PRIVACY-DRIVEN E2E SECURE DATA EXCHANGE FOR VALUE CHAINS

AUTHOR:
Aintzane Mosteiro Sánchez
2022

ADVISORS:
Marc Barcelo Llado
Jasone Astorga Burgo

Universidad del País Vasco
Euskal Herriko Unibertsitatea

UNIVERSITY OF THE BASQUE COUNTRY

UPV/EHU

THESIS DISSERTATION

Privacy-Driven E2E Secure Data Exchange for Value Chains

Author:
Aintzane Mosteiro Sanchez

Advisors:
Marc Barcelo Llado
Jasone Astorga Burgo

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy
in the

Deparment of Communications Engineering
January 15, 2023

(cc)2023 AINTZANE MOSTEIRO SANCHEZ (cc by-nc-sa 4.0)

Copyright © 2022 Aintzane Mosteiro Sanchez

Typeset using LaTeX

Cover 2022 Pablo Lopez Lopez

This work is licensed under a Creative Commons "Attribution-NonCommercial-ShareAlike 4.0 International" license.

The whole of life is just like watching a film. Only it's as though you always get in ten minutes after the big picture has started, and no-one will tell you the plot, so you have to work it out all yourself from the clues.

Terry Pratchett, Moving Pictures

ACKNOWLEDGEMENT

> Face your life, its pain, its pleasure,
> leave no path untaken.
>
> *Neil Gaiman, The Graveyard Book*

Qué razón tenía Neil Gaiman cuando comentaba eso de que hay que afrontar la vida como es: con sus placeres y su dolor. Y es que una tesis es un poco así: cosas maravillosas y cosas desesperantes. Momentos increíbles y momentos increíblemente duros. En sí, creo que no vi venir ni el 90% de las cosas que me han ocurrido desde aquel octubre de 2019 cuando a la pregunta de *"¿Oye, a ti te interesa la ciberseguridad?"* dije *"¡Claro, por qué no!"*. (Esto cuenta como grandes citas que predicen un desastre).

Desde luego, cuando comencé este camino de la tesis doctoral, no me esperaba la magnitud de todo. La cantidad de esfuerzo que hace falta (lo asumes, pero realmente no eres consciente de ello), la cantidad de ayuda y apoyo que hace falta, la fortaleza mental (que más de una vez falla), y la capacidad de saber desconectar. En cierto modo, una tesis es como mirar a ambos lados antes de cruzar la carretera y que te atropelle un avión. Por suerte, he tenido una buena cantidad de gente a mi lado que han podido sostenerme para que no me derribara el golpe.

Antes de nada, todo mi agradecimiento a mis padres, Arantza Sánchez y Norberto Mosteiro. Llevan 27 años aguantándome, y tras una carrera, un master, una tesis y una pandemia, aún no me han echado de casa. Reconozco no tener ni un 1% de la paciencia que han tenido ellos conmigo todos estos años. Su cariño y apoyo incondicionales han sido claves para no tirar la toalla en ningún momento.

También quiero dar mis agradecimientos a mis tutores de tesis, Jasone Astorga y Marc Barceló. Ellos también han mostrado mucha más paciencia conmigo de la que a veces he merecido. Pero gracias a ellos soy mejor investigadora: me han enseñado la necesidad de saber debatir sin discutir. No existe ciencia sin debate, al fin y al acabo.

A mis compañeros de tesis en Ikerlan, ¡qué puedo decir! En realidad, podría empezar a escribir y no parar nunca. Todos mis agradecimientos a Odei Olalde, Maialen Eceiza, Ángel Longueira, Unai Rioja, Servio Paguada, Gorka Abad, Xabi Pérez y Denis Stefanescu. Su apoyo, ánimos, ideas y cariño han sido imprescindibles para llegar a

donde estoy. Por esas pausas de café que salvan vidas y esos abrazos que curan todos los problemas. Gracias a ellos he descubierto nueva música, aficiones o lugares a los que viajar. Gracias, gracias, gracias. Todos vosotros me habéis hecho mejor investigadora, pero también mejor persona.

También quiero dedicarle unas palabras especiales a Estela Nieto. No han sido los tres años más fáciles de nuestras vidas, es más, han sido los peores y los mejores a la vez. Siempre estaré agradecida de que hayamos podido mantener la amistad desde que nos conocimos en aquel laboratorio de master hace cinco años. (No, no te acuerdas. Pero no importa, ya lo recuerdo yo por ti). Por más viajes, sesiones de videojuegos y fines de semana desayunando tortitas. No sé qué nos depara 2023, pero estoy deseando descubrirlo.

Por supuesto no podía terminar esto sin mencionar a mis amigos de Barakaldo. A Pablo López, mi infinito cariño. No solo me ha hecho una portada preciosa, sino que además (como tantísimas otras personas de mi vida) muestra una paciencia infinita conmigo, y su capacidad de reflexión me enseña cada día a ser mejor persona. A Amaia Vizuete, la vida sería un poco peor sin ti a mi lado: desde luego habría muchas menos risas y muchos menos musicales. A Itxaso Martín, Ander García, Enaitz Angoitia, Laura Gómez, Laura López y Alejandro Barranco, por esos pintxopotes, partidas de pádel y esas tardes de juegos de mesa. Aunque ya os sepáis el Trivial de memoria, espero poder echar muchas más partidas juntos. A Laura Gómez, Lucía Conde, Jimena Cabrejas, María Monzón y Araiz López. Por más viajes juntas (aunque siempre acaben siendo a sitios donde María ya ha ido), más imitaciones de estatuas, más musicales y más tortillas de patatas (de María, que es la que siempre acaba dándonos de comer). Esos viajes siempre acaban siendo de los más divertidos del año.

Finalmente, unas palabras especiales para Laura Gómez y María Monzón: yo ya he terminado, ahora quedáis vosotros. Pero ahora que estoy al final del túnel, puedo confirmar, ¡que hay luz! Vais a pasar por mil altibajos, os vais a desesperar y vais a llorar. Pero también vais a descubrir cosas nuevas sobre vosotras mismas que no habríais descubierto de otra manera. Una tesis es, al fin y al cabo, una aventura de la que saldréis totalmente exitosas. No tengo ni una sola duda al respecto. Cuidaros y dejad que os apoyen, y todo irá bien.

De verdad, mil gracias a todos. La vida es una aventura maravillosa, y estoy encantada de haber vivido una parte de ella a vuestro lado. Sin importar a donde nos lleve la vida, estos años siempre los guardaré en el corazón.

Aintzane Mosteiro Sánchez

Barakaldo, 2022

ABSTRACT

Digitalization is an unstoppable phenomenon affecting all aspects of life and society, including the industrial sector. Despite the typical resistance to change common in these environments, the increasing presence of distributed automated systems or wireless communications is changing the paradigm. One of the technologies driving these changes are the Internet of Things (IoT) devices, referred to in the industrial case as Industrial IoT (IIoT). These devices, together with other technologies such as artificial intelligence, cloud computing, or Big Data, have driven industrial transformation and have given rise to Industry 4.0. This industrial transformation has brought great benefits to the industrial environment and new risks and vulnerabilities. The presence of IIoT devices makes the presence of industrial cybersecurity solutions imperative, given that industrial cyberattacks are becoming more common and have worse consequences for the companies that suffer them.

From all of the above, it is crucial to have a system that can guarantee the security of all information transmitted in Industry 4.0. At plant level, many security solutions are regulated by institutions such as IEC, ENISA, or ICS-CERT. However, the regulation focuses on the security to be deployed within each industrial plant, while protecting information exchange between plants is not so regulated or defined. Thus, since the flow of information between partners in a value chain is vital for the industrial ecosystem, ensuring its security is also vital for developing Industry 4.0. Therefore, the goal of this thesis is to design an information exchange system that covers the entire value chain and can guarantee End-to-End (E2E) data confidentiality and integrity.

To achieve the objective, this thesis considers that *i)* the company sending or receiving information meets minimum security requirements and that *ii)* the exchange of information is secure. To this end, a minimum security architecture must be guaranteed, and how the information exchange is carried out must be defined. Currently, there is no widely used solution for the latter; instead, proprietary and ad-hoc solutions are used. Therefore, if a company is involved in several value chains, it has to manage a many different information exchange systems. In order to solve the identified problem, this thesis presents three proposals, as well as the state of the art, conclusions, and future lines of research.

The first proposal is related to premise *i)*, which considers that any plant connected to a value chain must have a minimum security architecture. For this purpose, various cybersecurity standards from ETSI, ENISA, or ICS-CERT are studied and used to define the objectives, layers, and security measures of a DiD-based security strategy. In addition, it is concluded that a plant-level security architecture is insufficient to ensure secure information exchange, so the requirements to be met by the proposed secure information exchange solution are also defined.

The second and third proposals are closely related and respond to premise *ii)*. They propose a secure information exchange system using application-layer encryption, specifically using CP-ABE. This type of encryption allows one-to-many encryption, identifying users through attributes and protecting information according to access policies. The system meets the industrial requirements defined by ETSI and can update the access policies used for encryption. This results in a secure E2E information exchange system that is flexible in the face of organizational changes and meets industrial requirements. In addition, a detailed study of various libraries that implement CP-ABE has also been carried out to choose the appropriate one in a value chain.

Regarding the third proposal, it is an attribute spoofing prevention system. The security of a CP-ABE system is based on users obtaining private keys that reflect the attributes they hold. However, how the key generating authorities obtain such information about users is not widely referenced in the literature. Therefore, the thesis identifies attribute spoofing as a potential vulnerability of CP-ABE-based systems. We establish certain assumptions about the system's operation, identify two attack vectors, and propose a system capable of preventing them through the combination of a DAG and a distributed storage system (IPFS).

Finally, to validate the different proposals quantitatively, their ability to meet the requirements established at different points of the thesis has been studied. In the case of the first proposal, the thesis verifies compliance with the DiD goals identified in different standards. In the case of the second proposal, compliance with the industrial requirements for ABE schemes defined by the ETSI is verified. Finally, the third proposal is evaluated regarding its ability to manage the studied attack vectors. Concerning the quantitative study, several experiments have been performed using an RPI0 to simulate

an IIoT device and an RPI4 to simulate an industrial device with higher capabilities.

LABURPENA

Digitalizazioa fenomeno geldiezina da, bizitzaren eta gizartearen alderdi askori eragiteaz gain, industria-sektoreari ere eragiten die. Ingurune horietan aldaketa komunarekiko erresistentzia ohikoa den arren, sistema automatizatu banatuen presentzia edo hari gabeko komunikazioen ugaritzeak paradigma aldatzen du. IoT gailuek aldaketa horiek bultzatzen dituzte, kasu industrialean *Industrial IoT* (IIoT) deituak. Gailu horiek, adimen artifiziala, *Cloud* konputazioa edo *Big Data* bezalako beste teknologia batzuekin batera, eraldaketa industriala bultzatu eta 4.0 industria sortu dute. Industria-eraldaketa horrek onura handiak ekarri dizkio industria-inguruneari, baina baita arrisku eta kalteberatasun berriak ere. IIoT gailuen presentzia ezinbestekoa da zibersegurtasun industrialeko soluzioetan, zibereraso industrialak gero eta ohikoagoak baitira eta hauek jasaten dituzten konpainientzat ondorioak larriak izaten dira.

Honekin guztiarekin, ondorioztatzen da funtsezkoa dela 4.0 Industrian transmititutako informazio ororen segurtasuna bermatzeko sistema bat izatea. Plantari dagokionez, segurtasun-irtenbide ugari daude, IEC, ENISA edo ICS-CERT erakundeek arautuak. Hala ere, erregulazioa industria-planta bakoitzaren barruan hedatu beharreko segurtasunean oinarritzen da, eta planten arteko informazio-trukearen babesa, berriz, ez dago hain araututa edo definituta. Beraz, balio-kate baten *partners*-en informazio fluxua funtsezkoa da ekosistema industrialerako, haren segurtasuna bermatzea ere ezinbestekoa da Industria 4.0 garatzeko. Beraz, tesi honen helburua informazioa trukatzeko sistema bat diseinatzea da, balio-kate osoa kontuan hartuko duena eta muturreko datuen konfidentzialtasuna eta osotasuna bermatzeko gai izango dena.

Helburua lortzeko, tesi honek bi helburu nagusi bete behar ditu, *i)* informazioa bidaltzen edo jasotzen duen konpainiak gutxieneko segurtasun-baldintza batzuk betetzea eta *ii)* informazio-trukea segurua izatea. Horretarako, gutxieneko segurtasun arkitektura bermatu behar da, eta informazio-trukea nola gauzatzen den zehaztu. Gaur egun, azken horrek ez du asko erabiltzen den soluziorik; aitzitik, soluzio jabeak eta *ad-hoc* soluzioak erabiltzen dira. Beraz, enpresa batek hainbat balio-katetan parte hartzen badu, informazioa trukatzeko sistema desberdinak kudeatu beharko ditu. Horren ordez, hobe da konponbide unibertsal bat izatea, horrela balio-kateko informazioaren kudeaketaren konplexutasuna murrizteko. Identifikatutako arazoa konpondu ahal izateko, tesi hau

garatu da. Tesia zazpi kapitulutan banatuta dago, bertan hiru proposamenak aurkezten dira, baita artearen egoera, ondorioak edo etorkizuneko ikerketa-ildoak ere.

Lehenengo proposamena *i)* helburuarekin lotuta dago, balio-kate bati konektatuta dagoen planta orok gutxieneko segurtasun-arkitektura izan behar duela uste baitu. Horretarako, ETSI, ENISA edo ICS-CERTen hainbat zibersegurtasun-estandar aztertzen dira, eta DiDn oinarritutako segurtasun-estrategiaren helburuak, geruzak eta segurtasun-neurriak definitzeko erabiltzen dira. Horretaz gain, segurtasun-arkitektura hori berez ez dela nahikoa informazio-truke segurua bermatzeko ondorioztatzen da, eta, beraz, informazio segurua trukatzeko irtenbideak bete beharreko baldintzak ere zehazten dira.

Bigarren eta hirugarren proposamenak estuki lotuta daude, eta *ii)* helburuari erantzuten diote. Horretarako, sekurizatutako informazioa trukatzeko sistema bat proposatzen da, aplikazio-geruzan zifratuta, zehazki CP-ABE erabiliz. Zifratze mota horrek aukera ematen du bat edo batzuk zifratzeko, erabiltzaileak atributuen bidez identifikatuz eta sarbide-politikekin bat datorren informazioa babestuz. Sistemak ETSIek definitutako industria-baldintzak betetzen ditu, eta zifratzeko erabilitako sarbide politikak eguneratzeko aukera ematen du. Horrela, informazioa trukatzeko E2E sistema segurua lortzen da, antolaketa-aldaketen aurrean malgua dena eta industria-baldintzak betetzen dituena. Horrez gain, esperimentuak ingurune industrialetara egokitutako mendekotasunekin egiteko, CP-ABE inplementatzen duten hainbat liburutegiren azterketa zehatza ere egin da, balio-kate batean egokia dena aukeratzeko.

Hirugarren proposamenari dagokionez, atributuak faltsutzeko prebentzio-sistema bat da. CP-ABE sistema baten segurtasuna erabiltzaileek dituzten ezaugarriak islatzen dituzten gako pribatuak lortzean oinarritzen da. Hala ere, literaturan ez da aipatzen gakoak sortzen dituzten agintariek erabiltzaileei buruzko informazio hori nola lortzen duten. Horregatik, atributuen faltsifikazioa CP-ABE.n oinarritutako sistemen ahultasun potentzial gisa identifikatzen dugu. Sistemaren funtzionamenduari buruzko suposizio batzuk ezartzen dira, bi eraso-bektore identifikatzen dira, eta DAG baten eta *Interplanetary File System (IPFS)* konbinazioaren bidez saihesteko gai den sistema bat proposatzen da.

Azkenik, proposamenak kuantitatiboki baliozkotzeko, memoriaren hainbat puntutan ezarritako baldintzak betetzeko gaitasuna aztertu da. Lehenengo proposamenaren

kasuan, estandar desberdinen baldintzak betetzea bilatu da. Bigarrenaren kasuan, ETSIen industria-baldintzak bete dira, 1. proposamena garatu bitartean zehaztu zirenak ere. Azkenik, hirugarren proposamena aztertutako eraso-bektoreei aurre egiteko gaitasunaren arabera ebaluatu da. Azterketa kuantitatiboari dagokionez, hainbat esperimentu egin dira IIoT gailu bat simulatzeko RPI0 bat eta gaitasun handiagoko gailu industrial bat simulatzeko RPI4 bat erabiliz.

RESUMEN EJECUTIVO

La digitalización es un fenómeno imparable, que afecta a todos los aspectos de la vida y la sociedad, incluyendo al sector industrial. A pesar a la típica resistencia al cambio común en estos entornos, el aumento de la presencia de sistemas automatizados distribuidos o las comunicaciones inalámbricas está cambiando el paradigma. Una de las tecnologías que actúan como motor e impulsan estos cambios son los dispositivos *Internet of Things*, o IoT. Este tipo de dispositivos, capaces de procesar, enviar y recibir información están presentes en todos los ámbitos y facetas de la tecnología actual, incluido en el ámbito industrial. De hecho, a los dispositivos IoT específicamente pensados para uso industrial se les ha denominado dispositivos *Industrial IoT* (IIoT). Estos dispositivos, junto a otras tecnologías como la inteligencia artificial, la computación *Cloud* o el *Big Data*, han impulsado la transformación industrial y han dado lugar a la Industria 4.0.

Esta transformación industrial ha traído grandes beneficios al entorno industrial, pero también nuevos riesgos y vulnerabilidades. La presencia de dispositivos IIoT vuelve imperativa la presencia de soluciones de ciberseguridad industrial. Sin embargo, a pesar de que la industria está comenzando a tomar medidas en este sentido, los ciberataques continúan, con cada vez peores consecuencias para las compañías que los sufren. Como la Industria 4.0 ha causado un aumento en la cantidad de información que se transmite tanto dentro de una misma planta industrial como entre plantas, los ataques cada vez afectan a más usuarios y son más caros de subsanar.

Por todo lo anterior, se deduce que es crucial contar con un sistema que pueda garantizar la seguridad de toda información transmitida en la Industria 4.0. A nivel de planta, existen multitud de soluciones de seguridad, reguladas por instituciones como el IEC, la ENISA o el ICS-CERT. Sin embargo, la regulación se centra en la seguridad a desplegar dentro de cada planta industrial, mientras que la protección del intercambio de información entre plantas no está tan regulado o definido. Asi pues, dado que el flujo de información entre *partners* de una cadena de valor es vital para ecosistema industrial, garantizar su seguridad también es vital para el desarrollo de la Industria 4.0. Por tanto, se puede definir que el objetivo de esta tesis es el diseño de un sistema de intercambio de información, que abarque toda la cadena de valor, y sea capaz de

garantizar la confidencialidad e integridad de los datos extremo a extremo.

Para lograr el objetivo, se considera que es necesario que se cumplan dos máximas, que *i)* la compañía que envía o recibe información cumpla unos requisitos de seguridad mínimos y que *ii)* el intercambio de información sea seguro. Para ello, hay que garantizar una arquitectura de seguridad mínima, y definir cómo se lleva a cabo el intercambio de información. Actualmente esto último no cuenta con una solución extensamente utilizada, sino que se utilizan soluciones propietarias y *ad-hoc*. Por tanto, si una empresa participa de varias cadenas de valor, tiene que gestionar una gran variedad de sistemas de intercambio de información diferentes. En su lugar, es mejor contar con una solución universal, para así reducir la complejidad de la gestión de la información en la cadena de valor. Para poder solucionar la problemática identificada, esta tesis se divide en 6 capítulos, cuyo contenido es el siguiente:

El Capítulo 1 introduce el contexto de la tesis y su motivación. También define las preguntas de investigación que vertebrarán los siguientes capítulos y ofrece un resumen de la principal contribución de cada uno con sus publicaciones asociadas.

El Capítulo 2 presenta el *background* necesario para seguir la tesis, así como el estado del arte general. Primero, resume el *background* principal de la industria 4.0. Para ello identifica los requisitos de seguridad exigidos por el NIST y el IEC, define las diferencias entre las redes de la Tecnología Operacional (OT) y la de Tecnología de la Información (IT), y presenta el modelo de referencia a alto nivel que propone la ENISA para la fabricación inteligente. A continuación, se resumen los requisitos esenciales que pide el IEC 62443, el principal estándar de ciberseguridad en industria. Una vez estudiadas las medidas de seguridad requeridas por los estándares dentro de una misma planta industrial, se estudian las propuestas existentes para securizar el intercambio de información entre miembros de una misma cadena de valor.

Tras la revisión de los aspectos más industriales de la tesis, se comienza con el estudio relacionado con el ámbito puramente de ciberseguridad. Concretamente, se centra en las diferentes propuestas para lograr confidencialidad e integridad E2E en la transmisión de datos. Se ofrece un breve resumen de la seguridad extremo-a-extremo (E2E) a nivel de capa de transporte, pero el capítulo da especial relevancia a la a seguridad E2E a nivel de aplicación. Para ello, se revisan diversas propuestas del IETF para la

obtención de la seguridad E2E a nivel de capa de aplicación. Sin embargo, la principal propuesta del IETF (OSCORE), está muy atada al despliegue de un protocolo de comunicaciones concreto (CoAP) y fuerza un sistema de comunicación cliente-servidor. Esta tesis considera que las cadenas de valor se benefician de sistemas de comunicación uno-a-varios, por lo que esta forma de transmisión debe tener en cuenta en las soluciones de securización de la información. Relacionado con esto último, se revisan las propuestas existentes para el desarrollo de sistemas de intercambio de datos vasados en un tipo de cifrado uno a varios denominado cifrado basado en atributos (ABE).

El capítulo 3 presenta una arquitectura basada en defensa en profundidad (DiD) para la protección de una planta industrial. El DiD ha sido identificado en el estado del arte como la principal estrategia de seguridad exigida por la norma IEC 62443 en ámbitos industriales, con instituciones como la ENISA o el ICS-CERT respaldando la adopción de esta estrategia. Por lo tanto, este capítulo unifica los criterios de las instituciones mencionadas e identifica los objetivos que debe cumplir una estrategia DiD para uso industrial, las capas que debe tener, así como las medidas de seguridad a desplegar en cada una dichas capas.

Sin embargo, la implementación de la arquitectura de seguridad por parte de cada planta de la cadena de valor no es suficiente para garantizar la seguridad E2E de la información intercambiada entre *partners*. Por lo tanto, hay que desarrollar una solución que sí sea capaz de garantizarla. Para ello, se han identificado los diferentes tipos de datos que se intercambian en una cadena de valor y se han definido los requisitos que debe cumplir la solución de intercambio de información.

El capítulo 4 evalúa y analiza diversas que ofrecen implementaciones de ABE. El estado del arte ha identificado multitud esquemas ABE, tanto para CP-ABE como KP-ABE y dCP-ABE. Sin embargo, todavía no existe un esquema de facto para ninguno de los tres. Como ilustra el Capítulo 2, mientras que algunos esquemas son vulnerables y debe evitarse su uso, otros requieren de un estudio más detallado para entender adecuadamente sus ventajas y desventajas. Además, existe un obstáculo añadido para el uso de ABE en proyectos industriales: la falta de librerías. Las dependencias utilizadas para desplegar una solución de seguridad tienen una influencia significativa en la eficiencia y la seguridad globales de la solución. Por tanto, el objetivo de este capítulo

es la elección de un esquema robusto y eficiente desde el punto de vista teórico, pero también considerando su implementación práctica.

Para conseguir todo esto, el capítulo realiza una evaluación cualitativa y de seguridad de los esquemas ofrecidos por varias librerías. La evaluación ha revelado qué librerías implementan esquemas identificados como rotos durante el estudio del estado del arte, qué dependencias matemáticas tienen y con qué modo AES se combinan. El capítulo también analiza qué librerías siguen recibiendo actualizaciones, cuáles reciclan código de otras y cuáles comparten desarrolladores. Al final de este análisis, se seleccionan cuatro candidatos potenciales para su despliegue en entornos industriales. Estas librerías se someten a un análisis experimental que revela el tiempo que necesitan dos dispositivos de diferentes capacidades para realizar operaciones criptográficas básicas. El esquema elegido para el uso industrial es denominado W11, concretamente la versión implementado por la librería OpenABE. Esta librería está escrita en C++, lo que la hace adecuada para dispositivos IIoT de baja capacidad. Su velocidad de generación de claves, cifrado y descifrado ha demostrado ser la más rápida de todas las combinaciones estudiadas.

El capítulo 5 presenta una arquitectura para el intercambio seguro de información dentro de una cadena de valor. Para ello, la flexibilidad que ofrecen los esquemas CP-ABE los convierte en una solución prometedora. Debido a la naturaleza dinámica e industrial de las cadenas de valor, el sistema de intercambio de información basado en ABE debe tener en cuenta los requisitos industriales, garantizar la confidencialidad e integridad E2E y ser flexible a cambios. En lo que respecta a la seguridad E2E, el estado del arte establece que la seguridad E2E en criptografía requiere que sólo el dispositivo que genera los datos y el dispositivo que es el destinatario final puedan acceder a la información. Por lo tanto, la información debe estar encriptada en el origen y ser descifrada únicamente por el dispositivo final.

Para garantizar la aplicabilidad industrial, se han identificado documentos especializados de la ETSI, que establecen los requisitos industriales y proponen el uso de un esquema CP-ABE con seguridad CCA. Además, los entornos industriales también tienen una larga vida útil, durante la cual los derechos de acceso a la información cambian. Por tanto, la solución de cifrado basada en CP-ABE también debe adaptarse a

los cambios del entorno para lograr esa flexibilidad que se mencionaba. Sin embargo, CP-ABE no dispone de un mecanismo propio para actualizar la política de cifrado. Por todo ello, es necesario desarrollar un sistema de actualización de las políticas de acceso y de los textos cifrados. Normalmente, esto implica desencriptar la clave simétrica y volver a encriptarla según la nueva política de acceso. Sin embargo, esto expone la clave simétrica y la deja vulnerable, rompiendo la seguridad E2E. En este sentido, ninguna de las soluciones estudiadas en el estado del arte es definitiva, y ninguna está orientada a entornos industriales. Además, la mayoría de ellas no consideran la seguridad CPA o CCA, o proponen esquemas propietarios cuya seguridad aún no ha sido analizada ampliamente por terceros.

Para solucionar lo arriba mencionado, este capítulo presenta Multi-Layered CP-ABE. Este sistema garantiza la confidencialidad mediante la combinación de CP-ABE y AES Galois/Counter Mode (AES-GCM). Para tener un sistema de actualización de políticas que cumpla los requisitos industriales de la ETSI, se presentan las funciones y los algoritmos de un sistema de actualización de políticas basado en el esquema CCA del ETSI que utiliza el esquema CP-ABE W11 como primitiva. También se identifican los roles dentro de la cadena de valor para desplegar Multi-Layered CP-ABE manteniendo la confidencialidad e integridad de los datos E2E. Finalmente, una vez cumplidas las restricciones industriales del ETSI, establecidos los roles del sistema de intercambio de información, y con la confidencialidad e integridad del E2E garantizadas, se analiza experimentalmente la propuesta.

Los dispositivos IIoT se benefician de esquemas ligeros, lo que hace que el coste computacional de CP-ABE puede limitar su aplicación en las redes IIoT. Por ello, es necesario cuantificar el tiempo necesario para las operaciones criptográficas básicas y el tamaño del texto cifrado generado. Los experimentos de este capítulo cifran datos industriales reales, utilizando una RPI4 y una RPI0 para modelar dispositivos industriales de alta capacidad y dispositivos IIoT. Los resultados experimentales demuestran la viabilidad práctica de la propuesta.

Por tanto, el esquema propuesto en este capítulo cumple con los requisitos industriales para los sistemas ABE establecidos por la ETSI, se basa en un esquema CP-ABE con CCA y cuenta con un sistema de actualización de políticas que mantiene la confi-

dencialidad y la integridad E2E.

El capítulo 6 presenta un sistema de prevención de la suplantación de atributos (ataque denominado en esta tesis como *attribute spoofing*). La seguridad del intercambio de datos basado en CP-ABE se basa en que los usuarios reciben claves que reflejan adecuadamente sus privilegios. También se basa en la suposición de que los atacantes no pueden obtener una clave privada para acceder a información a la que no deberían tener acceso. Estas suposiciones no son realistas y exponen el sistema a la suplantación de atributos. Por tanto, es necesario establecer un sistema de autenticación y validación de derechos de acceso para los usuarios del sistema basado en ABE. Para ello se despliega un sistema de almacenamiento de atributos distribuido y auditable, un sistema de validación de atributos y uno de autenticación de usuarios. El capítulo identifica dos vectores de ataque que pueden explotar la suplantación de atributos, así como las consecuencias que pueden tener para el sistema un ataque exitoso (escalada de privilegios, reducción de acceso, obtención ilegítima de claves privadas). Los ataques, además, se enmarcan en cuatro supuestos que definen el comportamiento del sistema.

Dado que el sistema de validación de atributos debe contar con un almacenamiento distribuido de atributos, este capítulo establece el uso de IPFS. Este protocolo de almacenamiento distribuido ofrece un funcionamiento de alta capacidad y garantiza la integridad de la información contenida en él. IPFS basa su funcionamiento en tecnologías ampliamente probadas, como los DHT. Además, no tiene un servidor central, y los nodos que participan en la red no necesitan confiar los unos en los otros. Para garantizar la auditabilidad de los atributos, IPFS se combina con IOTA, una DLT diseñada para dispositivos IoT. IOTA proporciona al sistema la auditabilidad que IPFS no puede garantizar por sí solo. Para enmarcar la propuesta en el contexto de una cadena de valor, el capítulo también establece cómo IPFS e IOTA se incluyen en el sistema de intercambio de información del Capítulo 5. Para ello, se asume que cada usuario del sistema pertenece a una empresa y que ésta se encarga de almacenar los atributos en la solución combinada de IPFS e IOTA. De esta manera, las autoridades que generan las claves privadas sólo tienen que recuperar la información de estos puntos. Esta solución se combina con FIM para garantizar la autenticación de los usuarios del sistema. Finalmente, tras el análisis cualitativo de la solución y la evaluación de cómo las medidas

desplegadas evitan los vectores de ataque identificados y cumplen con los requisitos técnicos establecidos, se evalúa la capacidad del sistema para ser desplegado en dispositivos IIoT. Para ello, se contabiliza el tiempo requerido para recuperar los atributos de un usuario y generar una clave, y se concluye que al ser inferior a 250ms, la solución es adecuada para IIoT.

El capítulo 7 presenta las conclusiones de la tesis, donde se resumen las principales contribuciones de cada capítulo, se listan las publicaciones resultantes de la investigación llevada a cabo durante estos años, y se indican las futuras líneas de investigación. Finalmente, el apéndice A indica la terminología criptográfica utilizada durante la escritura de esta memoria, y el apéndice B resume los conceptos de seguridad CPA y seguridad CCA.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF TERMS AND ABBREVIATIONS

ABE	Attribute-Based Encryption
AES	Advanced Encryption Standard
AES-CBC	AES Cipher-Block Chaining
AES-GCM	AES Galois/Counter Mode
AI	Artificial Intelligence
AP	Access Policy
CCA	Chosen Ciphertext Attack
CID	Content Identifier
CoAP	Constrained Application Protocol
COSE	CBOR Object Signing and Encryption
CP-AB-KEM	Ciphertext-Policy Attribute-Based KEM
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
CPA	Chosen Plaintext Attack
CPS	Cyber-Physical System
CRC	Cyclic Redundancy Check
CT	Ciphertext
DAG	Directed Acyclic Graph
dCP-ABE	Decentralized CP-ABE
DHT	Distributed Hash Table
DiD	Defense-in-Depth
DLT	Distributed Ledger Technologies
DMZ	Demilitarized Zone
DO	Data Owner
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
E2E	End-to-End
EDHOC	Ephemeral Diffie-Hellman Over COSE

ENISA	European Union Agency for Cybersecurity
ETSI	European Telecommunications Standards Institute
FIM	Federated Identity Management
FO	Fujisaki-Okamoto
HIDS	Host-Based IDS
HIPS	Host-Based IPS
IACS	Industrial Automation and Control System
IBE	Identity-Based Encryption
ICS	Industrial Control System
IdP	Identity Provider
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IIoT	Industrial IoT
IoT	Internet of Things
IPFS	Interplanetary File System
IPS	Intrusion Protection System
IT	Information Technology
ITRC	Identity Theft Resource Center
KEM	Key Encapsulation Method
KP-ABE	Key-Policy Attribute-Based Encryption
LSSS	Linear Secret Sharing Scheme
MITM	Man in the Middle
MPK	Master Public Key
MSK	Master Secret Key
NGFW	Next-Generation Firewall
NIST	National Institute of Standards and Technology
OSCORE	Object Security for Constrained RESTful Environments
OT	Operational Technology
PERA	Purdue Enterprise Reference Architecture

PLC	Programmable Logic Controller
PRNG	Pseudo Random Number Generator
PT	Plaintext
RBAC	Role-Based Access Control
RPI	Raspberry Pi
RPI0	Raspberry Pi Zero
RPI4	Raspberry Pi 4
SCADA	Supervisory Control And Data Acquisition
SK_{ABE}	CP-ABE Private Key
SSS	Secret Sharing Scheme
TLS	Transport Layer Security
TR	Technical Requirement
VPN	Virtual Private Network

"Begin at the beginning," the King said, gravely, "and go on till you come to an end; then stop".

Lewis Carroll, Alice in Wonderland

Chapter 1

Introduction

Digitalization is an unstoppable phenomenon that affects all facets of life and society. That includes the industrial sector, where the paradigm is shifting towards distributed automated systems, wireless communications, and the potential to connect to networks outside the Operational Technology (OT) network. The presence of Internet of Things (IoT) devices capable of processing, sending, or receiving information has increased in all environments, including industrial plants [8]. This niche has been dubbed Industrial IoT (IIoT) and has driven an industrial transformation along with other technologies like Artificial Intelligence (AI), Cloud Computing, and Big Data Analytics.

Thanks to this transformation, new benefits arise in the industrial environment, but also new risks and vulnerabilities. Because of the latter, it is essential to deploy an industrial security solution [9]. However, value chains continue to suffer attacks [10], which have increasingly worse consequences for the companies that suffer them. With the growth of Industry 4.0, the amount of information being transmitted between companies is also increasing. As a result, data breaches resulting from value chain attacks are affecting more and more users [11] and are more expensive [12].

Based on the above, it is crucial to establish a security system that guarantees the confidentiality of the information transmitted between partners in a value chain. There are many security solutions available at the plant level: Intrusion Protection System (IPS), hardening systems, and access control mechanisms, to name a few. These security systems are regulated by standards and regulations from a multitude of institutions like the IEC [13], the ENISA [1], or the ICS-CERT [14]. However, regulation is not so precise regarding data once it leaves the plant and is transmitted to another partner. While the general recommendation is to ensure its confidentiality and integrity, there is no indication of the 'how to.'

Therefore, this thesis strives to establish a secure End-to-End (E2E) information exchange system between value chain members. Traditionally, the industry has relied on

proprietary solutions for security and communications. However, many of the attacks it suffers today do not have an industrial origin; instead, they come from the Information Technology (IT) networks. Therefore, this thesis considers that security solutions initially developed for IT can be adapted to the industrial environment as long as industrial constraints are addressed. Their implementation allows us to abandon the aforementioned proprietary solutions by taking advantage of the capabilities of industrial equipment and the functionalities offered by IIoT devices. Formalized, the objective of this thesis is defined as:

> To design an information exchange system for value chains that guarantees E2E data confidentiality and integrity.

The following subsections outline the context in which this thesis is developed, as well as its requirements and limitations. Section 1.1 introduces the research questions that motivate this thesis dissertation and identifies the chapters that develop and answer them. Section 1.2 sets out the organization of this dissertation, summarizing each chapter and its main contribution.

1.1 Motivation and Research Questions

The concept "Industry 4.0" was introduced by the German government [15] to name a strategic initiative that promotes smart manufacturing by embracing technologies that increase the digitalization and interconnection of products. It is, therefore, a blanket term that encompasses many technologies and manufacturing strategies. To narrow it down, this dissertation considers that Industry 4.0 implies a network of interconnected industrial plants. In this sense, the information exchange required by smart manufacturing must include the supply chain and the value chain. It is important to clarify that historically, the supply chain encompassed all the activities involved in exchanging raw goods required to obtain a final product. In contrast, the value chain encompassed all actions that add value to the manufactured products. Since Industry 4.0 seeks added value in the manufacturing and exchanging of goods, this thesis uses the value chain as a concept that encompasses traditional supply and value chains.

As a consequence of the lack of a formal definition for “Industry 4.0", the concept of "security for Industry 4.0" also lacks a strict definition. Therefore, this dissertation considers that the supply and value chains generate sensitive information that needs to be exchanged securely. For the sake of writing convenience, we will refer only to the value chain, but the reader should consider that we are referring to all three.

The main security challenge in value chains is the management of the complex industrial ecosystem [9]. However, information flow between partners is vital for a robust industry environment [16], so it is vital to have a system that protects information from attackers and rival companies. To achieve this, it must be ensured that *i)* the data exchange is secure, and *ii)* the company sending or receiving the data meets the minimum security requirements. Therefore, Industry 4.0 security is the result of applying security within each plant and to the information flow between them.

It is essential to define a baseline security architecture value chain partners should deploy to establish a secure data exchange. However, due to the heterogeneity of industrial plants, it is not straightforward to establish a minimum security architecture that complies with the same guarantees along the entire value chain. This baseline security architecture must guarantee essential industrial requirements, such as uninterrupted plant operation [3], and comply with the standards and legislation of each sector and country [17]. However, these standards often define the requirements to be met but do not indicate how to achieve them. This brings us to the first question posed in the research process of this thesis, Question 1.

Question 1. — *Can we define and deploy a baseline industrial security architecture for the various partners in the value chain?*

— This question is answered in Chapter 3, "Security Architecture, Scenario and Requirements"

Establishing a security architecture by every partner in the value chain is necessary but insufficient to guarantee the E2E security of data exchanged between chain partners. Instead, it is imperative to define a system that can provide the exchanged data with confidentiality and integrity.

Value chains entail a massive data exchange between companies [18]. Moreover, sometimes the same information can be relevant to more than one partner. Industry 4.0, therefore, benefits from cryptographic algorithms that allow information to be encrypted once and decrypted by various users. Usually, cryptography is one-to-one: two users agree on the algorithm to be used, generate cryptographic keys, and use them to encrypt and decrypt the information. If the same message is sent to more than one user, the message must be encrypted for each one.

A straightforward solution to one-to-many encryption algorithms can be the deployment of group keys, where users are part of groups and receive a decryption key that allows them to decrypt messages sent to those groups. However, the effectiveness of this system is greatly affected by the size of the group, and users usually need as many

keys as there are groups to which they belong [19]. In contrast, one-to-many encryption algorithms allow the same encrypted message to be sent to several recipients and read by each authorized recipient without users having to manage several different keys.

A major algorithm family that meets this requirement is Attribute-Based Encryption (ABE) [20], which also supports one-to-many encryption, provides implicit authorization to data, and decouples encryption and decryption from user identities. However, there are numerous ABE schemes, so it is necessary to define which one is more appropriate for an industrial environment. Choosing the right ABE cipher for the industry is a trade-off between efficiency and security, which can be challenging. This leads to Question 2.

Question 2. — *Which ABE ciphers are suitable for deployment in industrial production environments?*

— This question is answered on Chapter 4, "ABE Libraries Analysis and Evaluation"

Once the most suitable ABE cipher for industrial deployment has been established, its compliance with industry guidelines issued by official institutions must be evaluated. Furthermore, information recipients may change during the lifetime of the industrial system. Similarly, previously encrypted information may have to be shared with new users or restricted to former collaborators. Therefore, the encryption system must be flexible in responding to these challenges while meeting E2E confidentiality and integrity requirements. To analyze this, we pose question 3.

Question 3. — *Can we establish an ABE-based system that can be deployed in value chains, adapt it to organizational changes, and guarantee E2E data confidentiality and integrity?*

— This question is answered on Chapter 5, "Multi-Layered CP-ABE with Access Policy Update"

Once data integrity and confidentiality have been addressed, the system's security depends on users obtaining the appropriate decryption keys. In ABE, users are not identified according to their identities but according to attributes that describe them. However, ABE algorithms do not integrate a validation system for these attributes.

Some solutions [21] consider that the key generators should handle and distribute the attributes. However, this solution implies that the generators know the attributes

of each user in the system. This reduces scalability and creates trust issues between collaborating and competing companies. Therefore, attribute spoofing is a security risk that must be considered to design a comprehensive security system to ensure E2E data confidentiality between partners in real-world scenarios. Consequently, we pose the last research question, 4.

Question 4. — *How do we deploy an attribute-validation system in the context of Industry 4.0?*

— This question is answered on Chapter 6, "Attribute-Spoofing Prevention"

1.2 Thesis Organization and Contributions

This thesis presents the results of the work undertaken to answer the research questions mentioned above. The dissertation's organization and each chapter's contributions are summarised below.

1.2.1 Chapter 2: Background and State-of-the-art

Chapter 2 introduces the background and state of the art for the rest of the thesis. First, it summarizes the important background of Industry 4.0: it identifies the essential security properties requested by the National Institute of Standards and Technology (NIST) and International Electrotechnical Commission (IEC), outlines the difference between OT and IT networks, and presents the high-level reference model for smart manufacturing proposed by the European Union Agency for Cybersecurity (ENISA). After that, it summarises the essential requirements presented in the IEC 62443, the leading industrial security standard. Afterward, it reviews the current proposals for securing information exchange in value chains and, related to this, also evaluates the impact of cryptography in industrial systems. With the industrial security background and state-of-the-art outlined, the chapter studies different approaches to achieve E2E data confidentiality. In particular, the chapter surveys application-layer security and ABE.

1.2.2 Chapter 3: Security Architecture, Scenario and Requirements

Chapter 3 presents the first contribution of the thesis. State of the art in Chapter 2 identifies DiD as a crucial component required by the IEC 62443 to implement an industrial security strategy. However, the standard does not define in detail how to deploy it. Thus,

this chapter identifies the objectives for the DiD strategy by bringing together the IEC, ENISA, and ICS-CERT criteria. It then defines the layers for the DiD strategy and the security measures to be enforced in each of them. In addition, the chapter concludes that using a DiD strategy alone does not guarantee that the exchange of information between value chain partners is secure. Therefore, it also defines the requirements that the E2E confidential data exchange system should meet.

Contribution of the Author: The author designed the proposed DiD architecture. This research work has resulted in a paper *"Securing IIoT using Defence-in-Depth: Towards an End-to-End Secure Industry 4.0"* [22], published in *Journal of Manufacturing Systems* in 2020.

1.2.3 Chapter 4: ABE Libraries' Analysis and Evaluation

Chapter 4 presents the second contribution of the thesis. One of the main deterrents to employing ABE is the complexity of knowing which of the many available schemes is suitable for industrial deployment. Therefore, this chapter analyses 11 ABE libraries considering security and efficiency. Since having a maintained library is crucial to deploy cryptographic schemes, the chapter proposes a methodology to select secure and maintained libraries. After this evaluation, only four libraries are considered potential candidates to be deployed in industrial environments.

The chapter analyses the ABE schemes provided by these libraries, identifies the symmetric ciphers they use underneath, and uncovers which libraries provide vulnerable ABE scheme implementations. The analysis also considers the libraries' mathematical dependencies and whether they are suitable for industrial deployment. After the qualitative evaluation, the chapter provides a practical implementation of every scheme provided by the chosen libraries, analyzing their efficiency in a Raspberry Pi 4 (RPI4) and a Raspberry Pi Zero (RPI0). The results provide insight into which libraries and schemes are best suited for industrial deployment.

