A multi-layered CP-ABE scheme for flexible policy update in Industry 4.0

1st Aintzane Mosteiro-Sanchez, 2nd Marc Barcelo, 4th Aitor Urbieta
Ikerlan Technology Research Centre (BRTA)
Arrasate/Mondragón, Spain
{amosteiro, mbarcelo, aurbieta}@ikerlan.es

3rd Jasone Astorga
University of the Basque Country
Bilbao, Spain
jasone.astorga@ehu.eus

***Abstract*—Industry 4.0 connectivity requires ensuring end-to-end (E2E) security for industrial data exchange. This requirement is critical when users external to the OT network retrieve data. CP-ABE guarantees E2E security by encrypting data according to a policy. To use this encryption scheme in dynamic environments, such as manufacturing, the policy must be updatable. This paper proposes a Multi-Layered Policy Key Encapsulation Method for CP-ABE that allows flexible policy update and revocation without modifying the original CP-ABE scheme.**

***Index Terms*—Industry 4.0, IIoT, E2E Security, hybrid encryption, CP-ABE, policy update**

I. Introduction

Adoption of IT in Industry 4.0 introduces security risks in the manufacturing environment, as it connects the OT network to the outside world. A common practice to protect this network is to separate it from the IT network using a Demilitarised Zone (DMZ), restricting data exchange to this channel. Meanwhile, sensitive information is secured by protecting the communication channel using security protocols such as TLS or DTLS.

However, TLS and DTLS work at the transport layer, so they do not protect the messages over multiple connections. This compromises confidentiality in decentralised systems. Security solutions must therefore provide strict End-to-End (E2E) security and ensure that the intermediaries in the communication path cannot read such messages. In addition, as the DMZ is considered an insecure area, data also has to be protected during storage. Ciphertext-Policy Attribute-Based Encryption (CP-ABE), first proposed by Bethencourt *et al.* [1] and later improved by Waters [2], is a promising approach for Industry 4.0 scenarios.

Figure 1 illustrates how CP-ABE encrypts data under a policy defined with attributes, giving Data Owners (DOs) control over who can retrieve the plaintext. Users are given secret keys (SKs) related to the attributes they own, and can only decrypt data if they match the policy. CP-ABE also allows DOs to encrypt information once and share it with various users, in contrast to PKI (Public Key Infrastructure) solutions. Furthermore, CP-ABE requires no key exchange, saving bandwidth and preventing key theft. This, combined with the fact that middleboxes cannot get a SK, results in E2E security.

However, CP-ABE schemes are computationally heavy. This limits the implementation of CP-ABE in manufacturing scenarios, which favour lightweight schemes. Therefore, we use the CP-ABE Key Encapsulation Method (CP-AB-KEM) [3]. CP-AB-KEM uses CP-ABE to produce symmetric keys associated with attributes, which can then be applied in symmetric ciphers such as AES-GCM. This combination, which will be explained in Section III, makes cryptographic operations lighter and avoids key negotiation.

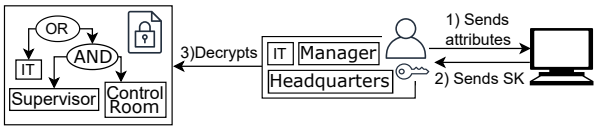

Fig. 1. CP-ABE. The user's attributes fulfil the policy and decrypts data.

II. CP-ABE in Industry

Encryption has been considered a threat to industrial availability [4], but Industry 4.0 can no longer neglect it. As introduced in Section I, this paper outlines the scenario in which users retrieve information generated in the OT network. To ensure E2E security, encryption is performed by the DOs, i.e., the field devices. They send the ciphertexts to be stored in the DMZ, delete the original plaintext, and no further interaction with them should be necessary.

To apply CP-ABE in Industry 4.0 schemes, we must define security goals compliant with industrial requirements and create a solution that accounts for industrial limitations. The mentioned combination of CP-AB-KEM and AES-GCM makes CP-ABE lighter, which favours its industrial use. The ETSI has defined some industrial ABE requirements in [5]. Among these, we have chosen those that apply to our scenario. Table I shows a summary with an indication of whether CP-AB-KEM with AES-GCM fulfils them on its own.

