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

Alexandr Silonosov and Lawrence Henesey

Blekinge Institute of Technology, Karlskrona, Sweden

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.

Keywords: AIS ship tracking data, Key encapsulation mechanism, end-to-end encryption, cryptographic agility, CRYSTALS-Kyber,Post-Quantum Cryptography.

1 Introduction

The Automatic Identification System (AIS) is a https://www.navcen.uscg.gov/automatic-identification-system-overviewvessel navigation safety communications system; its main aim is to improve maritime traffic awareness and safety [6]. 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 [13], [29]. To enhance the security of the AIS [1], [33] 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. [8], [9] 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 [15], [23] include a technique using the TESLA authentication protocol (RFC-4082 standard) to ensure AIS

David C. Wyld et al. (Eds): CCSIT, NLPCL, AISC, ITE, NCWMC, DaKM, BIGML, SIPP, SOEN, PDCTA – 2024
pp. 193-212, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141714

transceivers authentication and message integrity in transit only. Commercial AIS products https://www.saab.com/products/r6-secure-aisSAAB 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 https://www.imo.org/es/OurWork/Safety/Paginas/AIS.aspxInternational 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 [3] 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 [16] 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 [26], [25] 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 [4], 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.

2 Related work

According to surveys [20] and [34], 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

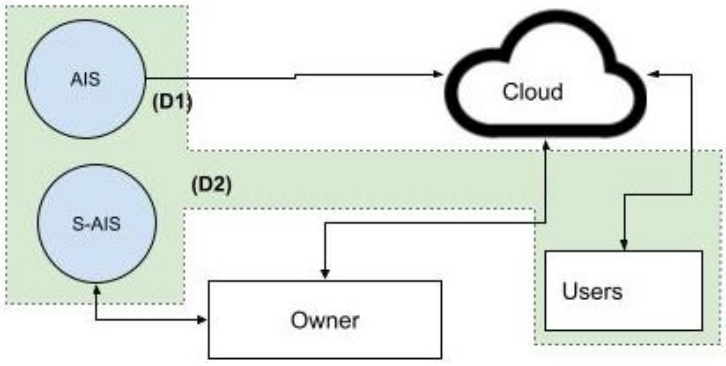

Fig. 1. Logical confidentiality perimeter (green) between data sources D1, D2, and Users in proposed E2EE architecture.

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[10] (2006) and Ciphertext-Policy ABE [32] (2008) are successors of Fuzzy Identity Based Encryption (IBE) [21], 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 [26]. Similar functionality implemented in traditional PKE known as Key Encapsulation Mechanism (KEM) to multiple parties proposed by [28] in 2005.

A Survey of ABE software libraries by Mosteiro-Sanchez et al. [18] 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 [32], [10] 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. [17] 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. [12] 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, [31], [19] 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. [36] , [37] 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. [35] 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 [30]. 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 [11], [7]. Gao et al. [7] 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. [14]

Ref	CP	CC	TM	DF	AM	SP	KD
[36]	+	F					
[37]	F	F					
[35]	+	+					
[7]	+		+	+	+	+	
[11]	+			+			
[14]	+	+	+	+		+	+

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. [18]. 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.

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 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

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

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

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

code library and data structure. Key Encapsulation approaches to multiple Parties also exist in conventional cryptosystems [28] 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 [22] .

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) [5] 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

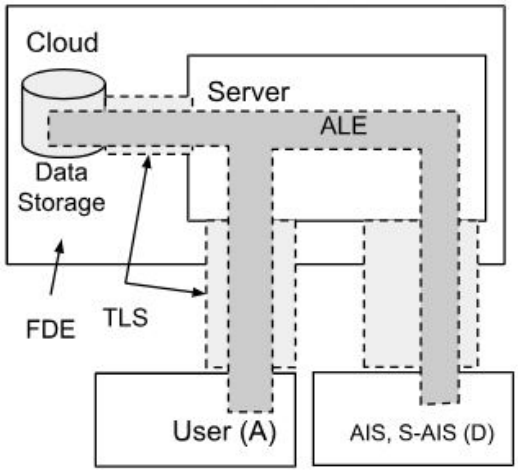

Fig. 2. Data encryption flow (gray color) in E2EE architecture.

challenge, the software variability issues [27] 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 [28] and comparing it with the ABE scheme is feasible to understand cryptographic migration capacity [25], [24]. As a second independent variable for experiment we choose CRYSTAL-Kyber [2] 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] https://csrc.nist.gov/pubs/fips/203/ipd

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 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.

