

[30] S. Hong, H. Pan, Y. Fang, J. Ma, X. Qi, and Y. Hu. “A Logistics Privacy Protection Scheme Based on Ciphertext Policy Attribute-Based Key Encapsulation”. In: *2022 International Conference on Blockchain Technology and Information Security (ICBCTIS)*. July 2022, pp. 218–224. DOI: 10.1109/ICBCTIS55569.2022.00057. URL: https://ieeexplore.ieee.org/document/9845073 (visited on 02/12/2024).

[31] Q. Gao, J. Zhang, J. Ma, C. Yang, J. Guo, and Y. Miao. “LIP-PA: A Logistics Information Privacy Protection Scheme with Position and Attribute-Based Access Control on Mobile Devices”. en. In: *Wireless Communications and Mobile Computing* 2018 (July 2018). Publisher: Hindawi, e9436120. ISSN: 1530-8669. DOI: 10.1155/2018/9436120. URL: https://www.hindawi.com/journals/wcmc/2018/9436120/ (visited on 02/12/2024).

[32] T. Li, R. Zhang, and Y. Zhang. “PriExpress: Privacy-preserving express delivery with fine-grained attribute-based access control”. In: *2016 IEEE Conference on Communications and Network Security (CNS)*. Oct. 2016, pp. 333–341. DOI: 10.1109/CNS.2016.7860501. URL: https://ieeexplore.ieee.org/document/7860501 (visited on 02/15/2024).

[33] S. Scheffler and J. Mayer. “SoK: Content Moderation for End-to-End Encryption”. en. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (Apr. 2023), pp. 403–429. ISSN: 2299-0984. DOI: 10.56553/popets-2023-0060. URL: https://petsymposium.org/popets/2023/popets-2023-0060.php (visited on 10/30/2023).

[34] Y. Ding and K. Klein. “Model-Driven Application-Level Encryption for the Privacy of E-health Data”. In: *2010 International Conference on Availability, Reliability and Security*. Feb. 2010, pp. 341–346. DOI: 10.1109/ARES.2010.91. URL: https://ieeexplore.ieee.org/abstract/document/5438071 (visited on 04/08/2024).

[35] M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch. “COVAMOF: A Framework for Modeling Variability in Software Product Families”. English. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3154 (2004). ISBN: 9783540229186, pp. 197–213. ISSN: 0302-9743. DOI: 10.1007/978-3-540-28630-1_12.

[36] D. Sikeridis, D. Ott, S. Huntley, S. Sharma, V. K. Dhanasekar, M. Bansal, A. Kumar, A. U. N, D. Beveridge, and S. Veeraswamy. *ELCA: Introducing Enterprise-level Cryptographic Agility for a Post-Quantum Era*. Publication info: Preprint. 2023. URL: https://eprint.iacr.org/2023/1539 (visited on 10/12/2023).

[37] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. London: IEEE, Apr. 2018, pp. 353–367. ISBN: 978-1-5386-4228-3. DOI: 10.1109/EuroSP.2018.00032. URL: https://ieeexplore.ieee.org/document/8406610/ (visited on 07/19/2024).

# Paper III

# The role of key encapsulation mechanism in end-to-end encryption agility - a case for telemetry data confidentiality.

*Alexandr Silonosov, Lawrence Henesey*
*In: 14th International Conference on Industry Sciences and Computer Science Innovation (ISCSI 2024), Porto Oct 29, 2024, Elsevier Publishing, 2024.*

Abstract

The study outlined in this paper assesses the implementation of Attribute-Based Encryption as a key encapsulation mechanism, culminating in a proposed end-to-end encryption solution for a cloud-based IoT management system and further simulation of migration to Kyber Post-Quantum Cryptography (PQC) key encapsulation mechanism. Though extensively used for efficiently gathering and sharing video surveillance data, these systems also collect telemetry information, which includes sensitive data. This paper presents a comprehensive study that examines the current understanding, methodologies, and challenges associated with supporting the End-to-End Encryption (E2EE) approach to ensure the confidentiality of telemetry data.

## 1 Introduction

In this paper, we study the feasibility and advantages of implementing the end-to-end encryption (E2EE) approach [1] for telemetry data confidentiality, particularly in addressing real-world data confidentiality challenges in cloud-based IoT systems.

This paper first introduces the experimentation approach of comparing different key encapsulation mechanisms (KEM) for telemetry data confidentiality. Our study shows the type of confidential information captured by the telemetry audit log in the cloud-based IoT system. State of practice shows cases when telemetry data was a valuable target of cyber-criminals; therefore, it is feasible to investigate confidentiality and protection controls for such data. Our contribution to this study is two-fold. First,

we introduce forward secrecy with Attribute-Based Encryption (ABE) for telemetry data confidentiality, second we contribute to crypto-agility research frontier [2] by providing experimental results of cryptographic migration performance from ABE to Kyber KEM for telemetry data protection.

The paper is structured as follows: The background section provides information related to problems arising from plain-text telemetry data logs, along with a proposal for comparing two types of encryption key encapsulation mechanisms. Section 2 encompasses a review of related work. Section 3 outlines the method and experiments. Section 4 discusses the results, while Section 6 concludes the paper with remarks and provides pointers for future research.

## 1.1 Background

Software Vendors and IoT manufacturers want to improve user experience with their products and therefore collect various product-related performance data such as error rates, resources use, accuracy of operations, delays in functioning, internal state of integrated components [3, 4]. Our work [5] demonstrated that payload strings in telemetry data contain confidential information such as facility coordinates, and authentication information to access cloud resources. Lyons et al. [6] generalized the problem by demonstrating that most devices create telemetry logs that include some amount of user-identifying information and discovered how 3rd party libraries could unintentionally generate telemetry data into the system logs.

The problem has two aspects, if telemetry data is not encrypted, I) a successful cyber attack on IoT vendor engineer's computer may lead to an information leakage of IoT user and vise-versa II) by successfully attacking IoT devices attackers may retrieve authentication information to access cloud server. Given that the payload data may be sensitive, it is feasible to investigate how encrypting it may influence the performance of IoT software and IoT cloud architecture. Recent supply chain attack on Cisco DUO multi-factor authentication service revealed how plain-text telemetry logs lead to DUO customer sensitive information leak [1].

Waters et al. [7] first proposed a solution for the problem by using Identify Based Encryption scheme, later Goyal et al. [8] proposed an application of an Attribute-Based Encryption scheme to encrypt audit logs of system activity. However, neither study assumed forward secrecy property and provided detailed performance evaluations and comparisons with quantum-safe public key encryption schemes.

[1] https://www.securityweek.com/cisco-duo-says-hack-at-telephony-supplier-exposed-mfa-sms-logs/

# 2 Related work

Key-Policy ABE[8] (2006) and Ciphertext-Policy ABE [9] (2008) are public key cryptosystem, a successor of Fuzzy Identity Based Encryption (IBE) [10], that allows to encrypt a message by using a combination of public keys, referred as access structure, instead of a single public key in conventional PKE systems like RSA. ABE cryptosystem consist of four algorithms: Setup, Keygen, Encrypt, Decrypt.

A range of attribute-based encryption (ABE) schemes have been proposed in healthcare for medical records data confidentiality and secure sharing [11], [12], [13], [14], [15], [16] two real-world implementations of ABE [17], [18] and none of the business implementation as a state of practice and readiness indicator.

In logistics information system privacy protection based on ABE was proposed by [19], [20], [20], [21]

Table 1 represents a comparison of 6 studies of real-world ABE applications in healthcare and logistics domains, and we identified that computational performance (CP) and communication costs (CC) are regular metrics measured in ABE proposals as time and number of bytes. However, some studies measure ABE performance in the amount of total number of bilinear pairings and exponentiations in the algebraic field for each operation of key generation, encryption, and decryption. The threat model (TM) and security properties (SP) are presented in two papers, assuming the presented application relies on the security properties of the underlying ABE scheme and its implementation. Detailed access model (AM) and key distribution (KD) are presented only in two studies. All studies for healthcare do not include detailed data flow (DF), access model, and key distribution scenario because all proposals use a single ABE encryption/decryption operation per each patient record treated as a solid data object.

**Table 1:** Comparison of studies of real-world ABE applications.

| Ref | CP | CC | TM | DF | AM | SP | KD |
|---|---|---|---|---|---|---|---|
| [13] | + | F | | | | | |
| [14] | F | F | | | | | |
| [15] | + | + | | | | | |
| [20] | + | | + | + | + | + | |
| [19] | + | | | + | | | |
| [21] | + | + | + | + | | + | + |

## 2.1 Key Encapsulation Mechanism

Public Key Encryption (PKE) and Key Encapsulation Mechanisms (KEM) are public-key cryptosystems, contains tuple of algorithms: Keygen, Encrypt, Decrypt. The introduction of KEMs as a separate concept from PKE became necessary for several reasons such as : efficiency in hybrid encryption in case of large messages, modularity,

security against adaptive attacks (such as chosen ciphertext attack) [22]. One of the main difference between PKE and KEM is that in KEM sender could not choose a message for encryption. Instead of it, KEM-Encrypt algorithm randomly generates a message for the sender, conforming security requirements of underlying algebra referred as 'secret', and encrypt it , i.e. encapsulates by using public key of the recipient [23]. The sender uses the provided secret message as a session encryption key to encrypt data of arbitrary length by using a traditional symmetric encryption algorithm.

Smart et al. [24] presented a notion of multiple recipients KEM (mKEM) to propose a solution for traditional two-party setting limitation in PKE. Alwin et al. [25] proposed multi-message multi-recipient (mmKEM) construction based on the modifications of CRYSTALS-Kyber and demonstrated 78% savings for the ciphertext size against the naive approach, i.e., executing Kyber encapsulation for a given number of recipients. As the side effect mmKEM requires additional parameters per each set of recipients referred as seed and introduced composite ciphertext structure.

# 3 Key encapsulation experiment for telemetry data protection

OpenABE code library implements KP-ABE, CP-ABE and provides key encapsulation (KEM) functionality as a separate abstraction layer [5] Thus, it is possible to design experiments comparing different KEM for performance and ciphertext size. For our experiment, we simulate migration from multiparty KEM implemented in OpenABE code library to CRYSTALS-Kyber KEM, thus to understand the overhead on performance, communication costs and ciphertext metadata. For the results to be closer to cloud-based taxonomy for virtual CPU time, it is feasible to measure performance in time related to CPU single-thread performance rating [26], [27], [28].

In general ABE encryption scheme can be represented as four procedures: Setup, Encrypt, KeyGen, and Decrypt.

**Setup**$\rightarrow PK, MK$. Generates the public parameters PK (i.e. public key) and a master key MK. This procedure is performed once by the authority as part of the system initialization. Master Key MK should be kept secret and available only for the Authority.

**Encrypt(*PK, M, A*)**$\rightarrow CT$**.** The encryption procedure takes as input the public key PK, a message M, which should be a random member of a specific algebraic group, and an access structure A, an array of strings identifying the message's recipients. The algorithm will encrypt M and produce a ciphertext CT, an array of elements, and its size depends on the size of A.

**KeyGen(*MK, S*)**$\rightarrow SK$**.** The key generation algorithm takes the master key MK and a set of attributes S as inputs, identifying the message's recipient decryption

capacity. It outputs a private key SK.

**Decrypt(*PK, CT, SK*)**$\rightarrow M$ The decryption algorithm takes as input the public parameters PK, a ciphertext CT, and a private key SK. If the key SK satisfies the access structure A of a ciphertext then the algorithm will decrypt the ciphertext and return a message M.

According to the description, in ABE, access structure $A$ represents the recipients, i.e., an array of public keys of recipients, but in ABE, it may be a plain text string and ABE authority uses it to generate an access structure cryptographically embedded into each secret key or message and as a result the key has a flexible decryption capability, i.e. can decrypt any message if there is a match between embedded access structures in a secret key and ciphertext.

To compare with ABE scheme, here we provide main procedures of KEM based on NIST standard called ML-KEM: KeyGen, Encaps and Decaps.

**KeyGen()**$\rightarrow EK, DK$**.** The key generation algorithm takes no input and generates an encapsulation key EK and decapsulation key DK. These keys serve as public and private keys in traditional PKE.

**Encaps(*EK, M*)**$\rightarrow CT$**.** The encapsulation (i.e. encryption) procedure takes as input the encapsulation key EK identifying the message's recipients, a message M. The algorithm will encrypt M and produce a ciphertext CT of constant size .

**Decaps(*CT, DK*)** $\rightarrow M$ The decapsulation algorithm takes as input the ciphertext CT and a decapsulation key DK. The algorithm will decapsulate (i.e. decrypt) the ciphertext and return a message M.

As we see, Authority does not exist in NIST standard ML-KEM, key-pairs can be generated by any member of the system and secret key DK is capable to decrypt any message for DK in past and future.

In the following section, we describe the proposed E2EE architecture for telemetry logs and experiments.

## 3.1 E2EE architecture for telemetry data

Figure 1 represents the proposed End-to-End Encryption approach for telemetry data in cloud based IoT system. For the proposed E2EE architecture, software appliance (Sw) employs Application Level Encryption (ALE) [29] to encrypt telemetry data generated from IoT hardware appliance (H) and its own to encrypt telemetry data. As shown, Software (Sw) application transfers the encrypted data in the gray ALE area to the server, and later, ALE-encrypted data arrives at cloud storage in always-encrypted form. In this case, the cloud also uses encryption in-transit (TLS) and encryption at-rest (FDE) implementations, but encryption key management for ALE belongs only to the 'ends', i.e., software (Sw) and engineering teams (T) at vendor side.

We assume the Vendor (Authority in terms of ABE) would like to be able to allow legitimate access to a subset of all telemetry log entries to the respective engineering

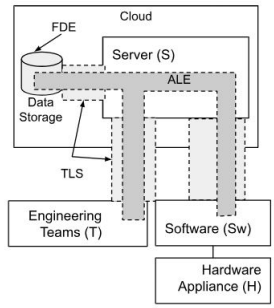


**Figure 1:** E2EE architecture for telemetry data generators (H, Sw) and engineering team (T) .

team or even individual engineers according to some criteria. In the architecture presented in Figure 1, the telemetry logs are encryped on IoT device (H + Sw) and should kept encrypted all the time in cloud storage, including the engineering team (T) computer and decrypted only after vendor authority generates a secret key. We present new criteria for the construction of secure telemetry logs, specifically to allow the secure access capabilities by means of decryption of log entries of a narrow scope. For example, if an engineering team dedicated to image recognition needed to investigate records generated only by image recognition code assets, or suppose a first tier-technical support department need to access Module name field from all of records generated by a specific customer ID or specific date. Another example is when Vendor must provide access for all telemetry records generated during a particular date only, thus limiting the decryption capacity for past or future data records. For such access to be considered secure, it must be impossible for an adversary to learn the content of telemetry records. We admit that the adversary can be an insider because it may have access to User on-premises assets [5] or may have some legitimate access given to it by the cloud infrastructure provider. Assuming the attacker does not compromise the Vendor engineering team itself, we would like to ensure that it cannot view the contents of any telemetry log record.

## 3.2 Telemetry data model

According to our case study [30], telemetry data consists of a series of log files (F0, F1, ..Fn) each file representing a IoT device operational session and contains a set of records R1, . . ., Rn. Each record Ri contains seven text fields presented Table 2. In our proposal, encrypted log file Fn begins with special Record R0 and then contains the records R1, R2 ... Rn:

1. $R0 = CT = C, C', C_{10}...C_n$ Encapsulation of random symmetric key SK with ABE scheme for n={10, 20, 40} access attributes.

2. $R1 = F1, F2, F3, F4, E_{SK}(F_{i=5..7})$, the symmetric encryption of the telemetry fields (i=5..7) to be logged under a key SK.

3. $R2 = F1, F2, F3, F4, E_{SK}(F_{i=5..7})$, $Rn = F1, F2, F3, F4, E_{SK}(F_{i=5..7})$,

## 3.3 Fine-grained access control and forward secrecy for telemetry data

In ABE, the access policy is a set of strings combined with logical operators. One example of an access policy is given below:

A = (”CustomerID=0xFEAFE46FF” or ”date=01/11/2024” or”event=7445” or ”level=info” ”Asset=Image Recognition” or ”version=4.2.09” or ”IoT-CloudID=0xFEABE46FF” or ”op-mode=plain”)

It defines access control with forward secrecy security property since the Authority can generate a secret key SK that will decrypt only records dated to ”01/11/2024”.

Our experiment relies on the different KEM schemes, assuming the data model and certain cryptographic API abstrations play a specific role in cryptographic agility performance regarding cryptographic migration capacity. Comparing the performance of different KEM schemes illustrates how practical aspects of the cryptographic API design, such as the interfaces and metadata control, can affect cryptographic agility in terms of dependence on mathematic library or code library and data structure. For the experiment, we simulate multiparty KEM with OpenABE and Kyber (indcpa-enc) code libraries for the first part of encrypted record R. The symmetric encryption for fields F5..7 of telemetry data records R1, R2 .. Rn is out of scope in this experiment. Key Encapsulation approaches to multiple recipients also exist in traditional cryptosystems [24] and comparing it with the ABE scheme is feasible to understand cryptographic migration capacity [31], [32]. As a second independent variable for experiment we choose CRYSTAL-Kyber [33] key encapsulation library as NIST KEM finalist for quantum-resistant cryptography [2] .

In the figure 1, we provide a schematically representation of ends (Sw and T) in E2EE architecture. In our experiment, we consider the IoT appliance (H, Sw) without user-provided encryption key management, and therefore, E2EE encryption starts in Software (Sw). Initially, IoT device generates plain data, unencrypted and encrypted by (Sw) before uploading it to the Cloud. For telemetry data, we propose the implementation of Application Level Encryption (ALE) on side (T) and (Sw).

[2]https://csrc.nist.gov/pubs/fips/203/ipd

**Table 2:** Telemetry data record fields

| Id | Description | Value example |
|---|---|---|
| F1 | Date | 01-01-2024 |
| F2 | Time | 12:00:00 245 |
| F3 | Event type | Info, Debug, Error |
| F4, | Process session id | System context number |
| F5 | Module name | Software asset name |
| F6 | Message | Performance indicator value |
| F7 | Payload | Other specific value |

The E2EE architecture for the proposed approach consists of the following four entities:

- Vendor: a key generation authority (KGA) of ABE setup primarily responsible for generating ABE public parameters (PK) and master key. KGA publishes PK to software development configuration as part of the telemetry library. KGA generates private keys (SK) for the engineering teams for each case of user experience improvement.

- Cloud: The Cloud is responsible for storing the encrypted telemetry data.

- The engineering team can access encrypted telemetry data stored in the Cloud and can decrypt a subset of it only if its attribute set satisfies the access policy in the private key (SK).

- The IoT device appliance that generates encrypted telemetry data.

The telemetry dataset described in table 2 contains seven fields. According to the proposed architecture, fields F2 and F6 define access policy for ABE key encapsulation for fine-grained access control. Our case study [30] identified that information such as IoT model, identity number, customer cloud identity ID, customer location, and software version number can also be used as access attributes in ABE. Engineers with different levels of access or purposes of user experience investigation will be able to decrypt log records data only if the telemetry field values match specific patterns, for example, logs in a particular date (F2), logs from a specific software module (F5) or version.

Simulation running flow:

- Setup CPU single thread performance rating to R=1000, 2000, 4000.

- Encapsulate 256-bit session key SK in CP-ABE [34]) and KP-ABE scheme by using OpenABE library for I =10, 20, 40 access attributes representing an arbitrary telemetry field values.

- Decapsulate the ciphertext from previous step for the same settings with OpenABE library.
- Encapsulate 256-bit session key SK with Kyber key encapsulation library (indcpa enc, 1024 security level) for I =10, 20, 40 public keys.

All tests were performed on the virtual host with Ubuntu 22 OS complemented by 16GB RAM, with three configurations of vCPU with single-thread performance rating [3] defined from 1000 to 4000, measured by linux command line "sysbench cpu –threads=1 –time=10 run", thus simulating three different vCPU starting from low-performance CPU up to Intel Core i9-12900F. The implementation of our simulation was realized using the x64 binaries compiled in the Ubuntu 22 system with the default Make file coming for each code library.

# 4 Results

Figure 2 represents the comparison of key encapsulation performance between OpenABE and PQC Kyber mechanisms to delegate access structure based on 10, 20, and 40 access attributes in ABE or public keys in Kyber, respectively. Figure 4 represents performance results for Kyber in a separate graph to see its timings more precisely due to significantly better performance over ABE. Figure 3 represents the comparison of key decapsulation performance between OpenABE and PQC Kyber. Results in table 3 shows that KEM in OpenABE produces composite ciphertext material with associated data, including access policy; this highlights the requirement of necessary abstraction layer to access metadata (AM) in the ciphertext. Opposite to it, Kyber KEM metadata includes only public-key bytes, assuming the metadata information will be coded directly by the higher code abstraction layers. OpenABE include integrated symmetric encryption abstraction (SYM) that allows the encryption of arbitrary data of any length alongside an encapsulated session key ( SK - session essential control) Kyber KEM was implemented in a stateful model (ST) to maintain the session key across multiple crypto operations. Ciphertext size comparison represents that Kyber requires at least 8 times more storage size for the single telemetry record R0 then OpenABE.

The results demonstrate time required for ABE key encapsulation for record R0, takes 100 times more time compared to the Kyber in CPU with a rating of 1000 and 50 times more on faster CPU with a rating of 4000. However, KP-ABE is the most effective regarding the small size of the encapsulated key (communication costs). The overhead on the size of the encrypted log Record R0, according to Table 3, estimates 650 bytes as the minimum for ten access attributes for the key encapsulation part only (KP-ABE). This means that in practice for each new session, i.e. each new initialization of Software (sw) inside IoT deivice, telemetry module requires at least

[3] www.cpubenchmark.net/singleThread.html

**Table 3:** Comparison of KEM criteria and storage costs (in bytes) between code libraries of OpenABE and CRYSTALS-Kyber in security level 3, for n-attributes, n-recipients.

| Criteria | OpenABE | Kyber-1024 |
|---|---|---|
| SYM | AES | - |
| Mathematic | RELIC BN P254 | LWE Lattices |
| SK | + | + |
| AM | + | - |
| ST | +/- | + |
| Ciphertext size | KP-ABE: 100+55*n, CP-ABE: 80+140*n | 1568 * n |
| Secret key size | KP-ABE : 120 *n, CP-ABE: 40 *n | 3168 |
| Public key size | KP-ABE 460, CP-ABE 510 | 1568 |

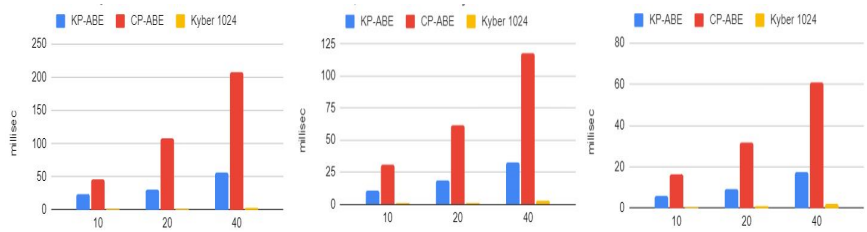


**Figure 2:** Key encapsulation performance comparison in ms. on vCPU with R=1000, 2000, 4000 single thread performance rating for 10, 20, 40 access attributes.