Contribution of the Author: The author proposed the methodology to select the libraries and performed the library analysis and evaluations. This research work has resulted in two papers, namely:

- *"All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) Libraries"* [23] presented at the *URSI 2022*

- *"Too Many Options: A Survey of ABE Libraries for Developers"* [24] that has been to submitted to "*Computer Networks*" and published as a preprint in *arXiv* in 2022.

1.2.4 Chapter 5: Multi-Layered CP-ABE with Access Policy Update

Chapter 5 presents the third contribution of the thesis, an architecture for secure information exchange within a value chain. The scheme guarantees E2E confidentiality and integrity through CP-ABE and AES Galois/Counter Mode (AES-GCM). The proposal also includes a system for updating policies, one of the CP-ABE open issues identified in Chapter 2. The scheme complies with the industrial requirements for ABE systems established by European Telecommunications Standards Institute (ETSI) and is based on a CP-ABE scheme with Chosen Ciphertext Attack (CCA) security. The chapter introduces the roles and algorithms needed to deploy the system and experimentally tests the feasibility of the system. The experiments are carried out on a RPI4 and a RPI0 and examine the size growth of the ciphertext and the time required by the devices to perform the encryption operations.

Contribution of the Author: The author proposed, designed, and tested the information exchange architecture. This research work has resulted in two papers, namely:

- *"Multi-Layered CP-ABE scheme for flexible policy update in Industry 4.0"* [25] presented at the *10th Mediterranean Conference on Embedded Computing (MECO 2021).*
- *""They got my keys!": On the Issue of Key Disclosure and Data Protection in Value Chains"* [26] that has been presented at the *2nd IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0 (ARCI' 2022)).*
- *"End to End Secure Data Exchange in Value Chains with Dynamic Policy Updates"* [27] that has been submitted to "*Future Generation Computer Systems*" and published as a preprint in *arXiv* in 2022.

1.2.5 Chapter 6: Attribute-Spoofing Prevention

Chapter 6 presents the last contribution of the thesis, an attribute spoofing prevention system. The information exchange scheme presented in Chapter 5 does not establish how the attribute authorities obtain the information needed to generate users' private keys. This situation is common in the literature, and most works do not consider the possibility of attribute spoofing. This chapter defines attribute spoofing, identifies two attack vectors, and proposes how to address them. To do so, certain assumptions are established for the system, and the attacks' consequences are identified. The solution combines Interplanetary File System (IPFS), IOTA, and Federated Identity Management (FIM) to prevent attribute spoofing. The chapter presents a qualitative analysis that discusses how the proposed technologies mitigate the attack vectors and an experimental analysis that evaluates how long it takes for the authorities to generate the users' private keys.

Contribution of the Author: The author designed and evaluated the attribute spoofing prevention system. This research work has resulted in two papers; namely:

- *"Are you what you claim to be?" Attribute Validation with IOTA for Multi Authority CP-ABE"* [28] presented at *BLOCKCHAIN 2022*.
- *"Trust your users as far as you can validate them: Secure Attribute Retrieval for ABE Schemes"* that has been submitted to *"Smart Cities Journal"* in 2022.

1.2.6 Chapter 7: Concluding remarks

This last chapter concludes this thesis by summarizing the general results from the research carried out throughout this thesis. This chapter also provides the results of research in the form of contributions published in specialized journals and congresses. Finally, it outlines the future research lines arising from this thesis's development.

Somewhere, something incredible is waiting to be known

Carl Sagan

Chapter 2

Background and State-of-the-art

Currently, the industry does not have a de facto solution for secure data exchange between supply chain members. Instead, companies rely on proprietary solutions. This means that if the same company is involved in several production and distribution environments, it has to manage several solutions for information exchange. Therefore, there is a need for a universal solution that can be adapted to the majority of use cases.

In order to exchange data securely, companies must have a minimum security architecture in place. This architecture, and the security solutions deployed, must be equivalent across members of a production environment. Section 2.1 of this chapter thus discusses the current state of security in Industry 4.0. Section 2.1.1 analyzes the background and state of the art of security within an industrial plant. Next, Section 2.1.2 reviews the solutions for secure data exchange in a value chain. It is further clarified that, in this thesis, the concept of information exchange in the value chain is used to cover any information exchange linked to the supply chain, as well as any data exchanged between members of the chain that can be used to gain an advantage over competitors. Linked to this information protection, Section 2.1.3 studies the impact of cryptography in the industry since it is the primary tool for protecting information confidentiality and integrity.

Section 2.2 examines E2E security solutions for information exchange. Specifically, Section 2.2.1 reviews E2E security solutions at the transport layer level, and Section 2.2.2 analyses security solutions at the application layer. Section 2.3 studies E2E security from the cryptographic point of view. Since the context of this thesis involves communication between several participants, Section 2.3.1 introduces broadcast encryption, one of the first one-to-many encryption ciphers. Following this, Section 2.3.2 reviews the current state of KP-ABE schemes, Section 2.3.3 CP-ABE, and Section 2.3.4 dCP-ABE. The chapter ends with Section 2.4, a summary of the essential ideas.

2.1 Security in Industry 4.0

The goal of Industry 4.0 is to manufacture higher-quality products and reduce production costs through key enabling technologies. The IIoT, a subset of IoT applied to industry and considered by some authors as the next step in industrial communications [29], is one of these technologies. IIoT devices make industrial networks more flexible and connect previously isolated industrial plants [29]. However, these devices significantly impact industrial security, exposing industrial systems to situations not foreseen during their design. Historically, industrial systems' complexity and limited accessibility have been considered a sufficient security measure to protect industrial environments [30]. However, the introduction of IIoT devices shifts the paradigm, and now security should be considered another parameter when designing industrial systems for Industry 4.0.

For a proper deployment of security in Industry 4.0 systems, it is necessary to consider the particular characteristics of IIoT devices. As already mentioned, the IIoT is a particularization of the IoT for industrial environments. Therefore, many of its features overlap, while others are specific to IIoT devices. Table 2.1 draws on the information identified by various sources like [29], and [31] to summarize the differences and similarities between these two families of devices.

Table 2.1 Feature comparison between IoT and IIoT devices. ◪ symbolizes that the feature may not exist in all Industry 4.0 environments, ■ indicates its presence, and □ indicates its non-existence.

Feature	**IoT**	**IIoT**
Battery Limitation	■	◪
Sleep-Mode	■	◪
Computing Limitation	■	■
Interdependence	■	■
Heterogeneity	■	■
Structured Nodes	□	■
Scalability	■	■
Interoperability	■	■
Very High Data Volume	□	■

Some features of IoT systems, such as battery limitation and sleep mode, are not always found in IIoT devices. Being part of an industrial network, these devices may have a continuous power supply. Therefore, although battery limitation may be present

in IIoT devices, it is not one of their major limitations. Meanwhile, computing limitations can severely affect IIoT devices. In industry, computing limitation describes the capability to perform complex tasks in a given amount of time. This limitation can derive from hardware computational constraints and the deadlines to be met by the device [31]. Therefore, these constraints severely restrict the security solutions deployed in the industry. In addition, interdependency between devices exacerbates the effect of computational constraints. Interdependency describes a device's ability to affect another's operation [31]. This trait can be exploited by attackers, altering the operation of one device to trigger unplanned actions in the industrial network. For example, a modification of sensor data can affect the actuators and the control system, putting the availability of the entire system at risk.

Therefore, the features described above, coupled with the heterogeneity of the IIoT devices and the large volume of data they manage, can severely limit the security solutions suitable for scenarios involving IIoT devices.

2.1.1 Security within the Enterprise

The Essential Security Properties

This section addresses the essential security principles already known in IT environments and relates them to the industrial environment. For this purpose, they are defined as outlined by entities like the NIST [4] or the IEC [13].

- **Availability:** Industrial systems should be designed considering fault tolerance to ensure availability [32]. Consequently, critical devices and networks should have a redundant counterpart that replaces the original in case of a failure or security breach [4]. Redundancy mechanisms minimize the effect of DoS attacks [13].
- **Authentication and authorization:** According to IEC 62443-4-2 [13], every user must be authenticated and authorized before performing any action. The way advised by NIST [4] to perform this access control is to use allow-lists and only enable communications between authenticated and authorized peers.
- **Integrity and confidentiality:** Integrity breaches can have dangerous consequences for companies, especially if they expose sensitive information or industrial secrets. Therefore, data must remain unaltered and confidential during capture, retrieval, update, storage, and transport. Only authorized users should be able to read or modify it.
- **Access control:** The NIST [4] recommends implementing fine-grained access control systems such as Role-Based Access Control (RBAC), as robust access

control systems decrease impersonation attacks and promote confidentiality. Preventing attackers from accessing sensitive information or control systems prevents them from compromising industrial devices [33].

- **Non-Repudiation:** This property ensures that the authenticity of the transmitted information cannot be questioned [34]. This feature is particularly relevant in user interfaces [13], where actions must be reflected in the system identifying the user who performed them.

The OT and IT Networks

Manufacturing networks are complex infrastructures composed of OT and IT networks. The IT network comprises the technologies used for information processing and communication equipment. In general, it is an umbrella term often used to group data and information management activities. In contrast, the OT network is related to the industrial equipment responsible for monitoring and controlling physical devices.

The stark differences between the two domains lead to OT and IT networks having different priorities in terms of security. These priorities are shown in Table 2.2, where it can be seen that IT networks follow the order of the CIA triad, i.e., confidentiality, integrity, and availability. In contrast, this prioritization is modified in the OT domain [3], where system availability becomes the top priority.

Table 2.2 Prioritisation of security requirements for IT and OT networks [3].

Priority Level	OT	IT
1	Availability	Confidentiality
2	Integrity	Integrity
3	Confidentiality	Availability

However, this distinction becomes blurred with the inclusion of IIoT devices in OT networks. On many occasions, IIoT devices reside in the OT network, obtaining process data that they then send to the IT network [35]. Moreover, sometimes IIoT devices send such data to the cloud, which is processed and returned to the OT network [36]. Therefore, Industry 4.0 security solutions must include both networks as a separate but interlocking whole.

Because of that blending between OT and IT networks, it is necessary to establish new industrial architectures that reflect this co-dependency. One of the most well-established architectures is Purdue Enterprise Reference Architecture (PERA) [37],

which focuses on planning and decision-making processes. However, it is a rigorous model unsuitable for the current industrial development derived from the presence of IIoT devices [38].

Therefore, new architecture models better suited for Industry 4.0 have been proposed. Among them is RAMI4.0 [39], from the German government's Industrie 4.0 platform. However, this architecture is complex and intricate for users to understand. Furthermore, it does not clearly define how new technologies such as artificial intelligence should be included or how information should be exchanged between processes and the cloud [40]. Thus, the ENISA introduced its high-level reference model based on the Purdue model [1] (Figure 2.1).

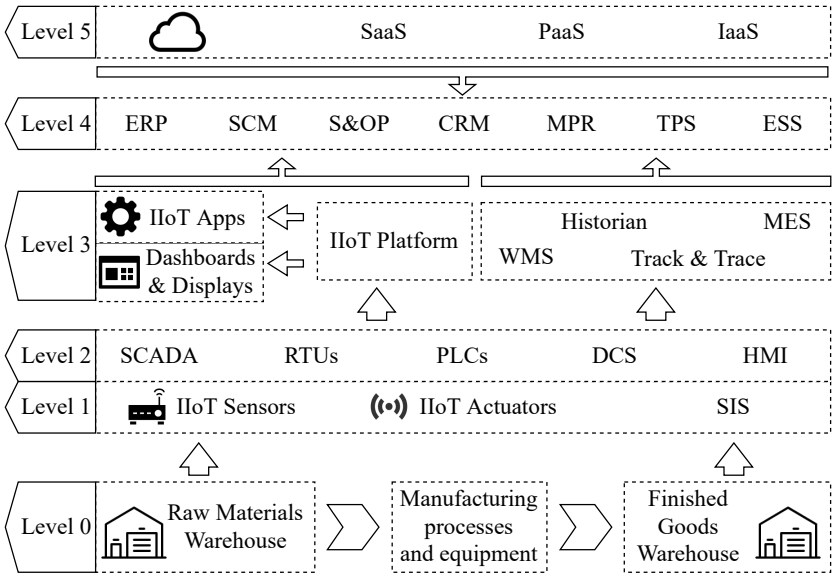


Fig. 2.1 ENISA high-level reference model [1].

In the high-level architecture proposed by the ENISA, OT network encompasses levels 1 and 2; the OT-IT connection level 3; the IT network level 4 and the third-party services level 5. The distribution is shown in Figure 2.1. Thus, due to the increased connectivity in OT network devices, communication between IT and OT is more flexible than in Industry 3.0. Proof of this is the existence of level 3, where both networks overlap.

However, despite the blending between IT and OT networks, a straightforward adaptation of IT security strategies to OT networks is not possible due to the different requirements of both types of networks. The main features that characterize OT networks are shown in Table 2.3, which summarizes the analysis presented in [4]. These

features affect the design of the security architecture. It is especially relevant to highlight the strict latency requirements, the need for a fault-tolerant design, or the much longer lifetime of OT systems compared to IT systems.

Table 2.3 Summary of OT and IT networks differences [4].

	OT	**IT**
Performance requirements	Real-Time Delays unacceptable	No Real-Time Delays acceptable
Fault-Tolerance	Essential	Not important
Updates	Should first be implemented in a controlled environment	Updates are straightforward
Communications	Proprietary protocols Complex Networks	Standard protocols IT networking practices
Lifetime	10-15 years	3-5 years
Device Location	May be remote and isolated	Local and easy to access

Various institutions worldwide, such as the NIST [4] or the Spanish INCIBE [41] have addressed the security issue within industrial ecosystems. Regarding the IEC, they proposed the IEC 62443 series, which defines the guidelines to protect industrial systems through their lifecycle. More about these series is explained in the following subsection.

IEC 62443

IEC 62443 is a series of standards developed by the IEC and dedicated to Industrial Automation and Control System (IACS) security. They establish the requirements for a secure IACS network since security requirements between IT and OT differ enough (Tables 2.2 and 2.3) that security solutions cannot be directly applied to OT.

Since availability is the main requirement for OT networks, the IEC 62443 standard takes a risk-based approach. It balances the level of security so that availability is affected as little as possible. Requiring the same level of security for all industrial systems is inefficient, and instead, the IEC establishes a holistic approach to system securitization. This implies that security solutions and procedures require the direct involvement of system users. These users can be one of the greatest vulnerabilities of the system and one of its strengths: they can identify the critical parts of the system and its vulnerabilities and apply the IEC 62443 standards accordingly.

To balance security and availability, the IEC establishes four security levels in IEC 62443-3-3 [5], presented in Table 2.4. These security levels make it possible to identify the particular needs of each system, thereby avoiding the establishment of generic security requirements that could put availability at risk in certain systems. Users should use the IEC 62443-1-1 [42] standard to identify the security levels required for their industrial system. For these security levels to be adequate, they must be studied during the system design phase and reviewed periodically [17].

Table 2.4 Security Levels established in IEC 62443-3-3 [5] and as summarised by [6].

Security Level	Skills required by the attacker	Motivation	Attacker's intent	Resources
SL 1	No skill	Mistakes	Non-intentional	None
SL 2	General	Low	Intentional	Few
SL 3	IACS-specific knowledge	Moderate	Sophisticated (attack)	Moderate
SL 4	IACS-specific knowledge	High	Sophisticated (campaign)	Extensive

With the security levels identified, users must develop the security strategy to deploy. For this, one of the requirements of IEC 62443 is DiD. DiD is identified and defined for the first time in IEC 62443-1-1. IEC considers it impossible to meet all the requirements of the different security levels with a single countermeasure. It, therefore, proposes the use of different security layers. DiD is mentioned again in IEC 62443-2-3 and IEC 62443-3-3. The latter considers that industrial networks should be divided into zones, that communications between these zones should be monitored and controlled, and that all of this should be included in the DiD strategy. Finally, IEC 62443-4-1 explains that DiD is required to achieve a risk-based approach to security. Thus, since DiD has been identified as a crucial strategy to achieve security by design according to IEC, it is one of the approaches studied in this thesis for securing industrial plants. Finally, it is worth noting that IEC is considered the leading industry standard. However, other countries may have similar standards and documents. Authors of [43] provide a mapping between IEC 62443 and other national standards from the USA, Germany, Norway, France, Spain, and The Netherlands.

2.1.2 Security in Value Chains

The concepts of value chain and supply chain are closely related in Industry 4.0. A supply chain comprises all the essential steps to deliver a product or service to a customer. These steps generally include logistics (i.e., distribution of the product), manufacturing, and the procurement of raw materials. Value chains were described by Porter [44] as

any activity devised to add value to a product, including but not limited to logistics, manufacturing, marketing, or after-sales service. Industry 4.0 is customer oriented and considers that manufacturing must add value to the final product. Thus, several companies collaborate in the value and supply chains to meet customers' needs. Exchanging data between retailers, customers or manufacturers can provide companies with a competitive advantage and is essential to increase efficiency and reduce costs in Industry 4.0 [45]. Therefore, this dissertation considers that any data exchanged in the supply or value chains are sensitive and shall be secured. For writing efficiency, we will refer to "value chains"; however, we also include "supply chains" in the definition.

Security is one of the critical issues for value chains. In particular, information exchange vulnerabilities are considered the primary security concern in these scenarios [46]. Real-world experiences show that such concerns are not unfounded. According to the 2020 IBM Data Breach Report, that year saw a surge in attacks on these infrastructures, exposing sensitive information and affecting vital industries [10]. Value chain attacks have grown in frequency and severity in the following years, causing IBM to include, for the first time, a chapter about data breaches resulting from supply chain attacks in their 2022 IBM Data Breach Report [12]. The report shows that the average total cost of a supply chain compromise was €4.43 million. Despite only one-fifth of the data breaches resulted from a supply chain attack, their individual cost is greater than the average of a data breach attack, €4.32 million.

The growth in value chain attacks leads to increased data breaches, exposed information, and affected users. This is analyzed by the Identity Theft Resource Center (ITRC) [11]. According to their Data Breach Analysis report, during the first quarter of 2021, the number of users affected by data breaches derived from value chains jumped from 8 million to 51, increasing by 564% compared to the last quarter of the previous year. Despite everything, the collaboration between partners in value chains improves the chain performance [47]. Therefore, partners must update their risk management strategies to prevent data breaches as business interconnections grow. In this way, they protect themselves from compromised value chain members.

In order to establish a security system to prevent data breaches, it is necessary to analyze how data is being exchanged. After analyzing the current situation of value chains, the researchers in [16] establish four main data flows: supplier-producer, maintenance provider-producer, producer-customer, and producer-collaborator. This is consistent with the analysis shown in [48], in which the authors point out that information exchange between the different hops of the value chain is limited. Company privacy and security must be guaranteed for efficient collaboration within value chains. Thus, steps have to be taken to secure the information exchange.

Security solutions developed for supply chains have to be adapted to value chains so that they can be applied to both physical and digital goods, as well as associated production parameters. Previous research [49] combines cloud computing and a cryptographic envelope to protect product delivery. However, this approach is limited to transferring finished digital products and does not consider the exchange of manufacturing information. Researchers in [50] combine Bloom filters [51] and Oblivious Transfer [52] to guarantee a private industrial parameters exchange. However, this solution reduces the control retained by data producers by allowing customers to retrieve data without the data producers knowing what has been transferred.

Regarding logistics security, the authors of [53] have developed a secure E2E sensing system for supply chains. Their solution focuses on ensuring that sensor readings are reliable and that each parameter is related to an existing physical event. Since the proposal is limited to sensor readings and does not consider other types of data, its direct application to digital assets in a value chain is not straightforward, which reduces its competitive advantage. Exclusively focused on OT network, researchers in [54] develop a security system for publish/subscribe communications regarding Cyber-Physical System (CPS). However, this proposal does not consider the idea of a value chain or the blending of IT and OT.

2.1.3 The impact of Cryptography in Industrial Systems

IIoT devices collect the exchanged information en masse, which has increased in volume, variety, and complexity [55]. This means that, as Table 2.1 introduced, the data volume IIoT devices manage is higher than typical IoT applications. Besides, because of the constrained nature of IIoT devices, sometimes data processing is carried out in edge devices or the cloud [56] [57]. This results in big volumes of data having to be processed, encrypted, and sent through different channels.

Industry 4.0 handles sensitive information related to the manufacturing process. Therefore, maintaining data confidentiality is vital to any Industry 4.0 security architecture, which is achieved through encryption. However, industrial data encryption is a sensitive issue in manufacturing due to the large volume of data to be managed and exchanged [58]. Furthermore, encryption and decryption are computationally expensive operations that may introduce latencies, and thus IIoT devices' constraints must be considered. To this end, lightweight encryption ciphers originally devised for IoT may be suitable for the IIoT. As explained in [17], IoT security techniques can be applied to smart manufacturing as long as the particularities of the new domain are addressed.

Regarding the properties of the different cryptographic approaches, asymmetric cryptography requires many resources compared to symmetric cryptography. There-

fore, the NIST [4] considers that asymmetric cryptography is more suitable for administrative purposes (e.g., finance or logistics), while the data flow and network traffic can be secured with symmetric cryptography. However, symmetric cryptography involves sharing a key beforehand, which is not always possible [29], and thus it is combined with asymmetric cryptography in what it is called public-key cryptography. Finally, it is essential to consider that not every IIoT node has the processing power or the required time for cryptographic operations. In this case, relegating cryptography to hardware accelerators [4] may be the only solution.

Intending to develop encryption schemes suitable for industrial environments, researchers have studied different options to achieve this. The authors of [59] use an Open Source Programmable Logic Controller (PLC) to embed the encryption in the controller. However, most PLCs in industrial environments are not open source, so this solution's applicability is significantly reduced in real environments. Researchers in [60], [61] and [62] analyze the possibility of using symmetric and asymmetric encryption in Supervisory Control And Data Acquisition (SCADA). However, the direct application of asymmetric cryptography is too heavy for industrial systems, and symmetric encryption requires exchanging keys beforehand. The key exchange and management systems, as [17] explains, require too much computational power and may interfere with communication times. A similar conclusion about not having an industrial-appropriate key management system is expressed by [63]. Therefore, there is still the need for an encryption scheme compatible with the constrained nature of industrial devices, which includes a key management solution and provides the system with E2E security.

2.2 E2E Security for Information Exchange

Information is one of the main assets to protect in industrial environments. As stated in Section 2.1.2, ensuring data confidentiality reduces the consequences of data breaches, and solutions that provide it should be explored. In this sense, E2E confidentiality guarantees that only the original owner of the data and the intended end user have access to the information. To this end, this section explores ways to achieve it through transport-layer and application-layer solutions.

2.2.1 E2E Security at the Transport Layer: TLS and DTLS

Transport layer E2E security ensures that client data reaches the server reliably [64]. This security is achieved through transport layer security protocols like TLS [65] and DTLS [66]. TLS protects the communications channel between a client and a server

located on the premises using TCP as the transport protocol. DTLS is based on TLS but works over UDP instead of TCP.

TLS can be used to secure data communications across partners in Industry 4.0. However, authors in [67] claim that TLS is vulnerable to Man in the Middle (MITM) attacks coming from application-layer proxies. Thus, they propose an authentication mechanism to prevent these impersonation attacks. However, they only consider MITM attacks in web applications. Regarding the use of DTLS, the authors in [68] present an IoT authentication scheme based on DTLS that protects data E2E through middleboxes such as routers or gateways. However, a later work [69] claims that the proposal in [68] breaks E2E security in the presence of forward proxies. Finally, in a more recent study [70], authors analyze the use of DTLS for IIoT. They compare the performance of DTLS with ABE and Blockchain and include the pros, cons, and appropriate domain of use for each solution. In the case of DTLS, they identify it as unsuitable for scenarios with brokered communications because it implements security hop-by-hop, meaning that information is always decrypted at the broker. Hop-by-hop security implies that when there are compromised intermediaries, E2E confidentiality is lost. Figure 2.2 shows that a node sends a message over a secured channel, but an attacker has compromised the intermediary node. This attacker can obtain the message, read or alter it, and send it to the recipient node. Thus, neither the sending node nor the receiving node would be aware that the confidentiality and integrity of the message have been breached.

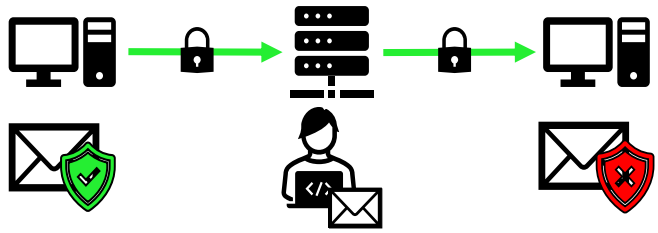


Fig. 2.2 TLS and DTLS security. Security is guaranteed for every security association, but not in the presence of middleboxes.

Although traditional industrial systems are hierarchical, Industry 4.0 systems tend to decentralization [71]. Therefore, messages pass through proxies, gateways, and brokers to save memory and bandwidth or perform protocol translation operations [72]. These middleboxes are crucial to grant scalability, efficiency, and interoperability to Industry 4.0. However, as already explained and shown in Figure 2.2, TLS and DTLS only secure the communication channel between client and server.

Therefore, security at the transport layer level is insufficient, and additional security

is required to guarantee data confidentiality and integrity. In this regard, security at the application level could solve this problem.

2.2.2 E2E Security at the Application Layer: Object Security

E2E security at the application level requires maintaining confidentiality and integrity all the way from the source to the end user, allowing proxies and gateways to do their job, as shown in Figure 2.3. The figure shows how, even if an attacker has compromised the middlebox, they get no access to the relayed information. However, the middlebox can still correctly forward the message to the final endpoint. To do this, these devices must only be given access to the indispensable parts of the message, such as the header, while the rest is hidden from them.

Fig. 2.3 Application layer E2E security. Middleboxes only have access to the information they need to forward the message to the next endpoint.

To ensure application layer security, a naive approach would be to assume that protocols such as Constrained Application Protocol (CoAP), the communication protocol for IoT devices proposed by the Internet Engineering Task Force (IETF), can avoid the problem identified in the previous section. However, CoAP does not include security, so works such as [73] use CoAP alongside DTLS to protect information. Although CoAP is specially designed for IoT devices, it is not so widely used in the industry to implement a solution highly dependent on CoAP. Furthermore, many CoAP-based solutions relegate security to TLS and DTLS, and not all include authentication and authorization [74]. Instead, implementing other mechanisms, such as object security, is necessary.

Object-based security protects the content of a message instead of the communication channel. This is achieved using "secure objects," consisting of a header, an encrypted payload, and a label that verifies integrity [75]. Secure objects are information containers; thus, a single message can carry several objects. This way, it is possible to obtain fine-grained access to the information.

The first object-based security solution for IoT was first proposed in the protocol

OSCAR [76]. This protocol achieved low energy consumption and latency while it ensured confidentiality through middleboxes. However, it did not include an object security format suitable for constrained devices, so the authors considered that there was potential for increasing the efficiency of the architecture in such scenarios [75]. Thus, years later, the IETF proposed OSCORE. This application-layer security protocol uses CBOR [77], an optimized binary data format for highly constrained environments, and CBOR Object Signing and Encryption (COSE) [78] to provide CBOR with security mechanisms, such as creating and processing signatures, message authentication codes, and encryption.To establish the encryption keys required to encrypt information, OSCORE uses Ephemeral Diffie-Hellman Over COSE (EDHOC) [79], a lightweight key exchange protocol. EDHOC has a small message overhead, making it efficient for technologies with duty-cycle or battery limitations. Finally, OSCORE uses CoAP for messaging.

OSCORE improves COSE's security by encrypting the request or response code in the original header and placing it in the encrypted payload. A dummy code is then placed in the new header: POST for requests and CHANGED for responses. This prevents attackers from changing a PUT to a DELETE and deleting a resource. Figure 2.4 shows how OSCORE messages are built upon CoAP messages.

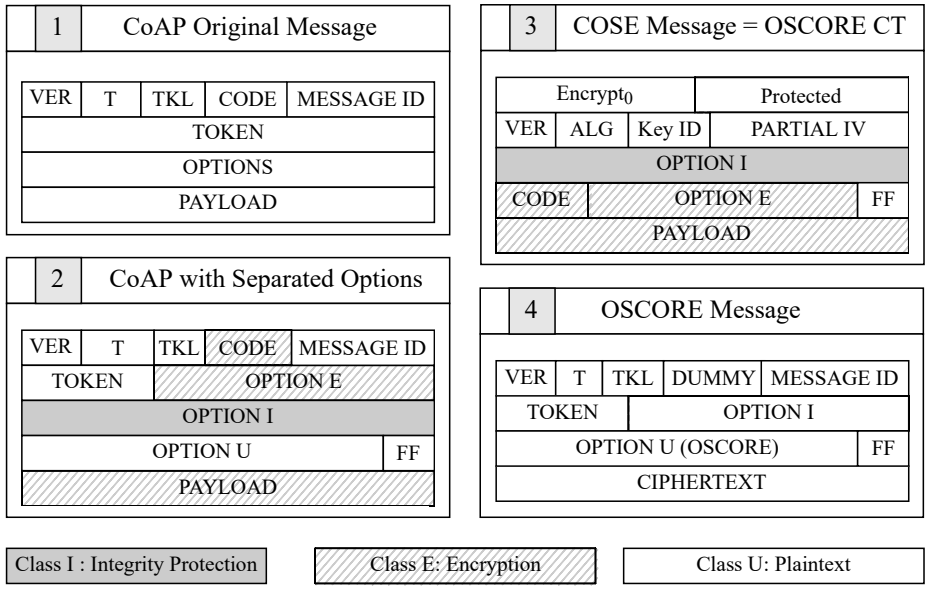


Fig. 2.4 Composition of OSCORE messages [2].

Message 1 shows the original form of an original CoAP message before protecting it. Message 2 identifies the sensitive fields to be protected. Payload and code have to

be encrypted, and the options field is encrypted, integrity protected, or left as plaintext, depending on the options it contains. The integrity of the options necessary for brokers to process the message must be guaranteed, while those not necessary for brokers can be encrypted. Message 3 shows how the fields identified as sensitive have been protected by encapsulating them in a COSE message. Finally, message 4 shows a typical OSCORE message built on top of a CoAP message. As code is protected within the COSE message, that field is replaced by the dummy code. The options field is replaced by the options required for message processing, and an identifier is added so brokers can identify the message as an OSCORE message. Finally, the COSE message is put into the ciphertext field. Thanks to this combination of protocols, OSCORE guarantees E2E security even in the presence of brokers while allowing them to operate normally with substantial security and privacy improvements [80].

Regarding performance, OSCORE is designed for constrained networks, so it is highly efficient for IIoT nodes. As shown in [81], OSCORE has less overhead than the combination of CoAP and DTLS, is faster in both single-hop and multi-hop scenarios, and handles retransmissions better. Furthermore, OSCORE performs better than DTLS while achieving application layer security [69]. Despite all of this, OSCORE has two significant disadvantages. The first one is that it secures information one by one. This implies that nodes must set up a key negotiation via EDHOC with each node they wish to communicate with. They must also perform a one-to-one OSCORE message exchange. All this derives from the second disadvantage, OSCORE being built on top of CoAP, a client-server messaging protocol. This dependency is something the previous section identified as unsuitable for Industry 4.0 due to the difficulty of extrapolating it to other communication protocols. Therefore, there is still a need for an application layer security solution capable of providing E2E confidentiality and integrity to sensitive industrial data.

2.3 Cryptography-Based E2E Security

As the previous section identified, the main disadvantages of OSCORE were its heavy reliance on CoAP and its focus on one-to-one information exchange. Instead, Industry 4.0, with its massively interconnected value chains, benefits more from solutions that allow sending information from one to many. Therefore, there is a need for a system that, like OSCORE, guarantees E2E data confidentiality at the application layer level but is not subject to the same constraints. To this end, this section studies one-to-many encryption algorithms, which are a potential solution.

2.3.1 Broadcast Encryption

Fiat and Naor outlined the first formal version of one-to-many encryption in their broadcast encryption proposal [82]. Their scheme allowed a user to broadcast a message to a privileged set of n users without allowing users from outside the privileged group to learn anything about it. This solution, however, was only secure against the collusion of a limited number of users. Thus, in 2005 Boneh *et al.* presented their version of broadcast encryption with short ciphertext and private keys, full collusion resistance and CCA security [83].

One of the main drawbacks of the broadcast encryption schemes presented above is the need to have a user list for encryption. To solve this, authors in [84] combined broadcast encryption with Identity-Based Encryption (IBE), resulting in what authors call Identity-based Broadcast Encryption (IBBE). Thanks to the combination, this scheme enhances efficiency by having a shorter public key and not limiting the total number of possible users during system setup. However, in a value chain information exchange system, it is more convenient to send messages to a group of users that meet certain characteristics, rather than to a specific identity.

With the emergence of the cloud and the rise of distributed networks, one-to-many encryption systems are increasingly applicable. For example, in a recent work, authors of [85] present a one-to-many encryption system for image encryption with homomorphic decryption. However, homomorphic encryption is computationally heavy, discouraging its application in resource-constrained devices like IIoT devices. Authors in [86] combine Blockchain and symmetric encryption. The proposed system allows authors to encrypt for multiple receivers, but the identity of those receivers must be known from the beginning. These limitations hinder scalability, making it difficult to add new users, and can have an unpredictable impact on system efficiency.

Currently, one of the most widespread one-to-many encryption systems is ABE, which provides encryption schemes that bind information encryption and decryption to attributes (e.g., $\mathbb{A} = [att_1,\ att_2,\ att_4,\ att_5]$) and an access policy (e.g., $AP = ((att_1\ AND\ att_2)\ OR\ \ att_3)$. Since access policies are defined according to attributes, access to information is only granted if the policy is fulfilled. As a result, ABE provides one-to-many encryption, offers implicit authorization to data, and decouples encryption and decryption from users' identities. Thus, ABE improves encryption efficiency for one-to-many communication models, a feature particularly relevant for brokered communications in industrial environments.

The first ABE algorithm was presented under the name of Fuzzy Identity-Based Encryption [20], giving more granularity to IBE schemes [87]. IBE schemes encrypt and decrypt information according to the receivers' identities. The scheme presented

in [20] provided more flexibility to encryption by using attributes instead of identities. Fuzzy Identity-Based Encryption eventually led to the term ABE and the development of a new family of asymmetric encryption algorithms. ABE continued to be developed in the cryptographic field, giving rise to many different modes. Authors in [88] present a complete taxonomy of ABE schemes. However, it can be considered that all of them may be grouped under three different modes: Key-Policy Attribute-Based Encryption (KP-ABE) [89], Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [90] and a decentralized version called Decentralized CP-ABE (dCP-ABE) [91]. These modes are explained in the following sections.

2.3.2 Key-policy Attribute-Based-Encryption

The operation of KP-ABE is based on the use of access policies to create the users' private keys and the generation of ciphertexts by applying attributes. Figure 2.5 shows that if the user policy meets the ciphertext attributes, access to the information is obtained.

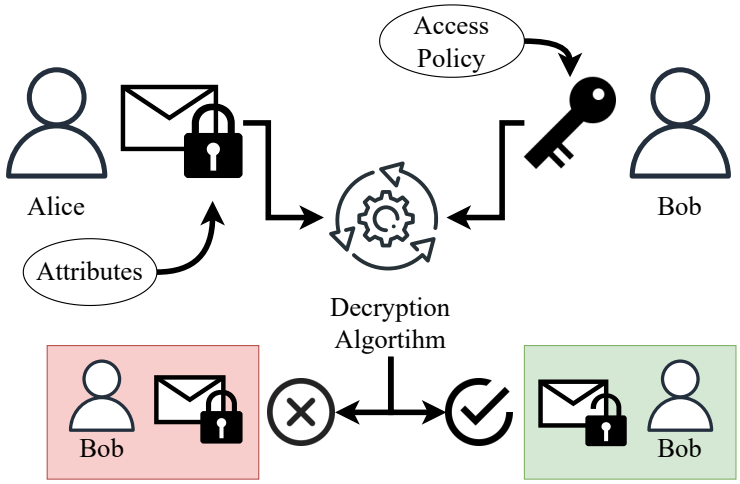


Fig. 2.5 KP-ABE Encryption and Decryption. Alice encrypts a message using attributes, and Bob tries to decrypt it. If the access policy in Bob's private key fulfills the attributes, he gets access to the message.

The first version of KP-ABE was presented by Goyal *et al.* in 2006 [89]. The access policy of this scheme is monotonic, meaning that the policy only supports ANDs and ORs. To provide more flexibility in policy definition, the authors in [92] present a scheme with non-monotonic access structures supporting AND, OR, and NOT. However, the scheme only admits a limited number of users and a fixed number of attributes. Years later, Lewko *et al.* [93] presented a KP-ABE scheme inspired by that of Goyal *et*

al. but which does not limit the number of users.

The drawback of the schemes in the previous paragraph is that the size of the ciphertext increases with the number of attributes contained in it. Therefore, one of the existing research directions seeks a constant ciphertext size. This is the case of [94], in which its authors present a KP-ABE scheme with a non-monotonic access structure and a constant ciphertext size. However, the authors claim that it is only selectively secure and that fully secure schemes should be pursued.

Seeking to maximize efficiency, Yao *et al.* [95] proposed a KP-ABE scheme for IoT. The authors consider previous schemes, based on bilinear pairings, to be too expensive for IoT devices. Instead, they propose a lightweight scheme based on elliptic curve cryptography, which solves IoT security and privacy issues. However, this scheme was broken in 2019 [96].

It can be concluded, therefore, that there are numerous KP-ABE schemes adapted to different use cases and requirements. As this section has reviewed, some seek size efficiency by generating ciphertexts of constant size. Others seek to speed up access policy operations by using elliptic cryptography or by making their definition more flexible. In general, although the mathematical operations vary, we can summarize the KP-ABE algorithms as the following:

Setup: Using a non-zero random value r, it generates the Master Secret Key (MSK) and the Master Public Key (MPK).

$$Setup(r) \rightarrow (MSK,\ MPK) \tag{2.1}$$

User Private Key Generation: It uses an access policy AP, the MSK and the MPK to generate the private key SK_{KP-ABE}. The key generation is randomized, so private keys generated with the same access policy are different.

$$KeyGen(AP,\ MPK,\ MSK) \rightarrow SK_{KP-ABE} \tag{2.2}$$

Plaintext encryption: This function takes a plaintext PT, an attribute set $\mathbb{A}$ and the MPK. It outputs the ciphertext CT.

$$Enc(PT,\ \mathbb{A},\ MPK) \rightarrow CT \tag{2.3}$$

Ciphertext decryption: This function takes a ciphertext CT, the user's SK_{KP-ABE}, and the system's MPK. PT is obtained if the access policy complies with the attributes; if not, it returns $\perp$.

$$Dec(CT,\ SK_{KP-ABE},\ MPK) \rightarrow PT \tag{2.4}$$

2.3.3 Ciphertext-policy Attribute-Based-Encryption

Bethencourt *et al.* [90] introduced the first CP-ABE scheme in 2007. The scheme allowed for a new mode of encrypted access control to information, shown in Figure 2.6. In CP-ABE, users have keys generated according to attributes, and ciphertext is generated according to access policies. This operation allows users access to information according to their privileges since the attributes of the key must define that user. However, the security of the scheme in [90] was proven under the generic group heuristic. This model has been proven to generate theoretically secure schemes, which are easily broken in practice [97]. Nevertheless, the CP-ABE approach was promising, and numerous CP-ABE schemes have since emerged, some of which are presented below.

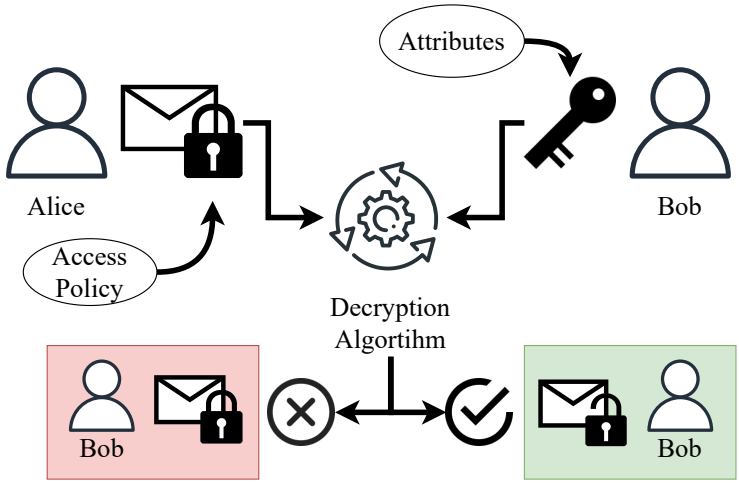


Fig. 2.6 CP-ABE Encryption and Decryption. Alice encrypts a message using an access policy, and Bob tries to decrypt it. If the attributes in Bob's private key fulfill the access policy, he gets access to the message.

Shortly after the presentation of the scheme in [90], a new work [98] enhanced the security proof of [90]. The new scheme uses AND gates, which increase the algorithm's efficiency. However, the definition of access policies using only ANDs is very limiting [99], so the authors of [100] use access trees to overcome this limitation. The proposed scheme uses AND and OR but forces the definition of a maximum depth for the access tree during the setup phase. The authors of [101] improve [100] by achieving better encryption times and reducing the size of the ciphertext and users' private key. The drawback of these schemes is that they use AND gates (which prevents the generation of policies through ORs), or access trees (which limits computational efficiency) to define policies. Waters presented a CP-ABE scheme in 2011 [102] that

uses the Linear Secret Sharing Scheme (LSSS) presented in [103] and can define policies with ANDs and ORs. LSSS achieves the same flexibility as the ones using access trees, while achieving security in the standard model. Most of these solutions define monotonic access policies and thus cannot be written with NOT. In order to reduce this limitation, the authors of [104] present a non-monotonic CP-ABE scheme whose policies and attributes are not bounded. This scheme is based on the KP-ABE scheme presented in [105].

Sometime later, the authors of [106] present a construction that prevents ciphertexts from leaking information about the attributes they are associated with. This scheme claims to reduce the ciphertext size up to 50% compared to similar works like [107] or [108]. The drawback of this scheme is that it restricts the flexibility of policies to be used. This limitation was solved by FAME [109], which overcomes the problems of [106] without compromising performance.

Despite the many different CP-ABE schemes, similar to what happened with KP-ABE, most of them use the same four functions: Setup, Key Generation, Encryption, and Decryption.

Setup: Using a non-zero random value r, it generates the MSK and the MPK.

$$Setup(r) \rightarrow (MSK,\ MPK) \tag{2.5}$$

User Private Key Generation: It uses an attribute set $\mathbb{A}$, the MSK and the MPK to generate the private key SK_{CP-ABE}. The key generation is randomized, so private keys generated with the same attributes are different and cannot be combined.

$$KeyGen(\mathbb{A},\ MPK,\ MSK) \rightarrow SK_{CP-ABE} \tag{2.6}$$

Plaintext encryption: This function takes a plaintext PT, an access policy AP and the MPK. It outputs the ciphertext CT.

$$Enc(PT,\ AP,\ MPK) \rightarrow CT \tag{2.7}$$

Ciphertext decryption: This function takes a ciphertext CT, the user's SK_{CP-ABE}, and the system's MPK. PT is obtained if the attributes in the key comply with the access policy in the ciphertext; if not, it returns $\perp$.

$$Dec(CT,\ SK_{CP-ABE},\ MPK) \rightarrow PT \tag{2.8}$$

Improving the Efficiency of CP-ABE

Despite the functions presented above, it is important to note that CP-ABE, an asymmetric encryption system based on elliptic curve cryptography, is computationally heavy. Therefore, although it is much more widely used than KP-ABE, it is not generally used on its own. Instead, it is combined with symmetric encryption schemes. Thus, the symmetric encryption scheme (much faster and computationally lighter) encrypts the message to be sent. The symmetric key used is then encrypted by CP-ABE. Symmetric keys are very small, 128 bits for the most common cases of Advanced Encryption Standard (AES), for example, and 256 bits for the strictest cases. However, even this combination can become cumbersome for certain deployments.