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 .

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

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 [28] and thus can be reviewed as an alternative to ABE. Table 4 represents the real-world ship tracking database we used during the experiment.

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

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

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 [6], 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 4 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 4, 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 4 and AIS specification [9], 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 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 6 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 5 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.

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,

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

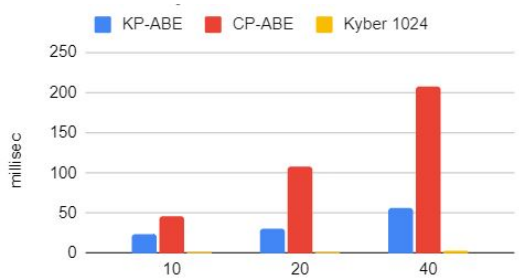

Fig. 3. ABE key encapsulation performance in ms. on vCPU with R=1000 single thread performance rating, for 10, 20, 40 access attributes.

.

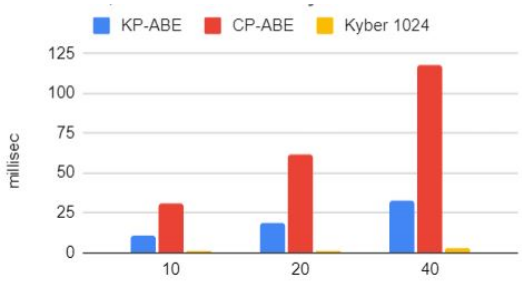

Fig. 4. ABE key encapsulation performance in ms. on vCPU with R=2000 single thread performance rating, for 10, 20, 40 access attributes.

.

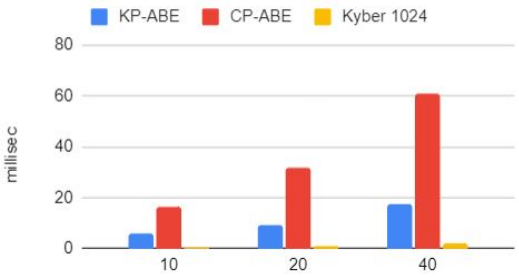

Fig. 5. ABE key encapsulation performance in ms. on vCPU with R=4000 single thread performance rating, for 10, 20, 40 access attributes.

.

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

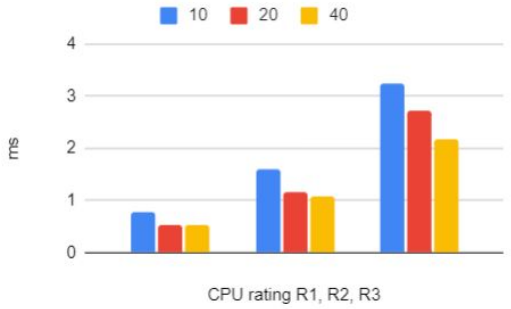

Fig. 6. Kyber-1024 multi-recipient key encapsulation performance in ms. on vCPU with R1=1000, R2=2000, R3=4000 single thread performance rating, for 10, 20, 40 public keys.

.

according to Table 5, 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 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.

6.1 Author contributions

The overall conceptualization, methodology, software, writing, result orchestration, and validation were carried out by Silonosov. Review, editing, and funding acquisition for the EU project were carried out by Henesey. Native language proofing was performed by Henesey. All authors have read and agreed to the published version of the manuscript.