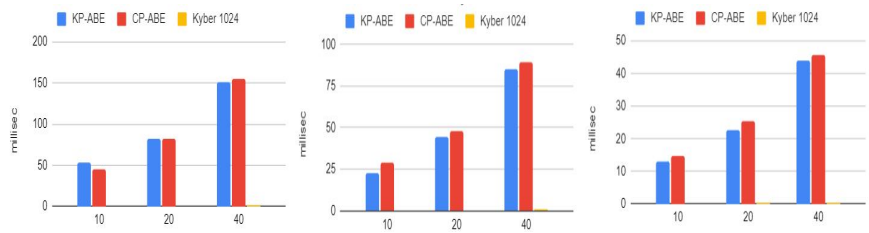


**Figure 3:** Key decapsulation performance comparison in ms. on vCPU with R=1000, 2000, 4000 single thread performance rating for 10, 20, 40 access attributes.

200 milliseconds more to initialize each log L. This is a significant delay for real-time cyber-physical systems.

# 5 Discussion

In the proposed approach for telemetry data protection, asymmetric KEM provides a security advantage since each deployed software only stores public parameters; there are no secret keys in IoT Appliance for an attacker to steal. Compromising a server does not allow the attacker to decrypt historical telemetry data. Compromising an engineering workstation enables the attacker to access only a subset of telemetry data and only stored locally. A drawback of this approach is the performance overhead

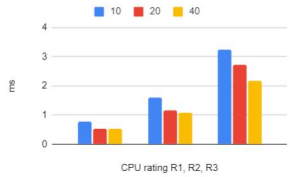


**Figure 4:** Kyber-1024 multi-recipient encapsulation performance in ms. on vCPU with R1=1000, R2=2000, R3=4000 single thread performance rating, for 10, 20, 40 public keys.

.

and requirements for KGA Authority in case of OpenABE use. This approach is also straightforward to modify to allow 1) separating the ability to access (i.e., decrypt) only specific fields from the log Record and 2) introducing 3rd party Cloud-based Hardware Security Module (HSM) service, 3) re-use session-key across multiple log records or days to minimize the overhead on computation and communication costs to save costs for cryptographic migration in cloud.

# 6 Conclusions and future research

This paper introduces a telemetry data confidentiality approach based on end-to-end encryption. It explores various challenges, with a central emphasis on applying KEM cryptosystem in IoT infrastructure. An examination of the existing ABE applications was performed and identified the necessity of further research for public-key cryptography applications for telemetry data confidential sharing. Our study supports the feasibility and efficiency of the proposed approach for cryptographic agility for telemetry data. This research paves the way for future developments in secure cloud-based telemetry data management, offering a scalable solution that can adapt to the evolving landscape of cybersecurity threats and the continuous advancement of end-to-end encryption adoptions for new applications.

# References

[1] B. Hale and C. Komlo. *On End-to-End Encryption*. Publication info: Preprint. MINOR revision. 2022. URL: https://eprint.iacr.org/2022/449 (visited on 04/02/2024).

[2] D. Ott, K. Paterson, and D. Moreau. "Where Is the Research on Cryptographic Transition and Agility?" en. In: *Communications of the ACM* 66.4 (Apr. 2023), pp. 29–32. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3567825. URL: https://dl.acm.org/doi/10.1145/3567825 (visited on 05/09/2023).

[3] C. Bird, T. Menzies, and T. Zimmermann. *The Art and Science of Analyzing Software Data*. English. The Art and Science of Analyzing Software Data. Pages: 648. 2015. ISBN: 978-0-12-411543-9. DOI: 10.1016/C2012-0-07289-4.

[4] B. Ding, J. Kulkarni, and S. Yekhanin. "Collecting Telemetry Data Privately". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/253614bbac999b38b5b60cae531c4969-Abstract.html (visited on 04/12/2024).

[5] A. Silonosov and L. Henesey. "Telemetry data sharing based on Attribute-Based Encryption schemes for cloud-based Drone Management system". In: *Proceedings of the 19th International Conference on Availability, Reliability and Security*. ARES '24. New York, NY, USA: Association for Computing Machinery, July 2024, pp. 1–8. ISBN: 9798400717185. DOI: 10.1145/3664476.3670909. URL: https://dl.acm.org/doi/10.1145/3664476.3670909 (visited on 08/25/2024).

[6] A. Lyons, J. Gamba, A. Shawaga, J. Reardon, J. Tapiador, S. Egelman, and N. Vallina-Rodriguez. "Log: {It's} Big, {It's} Heavy, {It's} Filled with Personal Data! Measuring the Logging of Sensitive Information in the Android Ecosystem". en. In: 2023, pp. 2115–2132. ISBN: 978-1-939133-37-3. URL: https://www.usenix.org/conference/usenixsecurity23/presentation/lyons (visited on 06/04/2024).

[7] B. Waters, D. Balfanz, G. Durfee, and D. Smetters. "Building an Encrypted and Searchable Audit Log". In: (Dec. 2003).

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters. *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data*. Publication info: Published elsewhere. Extended abstract to appear in ACM CCS 2006. This is the full version. 2006. URL: https://eprint.iacr.org/2006/309 (visited on 10/31/2023).

[9] B. Waters. *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*. Publication info: Published elsewhere. Unknown where it was published. 2008. URL: https://eprint.iacr.org/2008/290 (visited on 02/18/2024).

[10] A. Sahai and B. Waters. *Fuzzy Identity Based Encryption*. Publication info: Published elsewhere. Unknown where it was published. 2004. URL: https://eprint.iacr.org/2004/086 (visited on 02/22/2024).

[11] S. Mhatre, A. V. Nimkar, and S. N. Dhage. “Comparative study on attribute-based encryption for health records in cloud storage”. In: *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (May 2017). Conference Name: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) ISBN: 9781509037049 Place: Bangalore Publisher: IEEE, pp. 647–652. DOI: 10.1109/RTEICT.2017.8256677.

[12] R. Imam, K. Kumar, S. M. Raza, R. Sadaf, F. Anwer, N. Fatima, M. Nadeem, M. Abbas, and O. Rahman. “A systematic literature review of attribute based encryption in health services”. In: *Journal of King Saud University - Computer and Information Sciences* 34.9 (Oct. 2022), pp. 6743–6774. ISSN: 1319-1578. DOI: 10.1016/j.jksuci.2022.06.018. URL: https://www.sciencedirect.com/science/article/pii/S1319157822002269 (visited on 02/15/2024).

[13] X. Zhao and Q. Su. “Revocable Attribute-Base Scheme with Enhanced Security and Privacy for Healthcare Data Sharing”. en. In: *2023 IEEE 14th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*. Beijing, China: IEEE, Nov. 2023, pp. 1–8. ISBN: 9798350371024. DOI: 10.1109/PAAP60200.2023.10391571. URL: https://ieeexplore.ieee.org/document/10391571/ (visited on 02/07/2024).

[14] Y. Zhao, P. Fan, H. Cai, Z. Qin, and H. Xiong. “Attribute-based Encryption with Non-Monotonic Access Structures Supporting Fine-Grained Attribute Revocation in M-healthcare”. In: *Int. J. Netw. Secur.* (2017). (Visited on 02/12/2024).

[15] J. Zhao, K. Zhang, J. Gong, and H. Qian. “Lavida: Large-Universe, Verifiable, and Dynamic Fine-Grained Access Control for E-Health Cloud”. In: *IEEE Transactions on Information Forensics and Security* 19 (2024). Conference Name: IEEE Transactions on Information Forensics and Security, pp. 2732–2745. ISSN: 1556-6021. DOI: 10.1109/TIFS.2024.3350925. URL: https://ieeexplore.ieee.org/document/10382677 (visited on 02/12/2024).

[16] A. V, A. P. Nirmala, B. K, A. Christi, and N. A. “A Review on Cloud Cryptography Techniques to Improve Security in E-health Systems”. In: *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*. Mar. 2022, pp. 100–104. DOI: 10.1109/ICCMC53470.2022.9753999. URL: https://ieeexplore.ieee.org/document/9753999 (visited on 02/12/2024).

[17] C. Wang, X. Liu, and W. Li. *Implementing a Personal Health Record Cloud Platform Using Ciphertext-Policy Attribute-Based Encryption*. Journal Abbreviation: Proceedings of the 2012 4th International Conference on Intelligent

Networking and Collaborative Systems, INCoS 2012 Pages: 14 Publication Title: Proceedings of the 2012 4th International Conference on Intelligent Networking and Collaborative Systems, INCoS 2012. Sept. 2012. ISBN: 978-1-4673-2279-9. DOI: 10.1109/iNCoS.2012.65.

[18] S. Porwal and S. Mittal. "Implementation of Ciphertext Policy-Attribute Based Encryption (CP-ABE) for fine grained access control of university data". en. In: *2017 Tenth International Conference on Contemporary Computing (IC3)*. Noida: IEEE, Aug. 2017, pp. 1–7. ISBN: 978-1-5386-3077-8. DOI: 10.1109/IC3.2017.8284289.

[19] S. Hong, H. Pan, Y. Fang, J. Ma, X. Qi, and Y. Hu. "A Logistics Privacy Protection Scheme Based on Ciphertext Policy Attribute-Based Key Encapsulation". In: *2022 International Conference on Blockchain Technology and Information Security (ICBCTIS)*. July 2022, pp. 218–224. DOI: 10.1109/ICBCTIS55569.2022.00057. URL: https://ieeexplore.ieee.org/document/9845073 (visited on 02/12/2024).

[20] Q. Gao, J. Zhang, J. Ma, C. Yang, J. Guo, and Y. Miao. "LIP-PA: A Logistics Information Privacy Protection Scheme with Position and Attribute-Based Access Control on Mobile Devices". en. In: *Wireless Communications and Mobile Computing* 2018 (July 2018). Publisher: Hindawi, e9436120. ISSN: 1530-8669. DOI: 10.1155/2018/9436120. URL: https://www.hindawi.com/journals/wcmc/2018/9436120/ (visited on 02/12/2024).

[21] T. Li, R. Zhang, and Y. Zhang. "PriExpress: Privacy-preserving express delivery with fine-grained attribute-based access control". In: *2016 IEEE Conference on Communications and Network Security (CNS)*. Oct. 2016, pp. 333–341. DOI: 10.1109/CNS.2016.7860501. URL: https://ieeexplore.ieee.org/document/7860501 (visited on 02/15/2024).

[22] V. Shoup. *A Proposal for an ISO Standard for Public Key Encryption*. Publication info: Published elsewhere. Unknown where it was published. 2001. URL: https://eprint.iacr.org/2001/112 (visited on 02/20/2024).

[23] S. D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge: Cambridge University Press, 2012. ISBN: 978-1-107-01392-6. DOI: 10.1017/CBO9781139012843. (Visited on 09/20/2024).

[24] N. P. Smart. "Efficient Key Encapsulation to Multiple Parties". en. In: *Security in Communication Networks*. Ed. by C. Blundo and S. Cimato. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 208–219. ISBN: 978-3-540-30598-9. DOI: 10.1007/978-3-540-30598-9_15.

[25] J. Alwen, D. Hartmann, E. Kiltz, M. Mularczyk, and P. Schwabe. *Post-Quantum Multi-Recipient Public Key Encryption*. Publication info: Preprint. 2022. URL: https://eprint.iacr.org/2022/1046 (visited on 09/20/2024).

[26] A. Abraham and J. Yang. “A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services”. en. In: *Proceedings of the 2023 International Conference on Advances in Computing Research (ACR’23)*. Ed. by K. Daimi and A. Al Sadoon. Cham: Springer Nature Switzerland, 2023, pp. 408–422. ISBN: 978-3-031-33743-7. DOI: 10.1007/978-3-031-33743-7_33.

[27] B. Posey, C. Gropp, B. Wilson, B. McGeachie, S. Padhi, A. Herzog, and A. Apon. “Addressing the Challenges of Executing a Massive Computational Cluster in the Cloud”. In: *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. May 2018, pp. 253–262. DOI: 10.1109/CCGRID.2018.00040. URL: https://ieeexplore.ieee.org/abstract/document/8411029 (visited on 09/22/2024).

[28] X. Ding and J. Shan. “Diagnosing Virtualization Overhead for Multi-threaded Computation on Multicore Platforms”. In: *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*. Nov. 2015, pp. 226–233. DOI: 10.1109/CloudCom.2015.102. URL: https://ieeexplore.ieee.org/abstract/document/7396161 (visited on 09/22/2024).

[29] Y. Ding and K. Klein. “Model-Driven Application-Level Encryption for the Privacy of E-health Data”. In: *2010 International Conference on Availability, Reliability and Security*. Feb. 2010, pp. 341–346. DOI: 10.1109/ARES.2010.91. URL: https://ieeexplore.ieee.org/abstract/document/5438071 (visited on 04/08/2024).

[30] A. Silonosov. *SECMAR Project: Enhancing Maritime Cybersecurity with Drones and Drone Management in the South Baltic Region*. May 2023.

[31] A. Silonosov, O. Baranovskyi, and L. Henesey. “Poster: Towards cryptographic agility in end-to-end encryption systems for computer generated telemetry data.” In: *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*. HOTMOBILE ’24. New York, NY, USA: Association for Computing Machinery, Feb. 2024, p. 144. ISBN: 9798400704970. DOI: 10.1145/3638550.3643624. URL: https://dl.acm.org/doi/10.1145/3638550.3643624 (visited on 04/08/2024).

[32] D. Sikeridis, D. Ott, S. Huntley, S. Sharma, V. K. Dhanasekar, M. Bansal, A. Kumar, A. U. N, D. Beveridge, and S. Veeraswamy. *ELCA: Introducing Enterprise-level Cryptographic Agility for a Post-Quantum Era*. Publication info: Preprint. 2023. URL: https://eprint.iacr.org/2023/1539 (visited on 10/12/2023).

[33] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. “CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM”. In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. London: IEEE, Apr. 2018, pp. 353–367. ISBN: 978-1-5386-4228-3. DOI: 10.1109/EuroSP.2018.00032. URL: https://ieeexplore.ieee.org/document/8406610/ (visited on 07/19/2024).

[34] J. Bethencourt, A. Sahai, and B. Waters. “Ciphertext-Policy Attribute-Based Encryption”. en. In: *2007 IEEE Symposium on Security and Privacy (SP ’07)*. Berkeley, CA: IEEE, May 2007, pp. 321–334. ISBN: 978-0-7695-2848-9. DOI: 10.1109/SP.2007.11.

# Paper IV

# Towards cryptographic agility manifesto in end-to-end encryption systems: a position paper from the perspective of crypto-consumers.

*Alexandr Silonosov, Lawrence Henesey*
*In: 2024 IEEE Conference on Dependable, Autonomic and Secure Computing (DASC), November 2024.*

Abstract

This position paper presents the preliminary results from a research study focusing on cryptographic agility for consumers of open-source cryptographic libraries. We provide a concise overview of frontiers and recent advancements in cryptographic agility research frontier, examining the utilized approaches and emphasizing the perception of application layer encryption in end-to-end encryption systems. The paper delves into the state of practice of cryptographic libraries and programming interfaces, outlining recognized challenges and knowledge gaps that warrant exploration through new scientific research. Furthermore, we outline the values of cryptographic agility in a manifesto and propose a survey structure to validate our assumptions.

## 1 Introduction

Nowadays, as cryptography has become very accessible, [1] the variety of its applications continues to grow [2]. The state of practice in the industry shows that cryptographic API has become heterogeneous. For example, stand-alone crypto libraries may have dependencies from other crypto-libraries, while cloud-based crypto-API constantly evolves into new interfaces [3].

Regulation and standardization authorities frequently prompt the need for novel cryptographic approaches. This demand stems from evolving cyber-threat landscapes, ongoing scientific or community research, and adherence to regulatory compliance standards. Examples include emerging cryptographic fields like Post-quantum cryptography (PQC), lightweight cryptography, and Attribute-Based Encryption (ABE). In

response, cryptographers and conventional software engineers have devised various iterations of encryption elements such as algorithms, techniques, schemes, protocols, cryptosystems, or entirely new types of cryptography. Some have become defacto standards and are widely deployed, while others are experimental, hardly used, and offer poor performance. As a result, applied cryptography has increased complexity [4], [5], [6].

Cryptography for data under computation in a new scientific frontier [7] and leading industry figures actively implement and advocate this approach with End-to-End Encryption (E2EE) [8], customer-owned encryption, zero-knowledge encryption [9], application layer encryption [10] and this leads to the management of an enormous number of encryption keys and crypto-artifacts on client and server side [11] and therefore it is feasible to highlight the cases with intensive use of cryptography.

Furthermore cryptography consumers [7], must adapt to changes and ensure data protection in a heterogeneous distributed environment complying with data protection regulations such as FIPS, GDPR, HIPPA, PCI-DSS. In addition, cryptographic consumers face two additional sources of data-protection challenges: the research community and the cyber-threat landscape. Motivated by bug-bounty and darknet market trends [12], the research community identifies vulnerabilities and weaknesses in the specific implementation of encryption schemes that may be theoretically sound but lack the strength necessary for the required level of protection. Sridhar et al. [13], demonstrated the effects of bug-bounty adoption and its economic structure depending on the nature of the data: financial and healthcare being industries that are the most targeted. Fischer et al. [14] proposed a cryptography ecosystem mapping for crypto-consumers and identified five major challenges that hinder the adoption of cryptography, where usability and human factors are the weakest elements. On the other side, work [14] omit to map the role of cyber-criminal factors in the cyber-threat landscape for data confidentiality protection via growing number of Common Vulnerabilities and Exposures (CVE) records related to crypto-libraries. Examining CVE databases reveals numerous recent vulnerabilities that expose weaknesses in cryptographic implementations. Emerging cryptographic implementation challenges are actively exploited, including issues such as side-channels, attacks arising from shared resources in a cloud environment and the improper composition of crypto-artifacts.

As presented, cryptography consumers must effectively respond accordingly to three types of challenges: regulatory, community-driven, and those arising from cyber-threats. The table 1 represents examples for each challenge mapped to phases of the software development lifecycle (SDL) model, which are Requirements (R), Design (Ds), Implement (I), Test (T), Deploy (Dp) and Review (R) [15]. Regulatory compliance, such as FIPS, can be represented as a set of requirements (R), software architecture design solutions (Ds), and implementation (I) details [16], [17], [18]. Community-driven bug-bounty discover vulnerabilities that need to be fixed in implementation (I), verified during testing (T) and deployment (Dp), and later reviewed (R) during maintenance [19], [20]. Cyber-threats such as common vulnerabilities of

crypto-API or PQC may lead to cryptographic migration or redesign or refactoring on any phase (*) of SDL [21], [22].

| Challenge | Phase | Example |
|---|---|---|
| Regulatory | R, Ds, I | NIST, FIPS, ISO, HIPPA, PCI-DSS, NIS2 |
| Community | I,T,Dp,R | Bug-bounty |
| Cyber-threats | I,T,Dp,R | CVE |

**Table 1:** Challenges for crypto-users mapped to phases of software development lifecycle model.

Apart from the challenges in table 1 cryptographic consumers might need to update the existing software system or introduce new features that relate to data confidentiality, for example:

- Implement data protection for new data.
- Plan coordinated transition to a new cryptographic algorithm.
- Plan data migration for new business requirements.

To accomplish this development multiple tasks need to be solved, such as constructing new abstractions in the code base, selecting a new cryptographic library or tools, generating new encryption keys with distinct properties, designing re-encryption procedures, and establishing data security criteria for different tenants and data-regions in alignment with data-provenance policy. In many cases, achieving a large-scale cryptographic migration may involve complexities on multiple system levels such as data level, component level, and deployment level [23]. The work addresses the following research questions:

- (1) How do the challenges that have been identified earlier in Table 1, influence the nature of cryptographic code refactoring in cryptography-intensive system?
- (2) How do these challenges drive the evolution of Public Key Encryption (PKE) approaches and standards?
- (3) What are the structural and security differences between traditional PKE and PQC, and how do these differences influence software architecture, performance and resilience to cryptographic attacks?”
- (4) What are the key challenges and considerations in adopting cryptographic migrations from traditional PKE to the next generation of PKE and it’s approaches in E2EE systems?

A new research frontier has emerged in science that helps evaluate the efficiency of solving mentioned tasks at a scale referred to as cryptographic-agility (crypto-agility), and a literature search shows that there is a knowledge gap in this area [23].

Our work contributes to understanding the interplay between the architecture of the cryptographic library and Application Layer Encryption (ALE) principle with regard to engineers' perception of end-to-end encryption systems.

The paper is structured as follows: Section 1 introduces the challenges related to the problem. Section 2 encompasses a review of related work; in Section 3, we offer an overview categorizing the state of practice related to crypto-agility and provide our definition in the context of E2EE systems. Section 4 outlines the proposal for employing crypto-agility indicators and tools and discusses the challenges, while Section 5 concludes the paper with remarks and provides pointers for future research.

## 2 Related work

This section represents a summary of the knowledge and related work in the area of crypto-agility published in 2021-2023 by searching in Scopus by using this search phrase:

> "( crypt* W/3 agil* AND ( model OR framework OR review OR survey ) )"

The search results were then filtered to include only the following types of publication: "survey, models and framework." First, we outline the pre-conditions, challenges, and main drivers that motivate a new term of crypto-agility, and then in section III we synthesize our definition in context of E2EE systems. Crypto-agility was initially associated with Post-Quantum Cryptography (PQC) technology, where a new set of cryptographic algorithms should replace widely used algorithms based on mathematical problems that are easily solved by quantum computers [24]. The assumption was that PQC would not be vulnerable to quantum computers and all other threats that exist for current traditional cryptography. The state of the practice outlined below shows that apart from PQC, various threats and challenges exist that require continuous adaptation and transition of cryptographic tools, which is currently referred to as crypto-agility.

Ma et al. [25] proposed a quantitative approach to measure crypto-agility as a Crypto Agility Risk Assessment Framework (CARAF) that addresses challenges from cyber-threats, vulnerabilities in cryptographic tools, regulatory compliance, and supply-chain dependencies. The 5 phase risk assessment process in CARAF also includes crypto-inventory and timeline mappings. The inventory contains crypto-assets, normal assets, and ownership information, where the timeline maps the life-span of the assets time required for risk mitigation for such assets in case of particular treatment occurs within a 3rd time metric. However, CARAF is not adopted for zero-day vulnerabilities with an unpredictable timeline nature since they may occur at any time unexpectedly, and according to the framework, it produces the highest possible risk values for all assets.

Alnahavi et al. [26] provides a comprehensive review of the crypto-agility body of knowledge by surveying the literature on 64 topics identified on crypto-agility, divided by twelve categories such as crypto-agility, definitions, need, scope, prerequisites, etc. According to the survey, the topic 'Migration to Post Quantum Cryptography' is the most popular topic, as discussed in 10 publications. Other popular topics with crypto-agility mentioned include various perspectives, including managing all crypto components, Integrating hybrid keys into protocols, Algorithm-Agile hardware acceleration, and Documentation Issues. Mentioned in the paper is: "the need to develop common understanding in crypto-agility including definitions, terms, and language in general in order to support the discussion on crypto-agility".[26]

In Hohm et al. [27] proposed a crypto-agility maturity model (CAMM), designed a list of requirements from multiple domains, and proposed five levels of readiness for crypto-agility. The framework resembles the well-known Capability Maturity Model (CMM) in Software Engineering. Still, unlike CMM, which focuses only on the number of software development practices on each level, the proposed CAMM defines 25 requirements that focus on five domains such as, system features (having a crypto-inventory), software development life cycle (software system architecture and maintenance) and cyber-security governance (for example, to use the strongest encryption algorithm, or ability to exclude specific algorithm ).

Badertscher et al. [28] proposed formal language for cryptographic agility modeling in the setting of universally composable security formalization model [29] to real world attacks such as protocol substitution and downgrade.

Ott et al. [23] presented a viewpoint on crypto-agility challenges from the perspective of development operations of software-intensive systems in distributed computational platforms, referring to the lack of crypto-agility mechanisms and schemes to facilitate transition. This foresaw the need for next generation of security protocols that include information about data owners, data custodians, and cryptography users to address increased regulatory compliance challenges.