As Table I presents, the unmet specifications are related to policies and attributes. Although both attribute and policy management have a direct effect on the encryption algorithm, policy update and revocation are crucial for the sustainability of the stored industrial data. Due to the lifespan of manufacturing systems, policies related to encrypted data will vary. For instance, companies create new departments or disclose former confidential information. We believe that CP-AB-KEM can advance the inclusion of encryption in industrial environments, but these drawbacks must also be considered when designing

the security system. To this end, Section III explains how the combination of CP-ABE and AES-GCM work, and Section IV analyses our proposal for policy update.

TABLE I
COMPLIANCE WITH HIGH-LEVEL REQUIREMENTS DEFINED IN [5].

Required	Fulfilled by CP-AB-KEM + AES-GCM
Attribute update and expiration.	No
Policy update and expiration.	No
Addition of new policies to ciphertexts.	No
Non-identity based access control policies.	Yes
Time-based access control.	Yes[a]
Position-based access control.	Yes[b]
Role-based access control.	Yes[b]
Attribute based access control.	Yes
Emergency access control.	Yes[b]
Attribute management.	No
Integrity protection.	Yes

[a] Possible but very complex. [b] Added as an attribute.

III. CP-AB-KEM Symmetric Key Generation

CP-AB-KEM, by itself, only has Chosen Plaintext Attack (CPA) security [6]. It implies that attackers cannot recover the encryption key despite having encrypted a known plaintext and comparing the result with others produced with the same key. However, this type of security only protects against passive attackers. To be protected against active attackers, we need Chosen Ciphertext Attack (CCA) Security [7]. CCA security is the standard security notion for CP-AB-KEM and CP-ABE schemes. For KEM it means that attackers cannot get the keys even if they capture their encapsulation. And for CP-ABE, it means that even if attackers choose a ciphertext and compare it with its plaintext, they cannot discover the encryption key. In this context, the ETSI presents a CCA Secure CP-ABE [6] defined by four modules. The following subsections explain more about the encryption and decryption schemes.

A. Encryption

The ETSI requires a CPA-Secure CP-AB-KEM scheme like [2] or [8]. Over it they build three modules with security properties. If desired, the modules can be replaced by others with similar capabilities. Figure 2 shows the module construction. AP stands for Access Policy, M for message, and MPK for Master Public Key. The CPA secure CP-AB-KEM scheme is used unmodified as a black box and not defined.

1) CCA Secure CP-ABE. It requires a CCA-Secure CP-AB-KEM scheme to generate a symmetric key (K_{AE}) and its encapsulation (C_{ABE}). K_{AE} is used in ciphers like AES-GCM. It returns the encrypted M in C_{AE} and the encrypted key in C_{ABE}.
2) CCA Secure CP-AB-KEM. This module uses a CPA-Secure CP-ABE scheme to encrypt a randomly generated K_{AE} and outputting it in C_{ABE}. For CCA security, it uses a modified version of the Fujisaki-Okamoto transformation [7] that includes the AP in the Hash that generates R. It returns both K_{AE} and C_{ABE}.
3) CPA Secure CP-ABE. It calls a CPA Secure CP-AB-KEM scheme to generate a key (K_{ABE}) and its encapsulation (C_K) according to an AP and a random number R. K_{ABE} is used to generate a r_{num} of the same size as M. M is encrypted using a one-time pad and resulting in C_D. The algorithm returns both C_K and C_D.
4) A CPA Secure CP-AB-KEM. Any CPA secure scheme capable of generating a K_{ABE} and C_K using a random number R, an AP and a MPK can be used here.

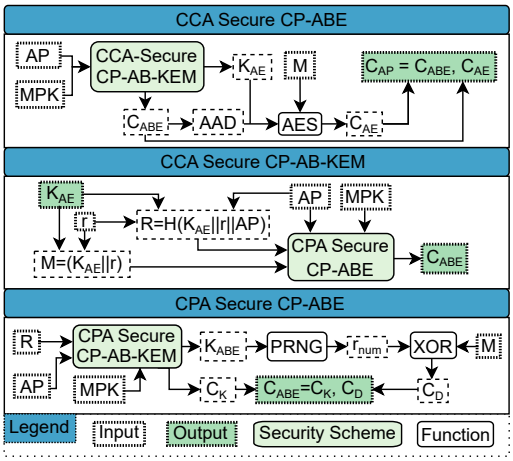

Fig. 2. ETSI's CCA Secure CP-ABE encryption modules.