References

1. Ahmed Aziz, Pietro Tedeschi, Savio Sciancalepore, and Roberto Pietro. *SecureAIS - Securing Pairwise Vessels Communications*. June 2020.
2. Joppe Bos, Leo Ducas, Eike Kiltz, T Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehle. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367, London, April 2018. IEEE.
3. William R. Cairns. AIS and Long Range Identification & Tracking. *The Journal of Navigation*, 58(2):181–189, May 2005. Publisher: Cambridge University Press.
4. Remi Challamel, Thibaud Calmettes, and Charlotte Neyret 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)*, pages 246–252, September 2012. ISSN: 2326-5949.
5. Yun Ding and Karsten Klein. Model-Driven Application-Level Encryption for the Privacy of E-health Data. In *2010 International Conference on Availability, Reliability and Security*, pages 341–346, February 2010.
6. Ties Emmens, Chintan Amrit, Asad Abdi, and Mayukh Ghosh. The promises and perils of Automatic Identification System data. *Expert Systems with Applications*, 178:114975, September 2021.
7. Qi Gao, Junwei Zhang, Jianfeng Ma, Chao Yang, Jingjing Guo, and Yinbin Miao. LIP-PA: A Logistics Information Privacy Protection Scheme with Position and Attribute-Based Access Control on Mobile Devices. *Wireless Communications and Mobile Computing*, 2018:e9436120, July 2018. Publisher: Hindawi.

8. Athanasios Goudosis and Sokratis Katsikas. Secure AIS with Identity-Based Authentication and Encryption. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, 14(2):287–298, 2020.
9. Athanasios Goudosis and Sokratis Katsikas. Secure Automatic Identification System (SecAIS): Proof-of-Concept Implementation. *Journal of Marine Science and Engineering*, 10(6):805, June 2022. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.
10. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data, 2006. Publication info: Published elsewhere. Extended abstract to appear in ACM CCS 2006. This is the full version.
11. Sheng Hong, Haowen Pan, Yijia Fang, Jie Ma, Xiaojing Qi, and Yanghong 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)*, pages 218–224, July 2022.
12. Raza Imam, Kaushal Kumar, Syed Mehran Raza, Rumi Sadaf, Faisal Anwer, Noor Fatima, Mohammad Nadeem, Mohamed Abbas, and Obaidur Rahman. A systematic literature review of attribute based encryption in health services. *Journal of King Saud University - Computer and Information Sciences*, 34(9):6743–6774, October 2022.
13. Silvie Levy, Ehud Gudes, and Danny Hendler. A Survey of Security Challenges in Automatic Identification System (AIS) Protocol. In Shlomi Dolev, Ehud Gudes, and Pascal Paillier, editors, *Cyber Security, Cryptology, and Machine Learning*, pages 411–423, Cham, 2023. Springer Nature Switzerland.
14. Tao Li, Rui Zhang, and Yanchao Zhang. PriExpress: Privacy-preserving express delivery with fine-grained attribute-based access control. In *2016 IEEE Conference on Communications and Network Security (CNS)*, pages 333–341, October 2016.
15. Robert E. Litts, Dimitrie C. Popescu, and Otilia Popescu. Authentication Protocol for Enhanced Security of the Automatic Identification System. In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–6, May 2021.
16. G. Longo, M. Martelli, E. Russo, A. Merlo, and R. Zaccone. Adversarial waypoint injection attacks on Maritime Autonomous Surface Ships (MASS) collision avoidance systems. *Journal of Marine Engineering & Technology*, 0(0):1–12, 2023. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/20464177.2023.2298521.
17. Siddhesh Mhatre, Anant V. Nimkar, and Sudhir N. Dhage. Comparative study on attribute-based encryption for health records in cloud storage. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 647–652, May 2017. Conference Name: 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) ISBN: 9781509037049 Place: Bangalore Publisher: IEEE.
18. Aintzane Mosteiro-Sanchez, Marc Barceló, Jasone Astorga, and Aitor Urbieta. *All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of CP-ABE Libraries*. September 2022.
19. Shardha Porwal and Sangeeta Mittal. Implementation of Ciphertext Policy-Attribute Based Encryption (CP-ABE) for fine grained access control of university data. In *2017 Tenth International Conference on Contemporary Computing (IC3)*, pages 1–7, Noida, August 2017. IEEE.
20. Marco Rasori, Michele La Manna, Pericle Perazzo, and Gianluca Dini. A Survey on Attribute-Based Encryption Schemes Suitable for the Internet of Things. *IEEE Internet of Things Journal*, 9(11):8269–8290, June 2022. Conference Name: IEEE Internet of Things Journal.
21. Amit Sahai and Brent Waters. Fuzzy Identity Based Encryption, 2004. Publication info: Published elsewhere. Unknown where it was published.
22. Sarah Scheffler and Jonathan Mayer. SoK: Content Moderation for End-to-End Encryption. *Proceedings on Privacy Enhancing Technologies*, 2023(2):403–429, April 2023.