The following section includes the state of the practice related to crypto-agility divided by compliance, community, and specific applications known as end-to-end encryption.

# 3 Present state of practice

We discuss three state-of-practice areas that provide technical foundations for our study, which are compliance, cyber security community, and cyber threats. We begin by briefly summarizing regulatory compliance related to cryptography, followed by the influence of the cyber-security community. Finally, we provide evidence of a specific use case for cryptography, namely end-to-end encryption.

Since 2016 NIST USA initiated standardization processes for new types of cryp-

tography such as : Post-Quantum Cryptography (PQC)[1] , Lightweight Cryptography [2], Multi-Party Threshold Cryptography (since 2011) [3], the work leads to the public release of FIPS publications in 2023. NIST SP 800-57 publication ”Recommendation for Key Management” (revision 5) in 2005 emphasizes the importance of cryptographic keys lifecycle, referred to as cryptoperiod. NIST assumes that the security strength provided by an algorithm or key may be reduced or lost completely, for example, the algorithm or key length used may no longer offer adequate security because of improvements in computational capability or cryptanalysis. And therefore algorithm or key with low-security strength will be prohibited to use.

PCI-DSS standard uses references to NIST standards and follows the term in strong cryptography, defined to use a minimum of 128-bits of effective key strength and functional requirements such a ”up-to-date inventory of all cryptographic cipher suites and protocols in use, including purpose and where used”.

NIS2 outlines a baseline with a list of seven mandatory vital elements that all companies must address or implement as part of their cybersecurity measures. [4]. These elements include the utilization of cryptography, encryption, validation of supply chain security, and vulnerability disclosure.

## 3.1 Cyber-security community and cyber-threats

Community-developed list of software and hardware weaknesses (CWE) contains fifteen types of weaknesses related to use of cryptography (out of 600 in total), for example CWE-327 Use of a Broken or Risky Cryptographic Algorithm [5], CWE-780: Use of RSA Algorithm without OAEP. OWASP also published Top Ten Cryptographic Failures in 2021 [6]. For CWE-326, Inadequate Encryption Strength, currently exist 326 known vulnerabilities in various software packages [7].

Research also demonstrates the rising role of bug-bounty adoption in industry [13], [30], as well as government sectors [31]. Bug-bounty or vulnerability reward program (VRP) is a software bug detection contest organized by an organization for the entire cybersecurity community, where an individual or team makes attempts to investigate the organization’s assets for cyber vulnerabilities. Research shows that software development companies and Open-source projects account for most programs listed on bug-bounty outsourcing platforms such as HackerOne and BugCrowd [31] and Only 10% of organizations provided source code for vulnerability search [32].

[1] https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization

[2] https://csrc.nist.gov/projects/lightweight-cryptography

[3] https://csrc.nist.gov/Projects/threshold-cryptography

[4] digital-strategy.ec.europa.eu/en/policies/nis2-directive

[5] cwe.mitre.org/data/definitions/327.html

[6] cwe.mitre.org/data/definitions/1346.html

[7] https://www.cvedetails.com/vulnerability-list/cwe-326/vulnerabilities.html

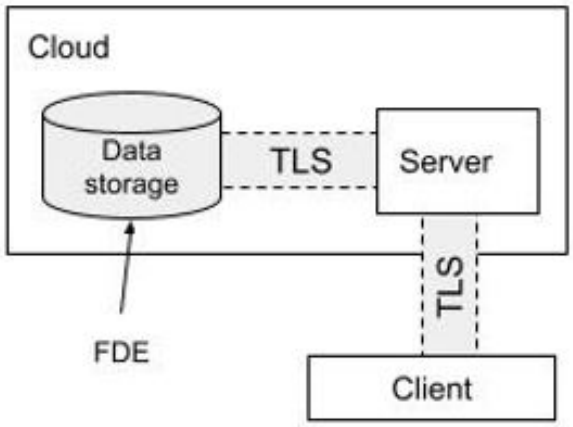


**Figure 1:** Encryption in-transit (TLS) and at-rest (FDE) in Client-Server systems

To conclude, from one side, NIST initiatives represented good examples of how regulation authority contributes to the main drivers of resilience, adaptation and modularity of cryptographic technology. Conversely, regulations like NIS2 require companies to significantly increase self-awareness about cryptographic usage, which may primarily result in the increased use of cryptography and thus transition of data model in existing systems. [18].

## 3.2 End-to-end encryption approach

End-to-end encryption (E2EE) approach assumes client-server architure with an integrated encryption scheme where the "ends" of the communication (a "sender" and one or more "recipients") can send messages to each other via an abstract channel and where the server does not have access to the cryptographic keys necessary to read or invisibly alter the message [33] .

Traditional Client-Server architecture in cloud-based systems employs only encryption In-Transit technology provided by Transport-Layer-Security protocol. Figure 2 shows how Encryption At-Rest is provided in next-generation cloud-based architecture by implementing dedicated data storage encrypted with Full-Disk Encryption (FDE) technology. In both pictures, the encryption key management is provided by the TLS protocol or Data storage, and Client is not involved in this process.

Figure 3 represents the End-to-End Encryption approach implemented for a Cloud-based system. For E2EE architecture, clients employ Application Level Encryption (ALE) [34] to encrypt business data inside the client. As shown, the client transfers the encrypted data in the gray ALE area to the server, and later, ALE-encrypted data arrives at cloud storage in always-encrypted form. In this case, the cloud also uses encryption in-transit and encryption at-Rest principles, but encryption key management for ALE

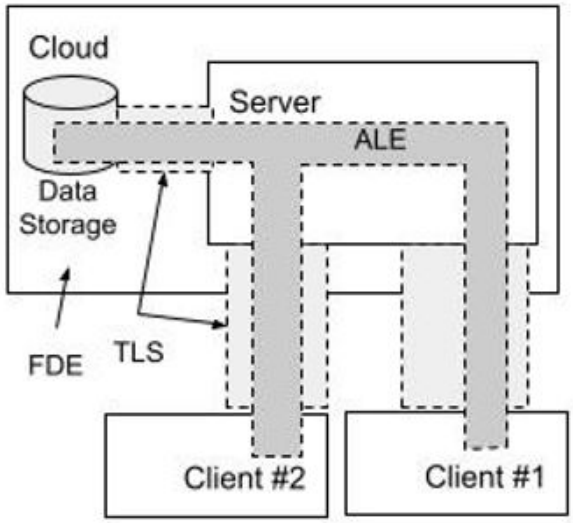


**Figure 2:** Crytography in-transit, at-rest and under-computation (ALE) in E2EE based Client-Server system.

belongs only to the Client side. Thus we conclude the client side has in

According to the E2EE protocol design, only the client owns the encryption and decryption keys. Thus, requests and approvals to access data are controlled only by the client, who manages the encryption key lifecycle and storage. Encryption key ownership by the client is an important requirement that distinguishes E2EE from other approaches. Before E2EE adoption, data protection architectures in cloud systems assume that the server always has access to decryption keys to process the data in plain, decrypted form.

| Type | In-transit | At-rest | Under computation |
|---|---|---|---|
| CS-1 | TLS | - | - |
| CS-2 | TLS | FDE, TDE | - |
| CS-E2EE | TLS | FDE, TDE | ALE |

**Table 2:** Comparison of cryptography use in cloud based client-server architectures.

Research of E2EE applications shows the increased complexity of internal key management, where each ”end” holds a particular set of own key material, including all recipients [35], [36], [37]. Currently, Signal protocol is literally a key factory, producing new encryption keys with each chat message [38]. This creates an overhead on the application layer to organize additional key management.

State of practice review shows that exist code libraries that implement ALE

approach for example, LogSentinel [8], Cloaked Search [9] and AWS Crypto Tools [10] and it heavily relies on various PKE approaches.

E2EE systems are cryptography-intensive since they contain a significant amount of cryptographic code in the ALE layer that first heavily relies on the PKE primitives in the underlying cryptographic library and second, solidified with the data model of the system. Depending on the mentioned challenge the software variability issues [39] also relate to ALE as software abstraction layer. To conclude, we synthesize the definition of crypto-agility as follows: The ability of cryptography-intensive systems to adapt seamlessly to new algorithms, protocols, or frameworks as security requirements. Ultimately, crypto-agility enables E2EE systems to maintain a certain level of cryptographic heterogeneity with determined level of security impact or performance ovehead, ensuring that they remain resilient against future cryptographic advancements and threat model. The PKE-to-PQC migration serves as a critical example of how agility is essential for the continued robustness of E2EE infrastructures.

# 4 Cryptographic-agility Manifesto

Ermoshina et al. [40] identified how Signal protocols influenced E2EE technology in a "quasi standardisation" way highlighting that cryptographic technology evolves by practitioners, business needs, and even socio-political effects. The study confirm the effects of open-source libraries and community-driven efforts to evolution of cryptographic library architecture and data model. Therefore, it raises a question of the feasibility of identifying new values and derived principles if we separate the concerns of cryptographic algorithms and parameters from the data model and code base.

Table 3 contains the list of the thirteen most popular, community-driven, open-source cryptographic libraries. This list represents maturity metrics such as; Year of the first version release, number of code contributors (Contrs.), number of found vulnerabilities (CVE), number of code reusages across community (Forks) in thousands (according to the data provided by GitHub.com and MITRE services).

According to a comparative analysis, OpenSSL library has the most significant impact on the community. The impact is measured by the total number of individuals (Contrs. column) who have contributed to the code base at least once. The Forks column demonstrates the level of interest by crypto-users since it shows the number of modified editions of the library during development. CVE [11]column represents the level of interest by the cyber-security community i.e. ethical hackers and criminals, which in turn works as a development driver for the library since after each fixed

[8]https://logsentinel.com/why-is-searchable-encryption-so-important

[9]https://ironcorelabs.com/products/cloaked-search/

[10]https://docs.aws.amazon.com/aws-crypto-tools/

[11]https://cve.mitre.org

CVE the library obtains more secure state. The data also shows that 4 libraries do not maintain a codebase repository which makes it difficult to track changes in the code.

| Name | Year | Contrs. | CVE | Forks |
|---|---|---|---|---|
| Crypto++ | 1995 | 88 | 13 | 1.8 |
| cryptlib | 1995 | - | - | - |
| Libgcrypt | 1997 | - | 48 | - |
| OpenSSL | 1998 | 864 | 443 | 10 |
| bouncycastle | 2000 | 90 | 13 | 1 |
| Nettle | 2001 | - | 9 | - |
| mbedTLS | 2006 | 315 | 31 | 2.4 |
| libsodium | 2012 | 128 | 3 | 1.7 |
| Nacl | 2012 | - | - | - |
| Botan | 2013 | 125 | 26 | 0.5 |
| LibreSSL | 2014 | 75 | 13 | 0.3 |
| liboqs | 2011 | 94 | - | - |
| RustCrypto | 2019 | 35 | 2 | 0.1 |
| HACL* | 2019 | 56 | - | 0.1 |
| PQClean | 2016 | 25 | - | - |
| pqm4 | 2018 | 28 | - | - |

**Table 3:** cryptographic libraries supported by the non-commercial community.

Table 4 represents the comparative analysis of open-source cryptographic libraries created by the biggest IT vendors. The data shows that starting in 2015, Big Tech decided to publish cryptographic code as part of an ongoing bug bounty program to attract the collective power of the community and search for vulnerabilities. Obviously, the volume of CVEs in 4 is less than in Table 3, which allows us to make an assumption that some vulnerabilities were fixed during bug-bounty reports before getting published in CVE database.

| Name | Year | Vendor | Contrs. | CVE | Forks |
|---|---|---|---|---|---|
| Libgcrypt | 1997 | GnuPG | - | 17 | 0.1 |
| JCA/JCE | 1997 | Oracle | - | - | - |
| NSS | 1997 | Mozilla,AOL,.. | - | - | - |
| WolfCrypt | 2006 | wolfSSL Inc | 114 | 7 | 0.7 |
| SymCrypt | 2006 | Microsoft | - | 3 | 0.1 |
| Signal | 2014 | Signal | 39 | 14 | |
| boringssl | 2015 | Google | 151 | 6 | 0.7 |
| conscrypt | 2017 | Google | 77 | 6 | 0.3 |
| Tink | 2017 | Google | 18 | 0 | 0 |
| SEAL | 2018 | Microsoft | 40 | 0 | 0.7 |
| circl | 2019 | Cloudflare | 24 | 1 | 0.1 |
| Corretto | 2019 | Amazon | 26 | 0 | 0.2 |
| aws-lc | 2020 | Amazon | 191 | 0 | 0.1 |
| swift-crypto | 2020 | Apple | 34 | 0 | 0.1 |

**Table 4:** corporate open-source cryptography libraries

As demonstrated by NIST SP 800-53, cryptographic artifacts have their own lifecycle. Such a dynamic perspective resembles Agility in software engineering, where it is iterative and incremental. However, for cryptographic consumers, it is cyclic and adaptational. First, it means that cryptographic strength should be increased over relatively larger periods of the information system's lifetime. Second, the cryptographic codebase and, thus, the associated data model should be adapted. On

the other side, research [21], [22] identified substantially increased code complexity in cryptographic software compared to non-cryptographic software. The number of contributors in public repositories shows that increasing community efforts may be a good strategy for addressing this issue.

According to our assumption [41], it is possible to identify and prioritize core values that contribute to cyber-resilience as a long-term interplay between an organization's technical capacity, human resources, technological trends, and cyber-threat landscape, ensuring success as a crypto-consumer. Using the mentioned data during state-of-practice analysis makes it possible to draw specific patterns. The following Table 5 represents how Crypto-agility can contribute to a cryptography-intensive system through a sample manifestation.

| **Cryptographic-agility Manifesto** |
|---|
| 1. Publish cryptographic code as an open-source library.<br>2. Announce security assumptions and decisions as technical notes.<br>3. Accumulate threat intelligence via bug-tracking artifacts in a public code repository. |

**Table 5:** Sample manifesto

As shown in state of practice, open-source cryptogrpahic libraries play a significant role on software industry as influencers for data protection practitioners. It is common in cryptography as a science to announce plausible assumptions and argue security properties while designing cryptosystems, but assumptions itself stay on theoretical level. Bug-bounty economy shows that code repository with integrated bug-tracking issues log enriches threat intelligence for systems with intensive use of cryptography. This can motivate study for the following research question: How to identify, validate and prioritize values in the cryptography-intensive system? Maturity models are well-known in the context of software process that had been introduced in theory and it target to evaluate the practices in areas such as Process, Technology and Culture. [42] and we aim to introduce indicators for maturity and readiness as cryptographic consumer in context of E2EE systems.

The following table 6 shows examples of crypto-agility values mapped to well-known trade-offs in cyber-security.

| Cryptographic-agility Values and trade-offs |
|---|
| 1. Open-source cryptographic library or costs for security audit.<br>2. Ongoing bug bounty in a public repository or rigorous security proofs.<br>3. Announce security assumptions or make privacy promise.<br>4. Plan adaptation strategy or respond to data breach in an emergency. |

**Table 6:** value mappings for the average cryptographic consumer

# 5 Conclusions and discussion

This paper introduces the crypto-agility manifesto and explores various challenges, emphasizing investigating the E2EE-based systems. Examining the cryptographic codebases, and its use in E2EE becomes imperative to determine its suitability for research on crypto-agility. Furthermore, the research explored the role of compliance, internet community and cyber-threats to cryptographic agility. As presented in a state of practice, crypto-agility can be described as cyclical and adaptive measures to support data confidentiality on the required level. Furthermore, a novel crypto-agility framework as a manifest was proposed to address the discussed challenges. Future research is to validate the manifesto with real-world data. To accomplish this, we aim to introduce indicators to measure cryptographic consumer maturity by proposing auditing tool for crypto-consumers in dimensions such as Management, Assets, Human capital, Tools and Functionality. From another side we aim to farther develop tools and indicators to evaluate the effectiveness of cryptographic agility of a system.

The primary emphasis of our research has been focused on crypto-agility in end-to-end encryption systems, and future research may explore various traditional methods to study the end-to-end encryption, create benchmarks and criteria for further studies, and gain additional knowledge about crypto-agility tasks in information systems based on end-to-end encryption approach. Closer cooperation between relevant industrial and academic experts could lead to valuable insights and further improvements in the developed manifesto.

# References


[1] M. Ryan. “Evolution of Applied Cryptography”. en. In: *Ransomware Revolution: The Rise of a Prodigious Cyber Threat*. Ed. by M. Ryan. Advances in Information Security. Cham: Springer International Publishing, 2021, pp. 33–41. ISBN: 978-3-030-66583-8. DOI: 10.1007/978-3-030-66583-8_3. URL: https://doi.org/10.1007/978-3-030-66583-8_3 (visited on 10/12/2023).

[2] L. H. Iwaya, M. A. Babar, and A. Rashid. “Privacy Engineering in the Wild: Understanding the Practitioners’ Mindset, Organizational Aspects, and Current Practices”. In: *IEEE Transactions on Software Engineering* 49.9 (Sept. 2023). Conference Name: IEEE Transactions on Software Engineering, pp. 4324–4348. ISSN: 1939-3520. DOI: 10.1109/TSE.2023.3290237. URL: https://ieeexplore.ieee.org/abstract/document/10167784 (visited on 10/12/2023).

[3] X. Huang and R. Chen. “A Survey of Key Management Service in Cloud”. In: *2018 IEEE 9th International Conference on Software Engineering and*

*Service Science (ICSESS)*. ISSN: 2327-0594. Nov. 2018, pp. 916–919. DOI: 10.1109/ICSESS.2018.8663805. URL: https://ieeexplore.ieee.org/document/8663805 (visited on 04/11/2024).

[4] J. Haney, M. Theofanos, Y. Acar, and S. S. Prettyman. *Organizational views of NIST cryptographic standards and testing and validation programs*. en. Tech. rep. NIST IR 8241. Gaithersburg, MD: National Institute of Standards and Technology, Dec. 2018, NIST IR 8241. DOI: 10.6028/NIST.IR.8241. URL: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8241.pdf (visited on 10/12/2023).

[5] L. Nurgalieva, A. Frik, and G. Doherty. "A Narrative Review of Factors Affecting the Implementation of Privacy and Security Practices in Software Development". In: *ACM Computing Surveys* 55.14s (July 2023), 320:1–320:27. ISSN: 0360-0300. DOI: 10.1145/3589951. URL: https://dl.acm.org/doi/10.1145/3589951 (visited on 10/12/2023).

[6] P. L. Gorski, L. L. Iacono, D. Wermke, C. Stransky, S. Möller, Y. Acar, and S. Fahl. "Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic {API} Misuse". en. In: 2018, pp. 265–281. ISBN: 978-1-939133-10-6. URL: https://www.usenix.org/conference/soups2018/presentation/gorski (visited on 10/12/2023).

[7] K. G. Paterson. "The Cyber Security Body of Knowledge v1.1.0, 2021". In: Section: Applied Cryptography. University of Bristol, 2021. URL: https://www.cybok.org/.

[8] B. Hale and C. Komlo. *On End-to-End Encryption*. Publication info: Preprint. MINOR revision. 2022. URL: https://eprint.iacr.org/2022/449 (visited on 04/02/2024).

[9] C. Manthiramoorthy, K. M. S. Khan, and N. A. A. "Comparing Several Encrypted Cloud Storage Platforms". en. In: *International Journal of Mathematics, Statistics, and Computer Science* 2 (2024), pp. 44–62. ISSN: 2704-1069. DOI: 10.59543/ijmscs.v2i.7971. URL: https://ijmscs.org/index.php/ijmscs/article/view/7971 (visited on 04/02/2024).

[10] B. Wall and P. Walsh. "Cryptographically Enforced Orthogonal Access Control at Scale". In: *Proceedings of the 6th International Workshop on Security in Cloud Computing*. SCC '18. New York, NY, USA: Association for Computing Machinery, May 2018, pp. 57–65. ISBN: 978-1-4503-5759-3. DOI: 10.1145/3201595.3201602. URL: https://dl.acm.org/doi/10.1145/3201595.3201602 (visited on 08/29/2023).

[11] E. Barker. *Recommendation for Key Management: Part 1 – General*. en. Tech. rep. NIST Special Publication (SP) 800-57 Part 1 Rev. 5. National Institute of Standards and Technology, May 2020. DOI: 10.6028/NIST.SP.800-57pt1r5. URL: https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final (visited on 10/12/2023).

[12] S. Atefi, A. Sivagnanam, A. Ayman, J. Grossklags, and A. Laszka. "The Benefits of Vulnerability Discovery and Bug Bounty Programs: Case Studies of Chromium and Firefox". In: *Proceedings of the ACM Web Conference 2023*. WWW '23. New York, NY, USA: Association for Computing Machinery, Apr. 2023, pp. 2209–2219. ISBN: 978-1-4503-9416-1. DOI: 10.1145/3543507.3583352. URL: https://dl.acm.org/doi/10.1145/3543507.3583352 (visited on 10/12/2023).

[13] K. Sridhar and M. Ng. "Hacking for good: Leveraging HackerOne data to develop an economic model of Bug Bounties". In: *Journal of Cybersecurity* 7 (Feb. 2021). DOI: 10.1093/cybsec/tyab007.

[14] K. Fischer, I. Trummová, P. Gajland, Y. Acar, S. Fahl, and M. A. Sasse. *On The Challenges of Bringing Cryptography from Papers to Products: Results from an Interview Study with Experts (Extended Version)*. eng. Publisher: Zenodo. Oct. 2023. DOI: 10.5281/zenodo.8404611. URL: https://zenodo.org/records/8404611 (visited on 06/08/2024).

[15] N. B. Ruparelia. "Software development lifecycle models". en. In: *ACM SIGSOFT Software Engineering Notes* 35.3 (May 2010), pp. 8–13. ISSN: 0163-5948. DOI: 10.1145/1764810.1764814. URL: https://dl.acm.org/doi/10.1145/1764810.1764814 (visited on 04/02/2024).

[16] I. Muslukhov, Y. Boshmaf, and K. Beznosov. "Source attribution of cryptographic API misuse in android applications". English. In: 2018, pp. 133–146. ISBN: 978-1-4503-5576-6. DOI: 10.1145/3196494.3196538.

[17] L. Piccolboni, G. Guglielmo, L. Carloni, and S. Sethumadhavan. "CRYLOGGER: Detecting crypto misuses dynamically". English. In: vol. 2021-May. ISSN: 1081-6011. 2021, pp. 1972–1989. ISBN: 978-1-72818-934-5. DOI: 10.1109/SP40001.2021.00010.

[18] M. Green and M. Smith. "Developers are Not the Enemy!: The Need for Usable Security APIs". In: *IEEE Security & Privacy* 14.5 (Sept. 2016). Conference Name: IEEE Security & Privacy, pp. 40–46. ISSN: 1558-4046. DOI: 10.1109/MSP.2016.111. URL: https://ieeexplore.ieee.org/document/7676144 (visited on 01/20/2024).

[19] M. Bortolozzo, M. Centenaro, R. Focardi, and G. Steel. "Attacking and fixing PKCS#11 security tokens". English. In: ISSN: 1543-7221. 2010, pp. 260–269. ISBN: 978-1-4503-0244-9. DOI: 10.1145/1866307.1866337.

[20] S. Rahaman, Y. Xiao, S. Afrose, F. Shaon, K. Tian, M. Frantz, M. Kantarcioglu, and D. Yao. “Cryptoguard: High precision detection of cryptographic vulnerabilities in massive-sized Java projects”. English. In: ISSN: 1543-7221. 2019, pp. 2455–2472. ISBN: 978-1-4503-6747-9. DOI: 10.1145/3319535.3345659.