Looking for ways to make CP-ABE more efficient, the authors of [110] study the complexity of various CP-ABE schemes and propose different strategies to optimize them. They conclude that there is no solution capable of reducing the complexity in every step. Instead, users must evaluate what is best for their system: reducing the complexity of encryption, decryption, or key generation. One way to reduce the computational impact during encryption is to use a technique called symmetric key encapsulation, also known as Ciphertext-Policy Attribute-Based KEM (CP-AB-KEM) in the case of CP-ABE. The encapsulation technique combines CP-AB-KEM with a one-time-pad built using XOR, as shown in Figure 2.7, and its working is explained below.

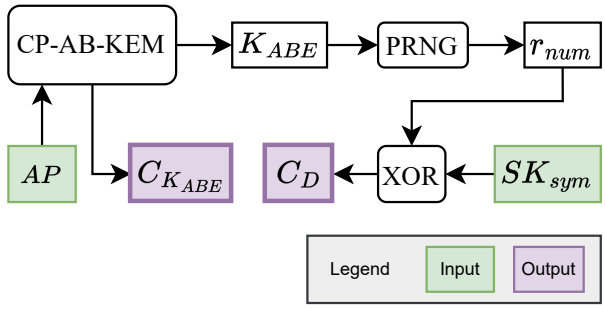


Fig. 2.7 CP-AB-KEM and XOR.

CP-AB-KEM uses an access policy to generate a random number K_{ABE} and its $C_{K_{ABE}}$ encapsulation. K_{ABE} is used as the seed of a Pseudo Random Number Generator (PRNG), which generates a r_{num} of the same size as the symmetric key (SK_{sym}) to be protected. r_{num} is used in a XOR operation to encrypt SK_{sym}, which is more efficient thant directly encrypting SK_{sym} using CP-ABE. Thus, the CP-AB-KEM and the one-time-pad combination is much lighter than the direct use of CP-ABE. Any CP-ABE scheme can be turned into a Key Encapsulation Method (KEM), and many researchers

have developed them. For example, authors in [111] present the KEM version of the CP-ABE scheme presented in [105]-Section 4.

Using CP-ABE to Exchange Industrial Data: The Issue of Policy Update in CP-ABE

Although the concept of fulfilling access policies in order to access information is similar in both ABE schemes, it can be seen how CP-ABE allows the data owners to retain control over who can access the information. In distributed environments like those of Industry 4.0, it is more secure that whoever generates the information is the one who decides the access policy to it. Moreover, combined with symmetric encryption like AES-GCM, CP-ABE provides E2E integrity protection, making it an ideal cipher for Industry 4.0.

With these positive features in mind, authors in [112] incorporate encryption to attribute-based access control by combining AES and the CP-ABE scheme in [90]. As mentioned at the beginning of this subsection, the scheme presented in [90] had only demonstrated security under an idealized model. Furthermore, they focus on protecting the RFID tags attached to the products, which limits its deployment in Industry 4.0 environments that exchange information not directly related to physical products. Authors in [21] combine CP-ABE with Blockchain in international supply chains. They consider a product exchanged through different partners, but to which not all of them can access. Applying CP-ABE to a trade infrastructure must ensure long-lasting encryption at rest. To achieve this, the access policies applied to encrypted data should be updatable while preserving E2E security.

The update of access policies is a known issue in CP-ABE. One of the works identifying it is [113], where authors solve it with a layered model allowing policy updates. However, their model requires knowing the information recipients beforehand. Another approach is [114], in which the authors focus on re-encrypting the symmetric ciphertext while data owners must produce a new ABE ciphertext with every update. However, the computational burden placed on data owners for updates makes this solution inadequate for IIoT devices. Authors in [115] modify the LSSS used to define the access policy embedded in the encrypted message. However, this update must be performed by data owners, which can have an unpredictable computational cost for them. To reduce the burden on data owners, authors in [116] propose a hybrid system in which the ciphertext is sliced, and one of the slices is randomly chosen to be updated. Thus, nothing stops an attacker from collecting slices at different stages and then combining and decrypting them. Finally, the authors of [117] combine CP-ABE with symmetric encryption. However, they apply the same symmetric key to each message and consider that it must be

updated every time an access policy changes. Therefore, their update system is focused on updating the symmetric key, not the CP-ABE ciphertext access policy.

Thus, the problem of updating policies is widely known, and various solutions have been proposed to deal with it. However, it is still considered an open issue, i.e., no de facto solution has been chosen. Ideally, any CP-ABE scheme should natively ensure that policies are updatable. However, this goal is neither possible nor realistic. Instead, a policy update solution should be developed that can be added to any CP-ABE scheme that does not support policy updates natively.

2.3.4 Decentralized Attribute-Based-Encryption

Although CP-ABE is ABE's most widely used mode, it has a major point of failure: the authority. The entire key generation depends on a single authority, which puts the encryption system at risk if it is compromised. For this reason, decentralizing CP-ABE key generation has been an issue that has been explored since the first ABE scheme was introduced.

One of the first attempts to decentralize CP-ABE key generation was presented in [91]. Their scheme allows users to interact with different authorities, but only after consulting the central authority, so it still retains the single point of failure. Therefore, the first scheme with true decentralized key generation was presented by Lewko *et al.* [118]. This scheme only uses a central authority to generate the global parameters of the system, but it can be removed afterwards. To avoid key collusion when keys are generated among several authorities, keys are tied to users' global identifiers. However, one authority is needed for each attribute in the system, which makes it inefficient for systems with a large universe of attributes. A work that breaks the constraint that each authority can only control one attribute is [119], where the authorities control the attributes between all of them. This scheme generates a master key that is distributed among various authorities using secret sharing schemes and users get their keys by contacting a threshold amount of authorities. However, this puts a heavy computational workload on users. Another paper that breaks the limitation of one authority-one attribute is [120]. This scheme is based on [118], but is more efficient and provides a large-unvierse construction. Authors also provide a practical open-source implementation. Thus, the apparent advantages offered by the system make it a potential solution, and it will be explored further in the following sections. Finally, another paper that avoids key collusion by relating users' keys to identities is [121], which presents a scheme with multiple central authorities. The authors propose that the identity-related part of the key is given to users by central authorities, and the attribute-related part is given to users by attribute authorities. However, a posterior work [122] identified the

central authorities as a security bottleneck and considers that the scheme in [121] has poor performance in distributed environments. The centralized security and lack of efficiency defeat the purpose of dCP-ABE.

One of the most popular dCP-ABE schemes for cloud storage, DAC-MACS [123], was proposed a few years later, in 2013. This scheme is intended for cloud storage and offers key revocation with forward and backward security. However, this scheme has been recently broken [124]. Another recently broken scheme is [125], where compromising one authority is enough to attack the whole scheme and cause it to lose Chosen Plaintext Attack (CPA) security [124]. This makes it vulnerable to attacks from both passive and active attackers. Another work that also considers the cloud is [126]. This scheme is intended for big data, and decryption operations are partially delegated to the cloud. This scheme, in addition, offers attribute revocation. However, every time this happens, the authority controlling that attribute must regenerate the *MSK* and *MPK*.

Therefore, there are many different dCP-ABE schemes, and unlike CP-ABE and KP-ABE, they are sufficiently different from each other that it is challenging to present generic functions. One of the security principles that ABE schemes must comply with is to avoid the collusion of users' keys. That is, different users cannot mix their respective private keys to gain access to new information. In single-authority schemes, it is enough to randomize the generation of keys. However, this same technique cannot be used for decentralized schemes since there must be coordination between the different parts of the key. Therefore, there are significant differences between schemes because the way to achieve this property varies. Therefore, below we indicate the functions of schemes that avoid collusion by using the identity of the users, as [118] or [120] do.

Global Setup: This algorithm takes a random security parameter k as input and outputs the system's global parameters, *GP*. These global parameters will be used to set up the different attribute authorities in the system.

$$GlobalSetup(K^k) \rightarrow GP \tag{2.9}$$

Attribute Authorities Setup: Every attribute authority (AA) in the system must run this algorithm. Each authority manages an attribute subset $\mathbb{Z}_i = \{att_1, att_2, \cdots, att_n\} \in AA$ where $1 \leq i \leq m$; such that $n \geq 1$ and m is the total number of authorities. Furthermore, $\mathbb{Z} \subset \mathbb{U}$ where $\mathbb{U}$ is the attribute universe. Thus, the authorities take the *GP* and their identity *AID* as inputs. Next, they use these parameters to generate the public key PK_{AID} and the private key SK_{AID} related to $\mathbb{Z}_i$. Authorities can be set at any time after *GP* generation.

$$AASetup(GP, AID) \rightarrow PK_{AID},\ SK_{AID} \tag{2.10}$$

User Private Key Generation: Users are defined according to their attribute subset, which is a combination of the attribute subsets of different authorities, and thus it is defined as $\mathbb{A} = \mathbb{Z}'_1 \cup \mathbb{Z}'_2 \cup \cdots \cup \mathbb{Z}'_p$ in which $1 \leq p \leq m$, and $\mathbb{A} \subset \mathbb{U}$. Data users ask authorities for their $SK_{UID,\ \mathbb{A}}$. For this, they provide their unique identifier UID to the authorities that fulfill $\mathbb{Z}'_p \subset \mathbb{Z}_i$. In return, the the authorities return $SK_{UID,\ \mathbb{Z}'_p}$. By combining those, users obtain the private key $SK_{UID,\ \mathbb{A}}$.

$$KeyGen(UID,\ AID,\ \mathbb{A},\ SK_{AID},\ GP) \rightarrow SK_{UID,\ \mathbb{A}} \tag{2.11}$$

Plaintext encryption: This function takes a plaintext PT, an access policy AP, the PK_{AID} of the required authorities and the global parameters GP. It outputs the ciphertext CT.

$$Enc(PT,\ AP,\ PK_{AID},\ GP) \rightarrow CT \tag{2.12}$$

Ciphertext decryption: This function takes a ciphertext CT, the user's $SK_{UID,\mathbb{A}}$, and the system's global parameters GP. PT is obtained if the attributes comply with the policy; if not, it returns $\perp$.

$$Dec(CT,\ SK_{UID,\mathbb{A}},\ GP) \rightarrow PT \tag{2.13}$$

2.4 Summary

This section has reviewed the state of the art and introduced the necessary background for the rest of the thesis. Considering that, as mentioned at the beginning, there is no de facto solution to protect data exchange in a value chain, it is necessary to develop one. For this purpose, this thesis considers that the potential solution is affected by the security measures implemented at the plant level and by the chosen solution to preserve confidentiality during the information exchange.

Concerning security at plant level, the chapter reviews the security measures required by institutions such as the IEC or NIST. The state of security at the value chain level has also been analyzed. Recent reports on compromises in the value chain and the data breaches resulting from them give a very worrying picture of the current state of value chain security. Therefore, this chapter analyzes security solutions for value chains, which are many and diverse, confirming the non-existence of a de-facto solution to the problem.

The main method to protect data confidentiality is encryption; thus, its impact in industrial environments has also been studied. It has been observed that industrial environments are reluctant to implement encryption in their communications due to the risk

it poses to availability. Therefore, industrial-level confidentiality-preserving solutions must not jeopardize availability.

With this idea in mind, the different ways of obtaining E2E confidentiality have been studied. We started by reviewing transport layer protocols such as TLS or DTLS and concluded that they are insufficient. By securing the communications channel from one point to the next, hop-by-hop security is obtained, which is insufficient in the presence of brokers. Solutions at the application layer level have been studied to prevent compromised brokers from gaining access to all the information transmitted. At this level, many IoT solutions are based on CoAP, which is secured by applying Object Security for Constrained RESTful Environments (OSCORE). However, OSCORE can only be applied to CoAP and is not easily adaptable to other communication protocols. Moreover, it is oriented to one-to-one communications, and value chains would benefit from one-to-many communications, which reduce the computational power needed by the sender and improve the efficiency of communications.

Finally, different one-to-many cryptographic algorithms have been studied, where the most relevant ones currently belong to the ABE family. In this sense, we review several solutions corresponding to the KP-ABE, CP-ABE, and dCP-ABE modes. The chapter identifies broken schemes or those whose efficiency is inadequate for industrial environments. Still, it has been considered a promising route, and several schemes are studied later in this dissertation.

Chapter 3

Security Architecture, Scenario and Requirements

Addressing security during industrial system design is essential for Industry 4.0. However, security solutions developed for Industry 4.0 are affected by the limitations of IIoT devices introduced in Table 2.1. These include computational capacity limitations, the obligation to comply with deadlines, or the interdependence between devices. In addition to that, due to the long lifetime of industrial equipment, old industrial devices that are not easy to substitute (formally called legacy devices) must also be considered.

Legacy devices can have security vulnerabilities resulting from a lack of security patches [127], limitations in implementing authentication [9], difficulties updating the devices [128], or interoperability problems with newer systems [129]. IIoT and legacy devices may need to interact with one another during industrial operations. Thus, due to the identified device interdependency, limitations of legacy devices eventually affect the entire industrial network. It is crucial to develop an industrial security solution that considers the limitations of legacy devices.

Therefore, the presence of IIoT and legacy devices, the increased connectivity, or device interdependence lead to highly complex industrial networks. Due to this, the IEC 62443, the leading industrial security standard, requires a risk-based approach to security, which balances industrial availability and security. To this end, they identify DiD as the industry's required strategy to deploy security. With this in mind, this chapter answers Question 1 by identifying how to deploy the DiD strategy (Section 3.1), outlining the value chain scenario (Section 3.2) and defining the requirements the data exchange solution should fulfill (Section 3.3). Finally, Section 3.4 summarises the chapter's main ideas.

Part of the research of this chapter has appeared in *"Securing IIoT using Defence-in-Depth: Towards an End-to-End Secure Industry 4.0"* [22] published in the "*Journal of Manufacturing Systems*" in 2020.

3.1 DiD in an Industrial Plant

As Section 2.1.1 introduced, deploying a DiD-based security strategy is crucial in Industry 4.0. According to IEC 62443-4-1 [130], the DiD approach applies layered security measures to limit the damage in case the system is attacked. DiD is an effective security method that relies on the security measures deployed at each layer to address various attack vectors.

The layers of a DiD strategy are created to ensure that if attackers enter the system, the security measures hinder them long enough to be detected [131]. All layers have monitoring and authentication systems, so legacy devices that cannot authenticate users have a certain degree of security: they belong to a network that has only been accessed by authenticated users. Traffic monitoring can also be implemented, reducing the possibility of unauthorized users accessing legacy systems. In general, the traditional hierarchical structure of the industrial domain [132] facilitates the establishment of DiD's layered approach.

3.1.1 Goals

In compliance with [4], security should be addressed during the design, use, maintenance, and removal of industrial systems. This includes hardware, software, and security policies. To this end, a myriad of institutions are participating in regulating Industry 4.0 security. However, the existence of many standards by different institutions can lead to an information avalanche for users [17]. To facilitate their interpretation, this section summarizes the common goals defined by institutions like the IEC [13], the ENISA [1], or the ICS-CERT [14], and that can be used to define a DiD strategy.

- **Availability.** As Table 2.2 introduced, availability is crucial in an industrial environment, and the deployed security solution must not compromise it. Essential functions that guarantee health, safety, environment maintenance, and equipment availability cannot be negatively affected by security measures or emergencies [13].

- **Confidentiality and Integrity.** Concerning integrity violations, a distinction must be made between accidental and deliberate violations. Accidental viola-

tions can be caused by interference in industrial communications, and solutions like Cyclic Redundancy Check (CRC) [133] can detect them. In contrast, if the integrity violation is intentional, attackers can alter the content of packets so that the CRC does not detect it and thus sabotage the system [134]. Intentional attacks can be prevented by using integrity-preserving algorithms like digital signatures. Finally, to ensure data confidentiality, data packets must be encrypted.

- **Logical and Physical Access Control**. The OT network is critical, and its connection to the IT network has to be restricted. This separation is usually achieved using a Demilitarized Zone (DMZ) [14] and reducing traffic to specific and documented services and ports. Combining a DMZ with unidirectional gateways and firewalls restricts the logical access to the OT network and restricts data flow as required in the IEC 62443-4-2 [13]. Biometric systems and smart cards can restrict physical access to critical systems, and a trusted entity should issue access rights following a least-privilege approach [1]. This entity should also keep them up-to-date to reflect the current situation and prevent security breaches.

- **Industrial Control System (ICS) Vulnerability Mitigation.** The long lifetime of industrial devices leads to the discovery of vulnerabilities not contemplated during their design. Companies must install patches provided by the manufacturers to mitigate them, as explained in Section 3.1.1. In case the vulnerability does not have a patch and the equipment has no further updates, a vulnerability assessment should be performed [135], and a hardening process should be considered [136], e.g., using allowlists, reducing application services to the minimum, or restricting users' privileges and roles as much as possible. Periodic security assessments can help identify vulnerabilities so that subsequent threats cannot use them to compromise the system.

- **System Monitorization.** Malfunctioning ICSs and misconfigured services create vulnerabilities in the systems. For example, in the case of wireless devices, an incorrect security configuration can give outsiders an access point to the industrial system [137]. IPSs or Intrusion Detection Systems (IDSs) can detect these intrusions on time and prevent future security breaches [138]. IDS and IPS systems detect abnormal behaviors by comparing the current and expected network status.

- **Critical System Isolation.** The Internet is an untrusted network [14], so ICS and control networks should not be connected to it. However, if necessary, communications should use only secure protocols (e.g., HTTPS instead of HTTP) and pass through a DMZ.

3.1.2 Layer Definition

Once the goals of a DiD strategy have been properly established, the next step is to define the layers. Standards do not establish which layers should be deployed, so approaches differ. However, a formal definition of the layers and their measures benefits industrial environments. To identify the DiD layers suitable for industrial systems, Table 3.1 compares the proposals of different authors.

Table 3.1 Comparison between different DiD strategies.

Reference	Layers	Layer Names	Designed for
Granzer, *et al.* [132]	3	1. Company/Internet 2. Intranet 3. Fieldbus	ICSs
Mavroeidakos, *et al.* [139]	4	1. Perimeter 2. Deceptive 3. Detection 4. Cryptography	Cloud Computing
Kuipers, *et al.* [140]	4	1. Internet and back-ups 2. Corporate 3. Control systems communications 4. Control system operations	ICSs
Zhou, *et al.* [141]	5	1. Physical Protection 2. Perimeter Security 3. Intranet 4. Control System 5. Production Process	SCADA
Nguyen [131]	5	1. Network 2. Enclave boundary 3. Computing environment 4. Identity 5. Application	Microsoft Azure
Knapp, *et al.* [127]	5	1. Physical Layer 2. Network Layer 3. Application Layer 4. Data Integrity 5. Data	Generic

Table 3.1 shows the high variability of DiD strategies. Most feature a minimum of four layers, except [132]. However, this proposal's reduced amount of layers raises questions about its scalability. In the case of [139], the presented strategy is designed for cloud computing. This proposal defines a deception layer and a detection layer. Adapting the layers of this proposal to the layout of an industrial plant may make it lose effectiveness.

In contrast with the previous option, the proposal of [140] is explicitly designed for ICS. However, the proposed layers have been divided considering the system architecture rather than the DiD strategy. Nevertheless, the proposed zones in the architecture are suitable, and [141] uses them as a basis for developing their own SCADA-oriented DiD strategy. The particularity of the last approach is that it places the continuity of the industrial system's operation at the core of the DiD strategy. This is aligned with what is indicated in Section 3.1.2 on the importance of plant availability. However, the system must be able to maintain it while considering that information is the core of the DiD strategy. This leads us to the last two options, out of which we discard [131] for being focused solely on Microsoft Azure. The remaining proposal [127] is based on the traditional DiD layers (Physical, Perimeter, Network, Host, Application, and Data). Since we consider information the core of the DiD strategy, we will follow these traditional layers.

Finally, combining the identified layers with network segmentation is advisable to reach the objectives presented in Section 3.1.1. Segmentation is one of the requirements of IEC 62443-4-2 [13] and increases security by subdividing the logical and physical network.

3.1.3 Security Measures

Network segmentation enhances availability [4], improves the system's reliability [13], and can be physical or logical. Logical segmentation is more flexible and easier to implement, but it may be bypassed and lead to single points of failure, while physical segmentation is more secure but also more complex and expensive [13]. Thus, segmentation techniques should be analyzed case-per-case since there is no universal solution.

The key to successful security frameworks lies in the combination of network segmentation (Figure 3.1) and a DiD approach. Each security zone should consist of assets with similar security needs [142], facilitating monitoring and logical access control. In agreement with the IEC 62443-4-1 [130], the DiD layers should provide additional defense mechanisms by supporting the secure design principles specified in the same standard. The choice of which security measures to implement in each layer is left to the user, e.g., IDSs, IPSs, firewalls, security gateways, or encryption algorithms. Figure

3.2 presents a DiD layered approach following the network segmentation required by IEC 62443-4-2 [13]. The purpose of each layer is explained below.

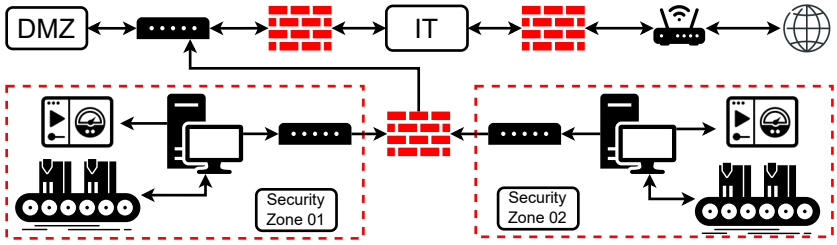


Fig. 3.1 OT network segmentation with two security zones and a DMZ separated by firewalls.

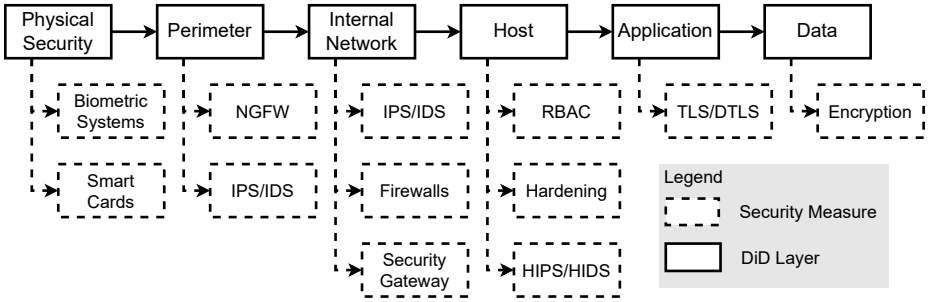


Fig. 3.2 Security layers in DiD with their corresponding security measures.

Layer Nº1: Physical Security

The first layer handles physical security and must be adapted to the organization's particularities. As introduced in Section 3.1.1, smart cards and biometric systems are potential solutions. Smart cards store user-related information for identification purposes. Contactless smart cards are particularly relevant for access in high-traffic areas, as they allow fast and accurate user identification [143]. For more critical areas, biometric systems like facial or iris recognition are more secure than smart cards [144]. Unlike cards, they are also slower and more expensive, but they cannot be easily stolen or duplicated.

It should be noted that although Figure 3.2 presents physical security as a single layer, it is highly complex. Physical security measures are distributed throughout the enterprise and can include various security mechanisms. For example, access rights to different areas can be related to features like time of day or user role. It is, therefore, suitable to deploy a context-dependent access system. This level of security is closely linked to the human factor: users can use USB sticks, cards, and other physical means

to interact with the system. If this access is not controlled or restricted, malicious users could use it to access and infect critical systems [145].

Layer Nº2: Perimeter

Perimeter security protects the OT network by restricting access and filtering unauthorized communications. However, smart manufacturing handles a large volume of traffic using equipment that can be outdated. It has been proven that attackers can overload these communication systems' buffers if suspicious traffic is limited to specific ports and cause accidental Denial of Service (DoS) attacks [127]. Instead, additional measures like Next-Generation Firewalls (NGFWs) are needed. These Firewalls offer deep-packet inspection and IDS/IPS functionalities, providing greater control over what enters and leaves the network, improving security, and facilitating updates [146].

Figure 3.1 shows how NGFWs can be deployed to isolate IT and OT networks while keeping a DMZ as a connection point. The filtering performed by NGFWs should be based on allowlists [41] since this filtering strategy can simplify firewall rules [147]. In addition, ICSs tend to be static [4], which makes allowlist more convenient as they make log analysis more manageable.

Regarding monitoring, it can be active or passive, depending on the particular requirements of the system. IDSs are suitable for analyzing incidents and alerting of system intrusions. On the other hand, if the objective is to stop the intrusion as soon as possible without further analysis, the appropriate approach is to use an IPS [14]. The deployment of a IPS requires a thorough knowledge of network traffic since a IPS that reacts to a false positive can lead to an accidental DoS [148]. Note that firewalls and IDS systems are complementary technologies, and one does not substitute the other.

Finally, this layer also manages remote access [14]. This can be done with secure Virtual Private Networks (VPNs), a temporal user in secured PCs, or by subjecting accessing users to vulnerability scans.

Layer Nº3: Internal Network

So far, the proposed layers protect the system from unauthorized access. In contrast, the following layers protect network resources when attackers have already entered the network, which is one of the vulnerabilities of digital manufacturing systems [149].

The security measures proposed for this layer must be deployed autonomously in each system sub-network. These measures include devices that control inbound and outbound traffic [127], i.e., IDSs/IPSs, firewalls, and security gateways.

The sophistication of cyberattacks is continuously growing, and stopping them requires increasingly complex security measures [150]. The enhanced complexity makes

security measures more efficient when applied to small networks, as they can be designed specifically for the subnetwork.

Layer Nº4: Host

This layer protects each device within a security zone. This individualized protection is of great relevance in OT security because targeted attacks on critical systems can cause significant damage [151]. For example, there have been attacks capable of modifying the firmware of a device to take control of it [152]. Therefore, detecting anomalies such as firmware or device configuration modifications by actively scanning for vulnerabilities is crucial.

The limitations of the system condition the security measures deployed in this layer. Authentication systems like RBAC should be implemented if devices allow them [4]. For legacy devices that cannot implement advanced authentication mechanisms [9], access control to the network they belong to has to be enforced.

Finally, additional security measures at the host level can be considered if the asset supports them, such as Host-Based IDSs (HIDSs) or Host-Based IPSs (HIPSs). These would provide another layer for monitoring and detecting abnormal situations in the host.

Layers Nº5: Application and Data

The application and data layers protect data and services from attacks that the previous layers have not detected. They are the layers most closely related to IT security, and although they are independent, they strongly influence each other.

Application and data layers deal with the worst-case scenario: an attacker who has infiltrated the system and can directly interact with the information generated in it. Thus, the main goal is to protect confidentiality and data integrity. To protect the information, communications between applications must be secured with protocols such as Transport Layer Security (TLS) or Datagram Transport Layer Security (DTLS), combined with application layer data encryption [3].

3.1.4 DiD Layers Summary

Table 3.2 shows how the proposed DiD layers meet the goals defined in Section 3.1.1. It can be seen that several of the goals are met in more than one layer. However, it should be noted that goals are redundant in their approaches since they are achieved by different means. Thus, if an attacker enters the system, they must overcome several security barriers before achieving their goal.

Table 3.2 Goals covered by the proposed security layers. ☐ No ; ■ Yes

		Physical Layer	Perimeter	Internal Network	Host	Application	Data
Availability		■	■	■	■	■	■
Confidentiality and Integrity		☐	☐	☐	☐	■	■
Physical Access Control		■	☐	☐	☐	☐	☐
Logical Access Control	*To Network*	☐	■	■	☐	☐	☐
	To Devices	☐	☐	☐	■	☐	☐
ICS vulnerability Mitigation		■	■	■	■	☐	☐
System Monitorization		☐	■	■	■	☐	☐
ICS Isolation		☐	☐	■	■	☐	☐

Finally, Question 1 discussed the possibility of creating a minimum security architecture. However, the measures defined above are aimed at security within a plant. This thesis approached the security of Industry 4.0, considering the interconnection between companies. Therefore, it is necessary to propose an industrial scenario that includes such connections, as well as the requirements to be met by the secure information exchange platform.

3.2 Industry 4.0 Scenario

As discussed in Chapters 1 and 2, Industry 4.0 security must protect plants and their information exchange. The DiD strategy proposed in the previous section fulfills the former but not the latter. Therefore, to develop a system for securing the data flow between partners in a value chain, it is first necessary to identify what data is exchanged.

Among the activities originally defined by Porter [44] for a value chain, which may require exchanging information between partners, are logistics, manufacturing, or technology development. Logistics is mainly related to the exchange of a physical product but generates digital information (e.g., location or transportation conditions). Regarding manufacturing, it includes all operations carried out to build a product and the maintenance and repair of equipment. In Industry 4.0, the exchange of production data and manufacturing parameters is closely linked to advanced manufacturing and predictive maintenance. Therefore, manufacturing physical goods also generates digital informa-

tion that partners will exchange. Finally, in terms of technology development, digital products should also be considered. It is common for companies that are usually competitors to collaborate in creating digital products or for a manufacturing company to hire another to create a product of this nature. Therefore, technology development is no longer done behind closed doors but also involves the interconnection of companies.

Therefore, it can be concluded from the above that the exchange of information in the value chain is wide and varied. There is no specific data type; rather, it encompasses information of all kinds, from manufacturing parameters to sensor information or digital products. Ideally, a value chain should establish a single information exchange system with E2E confidentiality suitable for all data types. Therefore, the solution must be applied by high-capacity industrial equipment, ordinary computers in IT networks, and IIoT devices. In addition, although most of the participants in the value chain are data consumers and data generators, some are only data generators, and others are only consumers.

Developing an information exchange system capable of encompassing this variety and complexity of data and roles is crucial since, even though information exchange between partners is currently limited [48], collaboration improves the performance of the entire chain [47].

3.3 Industry 4.0 E2E Security Requirements

Although some of the information handled in Industry 4.0 is generated by equipment with high performance, many other data are generated by IIoT devices. Therefore, this thesis considers IIoT devices as one limiting factor in deploying a security solution in Industry 4.0. With those IIoT devices in mind, this section defines the requirements that must be met to answer Questions 2, 3, and 4.

Regarding Question 2, it pondered which ABE algorithm was suitable to be deployed in the industry. As Chapter 2 has already shown, ABE is one of the most widely developed one-to-many cryptographic families nowadays, with many schemes available in the literature. However, not all schemes offer the same security and efficiency guarantees. Moreover, not all of them have practical implementations. Therefore, the scheme must have an implementation that still maintains the scheme's security and efficiency. As a consequence, we define requirement **R1**, which combines the need for the solution to have a robust and efficient cryptographic scheme.

*"**R1.** The solution shall be based on robust and efficient cryptographic ciphers"*

Question 3 wonders whether it is possible to deploy a flexible ABE-based information-exchange system in value chains that guarantees E2E data confidentiality and integrity.

Thus, we define requirements **R2** and **R3**.

*"**R2.** The solution shall provide E2E confidentiality and integrity for data management in industrial environments."*

*"**R3.** The solution shall be flexible to changes in data access rights."*

Finally, question 4 raised the issue of attribute validation in ABE. Similar to identity spoofing, ABE systems are vulnerable to attribute spoofing. Therefore, the problem of attackers taking advantage of the system by claiming attributes that do not belong to them and thus gaining access to sensitive data needs to be addressed. With this in mind, we formulate the following:

*"**R4.** The solution shall provide authentication and data access rights validation."*

3.4 Summary

Making industrial systems secure by design has become essential in smart manufacturing. DiD strategies tackle the complexity of this problem by providing multiple defense layers, each focusing on a particular set of threats.

This chapter defines the goals a DiD-based security strategy should fulfill according to institutions like the IEC, ENISA, or the ICS-CERT, and reviews the literature to define the DiD goals. Once goals have been defined, the chapter analyzes different DiD approaches to define the layers for Industry 4.0 security strategy.

The layer definition for DiD is accompanied by the security measures to be deployed in each of them. Furthermore, the chapter also verifies the fulfillment of the DiD goals identified at the beginning. Finally, the chapter introduces the Industry 4.0 scenario that will be followed in the next chapters and defines their requirements.

Chapter 4

ABE Libraries Analysis and Evaluation

Question 1 wondered about the existence of ABE ciphers suitable for deployment in industrial production environments. This question led to requirement **R1**, calling for a solution based on robust and efficient schemes. Chapter 2 presented many ABE schemes, and although some are broken and can be discarded, others require a more in-depth study. We only consider those ABE schemes with practical implementations for this analysis.

There is a wide variety of ABE schemes and libraries that implement them. To use this family of encryption algorithms in an industrial environment, it is necessary to identify an implementation capable of balancing efficiency and security. However, finding the balance is difficult, especially when users lack the necessary knowledge. To simplify this task, the authors of [153] propose eight metrics for users to choose the most suitable cryptographic libraries for their experiments. In a more recent work [154], other authors present up to fifteen metrics to make the choice. However, all these metrics require a thorough analysis of the libraries by the users who have to use them.

As already introduced by [155], library usability and the human factor directly influence the security of developments implemented with cryptographic libraries. The authors of [156] delve deeper into the problem, identifying up to 16 potential issues. These problems include some already presented by the authors of [155], such as lack of documentation and examples, and add new ones like lack of user knowledge (e.g., not knowing which methods are more suitable for their use cases, how to use the algorithms or lack of cryptographic knowledge).

Following a practical approach, this chapter analyzes eleven ABE libraries, studying their main characteristics and subjecting them to a qualitative and quantitative evaluation. The objective of the qualitative evaluation is to summarize some of the characteristics identified in the metrics of the works cited above. The quantitative evaluation is performed experimentally, so this practical analysis allows us to answer the question of

which libraries are suitable for industrial deployment. Libraries' efficiency is directly influenced by the architecture [157] or the constrained nature of the device [158] on which they are deployed.

We can therefore consider the experimental evaluation of ABE schemes of great interest since implementations are not always optimal. Third-party dependencies, mathematical libraries, implementation errors, or programming language affect the runtime and security of the library. In addition, mathematical dependencies also affect the security of the implementations. Therefore, current formal analyses of cryptographic schemes are insufficient, and this more practical approach is required. To this end, authors in [159] presented the first survey of ABE libraries in 2016, examining the implementation of fourteen libraries. However, most of the libraries studied in the paper have been deprecated or discontinued. In addition, although the authors conducted experiments to support their claims, they only performed experiments on one of the considered libraries.

The overall structure of the chapter takes the form of seven sections, including one that summarizes the chapter's main ideas. Section 4.1 explains the technical requirements defined to select the library that fulfills the requirement **R.1** presented in Section 3.4. Section 4.2 presents the ABE libraries' timeline and shows their evolution. Libraries identified as potential options for industry are qualitatively evaluated in Section 4.3. The experimental evaluation is defined in Section 4.4, and the results are presented in Section 4.5. Section 4.6 presents the discussion that answers Question 2 about which ABE ciphers are suitable for deployment in an industrial environment. Finally, Section 4.7 summarizes the chapter.

The research conducted to write this chapter has been included in two papers:

- *"All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of CP-ABE Libraries"* [23] presented at the *URSI 2022*.
- *"Too Many Options: A Survey of ABE Libraries for Developers"* [24] that has been to submitted to Computer Networks and published as a preprint in *arXiv* in 2022.

4.1 Requirements

This chapter provides two evaluations for ABE libraries: qualitative and experimental. Every library is qualitatively evaluated, but some are dismissed after the assessment and are not experimentally analyzed. In order to define the library selection methodol-

ogy, we define some Technical Requirements (TRs) that validate the fulfillment of **R.1** established in Section 3.3.

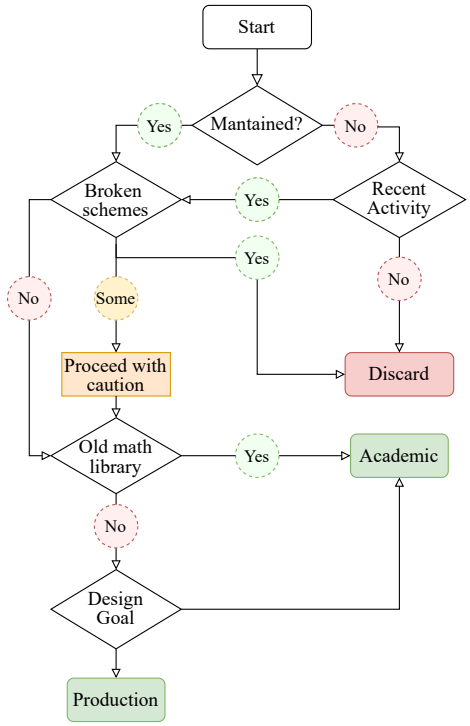


Fig. 4.1 Library analysis process.

TR 1.1. Libraries shall have active maintenance: A maintained library provides updates and upgrades, which are crucial for long-term security. Therefore, it is considered a significant aspect in the analysis and choice of libraries. We consider a library to be maintained when its main fork has had any update or upgrade in the last year as of November 2022.

TR 1.2. Libraries' repositories shall have recent activity: Activity is a sign that the library is in use, which encourages its maintenance. Many improvements and the inclusion of new features are often developed or suggested by users. We consider that there is recent activity when the main fork of the library has been maintained or if any branch has had activity in the last year as of November 2022.

TR 1.3. Libraries shall provide unbroken schemes: Some ABE schemes have been successfully attacked [124], although this is not always clearly disclosed in the implementations. We consider that a library should not implement any of these schemes and that it should provide a secure alternative if it does. Otherwise, it is considered a

vulnerable library and should be discarded.

TR 1.4. Libraries' mathematical dependencies shall be suitable for production: ABE schemes are based on elliptic curve cryptography but do not require specific curves; instead, they are chosen by the developers. Therefore, the chosen mathematical library is crucial for the security and efficiency of ABE. Some mathematical libraries only support obsolete curves and should only be used in academic or experimental settings.

TR 1.5. Libraries shall be designed for production: Not all libraries are intended or appropriate for production. Understanding the use of each library is critical to deploying a secure system. Libraries not designed for production environments should not be considered in these scenarios. Libraries flagged by their developers as unsuitable for production should only be used in academic environments.

All the libraries are studied according to these five technical requirements. Then, following the process shown in Figure 4.1, we choose the libraries to be experimentally analyzed. We consider that libraries implementing outdated curves are unsuitable for the production environment, regardless of their design goal.

4.2 ABE Libraries Timeline and Relationships

The existence of ABE libraries is essential for deploying this cryptographic algorithm beyond academia. However, these developments are notably scarcer than existing ABE proposals. Furthermore, many implementations are related to each other or share developers, so there is a lot of code reuse. There is also a wide variation in the life span of some projects: some libraries have been active for ten years, whereas some newer ones, after their first version, are abandoned shortly after. Therefore, it is interesting to make an initial assessment of when the libraries were created, why, which are the relationships between them, and how long they have been active.

When Bethencourt *et al.* presented CP-ABE [90], they included the *cpabe toolkit* [160]. This library was, until 2011, the only library offering ABE schemes. However, in 2011 *libfenc* [161] appeared. This project reused part of the *cpabe toolkit* code but extended the functions and schemes of the library. These relationships are shown in Figure 4.2 by solid line arrows.

The same year that *libfenc* appeared, the first version of *Charm* [162] was released. Several *libfenc* developers have also participated in the development and maintenance of *Charm*, which is still being maintained and updated. On the other hand, it is also interesting to mention that some of the *cpabe toolkit* and *Charm* developers worked together and released a new library for ABE in 2019, called *OpenABE* [163]. This

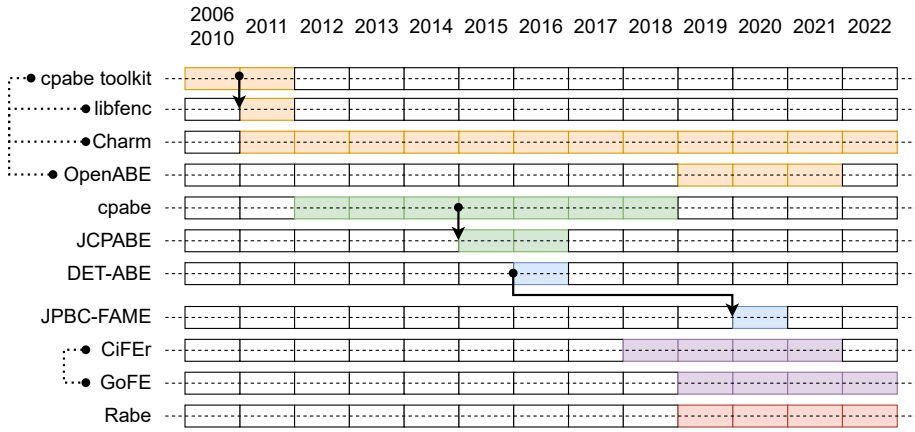


Fig. 4.2 Library timeline. Dashed lines represent libraries that share developers. Continuous line arrows represent libraries that reuse code. Colors show libraries with some relationship between them (code or developers).

relationship of developers is shown in Figure 4.2 by dashed lines next to the name of the libraries.

Although not every proposed ABE scheme includes a practical implementation, one exception is BDABE [164]. This scheme has been developed by Fraunhofer and includes a practical implementation in Rust, the Rabe library [165]. Moreover, *Rabe* implements several other ABE schemes, not only BDABE, and it is still maintained. Despite these advances, there is still a need for a production-grade library. For this reason, part of the results of the FENTEC[1] project, which aimed to develop functional encryption systems, resulted in two ABE libraries, *GoFE* [166] and *CiFEr* [167], between 2018 and 2019. Both implement the same ABE schemes, but *GoFE* is implemented in Go and *CiFEr* in C.

4.3 Qualitative Evaluation

This section uses the criteria defined in the previous section to evaluate 11 libraries. The information gathered about these libraries is presented in Table 4.1, which is analyzed according to the process shown in Figure 4.1. To better understand these sections, Annex A summarises the terminology used in this chapter.

Note that the table has also included information on whether there is an active community using the libraries. An active community can provide technical support, even if

[1] https://fentec.eu/

Table 4.1 Comparison of libraries. Part I.