B. Decryption

Along with encryption, the ETSI also defines the decryption modules presented in Figure 3. These modules are equivalent to those of Section III-A and have identical properties.

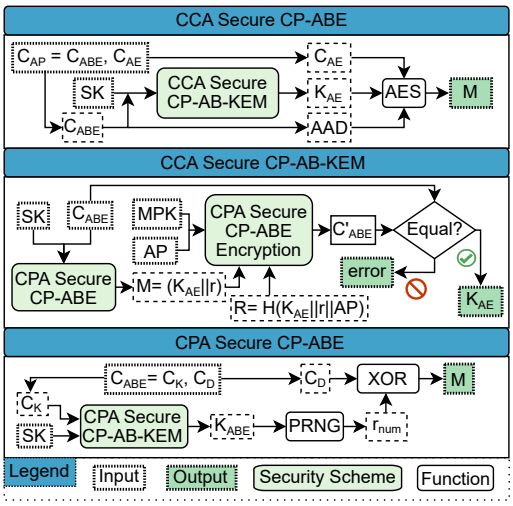

Fig. 3. ETSI's CCA Secure CP-ABE decryption with AES-GCM.

The decryption sequence is straightforward, the only difference being the CCA Secure CP-AB-KEM decapsulation. If this module is implemented as it is, it requires re-encrypting

the recovered $M = (K_{AE}||r)$ and comparing the resulting C'_{ABE} with the original C_{ABE}. If they are not equal, the recovered K_{AE} is incorrect, and the algorithm returns an error.

IV. Multi-Layered Policy KEM

CP-AB-KEM generates keys according to a policy, so every time the policy is redefined, the key changes. Thus, once keys are created, policy updates imply a reencryption, which is a computationally heavy operation. Sometimes it even implies adapting the original ABE encryption algorithm or requesting the DO to generate a delegation key, so a semitrusted third party can reencrypt the ciphertext. Therefore, this method does not fulfil the goal of ensuring E2E security, while also having a flexible policy update and revocation mechanism with a reduced number of interactions with DOs.

With the above goal in mind, in this section, we propose a multi-layered policy update and revocation scheme for CP-AB-KEM. This scheme structures the policies in multiple layers, where the outer layers comprise the most dynamic policies and the inner layers contain the more stable ones. With this approach, whenever a policy changes, rather than reencrypting the whole ciphertext, we only have to change a specific layer. Thus, this scheme only updates the symmetric key encapsulation, not the key itself. Hence, a full ciphertext reencryption is no longer necessary, and DOs do not need to create a delegation key or update the encryption policy themselves.

A. Encryption

In our proposed Multi-Layered KEM for CP-ABE, AP is still used to encrypt K_{AE} and r. Therefore, the policy update does not affect AES-GCM and we only have to modify the CCA Secure CP-AB-KEM encapsulation scheme. However, AP is now constituted by different layers, so the security scheme has to perform an iterative CPA Secure CP-ABE encryption. And since the nonce R is connected to the AP, it needs to be generated anew for every iteration. The resulting CCA Secure CP-AB-KEM encapsulation security scheme is shown in Figure 4. The innermost encryption layer is constructed following the ETSI security scheme. The next encryption layers are created by generating a new R according to the new AP and by encrypting the previous C'_{ABEi} using the CPA Secure CP-ABE Scheme. During encryption, the higher the value of i, the more external the encryption layer is. When all the AP layers are completed, the scheme outputs C_{ABE}, containing all the encryption layers.

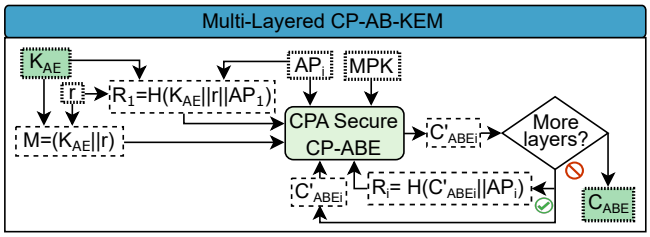

Fig. 4. Multi-Layered CP-AB-KEM encapsulation.