[21] J. Blessing, M. A. Specter, and D. J. Weitzner. *You Really Shouldn't Roll Your Own Crypto: An Empirical Study of Vulnerabilities in Cryptographic Libraries*. arXiv:2107.04940 [cs]. July 2021. DOI: 10.48550/arXiv.2107.04940. URL: http://arxiv.org/abs/2107.04940 (visited on 01/20/2024).

[22] D. Lazar, H. Chen, X. Wang, and N. Zeldovich. “Why does cryptographic software fail? A case study and open problems”. en. In: ().

[23] D. Ott, K. Paterson, and D. Moreau. “Where Is the Research on Cryptographic Transition and Agility?” en. In: *Communications of the ACM* 66.4 (Apr. 2023), pp. 29–32. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3567825. URL: https://dl.acm.org/doi/10.1145/3567825 (visited on 05/09/2023).

[24] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. *Report on Post-Quantum Cryptography*. en. Tech. rep. NIST IR 8105. National Institute of Standards and Technology, Apr. 2016, NIST IR 8105. DOI: 10.6028/NIST.IR.8105. URL: https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf (visited on 04/04/2024).

[25] C. Ma, L. Colon, J. Dera, B. Rashidi, and V. Garg. “CARAF: Crypto Agility Risk Assessment Framework”. English. In: *Journal of Cybersecurity* 7.1 (2021). ISSN: 2057-2085. DOI: 10.1093/cybsec/tyab013.

[26] N. Alnahawi, N. Schmitt, A. Wiesmaier, A. Heinemann, and T. Grasmeyer. *On the State of Crypto-Agility*. Publication info: Published elsewhere. 18. Deutscher IT-Sicherheitskongress. 2023. URL: https://eprint.iacr.org/2023/487 (visited on 08/25/2023).

[27] J. Hohm, A. Heinemann, and A. Wiesmaier. “Towards a Maturity Model for Crypto-Agility Assessment”. English. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13877 LNCS (2023). ISBN: 9783031301216, pp. 104–119. ISSN: 0302-9743. DOI: 10.1007/978-3-031-30122-3_7.

[28] C. Badertscher, M. Ciampi, and A. Kiayias. “Agile Cryptography: A Universally Composable Approach”. en. In: *Theory of Cryptography*. Ed. by G. Rothblum and H. Wee. Cham: Springer Nature Switzerland, 2023, pp. 480–509. ISBN: 978-3-031-48624-1. DOI: 10.1007/978-3-031-48624-1_18.

[29] R. Canetti. *Universally Composable Security: A New Paradigm for Cryptographic Protocols*. Publication info: Preprint. MINOR revision.Extended Abstract appeared in proceedings of the 42nd Symposium on Foundations of Computer Science (FOCS), 2001. 2000. URL: https://eprint.iacr.org/2000/067 (visited on 04/08/2024).

[30] T. Walshe and A. Simpson. "An Empirical Study of Bug Bounty Programs". English. In: 2020, pp. 35–44. ISBN: 978-1-72816-280-5. DOI: 10.1109/IBF50092.2020.9034828.

[31] A. Chatfield and C. Reddick. "Cybersecurity innovation in government: A case study of U.S. pentagon's vulnerability reward program". English. In: vol. Part F128275. 2017, pp. 64–73. ISBN: 978-1-4503-5317-5. DOI: 10.1145/3085228.3085233.

[32] A. Laszka, M. Zhao, A. Malbari, and J. Grossklags. "The Rules of Engagement for Bug Bounty Programs". In: *Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26 – March 2, 2018, Revised Selected Papers*. Berlin, Heidelberg: Springer-Verlag, Feb. 2018, pp. 138–159. ISBN: 978-3-662-58386-9. DOI: 10.1007/978-3-662-58387-6_8. URL: https://doi.org/10.1007/978-3-662-58387-6_8 (visited on 04/04/2024).

[33] S. Scheffler and J. Mayer. "SoK: Content Moderation for End-to-End Encryption". en. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (Apr. 2023), pp. 403–429. ISSN: 2299-0984. DOI: 10.56553/popets-2023-0060. URL: https://petsymposium.org/popets/2023/popets-2023-0060.php (visited on 10/30/2023).

[34] Y. Ding and K. Klein. "Model-Driven Application-Level Encryption for the Privacy of E-health Data". In: *2010 International Conference on Availability, Reliability and Security*. Feb. 2010, pp. 341–346. DOI: 10.1109/ARES.2010.91. URL: https://ieeexplore.ieee.org/abstract/document/5438071 (visited on 04/08/2024).

[35] K. G. Paterson, M. Scarlata, and K. T. Truong. "Three Lessons From Threema: Analysis of a Secure Messenger". en. In: 2023, pp. 1289–1306. ISBN: 978-1-939133-37-3. (Visited on 03/14/2025).

[36] M. Backendal, M. Haller, and K. G. Paterson. "MEGA: malleable encryption goes awry". In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 146–163.

[37] T. von Arx and K. G. Paterson. "On the cryptographic fragility of the Telegram ecosystem". In: *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*. 2023, pp. 328–341.

[38] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila. *A Formal Security Analysis of the Signal Messaging Protocol*. Publication info: Published elsewhere. Major revision. IEEE EuroS&P 2017. 2016. URL: https://eprint.iacr.org/2016/1013 (visited on 09/19/2023).

[39] M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch. "COVAMOF: A Framework for Modeling Variability in Software Product Families". English. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3154 (2004). ISBN: 9783540229186, pp. 197–213. ISSN: 0302-9743. DOI: 10.1007/978-3-540-28630-1_12.

[40] K. Ermoshina and F. Musiani. ""Standardising by running code": the Signal protocol and de facto standardisation in end-to-end encrypted messaging". In: *Internet Histories* 3.3-4 (Oct. 2019). Publisher: Routledge _eprint: https://doi.org/10.1080/24701475.2019.1654697, pp. 343–363. ISSN: 2470-1475. DOI: 10.1080/24701475.2019.1654697. URL: https://doi.org/10.1080/24701475.2019.1654697 (visited on 09/18/2023).

[41] A. Silonosov, O. Baranovskyi, and L. Henesey. "Poster: Towards cryptographic agility in end-to-end encryption systems for computer generated telemetry data." In: *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*. HOTMOBILE '24. New York, NY, USA: Association for Computing Machinery, Feb. 2024, p. 144. ISBN: 9798400704970. DOI: 10.1145/3638550.3643624. URL: https://dl.acm.org/doi/10.1145/3638550.3643624 (visited on 04/08/2024).

[42] *Andrew Shelley,Addressing Security Concerns with Chinese Dronesand DJI Products*. en.

# Paper V

## SoK: Evolution of KEM role in cryptographic migrations for IoT systems based on E2EE approach

*Alexandr Silonosov, Emiliano Casalicchio, Lawrence Henesey*
*In: Manuscript: urn:nbn:se:bth-27xxx, preprint. 2024 (c) Authors.*

Abstract

Key Encapsulation Mechanisms (KEM) is a special type of encryption method in Public Key Cryptography (PKC) that recently was popularized by standardization authorities. The emergence of a broader use of the term in industry practice was necessitated by the discovery of the malleability property of ciphertext, which led to new approaches to asymmetric encryption. This process required the refactoring of all cryptographic software libraries that relate to the problems of cryptographic agility.

Our SoK addresses the developments of public key encryption methods and the main challenges that drive to specialization of KEM. It survey literature, cryptographic software libraries and standardization efforts about the public key encryption, key-exchange and encapsulation. We structured the challenges for the users of cryptographic software, focusing on end-to-end encryption and real-world use cases, including IoT systems. We identified main milestones of KEM evolution and structured it into four development areas. We found that the evolution of KEM is defined by a variety of mathematical foundations which always reflects on various aspects of cryptosystem, which, in case of affected security properties, can be compensated by layered and hybrid approach. Our findings indicate that science, industry practitioners and standardization bodies, propagate such approaches into state of practice by additional abstraction layers in cryptographic software. However software still not in consensus and we observed increased redundancy for KEM toolset and associated layers. Moreover industry practitioners divide into those who only increase technological stack and those who propose both, new stack and new cryptographic methods. To structure the mentioned phenomena

we introduced novel, three-facet, consumer-centered mapping of the data security domain.

# 1 Introduction

In 2024, NIST (National Institute of Standards and Technology) announced finalists for quantum-resistant cryptography competition for digital signature and key-encapsulation mechanism (KEM). The CRYSTAL-Kyber [1] KEM algorithm, implemented in the C++, has been selected as the new FIPS 203 standard. [1]. This milestone marks the culmination of many years of efforts in colloboration between regulatory authority and the scientific community to develop the next generation of cryptographic tools designed to ensure privacy, data integrity, and confidentiality in a future where quantum computers pose a real threat. As of now, quantum computers, have demonstrated the capability to solve the RSA problems involving small integers [2]. NIST expects that by 2030, quantum computing technology will reach the maturity level to break current military-grade encryption with a security level of 256 bits [3], creating a serious risk for all IT systems using conventional cryptography. In addition, new quantum-computing algorithms could be discovered in the future that lead to attacks on existing Post-Quantum Cryptography (PQC) algorithms [4].

An outdated technology stack may serve as an additional incentive for regulatory authorities to accelerate the transition to enhanced cryptography standards in both industry and government sectors. The Cybersecurity and Infrastructure Security Agency (CISA) in the United States published its guidance on Product Security Bad Practices, identifying the use of the C++ programming language as a particularly high-risk software development practice. [2] This classification references the MITRE Common Weakness Enumeration (CWE) ID CWE-119, which highlights vulnerabilities related to memory safety. [3] First, CISA designates cryptographic operations as priority code components. Second, both MITRE and CISA encourage software developers to either migrate existing code or adopt memory-safe programming languages as part of the long-term mitigation plan. Considering that approximately 60% critical vulnerabilities over the last ten years have been linked to memory safety issues. To resolve this, CISA proposes software vendors to develop a comprehensive 'memory safety' roadmap, which includes six key elements, while acknowledging that in transitioning legacy code, updating processes, and training software developers may require more time and resources [5]. In the European Union, Directive (EU) 2022/2555 [6], states that end-to-end encryption, should, where necessary, be mandatory for providers of public electronic communications networks and must not be weakened for law enforcement purposes. However, several national-level regulations and proposal exist within the

[1] https://csrc.nist.gov/pubs/fips/203/ipd

[2] https://www.cisa.gov/resources-tools/resources/product-security-bad-practices

[3] https://cwe.mitre.org/data/definitions/119.html

EU have introduced obligations to vendors to collect and store encrypted communications and associated metadata to protect national security. For example, ongoing public clarification for SOU 2023:22 [4] in Sweden, aims to establish rules for targeted data retention. This includes the collection and storage of End-to-End Encrypted (E2EE) traffic along with metadata (e.g., IP address) based on predefined criteria such as geographic location or user identity. In addition, it mandates that vendors be technically capable of providing decrypted E2EE content to law enforcement authorities without delay [7]. According to recent clarifications, such measures must not involve the creation of systematic vulnerabilities or the insertion of deliberate backdoors into encryption systems. Instead, the proposed approach favors designing E2EE in a more selective or adaptive manner, so that it is only used between recipients that are not covered by coercive measures. In this architecture, neither the sender nor the recipient would aware of whether E2EE has been relaxed. Furthermore, any extracted data under such conditions would also be encrypted by the vendor for the law enforcement agency. These development bring light to the importance of cryptographic agility - the capability of live systems to adapt new cryptographic technologies without the need to substantially replace the existing infrastructure, including data formats, codebases, and technology stacks. Otto et al. [8] expanded the concept of cryptographic agility to eight modalities, including Implementation, Compliance, and Migration agility. Apart from these technical considerations, vendors must respond to additional challenges related to the use of cryptography stemming from community-driven standards and the evolving landscape of cyber-threats [9].

This work focuses on the role of KEM as a special use of Public Key Encryption and a fundamental building block for E2EE protocols and how the aforementioned challenges influence or impact the users of cryptographic tools, often referred to as crypto-consumers [8], [4]. For example, to maintain regulatory compliance, crypto-consumers need to migrate or integrate new cryptographic API. This often involves adapting their existing technology stack, particularly when the chosen API is implemented in a different programming language than the one currently in use.

This work addresses the following research questions:

- RQ1. What are the primary challenges face by crypto-consumers, and how do these challenges impact the process of cryptographic code refactoring in live systems?
- RQ2. What are the main drivers for the evolution of KEM and related cryptographic methods?
- RQ3. How is KEM abstracted within cryptographic software libraries in both pre-quantum and post-quantum implementations?

[4] www.regeringen.se/rattsliga-dokument/statens-offentliga-utredningar/2023/05/sou-202322/

The contribution of this study is threefold. First, we review and define the challenges for cryptographic-consumers. We then provide a comprehensive survey of the research field on KEM and associated cryptographic methods, such as encryption and key agreement, focusing on KEM's state-of-the-art and systematization of knowledge in this field, with the objective to answer RQ2. Second, we survey existing cryptographic software implementations with the objective to answer RQ1 and RQ3. Third, we provide knowledge systematization and first propose to extend the notion of crypto-agility with three facets of the data-security domain.

The remainder of this paper is structured as follows: Section II presents a review of related work; in Section III, provides background across three main areas: A) KEM , B) End-to-End Encryption, and C) challenges faced by cryptographic consumers. Each subsection begins with a critical review of the area, highlighting the role of KEM in supporting migration agility. In Section IV we outline the methodology used to collect and analyze the sources referenced in this work. Section V provides the state-of-the-art KEM, including a chronological systematization to address Research Question 2 (RQ2); Section VI explores how KEM is presented in cryptographic software libraries, providing insights in response to Research Questions 1 and 3 (RQ1, RQ3). Finally, Section VII discusses the results, introduces the three-facet mapping of the data-security domain, and outlines directions for future research.

## 2 Related work

This section presents a summary of secondary studies related to KEM, migration agility in End-to-End Encryption (E2EE) protocols, and state of practice in cryptographic software.

Collecting and categorizing KEM research has been a regular agenda of the NIST institute [10] summarizing the most significant advances after each round of quantum-safe cryptography competition, which finally led to ML-KEM standard [11]. As new insights are constantly gained, NIST publish recommendations and provides guidance on using KEMs [12]. However, NIST only intends to provide the information for implementing that KEM in FIPS-validated modules, not focusing on applications or protocol.

While the performance and security characteritics KEMs have been studied in various works [13], [14], [15], [16], [17] , [18]. These analyses typically focus only on quantum-safe KEMs, specific cryptographic protocols or individual application contexts [19]. Our work provide comprehensive systematization of both the state-of-the-art and the state-of-practice, especially one that integrates insights from software standardization and cryptographic tooling.

Notable recent surveys in the field of transition to quantum-safe KEM include those by Alnahawi et al. [13] systematizing the proposals for the transition of Transport Layer Security (TLS) protocol to quantum-safe cryptography. Howe et al. [16]

analyzed nine quantum-safe KEMs and implementation challenges for PQC, such as side-channel and implementation attacks. Multiple surveys [20], [21] investigate the challenges of migrating software and IT infrastructures to quantum-secure cryptography, providing categorization of issues while focusing only on state of art for crypto-agility in literature and standardization. The work [20] highlights the need to further research the cryptographic inventory concept, which is intended to bring information about the actual use of cryptographic primitives and algorithms encoded in source code to the system infrastructure level.

Recent surveys of cryptographic software state of the practice cover criteria such as number of vulnerabilities, programming language and API characteristics of code repository characteristics (lines of code, changes, etc) [22], [23] , [24], [25] or solely focusing on specific type of cryptography [26], [27], [28], [29]. To our knowledge, there exists no comprehensive survey to provide analysis of open-source cryptographic software covering all specific implementations of KEM, PKE, key exchange or high-level protocols such as TLS.

End-to-end encryption protocols and implementations other than transport layer TLS, surveyed mostly in the context of Instant Messenger (IM) applications [30], [31] and established as Messaging Layer Security (MLS) protocol [32]. Also exist reviews of E2EE approaches for cloud storages [33], [34] and AI assistants [35] where no specific protocol is used, but rather an approach by combining cryptosystems with primitives, since having a single cryptosystem such a Homorphic Encryption could not be a solution. Studies identified multiple gaps that demonstrate the need for further research to refine and promote the E2EE approach in the areas of secure compositions, design patterns, implementation examples in cryptographic software and standards.

The role of quantum-safe cryptography in the IoT area is actively surveyed for known attacks and performance in resource-constrained devices, but only in the context of the transport layer, i.e., protocols such as TLS, MQTT, Zigbee [36], [37], [38]. Since TLS provides security for data in transit, in peer-to-peer, device-to-device communication, the data encryption features surveyed as key-agreement and authentication protocols on physical and transport layers of IoT environment [39], [40]. The security issues for end users represented as privacy and confidentiality and addressed via trust establishment mechanisms, i.e authentication in the application layer or policy based mechanisms for data processing parties [40], [41], [42]. Existing surveys of end-user Privacy-Enhancing Technologies (PET) in IoT categorize asymmetric encryption, digital signature, and authentication as distinct PET, including the cases when data processing is delegated to untrusted service providers [43]. Identified solutions includes homomorphic encryption, attribute-based encryption, searchable encryption, however the type of cryptographic key management is completely overlooked. Systematizing the data encryption proposals in alignment with common principles of key management, such as E2EE, will substantially contribute to the growing area of PET in IoT.

# 3 Background

Here, We give definitions of KEM and E2EE in IoT environments, focusing on the versatile role of KEM i.e. authentication and confidentiality. As practice shows, innovations in KEM technology are quickly reflected in cryptographic software, creating prerequisites for code refactoring on the client side i.e. cryptographic consumers. Therefore, we further propose a definition of the main challenges for crypto-consumers, including those related to cryptographic software, which ultimately leads to the need for crypto-migration.

## 3.1 Key Encapsulation Mechanism

Traditionally, public key cryptography is defined through three fundamental primitives: key exchange, asymmetric encryption, and digital signatures. In this section, we begin by introducing the concept of asymmetric encryption, and its main security properties. We then follow by presenting the Key Encapsulation Mechanism (KEM) as a successor of encryption and a practical alternative to traditional encryption schemes. Finally, we present recent advancements in KEM design and explore its relationship to key exchange protocols.

The three building blocks of PKE algorithms implementations are GenerateKey, Encrypt and Decrypt:

1. Key generation algorithm $pk, sk \leftarrow$ **GenerateKeys** returns a public and secret key pair ($pk$, $sk$).

2. Encryption algorithm $c \leftarrow$ **Encrypt (*pk*, *m*)** takes as input a public key $pk$ of recipient, and message $m$, encrypt it with $pk$ and finally outputs a ciphertext $c$.

3. Decryption algorithm $m \leftarrow$ **Decrypt (*sk*, *c*)** takes as input a secret key $sk$ of recipient, ciphertext $c$ and outputs a message $m$.

We say PKE scheme is correct if the probability of decryption failure for associated key pairs is zero [44], that is, if for all ( $pk$, $sk$ ) $\leftarrow$ **KeyGen()** and ($c$) $\leftarrow$ **Encrypt (*pk*)**, it holds that :

Pr[ Decrypt ($sk$, $c$) $\neq m$] $\leq 0$

The mathematics, used as a foundation in first generation PKE schemes, poses specific requirements on that message $m$ [45], [46] i.e. assumming the message is a very large integer number. In software implementations, usually, there is the need to transform a plain text into a number while encrypting it. At this point requirements can be easily violated when the type-casting operator converts bits from plaintext variables into bits of $m$ variable of integer type without any modification in bit structure. In applied cryptography such an approach is known as textbook encryption and recognized as insecure [47].

`C++ pseudo-code code for textbook encryption of plaintext for public key 'pk' :`

char plaintext[8] = ”secret”; int i = static$_c ast < int > (plaintext); intc = Encrypt(pk, i)$;

In first-generation cryptosystems, such as RSA and ElGamal, the ciphertext has malleability property that if exists [48], an adversary can compute encryption of the product of *m1*, *m2*, which is the product of the respective encryptions *c1*, *c2*. The ability of producing a valid ciphertext, with predictable changes on *m*, just by arithmetic manipulation with known ciphertexts is well established in the literature as a notion ’choosen ciphertext attack’ (CCA).

Obviously, it becomes impossible enough to use PKE to protect general-purpose data. Bellare et al. [49] provided the relations between different types of CCA attacks, how non-malleability (NM) property implies indistinguishability (IND), and the use of notions IND-CCA and NM-CCA. To address this limitation, a special approach later was proposed, known as hybrid encryption [50]; it allows the combination of PKE (asymmetric) with symmetric encryption algorithms to provide resistance i.e. indistinguishability for CPA and CCA attacks. Later, the role of PKE in the hybrid encryption approach was separated as Key Encapsulation Mechanism [51] to support the modularity paradigm, i.e., message M is no longer needed for the Encrypt algorithm, but it outputs C and session key K, which is generated by the underlying foundation of PKE. Cramer and Shoup [51] demonstrated why resistance to CCA implies also resistance to ’choosen plaintext attack’ therefore it is the main and strongest security property of PKE, which motivates KEM specialization from PKE. Their work first introduced the notion of encasulation in hybrid encryption principle, to differentiate between symmetric and asymmetric encryption. Later works also motivate KEM specialization from PKE so that the encryption algorithm does not require any other input except the public key of the recipient [44].

A key encapsulation mechanism, denoted $\kappa$, is a triple of algorithms KeyGen, Encaps, Decaps, so that:

1. Key generation algorithm *pk*, *sk* $\leftarrow$ **KeyGen** returns a public and secret-key pair (*pk*, *sk*).

2. Encapsulation algorithm *c*, *k* $\leftarrow$ **Encaps (*pk*)** takes as input a public key *pk*, randomly generates value *k* $\in$ K keyspace, encrypt it with *pk* and finally outputs a ciphertext *c* as well as a key *k*.

3. Decapsulation algorithm *k* $\leftarrow$ **Decaps (*sk*, *c*)** takes as input a secret key *sk*, ciphertext *c* and outputs a key *k* $\in$ K keyspace or $\perp$ meaning failure.

A KEM K is correct if for all ( *pk*, *sk* ) $\leftarrow$ **KeyGen()** and (*c*, *k*) $\leftarrow$ **Encaps (*pk*)** , it holds that :

Pr[ Decaps (*sk*, *c*) $\neq$ *k*] $\leq$ 0

We say K is correct if probability of decapsulation failure for associated key pairs is zero [44]. Comparing to PKE scheme, main algorithms are renamed to Encaps and Decaps (Encrypt, Decrypt) and second, encapsulation algorithm takes as input only public key *pk*. The main innovation of KEM is that the encryption algorithm generates a random bit string of fixed length, key *k*, conforming all requirements, encrypts it, and outputs both values. Generated Key *k* should be used as an encryption key for a symmetric-key encryption scheme to encrypt arbitrary data. As shown by Cramer et al. [51], such a combination, referred to as a hybrid- encryption, has stronger security properties than if k is generated outside KEM.

It is obvious that the adversary can distinguish between correct and incorrect ciphertexts if he knows *pk* and *k*, i.e., plain text, because the Encapsulation algorithm is considered available. The main security property of KEM is defined in terms of the indistinguishability against CCA, denoted as IND-CCA [51]. In CCA secure KEM, it is very hard for an adversary to create new ciphertexts that decapsulate into key *k1* that holds some relation with the original key *k*, for which ciphertext form is only known to the adversary.

In public key cryptography, Key Exchange (KEX), also known as key agreement, is a method that allows two parties to obtain a shared secret key through an untrusted communication channel. The first key-exchange method is Diffie-Hellman (DH)[45] which established public key cryptography.