Name	Mantained [1] [2]	Active Community [1] [2]	Recent Activity [1] [2]	Broken Schemes [1] [2] [3]	Math Library [1] [2] [3] [4] [5] [6] [7] [8] [9]	Design Goal [1] [2]
	1. Yes 2. No	1. Yes 2. No	1. Yes 2. No	1. Yes 2. No 3. Some	1. Miracl [168] 2. libsodium [170] 3. bn-256 [172] 4. go-crypto 5. Rabe-bn [175] 6. Relic [169] 7. OpenSSL [171] 8. jPBC [173] 9. PBC [174]	1. Production 2. Research
libfenc [161]	□ ■	□ ■	□ ■	□ ■ □	□ □ □ □ □ □ □ □ ■	□ ■
cpabe-toolkit [160]	□ ■	□ ■	□ ■	□ ■ □	□ □ □ □ □ □ □ □ ■	□ ■
OpenABE [163]	□ ■	■ □	■ □	□ ■ □	□ □ □ □ □ ■ ■ □ □	■ □
JPBC-FAME [176]	□ ■	□ ■	□ ■	□ ■ □	□ □ □ □ □ □ □ ■ □	□ ■
cp-abe [177]	□ ■	□ ■	□ ■	□ ■ □	□ □ □ □ □ □ □ ■ □	□ ■
DET-ABE [178]	□ ■	□ ■	□ ■	□ ■ □	□ □ □ □ □ □ □ ■ □	□ ■
Rabe [165]	■ □	■ □	■ □	□ □ ■	□ □ □ □ ■ □ □ □ □	■ □
JCPABE [179]	□ ■	□ ■	□ ■	□ ■ □	□ □ □ □ □ □ □ ■ □	□ ■
Charm [162]	■ □	■ □	■ □	□ □ ■	■ □ □ □ □ ■ ■ □ ■	■ □
GoFE [166]	■ □	■ □	■ □	□ ■ □	□ □ ■ ■ □ □ □ □ □	□ ■
CiFEr [167]	□ ■	■ □	■ □	□ ■ □	■ ■ □ □ □ □ □ □ □	□ ■

the library is no longer maintained. The information contained in the table is discussed in the following subsections.

4.3.1 Generic Features

Table 4.1 shows that only three libraries continue to be maintained by the original developers in the main fork: *Rabe*, *Charm*, and *GoFE*. Of the remainder, it should be noted that most had no activity in the main fork in almost two years. The exceptions are *OpenABE* and *CiFEr*. *OpenABE* was last updated in the main fork on January 2021 and *CiFEr* on February 2021. However, both of them have had recent activity in new forks. In the case of *OpenABE* forks[2,3], they solve installation issues and updates dependencies like OpenSSL or Bison. In the case of the new *CiFEr* fork[4], it has received minor updates. It should also be noted that *CiFEr* is the twin of *GoFE* but implemented in C instead of Golang. In other words, although the main branch of *CiFEr* has no activity, the project is still going on. Therefore, this article considers that all five libraries, i.e., *OpenABE*, *Rabe*, *Charm*, *GoFE*, and *CiFEr*, have had recent activity.

Regarding community activity, *Rabe* and *Charm* are the most active, although in different ways. *Rabe* developers respond and resolve open issues in the repository and update the library with user-requested features[5]. In contrast, *Charm* is the most widely

[2] https://github.com/StefanoBerlato/openabe
[3] https://github.com/ivario123/openabe
[4] https://github.com/swanhong/CiFEr/tree/master
[5] https://github.com/Fraunhofer-AISEC/rabe/issues/9

used framework for cryptographic prototyping, and as such, it has an extensive community of users. An active community favors problem resolution, which makes it an interesting feature to be considered. However, we have not considered it that critical; therefore, its absence does not imply discarding the library.

4.3.2 Security Features

In this section, we analyze those features directly affecting the libraries' security: the chosen mathematical libraries, the implemented ABE schemes, and the AES schemes underneath. Regarding the mathematical libraries used, as Table 4.1 showed, some libraries use PBC, which supports obsolete elliptic curves that are currently not considered secure enough for production [110]. Therefore, we consider libraries using PBC only suitable for research. Java libraries use jPBC, the Java implementation of the original PBC, and therefore implements the same elliptic curves as PBC. Libraries like *Charm* use PBC but can be compiled with Miracl or Relic. Relic, PBC, and Miracl are pairing-based cryptographic libraries. However, Relic offers more efficient pairing constructions and faster implementations than PBC, and Miracl is newer and available in seven programming languages.

Miracl is also the mathematical library used by *CiFEr*, albeit with a crucial nuance compared to *Charm*: *Charm* uses the maintained branch of Miracl. In contrast, *CiFEr* uses Miracl-AMCL, a non-maintained branch of Miracl. Moreover, *CiFEr* implements a reduced version of Miracl-AMCL, using the BN-254 curve for 64 bits. As a result, *CiFEr* can only be implemented in a 64-bit architecture, limiting its deployability. In contrast, *Charm* imposes no restrictions on the curve to use from Miracl-Core. With the architecture limitation of *CiFEr*, *OpenABE* becomes the only library written in C++ with no architecture limitation. Note that a C++ library is especially relevant for IoT devices with minimal computational power.

Regarding "Design Goal," we refer to research or production quality. We consider cryptographic libraries using obsolete curves suitable for research and academic purposes but not production. Furthermore, some developers flag their libraries as unsuitable for production, either because they have not been adequately tested or are still in an early stage of development. Thus, *OpenABE*, *Rabe*, and *Charm* are the libraries currently suitable for production, while *GoFE* should still be limited to academic and research environments. However, as pointed out earlier, *Charm* should be compiled with Miracl or Relic, not PBC. With the four main libraries identified, we examine their implementation in Table 4.2. To enhance the results' readability, these four libraries have also been highlighted in bold in Table 4.1.

In terms of supported schemes, *Charm* and *Rabe* support YCT14, a KP-ABE scheme

Table 4.2 Comparison of libraries. Part II.

Name	Docs.	Language	KP-ABE	CP-ABE	dCP-ABE
				1. BSW07 [90]	1. MKE08 [91]
		1. C++	1. GPSW06 [89]	2. W11 [102]	2. LW11 [118]
	1. Yes	2. Rust	2. LSW10 [93]	3. YAHK14 [104]	3. DAC-MACS [123] ∗
	2. No	3. Go	3. YCT14 [95] ∗	4. CGW15 [106]	4. YJ14 [125] ∗
		4. Python	4. FAME [109]	5. TimePRE [180]	5. RW15 [120]
				6. FAME [109]	6. BDABE [164]
OpenABE [163]	■ □	■ □ □ □ □	■ □ □ □	□ ■ □ □ □ □	□ □ □ □ □ □
Rabe [165]	■ □	□ ■ □ □ □	□ ■ ■ ■	■ □ □ □ □ ■	■ ■ □ □ □ ■
Charm [162]	■ □	□ □ □ □ ■	□ ■ ■ □	■ ■ ■ ■ ■ ■	□ ■ ■ ■ ■ □
GoFE [166]	■ □	□ □ ■ □	■ □ □ □	□ □ □ □ □ ■	□ ■ □ □ □ □

∗Broken scheme

proposed in [95], which was broken in 2019 [96]. In addition, *Charm* also implements some vulnerable dCP-ABE schemes: YJ14 [125] and DAC-MACs [123], both broken in [124]. Broken schemes are insecure and should never be used. However, both *Charm* and *Rabe* support other ABE schemes, so there is no need to discard them. Instead, the libraries should be used cautiously, and vulnerable schemes should not be implemented. Another scheme that should be carefully considered is BDABE [164]. This is the only ABE scheme designed to work with Blockchain among all those offered by the libraries. Thus, as the authors of the scheme explain, the scheme's efficiency will be affected by the deployed Blockchain solution.

As mentioned earlier, ABE schemes are computationally heavy and are often used in hybrid encryption. Hybrid encryption uses a symmetric scheme (usually AES) to encrypt the message and an ABE scheme to encrypt the symmetric key. Therefore, we also study the implemented AES schemes to analyze the libraries properly. Table 4.3 shows the result of this evaluation.

Table 4.3 Implemented symmetric ciphers.

Library	AES-CBC	AES-GCM
OpenABE [163]	□	■
Charm [162]	■	□
Rabe [165]	□	■
GoFE [166]	■	□

AES Cipher-Block Chaining (AES-CBC) and AES-GCM are secure symmetric ciphers, but AES-GCM provides authenticated encryption. Using AES-GCM provides

confidentiality and guarantees the integrity of the encrypted message. This protection does not stop attackers from violating the integrity of the encrypted message but allows the receiver to detect tampering. Information exchange systems benefit from data integrity protection and should favor the use of AES-GCM. As can be seen from the libraries presented in Table 4.3, the only ones implementing AES-GCM are *OpenABE* and *Rabe*. It should be noted that *Rabe* has implemented AES-GCM in *rabe-0.3.1*[6]. Previous versions used the low-level AES block cipher function and should only be applied in academic and research environments.

4.4 Experimental Evaluation Definition

To properly deploy ABE encryption schemes, their efficiency and complexity cannot negatively impact the system. Since several factors influence the efficiency of a cryptographic scheme, the authors of [110] study the mathematical complexity of various ABE schemes and conclude that there is no solution capable of reducing the complexity of ABE schemes in every step. Instead, they propose several optimization strategies depending on which operation is to be enhanced: key generation, encryption, or decryption. The optimizations they propose, however, rely on mathematical modifications of the schemes and require expert knowledge.

Modifying the mathematical code of cryptographic libraries can lead to errors and create security vulnerabilities. It is hence unfeasible to implement the proposals of [110] in other libraries without adequate cryptographic and mathematical knowledge. Therefore, this section experimentally analyzes the four main libraries identified as having potential for industrial and academic use. As mentioned above, one of the challenges when implementing ABE schemes is the lack of knowledge about the performance of the libraries. Therefore, this section studies and compares how libraries behave on devices with different computational capabilities. This behavior is quantified by measuring the time each library scheme requires to perform the basic operations in ABE: encryption, decryption, key generation, and, in dCP-ABE, authority setup.

4.4.1 Experiment Definition

Our experiment relies on the different mode of ABE schemes. The classification of CP-ABE, KP-ABE, and dCP-ABE schemes offered by the libraries is shown in Table 4.2. dCP-ABE schemes have similar behavior to the conventional CP-ABE schemes, but the key generation is distributed among several key-generating authorities.

[6]https://docs.rs/rabe/0.3.1/rabe/index.html

Comparing the performance of different ABE schemes illustrates how practical aspects of the implementation, such as the math library or implementation language, can affect ABE schemes. Although the performance variation is expected, it should be quantified. Therefore, this section performs timing tests for each scheme in the libraries. To this end, we group the schemes according to whether they are CP-ABE, KP-ABE, or dCP-ABE and study how long each scheme takes to perform the key generation, encryption, and decryption operations. Finally, for the sake of comparability, note that the time related to Blockchain operations in the case of BDABE is not taken into account. This metric is highly variable and strongly dependent on the implemented technology, so it would not be comparable to the rest of the schemes.

4.4.2 Testbed Setup

The testbed consists of two Raspberry Pi (RPI) devices, i.e., a RPI0 and a RPI4. The RPI0 runs Raspbian Stretch, has 512MB of RAM, a single-core ARMv6, and Wi-Fi. We consider it a good representation of an IIoT device. The RPI4 has an ARMv8 processor, 8GB of RAM, and runs 32-bit Ubuntu Server TLS. This second device is considered representative of industrial devices with high processing capabilities. This setup provides a representative insight into the execution time of the libraries on IIoT devices, as well as the libraries' portability.

As explained in the discussion of AES schemes in Section 4.3.2, ABE schemes are paired with symmetric ciphers. Therefore, in this section's experiments, AES encrypts a 43-byte plaintext, and ABE encrypts the 256-bit AES key. The reason behind the small-sized plaintext is that this chapter evaluates the computational efficiency of different ABE implementations, so a small plaintext has been chosen to make the computational burden of ABE more significant than that of AES. This provides a representative view of the efficiency of each library. For the mentioned time measurements, experiments have been carried out by benchmarking, which depends on the library and its implementation language:

- OpenABE: The library offers its proprietary benchmark.
- Rabe: Criterion for Rust.
- Charm: Timeit for Python.
- GoFE: Golang benchmarking.

It is worth mentioning that *Rabe* runs on Rust nightly, the unstable API of Rust. Therefore, the native Rust benchmark could be used for experiments. However, this is

an unstable function with highly variable results and it is at risk of being deprecated. In addition, this function returns the runtime value in nanoseconds. In complex schemes, the execution time is greater than the maximum value that can be returned by the *u32* type variable used by the native benchmark. Therefore, the benchmark's output overflows, returning a value that does not capture the function's actual processing time. Criterion, which is statistically more reliable, has been used instead.

4.5 Experimental Evaluation Results

To properly discuss the results, we present Table 4.4, which summarizes the properties of each scheme. Every ABE scheme always has a linear growth for key generation, and most ABE schemes also have a linear time growth in encryption and decryption. However, some offer unique features, such as constant time for encryption and decryption.

Table 4.4 Scheme features.

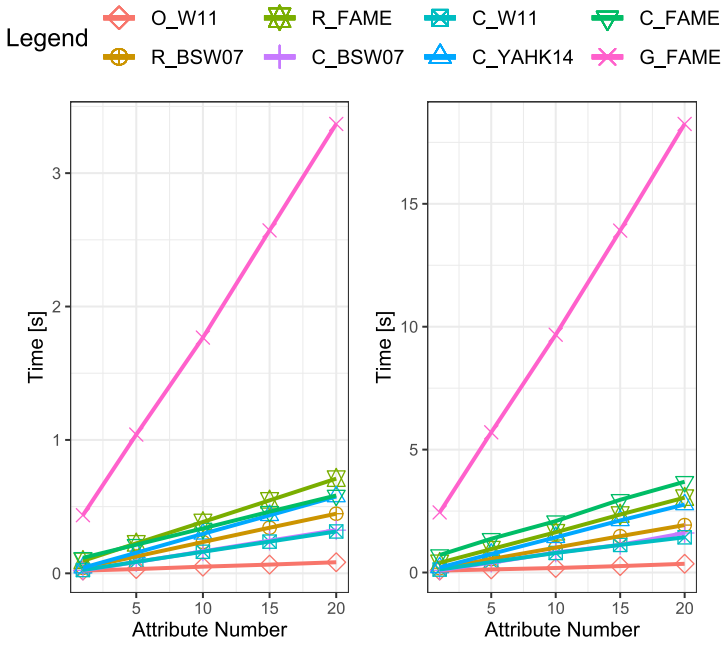
Scheme	Mode [1] [2] [3]	KeyGen [1] [2]	Enc. [1] [2]	Dec. [1] [2]
	1. KP-ABE 2. CP-ABE 3. dCP-ABE	1. Linear 2. Constant	1. Linear 2. Constant	1. Linear 2. Constant
GPSW06 [89]	■ □ □	■ □	■ □	■ □
LSW10 [93]	■ □ □	■ □	■ □	■ □
YCT14 [95]	■ □ □	■ □	■ □	■ □
FAME [109]	■ □ □	■ □	■ □	□ ■
BSW07 [90]	□ ■ □	■ □	■ □	■ □
W11 [102]	□ ■ □	■ □	■ □	■ □
YAHK14 [104]	□ ■ □	■ □	■ □	■ □
FAME [109]	□ ■ □	■ □	■ □	□ ■
MKE08 [91]	□ □ ■	■ □	□ ■	□ ■
LW11 [118]	□ □ ■	■ □	■ □	■ □
RW15 [120]	□ □ ■	■ □	■ □	■ □
BDABE [164]	□ □ ■	■ □	□ ■	□ ■

4.5.1 CP-ABE

For this experiment, we analyze the time evolution of the different operations (i.e., keygen, encryption, and decryption) for a different number of attributes. This way, we

quantify the variance of the time requirements according to the attributes contained in the Access Policy (AP) or CP-ABE Private Key (SK_{ABE}).

Figure 4.3 shows the time in seconds required to generate users' keys for a different number of attributes. The required time grows linearly with the number of attributes used for the SK_{ABE} generation. G_FAME is the slowest implementation for key generation. G_FAME takes 3.36s on RPI4 and 18s on RPI0 to generate a 20-attribute key. In contrast, R_FAME takes 0.71s to generate the same key in RPI4 and 3s in RPI0. In other words, G_FAME takes 374% longer than R_FAME in RPI4; and in the case of RPI0, the generation is 500% slower.



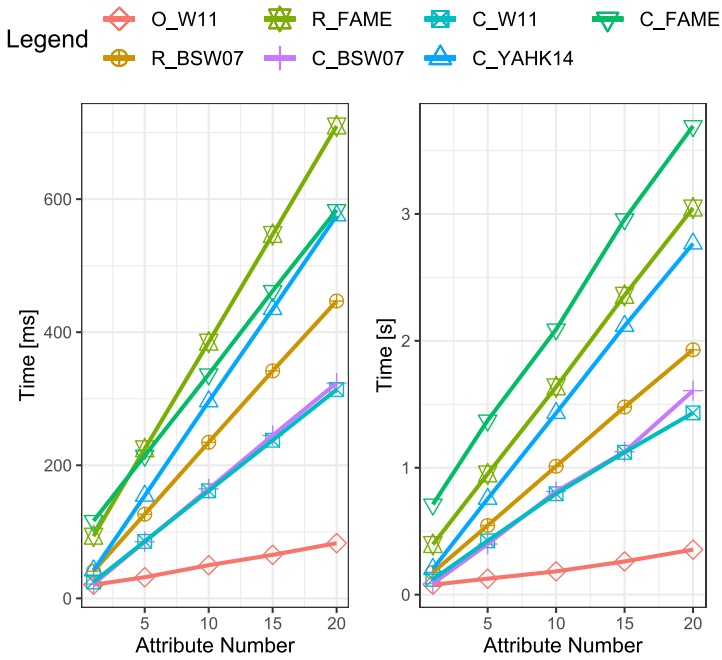
(a) Results in a RPI4 in seconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.3 CP-ABE key generation time.

The difference in time shown in Figure 4.3 is related to *GoFE* using crypt-go, Golang's cryptographic libraries, which are less efficient than consolidated libraries like OpenSSL. There are forks of Go that add additional security features[7], but experiments for this chapter have been carried out with the official Go distribution. This distribution is the most commonly used and, therefore, the one whose results are most meaningful. Meanwhile, *Charm* and *OpenABE* use well-established and consolidated math libraries like PBC, Miracl, or Relic. These libraries have had numerous releases and patches

[7]https://github.com/cloudflare/go

that have added functionality and increased efficiency. We can see that among the three other libraries, *Rabe* is the slowest one taking 0.88s compared to the 0.083s of *OpenABE*. *Charm* lies in-between, taking 0.58s. Since programming languages are designed for different purposes, their optimization level impacts performance. However, the impact of mathematical dependencies is also noticeable: Figure 4.3 shows that a library written in an interpreted language (i.e., Python) is faster than one written in a compiled language (e.g., Go or Rust). The growth rate of *GoFE* makes it challenging to visualize the rest of the results. Therefore, we provide Figure 4.4 for a better view.



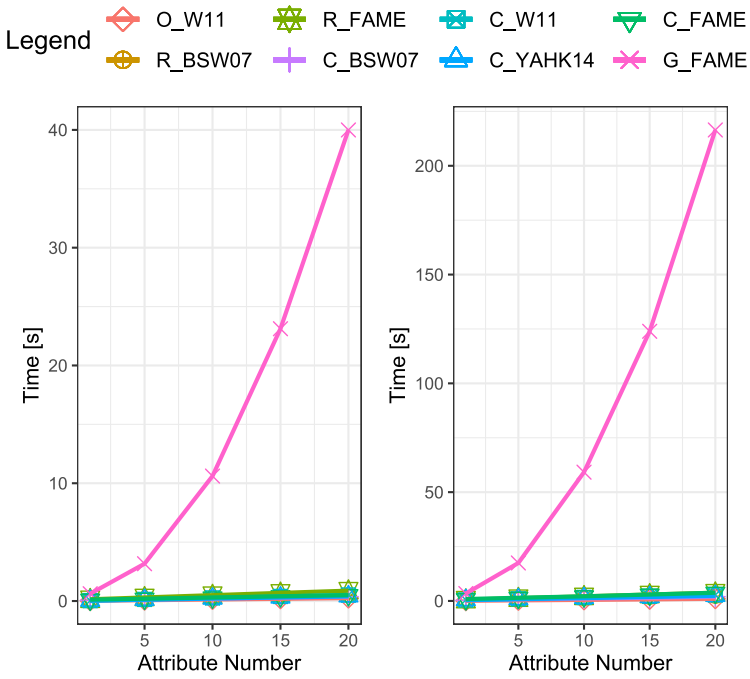
(a) Results in a RPI4 in milliseconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.4 CP-ABE key generation time without GoFE-FAME.

Once we establish G_FAME as the slowest scheme in Figure 4.3, it is interesting to see which one is the second slowest scheme in Figure 4.4. In RPI4, the second slowest scheme is R_FAME at 0.71s. Meanwhile, in the RPI0, the second slowest scheme is Charm-FAME (C_FAME) at 3.69s. This is because RPI0 favors compiled languages like Rust, and it is more affected by interpreted languages like Python.

Figure 4.4 also shows that the fastest scheme for key generation is OpenABE-Waters11 (O_W11), which takes 83ms for twenty attributes in the RPI4 and 355ms in the RPI0. The next best scheme is Charm-Waters11 (C_W11), with a notable differ-

ence from O_W11. In fact, O_W11 is approximately 74% faster than C_W11 in both RPI4 and RPI0. Another interesting finding is that Charm-BSW07 (C_BSW07) takes a similar time as C_W11 in both devices. In the RPI4, C_BSW07 takes 327ms for twenty attributes, barely 13ms longer than C_W11. In the case of RPI0, C_BSW07 takes 1.60s and C_W11 1.43s. Thus, the difference between C_BSW07 and C_W11 is 4.375% in RPI4 and 11.22% in RPI0.

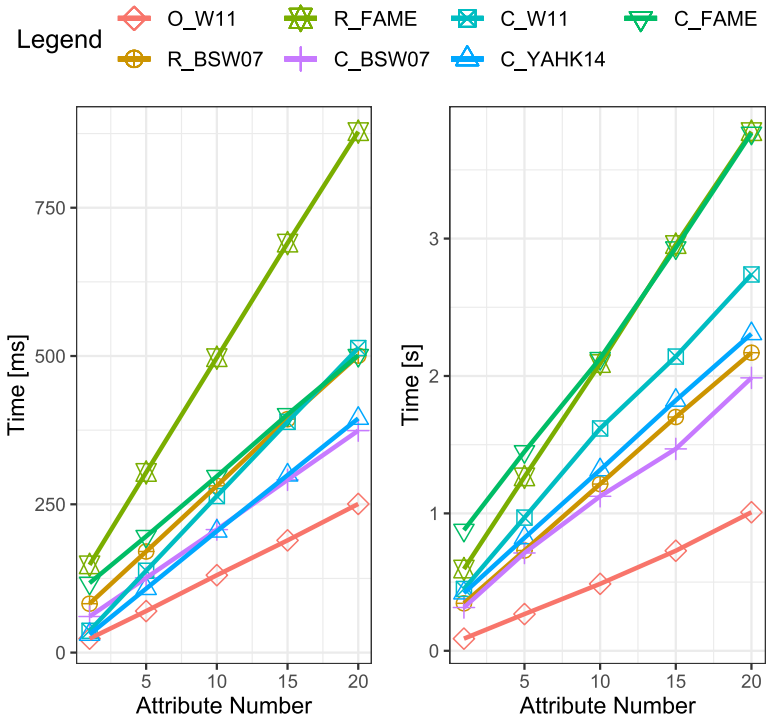


(a) Results in a RPI4 in seconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.5 CP-ABE encryption time.

Key generation is critical, and its running time should be measured. However, this operation is performed at a much lower frequency than encryption and decryption. We present Figure 4.5 to study the libraries' encryption time. It shows that encryption time of G_FAME grows exponentially, taking 40 seconds for twenty attributes in the RPI4 and 3 minutes for the RPI0. Meanwhile, the rest of the library-scheme combinations take less than a second in the RPI4 and less than four seconds in RPI0. Furthermore, FAME encryption time grows linearly with the complexity of the access policy, which G_FAME fulfills in neither device. Because of the long time taken by *GoFE* to encrypt information, we present Figure 4.6 to be able to visualize the rest of the results.

We can see in Figure 4.6a that, after G_FAME, R_FAME is the slowest encryption



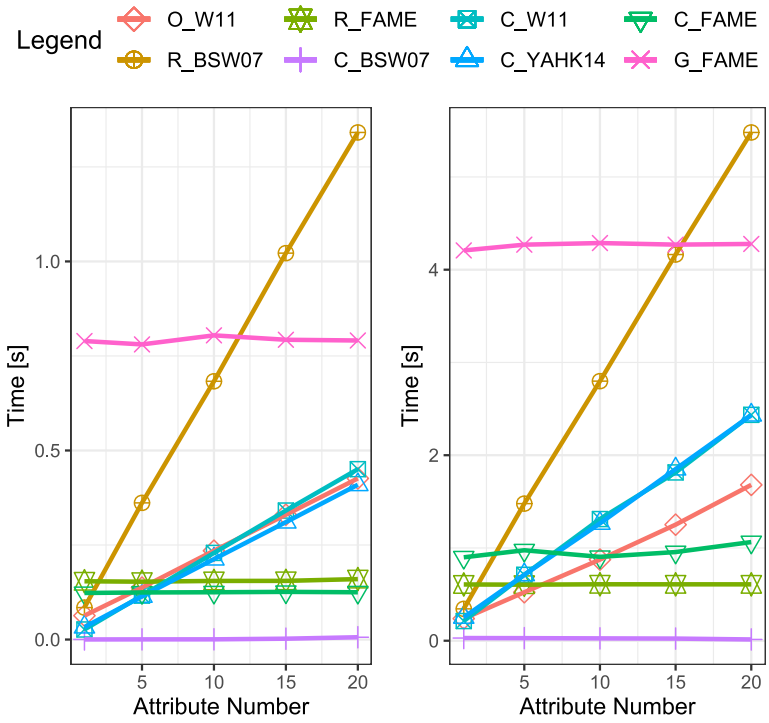
(a) Results in a RPI4 in milliseconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.6 CP-ABE encryption time without G_FAME.

scheme for every case in the RPI4. At 15 attributes, it holds a notable difference from the next-slowest one, C_FAME. For this case, R_FAME takes 690.36ms and C_FAME 400ms, making R_FAME 72.5% slower. However, this changes for more than 15 attributes, and C_W11 becomes the second slowest scheme. At 20 attributes, C_W11 takes 513.42ms and R_FAME 878.26ms, making this last scheme 70% slower. The fastest scheme is O_W11, which takes 250ms for 20 attributes, making it 33% faster than the second fastest scheme (C_BSW07) for the same case.

Figure 4.6b presents the results for the RPI0. It shows that C_FAME and R_FAME have a slight difference of 0.26% in the case of 20 attributes: 3.76s for C_FAME and 3.77s for R_FAME. This is a notable difference from the case of RPI4, in which this gap made R_FAME 72.5 slower than C_FAME. Overall, the fastest scheme is O_W11, which takes 1s for the case of 20 attributes. Due to being implemented in C++, it is highly efficient and more suitable for the RPI0 than the python version of the same scheme (i.e., C_W11). Regarding the second fastest scheme, C_BSW07 takes 1.98s for the worst case. This makes the difference between C_BSW07 and O_W11 noteworthy

since it makes O_W11 49% faster.



(a) Results in a RPI4 in seconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.7 CP-ABE decryption time.

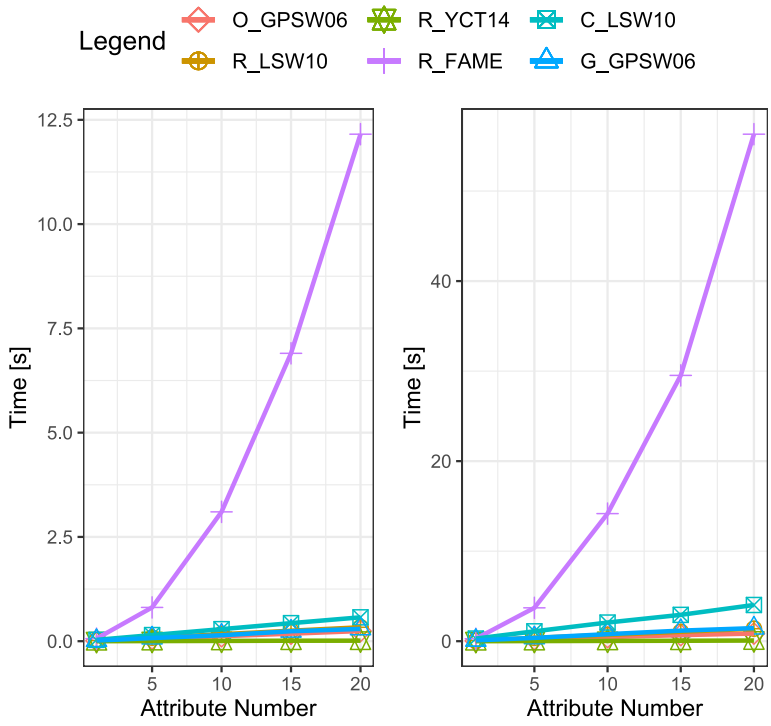
Although encryption time grows linearly with the number of attributes contained in the ciphertext, the pattern does not always hold for decryption. As was introduced in Table 4.4, some schemes (e.g., FAME) have the distinctive feature of constant decryption time, independent of the number of attributes in the ciphertext. This is seen in Figure 4.7, where G_FAME decryption takes 800ms for every case in a RPI4 and approximately 4s in a RPI0. However, the Python implementation (C_FAME) and Rust implementation (R_FAME) take 160ms and 130ms, approximately 80%-84% less time than G_FAME.

One of the differences between the RPI0 and RPI4 is that in RPI0, C_FAME and R_FAME no longer take a similar time. In fact, Rust's efficiency over Python is apparent in these results, with R_FAME taking 607.69ms and C_FAME taking approximately 950ms, 56% more time for the same scheme. Interestingly, G_FAME is not always the slowest scheme. In the case of RPI4, the slowest scheme for more than 12 attributes is Rabe-BSW07 (R_BSW07), taking up to 1.34s to decrypt a ciphertext of 20 attributes.

In the case of RPI0, the same happens for more than 16 attributes, with R_BSW07 taking 4.47s for a ciphertext of 20 attributes. Finally, the fastest option in both devices is C_BSW07.

4.5.2 KP-ABE

This section analyzes the libraries that implement KP-ABE schemes, analogous to the analysis conducted for CP-ABE in Section 4.5.1. To this end, we present Figure 4.8, which shows the time required to create users' private keys as a function of the number of attributes contained in the access policies.

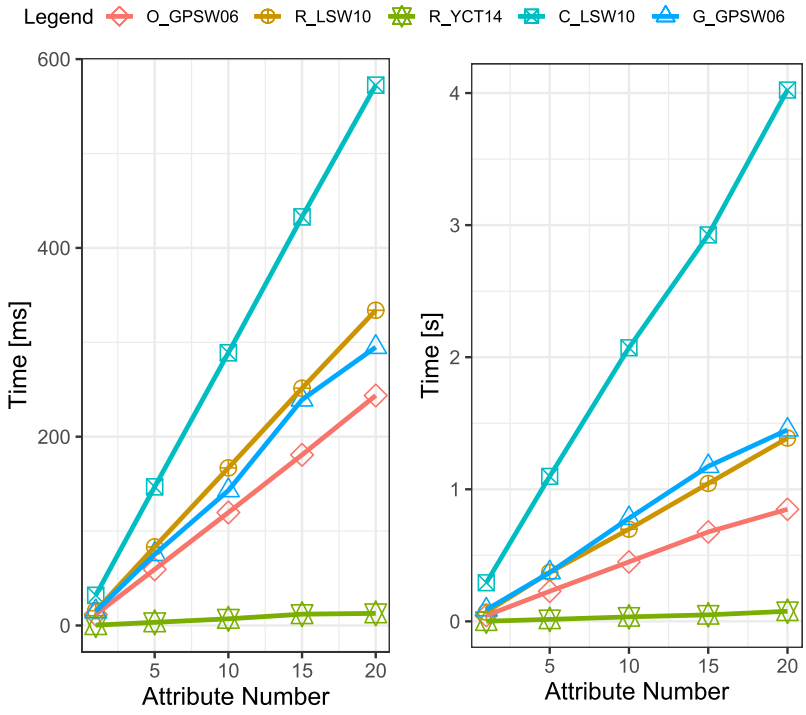


(a) Results in a RPI4 in seconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.8 KP-ABE key generation time.

Figure 4.8 shows that R_FAME has an exponential time evolution for user key generation. In fact, for the case of 20 attributes, the time required goes up to 12s for RPI4 and up to 56s for RPI0. The difference with the second slowest scheme is big enough that we provide Figure 4.9 to visualize it.

Figure 4.9 shows that Rabe-YCT14 (R_YCT14) is the fastest scheme-library com-



(a) Results in a RPI4 in milliseconds.

(b) Results in a RPI0 in seconds.

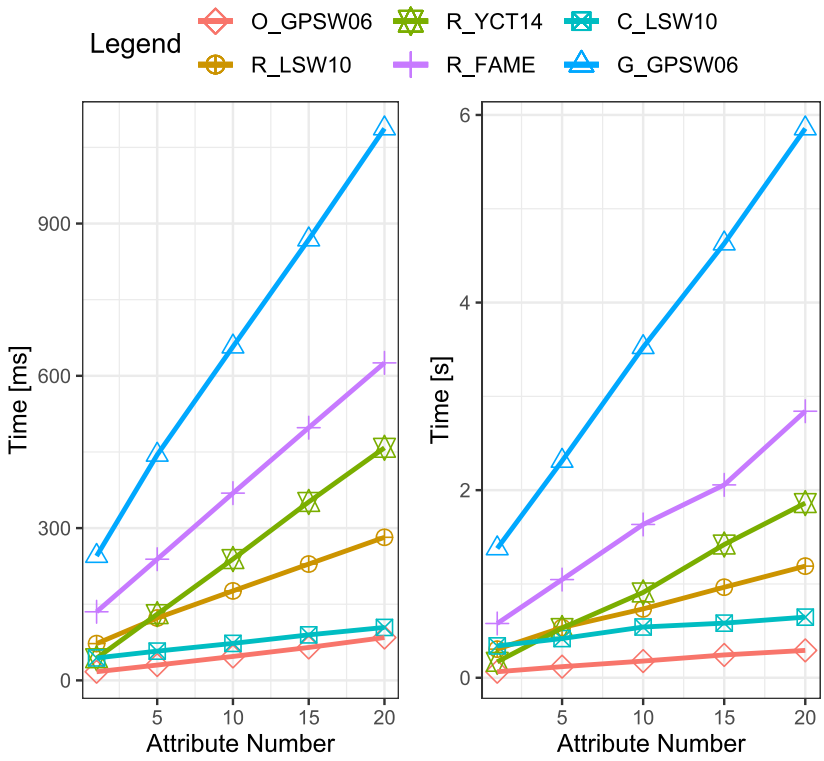
Fig. 4.9 KP-ABE key generation time without R_FAME.

bination. It takes 12ms to generate a key with 20 attributes for a RPI4 and 76.4ms for a RPI0. In RPI4, R_YCT14 shows a significant difference with the second fastest scheme, OpenABE-GPSW06 (O_GPSW06). For the same case of 20 attributes, O_GPSW06 requires 243s, a difference of 181%. Meanwhile, in the RPI0, O_GPSW06 takes 848ms, a difference of 166% from R_YCT14.

Regarding the second slowest scheme, Charm-LSW10 (C_LSW10), it is interesting to see that in the RPI4, it requires 572ms. This implies 71% more time than the same scheme in Rust, Rabe-LSW10 (R_LSW10), which takes 334ms. However, as Figure 4.9b presents, in RPI0, the difference between both schemes goes up to 185%: 4s for the worst case in C_LSW10 and 1.4s in the R_LSW10. The experiment, thus, clearly shows how devices with reduced computing capabilities favor libraries written in Rust and C++ over those implemented in Python.

Regarding encryption, the results can be seen in Figure 4.10. It is clear from the figure that the fastest scheme for encryption is O_GPSW06, taking 84ms for 20 attributes in a RPI4 and 292ms in a RPI0. Despite its slow key generation time, C_LSW10 is a

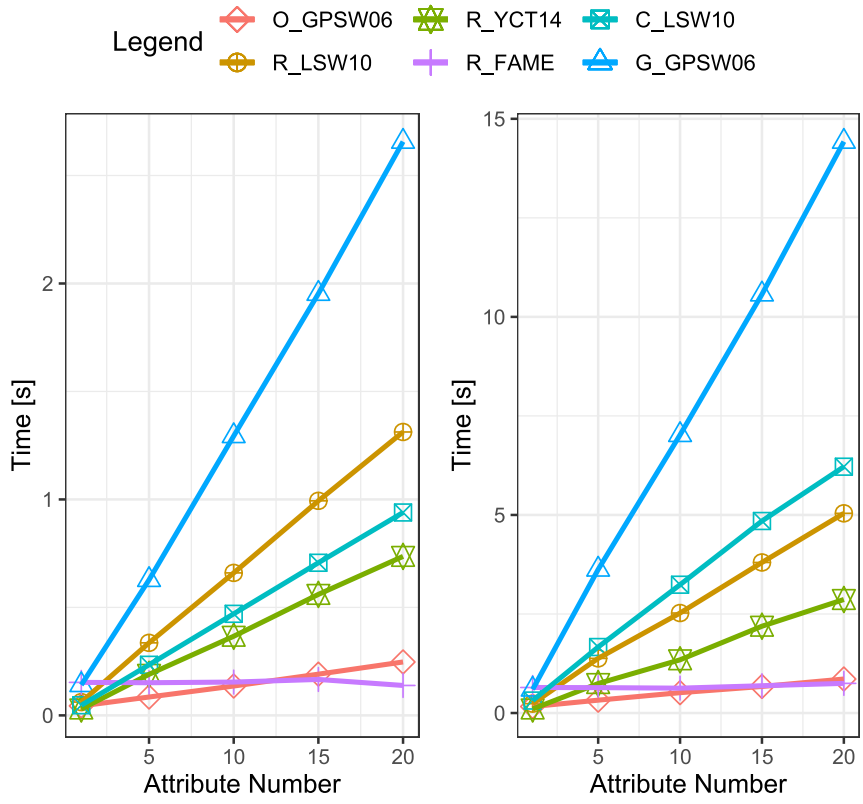
close second for the most efficient encryption scheme. In a RPI4, it takes 104ms for the worst case: barely 23% more than O_GPSW06. In a RPI0, however, C_LSW10 requires 646ms for the worst case: 121% more than O_GPSW06. In contrast, the slowest scheme in both devices is GoFE-GPSW06 (G_GPSW06).



(a) Results in a RPI4 in milliseconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.10 KP-ABE encryption time.

Finally, the decryption time for KP-ABE is depicted in Figure 4.11. It can be seen that KP-ABE FAME has constant decryption time, as with its CP-ABE version. However, the only library implementing it is *Rabe* (R_FAME). As a result of the FAME's constant decryption time, the fastest decryption scheme depends on the number of attributes of the Ciphertext (*CT*). Therefore, O_GPSW06 is the fastest scheme for less than 11 attributes in RPI4 and less than 15 attributes in RPI0. RPI4 takes 136 ms to decrypt a *CT* with 10 attributes using O_GPSW06; and for the same scheme, but with 15 attributes, RPI0 takes 670ms. Meanwhile, for more than 11 attributes, R_FAME becomes the fastest option in RPI4, taking 150ms; and for more than 15 in RPI0, taking 700ms. Meanwhile, the slowest scheme is G_GPSW06.



(a) Results in a RPI4 in seconds.

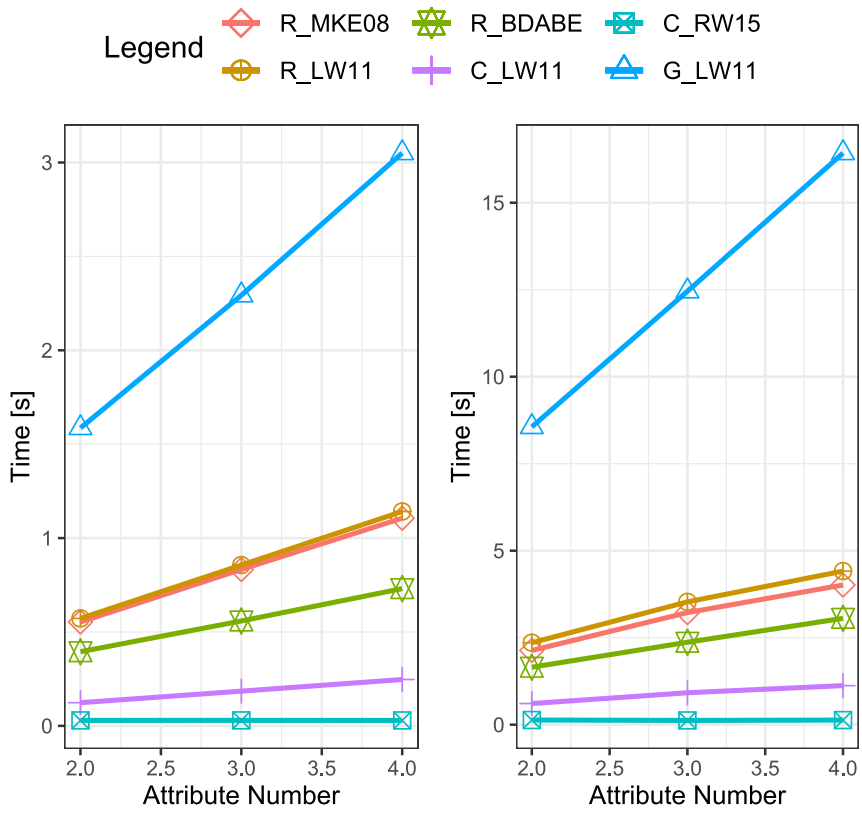
(b) Results in a RPI0 in seconds.

Fig. 4.11 KP-ABE decryption time.

4.5.3 dCP-ABE

Decentralized CP-ABE has an added operation to those shown for KP-ABE or CP-ABE: authority setup. Although every scheme requires a setup, decentralized schemes require the initialization of each authority. Some decentralized schemes allow users to setup authorities at any point in the system's lifetime. Thus, it is interesting to quantify how much time these setups can take. The time can be visualized in Figure 4.12.

As Figure 4.12 presents, authorities' setup time varies widely between schemes. The figure shows the time it takes to configure five authorities. Specifically, it shows the time evolution depending on how many attributes each manages. Charm-RW15 (C_RW15) is the fastest in both devices, taking 30ms for the five authorities to be set up in a RPI4, with each of them controlling four attributes. The RPI0 requires 132ms for the same operation. The closest scheme in execution performance to C_RW15 is



(a) Results in a RPI4 in seconds.

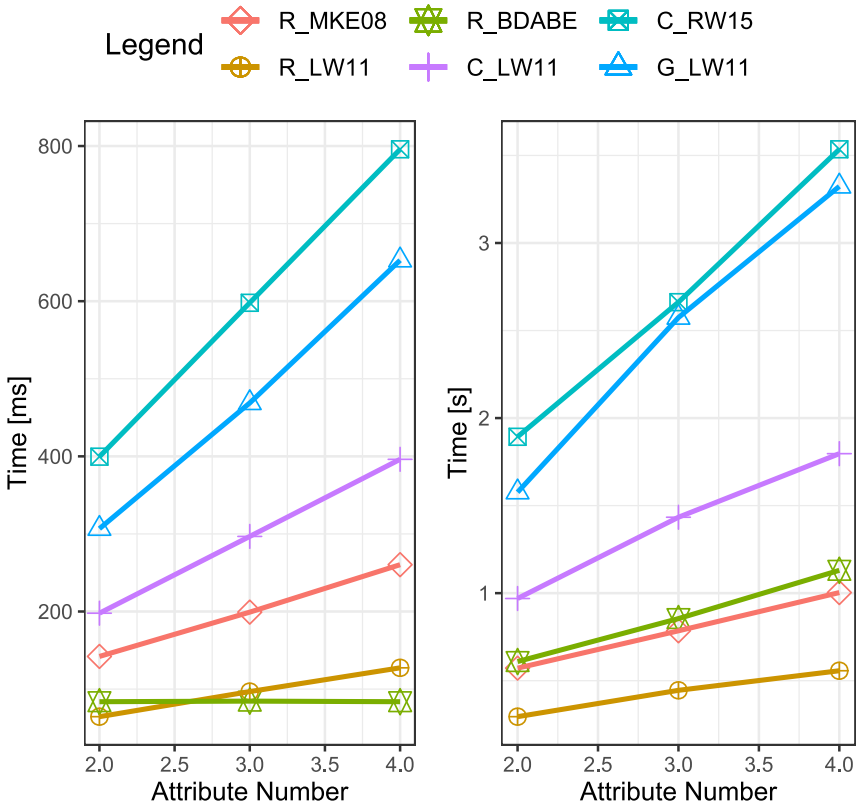
(b) Results in a RPI0 in seconds.