23. Savio Sciancalepore, Pietro Tedeschi, Ahmed Aziz, and Roberto Pietro. Auth-AIS: Secure, Flexible, and Backward-Compatible Authentication of Vessels AIS Broadcasts. *IEEE Transactions on Dependable and Secure Computing*, PP:1–1, March 2021.
24. Dimitrios Sikeridis, David Ott, Sean Huntley, Shivali Sharma, Vasantha Kumar Dhanasekar, Megha Bansal, Akhilesh Kumar, Anwitha U. N, Daniel Beveridge, and Sairam Veeraswamy. ELCA: Introducing Enterprise-level Cryptographic Agility for a Post-Quantum Era, 2023. Publication info: Preprint.
25. Alexandr Silonosov, Oleksii Baranovskyi, and Lawrence 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, page 144, New York, NY, USA, February 2024. Association for Computing Machinery.
26. Alexandr Silonosov and Lawrence 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, pages 1–8, New York, NY, USA, July 2024. Association for Computing Machinery.
27. M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch. COVAMOF: A Framework for Modeling Variability in Software Product Families. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3154:197–213, 2004. ISBN: 9783540229186.
28. N. P. Smart. Efficient Key Encapsulation to Multiple Parties. In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks*, Lecture Notes in Computer Science, pages 208–219, Berlin, Heidelberg, 2005. Springer.
29. Omer Soner, Gizem Kayisoglu, Pelin Bolat, and Kimberly Tam. Risk sensitivity analysis of AIS cyber security through maritime cyber regulatory frameworks. *Applied Ocean Research*, 142:103855, January 2024.
30. Asha. V, A. P. Nirmala, Bhavani. K, Aldred Christi, and Naveen. 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)*, pages 100–104, March 2022.
31. Changji Wang, Xuan Liu, and Wentao Li. *Implementing a Personal Health Record Cloud Platform Using Ciphertext-Policy Attribute-Based Encryption*. September 2012. 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.
32. Brent Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, 2008. Publication info: Published elsewhere. Unknown where it was published.
33. Gareth Wimpenny, Jan Šafář, Alan Grant, and Martin Bransby. Securing the Automatic Identification System (AIS): Using public key cryptography to prevent spoofing whilst retaining backwards compatibility. *The Journal of Navigation*, 75(2):333–345, March 2022. Publisher: Cambridge University Press.
34. Yinghui Zhang, Robert H. Deng, Shengmin Xu, Jianfei Sun, Qi Li, and Dong Zheng. Attribute-based Encryption for Cloud Computing Access Control: A Survey. *ACM Computing Surveys*, 53(4):83:1–83:41, August 2020.
35. Jun Zhao, Kai Zhang, Junqing Gong, and Haifeng Qian. Lavida: Large-Universe, Verifiable, and Dynamic Fine-Grained Access Control for E-Health Cloud. *IEEE Transactions on Information Forensics and Security*, 19:2732–2745, 2024. Conference Name: IEEE Transactions on Information Forensics and Security.
36. Xiaoping Zhao and Qianqian Su. Revocable Attribute-Base Scheme with Enhanced Security and Privacy for Healthcare Data Sharing. In *2023 IEEE 14th International Symposium*

on Parallel Architectures, Algorithms and Programming (PAAP), pages 1–8, Beijing, China, November 2023. IEEE.
37. Yang Zhao, Pengcheng Fan, H. Cai, Zhiguang Qin, and H. Xiong. Attribute-based Encryption with Non-Monotonic Access Structures Supporting Fine-Grained Attribute Revocation in M-healthcare. *Int. J. Netw. Secur.*, 2017.

Authors

Alexandr Silonosov received the M.Sc. degree in Computer Science from Blekinge Institute of Technology (BTH), Sweden in 2020. Currently, he is pursuing his PhD in Computer Science from the University of BTH. His research interests include Information security, Cryptographic agility in end-to-end protocols.

Dr. Lawrence Henesey received PhD in Information technology from Blekinge Institute of Technology, (BTH) Sweden. Since 2011 he is a senior researcher at BTH and focuses his work on the applications of technologies in Distributed Artificial Intelligence (Multi-Agents and ML) for improved performances in logistics systems, such as Container Ports and Terminals, which has culminated into 100+ published articles and two books.