By definition, key exchange method consists of two algorithms: GenrateKeys and CalculateKey as follows:

1. Key generation algorithm *pk*, *sk* ← **KeyGen** returns a public and secret-key pair (*pk*, *sk*).

2. Shared secret key calculation algorithm *ss* ← **Calculate (*pk1*, *sk2* )** takes as input a public key *pk1* of second party, secret key of first party , calculates value *ss* ∈ K keyspace and finally outputs *ss* value.

We say key exchange scheme is correct if for any pair of parties, and its corresponding key pairs (pk1,sk1) and (pk2,sk2) ← **KeyGen()**, the algorithm Calculate satisfies the constraint:

**Calculate (*pk1*, *sk2* ) = Calculate (*pk2*, *sk1* )**

By design, the DH method assumes key pairs *pk*, *sk* to be used only for one communication session and should not be stored for later use nor generated beforehand. The short-term, per-session context is a common use case for key exchange in protocols such as SSL, TLS, so that here key pair are referred to as ephemeral keys. While DH assumes two interactions between parties, one for each party, to exchange the ephemeral public keys for each session of communication, there exist alternatives that require more interactions.

It is obvious that, if the public key is known to one of the party in advance, KEM could be used as a key-exchange method in a non-interactive way, i.e., one of the

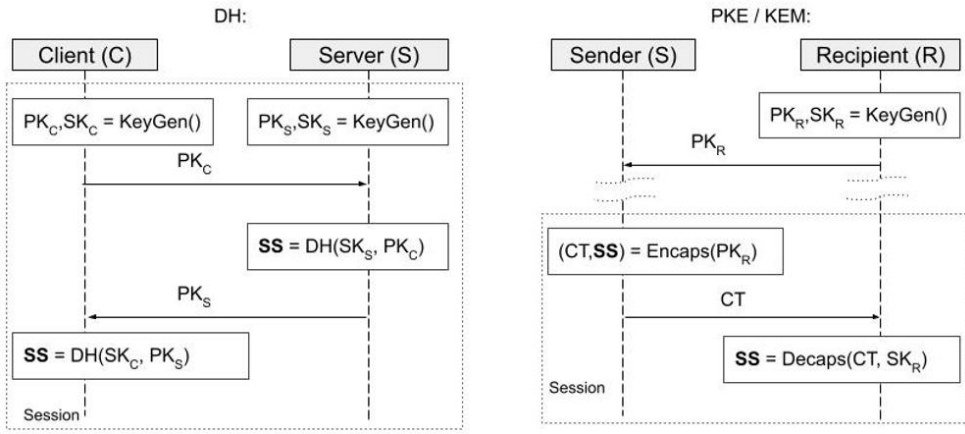


**Figure 1:** Comparison of session activity (rounds), in sequence diagrams for Diffie-Hellman (DH) key agreement and key encapsulation mechanism. SS is the session key, and CT is the ciphertext.

parties is not available at the moment. The commonalities between DH and KEM, if used for key exchange, are presented in Figure 1. Freie et al. [52] study the relationship between Non-Interactive Key Exchange (NIKE) and public key encryption (PKE) schemes. Where NIKE is a special case for key exchange protocols since the party requires no iteration to retrieve the secret key. In the NIKE case, one party can be the initiator, generate a secret key, and encrypt (encapsulate) it with the other party's private key. In this case, the key agreement process can be asynchronous and does not require both parties to be online, i.e., present or have a live communication session. Such use case is well defined in the literature, so KEM can also be viewed as a Non-Interactive Key Exchange (NIKE) method requiring only a single iteration over an untrusted cloud provider in non-interactive mode, i.e., as a message that must be delivered to the counter-party upon availability. [44]. A general solution against a man-in-the-middle attack in NIKE is to sign all protocol messages using digital signatures, thus resulting in an authenticated key exchange protocol (AKE). KEMs are also suitable for designing AKE for both authenticity and secrecy rather than using KEM only for secrecy and signatures for authenticity [53]. Jee Hea An [54] provided security analysis hybrid encryption with authentication as Encrypt-then-Sign, Sign-then-Encrypt, Encrypt-then-MAC in a setting where public and private key pair is not unique for the sender and receiver, i.e., ephemeral keys. Exist studies [55], [56], that investigate the security of combining two KEMs for authentication to replace signature algorithm. AKE with KEMs combined with several other cryptographic primitives represent an example of versatility in cryptographic inventory that needs to be handled during migration agility.

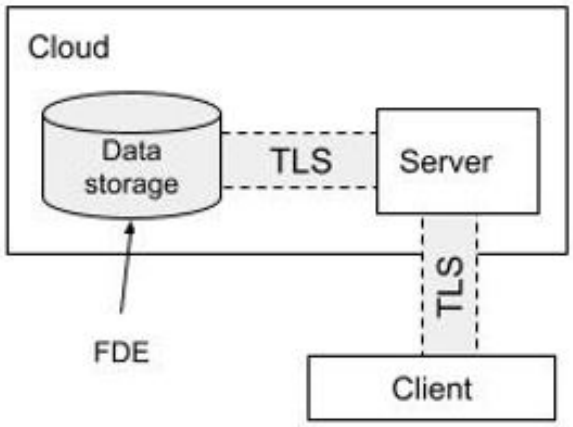


**Figure 2:** Encryption in-transit (TLS) and at-rest (FDE) in Client-Server systems

## 3.2 End-to-end encryption in IoT

The end-to-end encryption (E2EE) approach is defined in literature as client-server architecture with an integrated encryption scheme where the "ends" of the communication are situated at the client side, as a "sender" and one or more "recipients", so that they can send messages to each other via an abstract channel and where the server does not have access to the cryptographic keys necessary to read or invisibly alter the message [57] .

Usually Client-Server architecture in cloud-based systems employs encryption In-Transit provided by Transport-Layer-Security protocol, and encryption At-Rest for data storage. Figure 2 shows the data security implementation for data At-Rest and In-Transit with Full-Disk Encryption (FDE) technology and TLS protocol. For both types, Server process clear-text data.

Figure 3 represents the end-to-end encryption approach implemented for IoT system, in which clients are IoT devices i.e. sensor layer. For E2EE architecture, devices employ encryption on Application layer (ALE) [58] to encrypt data before passing for TLS protocol at transmission layer. As shown, devices transfer the encrypted data in the gray ALE area to the server, and the data arrives at cloud storage in always-encrypted form. In this case, the cloud also uses encryption in-transit and encryption at-Rest principles, but it cannot access the plain text data since it is encrypted in application layer, including key management procedures such as generating public key pairs. In IoT architecture, end users are presented in the Application Layer, which provides access to personalized services to the end users.

Hale et al. [31] proposed three core components of end-to-end encryption notion such as equivalence relations, encryption, and endness, across seven use cases in five applications such as IPSec, TLS, Signal, Zoom, and GCHQ Ghost User. The work concludes on the characteristics of each component, proposing a security notion

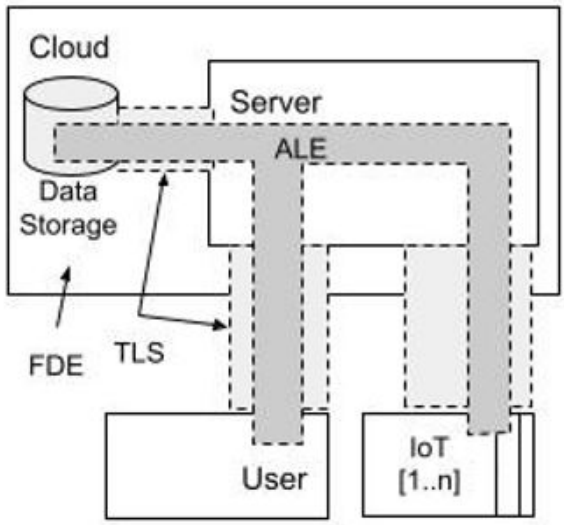


**Figure 3:** Crytography in-transit, at-rest and under-computation (ALE) in E2EE based Client-Server system.

as E2E–IND-CPA. Noorallahzade et al. [59] surveyed literature and provided a classification of existing schemes of Public-key Encryption with Keyword Search (PEKS). The study identified six classes of PEKS and forty-two constructions, which are compared by five attack models, security definition, and search method. In fact, PEKS is E2EE approach referred to as *encrypt data before outsourcing* or *searchable encryption*, for which already exist solutions in industry: LogSentinel [5] allows to search over encrypted SIEM data, Cloaked Search [6] allows E2EE approach for web-applications, and AWS Crypto Tools [7] to search over any type of encrypted data in the cloud.

## 3.3 Challenges for cryptographic consumers

Cryptographic technology became very accessible for practitioners via a wide variety of means, including software libraries, open-source data security software, data security standards, and community expertise. Initially, since OpenSSL and GnuPG cryptographic software in 1997, open-source software was a solid combination of a cryptographic library and an application, i.e., SSL protocol, email, or file encryption tools. Such cases of composite cryptographic software are still popular, for example, Bitcoin, Ethereum, and Signal. However, today, there exists a variety of cryptographic software that is not only application-neutral but also diversified in terms of

[5] https://logsentinel.com/why-is-searchable-encryption-so-important
[6] https://ironcorelabs.com/products/cloaked-search/
[7] https://docs.aws.amazon.com/aws-crypto-tools/

programming language, technological stack, and community-driven expertise for cryptographic technology. This facilitate rapid integration into software systems, however exist gaps and challenges [60]. In literature, it is referred to as a warning "don't roll your own cryptographic protocol" [61], [62]. As a result, multiple vulnerabilities have been reported for real-world software systems, where certain weaknesses relate to cryptographic implementation or incorrect use of cryptography in general.

Thus, it can be seen that exists common stages when integrating cryptography into an information system. It starts with the regulatory compliance step, which means employing standards such as FIPS [63]. Next usually follows the choice of cryptographic technology as an existing open-source software library [28]. At this stage exist, various alternatives such as integrating cryptographic software with the existing protocol or implementing its own cryptographic software as a layer on top of the existing one [60], [64]. An additional abstraction layer may evolve into its own security protocol combining different cryptographic primitives [61]. The next stage includes the changes and adaptations in the system after it turns out that there are vulnerabilities in the security protocol or in the underlying cryptographic software [23], [56]. This stage includes awareness and inclusiveness for a threat model of external cryptographic software in the IT system development lifecycle.

| Challenge | Example |
|---|---|
| Regulatory | NIST (FIPS,ISO), CISA, HIPPA, PCI-DSS, NIS2 |
| Community | External cryptographic software and expertise, Bug-bounty. |
| Cyber-threats | CVE Vulnerabilities |

**Table 1:** Challenges for consumers of cryptographic technology.

In our previous work [65], we identified two critical aspects of the state of practice related to data security in live IT systems: (1) the prevalence of vulnerabilities and (2) the role of bug-bounty programs. The increasing number of cryptographic vulnerabilities (CVE) originating in both IT systems and open-source cryptographic software libraries greatly stress live systems' robustness and adaptability due to heavy reliance on such software. On the other side, the cyber-security community can contribute to adaptability and resilience as bug-bounty programs and ethical hacking. Therefore, we define the challenges for crypto-consumers as ongoing development and adaptation of data security protocol throughout the entire system lifecycle. The challenges can be divided into three categories: 1) regulatory challenges – emerging from international data protection standards and national compliance requirements. 2) community-driven challenges – related to dependencies on third-party software, open-source libraries and external expertise. and 3) cybersecurity threats – represented by real-world cyber-security incidents and publicly reported vulnerabilities e.g., MITRE

CVE entries [8], CWE [9]. Table 1 shows examples for each challenge.

To conclude, regulations like NIS2 require companies to increase the use of cryptography for data security, resulting in the need for the transition of existing data into an encrypted form. According to reports [66] the challenge of quantum threats for data security is represented in the first category due to standards defining the implementation properties and baseline for the industry participants, i.e., cryptographic consumers.

# 4 Research Methodology

Our research process is structured in two parts. The first part focuses on a comprehensive review of the scientific literature in addressing RQ1 and RQ2. For this part, our work employs a two-step literature search process, first to review secondary studies to identify KEM state of the art in IoT area and then to review primary studies in Cryptology proceedings. The results for this part are presented in Section 5. In the second part, we address RQ3 by employing a literature search for secondary studies, coupled with an Internet search to identify the state of practice for KEM. The results for this part, including the answer for RQ1 are presented in Section 6 .

## 4.1 Data sources and search strategy for RQ1 and RQ2

For the first part, we first collect all secondary studies relevant to this SoK as the seed to build a reference set, which serves as a guideline to further identify articles that could be part of the final state of the art and state of practice. It is also used to formulate the inclusion and exclusion criteria which are then tested against the reference set. The initial reference set is based on:

- Our own pre-existing knowledge as well as knowledge from existing networks of experts in the PQC field;
- Searches in Google Scholar;
- Latest promiment works published in IACR archive [10].

The reference set was used to validate the outcomes of the search and inclusion step. We then refined the set searching Scopus databasee for works that include keywords according to the following search phase:

```
Search phrase for Scopus :
```

[8] https://cve.mitre.org
[9] https://cwe.mitre.org
[10] https://eprint.iacr.org/

( ( key W/2 ( encapsulation OR agreement OR exchange ) ) OR ( quantum W/2 ( safe OR secure ) ) ) AND ( sok OR reveiew OR survey ) )

Our inclusion criterion aims to only include secondary papers that provide survey, analysis, or SoK for traditional PKE, PQC KEM, and key exchange in the context of IoT, standardization, or cryptographic agility. Our exclusion criterion was topics of quantum key distribution. The objective of knowledge extraction at this step is to extract all relevant information to answer our research questions. Knowledge extraction protocol:

- Cryptosystem name.
- Cryptosystem abilities in terms of encryption, encapsulation or key-exchange.
- Cryptosystem definition in terms of algorithms, structure of keys and key-pair. The key generation process and structure of chipertext.
- Common cryptographic techniques that is nested into algortihms.
- Type of algebra or related mathematic in the foundation of cryptosystem or algorithm.
- Definition of security properties.
- Related cryptographic or data-security standards.
- Related software implementations.
- Associated use case, industry, application or context.
- Year of publication.

To support our research study, we then conducted a manual search to identify primary studies specifically focusing on practices and tools in PKE, KEM, and KEX in ten cryptographic “tier-one” proceedings: CRYPTO, Eurocrypt, Asiacrypt, CCS, S&P, PKC, Usenix Security, ”Theory of Cryptography” OR ”International Conference on Post-Quantum Cryptography” and the Journal of Cryptology. To define the search strategy and knowledge extraction protocol, first, we used a reference set. Next, according to Guo et al., [67], the contribution of cryptography can be read and analyzed from three main distinct perspectives: foundation, definition, and construction, which can be reflected in one of six types and seventeenth areas of contribution in total. It highlights what contribution elements we need to look at in each primary paper related to KEM. Our aim is to answer RQ2 to understand the structural and functional differences between pre-quantum and post-quantum KEM and PKE schemes.

We searched selected proceedings for works that include ”kem OR encapsulation” keywords in the Scopus database. We used the knowledge extraction protocol from the previous step with one additional item as a contribution area according to [67].

Additionally, we searched the NIST competition pubications [11] for the newest contributions in KEM area. Although some works found in the competition were not published in tier-1 proceedings, the NIST itself has an established reputation in the cryptology community . Since our work aims to survey the most up-to-date information, this suits our purpose as long as the quality of papers found is critically analyzed. The search results were then reviewed and manually filtered. In our scope, a relevant article should mention KEM, encapsulation or key-exchange or key-agreement algorithms in our scope. Results of the analysis that answer RQ1 and RQ2 are reported in Section 5.

## 4.2 Data sources and search strategy for RQ3

To address RQ3, we first collected all secondary studies in the field of cryptographic software implementations. We used the Scopus database as a primary source of data, to identify works that include keywords according to the following search phrase:

```
Search phrase for Scopus :
```

( ( crypto* W/2 ( software OR libr* ) ) ) AND ( sok OR review OR survey OR systematic OR analysis )

Our inclusion criterion aims to only include secondary papers that provide survey, analysis, or SoK for open-source cryptographic software libraries. Our exclusion criterion includes the proposals of reference implementations, i.e., the studies where only one cryptosystem implementation is proposed. Since NIST competitions, the notion of 'reference implementation' is well established in applied cryptography, and such works include a proposal of implementation of a concrete cryptosystem or primitive according to requirements defined in competition or standard. This type of software is out of our scope since it is only purposed for performance analysis. The objective of knowledge extraction at this step is to extract all relevant information used to formulate the inclusion and exclusion criteria for state of practice review. We defined the knowledge extraction protocol as follows:

- Name(s) of software library.
- Cryptosystem(s) implementations of KEM, PKE and KEX.
- Associated security protocols, industry, application or context.
- Year of publication.
- Related cryptographic or data-security standards.

In addition to retrieving data from Scopus, we conducted a manual search in the Google search engine to identify other prominent open-source cryptographic libraries

[11] https://csrc.nist.gov/projects/post-quantum-cryptography

by using the search phrase ”open source cryptographic libraries”. Building on the data collected from the first step and knowledge or insights gained by the first part of our study, we then proceeded to conduct a state of practice review. At this stage, we also applied a specific inclusion criterion with an aim to include only the software that contains two or more public key cryptosystems. The review process of selected items consists of manual analysis of each selected library’s source code repository and associated listings, following a structured extraction protocol, as outlined below:

- Source code repository type: public code repository with versioning system, or download link.
- Year of first release, and last release.
- Creator type: non-profit public community, individual person or business company.
- Cryptosystem(s) implementations of KEM, PKE, KEX, traditional and separatelly quantum safe.
- Associated security protocols.
- Related cryptographic or data-security standards.
- Programming language.
- Availability of proxy binding to other programming languages or associated technological stack.
- Dependencies from other Cryptographic software.
- Additional abstraction layers for key management or security protocols.
- Type of release configurations, target platform, and run-time parametrizations.

The protocol assuming the public availability of source code which provides the grounds of validation. We carried out a data analysis by examining and systematizing the content and trying to identify trends. The results for this part are presented in Section 6.

# 5 KEM state of the art

The initial search for secondary studies provided six hundred eleven results. Among the results, we selected surveys to include in the reference set, and we decided to limit inclusion by the 2020 year since the repeating of extracted data. Table 2 contains

| Ref/Year | PKE | KEM | KEX | Libs | Sec,..Def | Attacks | Applic. | Algs |
|---|---|---|---|---|---|---|---|---|
| [69] 04 | + | | + | | | | | Trad. |
| [70] 20 | | + | + | * | | | IoT | PQC |
| [16] 21 | | + | + | | | + | - | PQC |
| [17] 21 | | + | + | pqm4 | CCA | SCA, FA | NIST | Lattice-based PQC |
| [62] 21 | + | + | + | | CCA, CPA | SCA | Agility, Standards, | |
| [66] 22 | | + | + | | | | Agility | PQC |
| [71] 22 | | + | + | OpenSSL | | | IoT, MQTT | PQC, Trad. |
| [19] 22 | + | | | | | | IoT | Trad. ABE |
| [59] 22 | + | | | | | | IoT | Trad. ABE |
| [13] 23 | - | + | + | KemTLS | - | - | TLS | PQC, Hybrid |
| [15] 23 | + | | | | | - | | Delay-based |
| [72] 23 | + | | | | | - | NIST | PQC |
| [26] 23 | + | | | | | | | Homomorphic enc. |
| [29] 23 | + | | | * | | | | Homomorphic enc., Stand. |
| [73] 23 | + | | | | | | | Functional enc., PQC. |
| [72] 23 | | + | + | | | | | PQC. |
| [74] 23 | + | + | + | | + | | Key management | PQC. |
| [14] 24 | + | + | | | Openings | - | - | - |
| [36] 24 | | + | + | | | | IoT | PQC |
| [37] 24 | | + | + | | | | IoT | PQC |
| [75] 24 | | + | | | | | | PQC |
| [38] 24 | | + | | | | | Embedded systems | PQC |
| [20] 24 | | + | | | | | Agility | PQC |
| [75] 24 | | + | | Crypto++ | | | | PQC |

**Table 2:** Body of survey for quantum safe (PQC) and tradtional (Trad.) PKE, KEM, KEX .

the lists of twenty-four secondary studies present in our reference set, sorted in chronological order.

Alnahawi et al. [13] systematizing the proposals for transition of Transport Layer Security (TLS) protocol to quantum safe cryptography and compare between existing ephemeral Diffie-Helman key exchange [68] and lattice-based algorithms. Brunetta et al. [14] survey thirteen PKE constructions known to achieve security notions for scenarios, involving public key encryption (PKE KEM) with multiple users, some of whom may be corrupted to some extent, referred to as selective opening attacks. Medley et al. [15] mapped the role of various PKE constructions to the five encryption primitives in area of delay-based cryptography. Howe et al. [16] reviewed nine quantum-safe KEMs, various implementation challenges and provided an overview of the current and future integrations of the latest KEM into real-world systems. Ravi et al. [17] surveyed the evolution of lattice-based key-sharing schemes, including PKE and KEM, for mathematic foundations, security guarantees, general algorithmic structure and physical attacks.

The majority of the literature since 2017 has contributed to quantum-safe KEM cryptosystems motivated by the NIST competition process [11] as shown by the literature publication graph in Figure 4. These KEMs are Kyber (the finalist), FrodoKEM, NTRU, HQC, HIKE, McEliece and SIKE. These cryptosystems can be categorized by the type of mathematic problem that ensures computational difficulty. Quantum safe KEM schemes have built upon a variety of provably hard mathematic problems such as latices, isogenies, decoding a general linear code, error-correcting codes and code-based. Notably that NIST contributes to taxonomy by using the word *mechanism*, which is defined as a set of algorithms for the purpose of key-agreement, which should result to a shared secret key which then should be used with symmetric-key

cryptographic algorithms for data encryption and authentication.

Table 3 represents state of the art for traditional and quantum-secure KEM cryptosystems. The results includes comparison criteria such as mathematical foundation, number of keys in key-pair, the possibility to generate secret key asynchronously (if the secret key for the key-pair can be generated separately from public key), number of algorithms in the scheme and if key sizes have a variable size depending on number of cipher text or vise versa. The criteria *CT alg.* represents the algorithm designated for cloud service to modify the ciphertext according to provided cryptographic key. The first part of the table 3 includes traditional PKE cryptosystems and second part includes quantum safe KEM. Security of traditional PKE is based on the hardness of discrete logarithm or integer factorization in finite fields or groups based on elliptic curves or pairings. Classic KEM as well as PKE system, consists of three algorithms: GenKey, Encrypt, Decrypt. In case of RSA and Elgamal it also contain digital signature algorithms Sign, Verify and therefore it has five algorithms in total. The traditional approach for the GenKey algorithm assumes that public and private keys are generated at the same time, and it is always two keys in a key pair. Cryptosystems like IBE ABE are classic KEM with additional Setup algorithms that define some additional functionality as follows; the public key is already known since it is derived from a text string of arbitrary length and; Private key can be generated asynchronously, i.e. later at any time by the central authority; including the ability of generating multiple secret keys to decrypt the same ciphertext, therefore extending the number of possible keys in a key-pair. In some variations of IBE [76] exist four Key generation algorithm to avoid use of central authority while keeping extra functionality. Also some variations of ABE assumes a variable size of ciphertext (CT) for the same plaintext, depending on public key, while other variations assumes variable size of private key, by associating with multiple public keys. In proxy-encryption (PROXY-R) and homomorphic (FHE) systems for each key-pair it is possible to generate later unlimited number of additional keys to manipulate with ciphertext(s). Re-encrypt in PROXY-R allows to transform ciphertext for different public key. Add, mult, in FHE allows to perform multiplication or addition between two ciphertexts. In Threshold systems private key can be splitted into multiple parts therefore key-pair could be also n-size, but such parts should be generated at the same time synchronously with public key. Also these systems decryption process separated into two algorithms: partial decrypt and complete decrypt, and one more algorithm is responsible for verification of partial decryption, five algorithms in total. Broadcast systems have three algorithms, GenerateKey, Encrypt, and Decrypt, but the nature of key-pair is composite and flexible so that multiple private keys can be added to key-pair at any point in time, thus making the public key a data structure of variable length.