Fig. 4.12 dCP-ABE authority setup time.

Charm-LW11 (C_LW11). However, an analysis of the results shows that C_LW11 takes 246.84ms in RPI4, presenting a 157% difference compared to C_RW15. The same ratio is maintained at RPI0, where C_LW11 takes 1.12s. Therefore, even the second fastest scheme is no match for C_RW15. It is noteworthy that RW15 is the evolution of LW11, and thus stands to reason that RW15 is more efficient in some of its operations. Meanwhile, G_LW11 is the slowest, maintaining *GoFE* as the slowest library for many schemes and operations.

After the authorities are setup, they can start generating users' SK_{ABE}. For this experiment, we work with five authorities, each of which controls a variable amount of attributes, from 2 to 4. The result of key generation time can be visualized in Figure 4.13. It shows that the fastest dCP-ABE scheme for key generation in the RPI0 is Rabe-LW11 (R_LW11), which takes 557ms for the case of requesting 4 attributes from

each attribute authority. In contrast, on RPI4, the fastest scheme for 3 attributes or more is Rabe-BDABE (R_BDABE). For the case of asking 4 attributes to each attribute authority, R_BDABE takes 33% less than R_LW11 (85.25ms in R_BDABE vs 127ms in R_LW11). In the case of 3 attributes or less, the fastest scheme is R_LW11. However, the advantage presented by R_BDABE is so marginal that when the processing capabilities are reduced (case of RPI0), this advantage disappears. Moreover, as introduced during the qualitative evaluation in Section 4.3, to the result of R_BDABE, we have to add the time needed to interact with the Blockchain, which can be long and unpredictable. Thus, even for RPI4, R_LW11 can be considered the fastest key generation scheme, taking 127ms for the worst case.

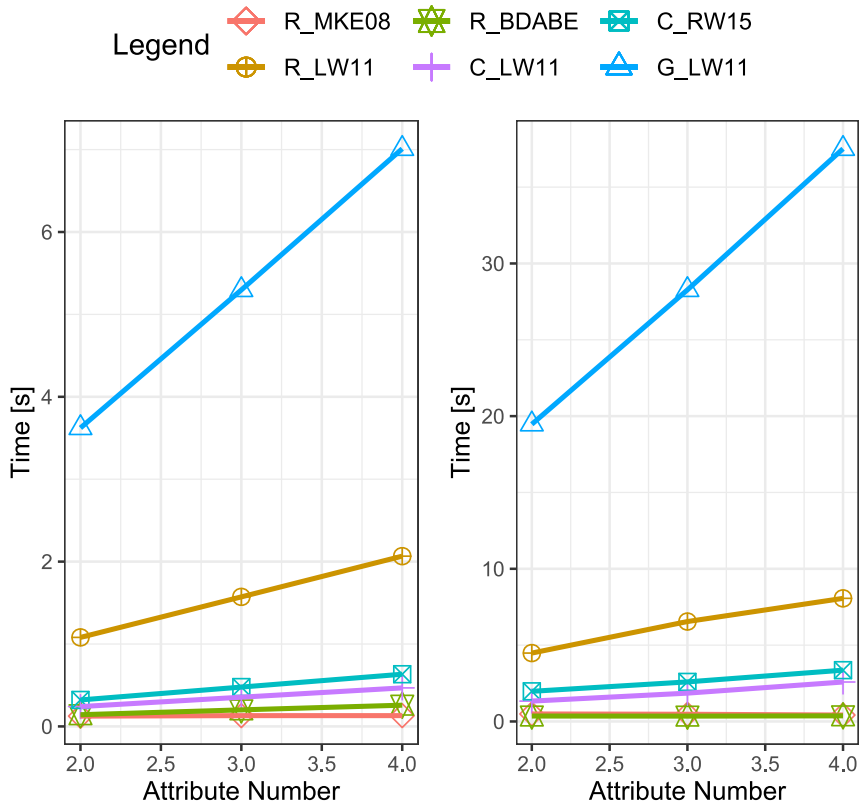


(a) Results in a RPI4 in milliseconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.13 dCP-ABE key generation time.

Finally, the slowest scheme is C_RW15 in both cases. However, the difference between RPI0 and RPI4 is noteworthy. While G_LW11 is the slowest scheme in RPI4

with a noticeable difference from C_RW15, this difference is much smaller in RPI0. Moreover, in the case of having to request 4 attributes, G_LW11 on RPI0 takes 3.32s while C_RW15 takes 3.53s. The little difference between the two schemes is related to the implementation of *GoFE*. In all the results, it has been observed that this library, when implemented in RPI0, achieves much worse results than in RPI4. However, key generation is an operation that takes place only when users require a key, and its impact is limited. Meanwhile, encryption and decryption take place more often. Encryption results are presented in Figure 4.14.



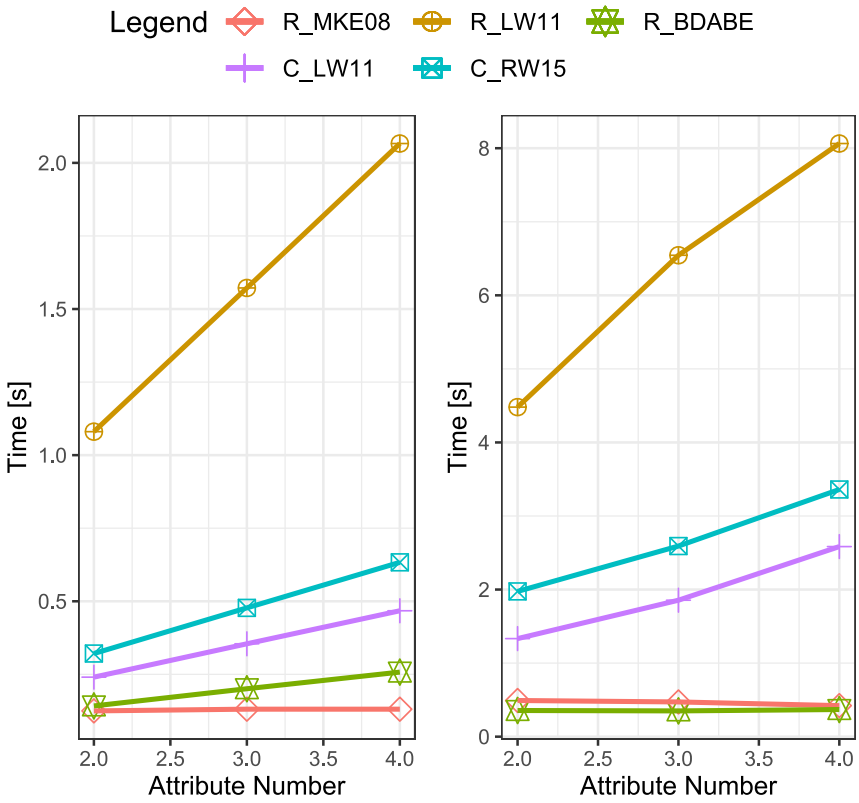
(a) Results in a RPI4 in seconds.

(b) Results in a RPI0 in seconds.

Fig. 4.14 dCP-ABE encryption time.

Figure 4.14 shows that the slowest encryption scheme is G_LW11, taking up to 5s for a policy of 20 attributes (distributed homogeneously among five authorities) in a RPI4. For the same operation, the RPI0 requires 37.5s. In fact, LW11 is generally one of the slowest schemes since it is also the slowest dCP-ABE scheme for *Rabe*

(R_LW11), which takes 1.3s for the worst case in the RPI4 and 8s for a RPI0. To better see the results of the rest of the schemes, we provide Figure 4.15. It shows that the fastest one is Rabe-MKE08 (R_MKE08), taking 131ms for encryption in RPI4. This same operation in the RPI0 takes 439ms. Although R_BDABE is almost as fast as R_MKE08, one should consider the time delays related to Blockchain operations.

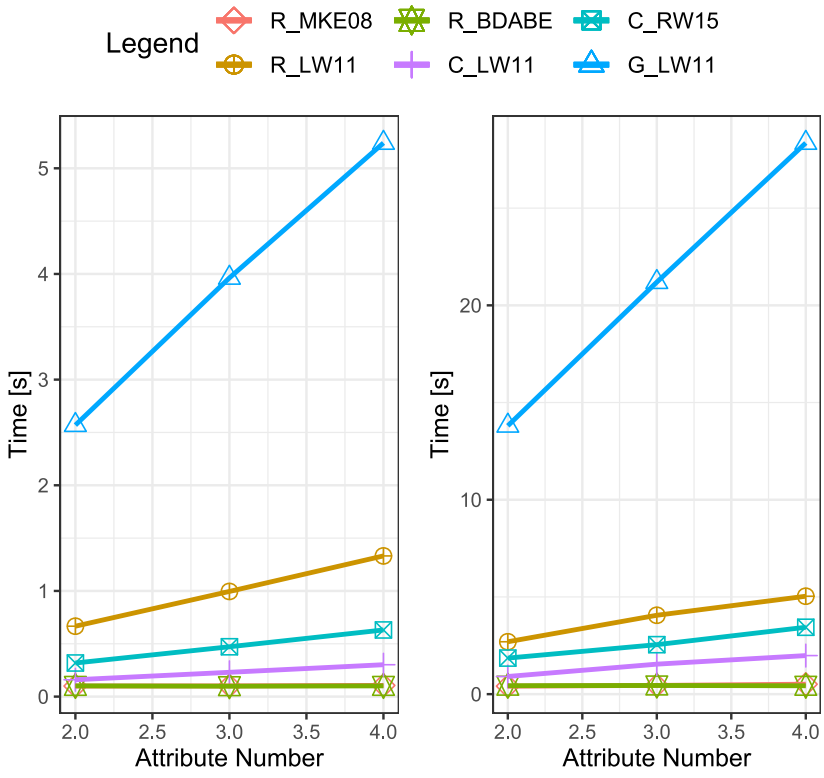


(a) Results in a RPI4 in seconds.

(b) Results in a RPI0 in seconds.

Fig. 4.15 dCP-ABE encryption time without G_LW11.

Finally, after encrypting information, we depict decryption time results in Figure 4.16. We see that results are somewhat analogous to those of encryption. Decryption is faster than encryption, but the schemes' performance is equivalent. Once again, the results show that G_LW11 is the slowest scheme and R_BDABE and R_MKE08 are the fastest schemes. However, as previously stated, when implemented, BDABE will have an added delay related to the Blockchain. Thus, the fastest scheme is R_MKE08, which takes 105ms on average for decryption in a RPI4 and 510ms for RPI0.



(a) Results in a RPI4 in seconds. **(b)** Results in a RPI0 in seconds.

Fig. 4.16 dCP-ABE decryption time.

4.6 Analysis

This section presents the analysis of the libraries.

4.6.1 Discussion

As a result of our experimental evaluations, we provide Table 4.5, which summarizes the most efficient schemes for each case. The performance of BDABE relies on the used Blockchain, so despite being included in the results, its lack of predictability makes us dismiss it as an option for the industrial ecosystem. The results are clustered into two scenarios. First, the schemes are classified according to whether they are CP-ABE, KP-ABE, or dCP-ABE, and second, they are classified according to the implementation language (Python, Go, C++ or Rust). The results in the table show the most efficient schemes for both assumptions, depending on the operation to be performed: authorities

setup, key generation, encryption, or decryption.

If we analyze the results according to the ABE mode, we can see that the fastest scheme for generating keys in CP-ABE is O_W11, which is also the fastest for encryption. However, since it is a CP-ABE scheme, its decryption is slower than its encryption. For fast decryption, the appropriate scheme is C_BSW07. This last scheme provides well-balanced time requirements in its three operations since, in addition to being the fastest in decryption, it is the second fastest in encryption and key generation.

Meanwhile, if we analyze the results for KP-ABE, the fastest scheme for key generation is R_YCT14. Industrial ecosystems have a long lifespan and benefit from solutions with efficient user adding. The second fastest scheme for key generation, O_GPSW06, is also the fastest scheme for encryption. Furthermore, O_GPSW06 is also the fastest scheme for decrypting access policies with less than 10 attributes in RPI4 and less than 15 attributes in RPI0. However, if the system has access policies with more attributes, R_FAME becomes a better choice. R_FAME has constant decryption time, so its decryption time is independent of the complexity of the access policies.

The last ABE mode is dCP-ABE. For this ABE mode, the time required to set up authorities must be considered. Some schemes only allow the creation of attribute authorities during the system setup, but others allow authorities to be set up throughout the system's lifetime. In this regard, the fastest scheme is C_RW15, but it is also one of the slowest schemes for key generation, encryption, and decryption. For faster key generation, the best option is R_LW11. However, this scheme is also the second slowest for encryption and the slowest for decryption. Thus, the fast key generation is hardly compensated by the rest of the delays since encryption and decryption are more common tasks. Finally, the best option for fast encryption and decryption is R_MKE08. It is noteworthy that this scheme is also the second fastest one for key generation.

Table 4.5 also summarizes the fastest schemes for each library. Not all libraries implement the same schemes, so the options for some may be reduced. For example, *OpenABE* only provides one CP-ABE and one KP-ABE scheme. However, we can see that W11 is the fastest scheme for setupkey generation in both *OpenABE* and Charm, and GPSW06 is the fastest for encryption in *GoFE* and *OpenABE*. In the case of *Rabe*, we can see how MKE08 is one of the most balanced schemes: it is the fastest scheme provided by *Rabe* for authority generation, encryption, and decryption.

Table 4.5 Fastest schemes for each considered function.

		Auth Setup		KeyGen		Enc.		Dec.	
		RPI0	RPI4	RPI0	RPI4	RPI0	RPI4	RPI0	RPI4
Scheme Mode	CP-ABE	—	—	O_W11	O_W11	O_W11	O_W11	C_BSW07	C_BSW07
	KP-ABE	—	—	R_YCT14	R_YCT14	O_GPSW06	O_GPSW06	O_GPSW06 (<15 att) R_FAME(>15 att)	O_GPSW06 (<10 att) R_FAME(>10 att)
	dCP-ABE	C_RW15	C_RW15	R_LW11	R_LW11	R_MKE08	R_MKE08	R_MKE08	R_MKE08
Language	Python	RW15	RW15	BSW07(<5 att) W11 (>5 att)	BSW07(<10 att) W11(>10 att)	LSW10	LSW10	BSW07	BSW07
	Golang	LW11	LW11	GPSW06	GPSW06	GPSW06	GPSW06	FAME (>6 att) GPSW06 (<6 att)	FAME (>6 att) GPSW06 (<6 att)
	C++	—	—	W11	W11	GPSW06	GPSW06	GPSW06	GPSW06
	Rust	MKE08	MKE08	YCT14	YCT14	MKE08 (>5 att) YCT14 (<5 att)	MKE08 (>5 att) YCT14 (<5 att)	MKE08	MKE08

4.6.2 Library and Scheme Choice

CP-ABE schemes are the most suitable for industrial use of all the proposed ABE modes. The key to this is that CP-ABE allows encryption employing access policies. Thus, the encrypting device decides under what conditions the information that the device itself has generated can be accessed.

Once this has been decided, choosing a CP-ABE or dCP-ABE scheme is necessary. Although the operation is similar, decentralized systems base their operation on the existence of several attribute authorities, which are the ones that deliver the keys to the users.

The results show that the only C++ library (OpenABE) does not include dCP-ABE schemes. In an Industry 4.0 environment, with a massive deployment of IIoT devices, having C++ libraries is a great advantage; instead, it is observed that among the fastest dCP-ABE schemes, the libraries that implement them are in Python or Rabe.

Furthermore, the encryption time in dCP-ABE is longer than for CP-ABE. This is detrimental to their deployment in an industrial environment where IIoT devices with limited capabilities perform the encryption operations. Therefore, the library-scheme combination chosen for the following chapters will be O_W11.

4.7 Summary

This chapter provides a qualitative and quantitative evaluation of the 11 existing ABE libraries. Rabe and Charm have been identified as the libraries with vulnerable schemes (i.e., YCT14, YJ14, and DAC-MACS). The qualitative study has also identified the mathematical dependencies of each library, and since the schemes are used in hybrid mode, the AES modes used by each.

Four libraries are selected based on this qualitative evaluation, and their efficiency is quantitatively assessed. We measure efficiency in terms of how much time each scheme takes to perform basic operations like key generation, encryption, and decryption.

As a result of our experimental evaluations, we provide a table in which we indicate which schemes are faster based on their approach, programming language, and the device running them. Using these results, we answer Question 2 and conclude in the discussion that the best library for deploying ABE in Industry 4.0 is *OpenABE*, with the W11 scheme.

Chapter 5

Multi-Layered CP-ABE with Access Policy Update

The adoption of IT solutions and technologies in Industry 4.0 provides numerous benefits by increasing network connectivity. However, this increased connectivity introduces additional risks in manufacturing environments, creating potential access points from partners in the value chain.

For the above reason, industrial information must be protected at all times, and, as mentioned in Chapter 3, an E2E security solution is necessary. To this end, the flexibility provided by CP-ABE schemes makes them a promising solution. However, IIoT devices benefit from lightweight schemes, so the computational cost of CP-ABE may limit its application in IIoT networks. Although the operation times have been explored in Chapter 4, the ciphertext size is another limitation of CP-ABE. Encryption operations expand the original plaintext, which must also be addressed.

Furthermore, industrial environments also have a long lifespan, during which access rights to information change. Therefore, the CP-ABE-based encryption solution must also be flexible and resilient and adapt to changes in the organizational structure. However, CP-ABE schemes do not have their own mechanism to update the ciphertext policy.

For all these reasons, it is necessary to develop a system for updating access policies and ciphertexts. Usually, this involves decrypting the symmetric key and re-encrypting it according to the new access policy. However, this approach exposes the symmetric key and leaves it vulnerable, breaking E2E security. This chapter proposes a Multi-Layered CP-ABE that meets industrial requirements and can update the access policies without exposing the symmetric key.

This chapter is divided into five sections. Section 5.1 explains the methodology designed to comply with **R2** and **R3** and presents several technical requirements to verify

compliance. Section 5.2 discusses the industrial deployment of CP-ABE schemes in IIoT networks following several ETSI guidelines. Section 5.3 presents our proposed CP-ABE scheme with policy update, and Section 5.4 evaluates it experimentally. Finally, Section 5.5 summarizes the chapter's main ideas.

The research conducted to write this chapter has been included in three articles:

- *"Multi-Layered CP-ABE scheme for flexible policy update in Industry 4.0"* [25] that has been presented at the *10th Mediterranean Conference on Embedded Computing (MECO 2021)*.
- *""They got my keys!": On the Issue of Key Disclosure and Data Protection in Value Chains"* [26] that has been presented at the *2nd IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0 (ARCI' 2022))*.
- *"End to End Secure Data Exchange in Value Chains with Dynamic Policy Updates"* [27] that has been submitted to *Future Generation Computer Systems* and published as a preprint in *arXiv* in 2022.

5.1 Requirements

Section 3.3 introduced the requirement *"**R2.** The solution shall provide E2E confidentiality and integrity for data management in industrial environments."* To fulfil it, below are defined the technical requirements to be met to validate **R2**.

TR 2.1. Data shall be encrypted by the devices that generate it. E2E security means that information must be secured before leaving the device that generated it. Therefore, data must be encrypted by the device that generated it.

TR 2.2. The solution shall fulfill industrial guidelines for ABE. Chapter 4 concluded that the chosen scheme for industrial use was W11. However, this scheme was not designed with industry in mind. Thus, the solution shall identify the industrial guidelines for ABE and apply them to W11.

TR 2.3. The solution shall be independent of the data type. Section 3.2 identified different information exchange cases. This information exchange gives members of value chains an advantage over their competitors. Therefore, the information protection system must be independent of the data format.

Section 3.3 also defined *"**R3.** The solution shall be flexible to changes in data access rights."*. To this end, below are the technical requirements to be met to validate

its fulfilment.

TR 3.1. The solution shall provide dynamic policy update. The data exchange solution requires flexibility and responsiveness. In the case of ABE, this implies being flexible enough to evolve when access policies change. Thus, a policy update system shall be designed to maximize the lifetime of the defined system.

TR 3.2. The solution shall provide dynamic policy revocation. Similar to the case of **TR 3.1.**, the policy management system should also account for revoked policies.

Fulfilling **R2** and **R3** involves meeting the technical requirements presented here. Secure data exchange solutions must balance security, industrial availability, and system flexibility to answer Question 3.

5.2 Industrial Constraints Defined by the ETSI

ABE schemes offer many advantages for information exchange, most notably their one-to-many encryption capability. Among all the ABE schemes analyzed in Chapter 4, it has been concluded that CP-ABE is the most suitable mode for industrial use. Moreover, specifically, the CP-ABE scheme known as W11.

However, ABE schemes are computationally heavy, and it is known that encryption schemes can put the availability of industrial systems at risk [181]. Therefore, before applying W11 directly to the industrial domain, ABE's industrial applicability must be guaranteed by fulfilling **TR 2.2**. To this end, this section discusses two ETSI documents that are crucial for this:

- ETSI-TS-103-458 [7] defines the requirements for any ABE scheme deployed in an environment with IoT and IIoT devices that handle sensitive information.
- ETSI-TS-103-532 [182] defines the trust models and functions to use a generic CCA-Secure ABE scheme.

5.2.1 Industrial Requirements

As mentioned above, ETSI-TS-103-458 identifies up to 35 high-level requirements for using ABE schemes with IoT devices. These requirements relate to 7 different use cases. Table 5.1 shows a summary of the most relevant requirements for the industrial case addressed in this thesis, in which the use of the CP-ABE W11 scheme is combined with AES-GCM to guarantee integrity verification.

ETSI 01 is covered by the W11 scheme since users are identified by attributes. The use of the CP-ABE scheme also covers ETSI 02-05 since the information is encrypted

Table 5.1 Compliance with high-level requirements defined in [7].

Code	Requirement	Fulfilled by W11 + AES-GCM
ETSI 01	**Non-identity based access control policies**	Yes
ETSI 02	**Time-based access control**	Yes
ETSI 03	**Position-based access control**	Yes
ETSI 04	**Role-based access control**	Yes
ETSI 05	**Attribute based access control**	Yes
ETSI 06	**Emergency access control**	Yes
ETSI 07	**Integrity protection**	Yes
ETSI 08	**Attribute update and expiration**	No
ETSI 09	**Policy update and expiration**	No
ETSI 10	**Addition of new policies to ciphertexts**	No
ETSI 11	**Attribute management**	No

according to access policies, and these policies can reflect any of these requirements. As for integrity protection (ETSI 07), it is supported by AES-GCM. However, we can see that ETSI 08-11 requirements, aligned with **TR 3.1.** and **TR 3.2.**, are not covered by either W11 or AES-GCM.

5.2.2 Recommended ABE Scheme

The CP-ABE schemes proposed by ETSI in ETSI-TS-103-532 are based on a KEM construction like the one introduced in Section 2.3.3. However, KEM constructions only have CPA security [182] and they only protect against passive attackers. Therefore, attackers can still obtain the symmetric key by attacking the KEM even if the symmetric algorithm has a higher security level. This thesis considers that to fulfill **TR 2.2** satisfactorily, the symmetric key must be protected against both passive and active attackers. In other words, the scheme must have CCA security [183]. The concepts of CPA and CCA security are explained in Annex B.

In a KEM scheme, CCA security implies that attackers cannot recover keys, even if they capture their encapsulation. Following this premise, the CP-ABE schemes proposed by ETSI [182] provide CCA security to several KEM constructs: FAME [109], W11 [102], and GPSW [89]. The ETSI CP-ABE scheme exploits these KEM schemes and creates four modules combining the KEM construct, the one-time-pad of Section 2.3.3, the Fujisaki-Okamoto (FO) transformation [183], and AES-GCM. The constructions are explained below.

CCA Secure CP-ABE Encryption Algorithm

Figure 5.1 shows the layered CCA secure encryption system proposed by the ETSI. The reason for this layered system lies in the CPA security of KEM schemes, explained above. Since KEM schemes do not have the CCA security needed by the industry, the ETSI starts from a KEM with CPA security and, based on layers or security modules, builds a system with CCA security. Thus, in this thesis, we start from the CP-AB-KEM based on W11, and three security layers are built on it, which are explained below.

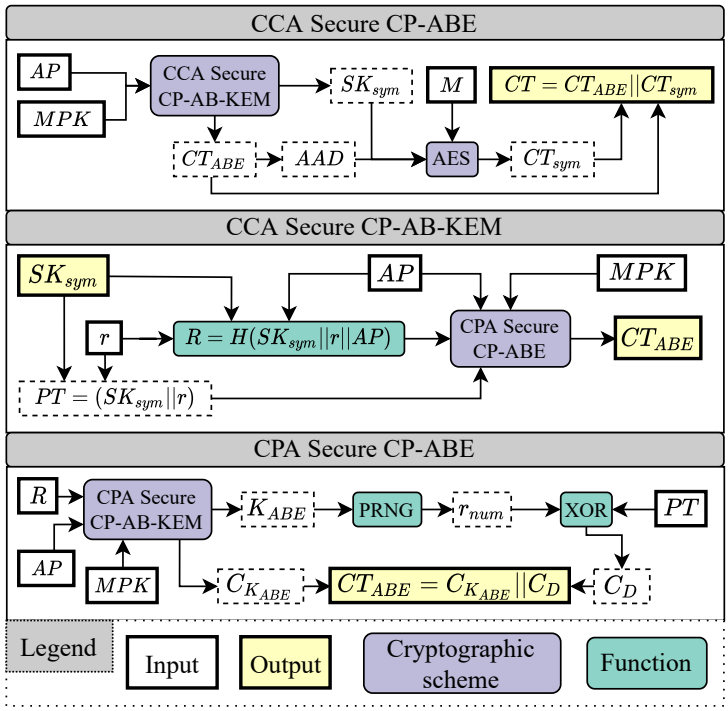


Fig. 5.1 ETSI's CCA Secure CP-ABE encryption modules. *AP* stands for the access policy, *M* for the message, and *MPK* for the Master Public Key. Each security layer uses the previous layer as a black box, so the system is highly modular.

CCA Secure CP-ABE: This algorithm inputs the original message M, an access policy AP and a MPK in a CCA-Secure CP-AB-KEM scheme to generate an AES-GCM symmetric key (SK_{sym}) and its encapsulation (CT_{ABE}). It returns M encrypted with AES-GCM in CT_{sym} and the encrypted symmetric key in CT_{ABE}.

$$Enc_{CP-ABE_{CCA}}(AP,\ MPK,\ M) \rightarrow CT = (CT_{ABE} \| CT_{sym}) \tag{5.1}$$

CCA Secure CP-AB-KEM: This module randomly generates SK_{sym} and a nonce r. It applies a modified version of FO [183] by including the AP in the hash that generates R. It then inputs R, AP, $PT = (SK_{sym}||r)$ and MPK in a CPA-Secure CP-ABE scheme to generate CT_{ABE}. The function returns both SK_{sym} and CT_{ABE}.

$$Enc_{CP-ABKEM_{CCA}}(SK_{sym},\ r,\ AP,\ MPK) \rightarrow CT_{ABE},\ SK_{sym} \tag{5.2}$$

CPA Secure CP-ABE: It calls the W11 CPA Secure CP-AB-KEM scheme to generate a key (K_{ABE}) and its encapsulation ($C_{K_{ABE}}$) according to an AP and a random number R. K_{ABE} is used as input for the one-time-pad used to encrypt Plaintext (PT). The result of encrypting PT with the one-time-pad of Figure 2.7 is C_D. The algorithm returns $CT_{ABE} = (C_{K_{ABE}}||C_D)$.

$$Enc_{CP-ABE_{CPA}}(AP,\ MPK,\ R,\ PT) \rightarrow CT_{ABE} = (C_{K_{ABE}}||C_D) \tag{5.3}$$

CCA Secure CP-ABE Decryption Algorithms

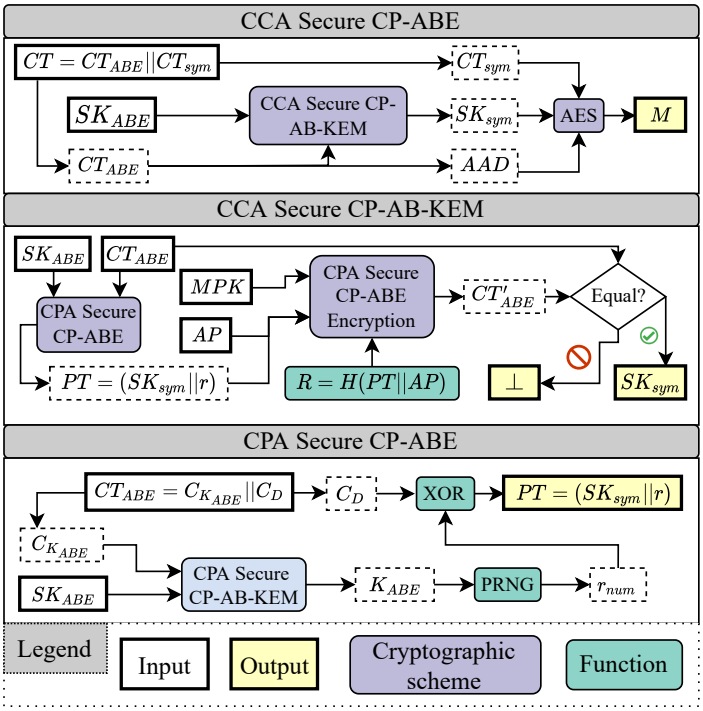


Fig. 5.2 ETSI's CCA Secure CP-ABE decryption modules.

This section outlines the decryption modules proposed by ETSI, depicted in Figure 5.2.

CCA Secure CP-ABE: This algorithm takes the CT such as $CT = (CT_{ABE} \| CT_{sym})$ generated during encryption and SK_{ABE}. It obtains the SK_{sym} from CT_{ABE} and uses as input in AES-GCM to obtain the original message M.

$$Dec_{CP-ABE_{CCA}}(CT,\ SK_{ABE}) \rightarrow M \tag{5.4}$$

CCA Secure CP-AB-KEM: This layer takes as inputs a SK_{ABE}, CT_{ABE}, and the MPK. It uses a CPA Secure CP-ABE decryption scheme to obtain $PT = (SK_{sym} \| r)$. The ETSI applies FO to achieve CCA security, for which FO requires a re-encryption during decryption. Thus, FO re-encrypts the recovered PT using a CPA Secure CP-ABE scheme. The result of this operation is $(CT')_{ABE}$. The scheme compares CT_{ABE} with CT'_{ABE} and returns SK_{sym} if they are the same. If they are not, it outputs $\perp$.

$$Dec_{CP-ABKEM_{CCA}}(SK_{ABE}, CT_{ABE},\ MPK) \rightarrow \perp \ or\ SK_{sym} \tag{5.5}$$

CPA Secure CP-ABE: It takes $CT_{ABE} = (C_{K_{ABE}} \| C_D)$ and SK_{ABE} as inputs. It uses a CPA Secure CP-AB-KEM scheme to obtain K_{ABE} from $C_{K_{ABE}}$ and thus recover PT from C_D.

$$Dec_{CP-ABE_{CPA}}(CT_{ABE},\ SK_{ABE}) \rightarrow PT = (SK_{sym} \| r) \tag{5.6}$$

Thus, it can be seen that the decryption sequence is straightforward, the only difference with encryption being the CCA Secure CP-AB-KEM decapsulation.

5.3 Architecture Design

Two of the technical requirements established at the beginning of this chapter were **TR 3.1** and **TR 3.2**, which require a policy update and revocation system. In addition, following Section 5.2.1, it became apparent that complying with these two technical requirements also enforces **TR 2.2** since, among the requirements established by ETSI for deploying ABE in environments with IIoT devices, they also included policy updating and revocation. Therefore, having a system that performs these operations is mandatory.

However, updating policies in CP-ABE is a delicate operation since the ciphertext generated with the old policies must also be updated without breaking E2E security. Therefore, the solution proposed in this chapter manages ABE ciphertext policy up-

dates without overloading IIoT devices or violating the confidentiality of the original ciphertext. The solution, moreover, is built on top of ETSI's own Secure CCA scheme to ensure **TR 2.2** compliance.

5.3.1 Layered System Overview

This chapter proposes the layered encryption system schematized in Figure 5.3, Multi-Layered CP-ABE. This system allows policy updates and revocation.

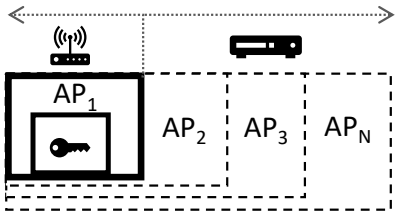


Fig. 5.3 Layered encryption. The IIoT devices apply AP_1. Any other device can apply the following policies.

Our solution encrypts the original message with AES-GCM and SK_{sym} Multi-Layered CP-ABE, based on the ETSI CCA Secure scheme. Multi-Layered CP-ABE also defines a set of access policies, $\mathbb{AP}$ such as $\mathbb{AP} \leftarrow (AP_1 \| AP_2 \| \cdots \| AP_{nLayers} \| AP_{time})$. Of these APs, AP_1 is immutable and will not be updated or revoked to guarantee E2E security. Meanwhile, AP_2 to $AP_{nLayers}$ are updatable and revocable. To determine the order of AP_2 to $AP_{nLayers}$, it is necessary to analyze which policies are most likely to change in the deployed system. The policies most likely to be upgradable are in the upper layers, starting from $AP_{nLayers}$. Likewise, AP_2 is upgradable but also the least change-prone policy. Finally, AP_{time} is a policy related to the last security event in the system. It is used to control when user keys are created; the following paragraphs and sections will explain more about AP_{time} and security events.

The policy update system must met **TR 2.1** to maintain E2E confidentiality and integrity. This is where AP_1 comes in. As explained, this policy is immutable and therefore cannot be updated or revoked. As shown in Figure 5.3, devices that generate data use AP_1 to encrypt SK_{sym} and AES-GCM to encrypt M before sending the resulting CT_1 to a repository. This is the only way to guarantee E2E confidentiality: by ensuring that data can only be read by those who produce and request it.

The following policies, AP_2 to $AP_{nLayers}$, are added to CT_1 by a device with higher performance. The original device could also add these policies, but it is an IIoT device, so its resources and computational capabilities are limited. Therefore, the device with higher capacities uses the modified ETSI scheme to add the upgradable access poli-

cies, using a step-by-step process explained in the following sections. After adding the updatable policies, CT_2 is generated.

Finally, the last AP, AP_{time}, remains to be added. Encryption systems generate decryption keys for users, and it is considered a good security practice to have a key revocation system. Typically, this can be done with revocation lists that tell the system which users should be denied access despite having a valid decryption key. However, value chains have many users from different companies, and revocation lists are not particularly efficient in this scenario [184]. Thus, Multi-Layered CP-ABE binds user keys to the time they were created. Hence, the attribute set $\mathbb{A}$ of users in Multi-Layered CP-ABE is defined as $\mathbb{A} \leftarrow (Att_1 \| Att_2 \| \cdots \| Att_n)$, while their keys will be generated according to $\mathbb{A}' \leftarrow (Att_1 \| Att_2 \| \cdots \| Att_n \| T_{SK})$. T_{SK} is the timestamp indicating when the key was generated. Thus, AP_{time} ensures that the user's key was generated after the last security incident. Whenever a user requests a ciphertext, the final encryption layer containing AP_{time} will be issued.

Details about the system functions and the design of the encryption and decryption algorithms are provided in the following sections.

5.3.2 Role Definition

After outlining how Multi-Layered CP-ABE works, we use this section to introduce the roles required to set the policy update system in Industry, which fulfills **TR 3.1**. Furthermore, the data exchange system design also considers the heterogeneity of the information to be exchanged (**TR 2.3**) and that information is encrypted by the devices that generate it (**TR 2.1**). The roles involved in deploying Multi-Layered CP-ABE are presented in Figure 5.4 and are defined below.

- **Data Owners.** These are the IIoT devices that generate the source data. They apply AES-GCM to the information they have to send and CP-ABE with AP_1 to the symmetric key. Since only the original device handles the unencrypted data, this fulfills **TR 2.1**. The information generated and protected by data owners can be any data type (**TR 2.3**).

- **Attribute Authority.** During system setup, the authority generates the MSK and MPK. It stores and protects the MSK and sends the MPK to data owners and CT engines. Finally, it generates consumers' SK_{ABE} based on $\mathbb{A}' \leftarrow (Att_1 \| Att_2 \| \cdots \| Att_n \| T_{SK})$.

- **System Manager.** It manages system access policies. When a policy is updated, it is in charge of sending it to the policy engine.

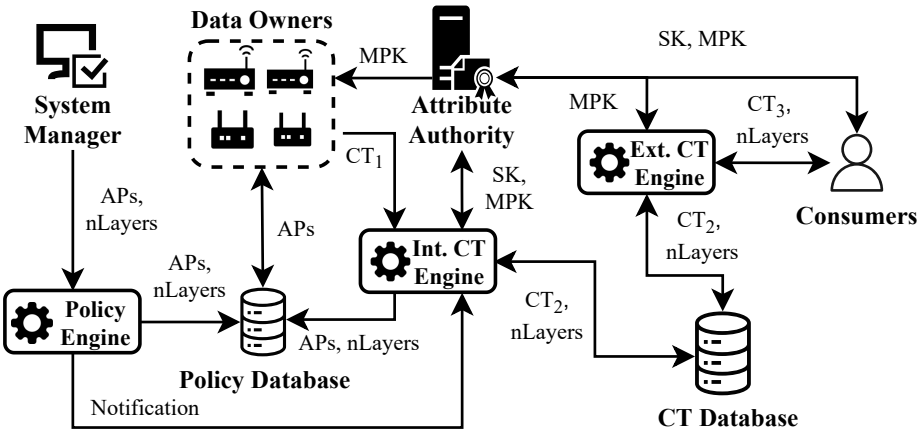


Fig. 5.4 System Design. AP stands for access policy, CT for ciphertext, and $nLayers$ is the number of access policy layers.

- **Policy Engine.** It pushes the new $\mathbb{AP}$ to the Policy Database. It also notifies the Internal CT Engine when a policy update occurs.

- **Internal CT Engine.** It adds the revocable and updatable policies (i.e., AP_2 to $AP_{nLayers}$) to CT_1 and generates CT_2. CT_2 is stored in the the CT database. Whenever an access policy update occurs, the Internal CT Engine receives a notification from the Policy Engine and retrieves the old CT_2 from the CT database, updates and stores it again.

- **External CT Engine.** Once system users obtain their SK_{ABE}, they can use it to access all information whose $\mathbb{AP}$ they comply with. However, as introduced in Section 5.3.2, there must exist a system that manages SK_{ABE} expiration. To this end, when a consumer requests data, the External CT Engine adds a new layer to CT_2 using AP_{time} and outputs CT_3 before sending it to the requesting consumer.

- **Policy Database.** It stores all the access policies contained in different $\mathbb{AP}s$ to be used for encryption.

- **CT database.** It stores CT_2. The ideal solution is to set up a distributed storage solution. This way, ciphertexts are always accessible, even if one of the nodes goes down. Solutions like IPFS may be suitable since it distributes the storage, is immutable, and is tamper-resistant.

Industrial High-Level Reference Model

The beginning of this chapter established ***TR 2.2.*** *"The solution shall fulfill industrial guidelines for ABE"* to validate compliance with **R2**. The previous sections have established different mechanisms to guarantee this technical requirement. However, this thesis considers that security in Industry 4.0 stems from combining plant-level security with the secure exchange of information between members of the value chain.

Chapter 2 presented Figure 2.1, which illustrated the high-level reference model of ENISA for a connected plant. Therefore, we present Figure 5.5, which illustrates at what level of the model the Multi-Layered CP-ABE roles identified in this section would be.

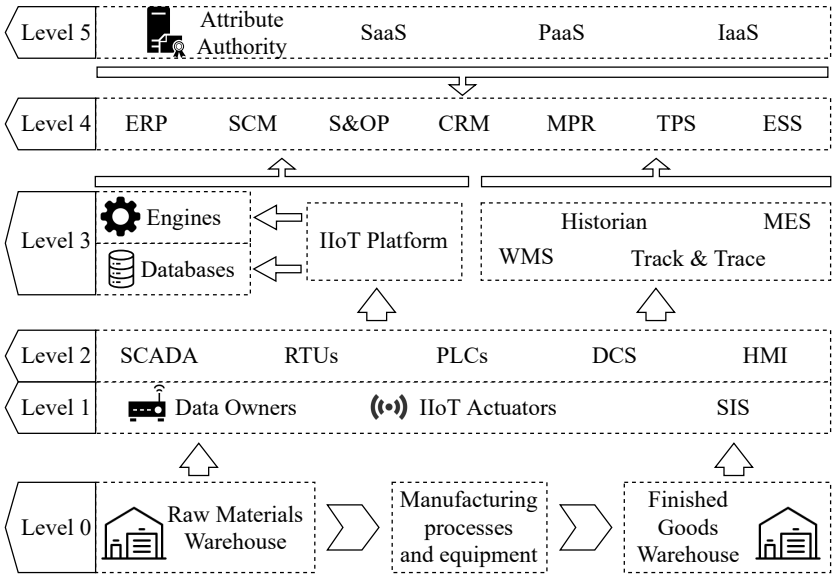


Fig. 5.5 Components of the proposed system mapped to the ENISA high reference model [1].

The original ENISA model dictates that devices interacting directly with the production line must be located at level 1. Therefore, the massively deployed IIoT devices that this thesis considers must be placed at this level. In the case of production parameters, these devices act as data owners in the system. Once they generate CT_1, they send it through level 2 of the model until it reaches the IIoT platform at level 3.

Once CT_1 is at level 3 along with the databases and engines, it can be transformed to CT_2 by inputting it alongside $\mathbb{AP}$ in Multi-Layered CP-ABE. The data is stored here until a user in the value chain requests it. When this happens, the engines generate CT_3 and send it to level 5. Level 5 connects the plant to the cloud and other external

services. In this case, it also connects the plant to other value chain members. Finally, the attribute authority generates decryption keys for data consumers in the system. This role must interact with users inside and outside the enterprise, placing it at level 5.

5.3.3 Update and Revocation Algorithms

Multi-Layered CP-ABE consists of four algorithms: system setup, key generation, encryption, and decryption. Policy addition and policy revocation are achieved by encryption and decryption algorithms. Of these, two algorithms are the same as those presented in W11 and are not modified in Multi-Layered CP-ABE: System Setup and Key Generation.

$SystemSetup\,(K) \rightarrow MPK,\ MSK$

This is the original CP-ABE setup algorithm as defined by W11 [102], performed by the attribute authority. After obtaining MPK and MSK, it sends the MPK to the data owners and the Engines.