B. Decryption

Figure 5 shows the multi-layered decryption scheme. As with encryption, changes occur in the CCA Secure CP-AB-KEM decapsulation scheme, which uses the security schemes below as a black box. Because several layers compose the final key encapsulation, the decryption of the outermost ones returns another ciphertext. Finally, the innermost layer contains $M' = (K_{AE}||r)$. Only this last layer is checked, comparing the last C'_{APi} with the result of reencrypting M'. Like in the ETSI scheme, if both ciphertexts are the same, the scheme outputs K_{AE}. If not, it returns an error and users do not get the original plaintext.

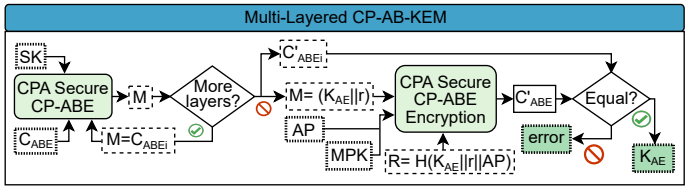

Fig. 5. Multi-Layered CP-AB-KEM Decapsulation.

V. Results

The iterative encryption of the Multi-Layered CP-AB-KEM increases the final CP-ABE ciphertext size. Since ciphertexts are stored, generating excessively large ciphertexts is a problem. Limiting the growth is essential to achieve a feasible solution for IIoT environments.

To accomplish E2E security in an industrial setting, DOs apply a first encryption layer, defined with a policy that is not expected to change. They send the result to the DMZ, where it is reencrypted using the new encapsulation method introduced in Section IV-A. To speed up update and revocation, policies more prone to variation must be placed in the outermost layer. Thus, the final ciphertext has two parts. The first one generated by the DO, and the second one composed of the encryption layers computed by the DMZ. Since the DO-generated ciphertext follows the ETSI scheme, this section focuses on the analysis of our proposal.

The used library is OpenABE [9]. It implements the ETSI CCA Secure CP-ABE and uses NIST-approved industry-standard cryptographic functions. We have run the tests in an Ubuntu Virtual Machine with an allocated RAM of 4GB, a plaintext of 18 kBytes and using AES-256-GCM.

Figure 6 compares the ciphertext size of Multi-Layered CP-ABE with an Iterative application of the ETSI scheme. As it can be observed, applying the policy layers to AES key encryption and using AES-GCM only once prevents exponential ciphertext growth. Compared with Iterative CP-ABE, with four layers, the ciphertext size reduction is almost 52% and with five the reduction is 61%.

To further study the relationship between layers and bytes added to the ciphertext, we analyse the ciphertext size evolution for a fixed number of layers (Figure 7). It shows a case where a total of 16 attributes are divided into four layers.

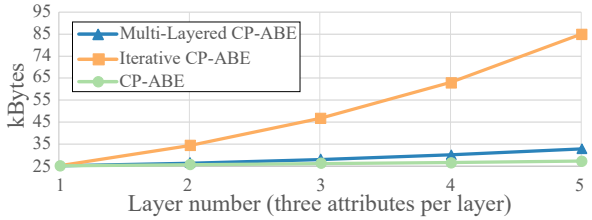

Fig. 6. CT size evolution. In CP-ABE, one layer contains all the attributes.

This combination yields a 31.8 kBytes ciphertext, which is approximately 18% bigger than the one generated by the ETSI scheme (27 kBytes). In contrast, in the case of having 15 attributes distributed over five layers, the generated ciphertext is approximately 33 kBytes (Figure 6) which is 22% larger than that of the ETSI scheme. In other words, if we would like to use a six-attribute policy, having two layers with three attributes is more efficient than having three layers with two.

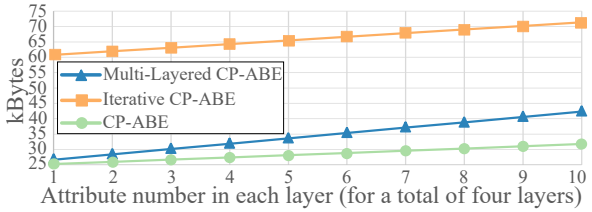

Fig. 7. CT size for a constant number of four layer.