The research for quantum safe cryptosystems was motivated after the Shor's algorithm in 1997 [77]. Some of the cryptosystems which are quantum safe exists prior to NIST competition initiative, for example McElicie [78] based on algebraic

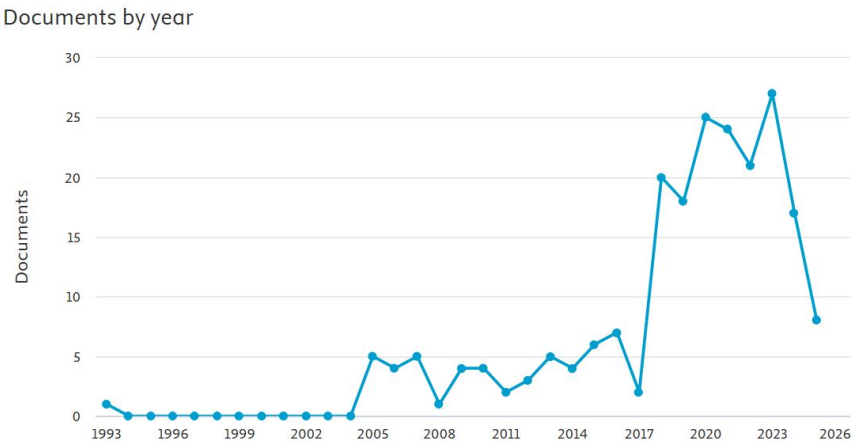


**Figure 4:** Scopus results for search query "kem OR encapsulation" for "tier-one" ten venues

coding and fully holomorphic encryption scheme BGV [79] based on the learning with error (LWE) or ring-LWE (RLWE) problems. The last one explicitly propose to exploit the malleability property of ciphertext in a robust and useful way therefore contains algorithms that generate new ciphertext from two existing ciphertexts (Mult, Add), therefore such cryptosystems may contain five additional algorithms.

Since 1996, Ajtai [80] proposed provably hard problems based on lattices. Concurrently (98), Hoffstein et al. [81] proposed the Nth order Truncated Polynomial Ring Unit (NTRU) public-key encryption scheme (NTRUEncrypt), based on hard problems over algebraically structured lattices. Regev [82] in 2005 proposed the Learning with Errors (LWE) problem, based on lattices. Over the years, new proposals based on LWE and NTRU problems have been developed, contributing to security definition, performance, algorithm, or construction. All lattice-based KEM or signature algorithms in the NIST competition are based on either of these two problems. As shown in Table 3 quantum-secure KEM cryptosystems participated in NIST competition, have strict structure with three algorithms and key generation algorithm creates a key-pair with two keys only.

The classification of cryptographic tools in Figure 5 demonstrates relations between PKE, KEM and key-agreement tools. it is possible to implement Key Agreement Protocol by using PKE schema and the KEM standard, but at the same time, additional security properties can be employed since specific schemes as HE, ABE, IBE since all them has KEM functionality.

The search results on the next step, provided 241 results. After the initial filtering of search results, eighty-seven publications remained for detailed review and categorization. For each relevant work we identified, we conducted a forward-backward search, i.e., we looked at which publications the work referred to and which publications referred to the work to find additional relevant information. We found that

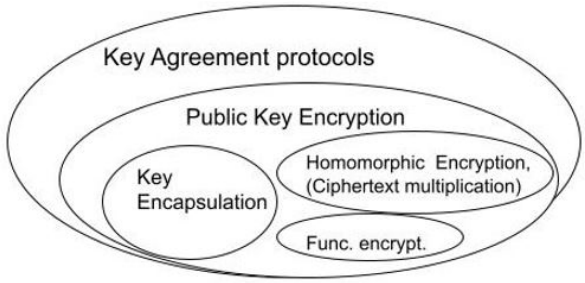


**Figure 5:** Relationship between PKC tools: key exchange, public key encryption and Key Encapsulation

a total of 34 publications were relevant to the scope of the survey. Table 5 contains KEM proposals since the first public key cryptosystem RSA, sorted by chronological order and categorized by their area of contribution and aligned with the body of standardization.

The results show that progress in quantum-safe cryptography for 30 years coexists with cryptography based on mathematic problems of the first generation, such as the difficulty of factorization and discrete logarithm, but in various different algebraic settings such as elliptic curves and pairings. However, new mathematic foundations make it obvious that the malleability property of the ciphertext produced by PKE threatens the security of encryption if it is used for arbitrary data. This leads to the contribution of new cryptographic schemes that combine PKE with symmetric encryption and other primitives, referred to as hybrid encryption. To specialize the role of PKE in such encryption schemes, the new term encapsulation was proposed in the literature. indistinguishability to choosen cipher text attack (IND-CCA) defines the security strenghts of PKE to encapsulate session key in such schemes. The body of standartization demonstrated variability while adopted the hybrid encryption schemes which resultinp to multiple concurent standards such as RSA-OAEP and then DHIES, ECIES.

Simultaneously to the studies of quantum-safe KEMs, the coming quantum threat has also motivated the research of key exchange, as well as hybrid approaches that use a combination of key material from conventional KEM and PQC-KEM primitives. Giacon et al. [83] proposed 'KEM combiners', a parallel combination of $n$ KEMs, which are supposed to be based on different hardness assumptions, i.e., conventional and quantum-safe algorithms. Other than achieving aggregated security strength, the approach also implies the composite structure of the public key. In combined KEM, the resulting key-pair consists of a vector of $n$ public keys and, respectively, $n$ secret keys. X-Wing is a hybrid KEM, proposed by Barbosa et al. [84], which uses such an approach by proposing a concrete combination of X25519 and ML-KEM, and thus first having a composite key-pair. In X-Wing, the public key contains two elements: ML-KEM public key, and X25519 public key. And since the ML-KEM standard includes three types of security strengths, each producing key-pair of different size,

| KEM/PKE | Mathematic foundation | Key-pair | Asynchronous key generation | Algorithms | Variable key size | CT algs. |
|---|---|---|---|---|---|---|
| RSA | Factoring | 2 | | 5 | | * |
| Elgamal | Discrete log. | 2 | | 5 | | * |
| IBE | Pairings | 1..n | + | 4,7 | | |
| ABE | Pairings | 1..n | + | 4 | Priv or CT | |
| Proxy. | Pairings | 2..n | + | 5 | | Re-encrypt |
| Thresh. | Pairings | 2..n | | 5 | | |
| Broadc. | Pairings | 2..n | + | 3 | Pub | |
| FHE | Lattices | 2..n | | 8 | | Add, Mult |
| Kyber | Lattices | 2 | | 3 | | |
| Saber | Lattices | 2 | | 3 | | |
| FrodoKEM | Lattices | 2 | | 3 | | |
| NTRU | Lattices | 2 | | 3 | | |
| HQC | Code | 2 | | 3 | | |
| Bike | Code | 2 | | 3 | | |
| McEliece | Code | 2 | | 3 | | |
| NewHope | Lattices | 2 | | 3 | | |
| CSIDH | Isogeni | 2 | | 3 | | |
| X-Wing | Discrete log. Lattices | 2 | | 3 | | |

**Table 3:** Comparisons of KEM and PKE cryptosystems structure . Algebraic foundation, Number of elements in a key-pair, The possibility of asynchronous key-pair generation (+), Number of algorithms, Elements of variable size: Private or Public key or Ciphertext (CT). CT Algs - algorithms to manipulate ciphertext or (*) for malleable CT.

therefore assuming also a variable key size produced by X-Wing. Table 5 represents a structured chronology of PKE and KEM studies categorized by improvement area and aligned with related methods and the body of standardization.

To answer RQ2, we find that the evolution of KEM is defined by a few key factors. First is the variety of mathematical foundations. For example, schemes based on Pairings first extended the capabilities of key-pair with the composite structure of the public key or private key, which led to the principle of flexible power for decryption. Some mathematic foundations also allow the exploitation of the malleability property of ciphertext, which leads to expanding PKE cryptosystems with new algorithms that manipulate it or create new ciphertexts from existing ones. The extent of such manipulations also can be limited by special keys. Such a concept of power is referred to as constrained power in Cryptology. The second factor is the introduction of the CCA property, which led to the KEM scheme design allowing long-term public keys if coupled with symmetric encryption, i.e., hybrid encryption. We also identified that the threat of quantum computers has given rise to a new class of KEM algorithms based on new computational hardness. But from another side, the possibility to aggregate the merits of both classes of algorithms, with a proven increase of security in terms of hardness assumptions, chose redundancy as a new direction of development.

# 6 Cryptographic software

The Scopus search conducted for the second part yielded 470 results. From these results, we selected 12 relevant surveys for analysis and knowledge extraction. A

| Years | Definition | Foundation | Construction | Standardization |
|---|---|---|---|---|
| 1978 | RSA [46] | | | |
| 1985 | ElGamal [45] | | | |
| 1986 | ECDH [85] | | | |
| 1993 | | LUC [86] | | |
| 1994 | | | RSA-OAE [87] | |
| 1997 | | | DLAES [88] | |
| 1998 | NTRU [81] | | Hybrid PKE [50] , Proxy Enc. [89] | rfc-2437 (RSAES-OAEP) |
| 1999 | | | FO-transform [90] , DHAES (ECIES) [91] | |
| 2000 | | | | SECG SEC1 (ECIES), IEEE 1363a (DLIES) |
| 2001 | | IBE-Pairings [92] | REACT [93] , KEM, RSA-KEM[94] | ANSI X9.63 (DHIES, ECIES) |
| 2002 | | | HIBE [95] | |
| 2003 | | | Hyb.PKE, KEM [51] | rfc-3447 (RSAES-OAEP) |
| 2004 | FuzzyIBE[96] | | | |
| 2005 | | | TAG-KEM [97] | |
| 2006 | KP-IBE [98] | | X25519 [99] | ISO/IEC 18033-2 (ECIES/RSA-KEM) |
| 2007 | CP-ABE [100] | | | |
| 2009 | | | | SECG SEC1 v2 |
| 2011 | | SIDH [101] | | |
| 2012 | | | | |
| 2013 | | ABE-Lattices [102] | | |
| 2015 | | | ed448 [103] | |
| 2016 | NewHope [104] | | | rfc-8017 (RSA-OAEP) |
| 2017 | | BIKE [105], SIKE [106] | | |
| 2018 | KYBER [1], Saber CSIDH[107] | | Kem Combiners [83], LEDAKem [108] | ETSI TS 103-458, 103-532 (ABE KEM) |
| 2019 | | | LIMA [109] | |
| 2020 | HAKE [110] | | | |
| 2022 | | | mmPKE-Kyber [18] | RFC-9180 (HPKE) [111] |
| 2023 | | | | RFC-9420 (MLS) |
| 2024 | | | X-Wing [84] | FIPS-203 (ML-KEM) |
| 2023 | | | | |

**Table 4:** Development chronology of PKE and KEM, separated by improvement areas and standardization (Std) efforts. Performance (Perf) includes size, time, iteractions costs. Proofs include model, results. SM-security model, Profs - proofs. Hard.

summary of this analysis is provided in table 5. As indicated in the application column, these studies examine cryptographic software with varying scopes, including timing attacks, vulnerabilities, specific cryptography algorithms, platform compatibility, or usability. Each study reviews a set of cryptographic software according to our inclusion criterion, and our analysis identified 19 cryptographic libraries that are valid for our study. An additional internet-based search yielded 36 results that comply with the inclusion criterion, 19 of which are already validated by secondary studies.

Table 6 represents a survey of 30 open-source cryptographic software libraries according to our knowledge extraction protocol. The table is divided into two parts; the upper part consists of libraries provided and maintained by commercial companies, and the bottom part contains community-driven and research-driven libraries. Within the second group, seven software libraries have a scientific report citation since it was proposed and developed by research groups. The majority of corporate

| Ref/Year | Software name(s) | Application | Standards |
|---|---|---|---|
| [112] 24 | Python Libs | | RSA, ECC, DH |
| [113] 24 | OpenSSL, GnuTLS, WolfSSL... | Timing attacks, Embedded systems | AES, HMAC |
| [114] 24 | * | Android OS | |
| [56] 24 | * | Timing attacks | |
| [22] 24 | * | CVE | |
| [25] 23 | * | Usability | |
| [24] 23 | * | PQC | |
| [29] 23 | * | homomorphic encryption | |
| [29] 22 | * | ABE | |
| [115] 22 | PQClean | PQC, Standartization | NIST |
| [116] 09 | * | Java | ECC |
| [117] 07 | * | mathematical functions | |

**Table 5:** Summary of surveys for cryptographic software. Surveys marked with (*) if covering more than three software libraries.

software contains ML-KEM, the current standard as FIPS-203 (Kyber algorithm [1]), complimented with HPKE layer for data encryption. The X25519, X448, and P-* are elliptic curve specifications that implement the ECDH key exchange protocol and are presented in the majority of software. ECIES is hybrid asymmetric encryption, where ECDH coupled with AEAD, authenticated encryption with associated data, performed by symmetric cipher ChaCha20 and Poly1305 hash function. DLIES - DH with AEAD. Hybrid KEM represent a KEM combiners approach, most often x25519 with ML-KEM as seen in software. CECPQ2 is an experimental key exchange combiner introduced by practitioners to enforce X25519 by quantum-safe NTRU-HRSS KEM [12]. T - C++ class Templates (Object-Oriented Programming). Table 7 represents cryptographic software libraries that use cryptographic primitives from other libraries or represent bindings to implementation in different programming languages like JavaScript and Go. As shown, Themis library provides bindings for more than 9 programming languages with unified abstraction layers.

To answer RQ3, we identified key-exchange protocols and hybrid encryption functionality interface (API) where KEM is used to encrypt the symmetric keys, such as ECIES, cryptobox, EVP, and HPKE. Software with HPKE layer allows the selection of two or more KEM alternatives, traditional and quantum secure, by specifying the respective API interface in a call. HPKE availability is very limited in community-driven software and is present in 80% of corporate software. *Hybrid Kem* interface provides up to six predefined combinations of KEX, such as X25519, X448 with PQC KEM such as ML-KEM, in KEM Combiner scheme [125], X-Wing [84] or experimental combiners such as CECPQ2. Combiners such as X-Wing solidify underlying KEMs' key-pairs into the resulting key-pair or composite structure. The *Secure Message* layer in Themis from Table 7, integrates asymmetric encryption and authenticated symmetric encryption based on KEX and PKE available in OpenSSL, LibreSSL or BoringSSL depending on build settings of Themis. The latter represents the solution to migrate between different cryptographic backends, however, in a

[12]https://www.chromium.org/cecpq2/

| Name | PKE, KEX | PQC KEM | Protocols | Language(s) | Abstraction Layer | CP |
|---|---|---|---|---|---|---|
| JCA/JCE | X25519,X448, RSA-OAEP | | | Java | | |
| NSS | X25519, RSA-OAEP, P* | ML-KEM | TLS, SSL | C | HPKE | |
| bouncycastle | X25519,X448,P521<br>ElGamal, RSA, GOST | ML-KEM, Bike, HQC<br>FrodoKEM, NTRU*,<br>NewHope,<br>mceliece, X-Wing | TLS, JSSE | Java, C<br>Kotlin | ECIES, HPKE | |
| WolfCrypt | X25519, X448, RSA-OAEP | ML-KEM | | C | ECIES, HPKE,<br>Hybrid KEM | |
| SymCrypt<br>by Microsoft | RSA-OAEP, P*, X25519 | ML-KEM | | C | | |
| mbedTLS | RSA, X25519, X448, P* | | SSL, TLS | C | | |
| Libsignal<br>by Signal | X25519, X448, X3DH | ML-KEM , PQXDH | Signal | Rust | | |
| libcrypto<br>in boringssl<br>fork openssl | RSA-OAEP, x25519, P* | ML-KEM | | C | HPKE<br>EVP, CMS | |
| Tink<br>Google | X25519, P* | ML-KEM | | Java<br>Go, C++<br>Obj-C<br>Python | HPKE, ECIES<br>CECPQ2 | |
| CIRCL<br>Cloudflare | X25519, X448, P*<br>Curve4Q, CPABE | ML-KEM, X-Wing<br>FrodoKEM,<br>CSIDH SIKE | | Go | HPKE<br>Hybrid KEM | |
| aws-lc fork of<br>Boringssl | | ML-KEM, | TLS | C, C++ | HPKE<br>Hybrid KEM | |
| CryptoKit<br>Apple | DH, x25519, P* | | | Obj-C<br>Swift | HPKE | |
| Crypto++ | DH, DH2, MQV, HMQV<br>FHMQV, RSA-OAEP<br>ElGamal, x25519<br>, Rabin-W, LUC<br>ECDH, ECMQV, x25519 | | | C++ | DLIES, ECIES | T |
| Cryptlib | DH, RSA-OAEP, Elgamal, P* | | SSL/TLS<br>SSH | C | Envelope | |
| Libgcrypt | X25519, X448, P*,<br>RSA-OAEP, ElGamal , | ML-KEM | | C | HPKE | S-Exp. |
| libcrypto in<br>OpenSSL | RSA-OAEP, x25519, X448, P* | | | C | HPKE<br>EVP, CMS | |
| libtomcrypt | RSA-OAEP, x25519 | | | C | | |
| Nettle | RSA-OAEP | | | C | | |
| RELIC [118] | RSA-OAEP, ECMQV , IB-NIKE | | | C | ECIES | |
| NaCl [119] | ECDH | | | C | ECIES, cryptobox | |
| libsodium<br>fork of NaCl | ECDH | | | C | ECIES, cryptobox | |
| Botan | DH, X25519, X448<br>RSA-OAEP, ElGamal<br>RSA-KEM,SM2 | ML-KEM,<br>FrodoKEM<br>Classic McEliece | TLS | | C<br>Python | DLIES, ECIES<br>ECIES-Encryptor |
| CHARM [120] | IB-NIKE, IBE | | | Python | | |
| Libcrypto in<br>LibreSSL | DH, X25519, RSA-OAEP | ML-KEM | | C | EVP, CMS | |
| RustCrypto | P-256*, SM2, RSA-OAEP | ML-KEM, X-Wing,<br>FrodoKEM | | | Rust | |
| HACL [121] | X25519 | | | F* to C | ECIES, Box | |
| PQClean [115] | | Kyber, HQC,<br>Classic McEliece | | C | - | |
| crates/HPKE | X25519, X448<br>P-256 | | | Rust | HPKE | |
| pqm4 ([122]) | | BIKE,ML-KEM | | C | | |
| LibOQS [123] | | BIKE, McEliece,<br>FrodoKEM<br>HQC, Kyber,<br>ML-KEM,<br>NTRU-Prime | | C | Separated to<br>OQSprovider | |
| Libjade [124] | X25519 | ML-KEM | | Jasmin | | |

**Table 6:** List of PKE, KEX, KEM constructions in cryptographic libraries. CP - cryptographic parametrization with software-level. Abstraction L. - layers of high-level data encryption functionality.

limited mode where the alternative is solidified into application in the software release build phase.

By using the defined challenges for crypto-consumers, provided in section 3.3 and state of practice for KEM in cryptographic software, we can elaborate the answer for RQ1 as follows. The process of cryptographic code refactoring in live systems is impacted by standards, cryptographic software, and cyber threats.

The regulatory challenge affects the selection and use of particular KEM standards

in cryptographic software. The adoption of new KEM create incompatibility with existing encrypted data. The KEM parametrization metadata does not always exist in cryptographic software, which may lead to the necessity of a higher abstraction layer over cryptographic software. The case with KEM combiners and hybridization of asymmetric encryption are good examples of when the identification of such construction should be attached to encrypted data to provide compatibility tools for future standards. The process of binding cryptographic software to different programming languages seems like a wrapping around core cryptographic primitives, but it may lead to new quasi-protocol semantics where data and process affect security properties. The vulnerabilities in the core cryptographic backend, mentioned in section 3 will stimulate further separation and development of more quasi-cryptographic code.

| Name | Language (Bindings) | Backend | Proto-s | Abstraction L. | Key Mngmt. |
|---|---|---|---|---|---|
| Conscrypt | Java | BoringSSL | | | |
| Corretto | Java | aws-lc | | | |
| GnuTLS | C | Nettle | TLS | | |
| Matrixssl | C | own, OpenSSL, Libsodium | TLS | | |
| Themis | C++ (Java, Go, JavaScript Obj-C, Swift, PHP, Python, Ruby, Rust) | own, OpenSSL, LibreSSL, BoringSSL | | Secure Cell, Secure Message | Function Name: themis-gen-ec-key-pair, themis-gen-rsa-key-pair |
| libVES | OpenSSL , liboqs | | | | |

**Table 7:** Open-source cryptographic libraries that use another(s) libraries as cryptographic backend.

# 7 Conclusions and discussion

In this section we discuss and illustrate the main results of this study, which are :

1. Challenges for cryptographic consumers
2. KEM state of (the art and state of) practice
3. Mapping data security domain.

We then outline Future research directions for identified results.

We find that asymmetric encryption evolved into the HPKE abstraction layer, which is available in C, Java, Kotlin, Go, Python, and Rust languages in different libraries, and only eight of them provide quantum-safe KEM for HPKE. For example, software in C proposes constant variables for HPKE parametrization, which solidifies the use of certain HPKE configurations with encrypted data, suggesting here open problem to tackle in future research.

The data encryption on the IoT devices and further accessing it on the applications on the end-user computer involves cross-platform software development stack and thus the need for cryptographic software availability in multiple programming languages, including all types of KEMs.

By answering RQ2, we identified multi-layered, hybrid KEMs are already present in the software, thus the next candidates for standardization. CCA property may not present for certain KEM, thus limiting its use only for key-exchange with short term public keys. By answering RQ3, We also identified how practitioners contribute with new key exchange, such as PQXDH and CECPQ2, directly into cryptographic software and state of practice. From another side, practitioners also propose software to wrap around or bind to existing cryptographic software(s) as backend, configurable during the build phase. Which assuming the work with borrowing vulnerabilities into nested chains of software layers, and additional abstraction layers for KEM parametrization and thus additional key management in case of heterogeneous cryptographic backend.

Our results proposed progress for cryptographic inventory concept and identified gaps for future research as well.

To generalize results, we propose to align and systematize the given observations, facts, phenomena, and challenges into a map of the data security domain, where the

cryptographic consumer is placed in the center, as per Figure 6. The increasing complexity of technologies and processes in the field of data security creates distillation conditions for dividing it into constituent parts. The role of participants and processes that occur in each of these parts suggests three distinct facets. The aim of the map is to help gain insights and a quick understanding of key players in the data security domain and the processes and outcomes in each facet.