$KeyGeneration\,(MSK,\ \mathbb{A}) \rightarrow SK_{ABE}$

Whenever users request a SK_{ABE}, the attribute authority generates a timestamp for the request (T_{SK}). Then, the authority generates SK_{ABE} using the MSK, according to an attribute set $\mathbb{A}'$ that contains the user's attributes and T_{SK}. Thus, $\mathbb{A}'$ is defined as $\mathbb{A}' \leftarrow (Att_1 \| Att_2 \| \cdots \| Att_n \| T_{SK})$. Including T_{SK} in the attribute set dates its generation, ensuring that consumers with an outdated SK_{ABE} cannot access the system.

Meanwhile, the encryption and decryption algorithms are explained below. Encryption consists of three phases. In the first phase, the device that generated the information encrypts and protects the symmetric key with AP_1. In the second phase, the Internal CT Engine adds the remaining access policies contained in $\mathbb{AP}$. Finally, the third encryption phase is done on demand: whenever an external user requests information, the External CT Engine performs a time-based encryption. As mentioned, all of this is further explained below.

Encryption and Update

Multi-Layered CP-ABE must guarantee E2E security and maintain the CCA security the ETSI scheme provides. The proposed encryption scheme is shown in Figure 5.6. To maintain CCA security, the following parameters are taken into account:

- The R value, which contains the random nonce r used by CPA-Secure CP-ABE. The encapsulation of nonce r is one of the pillars of CCA security.

- The immutability of AP_1, which is used in the first layer of SK_{sym} encryption, and it is central to E2E security.

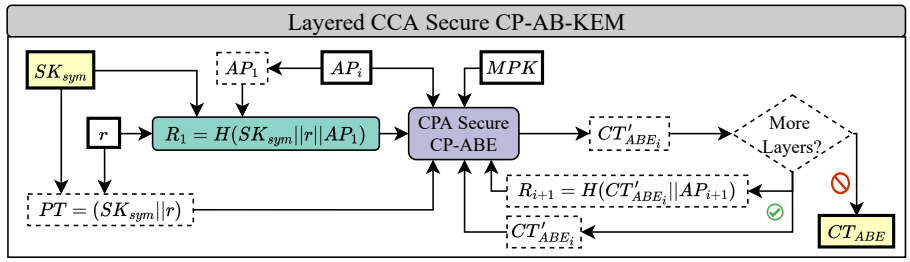


Fig. 5.6 Multi-Layered CP-ABE encryption.

The encryption process is explained below, and Figure 5.7 illustrates the message exchange between the data owners and the Internal CT Engine. To explain the phases, we distinguish the third phase from the first and second one since the third phase is performed on demand by a specific user. The first and second phases are explained below:

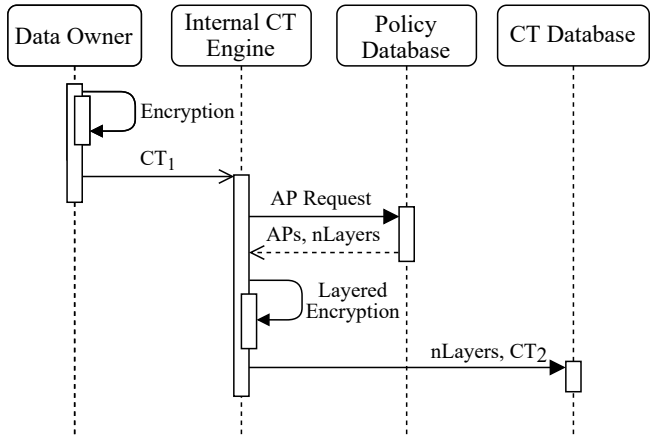


Fig. 5.7 Encryption message exchange.

Encryption $(MPK,\ \mathbb{AP},\ M) \rightarrow CT_2$

The first phase of encryption takes the original message, M, and generates an SK_{sym}. These operations are performed by the data owners and guarantee E2E data confidentiality and integrity. The results of these operations is $CT_1 = (CT_{AES}, CT_{ABE_1})$. Figure 5.7 shows that phase 1 is performed by the data owner, and how it sends CT_1 to the Internal CT Engine. The process is explained step by step below:

Algorithm 1: Policy Addition // Policy Update

Input: CT_{ABE_1}, $\mathbb{AP}$, MPK
Output: $nLayers$, CT_{ABE_2}

```
for i ← 0 to nLayers do
    if i == 0 then
        C ← CT_ABE1
        R' ← H(C‖AP_i)
        C_i ← Enc_ABE_CP(MPK, AP_i, C, R')
        nLayers++
                /* nLayers reflects the current amount of encryption
         layers.  */
    else
        R' ← H(C_{i-1}‖AP_i)
        C_i ← Enc_ABE_CP(MPK, AP, C_{i-1}, R')
        nLayers++
CT_ABE2 ← C_i
```

1. The data owner chooses a random AES-GCM key $SK_{sym} \in \{0,1\}^n$ and nonce $r \in \{0,1\}^n$. The data owner generates a new SK_{sym} for every piece of data they create and encrypt.

2. The data owner applies a modification of FO by including AP_1 in R such as $R \leftarrow H(SK_{sym} \| r \| AP_1)$.

3. FO also requires encrypting r, so it is appended to SK_{sym} which results in $PT \leftarrow (SK_{sym} \| r)$. PT is then used as input in CP-ABE along with AP_1. This encryption process is summarized as follows: $CT_{ABE_1} \leftarrow Enc_{ABE_{CP}}$ (MPK, AP_1, PT, R). The performed FO transformation ensures the CCA Security, which, as mentioned in Section 5.2.2, protects SK_{sym} against passive and active attackers.

4. After encapsulating SK_{sym}, the original message, M, is encrypted using AES-GCM. To guarantee the integrity of the encrypted SK_{sym}, the header of CT_{ABE_1} is used as AES-GCM's encrypted data. Thus, the result of encrypting M is $CT_{AES} \leftarrow Enc_{AES}$ (SK_{sym}, M, AAD).

5. Finally, the devices send the resulting ciphertext $CT_1 \leftarrow (CT_{AES}, CT_{ABE_1})$ to the Internal CT Engine.

Once the first phase is completed, the Internal CT Engine generates CT_2 during the second phase. CT_2 is generated using the updatable policies contained in $\mathbb{AP}$.

1. The Internal CT Engine requests $\mathbb{AP}$ to the Policy Database.
2. The Engine inputs CT_{ABE_1} in Algorithm 1, which outputs CT_{ABE_2}. This same algorithm is the one used for Policy Update. The encryption process presented in the algorithm is explained step by step below:
   (a) The Internal CT Engine takes $\mathbb{AP}$.
   (b) For the first iteration, it takes the inputed CT_{ABE_1} and proceeds to rename it C.
   (c) Then, for each policy AP_i, it creates $R' \leftarrow H(C \| AP_i)$.
   (d) With R' generated, the Internal CT Engine applies the CP-ABE encryption resulting in C_i.
   (e) When $i = nLayers$, the iterations finish.
   (f) Finally, Algorithm 1 produces $CT_{ABE_2} \leftarrow C_i$.
3. The Internal CT Engine returns $CT_2 \leftarrow (CT_{AES},\ CT_{ABE_2})$.
4. It finally sends CT_2 to the CT Database alongside information about the total number of layers, $nLayers$.

Time-Based Encryption

As introduced in Section 5.3.1, a key revocation system must also be deployed when setting up an encryption system. Therefore, the third phase of the encryption consists of time-based encryption that prevents users with a SK_{ABE} that predate the last security incident from accessing the information.

Security incidents are events that can damage or compromise data security. Not all incidents involve direct damage, but all pose a potential risk. Accidental key disclosures, system intrusions, or data breaches are security incidents that require SK_{ABE} renewal. However, it is worth noting that it should also be performed after certain security events, such as the installation of security patches or periodic key renewals.

To ensure that every SK_{ABE} has been issued after the last security incident, each time a user requests a ciphertext, the External CT Engine performs time-based encryption of the requested CT_2. The result of this operation, CT_3, is returned to the supply chain partner who requested the data.

1. The engine defines a new AP_{time} that requires SK_{ABE} to have been generated after the last security incident. For this purpose the policy takes the form of $AP_{time} \leftarrow (T_{SK} > T_{incident})$. In it, $T_{incident}$ is the timestamp of the last registered security incident in the system.

2. The External CT Engine generates R_{time} such as $R_{time} \leftarrow H(CT_{ABE_2} \| AP_{time})$.

3. CT_{ABE_2} is re-encrypted, generating $CT_{ABE_3} \leftarrow Enc_{ABE_{CP}}(MPK,\ AP_{time}, CT_{ABE_2}, R_{time})$.

4. Finally, the engine sends the consumer the resulting ciphertext $CT_3 \leftarrow (CT_{AES}, CT_{ABE_3})$ and the amount of policies contained in it, $nLayers$.

Decryption and Revocation

Figure 5.8 shows the modifications made to the ETSI scheme to generate the Multi-Layered CP-ABE decryption scheme. Whenever a user of the value chain requests a CT, what they receive is $CT_3 = (CT_{AES},\ CT_{ABE_3})$, which is encrypted under a policy of the form $\mathbb{AP} = (AP_1,\ AP_2,\ \ldots,\ AP_{nLayers},\ AP_{time})$. AP_1 is the immutable policy under which SK_{sym} has been encrypted. Therefore, to decrypt M, the user must first retrieve the symmetric key by decrypting CT_{ABE_3}.

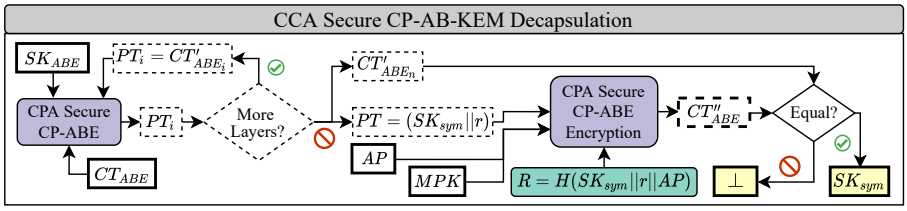


Fig. 5.8 Multi-Layered CP-ABE decryption.

In addition to this, encryption has been performed by applying FO to ensure the CCA security of SK_{sym}. FO's approach to obtaining CCA security requires an encryption during the decryption process to verify that decryption operations are correct. This check is executed on CT_{ABE_1}.

The FO security check needs to cache the value of CT_{ABE_1} and then decrypt it. The value obtained is called PT' and it is assumed to be of the form $PT' = (SK'_{sym} \| r')$. The original r was a random value, and CCA security depends on its correct decryption. If the decryption operation has been successful, r' is equal to r. Therefore, R' can be generated using r', AP_1, and SK'_{sym}. If R' is used with the rest of the decrypted values in a new encryption process, this results in CT'_{ABE_1}. FO compares CT_{ABE_1} and CT'_{ABE_1}, and only returns SK_{sym} to the user if both CTs are equal. If not, the user gets $\perp$.

The process of message exchange during decryption is presented in Figure 5.9. This Figure also includes the time encryption that generates CT_3 to reflect every step since

Algorithm 2: Layered Decryption // Policy Revocation

Input: CT_{ABE_3}, $nLayers$, SK_{ABE}
Output: SK'_{sym}, r', AAD

$C \leftarrow CT_{ABE_3}$
for $i \leftarrow 1$ *to* $nLayers$ **do**
  $C_i \leftarrow Dec_{ABE_{CP}}(MPK, SK_{ABE}, C_{i-1})$
$CT_{ABE_1} \leftarrow C_i$
$AAD \leftarrow ExtractHeader\,(CT_{ABE_1})$
$(SK'_{sym} \| r') \leftarrow Dec_{ABE_{CP}}(MPK, SK, CT_{ABE_1})$

a user requests a piece of information until it is decrypted. The decryption process and algorithm are also described below.

$Decryption(CT_3, SK_{ABE}, nLayers) \rightarrow M$

1. The Consumer starts by applying Algorithm 2 to CT_{ABE_3}, which outputs SK'_{sym}, R' and AAD. This algorithm can be used for Policy Revocation of i layers until the immutable policy AP_1 is reached.
   (a) The Supply Chain Partner takes CT_{ABE_3} and loads it as C_i.
   (b) For every layer until $nLayers$, it decrypts C_i by using its SK_{ABE}.
   (c) For the layer $nLayers$, the decryption of C_i returns CT_{ABE_1}. CT_{ABE_1} was the original ciphertext produced by the data owner, and thus contains the SK_{sym} used to generate the corresponding CT_{AES}.

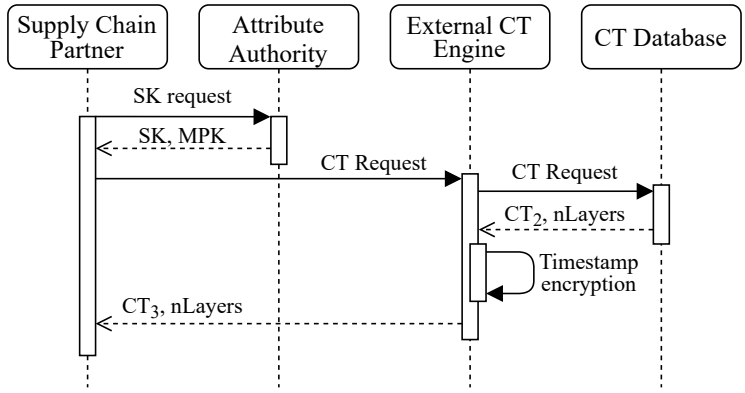


Fig. 5.9 Decryption message exchange.

(d) The user extracts the header from CT_{ABE_1} and stores it to use as AAD.

(e) Decrypting CT_{ABE_1} returns $(SK'_{sym} \| r')$.

(f) Finally, Algorithm 2 returns AAD, SK'_{sym} and r'.

2. The Supply Chain Partner performs the security encryption required by FO by defining R' as $R' \leftarrow H(r' \| AP' \| SK'_{sym})$.

3. A new $PT' = (r' \| SK'_{sym})$ is defined, and the security encryption computed as: $CT'_{ABE_1} \leftarrow Enc_{ABE_{CP}}(MPK, AP', PT', R')$.

4. If $CT'_{ABE_1} == CT_{ABE_1}$, the consumer obtains SK_{sym}; if not, it obtains $\perp$.

5. With SK_{sym}, the consumer can obtain the original message $M \leftarrow Dec_{AES}$ $(SK_{sym}, CT_{AES}, AAD)$. Note that each M is encrypted with a different SK_{sym}, so the decryption process has to be performed for every required CT.

5.4 Architecture Evaluation

This chapter has so far covered all the technical requirements defined in Section 5.1. However, the solution's efficiency can only be tested experimentally. Thus, this section presents a testbed for the roles defined in Section 5.3.2 executing Multi-Layered CP-ABE as defined in Section 5.3.3.

5.4.1 Experiment Definition

This section defines the experiments carried out to verify the fulfillment of **TR 2.2**.

Ciphertext Size Measure

CP-ABE schemes generate ciphertexts larger than the original messages, and the same is true for Multi-Layered CP-ABE. Ciphertexts must be temporarily stored in the devices that generate them; thus, their expansion cannot be too large. Furthermore, sometimes they are sent using narrowband technologies, which benefit from smaller-sized messages. Thus, quantifying the growth of the ciphertext is crucial to develop a feasible solution for manufacturing environments. Two experiments are conducted to study the expansion of the ciphertext. In **Experiment 1**, three scenarios are compared to show the overall ciphertext size evolution of Multi-Layered CP-ABE.

- **Scenario S1**: It is the CP-ABE case, the baseline. It involves the application of the ETSI scheme using W11. To make it comparable with the following scenarios, in this scenario, a single policy contains all the attributes in the form $AP = (att_1\ AND\ att_2\ AND\ \ldots\ AND\ att_n)$.
- **Scenario S2**: The iterative case. This scenario studies the result of applying several access policies directly over the original ETSI scheme. It involves iteratively encrypting the original message using the CCA Secure CP-ABE layer. Each policy has three attributes: $AP = (att_1\ AND\ att_2\ \ AND\ att_3)$.
- **Scenario S3**: Application of Multi-Layered CP-ABE. The algorithm is applied as described in Section 5.3, in which additional layers are only added to the symmetric key encryption. As in **Scenario S2**, each AP has three attributes.

Experiment 2 studies the influence of the expansion suffered by CT_{ABE} on the final $CT = (CT_{AES} \| CT_{ABE})$. For this purpose, Multi-Layered CP-ABE is applied to the same message but with a different number of attributes in $\mathbb{AP}$. Then, the CT size is measured, distinguishing which part of the final size comes from CT_{AES} and which from CT_{ABE}.

The purpose of the two experiments used to measure the size of CT is to test whether the size of the generated CT poses a limitation to the applicability of Multi-Layered CP-ABE in IIoT devices.

Encryption Times Measure

The fulfillment of **TR 2.2** can be affected not only by the size of the generated ciphertext but also by the time required by the encryption operations. Therefore, analyzing the speed of cryptographic operations is crucial to verify the feasibility of the proposed solution.

IIoT devices (fulfilling the role of data owners) encrypt two elements: M, the original message, and the SK_{sym} used in AES-GCM. Although the time required to encrypt M with AES-GCM depends on the size of said M, that time is not affected by Multi-Layered CP-ABE. Meanwhile, SK_{sym} is encrypted with Multi-Layered CP-ABE, and the size of the resulting CT_{ABE} is affected by the number of policies contained in $\mathbb{AP}$. To analyze the efficiency of the solution, **Experiment 3** studies the time required for encryption in three scenarios, which reflect the different approaches to deploying Multi-Layered CP-ABE according to the roles defined in Section 5.3.2.

- **Scenario T1:** Data owners apply the whole policy, represented with red triangles. This case reflects the result of the data owner (i.e., RPI0) applying the entire

policy $AP = (Att_1\ AND\ Att_2\ AND \ldots\ AND\ Att_n)$ in which n = size of $\mathbb{AP}$.

- **Secenario T2:** The encryption is performed entirely by the Internal CT Engine, without any intervention by the data owner and thus not achieving E2E security. It is represented with blue dots. This case reflects the result of the Internal CT Engine (i.e., RPI4) applying $\mathbb{AP}$ using Multi-Layered CP-ABE. Each of the APs contained in $\mathbb{AP}$ have three attributes, e.g., $AP = (Att_1\ AND\ Att_2\ AND\ Att_3)$.

- **Scenario T3:** Data owners and the Internal CT Engine work together to encrypt data, represented with orange squares. This case is the one designed for Multi-Layered CP-ABE. In it, Data Owners (DOs) generate CT_{ABE_1} and the Internal CT Engine generates CT_{ABE_2}. The DOs apply $AP_1 = (Att_1\ AND\ Att_2\ AND\ Att_3)$. Then the Internal CT Engine applies the rest of APs contained in $\mathbb{AP}$.

Comparing **Scenario T1** with **Scenario T3** illustrates the amount of work offloaded to the Internal CT Engine, and **Scenario T2** shows how the size of $\mathbb{AP}$ affects the Internal CT Engine. Finally, it should be noted that the time needed to add policies represents the time it takes to update CT_{ABE} since the same device performs the update using the same algorithm.

5.4.2 Testbed Setup

As discussed in Chapter 4, the library chosen to implement CP-ABE in industry is OpenABE. A thorough analysis of the library has revealed that OpenABE implements the CCA Secure CP-ABE scheme proposed by the ETSI and uses industry-grade cryptographic functions approved by the NIST. Therefore, since the use of OpenABE is **TR 2.2** compliant, the practical implementation of Multi-Layered CP-ABE is based on this library.

The first experiment measures the size of the generated CT_2 to determine whether the generated ciphertext sizes are manageable by an IIoT device. Therefore, since the ciphertext size is only related to the algorithm and not the device that implements it, a preliminary experiment is performed on an Ubuntu virtual machine with an allocated RAM of 4 GB and AES-GCM with a 256 bits SK_{sym}. The chosen M is a real-case manufacturing data stored in 160 kBytes JSON with the form shown in Listing 1.

Once the ciphertext expansion has been analyzed as a function of $\mathbb{AP}$ complexity, the rest of the tests are performed on a testbed that implements the following roles, previously defined in Section 5.3.2:

- **Data Owners**: The tasks assigned to this role are performed by a RPI0 with Raspbian Stretch. It has 512MB RAM, a single-core ARMv6, and Wi-Fi. It is

```
{
    "ts": <timestamp>
    "device": <Device ID>,
    "metadata": <Device Metadata>,
    "data": <data>
}
```

Listing 1: Industrial data to be encrypted.

thus a good representation of an IIoT device.

- **Internal CT Engine**: A RPI4 with 32 bits Ubuntu Server TLS. The RPI4 has a 8GB LPDDR4-3200 SDRAM and a Quad core Cortex-A72 (ARMv8).
- **Consumer**: This role is performed by an Ubuntu Virtual Machine with an allocated RAM of 4GB within a host machine with an Intel Core i7-8850H CPU processor.

The testbed has been used for **Experiment 2** and **Experiment 3**. The testbed has the same set-up as the virtual machine: a 160kByte M, Multi-Layered CP-ABE implemented in OpenABE, and AES-GCM with a 256-bit SK_{sym}. As for the time measurements, they have been obtained by performing each operation 500 times and calculating the average operation time. Moreover, to correctly show the evolution of the results, we have considered up to 15 APs consisting of three attributes in the form $AP = (att_1\ AND\ att_2\ AND\ att_3)$. Thus, for the worst case, a total of 45 attributes are used.

5.4.3 Results

Ciphertext Size Results

Figure 5.10 shows on the y-axis the size of $CT_2 = (CT_{AES} \| CT_{ABE_2})$ in kBytes, and on the x-axis, the number of APs in $\mathbb{AP}$. It shows that **Scenario S2** generates an exponential growth of CT_2. This can be seen in the graph since, for an $\mathbb{AP}$ consisting of 5 APs, CT_2 has a size of 697 kBytes. However, when $\mathbb{AP}$ is formed by 15 APs, the size of CT_2 increases to 12,420.49 kBytes. That is, it has an expansion of 1682%. In contrast, those two same cases in **Scenario S3** and **Scenario S1** give notably more efficient results. **Scenario S3** increases from 226 kBytes to 281 kBytes, a 24% increase, and **Scenario S1** increases from 221 kBytes to 226 kBytes, an increase of just 2.5%. Therefore, the growth of Multi-Layered CP-ABE is controlled, contrary to **Scenario 2**.

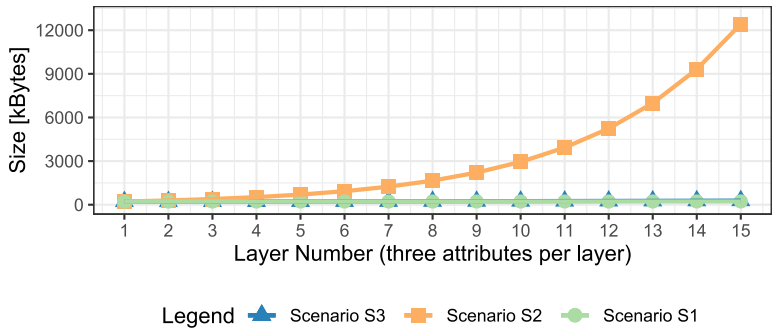


Fig. 5.10 CT_2 size evolution in three scenarios: **Scenario S1** (green circle), **Scenario S2** (orange square) and **Scenario S3** (blue triangle).

On the other hand, the time required by **Scenario S2** is compared with **Scenario S1**, and the results show that the **Scenario S2** is noticeably inefficient. For the case of 15 *AP*s, **Scenario S2** generates a CT_2 5387% larger than **Scenario S1**. **Scenario S3**, on the other hand, generates a CT 24% larger **Scenario S1**. Therefore, Multi-Layered CP-ABE generates acceptable CT_2 sizes for an IIoT device.

Finally, as explained in Section 5.4.1, we analyze the impact of CT_{ABE_2} on $CT_2 = (CT_{AES} \| CT_{ABE_2})$. Figure 5.11 shows the total length of a CT_2 requested by a consumer for an original M of 160 kBytes. The figure shows the distribution in kBytes of CT_{ABE_2} and CT_{AES} as a function of the number of *AP*s in $\mathbb{AP}$.

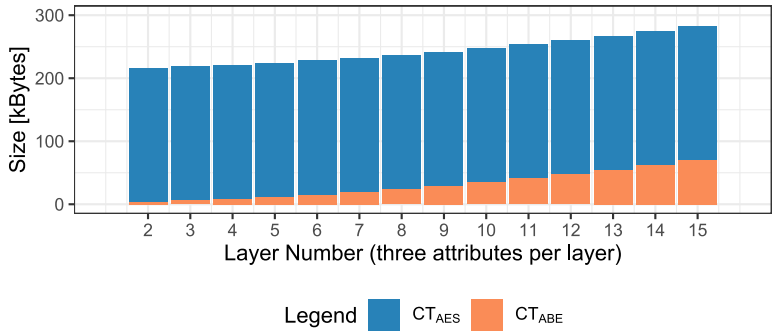


Fig. 5.11 Sizes of the generated CT_{ABE_2} and CT_{AES}.

Thus, Figure 5.11 shows how CT_{ABE_2} has a significantly larger size than the original 256 bits SK_{sym}. However, the much larger size of CT_{AES} makes the expansion of CT_{ABE_2} less relevant. Thus, since $CT_{ABE_2} << CT_{AES}$, the solution is considered suitable for IIoT devices. The algorithm used to add policies is the same as the one used

to update them, so it is proven that the update does not generate a CT_{ABE} larger than CT_{AES} for a reasonable number of attributes.

Encryption Times Results

Figure 5.12 shows the time required to encrypt SK_{sym} on the y-axis and the number of *AP*s contained in *AP* on the x-axis. The graph only shows the time required to generate CT_{ABE_2}. This is because Multi-Layered CP-ABE always uses a 256-bit SK_{sym}, whose encryption is directly influenced by *AP*. In contrast, the size of *M* is variable and depends on the system's needs. Therefore, it is more relevant to measure the CT_{ABE_2} generation time because of its impact and predictability.

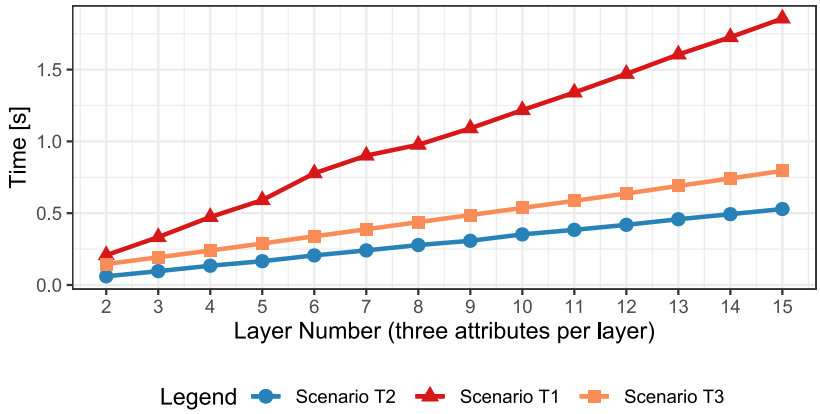


Fig. 5.12 Total time required to generate CT_{ABE_1} and transform it to CT_{ABE_2} in **Scenario T1** (red triangle), **Scenario T2** (blue dot) and **Scenario T3** (orange square).

The graph shows that the combined use of data owner and Internal CT Engine (Scenario T3) is more efficient than using data owners alone (**Scenario T1**). For an $\mathbb{AP}$ with 15 *AP*s, **Scenario T1** takes 1.85 s to generate CT_{ABE_2}. In contrast, for the same situation, **Scenario T3** generates CT_{ABE_2} in 0.79 s, a time reduction of 57%. Offloading computational expensive operations to more powerful devices is a common way to alleviate IIoT devices. However, this is usually accompanied by a security loss. In contrast, with the implementation of Multi-Layered CP-ABE, E2E confidentiality and integrity are maintained using AP_1.

Additionally, it is also observed that the involvement of the IIoT device delays the generation of CT_{ABE_2}. Note that for 15 *AP*s, **Scenario T3** takes 0.79 s, whereas **Scenario T2** takes 0.52 s. The delay added by the IIoT device is more than acceptable for obtaining confidentiality and E2E integrity in return. Finally, to visualize how the

encryption times are distributed between CT_{ABE_1} and CT_{ABE_2}, we present Figure 5.13.

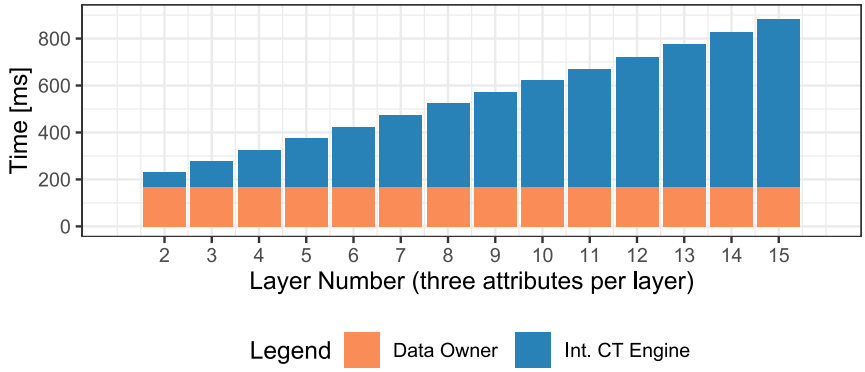


Fig. 5.13 Total time required by the data owner to generate CT_{ABE_1} (orange) and the total time required for the Internal CT Engine to generate CT_{ABE_2} (blue).

Figure 5.13 shows the total time required for a data owner to create CT_{ABE_1} and for Internal CT Engine to generate CT_{ABE_2} in **Scenario T3**. The time consumed by the data owner is constant, while the time consumed by the Internal CT Engine grows linearly as *AP*s are added to $\mathbb{AP}$. This linear growth implies that the required time to add more layers is predictable and that encryption times do not escalate out of control.

5.5 Summary

Section 5.1 proposed three technical requirements to verify compliance with **R2** and two to verify compliance with **R3**. Adherence to these technical requirements is complex, and various aspects must be considered. Table 5.2 shows a summary of the sections in which each has been evaluated, and they are discussed below.

- Section 5.2.1 summarized the requirements ABE schemes must fulfill to be deployed in environments with IIoT devices and comply with **TR 2.2**.
- Section 5.2.2 presents ETSI' CCA Secure CP-ABE scheme. CCA Security guarantees E2E security (and thus **TR 2.2**) by protecting SK_{sym} against active and passive attackers. ETSI's scheme can be applied by IIoT devices, favoring **TR 2.1**.
- Section 5.3.3 presents Multi-Layered CP-ABE. This scheme has a policy update and revocation system based on ETSI's CCA Secure CP-ABE scheme. The proposed scheme fulfils **TR 2.1**, **TR 3.1** and **TR 3.2**.

Table 5.2 Technical Requirement fulfillment during the chapter.

	TR 2.1	**TR 2.2**	**TR 2.3**	**TR 3.1**	**TR 3.2**
ETSI Industrial Requirements *(Section 5.2.1)*	□	■	□	□	□
Application of a CCA Secure Scheme *(Section 5.2.2)*	■	■	□	□	□
Multi-Layered CP-ABE Role Definition *(Section 5.3.2)*	□	■	■	□	□
Multi-Layed CP-ABE Algorithms (Section 5.3.3)	■	□	□	■	■

- Section 5.3.2 presents the definition of the roles and how Multi-Layered CP-ABE can be integrated into an Industrial environment (**TR 2.2**). The proposed roles are independent of the data type and thus fulfill **TR 2.3**.
- Finally, Section 5.4.3 proves the feasibility of our Multi-Layered CP-ABE. The experiments deploy Multi-Layered CP-ABE on a RPI0 emulating IIoT devices and on a RPI4 emulating devices with higher computational capabilities.

Chapter 6

Attribute-Spoofing Prevention

The previous chapter discussed an information exchange system between value chain partners based on CP-ABE. The proposal also solved the well-known policy update problem inherent to ABE schemes. However, Section 5.2 of the same chapter presented the high-level requirements requested by ETSI for deploying an ABE system in the industry. Among those requirements was attribute management, which Chapter 5 does not address.

This chapter completes the proposal of the previous chapter, focusing on attribute management. Since CP-ABE relies on user attributes to generate private keys, it is vital to establish a system to manage these attributes. However, attribute management in distributed systems like the value chains defined in this thesis is a complex problem. For example, some solutions, such as [21], consider that attributes are distributed and managed by the attribute authority. Such solutions also consider that the authority always generates a decryption key that accurately reflects the users' real privileges. However, this approach can lead to performance bottlenecks [99] if the authority's computational capabilities are limited, trust issues between different partners in the value chain, and a single point of failure by centralizing all the management on a single node.

We have identified attribute spoofing among the possible vulnerabilities derived from centralizing knowledge management in a single node. Attribute spoofing, like its namesake identity spoofing, seeks to deceive the attribute authority. The purpose of the deception is to obtain a SK_{ABE} that does not reflect their real attributes. For example, the system presented in the previous chapter would be vulnerable to this situation because the chapter does not establish any security measures to address it.

Therefore, to further enhance the system in Chapter 5, it is necessary to have a system that allows distributed attribute management, freeing the attribute authority from storing and managing this information internally. The literature review [185] [186] has shown that solutions like Distributed Ledger Technologiess (DLTs) can provide a secure

attribute management system capable of preventing attribute impersonation. They also guarantee integrity, immutability, and auditability, which builds trust among all chain members.

The rest of the chapter is organized as follows: Section 6.1 presents the technical requirements needed to fulfill **R4**. Section 6.2 defines attribute spoofing and the attack vectors capable of exploiting it, and Section 6.3 discusses the proposed solution. Section 6.4 evaluates the proposal, and the chapter closes with the summary in Section 6.5.

The research conducted to write this chapter has been included in two papers:

- *"Are you what you claim to be?" Attribute Validation with IOTA for Multi Authority CP-ABE* [28] that has been presented at *BLOCKCHAIN 2022*.
- *"Trust your users as far as you can validate them: Secure Attribute Retrieval for ABE Schemes"* [27] that has been submitted to *"Smart Cities Journal"* in 2022.

6.1 Requirements

Section 3.3 introduced requirement *"**R4.** The solution shall provide authentication and data access rights validation"* For this purpose, this section defines the technical requirements that have been considered the most relevant to validate its fulfillment.

- **TR 4.1. The solution shall provide distributed attribute storage:** The attribute storage system must have no single-point failures. Thus, it will benefit from distributed storage solutions that also allow for redundancy. The objective is for the authority to have the attributes always available.
- **TR 4.2. The solution shall validate the attributes ensuring the trust of the system.** Although not a native capability of CP-ABE, validating user attributes protects value chain members and builds trust in the data exchange. Attribute validation cannot rely solely on the attribute authority since that can result in a performance bottleneck [99]. Furthermore, there must be an efficient and scalable authentication system capable of avoiding unnecessary trust relationships to maximize trust.
- **TR 4.3. The solution shall provide reliable and auditable attribute management:** The distributed storage of attributes must be accompanied by a system that

enables its auditability. Hence, the integrity of the attributes is protected, and the nodes that store them are guaranteed not to make unauthorized modifications.

- **TR 4.4. The solution shall be suitable for IIoT devices:** The solution proposed in Chapter 5 does not define the computational power of the device that generates the user keys. Therefore, the attribute validation system must be deployable on all kinds of devices, regardless of their computational capabilities.

6.2 Attribute Spoofing Definition and Attack Vectors

Information exchange solutions based on CP-ABE usually overlook how the attribute authority determines the users' attributes. Instead, it is presumed that the authority always generates the users' keys correctly [187] [188] or that it already knows the attributes because it is the one managing them [21]. However, the latter can result in performance bottlenecks in systems with a large attribute universe and many users [99]. To cope with this, some authors consider that the authority can take those attributes from LDAP services, enterprise databases, or SAML authorities [189] [190]. However, they do not detail how to carry out such integration, nor do they consider that there must be a shared attribute universe between the different companies.

The main drawback of the approaches presented above is that they do not consider the risk of attribute spoofing. Attribute spoofing forces the authority to generate keys based on attributes the user does not possess. This attack can be based on escalation, i.e., users request a key with higher privileges than they have; or on pure forgery: attackers from outside the system force the authority to grant them attributes they do not possess. Therefore, this section analyses how this vulnerability can be exploited in order to design a system to prevent it. The analysis is based on the following assumptions.

- **Assumption 1:** The authority does not know the defined attribute universe $\mathbb{U}$.
- **Assumption 2:** The authority does not know users' attribute sets, $\mathbb{A}$.
- **Assumption 3:** Every legitimate user belongs to one of the participating companies in the value chain.
- **Assumption 4:** Every participating company knows which users depend on them and what attributes they have.

The following subsections define the two attack vectors identified as capable of taking advantage of the established assumptions. The attack vectors are defined considering that the main attack points for attackers are either the attribute authority or the

shared attribute storage database. Companies are also considered to have implemented security measure that prevents them from being attacked.

6.2.1 Attack Vector 1: Directly Interfering with the Attribute Authority

We number this attack vector as **AV 1**. The key generation in CP-ABE is randomized to guarantee that users with the same set $\mathbb{A}$ get a different CP-ABE private key (SK_{ABE}). Therefore, colluding keys and combining them to obtain a new SK_{ABE} with higher privileges does not work. Instead, if a malicious user or an attacker wants to obtain a SK_{ABE} according to an illegitimate $\mathbb{A}$, they can try to obtain it from the attribute authority. Two ways of exploiting this vector have been identified: by a malicious user and by an external attacker.

In the case of the malicious user, assume they have a legitimate attribute set $\mathbb{A}$. However, when interacting with the attribute authority, they request a key for attribute set $\mathbb{A}'$, which contains more attributes than $\mathbb{A}$. Therefore, the authority delivers a private key that entitles the user to access data they should not be able to access. The success of this attack is based on the following:

- It is a legitimate user, so it passes the authentication process with the Authority.
- Based on **Assumption 2**, the authority does not know what attributes the user has and therefore does not distinguish between $\mathbb{A}$ and $\mathbb{A}'$.

The case of the external attacker is based on credential theft. If an external attacker has been able to steal a legitimate system user's credentials, they can interact with the authority and bypass the authentication process. The attacker could then request keys for any possible attribute set. The success of this attack lies in the following:

- Based on **Assumption 2**, the authority does not know what attributes the original user to whom the credentials belong has and, therefore, will generate any key requested by the attacker.

Finally, it is necessary to consider that the generation of keys in CP-ABE is randomized, so examining them is insufficient to detect overprivileged keys.

6.2.2 Attack Vector 2: Interfering with the Attribute Storage

We number this attack as **AV 2**. As stated by **Assumption 3**, every user belongs to one of the companies in the value chain. Furthermore, according to **Assumption 4**, companies know which attributes their users hold. Therefore, a straightforward solution would

be for companies to store attributes in a shared centralized database so the attribute authority can retrieve them. However, attackers can interfere with attribute storage by compromising the database and modifying the information related to users' attributes, as Figure 6.1 shows. Thus, additional security mechanisms allowing the attribute authority to retrieve the attributes reliably, ensuring their integrity and confidentiality, are needed.

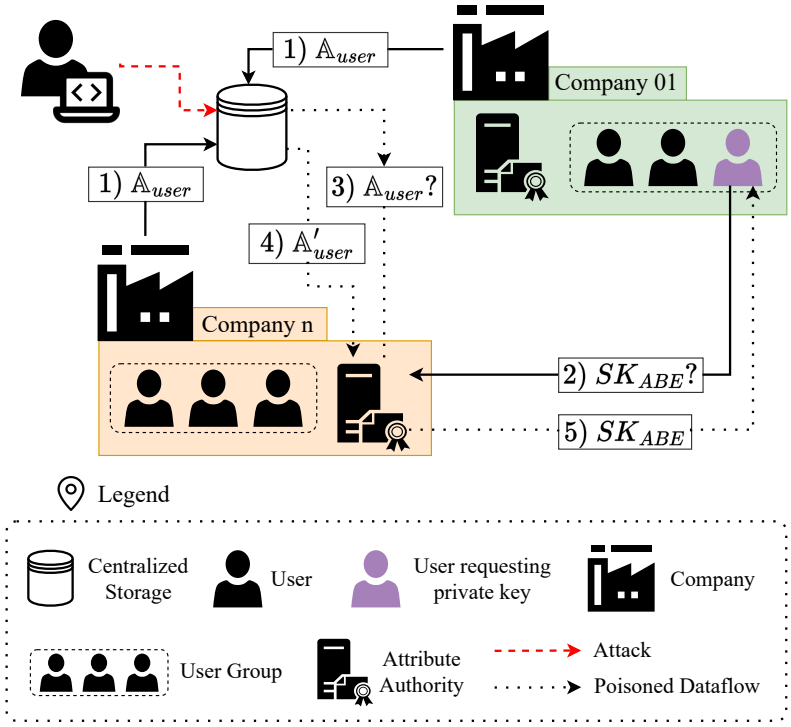


Fig. 6.1 AV 2 - Interfering with the Attribute Storage. The attacker poisons the database, resulting in the user getting SK'_{ABE} instead of SK_{ABE}.

This attack can create a significant disruption. The attribute authority would not know that they are generating incorrect keys, and the users and their companies would not be aware that users' private keys do not reflect their privileges. In addition, since key analysis does not provide information on the attributes contained in the keys, it is not easy to detect which keys in the system have been altered and which have not. The success of this attack is based on the following:

- According to **Assumption 1**, the attribute authority does not know the attribute universe, so modification on existing attributes would go unnoticed.

- According to **Assumption 2**, the authority does not know users' attributes.

Therefore, a system that guarantees that the attribute authority generates correct keys to legitimate users, that fake users cannot obtain a SK_{ABE} is needed. Furthermore, attributes must be auditable to detect when false information has been stored and by whom.

6.3 Attribute-Spoofing Prevention System Definition

This section presents the proposed solution to prevent attribute spoofing. Table 6.1 summarizes the assumptions on which the attack vectors identified in the previous section are based and the consequences for the system.

Table 6.1 Attack vector, assumption, and solution summary.

Attack Vector	Assumption 1 2	Consequence 1 2 3 4	Technical Requirement 1 2 3
	1. Assumption 1 2. Assumption 2	1. Privilege Scalation 2. Privilege Reduction 3. SK_{ABE} Adquisiton 4. Cut-off Data Access	1. TR 4.1 2. TR 4.2 3. TR 4.3
AV 1	□ ■	■ □ ■ □	■ ■ □
AV 2	■ ■	■ ■ ■ ■	□ ■ ■

As seen in the table and previously mentioned, the success of **AV 1** is based on **Assumption 2**. The mitigation of this attack requires compliance with technical requirements **TR 4.1** and **TR 4.2**. The first requires distributed attribute storage, which allows the authority to retrieve this information from a database instead of relying on user messages. The second requirement, meanwhile, calls for attribute validation and user authentication. Attribute validation guarantees that the attributes come from a trusted source, generating confidence for the authority. User authentication guarantees that only legitimate users connect to the system.

Regarding **AV 2**, it is based on **Assumption 1** and **Assumption 2**, and it has more consequences than **AV 1**. As before, malicious users can escalate privileges by claiming more privileges than they have, and external attackers can force the system to grant them private keys. This time, however, privilege reductions and cutting off access to data stand out. These actions would be carried out by an attacker, who seeks to harm certain users by reducing their privileges or taking them away altogether. This attack vector

can be mitigated by complying with **TR 4.2** and **TR 4.3**. If attributes are validated, authorities can detect potential spoofings. Regarding auditability, it provides a record of attribute modification, enabling it to detect integrity violations, trace the changes, identify the attackers and discover the affected private keys.