Finally, as revealed by Figure 8, since the added operations to Multi-Layered CP-ABE only affect the symmetric key encryption, encryption times are quite similar to the ETSI scheme. In the case of decryption, Multi-Layered CP-ABE is faster than the ETSI scheme when the number of layers grows. This is because our proposal only validates the innermost layer, which is less complex than the ETSI case, making it faster. Finally, the time improvement of Multi-Layered CP-ABE over the iterative CP-ABE is clear, with decryption being almost 61% faster for the case of three layers.

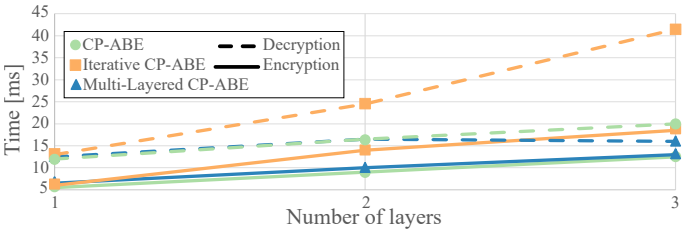

Fig. 8. Encryption and decryption time.

VI. Conclusion

The goal of this paper is to create a CP-ABE encryption scheme suitable for manufacturing environments. For this purpose, we have used ETSI's CCA Secure CP-ABE scheme, which fulfils many of the industrial requirements identified. Furthermore, we present a modification to that scheme that allows policies to be defined in layers, achieving easy update and revocation. The proposed Multi-Layered CP-ABE gives the possibility of revoking or updating policies with no need for a full ciphertext reencryption or symmetric key regeneration. We consider different layer and attribute combinations and present a possible industrial scenario. In the potential industrial scenario, we also analyse the roles each entity should play in communication. As a result, by using our scheme middleboxes have no access to the plaintext, true E2E security is achieved, and ABE flexibility in response to changes is increased. Moreover, since middleboxes perform policy updates, interactions with data owners are avoided.

Acknowledgment

This work was financially supported by European commission through ECSEL-JU 2018 program under the COMP4DRONES project (grant agreement N° 826610), with national financing from France, Spain, Italy, Netherlands, Austria, Czech, Belgium and Latvia. It was also partially supported by the *Ayudas Cervera para Centros Tecnológicos* grant of the Spanish Centre for the Development of Industrial Technology (CDTI) under the project EGIDA (CER-20191012), and by the Basque Country Government under the ELKARTEK program, project TRUSTIND - Creating Trust in the Industrial Digital Transformation (KK-2020/00054).

References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symp. Secur. Priv. (SP '07)*. Berkeley, USA: IEEE, may 2007, pp. 321–334. [Online]. Available: https://ieeexplore.ieee.org/document/4223236

[2] 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. Springer Berlin Heidelberg, mar 2011, pp. 53–70. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-19379-8_4

[3] M. C. Gorantla, C. Boyd, and J. M. González Nieto, "Attribute-Based Authenticated Key Exchange," in *Inf. Secur. Priv.*, R. Steinfeld and P. Hawkes, Eds. Sydney, Australia: Springer Berlin Heidelberg, jul 2010, pp. 300–317. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-14081-5_19

[4] 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. [Online]. Available: https://doi.org/10.1080/23742917.2016.1252211

[5] 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," ETSI, Tech. Rep., 2018.

[6] ——, "Attribute Based Encryption for Attribute Based Access Control," ETSI, Tech. Rep., 2018.

[7] E. Fujisaki and T. Okamoto, "How to Enhance the Security of Public-Key Encryption at Minimum Cost," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E83-A, no. 1, pp. 24–32, 2000. [Online]. Available: https://search.ieice.org/bin/summary.php?id=e83-a_1_24&category=A&year=2000&lang=E&abst=

[8] S. Agrawal and M. Chase, "FAME: Fast Attribute-Based Message Encryption," in *Proc. 2017 ACM SIGSAC Conf. Comput. Commun. Secur.*, ser. CCS '17. Dallas, Texas, USA: Association for Computing Machinery, oct 2017, pp. 665–682. [Online]. Available: https://dl.acm.org/doi/10.1145/3133956.3134014

[9] Zeutro, "The OpenABE Design Document Version 1.0," pp. 0–29, 2018. [Online]. Available: https://github.com/zeutro/openabe/blob/master/docs/libopenabe-v1.0.0-design-doc.pdf