Thus we introduce a novel mapping layout for three distinct facets of the data-security domain: Governance, Operations by crypto-consumers, and Engineering facet. This framework categorizes each artifact or role into one of three distinct facets. Each facet operates with KEM technology, which defines an interplay to understand the evolution of KEM technology as well as for any other cryptographic tool. The Engineering facet represents the origin and development of KEM technologies. Our findings show that in addition to Theoretical and Applied cryptography, the industry sector also contributes to KEM development with practical proposals directly as software code. Notable examples include the Signal proposal for PQXDH, CECPQ2 key agreement [126], and the abstraction layers for key management introduced by leading cryptographic libraries such as (OpenSSL, Crypto++ and Tink) for hybrid asymmetric encryption. Such contributions demonstrate the lack of stronger leadership for practical solutions as a response to long-standing PQC threats. These findings also highlight the importance of advancing the cryptographic modularity paradigm, which is a design principle emphasizing flexibility, scalability, and interoperability in cryptographic systems. The big picture of data security also aids in capitalizing on knowledge transfer for practitioners by providing new specialists with a quick and clear understanding of contributions and tasks at a specific facet within the system context.

From another side, we note that Applied Cryptography, as a science, keeps expanding to the operational facet by proposing new constructions with combinational and layered approaches, thus reflecting on software implementations with new abstraction layers. While most IoT products in real life are multi-platform and thus multi-language, cryptographic consumers may end up with quasi-protocols for data security to keep uniform software architecture. This creates a need for high-level bindings from a single crypto-software to different programming languages i.e. multiple software packages to supporting cross-platform capacity of the system code base. To avoid *roll your own crypto* cases, practitioners choose to further separate the layer of cryptographic primitives from the system code base. And even more adding a capability to switch between different backends assuming one may opt-out soon. If this keeps going, an emergent area between cryptography and software engineering will further increase. The need to validate cryptographic security properties across the entire technological stack, including bindings, will create a new research direction. On another side note, we may conclude that a certain amount of cryptographic software is derivatives (forks, partial copies) from other software (boringssl), i.e. contains the same cryptographic primitives with certain improvements, which creates a redundancy

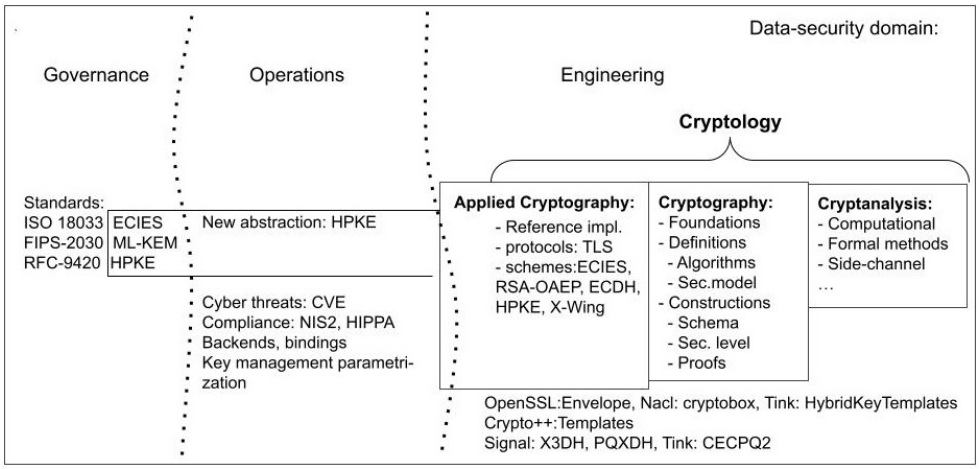


**Figure 6:** Mapping results of knowledge systematization to three facets of the data-security domain.

of alternatives. Such a redundancy lacks documentation of differentiating aspects and it propagates further to practitioners as a burden for cryptographic consumers. As an example, the list of cryptographic inventory should include not only all the primitives that were used to encrypt specific data but also the name and the version of the cryptographic backend applied in the configuration.

Also, future research may focus on bridging the gap between Applied Cryptography and operational faces, with contributions in areas such as:

- Cryptographic inventory embeddings from backends into high-level bindings across the system technological stack.
- Layered approaches in verified cryptographic backends such as HACL.
- Operational integration of modular cryptographic software within technological stack of real-world IoT systems.

# 8 Acknowledgements

Thanks to the anonymous reviewers for their feedback. This work was supported by the DigiTechPort2030 – Decarbonisation of Small Ports' Ecosystems for Efficient Environmental and Energy Management towards South Baltic Fit for 55. Interreg South Baltic.

# References

[1] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. London: IEEE, Apr. 2018, pp. 353–367. ISBN: 978-1-5386-4228-3. DOI: 10.1109/EuroSP.2018.00032. URL: https://ieeexplore.ieee.org/document/8406610/ (visited on 07/19/2024).

[2] C. Wang, Q. Hu, H. Yao, S. Wang, and Z. Pei. "Deciphering a Million-Plus RSA Integer with Ultralow Local Field Coefficient *h* and Coupling Coefficient *J* of the Ising Model by D-Wave 2000Q". en. In: *Tsinghua Science and Technology* 29.3 (June 2024). Publisher: □□□□□□□, pp. 874–882. ISSN: 1007-0214. DOI: 10.26599/TST.2023.9010059. URL: https://www.sciopen.com/article/10.26599/TST.2023.9010059 (visited on 11/21/2024).

[3] D. Moody, R. Perlner, A. Regenscheid, A. Robinson, and D. Cooper. *Transition to post-quantum cryptography standards*. Tech. rep. National Institute of Standards and Technology, 2024.

[4] L. Chen. "Cryptography Standards in Quantum Time: New Wine in an Old Wineskin?" In: *IEEE Security & Privacy* 15.4 (2017). Conference Name: IEEE Security & Privacy, pp. 51–57. ISSN: 1558-4046. DOI: 10.1109/MSP.2017.3151339. URL: https://ieeexplore.ieee.org/document/8012315 (visited on 12/13/2024).

[5] U. Cybersecurity et al. "The Case for Memory Safe Roadmaps: Why Both C-Suite Executives and Technical Experts Need to Take Memory Safe Coding Seriously". In: (2023). Publisher: CISA: Cybersecurity and Infrastructure Security Agency.

[6] *Directive - 2022/2555 - EN - EUR-Lex*. en. Doc ID: 32022L2555 Doc Sector: 3 Doc Title: Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (EU) 2016/1148 (NIS 2 Directive) (Text with EEA relevance) Doc Type: L Usr_lan: en. 2022. URL:

https://eur-lex.europa.eu/eli/dir/2022/2555/oj/eng (visited on 03/05/2025).

[7] R. o. Regeringskansliet. *Datalagring och åtkomst till elektronisk information*. sv. Text. Publisher: Regeringen och Regeringskansliet. May 2023. URL: https://www.regeringen.se/rattsliga-dokument/statens-offentliga-utredningar/2023/05/sou-202322/ (visited on 03/05/2025).

[8] D. Ott, C. Peikert, and o. w. participants other workshop. *Identifying Research Challenges in Post Quantum Cryptography Migration and Cryptographic Agility*. arXiv:1909.07353. Sept. 2019. DOI: 10.48550/arXiv.1909.07353. URL: http://arxiv.org/abs/1909.07353 (visited on 11/20/2024).

[9] A. Silonosov and L. Henesey. "Towards cryptographic agility manifesto in end-to-end encryption systems: a position paper from the perspective of crypto-consumers". In: *2024 IEEE Conference on Dependable, Autonomic and Secure Computing (DASC)*. IEEE, 2024, pp. 65–72. URL: https://ieeexplore.ieee.org/abstract/document/10798720/ (visited on 01/23/2025).

[10] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, and Y.-K. Liu. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*. en. Tech. rep. NIST Internal or Interagency Report (NISTIR) 8413. National Institute of Standards and Technology, Sept. 2022. DOI: 10.6028/NIST.IR.8413-upd1. URL: https://csrc.nist.gov/pubs/ir/8413/upd1/final (visited on 03/06/2025).

[11] N. I. o. S. a. Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. en. Tech. rep. Federal Information Processing Standard (FIPS) 203. U.S. Department of Commerce, Aug. 2024. DOI: 10.6028/NIST.FIPS.203. URL: https://csrc.nist.gov/pubs/fips/203/final (visited on 03/06/2025).

[12] G. Alagic. *Recommendations for Key-Encapsulation Mechanisms*. en. Tech. rep. NIST SP 800-227 ipd. Gaithersburg, MD: National Institute of Standards and Technology, 2025, NIST SP 800–227 ipd. DOI: 10.6028/NIST.SP.800-227.ipd. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-227.ipd.pdf (visited on 03/06/2025).

[13] N. Alnahawi, J. Müller, J. Oupický, and A. Wiesmaier. *SoK: Post-Quantum TLS Handshake*. Publication info: A major revision of an IACR publication in CIC 2024. 2023. URL: https://eprint.iacr.org/2023/1873 (visited on 12/17/2024).

[14] C. Brunetta, H. Heum, and M. Stam. “SoK: Public Key Encryption with Openings”. en. In: *Public-Key Cryptography – PKC 2024*. Ed. by Q. Tang and V. Teague. Cham: Springer Nature Switzerland, 2024, pp. 35–68. ISBN: 978-3-031-57728-4. DOI: 10.1007/978-3-031-57728-4_2.

[15] L. Medley, A. F. Loe, and E. A. Quaglia. “SoK: Delay-Based Cryptography”. English. In: IEEE Computer Society, July 2023, pp. 169–183. ISBN: 9798350321920. DOI: 10.1109/CSF57540.2023.00028. URL: https://www.computer.org/csdl/proceedings-article/csf/2023/219200a426/1On91gWwHsI (visited on 12/18/2024).

[16] J. Howe, T. Prest, and D. Apon. *SoK: How (not) to Design and Implement Post-Quantum Cryptography*. Publication info: Published elsewhere. Major revision. CT-RSA 2021. 2021. URL: https://eprint.iacr.org/2021/462 (visited on 08/25/2023).

[17] P. Ravi, J. Howe, A. Chattopadhyay, and S. Bhasin. “Lattice-based Key-sharing Schemes: A Survey”. In: *ACM Comput. Surv.* 54.1 (Jan. 2021). Place: New York, NY, USA Publisher: Association for Computing Machinery. ISSN: 0360-0300. DOI: 10.1145/3422178. URL: https://doi.org/10.1145/3422178.

[18] J. Alwen, D. Hartmann, E. Kiltz, M. Mularczyk, and P. Schwabe. *Post-Quantum Multi-Recipient Public Key Encryption*. Publication info: Preprint. 2022. URL: https://eprint.iacr.org/2022/1046 (visited on 09/20/2024).

[19] M. Rasori, M. L. Manna, P. Perazzo, and G. Dini. “A Survey on Attribute-Based Encryption Schemes Suitable for the Internet of Things”. In: *IEEE Internet of Things Journal* 9.11 (June 2022). Conference Name: IEEE Internet of Things Journal, pp. 8269–8290. ISSN: 2327-4662. DOI: 10.1109/JIOT.2022.3154039. URL: https://ieeexplore.ieee.org/abstract/document/9721417 (visited on 05/04/2024).

[20] C. Näther, D. Herzinger, S.-L. Gazdag, J.-P. Steghöfer, S. Daum, and D. Loebenberger. *Migrating Software Systems towards Post-Quantum Cryptography – A Systematic Literature Review*. arXiv:2404.12854 [cs]. June 2024. DOI: 10.48550/arXiv.2404.12854. URL: http://arxiv.org/abs/2404.12854 (visited on 12/20/2024).

[21] A. Wiesmaier, N. Alnahawi, T. Grasmeyer, J. Geißler, A. Zeier, P. Bauspieß, and A. Heinemann. *On PQC Migration and Crypto-Agility*. arXiv:2106.09599 [cs]. June 2021. DOI: 10.48550/arXiv.2106.09599. URL: http://arxiv.org/abs/2106.09599 (visited on 01/07/2025).

[22] J. Blessing, M. A. Specter, and D. J. Weitzner. “Cryptography in the Wild: An Empirical Analysis of Vulnerabilities in Cryptographic Libraries”. In: *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*. ASIA CCS ’24. New York, NY, USA: Association for Computing Machinery, July 2024, pp. 605–620. ISBN: 9798400704826. DOI: 10.1145/3634737.3657012. URL: https://dl.acm.org/doi/10.1145/3634737.3657012 (visited on 12/27/2024).

[23] D. Lazar, H. Chen, X. Wang, and N. Zeldovich. “Why does cryptographic software fail? a case study and open problems”. In: *Proceedings of 5th Asia-Pacific Workshop on Systems*. APSys ’14. New York, NY, USA: Association for Computing Machinery, June 2014, pp. 1–7. ISBN: 978-1-4503-3024-4. DOI: 10.1145/2637166.2637237. URL: https://dl.acm.org/doi/10.1145/2637166.2637237 (visited on 12/27/2024).

[24] J. Hekkala, M. Muurman, K. Halunen, and V. Vallivaara. “Implementing Post-quantum Cryptography for Developers”. en. In: *SN Computer Science* 4.4 (Apr. 2023), p. 365. ISSN: 2661-8907. DOI: 10.1007/s42979-023-01724-1. URL: https://doi.org/10.1007/s42979-023-01724-1 (visited on 01/04/2025).

[25] J. Luo, X. Yang, X. Yi, F. Han, I. Gondal, and G.-B. Huang. “A Comparative Study on Design and Usability of Cryptographic Libraries”. In: *Proceedings of the 2023 Australasian Computer Science Week*. ACSW ’23. New York, NY, USA: Association for Computing Machinery, Mar. 2023, pp. 102–111. ISBN: 9798400700057. DOI: 10.1145/3579375.3579388. URL: https://dl.acm.org/doi/10.1145/3579375.3579388 (visited on 03/10/2025).

[26] C. Gouert, D. Mouris, and N. G. Tsoutsos. “SoK: New Insights into Fully Homomorphic Encryption Libraries via Standardized Benchmarks”. In: *Proc. Priv. Enhancing Technol.* 2023.3 (2023), pp. 154–172. DOI: 10.56553/POPETS-2023-0075. URL: https://doi.org/10.56553/popets-2023-0075.

[27] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta. *Too Many Options: A Survey of ABE Libraries for Developers*. arXiv:2209.12742 [cs]. Sept. 2022. URL: http://arxiv.org/abs/2209.12742 (visited on 02/09/2024).

[28] Y. Acar, M. Backes, S. Fahl, S. Garfinkel, D. Kim, M. L. Mazurek, and C. Stransky. “Comparing the Usability of Cryptographic APIs”. In: *2017 IEEE Symposium on Security and Privacy (SP)*. ISSN: 2375-1207. May 2017, pp. 154–171. DOI: 10.1109/SP.2017.52. URL: https://ieeexplore-ieee-org.miman.bib.bth.se/abstract/document/7958576 (visited on 06/08/2024).

[29] T. V. T. Doan, M.-L. Messai, G. Gavin, and J. Darmont. “A survey on implementations of homomorphic encryption schemes”. en. In: *The Journal of Supercomputing* 79.13 (Sept. 2023). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 13 Publisher: Springer US, pp. 15098–15139. ISSN: 1573-0484. DOI: 10.1007/s11227-023-05233-z. URL: https://link.springer.com/article/10.1007/s11227-023-05233-z (visited on 03/10/2025).

[30] A. Dimeo, F. Gohla, D. Goßen, and N. Lockenvitz. *SoK: Multi-Device Secure Instant Messaging*. Publication info: Preprint. MINOR revision. 2021. URL: https://eprint.iacr.org/2021/498 (visited on 03/07/2025).

[31] B. Hale and C. Komlo. *On End-to-End Encryption*. Publication info: Preprint. MINOR revision. 2022. URL: https://eprint.iacr.org/2022/449 (visited on 04/02/2024).

[32] R. Barnes, B. Beurdouche, R. Robert, J. Millican, E. Omara, and K. Cohn-Gordon. *RFC 9420: The Messaging Layer Security (MLS) Protocol*. USA: RFC Editor, June 2023.

[33] C. Manthiramoorthy, K. M. S. Khan, and N. A. A. “Comparing Several Encrypted Cloud Storage Platforms”. en. In: *International Journal of Mathematics, Statistics, and Computer Science* 2 (2024), pp. 44–62. ISSN: 2704-1069. DOI: 10.59543/ijmscs.v2i.7971. URL: https://ijmscs.org/index.php/ijmscs/article/view/7971 (visited on 04/02/2024).

[34] J. Hofmann and K. T. Truong. “End-to-End Encrypted Cloud Storage in the Wild: A Broken Ecosystem”. In: *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*. CCS ’24. New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 3988–4001. ISBN: 9798400706363. DOI: 10.1145/3658644.3690309. URL: https://dl.acm.org/doi/10.1145/3658644.3690309 (visited on 03/14/2025).

[35] M. Knodel, A. Fábrega, D. Ferrari, J. Leiken, B. L. Hou, D. Yen, S. d. Alfaro, K. Cho, and S. Park. *How To Think About End-To-End Encryption and AI: Training, Processing, Disclosure, and Consent*. Publication info: Preprint. 2024. URL: https://eprint.iacr.org/2024/2086 (visited on 03/14/2025).

[36] P. R. Babu, S. A. P. Kumar, A. G. Reddy, and A. K. Das. “Quantum secure authentication and key agreement protocols for IoT-enabled applications: A comprehensive survey and open challenges”. In: *Computer Science Review* 54 (Nov. 2024), p. 100676. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2024.100676. URL: https://www.sciencedirect.com/science/article/pii/S1574013724000601 (visited on 01/07/2025).

[37] T. Liu, G. Ramachandran, and R. Jurdak. *Post-Quantum Cryptography for Internet of Things: A Survey on Performance and Optimization*. arXiv:2401.17538 [cs]. Jan. 2024. DOI: 10.48550/arXiv.2401.17538. URL: http://arxiv.org/abs/2401.17538 (visited on 12/13/2024).

[38] M. Pursche, N. Puch, S. N. Peters, and M. P. Heinl. *SoK: The Engineer's Guide to Post-Quantum Cryptography for Embedded Devices*. Publication info: Preprint. 2024. URL: https://eprint.iacr.org/2024/1345 (visited on 11/19/2024).

[39] B. Gupta and M. Quamara. "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols". en. In: *Concurrency and Computation: Practice and Experience* 32.21 (2020). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.4946, e4946. ISSN: 1532-0634. DOI: 10.1002/cpe.4946. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.4946 (visited on 03/06/2025).

[40] P. P. Ray. "A survey on Internet of Things architectures". In: *Journal of King Saud University - Computer and Information Sciences* 30.3 (July 2018), pp. 291–319. ISSN: 1319-1578. DOI: 10.1016/j.jksuci.2016.10.003. URL: https://www.sciencedirect.com/science/article/pii/S1319157816300799 (visited on 03/06/2025).

[41] M. b. Mohamad Noor and W. H. Hassan. "Current research on Internet of Things (IoT) security: A survey". In: *Computer Networks* 148 (Jan. 2019), pp. 283–294. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2018.11.025. URL: https://www.sciencedirect.com/science/article/pii/S1389128618307035 (visited on 03/13/2025).

[42] S. R. Yagna Dharmendrakumar Patel. *(PDF) SoK: Evaluating IoT Security Strategies across Diverse Architectures*. en. Feb. 2025. URL: https://www.researchgate.net/publication/388722445_SoK_Evaluating_IoT_Security_Strategies_across_Diverse_Architectures (visited on 03/13/2025).

[43] L. Malina, P. Dzurenda, S. Ricci, J. Hajny, G. Srivastava, R. Matulevičius, A.-A. O. Affia, M. Laurent, N. H. Sultan, and Q. Tang. "Post-Quantum Era Privacy Protection for Intelligent Infrastructures". In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 36038–36077. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3062201. URL: https://ieeexplore.ieee.org/document/9363165 (visited on 03/13/2025).

[44] S. Coretti, U. Maurer, and B. Tackmann. "A Constructive Perspective on Key Encapsulation". en. In: *Number Theory and Cryptography: Papers in Honor of Johannes Buchmann on the Occasion of His 60th Birthday*. Ed. by M. Fischlin and S. Katzenbeisser. Berlin, Heidelberg: Springer, 2013, pp. 226–

239. ISBN: 978-3-642-42001-6. DOI: 10.1007/978-3-642-42001-6_16. URL: https://doi.org/10.1007/978-3-642-42001-6_16 (visited on 12/12/2024).

[45] T. Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (July 1985). Conference Name: IEEE Transactions on Information Theory, pp. 469–472. ISSN: 1557-9654. DOI: 10.1109/TIT.1985.1057074. URL: https://ieeexplore.ieee.org/document/1057074 (visited on 01/19/2025).

[46] R. L. Rivest, A. Shamir, and L. Adleman. “A method for obtaining digital signatures and public-key cryptosystems”. In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342. URL: https://dl.acm.org/doi/10.1145/359340.359342 (visited on 12/16/2024).

[47] D. Boneh, A. Joux, and P. Q. Nguyen. “Why Textbook ElGamal and RSA Encryption Are Insecure”. en. In: *Advances in Cryptology — ASIACRYPT 2000*. Ed. by T. Okamoto. Berlin, Heidelberg: Springer, 2000, pp. 30–43. ISBN: 978-3-540-44448-0. DOI: 10.1007/3-540-44448-3_3.

[48] D. Dolev, C. Dwork, and M. Naor. “Non-malleable cryptography”. In: *Proceedings of the twenty-third annual ACM symposium on Theory of Computing*. STOC ’91. New York, NY, USA: Association for Computing Machinery, Jan. 1991, pp. 542–552. ISBN: 978-0-89791-397-3. DOI: 10.1145/103418.103474. URL: https://dl.acm.org/doi/10.1145/103418.103474 (visited on 01/17/2025).

[49] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. “Relations among notions of security for public-key encryption schemes”. en. In: *Advances in Cryptology — CRYPTO ’98*. Ed. by H. Krawczyk. Berlin, Heidelberg: Springer, 1998, pp. 26–45. ISBN: 978-3-540-68462-6. DOI: 10.1007/BFb0055718.

[50] R. Cramer and V. Shoup. “A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack”. en. In: *Advances in Cryptology — CRYPTO ’98*. Ed. by H. Krawczyk. Berlin, Heidelberg: Springer, 1998, pp. 13–25. ISBN: 978-3-540-68462-6. DOI: 10.1007/BFb0055717.

[51] R. Cramer and V. Shoup. “Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack”. In: *SIAM Journal on Computing* 33 (Dec. 2003). Publisher: Society for Industrial & Applied Mathematics, pp. 167–226. ISSN: 00975397. DOI: 10.1137/S0097539702403773. URL: http://miman.bib.bth.se/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=12338543&site=ehost-live&scope=site (visited on 12/12/2024).

[52] E. S. V. Freire, D. Hofheinz, E. Kiltz, and K. G. Paterson. “Non-Interactive Key Exchange”. en. In: *Public-Key Cryptography – PKC 2013*. Ed. by K. Kurosawa and G. Hanaoka. Berlin, Heidelberg: Springer, 2013, pp. 254–271. ISBN: 978-3-642-36362-7. DOI: 10.1007/978-3-642-36362-7_17.

[53] S. Galbraith. “Authenticated key exchange for SIDH”. In: *IACR Cryptol. ePrint Arch.* (2018). (Visited on 03/16/2025).

[54] J. H. An. *Authenticated Encryption in the Public-Key Setting: Security Notions and Analyses*. Publication info: Published elsewhere. Unknown where it was published. 2001. URL: https://eprint.iacr.org/2001/079 (visited on 12/16/2024).

[55] C. de Saint Guilhem, N. P. Smart, and B. Warinschi. “Generic Forward-Secure Key Agreement Without Signatures”. en. In: *Information Security*. Ed. by P. Q. Nguyen and J. Zhou. Cham: Springer International Publishing, 2017, pp. 114–133. ISBN: 978-3-319-69659-1. DOI: 10.1007/978-3-319-69659-1_7.