The following subsections present the proposed solution to avoid attribute spoofing. Sections 6.3.1 and 6.3.2 describe how IPFS and IOTA are used to fulfill various technical requirements and prevent the identified attack vectors. Section 6.3.3 presents the proposal for user authentication, which is necessary to comply with **TR 4.2**. Finally, Section 6.3.4 integrates the solution with the information exchange system in Chapter 5.

6.3.1 IPFS for Attribute Distribution

TR 4.1 requires a distributed storage of attributes. The purpose of this distribution is twofold: to relieve the attribute authority from attribute storage and management and to make the information accessible at any time without single points of failure. Among the systems for information distribution, infrastructures based on Distributed Hash Tables (DHTs) have shown their potential in CP-ABE private key distribution [191]. The referenced paper combines a DHT infrastructure with a Secret Sharing Scheme (SSS) to distribute private keys. This chapter addresses the distribution of the attributes used to generate the private keys, not the distribution of the keys themselves. Even so, distributed storage solutions based on DHT show great potential to be part of the attribute spoofing prevention solution. To this end, one of the most prominent distributed systems based on DHTs is IPFS.

IPFS is a peer-to-peer protocol that offers content discovery through DHTs and can be used to establish a high-performance distributed storage model. IPFS solutions have no single point of failure, nodes do not need to trust each other, and every distributed file has a timestamp [192]. Another advantage of IPFS within a value chain is that there is no central server; instead, the data is distributed and stored in different locations. These properties have made IPFS one of the most supported solutions for distributed storage of industrial information [86] since it does not cause bottlenecks [193] and can be used by IIoT devices [194]. Because of this, the distributed attributes storage for the attribute spoofing prevention solution is built with IPFS.

The proposed solution is based on **Assumption 4**, in which companies manage their users' attributes. To do so, the companies in the value chain agree on an attribute universe called $\mathbb{U}$. Afterward, each company defines a set of attributes $\mathbb{A}$ for each of their users, such that $\mathbb{A}_{user} \subset \mathbb{U}$. With the $\mathbb{A}$ defined, companies stores them in IPFS, as Figure 6.2 shows. The process to store $\mathbb{A}_{users}$ in IPFS is detailed below:

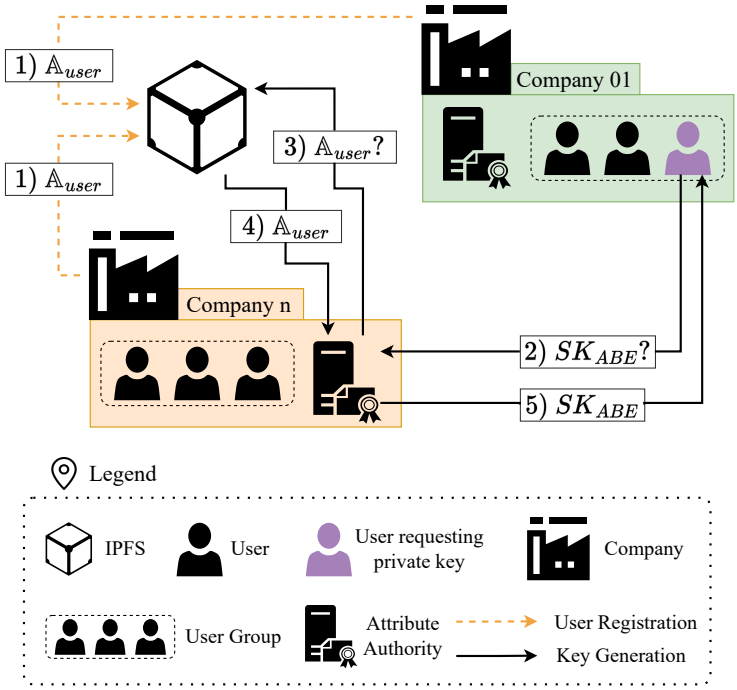


Fig. 6.2 CP-ABE attribute storage in IPFS.

1. During a preliminary phase, companies register their users by storing their users' set $\mathbb{A}$ in IPFS.
2. When users need a private key, they request SK_{ABE} from the attribute authority.
3. The attribute authority requests the user's attribute set from IPFS.
4. The attribute authority takes $\mathbb{A}$ as input to generate the user's SK_{ABE}.
5. Users get their SK_{ABE}.

Once the information has been stored in IPFS, it can be retrieved by any node that connects to the network, as long as the node knows the Content Identifier (CID) of the file it needs to retrieve. CIDs are generated from the hash of the content, which prevents data duplication and allows for data integrity verifications.

The content of the IPFS is retrieved by the attribute authority that needs it to generate the SK_{ABE}. The CID allows the attribute authority to detect if the file information in IPFS has been modified. In addition, IPFS relieves the attribute authority from managing the attributes and transfers that responsibility to the companies.

6.3.2 IOTA for Attribute Auditability

TR 4.3 requires attribute auditability. Attribute storage in IPFS provides distributed storage, and the hash from which the CIDs are generated allows the attribute authority to detect potential manipulations. However, while IPFS can detect modified content, it cannot track which user has modified it. In this regard, DLTs can provide the auditability and immutability that the system requires to prevent attribute forgery.

DLTs are composed of nodes that contain distributed, replicated and synchronized data. Nodes forego a central authority and instead agree on the ledger's state using a consensus protocol. Hence, DLTs are resilient and provide traceability to the attribute validation system [185]. It should be noted that, reading this description, the straightforward solution would be to directly use a DLTs for distributed storage. However, storing credentials directly on a DLTs requires a high-capacity networks since this operation generates several transactions [187].

There are many different DLT systems, each of which has unique virtues and capabilities. Therefore, choosing one whose performance and efficiency are suitable for the industrial environment is crucial. In general, the usefulness of DLTs in industry has already been proven, and its effect is considered positive in improving data confidentiality, privacy, and security in IIoT networks [195]. Therefore, the choice is not about proving the suitability of DLT technology but choosing the best one. In this sense, the literature considers that Directed Acyclic Graph (DAG)-type DLTs are the most promising for the industry due to their scalability and transaction speed [186] [196]. In fact, DAG-type DLTs are faster and more secure as transactions increase, which provides high performance [197].

Currently, the main DAG-based DLT is IOTA[1]. IOTA is specially designed for IoT and has proven applicable in a network formed by IIoT devices [198]. IOTA's feeless microtransactions and high throughput make it the selected DLT technology to build the attribute spoofing prevention system. Although less numerous than proposals with other technologies, there are already works that study how to combine IOTA and IPFS. The authors of [193] combine IOTA, IPFS, and Ethereum's Smart Contracts to create a preliminary architecture for data mining in intelligent transport networks. More directed to the industrial environment comes the proposal presented in [186], which seeks to maximize the efficiency of IOTA for IIoT devices. This last work is designed for production lines, so its application in a value chain context is appropriate. Matching the case presented in this chapter, the authors of [186] also seek the implementation of IOTA with IIoT devices. Therefore, their proposal seeks to maximize efficiency and reduce the work these devices with reduced capacities have to perform. In the proposal,

[1] https://www.iota.org/

the authors store the information in IPFS and the CID in IOTA.

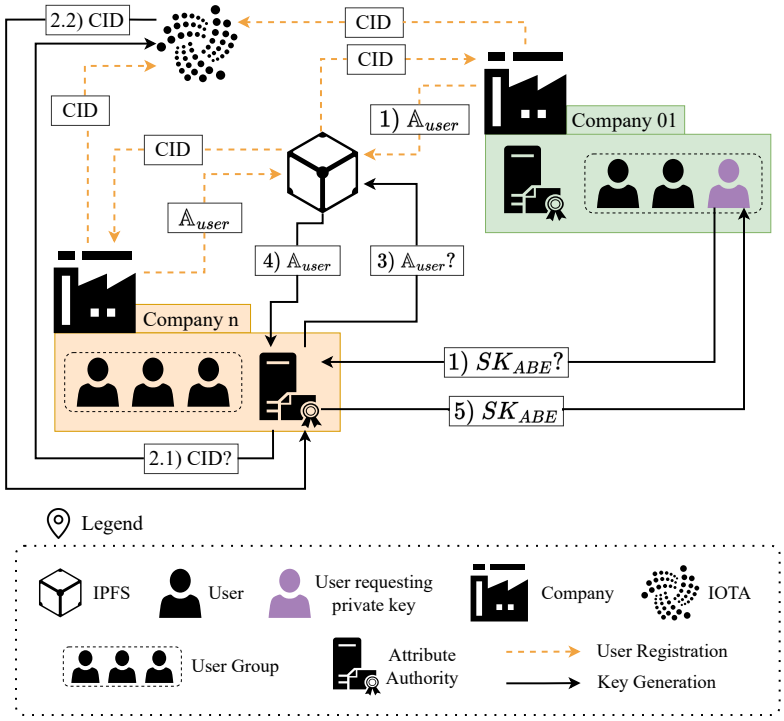


Fig. 6.3 CP-ABE attribute validation with IPFS and IOTA.

The attribute prevention architecture based on IOTA and IPFS is presented in Figure 6.3. As can be seen, it is an extension of Figure 6.2, and it is formed by the same two phases: user registration and private key retrieval. During registration (Algorithm 3), a company node uploads the attributes to IPFS and stores the CID in the IOTA Tangle. This secure storage ensures that the CID has not been modified since IOTA provides the auditability that IPFS alone cannot guarantee.

Algorithm 3: User Registration

Input: $\mathbb{A}$
Output: *msgID*
1 *Node = IPFS.Create*
2 *CID = Node.add* ($\mathbb{A}_{user}$)
3 *msgID = IOTA.send (CID)*

Once users' have their $\mathbb{A}$ stored, they can require their private keys from the attribute authority. The attribute authority generates users' private keys following Algorithm 4. The step-by-step process is defined below:

Algorithm 4: User SK_{ABE} Generation

Input: *msgID*
Output: SK_{ABE}
1 $CID = IOTA.retrieve(msgID)$
2 $\mathbb{A}_{user} = IPFS.retrieve(CID)$
3 $KeyGen(\mathbb{A},\ MPK,\ MSK) \rightarrow SK_{ABE}$

1. A user requests a private key SK_{ABE} from an attribute authority.
2. The attribute authority obtains CID from IOTA.
3. With the CID, the authority retrieves the content (i.e., user's $\mathbb{A}$) from IPFS.
4. If the retrieved content passes the integrity check, the attribute authority uses $\mathbb{A}$ as input to generate the user's private key SK_{ABE}.
5. Finally, the user gets their SK_{ABE}, according to the attribute set $\mathbb{A}$ their company conceded them.

Thus, with the combination of IPFS, IOTA, Algorithm 3, and Algorithm 4, the attribute-spoofing prevention system is implemented. The attribute authority does not have to manage all the information; instead, attribute management is distributed by the combination of IPFS and IOTA. Thanks to IPFS, the solution obtains auditability, which allows the system to know who has stored the information, and thanks to the CID, integrity violations in IPFS files can be detected. Finally, the distributed nature of both technologies (IPFS and IOTA) protects the system against single-point failures.

6.3.3 Federated Identity Management for User Authentication

Finally, **TR 4.3** also called for user authentication. The scenario considered in this chapter requires value chain users to authenticate with the attribute authority from which they require SK_{ABE}. Similarly, the attribute authority must identify users coming from different environments. This reduces the system's scalability by adding operations prior to key generation. Therefore, it is necessary to have an efficient and scalable authentication system. In addition, the no trust assumption [48] is deemed desirable: it should be unnecessary for different companies in the value chain to establish trust relationships between them.

The authentication system chosen for our attribute spoofing prevention system is FIM [199], whose behavior is shown in Figure 6.4. FIM allows users from different companies to use their company credentials to authenticate to different attribute authorities. This also implies that attribute authorities do not have to manage the credentials of

multiple users from different companies. Instead, they verify the token issued through an Identity Provider (IdP). This IdP establishes a trust relationship with the different companies and acts as an intermediary between the attribute authority and the companies. This way, attribute authority only has to manage the trust relationship with the IdP. The FIM message exchange is performed through the users' browser and is described below:

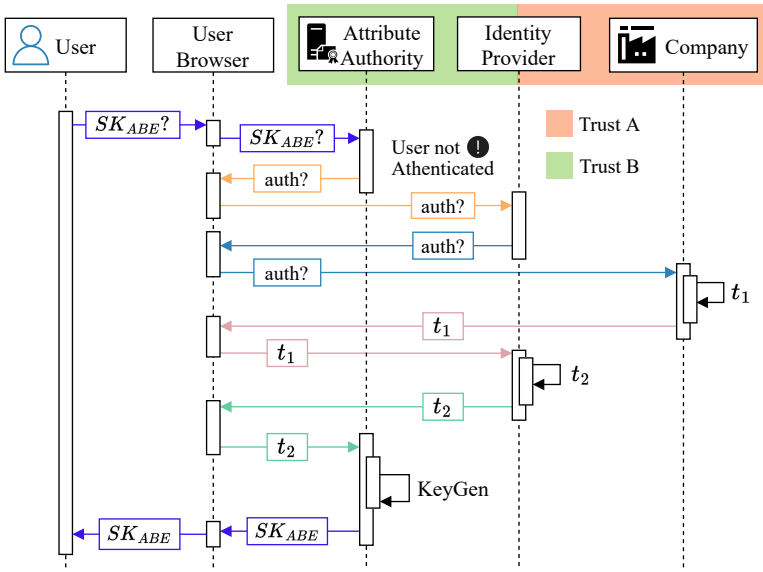


Fig. 6.4 FIM message exchange.

1. The user requests SK_{ABE} from the attribute authority through their browser.

2. If the authority detects that the user has not been authenticated, it triggers the authentication process by sending an *auth* authentication request through the user's browser.

3. The request is sent to the identity provider, the only role with which the attribute authority has a trusted connection during the authentication process.

4. The identity provider has a trusted connection with the various companies in the value chain. Thus, it redirects the authentication request to the company to which the user belongs via the user's browser.

5. The company to which the user belongs sends an authentication token t_1 to the identity provider that requested it.

6. The identity provider uses the token t_1 to generate a second token, t_2.

7. The identity provider relies on the user's browser to send token t_2 to the attribute authority.

8. Once the attribute authority receives t_2, it runs the Algorithm 4 presented in the previous section to generate the user's SK_{ABE}.

6.3.4 Integration in the Value Chain

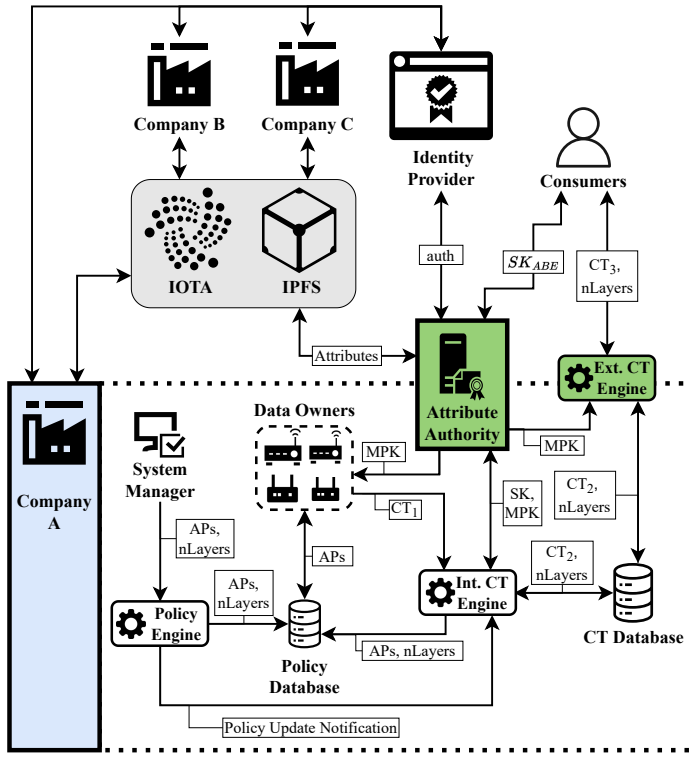


Fig. 6.5 Attribute spoofing prevention system for CP-ABE in a value chain.

Having defined the integration of IOTA and IPFS for attribute spoofing prevention, it is necessary to integrate the proposal into the E2E secure information exchange solution for the value chain. The previous chapter defined the data exchange functions and outlined the infrastructure's roles. Figure 6.5 shows how the solution proposed in this chapter is integrated with the data exchange architecture presented in the previous chapter to make the system resistant to attribute spoofing attacks.

By including the attribute spoofing prevention system, each company's attribute authority and the external *CT* Engine (in green) are the only points with which external users can interact. Although the figure only shows the case of Company A, every company shall expose its authority and external Engine.

Finally, Figure 6.5 also shows how trust relationships and communication channels are reduced to the minimum. The attribute authorities and companies are the only entities interacting with the attribute spoofing solution. Using FIM also makes the trust relationship among companies unnecessary since authentication goes through the identity provider. Finally, consumers retrieve their private key by only interacting with the attribute authority and the ciphertexts by only interacting with the External CT Engine.

6.4 Attribute-Spoofing Prevention System Evaluation

The previous sections introduced the security risks posed by attribute spoofing in CP-ABE schemes. Section 6.1 presents the technical requirements to comply with **R4**, which requires authentication and validation of access rights. Section 6.2 presents the two identified attack vectors capable of exploiting attribute spoofing. Table 6.2 summarizes how meeting the technical requirements prevents the identified attack vectors and which technology achieves that compliance. Meeting the technical requirements is necessary to consider **R4** fulfilled.

Table 6.2 Technical Requirement fulfilment. ▣ stands for not relevant.

Technical Requirement	Attack Vector 1. AV 1 2. AV 2		Proposed Solution 1. IOTA 2. IPFS 3. FIM		
	1	2	1	2	3
TR 4.1	■	□	□	■	□
TR 4.2	■	■	■	□	■
TR 4.3	□	■	■	□	□
TR 4.4	▣	▣	▣	▣	▣

The fulfilment of **TR 4.1**, **TR 4.2** and **TR 4.3** is analysed in Subsection 6.4.1, while the fulfilment of **TR 4.4** is experimentally verified in Subsection 6.4.2.

6.4.1 Qualitative Evaluation

TR 4.1 requires distributed attribute storage, and meeting this requirement by using IPFS deters **AV 1**. This attack depends on users transmitting their attributes to the authority. Therefore, although the authority still does not know which attributes belong to which users (**Assumption 2**), that information is provided by the companies storing the information in IPFS. Furthermore, users and attackers cannot claim a private key containing more attributes than they possess since they do not interact with the information stored in the IPFS. Therefore, the main consequences of **AV 1** (privilege escalation and illegal acquisition of SK_{ABE}) are stopped.

TR 4.2 requires attribute validation and user authentication. The ones that store information in IPFS are the original companies, which we presume are honest at the beginning. However, if one is compromised or an external attacker gets access to the stored information, the CID stored in IOTA changes. Thus, IPFS detects the integrity violation, and the attacker or the compromised node can be traced back using IOTA. In addition, knowing the set of compromised attributes also allows partners in the value chain to know which private keys have been compromised. Regarding user authentication, it can be provided by FIM based on **Assumption 3** and **Assumption 4**. Since the distributed storage is protected, the solution prevents **AV 2**. Attackers can no longer modify the information contained in the storage solution, preventing the consequence shown in Table 6.1. Furthermore, the attack is prevented even in the presence of **Assumption 1** and **Assumption 2**. The attribute authority does not need to know the attribute universe or users' attributes: all that information is retrieved from IOTA and IPFS.

TR 4.3 is fulfilled by combining IPFS with IOTA. As explained, IOTA provides traceability and, combined with IPFS, ensures attribute integrity and auditability. Auditability protects the system against AV 2 since data modifications can be traced back to the user who made them.

Therefore, the chosen combination of technologies meets most established technical requirements and avoids the identified attack vectors. However, the last technical requirement (**TR 4.4**) alludes to the suitability of the solution for IIoT devices. In order to validate this last technical requirement, the following subsections present the experimental evaluation.

6.4.2 Experimental Evaluation

The experimental evaluation assesses the fulfillment of **TR 4.4**. The experiment tests the time required by the authority to retrieve the attributes from the attribute spoofing

prevention system and use them to generate the users' private keys. The experiment uses a RPI4 with a 32-bit Ubuntu Server TLS and 8GB of RAM as the attribute authority, while data has been uploaded to IOTA and IPFS using the WSL running Ubuntu 20.04.5 LTS. IPFS stores a local copy of the uploaded data; thus, uploading it from a different device is crucial to measure the time correctly. Users' attributes are stored in JSON files, following the structure shown in Listing 2.

```
{
    "issuer": <Company ID>
    "userID" : <User ID>,
    "attribute" : "(Attr1||Attr2||...||Attrn)",
    "timestamp": <timestamp>
}
```

Listing 2: JSON with users' attribute set $\mathbb{A}$.

We name the time elapsed between the attribute authority requesting the user's attributes and obtaining them as T_{AS}. The obtained user's attribute set is $\mathbb{A}$ = (Attr1‖ Attr2‖Attr3‖Attr4‖Attr5‖ ⋯ ‖Attrn). Measurements are performed from n = 1 to n = 20. Afterward, the native OpenABE benchmarking tool measures the time required to generate SK_{ABE} in W11 according to the $\mathbb{A}$ retrieved from IPFS. We run 100 iterations for each $\mathbb{A}$ and calculate the mean time. We denote this time as T_{GEN}.

To establish whether the solution is deployable on IIoT devices, we need to establish a baseline against which to compare. Obtaining attributes and generating keys is a process that is carried out as a step prior to the exchange of information. In this sense, conceptually, the objective is similar to that of the TLS/DTLS handshake. Therefore, we rely on the work by [200], in which the authors measure the time required to improve the efficiency of the DTLS handshake. Their proposal achieves an average time of 250ms, which they consider adequate for IIoT devices, which we take as the baseline for our experiment. Thus, the solution proposed in this chapter is feasible if $T_{AS} + T_{GEN} < 250ms$.

Figure 6.6 presents the experimental evaluation results, which clearly show that the limit of 250ms is not exceeded in any case, which is the threshold set to consider the solution acceptable. One aspect to highlight in the figure is the constant time required to obtain the data from the attribute spoofing prevention solution. It is observed that, regardless of the size of the file storing the attributes, the average time to obtain them from IPFS is 145ms. This result is related to how IPFS stores the information in 256kB blocks. If the stored file is smaller than 256kB, a single block is enough to store it.

If it is larger, the file is split into several 256kB blocks, and a last block is generated whose content is used to link the previous ones. Several tests have been performed with JSONS of different sizes, and the IPFS block limit is not exceeded in any cases. As a consequence, the amount of blocks to get from IPFS is always the same, which causes similar times for their download. On the other hand, before getting the IPFS data, the CID has to be retrieved from IOTA. However, the CID is created from the hash of the IOTA file and thus always has the same size. Consequently, the amount of transactions to recover from IOTA is always the same. The consequence of this is that, although $T_{AS} > T_{GEN}$, the one that can cause the limit set in the baseline to be exceeded is T_{GEN}.

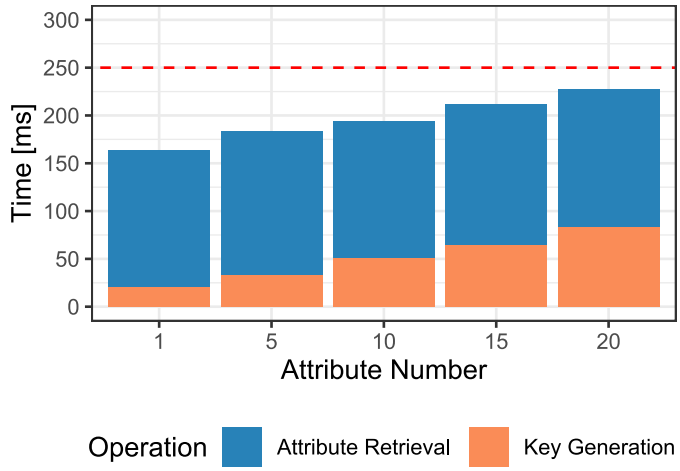


Fig. 6.6 Time in ms to generate SK_{ABE} with attribute validation.

As Chapter 4 shows and as Figure 6.6 reinforces, the time required for key generation in W11 increases linearly with the number of attributes contained in it. Therefore, knowing that the authority takes an average of 145ms to obtain the attributes and that it takes 20ms to generate a key with 1 attribute and 83ms to generate a key with 20 attributes, it can be calculated that the number of attributes that will exceed the limit will be 26. Since having 26 attributes is a far-fetched assumption for a real environment, it is implausible that the 250ms limit will be exceeded.

It can be concluded that the inclusion of the attribute spoofing prevention system satisfies the imposed constraint of $T_{AS} + T_{KG} < 250ms$. The delay added by the solution falls within the margin considered for it to be acceptable, especially given the added security; thus, **TR 4.4** is satisfied.

6.5 Summary

This chapter identifies the issue of attribute spoofing in CP-ABE. It outlines the basis of the attack and proposes an attribute spoofing prevention system that relies on IOTA, IPFS, and FIM. The combination of IOTA and IPFS ensures secure attribute storage, and using FIM for user authentication prevents impersonation. This way, the system distributes responsibility and trust among each value chain member and protects the system against single-point failures, identity theft, and attribute spoofing. This enhances E2E security and ensures the integrity and confidentiality of industrial data.

I may not have gone where I intended to go, but I think I have ended up where I needed to be.

Douglas Adams, The Long Dark Tea-Time of the Soul

Chapter 7

Concluding remarks

7.1 Conclusions

The beginning of this dissertation defined the overall objective of the thesis as *"To design an information exchange system for value chains that guarantees E2E data confidentiality and integrity"*. The goal is linked to the lack of a de-facto solution to accomplish this task, and the dissertation has provided a way to bridge this gap. In general terms, the thesis presents the design of a basic security architecture for the industry and establishes the information exchange solution using ABE. More specifically, these solutions are contextualized in the resolution of four research questions, the answers and analysis of which are provided below:

Question 1: *"Can we define and deploy a baseline industrial security architecture for the various partners in the value chain?"* – Answered in Chapter 3.

Yes, it is possible to establish a baseline architecture for industrial security following the standards of the leading institutions in each country.

State of the art has revealed that the main security strategy demanded by IEC 62443 is DiD. Moreover, ENISA and ICS-CERT agree on adopting this strategy. Therefore, this chapter unifies the criteria of the mentioned institutions and identifies the objectives to be met by a DiD strategy for industrial use and condenses them into 6: Availability, Confidentiality and Integrity, Logical and Physical access control, ICS vulnerability mitigation and system monitorization. Based on these objectives, the chapter clearly defines the layers that need to be deployed (i.e., physical security, perimeter, internal network, host, application, and data), and it also identifies the security measures in each of them.

However, establishing a security architecture for each plant in the value chain is not enough to guarantee the E2E confidentiality and integrity of the information exchanged between partners. Therefore, it also defines the 4 requirements to be met by an

information exchange solution for value chains:

- **R1.** The solution shall be based on robust and efficient cryptographic ciphers.
- **R2.** The solution shall provide E2E confidentiality and integrity for data management in industrial environments.
- **R3.** The solution shall be flexible to changes in data access rights.
- **R4.** The solution shall provide authentication and data access rights validation.

As a consequence of the answer to question 1, and based on state of the art, one-to-many encryption solutions such as ABE are known to be a potential solution for securing the information exchange system. As a consequence, the following questions arose.

Question 2: *"Which ABE ciphers are suitable for deployment in industrial production environments?"* – Answered in Chapter 4 - Associated Requirement **R1**.

The scheme chosen for industrial use is W11 [102], implemented by the OpenABE library.

State of the art has identified many ABE schemes. While it is true that CP-ABE is more developed than KP-ABE and dCP-ABE, there is still no de-facto scheme for any of the three. As Chapter 2 illustrates, there are several potential ABE schemes from which the most suitable for industrial use must be chosen. However, there is an added obstacle to using ABE in industrial projects: the lack of libraries. The dependencies used to deploy a security solution significantly influence overall efficiency and security, but no survey evaluates the different libraries' features or computational requirements. Therefore, Chapter 4 establishes a methodology based on **R1** to choose a suitable library for industrial applications. Fulfilling **R1** requires choosing a robust and efficient scheme from a theoretical point of view and considering its practical implementation. The fulfillment of this requirement, in turn, makes it possible to answer Question 2.

In order to achieve all this, the chapter carries out a qualitative and security evaluation of 11 different libraries. The qualitative evaluation considers which libraries are maintained and identifies which ones implement vulnerable ABE schemes and their mathematical dependencies. At the end of this analysis, four potential candidates for deployment in industrial environments are selected. These libraries are subjected to an experimental analysis in which we implement 20 combinations of libraries and schemes in two devices of different computational power (i.e., a RPI0 and a RPI4). The experimental analysis measures the time required by the devices to perform basic cryptographic operations. The results of these experiments are shown in Table 4.5, which allowed us to conclude that the best scheme-library combination to be deployed in value

chains is the W11 scheme of OpenABE. This library is written in C++, which makes it suitable for low-capacity IIoT devices. Its key generation, encryption, and decryption speeds have proven to be the fastest of all the combinations studied.

Question 3: *"Can we establish an ABE-based system that can be deployed in value chains, adapt it to organizational changes, and guarantee E2E data confidentiality and integrity?"* – Answered in Chapter 5 - Associated Requirements **R2** and **R3**.

Yes, it is possible to deploy an ABE-based system for value chains.

Due to the dynamic nature of value chains, the ABE-based information exchange system needs to be flexible to changes while guaranteeing the E2E confidentiality and integrity of the deployed system. To this effect, state of the art established that E2E security in cryptography requires that only the device that generates the data and the device that is the final recipient can access the information. Therefore, the information must be encrypted at the source and be decrypted only by the end device. In addition, the information exchange solution must be deployable in value chains. To ensure industrial applicability, two industrial security oriented ETSI documents have been identified: [7] and [182]. These documents establish industry requirements and propose using a CP-ABE scheme with CCA security. Policy updates are included among the requirements, which state of the art had already identified as an open issue. In this sense, none of the solutions studied in the state of the art is widely adopted, and none is oriented to industrial environments. Moreover, most of them do not consider CPA or CCA security, or they propose proprietary schemes whose security has not yet been extensively analyzed by third parties.

To meet the industrial requirements, the chapter proposes Multi-Layered CP-ABE, an information exchange system with policy updates based on the ETSI CCA scheme, which uses the W11 CP-ABE schema as a primitive. Roles within the value chain are also identified to deploy Multi-Layered CP-ABE while maintaining the E2E data security. Therefore, having met the ETSI industrial constraints, established the roles of the information exchange system, and with E2E security, the proposal is analyzed experimentally. For this purpose, six scenarios (3 for times and 3 for sizes) have been defined using real industrial data. The tests encrypt real industrial data, using an RPI4 and an RPI0 to model high-capacity industrial devices and IIoT devices. The results of the experiments demonstrate the practical feasibility of the proposal, where the industrial data is encrypted with AES-GCM and the AES key with Multi-Layered CP-ABE. In the case of sizes, for an original data of 160kBytes, an AES-based ciphertext of 213kBytes is generated. However, the encrypted key size never exceeds 70kBytes, thus occupying up to 67% less than the original encrypted message.

In terms of time, how long RPI0 and RPI4 take to perform the operations has been

calculated. The results show that the proposed architecture makes the solution deployable on IIoT devices. This is achieved by delegating part of the computational load to a higher-capability device. However, even after delegating 80% of the workload, E2E security is still maintained.

Question 4: *"How do we deploy an attribute-validation system in the context of Industry 4.0?"* – Answered in Chapter 6 - Associated Requirement **R4**.

The attribute validation system can be deployed by combining IPFS, IOTA, and FIM.

The security of CP-ABE-based data exchange is founded on users receiving keys that unambiguously reflect their attributes. It is also founded on the assumption that attackers cannot obtain a private key to access the information they should not have access to. These assumptions are unrealistic and expose the system to attribute spoofing. For this reason, Chapter 6 defines the vulnerability of attribute spoofing and defines two attack vectors that can exploit attribute spoofing and the consequences they can have for the system (e.g., privilege escalation, access reduction, or illegitimate obtaining of private keys). The attacks are framed in four assumptions that define how the system behaves.

The attribute validation system must have a distributed attribute storage, for which this chapter establishes the use of IPFS. This distributed storage protocol offers high-capacity operation and guarantees the integrity of the information contained in it. IPFS bases its operation on widely proven technologies, such as DHTs. In addition, it has no central server, and the nodes participating in the network do not need to trust each other. To ensure attribute auditability, IPFS is combined with IOTA, a DLT designed for IoT devices. IOTA gives the system the auditability that IPFS alone cannot guarantee. To frame the proposal in the context of a value chain, Chapter 6 also establishes how IPFS and IOTA are included in the information exchange system presented in Chapter 5. For this purpose, it is assumed that every system user belongs to a company and that this company is in charge of storing the attributes in the combined IPFS and IOTA solution. This way, the authorities that generate private keys only have to retrieve the information from these points. This solution is combined with FIM to guarantee the authentication of system users. Finally, after the qualitative analysis of the solution and evaluating how the deployed measures avoid the identified attack vectors and comply with the established technical requirements, the system's capacity to be deployed in IIoT devices is evaluated. For this purpose, the result of the article [200] is established as a baseline, in which the authors consider a key negotiation for IoT to be adequate when it only requires around 250ms. In our proposal, the time required to recover the attributes of a user and generate a key is measured, and it never exceeds 250ms, and

thus the solution is suitable for IIoT.

Therefore, the thesis's overall objective has been achieved thanks to the three Industry 4.0 security solutions presented in this dissertation: the definition of the DiD strategy for each plant in a value chain, the design and experimental implementation of an application-layer E2E secure information exchange system with access right updates, and the design and implementation of the attribute spoofing prevention system. Additionally, the dissertation also evaluates different ABE implementations to guarantee that the experiments to validate the proposals are performed using industrial-grade dependencies. Thus, the thesis achieves a privacy-driven E2E secure data exchange for value chains and proposes an adaptable and open solution for Industry 4.0.

7.2 List of Publications

International Journals

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "Securing IIoT using Defence-in-Depth: Towards an End-to-End secure Industry 4.0," Journal of Manufacturing Systems, vol. 57, pp. 367–378, 2020.

Conference Proceedings

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "Multi-Layered CP-ABE Scheme for Flexible Policy Update in Industry 4.0," in 2021 10th Mediterranean Conference on Embedded Computing (MECO), (Budva, Montenegro), pp. 1–4, June 2021.

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, ""They got my keys!": On the Issue of Key Disclosure and Data Protection in Value Chains," in 2nd IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0 (ARCI' 2022), (Andorra la Vella, Andorra), pp. 42–45, January 2022.

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, ""Are you what you claim to be?" Attribute Validation with IOTA for Multi Authority CP-ABE," in Blockchain 2022, (L'Aquila, Italy), July 2022. – Awarded the Best Paper Application in BLOCKCHAIN'22.

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of CP-ABE Libraries," in URSI 2022, (Málaga, Spain), pp. 1–4, Sept. 2022.

Preprints – Submitted to Journal

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, “End to End Secure Data Exchange in Value Chains with Dynamic Policy Updates,” 2022. *Submitted to Future Generation Computer Systems.* Preprint available on *Arxiv*.

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, “Too Many Options: A Survey of ABE Libraries for Developers,” 2022. *Submitted to Computer Networks.* Preprint available on *Arxiv*.

- A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, “Trust your users as far as you can validate them: Secure Attribute Retrieval for ABE Schemes," 2022. *Submitted to Smart Cities.*

7.3 Future Research Lines

There are several ways to extend the work presented in this thesis:

Chapters 5 and 6 use CP-ABE as the encryption algorithm to achieve one-to-many encryption. However, IBE and ABE algorithms have a known problem: key escrow. A compromised attribute authority could generate keys for itself and access all shared information in the system. A few years ago, researchers began to address this problem in IBE schemes with what they called Registration-Based Encryption (RBE) [201]. The authors established the existence of a “key curator" that managed users’ private keys but did not have the power to generate keys for itself. However, the proposal considered the key curator honest, lacked strong security proof, and was inefficient. Still, interest in this research path is increasing, with recent works proposing an efficient RBE that solves several of the problems of previous constructions [202] or combining it with Blockchain [203]. Proposals for IBE and ABE schemes tend to align, and by the end of 2022, registered ABE-based encryption was proposed [204]. Seeing the interest raised recently and the now existing ABE scheme, it is a research direction to consider, as this type of encryption could be applied to our data exchange system. These schemes could eliminate the central point failure posed by authorities and further reduce the attack surface for attribute spoofing.

Chapter 6 uses IOTA to achieve auditability in the attribute spoofing prevention system. However, IOTA is a DLT under continuous development. In fact, IOTA 2.0, the version used in this dissertation, was announced in June 2021. One of the features of IOTA 2.0 is the existence of Smart Contracts. However, as with many technologies under development, IOTA Smart Contracts are still at an early stage of development.

Nevertheless, it is a promising technology that could be incorporated into Chapter 6 proposal to relieve companies and the authority of some of the tasks they must perform.

Finally, Post Quantum computing is an emerging field for cryptography. In this sense, researchers have proposed various ABE-based schemes resistant to quantum cryptanalysis [205] [206], and it is a field shown to be promising for Big Data [207]. Furthermore, post-q cryptography is being used to solve some of the classical problems on ABE schemes, like reducing its computational cost [208] and enhancing the security of dCP-ABE [209]. Thus, it is a growing field whose features could benefit the data exchange system proposed in this thesis.

REFERENCES

[1] European Union Agency for Network and Information Security (ENISA), "Good Practices for Security of Internet of Things in the context of Smart Manufacturing," Tech. Rep. November, ENISA, 2018.

[2] G. Selander, J. P. Mattsson, F. Palombini, and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)," Tech. Rep. 8613, Internet Engineering Task Force, 2019.

[3] T. Heer, M. Heintel, S. Hiensch, L. Jänicke, M. Jochem, B. Kärcher, M. Kisch, J. Mehrfeld, G. Oeynhausen, T. Pfeiffer, F. Schewe, M. Schmitt, D. Schulz, D. Tenhagen, K. Theuerkauf, A. Teuscher, and T. Walloschke, "Secure Communication for Industrie 4.0," tech. rep., Plattform Industrie 4.0, Berlin, 2016.

[4] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn, "Guide to Industrial Control Systems (ICS) Security," Tech. Rep. Revision 2, US Department of Commerce, 2015.

[5] International Electrotechnical Commission, "IEC 62443-3-3: System Security Requirements and Security Levels," tech. rep., IEC, Geneva, Switzerland, 2013.

[6] I. Mugarza, J. L. Flores, and J. L. Montero, "Security Issues and Software Updates Management in the Industrial Internet of Things (IIoT) Era," *Sensors*, vol. 20, no. 24, p. 7160, 2020.

[7] ETSI - CYBER, "Application of Attribute Based Encryption (ABE) for PII and personal data protection on IoT devices, WLAN, cloud and mobile services - High level requirements," tech. rep., ETSI, 2018.

[8] A. Rojko, "Industry 4.0 Concept: Background and Overview," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, no. 5, pp. 77–90, 2017.

[9] European Union Agency for Network and Information Security (ENISA), "Industry 4.0 Cybersecurity: Challenges and Recommendations," tech. rep., ENISA, 2019.

[10] IBM, "Cost of a Data Breach Report 2020 | IBM," tech. rep., Ponemon Institute for IBM, 2020.

[11] ITRC, "Q1 2021 Data Breach Analysis," tech. rep., Identity Theft Resource Center (ITRC), 2021.

[12] IBM, "Cost of a Data Breach Report 2022 | IBM," tech. rep., Ponemon Institute for IBM, 2022.

[13] International Electrotechnical Commission, "IEC 62443-4-2: Technical Security Requirements for IACS Components," tech. rep., IEC, Geneva, Switzerland, 2019.

[14] Industrial Control Systems Cyber Emergency Response Team, "Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-In-Depth Strategies," Tech. Rep. September, US Department of Homeland Security, 2016.

[15] Bundesministerium für Bildung und Forschung, "Industrie 4.0: Innovationen für die Produktion von morgen," tech. rep., BMBF - Bundesministerium für Bildung und Forschung, Berlin, 2017.

[16] J. Pennekamp, M. Henze, S. Schmidt, P. Niemietz, M. Fey, D. Trauth, T. Bergs, C. Brecher, and K. Wehrle, "Dataflow Challenges in an Internet of Production: A Security and Privacy Perspective," in *Proceedings of the ACM Workshop on Cyber-Physical Systems Security and Privacy*, CPS-SPC'19, (New York, NY, USA), pp. 27–38, Association for Computing Machinery, 2019.

[17] N. Tuptuk and S. Hailes, "Security of Smart Manufacturing Systems," *Journal of Manufacturing Systems*, vol. 47, no. February, pp. 93–106, 2018.

[18] F. Alshohoumi, M. Sarrab, A. AlHamadani, and D. Al-Abri, "Systematic review of existing IoT architectures security and privacy issues and concerns," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 7, 2019.

[19] C. Boyd, A. Mathuria, and D. Stebila, *Group Key Establishment*, pp. 389–440. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020.

[20] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Advances in Cryptology – EUROCRYPT 2005* (R. Cramer, ed.), (Berlin, Heidelberg), pp. 457–473, Springer Berlin Heidelberg, 2005.

[21] J. Pennekamp, L. Bader, R. Matzutt, P. Niemietz, D. Trauth, M. Henze, T. Bergs, and K. Wehrle, "Private Multi-Hop Accountability for Supply Chains," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, (Dublin, Ireland), pp. 1–7, IEEE - Institute of Electrical and Electronics Engineers Inc., June 2020.

[22] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "Securing IIoT using Defence-in-Depth: Towards an End-to-End secure Industry 4.0," *Journal of Manufacturing Systems*, vol. 57, pp. 367–378, 2020.

[23] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "All Cryptolibraries Are Beautiful, But Some Are More Beautiful Than Others: A Survey of CP-ABE Libraries," in *URSI 2022*, (Málaga, Spain), pp. 1–4, Sept. 2022.

[24] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "Too Many Options: A Survey of ABE Libraries for Developers," 2022. Preprint.

[25] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "Multi-Layered CP-ABE Scheme for Flexible Policy Update in Industry 4.0," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, (Budva, Montenegro), pp. 1–4, June 2021.

[26] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, ""They got my keys!": On the Issue of Key Disclosure and Data Protection in Value Chains," in *2nd IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0 (ARCI' 2022)*, (Andorra la Vella, Andorra), pp. 42–45, Jan. 2022.

[27] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, "End to End Secure Data Exchange in Value Chains with Dynamic Policy Updates," 2022. Preprint.

[28] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga, and A. Urbieta, ""Are you what you claim to be?" Attribute Validation with IOTA for Multi Authority CP-ABE," in *Blockchain 2022*, (L'Aquila, Italy), pp. 279–288, July 2022.

[29] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[30] R. Setola, L. Faramondi, E. Salzano, and V. Cozzani, "An overview of Cyber Attack to Industrial Control System," *Chemical Engineering Transactions*, vol. 77, no. July, pp. 907–912, 2019.

[31] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1606–1616, 2019.

[32] A. Mikhail, I. A. Kamil, and H. Mahajan, "Increasing SCADA System Availability by Fault Tolerance Techniques," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, (Pune, India), pp. 1–5, IEEE - Institute of Electrical and Electronics Engineers Inc., Aug. 2017.

[33] F. Maggi, D. Quarta, M. Pogliani, M. Polino, A. M. Zanchettin, and S. Zanero, "Rogue Robots: Testing the Limits of an Industrial Robot's Security," 2017.