[56] J. Jancar, M. Fourné, D. De Almeida Braga, M. Sabt, P. Schwabe, G. Barthe, P.-A. Fouque, and Y. Acar. “They’re not that hard to mitigate: What Cryptographic Library Developers Think About Timing Attacks”. en. In: Gesellschaft für Informatik e.V., 2024, pp. 143–144. ISBN: 978-3-88579-737-1. URL: https://dl.gi.de/handle/20.500.12116/43592 (visited on 03/15/2025).

[57] S. Scheffler and J. Mayer. “SoK: Content Moderation for End-to-End Encryption”. en. In: *Proceedings on Privacy Enhancing Technologies* 2023.2 (Apr. 2023), pp. 403–429. ISSN: 2299-0984. DOI: 10.56553/popets-2023-0060. URL: https://petsymposium.org/popets/2023/popets-2023-0060.php (visited on 10/30/2023).

[58] Y. Ding and K. Klein. “Model-Driven Application-Level Encryption for the Privacy of E-health Data”. In: *2010 International Conference on Availability, Reliability and Security*. Feb. 2010, pp. 341–346. DOI: 10.1109/ARES.2010.91. URL: https://ieeexplore.ieee.org/abstract/document/5438071 (visited on 04/08/2024).

[59] M. H. Noorallahzade, R. Alimoradi, and A. Gholami. “A Survey on Public Key Encryption with Keyword Search: Taxonomy and Methods”. en. In: *International Journal of Mathematics and Mathematical Sciences* 2022 (Mar. 2022). Publisher: Hindawi, e3223509. ISSN: 0161-1712. DOI: 10.1155/2022/3223509. URL: https://www.hindawi.com/journals/ijmms/2022/3223509/ (visited on 08/30/2023).

[60] K. Fischer, I. Trummová, P. Gajland, Y. Acar, S. Fahl, and M. A. Sasse. *On The Challenges of Bringing Cryptography from Papers to Products: Results from an Interview Study with Experts (Extended Version)*. eng. Publisher: Zenodo.

Oct. 2023. DOI: 10.5281/zenodo.8404611. URL: https://zenodo.org/records/8404611 (visited on 06/08/2024).

[61] K. G. Paterson, M. Scarlata, and K. T. Truong. “Three Lessons From Threema: Analysis of a Secure Messenger”. en. In: 2023, pp. 1289–1306. ISBN: 978-1-939133-37-3. (Visited on 03/14/2025).

[62] K. G. Paterson. “The Cyber Security Body of Knowledge v1.1.0, 2021”. In: Section: Applied Cryptography. University of Bristol, 2021. URL: https://www.cybok.org/.

[63] N. Huaman, J. Suray, J. H. Klemmer, M. Fourné, S. Amft, I. Trummová, Y. Acar, and S. Fahl. “”You have to read 50 different {RFCs} that contradict each other”: An Interview Study on the Experiences of Implementing Cryptographic Standards”. en. In: 2024, pp. 7249–7266. ISBN: 978-1-939133-44-1. URL: https://www.usenix.org/conference/usenixsecurity24/presentation/huaman (visited on 09/27/2024).

[64] P. L. Gorski, L. L. Iacono, D. Wermke, C. Stransky, S. Möller, Y. Acar, and S. Fahl. “Developers Deserve Security Warnings, Too: On the Effect of Integrated Security Advice on Cryptographic {API} Misuse”. en. In: 2018, pp. 265–281. ISBN: 978-1-939133-10-6. URL: https://www.usenix.org/conference/soups2018/presentation/gorski (visited on 10/12/2023).

[65] A. Silonosov and L. Henesey. “Towards cryptographic agility manifesto in end-to-end encryption systems: a position paper from the perspective of crypto-consumers”. In: *2024 IEEE Conference on Dependable, Autonomic and Secure Computing (DASC)*. ISSN: 2837-0740. Nov. 2024, pp. 65–72. DOI: 10.1109/DASC64200.2024.00015. URL: https://ieeexplore.ieee.org/abstract/document/10798720 (visited on 01/23/2025).

[66] D. J. Bernstein and A. H. Tanja Lange. *Post-Quantum Cryptography - Integration study | ENISA*. en. Nov. 2022. URL: https://www.enisa.europa.eu/publications/post-quantum-cryptography-integration-study (visited on 01/21/2025).

[67] F. Guo, W. Susilo, X. Chen, P. Jiang, J. Lai, and Z. Zhao. *SOK: Research Motivations of Public-Key Cryptography*. Publication info: Preprint. 2023. URL: https://eprint.iacr.org/2023/715 (visited on 11/01/2024).

[68] W. Diffie and M. Hellman. “New directions in cryptography”. In: *IEEE Transactions on Information Theory* 22.6 (Nov. 1976). Conference Name: IEEE Transactions on Information Theory, pp. 644–654. ISSN: 1557-9654. DOI: 10.1109/TIT.1976.1055638. URL: https://ieeexplore.ieee.org/document/1055638 (visited on 03/06/2025).

[69] R. Dutta, R. Barua, and P. Sarkar. *Pairing-Based Cryptographic Protocols : A Survey*. Publication info: Published elsewhere. Unknown where it was published. 2004. URL: https://eprint.iacr.org/2004/064 (visited on 01/20/2025).

[70] T. M. Fernández-Caramés. "From Pre-Quantum to Post-Quantum IoT Security: A Survey on Quantum-Resistant Cryptosystems for the Internet of Things". In: *IEEE Internet of Things Journal* 7.7 (July 2020). Conference Name: IEEE Internet of Things Journal, pp. 6457–6480. ISSN: 2327-4662. DOI: 10.1109/JIOT.2019.2958788. URL: https://ieeexplore.ieee.org/document/8932459 (visited on 10/31/2024).

[71] J.-A. Septien-Hernandez, M. Arellano-Vazquez, M. A. Contreras-Cruz, and J.-P. Ramirez-Paredes. "A Comparative Study of Post-Quantum Cryptosystems for Internet-of-Things Applications". en. In: *Sensors* 22.2 (Jan. 2022). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute, p. 489. ISSN: 1424-8220. DOI: 10.3390/s22020489. URL: https://www.mdpi.com/1424-8220/22/2/489 (visited on 01/03/2025).

[72] D.-T. Dam, T.-H. Tran, V.-P. Hoang, C.-K. Pham, and T.-T. Hoang. "A Survey of Post-Quantum Cryptography: Start of a New Race". en. In: *Cryptography* 7.3 (Sept. 2023). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 40. ISSN: 2410-387X. DOI: 10.3390/cryptography7030040. URL: https://www.mdpi.com/2410-387X/7/3/40 (visited on 01/07/2025).

[73] C. Mascia, M. Sala, and I. Villa. "A SURVEY ON FUNCTIONAL ENCRYPTION". English. In: *Advances in Mathematics of Communications* 17.5 (2023), pp. 1251–1289. ISSN: 1930-5346. DOI: 10.3934/amc.2021049.

[74] S. Rana, F. K. Parast, B. Kelly, Y. Wang, and K. B. Kent. "A comprehensive survey of cryptography key management systems". In: *Journal of Information Security and Applications* 78 (Nov. 2023), p. 103607. ISSN: 2214-2126. DOI: 10.1016/j.jisa.2023.103607. URL: https://www.sciencedirect.com/science/article/pii/S2214212623001916 (visited on 03/21/2025).

[75] M. Harmalkar, K. Jain, and P. Krishnan. "A Survey of Post Quantum Key Encapsulation Mechanism". In: *2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*. Jan. 2024, pp. 141–149. DOI: 10.1109/ICMCSI61536.2024.00028. URL: https://ieeexplore.ieee.org/abstract/document/10493946 (visited on 01/01/2025).

[76] S. S. Al-Riyami and K. G. Paterson. "Certificateless Public Key Cryptography". en. In: *Advances in Cryptology - ASIACRYPT 2003*. Ed. by C.-S. Laih. Berlin, Heidelberg: Springer, 2003, pp. 452–473. ISBN: 978-3-540-40061-5. DOI: 10.1007/978-3-540-40061-5_29.

[77] C. Gidney and M. Ekerå. “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits”. en-GB. In: *Quantum* 5 (Apr. 2021). Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, p. 433. DOI: 10.22331/q-2021-04-15-433. URL: https://quantum-journal.org/papers/q-2021-04-15-433/ (visited on 01/21/2025).

[78] R. McEliece. “A public key cryptosystem based on algebraic coding theory”. In: 1978. (Visited on 03/17/2025).

[79] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. “(Leveled) fully homomorphic encryption without bootstrapping”. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS ’12. New York, NY, USA: Association for Computing Machinery, Jan. 2012, pp. 309–325. ISBN: 978-1-4503-1115-1. DOI: 10.1145/2090236.2090262. URL: https://doi.org/10.1145/2090236.2090262 (visited on 03/17/2025).

[80] M. Ajtai. “Generating hard instances of lattice problems (extended abstract)”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*. STOC ’96. New York, NY, USA: Association for Computing Machinery, July 1996, pp. 99–108. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237838. URL: https://dl.acm.org/doi/10.1145/237814.237838 (visited on 03/16/2025).

[81] J. Hoffstein, J. Pipher, and J. H. Silverman. “NTRU: A ring-based public key cryptosystem”. en. In: *Algorithmic Number Theory*. Ed. by J. P. Buhler. Berlin, Heidelberg: Springer, 1998, pp. 267–288. ISBN: 978-3-540-69113-6. DOI: 10.1007/BFb0054868.

[82] O. Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *Journal of the ACM (JACM)* 56.6 (2009). Publisher: ACM New York, NY, USA, pp. 1–40.

[83] F. Giacon, F. Heuer, and B. Poettering. “KEM Combiners”. en. In: *Public-Key Cryptography – PKC 2018*. Ed. by M. Abdalla and R. Dahab. Cham: Springer International Publishing, 2018, pp. 190–218. ISBN: 978-3-319-76578-5. DOI: 10.1007/978-3-319-76578-5_7.

[84] M. Barbosa, D. Connolly, J. D. Duarte, A. Kaiser, P. Schwabe, K. Varner, and B. Westerbaan. “X-Wing”. en. In: *IACR Communications in Cryptology* 1.1 (Apr. 2024). Number: 1. ISSN: 3006-5496. DOI: 10.62056/a3qj89n4e. URL: https://cic.iacr.org/p/1/1/21 (visited on 02/08/2025).

[85] V. S. Miller. “Use of Elliptic Curves in Cryptography”. en. In: *Advances in Cryptology — CRYPTO ’85 Proceedings*. Ed. by H. C. Williams. Berlin, Heidelberg: Springer, 1986, pp. 417–426. ISBN: 978-3-540-39799-1. DOI: 10.1007/3-540-39799-X_31.

[86] P. J. Smith and M. J. J. Lennon. “LUC: A New Public Key System”. In: *Proceedings of the IFIP TC11, Ninth International Conference on Information Security: Computer Security*. IFIP/Sec ’93. NLD: North-Holland Publishing Co., May 1993, pp. 103–117. ISBN: 978-0-444-81748-8. (Visited on 01/15/2025).

[87] M. Bellare and P. Rogaway. “Optimal asymmetric encryption–How to encrypt with RSA”. In: *Eurocrypt*. Vol. 94. 1994, pp. 92–111.

[88] M. Bellare and P. Rogaway. “Minimizing the use of random oracles in authenticated encryption schemes”. en. In: *Information and Communications Security*. Ed. by Y. Han, T. Okamoto, and S. Qing. Berlin, Heidelberg: Springer, 1997, pp. 1–16. ISBN: 978-3-540-69628-5. DOI: 10.1007/BFb0028457.

[89] M. Blaze, G. Bleumer, and M. Strauss. “Divertible protocols and atomic proxy cryptography”. en. In: *Advances in Cryptology — EUROCRYPT’98*. Ed. by K. Nyberg. Berlin, Heidelberg: Springer, 1998, pp. 127–144. ISBN: 978-3-540-69795-4. DOI: 10.1007/BFb0054122.

[90] E. Fujisaki and T. Okamoto. “Secure Integration of Asymmetric and Symmetric Encryption Schemes”. en. In: *Advances in Cryptology — CRYPTO’ 99*. Ed. by M. Wiener. Berlin, Heidelberg: Springer, 1999, pp. 537–554. ISBN: 978-3-540-48405-9. DOI: 10.1007/3-540-48405-1_34.

[91] M. Abdalla, M. Bellare, and P. Rogaway. *DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem*. Publication info: Published elsewhere. Appeared in the THEORY OF CRYPTOGRAPHY LIBRARY and has been included in the ePrint Archive. 1999. URL: https://eprint.iacr.org/1999/007 (visited on 01/13/2025).

[92] D. Boneh and M. Franklin. *Identity Based Encryption From the Weil Pairing*. Publication info: Published elsewhere. Unknown where it was published. 2001. URL: https://eprint.iacr.org/2001/090 (visited on 12/12/2023).

[93] T. Okamoto and D. Pointcheval. “REACT: Rapid Enhanced-Security Asymmetric Cryptosystem Transform”. en. In: *Topics in Cryptology — CT-RSA 2001*. Ed. by D. Naccache. Berlin, Heidelberg: Springer, 2001, pp. 159–174. ISBN: 978-3-540-45353-6. DOI: 10.1007/3-540-45353-9_13.

[94] V. Shoup. *A Proposal for an ISO Standard for Public Key Encryption*. Publication info: Published elsewhere. Unknown where it was published. 2001. URL: https://eprint.iacr.org/2001/112 (visited on 02/20/2024).

[95] C. Gentry and A. Silverberg. “Hierarchical ID-Based Cryptography”. en. In: *Advances in Cryptology — ASIACRYPT 2002*. Ed. by Y. Zheng. Berlin, Heidelberg: Springer, 2002, pp. 548–566. ISBN: 978-3-540-36178-7. DOI: 10.1007/3-540-36178-2_34.

[96] A. Sahai and B. Waters. *Fuzzy Identity Based Encryption*. Publication info: Published elsewhere. Unknown where it was published. 2004. URL: https://eprint.iacr.org/2004/086 (visited on 02/22/2024).

[97] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. "Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM". en. In: *Advances in Cryptology – EUROCRYPT 2005*. Ed. by R. Cramer. Berlin, Heidelberg: Springer, 2005, pp. 128–146. ISBN: 978-3-540-32055-5. DOI: 10.1007/11426639_8.

[98] V. Goyal, O. Pandey, A. Sahai, and B. Waters. *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data*. Publication info: Published elsewhere. Extended abstract to appear in ACM CCS 2006. This is the full version. 2006. URL: https://eprint.iacr.org/2006/309 (visited on 10/31/2023).

[99] D. J. Bernstein. "Curve25519: New Diffie-Hellman Speed Records". en. In: *Public Key Cryptography - PKC 2006*. Ed. by M. Yung, Y. Dodis, A. Kiayias, and T. Malkin. Berlin, Heidelberg: Springer, 2006, pp. 207–228. ISBN: 978-3-540-33852-9. DOI: 10.1007/11745853_14.

[100] J. Bethencourt, A. Sahai, and B. Waters. "Ciphertext-Policy Attribute-Based Encryption". en. In: *2007 IEEE Symposium on Security and Privacy (SP '07)*. Berkeley, CA: IEEE, May 2007, pp. 321–334. ISBN: 978-0-7695-2848-9. DOI: 10.1109/SP.2007.11.

[101] L. D. Feo, D. Jao, and J. Plût. *Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies*. Publication info: Published elsewhere. PQCrypto 2011. 2011. URL: https://eprint.iacr.org/2011/506 (visited on 01/13/2025).

[102] X. Boyen. "Attribute-Based Functional Encryption on Lattices". en. In: *Theory of Cryptography*. Ed. by A. Sahai. Vol. 7785. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 122–142. ISBN: 978-3-642-36593-5 978-3-642-36594-2. DOI: 10.1007/978-3-642-36594-2_8. URL: http://link.springer.com/10.1007/978-3-642-36594-2_8 (visited on 02/27/2024).

[103] M. Hamburg. *Ed448-Goldilocks, a new elliptic curve*. Publication info: Published elsewhere. NIST ECC Workshop 2015. 2015. URL: https://eprint.iacr.org/2015/625 (visited on 01/13/2025).

[104] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. "Post-quantum key {Exchange—A} new hope". In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016, pp. 327–343. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim (visited on 01/10/2025).

[105] N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. Aguilar Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, and G. Zémor. *BIKE: Bit Flipping Key Encapsulation*. Dec. 2017. URL: https://hal.science/hal-01671903 (visited on 01/11/2025).

[106] R. A. M. C. C. C. L. D. F. B. H. D. Jao A. Jalali B. Kozie B. LaMacchia P. Longa M. Naehrig J. Renes V. Soukharev and D. Urbanik. *SIKE: Supersingular Isogeny Key Encapsulation.* 2017. URL: https://sike.org.

[107] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. "CSIDH: An Efficient Post-Quantum Commutative Group Action". In: *Advances in Cryptology – ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, Dec. 2018, pp. 395–427. ISBN: 978-3-030-03331-6. DOI: 10.1007/978-3-030-03332-3_15. URL: https://doi.org/10.1007/978-3-030-03332-3_15 (visited on 02/12/2025).

[108] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini. "LEDAkem: A Post-quantum Key Encapsulation Mechanism Based on QC-LDPC Codes". en. In: *Post-Quantum Cryptography*. Ed. by T. Lange and R. Steinwandt. Cham: Springer International Publishing, 2018, pp. 3–24. ISBN: 978-3-319-79063-3. DOI: 10.1007/978-3-319-79063-3_1.

[109] M. Kraitsberg, Y. Lindell, V. Osheter, N. P. Smart, and Y. Talibi Alaoui. "Adding Distributed Decryption and Key Generation to a Ring-LWE Based CCA Encryption Scheme". en. In: *Information Security and Privacy*. Ed. by J. Jang-Jaccard and F. Guo. Cham: Springer International Publishing, 2019, pp. 192–210. ISBN: 978-3-030-21548-4. DOI: 10.1007/978-3-030-21548-4_11.

[110] B. Dowling, T. B. Hansen, and K. G. Paterson. "Many a Mickle Makes a Muckle: A Framework for Provably Quantum-Secure Hybrid Key Exchange". en. In: *Post-Quantum Cryptography*. Ed. by J. Ding and J.-P. Tillich. Cham: Springer International Publishing, 2020, pp. 483–502. ISBN: 978-3-030-44223-1. DOI: 10.1007/978-3-030-44223-1_26.

[111] J. Alwen, B. Blanchet, E. Hauck, E. Kiltz, B. Lipp, and D. Riepel. "Analysing the HPKE Standard". en. In: *Advances in Cryptology – EUROCRYPT 2021*. Ed. by A. Canteaut and F.-X. Standaert. Cham: Springer International Publishing, 2021, pp. 87–116. ISBN: 978-3-030-77870-5. DOI: 10.1007/978-3-030-77870-5_4.

[112] X. Ye, T. G. Tan, and J. Zhou. “Towards Discovering Quantum-Threats for Applications Using Open-Source Libraries”. en. In: *Applied Cryptography and Network Security Workshops*. ISSN: 1611-3349. Springer, Cham, 2024, pp. 283–302. ISBN: 978-3-031-61486-6. DOI: 10.1007/978-3-031-61486-6_17. URL: https://link.springer.com/chapter/10.1007/978-3-031-61486-6_17 (visited on 03/22/2025).

[113] A. Stegmeier, P. Knauer, P. Schubaur, C. Piatka, D. Merli, and S. Altmeyer. “Safe and Secure? On the Timing Analysability of Cryptographic Implementations”. In: *2024 IEEE 30th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. ISSN: 2642-7346. May 2024, pp. 68–80. DOI: 10.1109/RTAS61025.2024.00014. URL: https://ieeexplore.ieee.org/abstract/document/10568064 (visited on 03/22/2025).

[114] A. Kanaoka and M. Abe. “Enhanced Analysis of Cryptographic Library Usage Patterns and Trends in Android Applications”. In: *2024 IEEE Conference on Dependable and Secure Computing (DSC)*. Nov. 2024, pp. 88–93. DOI: 10.1109/DSC63325.2024.00031. URL: https://ieeexplore.ieee.org/abstract/document/10763698 (visited on 03/22/2025).

[115] M. J. Kannwischer, P. Schwabe, D. Stebila, and T. Wiggers. “Improving Software Quality in Cryptography Standardization Projects”. In: *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*. Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 19–30. DOI: 10.1109/EuroSPW55150.2022.00010. URL: https://eprint.iacr.org/2022/337.

[116] V. Gayoso Martínez, L. Hernández Encinas, and C. Sánchez Ávila. “Elliptic curve cryptography. Java platform implementations”. eng. In: (2009). Accepted: 2010-02-17T12:40:38Z. ISSN: 1313-8251. URL: https://digital.csic.es/handle/10261/21232 (visited on 02/19/2025).

[117] A. Abusharekh and K. Gaj. “Comparative analysis of software libraries for public key cryptography”. In: *Software Performance Enhancement for Encryption and Decryption, SPEED* (2007), pp. 11–12.

[118] D. F. Aranha, C. P. L. Gouvêa, T. Markmann, R. S. Wahby, and K. Liao. *RELIC is an Efficient LIbrary for Cryptography*. 2007. URL: https://github.com/relic-toolkit/relic.

[119] D. J. Bernstein, T. Lange, and P. Schwabe. “The Security Impact of a New Cryptographic Library”. en. In: *Progress in Cryptology – LATINCRYPT 2012*. Ed. by A. Hevia and G. Neven. Berlin, Heidelberg: Springer, 2012, pp. 159–176. ISBN: 978-3-642-33481-8. DOI: 10.1007/978-3-642-33481-8_9.

[120] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin. “Charm: a framework for rapidly prototyping cryptosystems”. In: *Journal of Cryptographic Engineering* 3 (2013). Publisher: Springer, pp. 111–128.

[121] J.-K. Zinzindohoué, K. Bhargavan, J. Protzenko, and B. Beurdouche. “HACL*: A Verified Modern Cryptographic Library”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’17. New York, NY, USA: Association for Computing Machinery, Oct. 2017, pp. 1789–1806. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3134043. URL: https://dl.acm.org/doi/10.1145/3133956.3134043 (visited on 02/11/2025).

[122] M. J. Kannwischer, R. Petri, J. Rijneveld, P. Schwabe, and K. Stoffelen. *PQM4: Post-quantum crypto library for the ARM Cortex-M4*.

[123] D. Stebila and M. Mosca. “Post-quantum Key Exchange for the Internet and the Open Quantum Safe Project”. en. In: *Selected Areas in Cryptography – SAC 2016*. Ed. by R. Avanzi and H. Heys. Cham: Springer International Publishing, 2017, pp. 14–37. ISBN: 978-3-319-69453-5. DOI: 10.1007/978-3-319-69453-5_2.

[124] B. Grégoire. “Provable Security of Cryptographic Primitives: from Algorithms to Assembly”. en. thesis. Université Côte d’Azur, Dec. 2024. URL: https://hal.science/tel-04911253 (visited on 04/15/2025).

[125] F. Giacon, E. Kiltz, and B. Poettering. “Hybrid Encryption in a Multi-user Setting, Revisited”. en. In: *Public-Key Cryptography – PKC 2018*. Ed. by M. Abdalla and R. Dahab. Cham: Springer International Publishing, 2018, pp. 159–189. ISBN: 978-3-319-76578-5. DOI: 10.1007/978-3-319-76578-5_6.

[126] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila. *A Formal Security Analysis of the Signal Messaging Protocol*. Publication info: Published elsewhere. Major revision. IEEE EuroS&P 2017. 2016. URL: https://eprint.iacr.org/2016/1013 (visited on 09/19/2023).

Printed by Media-Tryck, Lund 2025 NORDIC SWAN ECOLABEL 3041 0903

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2025:04
ISSN: 1650-2140
ISBN: 978-91-7295-498-4