[34] M. El-Hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A Survey of Internet of Things (IoT) Authentication Schemes," *Sensors (Switzerland)*, vol. 19, no. 5, p. 1141, 2019.

[35] H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.

[36] L. Haghnegahdar, S. S. Joshi, and N. B. Dahotre, "From IoT-based cloud manufacturing approach to intelligent additive manufacturing: Industrial Internet of Things—An overview," *The International Journal of Advanced Manufacturing Technology*, pp. 1–18, 2022.

[37] T. J. Williams, "The Purdue Enterprise Reference Architecture," *Computers in Industry*, vol. 24, no. 2, pp. 141–158, 1994.

[38] A. Bécue, I. Praça, and J. Gama, "Artificial Intelligence, Cyber-Threats and Industry 4.0: Challenges and Opportunities," *Artificial Intelligence Review*, vol. 54, pp. 3849–3886, June 2021.

[39] Platform Industrie 4.0, "Reference Architectural Model Industrie 4.0 (RAMI4.0) - An Introduction," tech. rep., Plattform Industrie 4.0, 2018.

[40] R. M., P. M., S. M., and H. N., "A new architecture model for smart manufacturing: A performance analysis and comparison with the RAMI 4.0 reference model," *Advances in Production Engineering and Management*, vol. 14, no. 2, pp. 153–165, 2019.

[41] M. Herrero Collantes and A. López Padilla, "Protocolos y Seguridad de Red en SCI," tech. rep., INCIBE, 2015.

[42] International Electrotechnical Commission, "IEC 62443-1-1: Terminology, Concepts and Models.," tech. rep., IEC, Geneva, Switzerland, 2009.

[43] I. Mugarza, J. L. Flores, and J. L. Montero, "Security Issues and Software Updates Management in the Industrial Internet of Things (IIoT) Era," *Sensors*, vol. 20, no. 24, 2020.

[44] M. E. Porter, *Competitive advantage of nations: creating and sustaining superior performance*. simon and schuster, 2011.

[45] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

[46] L. Urciuoli and J. Hintsa, "Adapting supply chain management strategies to security – an analysis of existing gaps and recommendations for improvement," *International Journal of Logistics Research and Applications*, vol. 20, no. 3, pp. 276–295, 2017.

[47] B. B. Flynn, B. Huo, and X. Zhao, "The impact of supply chain integration on performance: A contingency and configuration approach," *Journal of Operations Management*, vol. 28, no. 1, pp. 58–71, 2010.

[48] L. Bader, J. Pennekamp, R. Matzutt, D. Hedderich, M. Kowalski, V. Lücken, and K. Wehrle, "Blockchain-based privacy preservation for supply chains supporting lightweight multi-hop information accountability," *Information Processing and Management*, vol. 58, no. 3, p. 102529, 2021.

[49] G. A. Vazquez-Martinez, J. Gonzalez-Compean, V. J. Sosa-Sosa, M. Morales-Sandoval, and J. C. Perez, "CloudChain: A novel distribution model for digital products based on supply chain principles," *International Journal of Information Management*, vol. 39, pp. 90–103, 2018.

[50] J. Pennekamp, E. Buchholz, Y. Lockner, M. Dahlmanns, T. Xi, M. Fey, C. Brecher, C. Hopmann, and K. Wehrle, "Privacy-Preserving Production Pro-

cess Parameter Exchange," in *Annual Computer Security Applications Conference*, ACSAC '20, (Austin, USA), pp. 510–525, Association for Computing Machinery, Dec. 2020.

[51] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[52] M. O. Rabin, "How to exchange secrets with oblivious transfer," *Cryptology ePrint Archive*, 2005.

[53] J. Pennekamp, F. Alder, R. Matzutt, J. T. Mühlberg, F. Piessens, and K. Wehrle, "Secure End-to-End Sensing in Supply Chains," in *2020 IEEE Conference on Communications and Network Security (CNS)*, (Avignon, France), pp. 1–6, IEEE - Institute of Electrical and Electronics Engineers Inc., June 2020.

[54] M. Dahlmanns, J. Pennekamp, I. B. Fink, B. Schoolmann, K. Wehrle, and M. Henze, "Transparent End-to-End Security for Publish/Subscribe Communication in Cyber-Physical Systems," in *Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems*, SAT-CPS '21, (Virtual Event, USA), pp. 78–87, Association for Computing Machinery, Apr. 2021.

[55] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 157–169, 2018.

[56] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying Fog Computing in Industrial Internet of Things and Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 4674–4682, Oct. 2018.

[57] D. Wu, X. Liu, S. Hebert, W. Gentzsch, and J. Terpenny, "Democratizing digital design and manufacturing using high performance cloud computing: Performance evaluation and benchmarking," *Journal of Manufacturing Systems*, vol. 43, pp. 316–326, 2017.

[58] R. Liu and A. Kumar, "Leveraging information sharing to configure supply chains," *Information Systems Frontiers*, vol. 13, no. 1, pp. 139–151, 2011.

[59] T. Alves, T. Morris, and S.-M. Yoo, "Securing SCADA Applications Using OpenPLC With End-To-End Encryption," in *Proceedings of the 3rd Annual Industrial Control System Security Workshop*, ICSS 2017, (San Juan, PR, USA), pp. 1–6, Association for Computing Machinery, Dec. 2017.

[60] R. J. Robles, M. Balitanas, R. Caytiles, Y. Gelogo, and T. Kim, "Comparison of Encryption Schemes as Used in Communication between SCADA Components," in *2011 International Conference on Ubiquitous Computing and Multimedia Applications*, (Daejeon, South Korea), pp. 115–118, IEEE - Institute of Electrical and Electronics Engineers Inc., Apr. 2011.

[61] R. J. Robles and M.-K. Choi, "Symmetric-Key Encryption for Wireless Internet SCADA," in *International Conference on Security Technology* (D. Ślęzak, T.-h. Kim, W.-C. Fang, and K. P. Arnett, eds.), (Jeju Island, Korea (Republic of)), pp. 289–297, Springer Berlin Heidelberg, Dec. 2009.

[62] R. J. Robles, M. Balitanas, and T.-h. Kim, "Security Encryption Schemes for Internet SCADA: Comparison of the Solutions," in *Security-Enriched Urban Computing and Smart Grid* (R.-S. Chang, T.-h. Kim, and S.-L. Peng, eds.), (Hualien, Taiwan), pp. 19–27, Springer Berlin Heidelberg, Sept. 2011.

[63] L. Pietre-Cambacédès and P. Sitbon, "Cryptographic Key Management for SCADA Systems-Issues and Perspectives," in *2008 International Conference on Information Security and Assurance (isa 2008)*, (Busan, South Korea), pp. 156–161, IEEE - Institute of Electrical and Electronics Engineers Inc., Apr. 2008.

[64] J. Granjal, E. Monteiro, and J. S. Silva, "End-to-End Transport-Layer Security for Internet-Integrated Sensing Applications with Mutual and Delegated ECC Public-Key Authentication," in *2013 IFIP Networking Conference*, pp. 1–9, May 2013.

[65] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3." RFC 8446, Aug. 2018.

[66] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security Version 1.2." RFC 6347, Jan. 2012.

[67] A. Esfahani, G. Mantas, J. Ribeiro, J. Bastos, S. Mumtaz, M. A. Violas, A. M. De Oliveira Duarte, and J. Rodriguez, "An Efficient Web Authentication Mechanism Preventing Man-In-The-Middle Attacks in Industry 4.0 Supply Chain," *IEEE Access*, vol. 7, pp. 58981–58989, 2019.

[68] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS Based Security and Two-Way Authentication for the Internet of Things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2710–2723, 2013.

[69] M. Gunnarsson, J. Brorsson, F. Palombini, L. Seitz, and M. Tiloca, "Evaluating the performance of the OSCORE security protocol in constrained IoT environments," *Internet of Things*, vol. 13, p. 100333, 2021.

[70] J. Astorga, M. Barcelo, A. Urbieta, and E. Jacob, "Revisiting the Feasibility of Public Key Cryptography in Light of IIoT Communications," *Sensors*, vol. 22, no. 7, 2022.

[71] M. Moghaddam, M. N. Cadavid, C. R. Kenley, and A. V. Deshmukh, "Reference Architectures for Smart Manufacturing: A Critical Review," *Journal of Manufacturing Systems*, vol. 49, pp. 215–225, 2018.

[72] O. Garcia-Morchon, S. Kumar, and M. Sethi, "Internet of Things (IoT) Security: State of the Art and Challenges." RFC 8576, 2019.

[73] P. M. Kumar and U. D. Gandhi, "Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application," *The Journal of Supercomputing*, vol. 76, pp. 3963–3983, 2020.

[74] G. Nebbione and M. C. Calzarossa, "Security of IoT Application Layer Protocols: Challenges and Findings," *Future Internet*, vol. 12, no. 3, 2020.

[75] J. P. Mattsson, G. Selander, and G. A. Eriksson, "Object Security in Web of Things," in *W3C Workshop on the Web of Things: Enablers and services for an open Web of Devices*, (Berlin, Germany), pp. 1–5, June 2014.

[76] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, and R. Guizzetti, "OSCAR: Object security architecture for the Internet of Things," *Ad Hoc Networks*, vol. 32, pp. 3–16, 2015.

[77] C. Bormann and P. E. Hoffman, "Concise Binary Object Representation (CBOR)," Tech. Rep. 7049, Internet Engineering Task Force, 2015.

[78] J. Schaad, "CBOR Object Signing and Encryption (COSE)," Tech. Rep. 8152, Internet Engineering Task Force, 2017.

[79] G. Selander, J. P. Mattsson, and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)," Internet-Draft draft-ietf-lake-edhoc-17, Internet Engineering Task Force, Oct. 2022. Work in Progress.

[80] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[81] C. Gündoğan, C. Amsüss, T. C. Schmidt, and M. Wählisch, "IoT Content Object Security with OSCORE and NDN: A First Experimental Comparison," in *2020 IFIP Networking Conference (Networking)*, pp. 19–27, 2020.

[82] A. Fiat and M. Naor, "Broadcast Encryption," in *Advances in Cryptology — CRYPTO' 93* (D. R. Stinson, ed.), (Berlin, Heidelberg), pp. 480–491, Springer Berlin Heidelberg, 1994.

[83] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," in *Advances in Cryptology – CRYPTO 2005* (V. Shoup, ed.), (Berlin, Heidelberg), pp. 258–275, Springer Berlin Heidelberg, 2005.

[84] C. Delerablée, "Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys," in *Advances in Cryptology – ASIACRYPT 2007* (K. Kurosawa, ed.), (Berlin, Heidelberg), pp. 200–215, Springer Berlin Heidelberg, 2007.

[85] D. Xie, F. Chen, Y. Luo, and L. Li, "One-to-many image encryption with privacy-preserving homomorphic outsourced decryption based on compressed sensing," *Digital Signal Processing*, vol. 95, p. 102587, 2019.

[86] G. Epiphaniou, P. Pillai, M. Bottarelli, H. Al-Khateeb, M. Hammoudesh, and C. Maple, "Electronic Regulation of Data Sharing and Processing Using Smart Ledger Technologies for Supply-Chain Security," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1059–1073, 2020.

[87] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Advances in Cryptology* (G. R. Blakley and D. Chaum, eds.), (Berlin, Heidelberg), pp. 47–53, Springer Berlin Heidelberg, 1985.

[88] P. K. P., S. K. P., and A. P.J.A., "Attribute based encryption in cloud computing: A survey, gap analysis, and future directions," *Journal of Network and Computer Applications*, vol. 108, pp. 37–52, 2018.

[89] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, (New York, NY, USA), pp. 89–98, Association for Computing Machinery, 2006.

[90] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, (Berkeley,

USA), pp. 321–334, IEEE - Institute of Electrical and Electronics Engineers Inc., May 2007.

[91] S. Müller, S. Katzenbeisser, and C. Eckert, "Distributed Attribute-Based Encryption," in *Information Security and Cryptology – ICISC 2008* (P. J. Lee and J. H. Cheon, eds.), (Berlin, Heidelberg), pp. 20–36, Springer Berlin Heidelberg, 2009.

[92] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-Based Encryption with Non-Monotonic Access Structures," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, (New York, NY, USA), pp. 195–203, Association for Computing Machinery, 2007.

[93] A. Lewko, A. Sahai, and B. Waters, "Revocation Systems with Very Small Private Keys," in *2010 IEEE Symposium on Security and Privacy*, pp. 273–285, May 2010.

[94] N. Attrapadung, B. Libert, and E. de Panafieu, "Expressive Key-Policy Attribute-Based Encryption with Constant-Size Ciphertexts," in *Public Key Cryptography – PKC 2011* (D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, eds.), (Berlin, Heidelberg), pp. 90–108, Springer Berlin Heidelberg, 2011.

[95] X. Yao, Z. Chen, and Y. Tian, "A lightweight attribute-based encryption scheme for the Internet of Things," *Future Generation Computer Systems*, vol. 49, pp. 104–112, 2015.

[96] S.-Y. Tan, K.-W. Yeow, and S. O. Hwang, "Enhancement of a Lightweight Attribute-Based Encryption Scheme for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, pp. 6384–6395, Aug. 2019.

[97] A. W. Dent, "Adapting the Weaknesses of the Random Oracle Model to the Generic Group Model," in *Advances in Cryptology — ASIACRYPT 2002* (Y. Zheng, ed.), (Berlin, Heidelberg), pp. 100–109, Springer Berlin Heidelberg, 2002.

[98] L. Cheung and C. Newport, "Provably Secure Ciphertext Policy ABE," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, (New York, NY, USA), pp. 456–465, Association for Computing Machinery, 2007.

[99] Y. Lu, Y. Wang, X. Dai, J. Li, J. Li, and M. Chen, "Survey of Attribute-Based Encryption in Cloud Environment," in *Cognitive Cities* (J. Shen, Y.-C. Chang,

Y.-S. Su, and H. Ogata, eds.), (Singapore), pp. 375–384, Springer Singapore, 2020.

[100] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded Ciphertext Policy Attribute Based Encryption," in *Automata, Languages and Programming* (L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, eds.), (Berlin, Heidelberg), pp. 579–591, Springer Berlin Heidelberg, 2008.

[101] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, (New York, NY, USA), pp. 343–352, Association for Computing Machinery, 2009.

[102] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," in *14th Int. Conf. Pract. Theory Public Key Cryptogr.* (D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, eds.), (Taormina, Italy), pp. 53–70, Springer Berlin Heidelberg, Mar. 2011.

[103] A. Beimel *et al.*, "Secure schemes for secret sharing and key distribution," 1996.

[104] S. Yamada, N. Attrapadung, G. Hanaoka, and N. Kunihiro, "A Framework and Compact Constructions for Non-monotonic Attribute-Based Encryption," in *Public-Key Cryptography – PKC 2014"* (H. Krawczyk, ed.), (Berlin, Heidelberg), pp. 275–292, Springer Berlin Heidelberg, 2014.

[105] Y. Rouselakis and B. Waters, "Practical Constructions and New Proof Methods for Large Universe Attribute-Based Encryption," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, CCS '13, (New York, NY, USA), pp. 463–474, Association for Computing Machinery, 2013.

[106] J. Chen, R. Gay, and H. Wee, "Improved Dual System ABE in Prime-Order Groups via Predicate Encodings," in *Advances in Cryptology - EUROCRYPT 2015* (E. Oswald and M. Fischlin, eds.), (Berlin, Heidelberg), pp. 595–624, Springer Berlin Heidelberg, 2015.

[107] A. Lewko, "Tools for simulating features of composite order bilinear groups in the prime order setting," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 318–335, Springer, 2012.

[108] T. Okamoto and K. Takashima, "Fully secure functional encryption with general relations from the decisional linear assumption," in *Annual cryptology conference*, pp. 191–208, Springer, 2010.

[109] S. Agrawal and M. Chase, "FAME: Fast Attribute-Based Message Encryption," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pp. 665–682, Association for Computing Machinery, 2017.

[110] A. de la Piedra, M. Venema, and G. Alpár, "ABE Squared: Accurately Benchmarking Efficiency of Attribute-Based Encryption." Cryptology ePrint Arch., Report 2022/038, 2022.

[111] S. Hohenberger and B. Waters, "Online/Offline Attribute-Based Encryption," in *Public-Key Cryptography – PKC 2014* (H. Krawczyk, ed.), (Berlin, Heidelberg), pp. 293–310, Springer Berlin Heidelberg, Mar. 2014.

[112] S. Qi, Y. Zheng, M. Li, Y. Liu, and J. Qiu, "Scalable Industry Data Access Control in RFID-Enabled Supply Chain," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3551–3564, 2016.

[113] S. Fugkeaw and H. Sato, "Updating Policies in CP-ABE-Based Access Control: An Optimized and Secure Service," in *Service-Oriented and Cloud Computing* (M. Aiello, E. B. Johnsen, S. Dustdar, and I. Georgievski, eds.), (Vienna, Austria), pp. 3–17, Springer International Publishing, Aug. 2016.

[114] Y. Yasumura, H. Imabayashi, and H. Yamana, "Attribute-based proxy re-encryption method for revocation in cloud storage: Reduction of communication cost at re-encryption," in *2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA)*, (Shanghai, China), pp. 312–318, IEEE - Institute of Electrical and Electronics Engineers Inc., Mar. 2018.

[115] J. Li, S. Wang, Y. Li, H. Wang, H. Wang, H. Wang, J. Chen, and Z. You, "An Efficient Attribute-Based Encryption Scheme with Policy Update and File Update in Cloud Computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6500–6509, 2019.

[116] M. Bouchaala, C. Ghazel, and L. A. Saidane, "Revocable Sliced CipherText Policy Attribute Based Encryption Scheme in Cloud Computing," in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, (Tangier, Morocco), pp. 1860–1865, IEEE - Institute of Electrical and Electronics Engineers Inc., June 2019.

[117] C. Wang and Y. Yuan, "An Efficient Ciphertext-Policy Attribute-Based Encryption Scheme with Policy Update," *Computers, Materials and Continua*, vol. 63, no. 2, pp. 1031–1041, 2020.

[118] A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," in *Advances in Cryptology – EUROCRYPT 2011* (K. G. Paterson, ed.), pp. 568–588, Springer Berlin Heidelberg, May 2011.

[119] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A Robust and Verifiable Threshold Multi-Authority Access Control System in Public Cloud Storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484–1496, 2016.

[120] Y. Rouselakis and B. Waters, "Efficient Statically-Secure Large-Universe Multi-Authority Attribute-Based Encryption," in *Financ. Cryptogr. Data Secur.* (R. Böhme and T. Okamoto, eds.), (San Juan, Puerto Rico), pp. 315–332, Springer Berlin Heidelberg, 2015.

[121] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, "Fully Secure Multi-authority Ciphertext-Policy Attribute-Based Encryption without Random Oracles," in *Computer Security – ESORICS 2011* (V. Atluri and C. Diaz, eds.), (Berlin, Heidelberg), pp. 278–297, Springer Berlin Heidelberg, 2011.

[122] H. Zhong, W. Zhu, Y. Xu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," *Soft Computing*, vol. 22, no. 1, pp. 243–251, 2018.

[123] K. Yang and X. Jia, *DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems*, pp. 59–83. New York, NY: Springer New York, 2014.

[124] M. Venema and G. Alpár, "A Bunch of Broken Schemes: A Simple yet Powerful Linear Approach to Analyzing Security of Attribute-Based Encryption," in *CT-RSA 2021*, (Virtual Event), pp. 100–125, Springer, May 2021.

[125] K. Yang and X. Jia, "Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 1735–1744, July 2014.

[126] M. Xiao, M. Wang, X. Liu, and J. Sun, "Efficient distributed access control for big data in clouds," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 202–207, 2015.

[127] E. D. Knapp and J. T. Langill, *Industrial Network Security, Second Edition: Securing Critical Infrastructure Networks for Smart Grid, SCADA, and Other Industrial Control Systems*. Syngress, 2 ed., 2014.

[128] M. Brachmann, S. L. Keoh, O. G. Morchon, and S. S. Kumar, "End-to-end Transport Security in the IP-Based Internet of Things," in *2012 21st International Conference on Computer Communications and Networks, ICCCN 2012 - Proceedings*, (Munich, Germany), pp. 1–5, IEEE - Institute of Electrical and Electronics Engineers Inc., July 2012.

[129] O. Givehchi, K. Landsdorf, P. Simoens, and A. W. Colombo, "Interoperability for Industrial Cyber-Physical Systems: An Approach for Legacy Systems," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3370–3378, 2017.

[130] International Electrotechnical Commission, "IEC 62443-4-1: Secure Product Development Lifecycle Requirements," tech. rep., IEC, Geneva, Switzerland, 2018.

[131] T. Nguyen, *Microsoft Azure IaaS Defense in Depth Guide*. T. Nguyen, 2017.

[132] W. Granzer and A. Treytl, "Security in Industrial Communication Systems," in *The industrial electronics handbook - Industrial communication systems* (B. M. Wilamowski and J. D. Irwin, eds.), ch. 22, pp. 318–332, CRC Press, second ed., 2011.

[133] P. Koopman, K. Driscoll, and B. Hall, "Selection of Cyclic Redundancy Code and Checksum Algorithms to Ensure Critical Data Integrity," Tech. Rep. March, U.S. Department of Transportation - Federal Aviation Administration, 2015.

[134] B. Maxwell, D. R. Thompson, G. Amerson, and L. Johnson, "Analysis of CRC Methods and Potential Data Integrity Exploits," in *International Conference on Emerging Technologies*, (Minneapolis, Minnesota, US), pp. 25–26, Aug. 2003.

[135] Z. DeSmit, A. E. Elhabashy, L. J. Wells, and J. A. Camelio, "An Approach to Cyber-Physical Vulnerability Assessment for Intelligent Manufacturing Systems," *Journal of Manufacturing Systems*, vol. 43, pp. 339–351, 2017.

[136] A. A. Creery and E. J. Byres, "Industrial Cybersecurity for Power System and SCADA networks," in *Record of Conference Papers Industry Applications Society 52nd Annual Petroleum and Chemical Industry Conference*, (Denver, CO, USA, USA), pp. 303–309, IEEE - Institute of Electrical and Electronics Engineers Inc., Sept. 2005.

[137] B. Reaves and T. Morris, "Analysis and Mitigation of Vulnerabilities in Short-Range Wireless Communications for Industrial Control Systems," *International Journal of Critical Infrastructure Protection*, vol. 5, no. 3, pp. 154–174, 2012.

[138] T. Roosta, D. K. Nilsson, U. Lindqvist, and A. Valdes, "An Intrusion Detection System for Wireless Process Control Systems," in *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, (Atlanta, GA, USA), pp. 866–872, IEEE - Institute of Electrical and Electronics Engineers Inc., Oct. 2008.

[139] T. Mavroeidakos, A. Michalas, and D. D. Vergados, "Security Architecture Based on Defense in Depth for Cloud Computing Environment," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, (San Francisco, USA), pp. 334–339, IEEE - Institute of Electrical and Electronics Engineers Inc., Apr. 2016.

[140] D. Kuipers and M. Fabro, "Control Systems Cyber Security:Defense in Depth Strategies," tech. rep., Idaho National Laboratory (INL), Idaho Falls, US, 2006.

[141] X. Zhou, Z. Xu, L. Wang, K. Chen, C. Chen, and W. Zhang, "Construction and evaluation of defense-in-depth architecture in SCADA system," *MATEC Web of Conferences*, vol. 173, pp. 1–5, 2018.

[142] ISA, "ISA-62443-3-2: Security for Industrial Automation and Control Systems - Security Risk Assessment and System Design," tech. rep., ISA, 2015.

[143] Smart Card Alliance, "Using smart cards for secure physical access," Tech. Rep. July, Smart Card Alliance, 2003.

[144] S. Liu and M. Silverman, "A Practical Guide to Biometric Security Technology," *IT Professional*, vol. 3, no. 1, pp. 27–32, 2001.

[145] E. Byres, A. Ginter, and J. Langill, "How Stuxnet Spreads – A Study of Infection Paths in Best Practice Systems," Tech. Rep. Version 1.0, Tofino Security, Abterra Technologies, ScadaHacker, 2011.

[146] S. Thomason, "Improving Network Security: Next Generation Firewalls and Advanced Packet Inspection Devices," *Global Journal of Computer Science and Technology*, vol. 12, no. 13, pp. 47–50, 2012.

[147] N. Jiang, H. Lin, Z. Yin, and C. Xi, "Research of Paired Industrial Firewalls in Defense-in-Depth Architecture of Integrated Manufacturing or Production System," in *2017 IEEE International Conference on Information and Automation*

(ICIA), (Macau, China), pp. 523–526, IEEE - Institute of Electrical and Electronics Engineers Inc., July 2017.

[148] H. K. Patil, D. Wing, and T. M. Chen, "Chapter 60 - VoIP Security," in *Computer and Information Security Handbook (Third Edition)* (J. R. Vacca, ed.), ch. 60, pp. 859–873, Boston: Morgan Kaufmann, 3 ed., 2013.

[149] D. Wu, A. Ren, W. Zhang, F. Fan, P. Liu, X. Fu, and J. Terpenny, "Cybersecurity for digital manufacturing," *Journal of Manufacturing Systems*, vol. 48, pp. 3–12, 2018.

[150] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014.

[151] M. B. Line, A. Zand, G. Stringhini, and R. Kemmerer, "Targeted Attacks against Industrial Control Systems: Is the Power Industry Prepared?," in *Proceedings of the 2nd Workshop on Smart Energy Grid Security*, SEGS '14, (Scottsdale, USA), pp. 13–22, Association for Computing Machinery, Nov. 2014.

[152] S. Abe, M. Fujimoto, S. Horata, Y. Uchida, and T. Mitsunaga, "Security Threats of Internet-Reachable ICS," in *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, (Tsukuba, Japan), pp. 750–755, IEEE, Sept. 2016.

[153] F. López de la Mora and S. Nadi, "Which Library Should I Use?: A Metric-Based Comparison of Software Libraries," in *2018 IEEE/ACM 40th International Conference on Software Engineering: New Ideas and Emerging Technologies Results (ICSE-NIER)*, (Gothenburg, Sweden), pp. 37–40, Aug. 2018.

[154] J. Wohlwender, R. Huesmann, A. Heinemann, and A. Wiesmaier, "Cryptolib: Comparing and Selecting Cryptography Libraries," in *Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference*, EICC '22, (New York, NY, USA), pp. 6–11, Association for Computing Machinery, 2022.

[155] 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)*, (San Jose, CA, USA), pp. 154–171, May 2017.

[156] N. Patnaik, J. Hallett, and A. Rashid, "Usability Smells: An Analysis of Developers' Struggle With Crypto Libraries," in *Fifteenth Symposium on Usable*

Privacy and Security (SOUPS 2019), (Santa Clara, CA), pp. 245–257, USENIX Association, Aug. 2019.

[157] O. Hyncica, P. Kucera, P. Honzik, and P. Fiedler, "Performance Evaluation of Symmetric Cryptography in Embedded Systems," in *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, vol. 1, (Prague, Czech Republic), pp. 277–282, Sept. 2011.

[158] U. Kumar, T. Borgohain, and S. Sanyal, "Comparative Analysis of Cryptography Library in IoT," 2015.

[159] S. Zickau, D. Thatmann, A. Butyrtschik, I. Denisow, and A. Küpper, "Applied Attribute-based Encryption Schemes," in *19th International ICIN Conference-Innovations in Clouds, Internet and Networks*, Mar. 2016.

[160] J. Bethencourt, A. Sahai, and B. Waters, "cpabe-toolkit," 2011.

[161] M. D. Green, J. A. Akinyele, and M. A. Rushanan, "libfenc," 2011.

[162] JHU Security and Crypto Lab, "Charm," 2022.

[163] B. Waters, M. Green, S. H. Waters, J. A. Akinyele, A. M. Dunn, and M. Rushanan, "OpenABE," 2011.

[164] G. Bramm., M. Gall., and J. Schütte., "BDABE - Blockchain-based Distributed Attribute based Encryption," in *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - SECRYPT,*, pp. 99–110, INSTICC, SciTePress, 2018.

[165] J. Schütte, G. Bramm, and D. Bücheler, "Rabe," 2022.

[166] M. Bizjak, J. Hartman, T. Marc, and M. Stopar, "GoFE," 2021.

[167] M. Bizjak, J. Hartman, T. Marc, and M. Stopar, "CiFEr," 2021.

[168] MIRACL, "Mircl," 2019.

[169] D. de Freitas Aranha, C. Porto Lopes Gouvêa, T. Markmann, R. S. Wahby, and K. Liao, "Relic," 2022.

[170] Denis, Frank, "libsodium," 2022.

[171] OpenSSL Software Foundation, "OpenSSL," 2022.

[172] FENTEC Project, "bn256," 2019.

[173] De Caro, Angelo, "jPBC," 2015.

[174] Lynn, Ben, "PBC," 2022.

[175] Bramm, Georg and Bowe, Sean, "Rabe-bn," 2022.

[176] C. Chow, "JPBC-FAME," 2020.

[177] J. Wang, J. Perrochet, M. Grossi, and S. Weiland, "cp-abe," 2018.

[178] M. Morales-Sandoval and A. Diaz-Perez, "DET-ABE: A Java API for Data Confidentiality and Fine-Grained Access Control from Attribute Based Encryption," in *WISTP*, pp. 104–119, Springer, Aug. 2015.

[179] TU-Berlin-SNET, "JCPABE," 2016.

[180] Q. Liu, G. Wang, and J. Wu, "Time-based Proxy Re-encryption Scheme for Secure Data Sharing in a Cloud Environment," *Information Sciences*, vol. 258, pp. 355–370, 2014.

[181] U. P. D. Ani, H. M. He, and A. Tiwari, "Review of cybersecurity issues in industrial critical infrastructure: manufacturing in perspective," *J. Cyber Secur. Technol.*, vol. 1, no. 1, pp. 32–74, 2017.

[182] ETSI - CYBER, "Attribute Based Encryption for Attribute Based Access Control - ETSI TS 103 532 V1.2.1," tech. rep., ETSI, 2021.

[183] E. Fujisaki and T. Okamoto, "How to Enhance the Security of Public-Key Encryption at Minimum Cost," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, no. 1, pp. 24–32, 2000.

[184] A. D. Keromytis and J. M. Smith, "Requirements for Scalable Access Control and Security Management Architectures," *ACM Trans. Internet Technol.*, vol. 7, p. 8–es, may 2007.

[185] J. Hu, J. Deng, N. Gao, and J. Qian, "Application Architecture of Product Information Traceability Based on Blockchain Technology and a Lightweight Secure Collaborative Computing Scheme," in *2020 International Conference on E-Commerce and Internet Technology (ECIT)*, pp. 335–340, 2020.

[186] D. Stefanescu, P. Galán-García, L. Montalvillo, J. Unzilla, and A. Urbieta, "Towards a Holistic DLT Architecture for IIoT: Improved DAG for Production Lines," in *Blockchain and Applications* (J. Prieto, A. Partida, P. Leitão, and

A. Pinto, eds.), (Salamanca, Spain), pp. 179–188, Springer International Publishing, 2022.

[187] R. Nakanishi, Y. Zhang, M. Sasabe, and S. Kasahara, "Combining IOTA and Attribute-Based Encryption for Access Control in the Internet of Things," *Sensors*, vol. 21, no. 5053, pp. 1–15, 2021.

[188] D. Preuveneers, W. Joosen, J. Bernal Bernabe, and A. F. Skarmeta, "Distributed Security Framework for Reliable Threat Intelligence Sharing," *Secur. Commun. Networks*, vol. 2020, p. 8833765, 2020.

[189] D. Di Francesco Maesa, A. Lunardelli, P. Mori, and L. Ricci, "Exploiting Blockchain Technology for Attribute Management in Access Control Systems," in *Econ. Grids, Clouds, Syst. Serv.* (K. Djemame, J. Altmann, J. Á. Bañares, O. Agmon Ben-Yehuda, and M. Naldi, eds.), (Leeds, UK), pp. 3–14, Springer International Publishing, Sept. 2019.

[190] D. Di Francesco Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable Access Control systems," *Comput. Secur.*, vol. 84, pp. 93–119, 2019.

[191] D. Thatmann, A. Butyrtschik, and A. Küpper, "A Secure DHT-Based Key Distribution System for Attribute-Based Encryption and Decryption," in *2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–9, 2015.

[192] T. M. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, and P. Fraga-Lamas, "Towards an Autonomous Industry 4.0 Warehouse: A UAV and Blockchain-Based System for Inventory and Traceability Applications in Big Data-Driven Supply Chain Management," *Sensors*, vol. 19, no. 10, 2019.

[193] M. Zichichi, S. Ferretti, and G. D'Angelo, "A Distributed Ledger Based Infrastructure for Smart Transportation System and Social Good," in *2020 IEEE 17th Annual Consumer Communications and Networking Conference (CCNC)*, pp. 1–6, 2020.

[194] M. Shahjalal, M. M. Islam, M. M. Alam, and Y. M. Jang, "Implementation of a Secure LoRaWAN System for Industrial Internet of Things Integrated With IPFS and Blockchain," *IEEE Systems Journal*, pp. 1–10, 2022.

[195] T. M. Fernández-Caramés and P. Fraga-Lamas, "A Review on the Application of Blockchain to the Next Generation of Cybersecure Industry 4.0 Smart Factories," *IEEE Access*, vol. 7, pp. 45201–45218, 2019.

[196] N. Sealey, A. Aijaz, and B. Holden, "IOTA Tangle 2.0: Toward a Scalable, Decentralized, Smart, and Autonomous IoT Ecosystem," *arXiv preprint arXiv:2209.04959*, 2022.

[197] L. Cui, S. Yang, Z. Chen, Y. Pan, M. Xu, and K. Xu, "An Efficient and Compacted DAG-Based Blockchain Protocol for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 4134–4145, June 2020.

[198] J. Rosenberger, F. Rauterberg, and D. Schramm, "Performance study on IOTA Chrysalis and Coordicide in the Industrial Internet of Things," in *2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, pp. 88–93, 2021.

[199] D. Hardt, "The OAuth 2.0 Authorization Framework." RFC 6749, Oct. 2012.

[200] A. Atutxa, J. Astorga, M. Barcelo, A. Urbieta, and E. Jacob, "Improving efficiency and security of IIoT communications using in-network validation of server certificate," *Computers in Industry*, p. 103802, 2022.

[201] S. Garg, M. Hajiabadi, M. Mahmoody, and A. Rahimi, "Registration-based encryption: removing private-key generator from IBE," in *Theory of Cryptography Conference*, pp. 689–718, Springer, 2018.

[202] N. Glaeser, D. Kolonelos, G. Malavolta, and A. Rahimi, "Efficient registration-based encryption," *Cryptology ePrint Archive*, 2022.

[203] Q. Wang, R. Li, D. Galindo, Q. Wang, S. Chen, and Y. Xiang, "Transparent registration-based encryption through blockchain," *Distributed Ledger Technologies: Research and Practice*, 2022.

[204] S. Hohenberger, G. Lu, B. Waters, and D. J. Wu, "Registered attribute-based encryption." Cryptology ePrint Archive, Paper 2022/1500, 2022. `https://eprint.iacr.org/2022/1500`.

[205] X. Fu, Y. Ding, H. Li, J. Ning, T. Wu, and F. Li, "A survey of lattice based expressive attribute based encryption," *Computer Science Review*, vol. 43, p. 100438, 2022.

[206] E.-S. Zhuang, C.-I. Fan, and I.-H. Kuo, "Multiauthority attribute-based encryption with dynamic membership from lattices," *IEEE Access*, vol. 10, pp. 58254–58267, 2022.

[207] Z. B. Jemihin, S. F. Tan, and G.-C. Chung, "Attribute-Based Encryption in Securing Big Data from Post-Quantum Perspective: A Survey," *Cryptography*, vol. 6, no. 3, 2022.

[208] X. Fu, Y. Wang, L. You, J. Ning, Z. Hu, and F. Li, "Offline/online lattice-based ciphertext policy attribute-based encryption," *Journal of Systems Architecture*, vol. 130, p. 102684, 2022.

[209] B. Waters, H. Wee, and D. J. Wu, "Multi-authority abe from lattices without random oracles." Cryptology ePrint Archive, Paper 2022/1194, 2022. `https://eprint.iacr.org/2022/1194`.

[210] R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," in *Advances in Cryptology - EUROCRYPT 2004* (C. Cachin and J. L. Camenisch, eds.), (Berlin, Heidelberg), pp. 207–222, Springer Berlin Heidelberg, 2004.

Appendices

Appendix A

Cryptographic Terminology

Cryptographic terminology can be complex, including terms like cipher, algorithm, or primitives. Although they have different meanings, sometimes, this difference is minimal and can be confusing. This appendix clarifies some of the cryptographic terminology used in the thesis, especially that used during the Chapter 2 and Chapter 4.

- **Private key:** It is a term inherited from asymmetric cryptography. This thesis uses it to refer to the decryption keys generated by ABE schemes. The reader should note, however, that this key is also called secret key in the literature. Since the latter term comes from symmetric cryptography and can lead to confusion, the concept of private key has been chosen.

- **Master public key:** In ABE-based cryptography, it refers to a parameter needed to encrypt, decrypt and generate private keys. Although this thesis has decided to use this denomination, in the literature, different authors have referred to this variable as public parameters, master key, or public key.

- **Family (of cryptographic algorithms):** It is used in this thesis to refer to collections of cryptographic algorithms that share unique ways of characterizing encryption and decryption—for example, IBE or ABE.

- **Mode (of operation):** A term inherited from symmetric cryptography refers to AES's modes of operation, e.g., AES-CBC, AES-GCM. For lack of a better term, this thesis recycles the concept and uses it to refer to KP-ABE, dCP-ABE, and CP-ABE.

- **Scheme:** The term is used throughout this thesis to refer to an algorithm that details what precise mathematical operations must be performed to encrypt and decrypt. In the literature, this is sometimes also called cipher—for example,

FAME [109], DAC-MACS [123], LSW10 [93] or W11 [102]. Literature also refers to these as ciphers and cryptographyic algorithms.

Thus, if we were to talk about W11, it is an asymmetric scheme belonging to the ABE family of cryptographic algorithms whose mode is CP-ABE. Figure A.1 shows an example of how the above terminology is used to classify algorithms scenes. The figure is not intended to show a thorough classification of the algorithms, since a detailed classification of the ABE schemes is already given in Chapter 4.

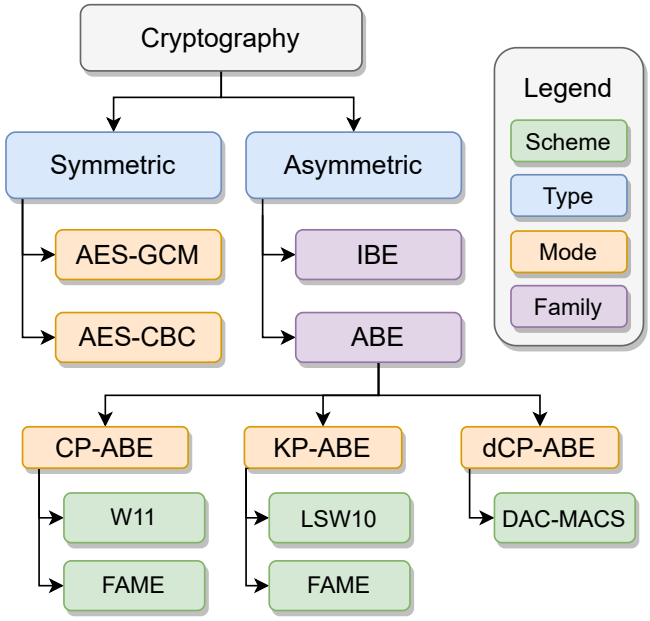


Fig. A.1 Cryptography Classification.

Appendix B

CPA and CCA Security in Cryptography

The functionality of encryption algorithms is based on entropy, which hinders relating the original message to the ciphertext without knowing the decryption key and the encryption algorithm. However, achieving such entropy requires random functions, which is impossible in deterministic machines like computers. Instead, pseudo-random approximations and security models are used, which make certain assumptions to measure the security of algorithms.

Cryptographers use security models to build an algorithm's security. These models are based on assumptions related to the randomness of the algorithms. If the assumption is that a truly random function exists, it is called the random oracle model. This model represents an idealized scenario that does not represent reality since, as stated above, there is no truly random function. By contrast, the standard model assumes that there is no true randomness. In this model, security must depend exclusively on the algorithm's mathematical robustness. The algorithms developed in the standard model are less efficient than those developed on the random oracle, but they represent the real world.

However, these models are intended for designing cryptographic algorithms. Meanwhile, when deploying cryptographic schemes in an environment, it is more useful to know whether they are resistant to certain attacks or not. The security model cannot provide this information by itself. Instead, the property of resisting attacks is modeled from security games, which lead to CPA security and CCA security.

B.1 CPA Security

CPA security is the minimum security level a scheme must meet to be deployed in production. CPA security guarantees security against passive attackers. For example, most ABE schemes, their KEM versions, the one-time-pad, or AES-CBC have CPA security. The game to prove CPA security involves an attacker, a challenger, and an

encryption oracle.

1. The attacker chooses two plaintexts, PT_0 and PT_1, and sends them to the challenger.

2. The challenger flips a coin and randomly chooses a value for n between 0 and 1. They send the resulting PT_n to the encryption oracle.

3. The encryption oracle is a black box for the attacker. The attacker does not know what algorithm is inside; they can only know its output. The oracle returns CT_n to the attacker.

4. If the attacker can know whether $n = 1$ or $n = 0$, the attacker can infer the encryption key used and win the game.

An algorithm is considered to have CPA security when the attacker can only win half the time, no more and no less. Otherwise, it means that the attacker has information about the type of algorithm that is or is not in the oracle. Winning half the time is called having a negligible advantage.

B.2 CCA Security

Cryptographic algorithms aspire to have CCA security, which guarantees protection against active attackers. Authenticated encryption like AES-GCM is CCA-secure, although there are also operations that can achieve CCA security from a CPA-secure algorithm [183]. Similarly, an IBE scheme with CCA security can also be obtained from one with CPA security [210]. In the case of CP-ABE, for example, the authors of [102] consider that their CP-ABE algorithm could achieve CCA security as dictated by [210].

The game that proves that an encryption algorithm has CCA security requires an attacker, a challenger, an encryption oracle, and a decryption oracle. In contrast to CPA security, the attacker has access to the encryption and decryption oracles.

1. The attacker chooses two plaintexts, PT_0 and PT_1, and sends them to the challenger.

2. The challenger flips a coin and chooses an n between 0 and 1 at random. He sends the resulting PT_n to the encryption oracle.

3. The encryption oracle returns a challenge ciphertext to the attacker called CT_{CH}.

4. The attacker can still interact with the decryption oracle and send any ciphertext except CT_{CH}. Thus, it chooses CT_0 and CT_1.

5. The challenger flips a coin and randomly chooses n between 0 and 1. He sends the resulting CT_n to the decryption oracle.

6. The decryption oracle returns a PT_n to the challenger.

7. If using all the collected information, the attacker can know whether n equals 1 or 0, it means that the attacker can infer the decryption key used and wins the game.

As in the case of CPA security, there is CCA security if the attacker wins with a negligible advantage. CCA security protects against active attackers because it simulates a case where an attacker tricks the decryption algorithm and gets it to decrypt ciphertexts chosen by the attacker. These ciphertexts can result from intercepting a legitimate ciphertext and modifying it, violating its integrity. The attacker is looking for in the CCA game to determine if the cryptographic algorithm is malleable. That is if a known modification in a ciphertext generates a predictable change in the recovered plaintext.