Fine-Grained Access Control in Industrial Internet of Things

Dominik Ziegler, Josef Sabongui, Gerald Palfinger

Fine-Grained Access Control in Industrial Internet of Things

Evaluating Outsourced Attribute-Based Encryption

Dominik Ziegler[1][0000−0002−5930−8216], Josef Sabongui[2], and Gerald Palfinger[3]

[1] Know Center GmbH, Inffeldgasse 13, 8010 Graz, Austria
dominik.ziegler@tugraz.at
http://www.know-center.tugraz.at
[2] Graz University of Technology, Institute for Applied Information Processing and Communications, Inffeldgasse 16a, 8010 Graz, Austria
josef.sabongui@student.tugraz.at
https://www.iaik.tugraz.at
[3] A-SIT Secure Information Technology Center Austria, Seidlgasse 22 / Top 9, 1030 Vienna, Austria
gerald.palfinger@a-sit.at
https://www.a-sit.at

Abstract. Putting Attribute-Based Encryption (ABE) to the test, we perform a thorough performance analysis of ABE with outsourced decryption. In order to do so, we implemented a purely Java and Kotlin based Ciphertext-Policy Attribute-Based Encryption (CP-ABE) system. We specifically focus on the requirements and conditions of the Industrial Internet of Things (IIoT), including attribute revocation and limited computing power. We evaluate our system on both resource-constrained devices and high-performance cloud instances. Furthermore, we compare the overhead of our implementation with classical asymmetric encryption algorithms like RSA and ECC.
To demonstrate compatibility with existing solutions, we evaluate our implementation in the Siemens MindSphere IIoT operating system. Our results show that ABE with outsourced decryption can indeed be used in practice in high-security environments, such as the IIoT.

Keywords: Fine-grained access control · IIoT · Performance analysis · ABE

1 Introduction

The integration of Internet of Things (IoT) [4] into classical enterprise systems promises fundamental improvements to existing workflows, efficiency gains or optimised decision making. This trend towards interconnected devices in the manufacturing industry is called the Industrial Internet of Things (IIoT) or Industry 4.0 [12]. Its goal is to integrate and connect manufacturing environments via global networks. In fully automated IIoT systems, arbitrary sensors, production facilities or heavy machinery are all communicating and controlling each

other independently. As a result, the IIoT will allow enterprises to quickly adapt to customer requirements and individualisation in production processes while maintaining resource and energy efficiency.

However, Sadeghi et al. [20] show that devices in enterprise environments can generate vast amounts of security critical or personal data. Furthermore, processing of such data might be restricted by laws, like the General Data Protection Regulation (GDPR). As a result, these challenges call for an efficient and flexible access-control mechanism, to protect sensitive data from unauthorised access.

A cryptographic solution for fine-grained access control is Attribute-Based Encryption (ABE), first introduced by Sahai and Waters [21]. It represents a generalisation of Identity-Based Encryption (IBE), a concept proposed by Shamir [22]. In contrast to conventional public-key cryptography schemes, ABE defines the recipient of a message as a set of attributes. It does so, by combining ciphertexts with distinctive attributes and access control policies. There exist two main flavours of ABE. Key-Policy Attribute-Based Encryption (KP-ABE) [10] and Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [6]. KP-ABE schemes encrypt messages under a set of attributes. Access control policies, embedded into a party's secret key, determine which ciphertexts a key can decrypt. In CP-ABE constructions the role of ciphertexts and keys are reversed. Access control structures are directly embedded into ciphertexts and attributes are associated with a party's secret key. Parties can only decrypt ciphertexts if their attributes match the embedded policy.

A central issue with classical ABE systems, however, is that they typically do not have strong security guarantees or are based on expensive bilinear pairings and are hence not very efficient. Indeed, Wang et al. [23] show that classical ABE systems should only be used "when the computing device has relatively high computing power and the applications demand low to medium security". In contrast, we typically find a variety of devices with usually limited resources, e.g. sensors with low computing power or memory, in the IIoT. Hence, performance is is a critical aspect of ABE in the IIoT. As a result, several approaches aim at improving ABE, in terms of efficiency. For example, approaches based on Elliptic Curve Cryptography (ECC) or Linear Secret Sharing Schemes (LSSSs) [1, 19], address performance issues of ABE. Another promising approach for resource-constrained environments is outsourcing heavy computations to cloud servers [11, 17, 18]. Clients only need to perform lightweight operations. Recent advances in Outsourced Attribute-Based Encryption (OABE), promise applicability in resource-constrained environments, such as the IIoT. Hence, in this paper, we study OABE from two perspectives.

– **How does ABE with outsourced decryption compare to established cryptographic primitives like RSA and ECC, in practice?** We implemented a purely Java and Kotlin based CP-ABE system with outsourced decryption. We specifically focus on industrial requirements such as attribute revocation, key escrow and key exposure. We perform a thorough evaluation of the execution time of operations and overhead on both, resource-constrained devices and powerful cloud servers. Although a fair comparison of these prim-

itives is not possible, our goal is to highlight the overall performance of ABE with outsourced decryption with established primitives.

– **How can ABE efficiently be deployed in IIoT-based systems with high security requirements?** To demonstrate compatibility with existing infrastructure, we successfully deployed our system in the Siemens MindSphere cloud. MindSphere is the cloud-based, open IoT operating system from Siemens for the Industrial Internet of Things. It focuses on data acquisition and access control in IIoT environments. Our evaluation shows that, while adding some computational overhead, ABE can provide fine-grained access control in IIoT, in practice.

Motivated by our findings, we first provide an introduction to ABE and relevant aspects. Next, we evaluate execution time and overhead of our implementation on a Raspberry Pi 3 Model B+ and high-performance cloud instances. We evaluate the performance of different security levels. Furthermore, we show how OABE can successfully be deployed in IIoT environments such as the Siemens MindSphere platform. Subsequently, we highlight relevant work. Finally, we discuss our findings and give an outlook.

2 Background: ABE with Outsourced Decryption

We evaluate how ABE can be used in professional environments in practice. The work of Lin et al. [19] addresses several typical challenges of IIoT systems and was therefore chosen to serve as the foundation of the implemented system, concerning the ABE framework.

In our system, a Client's (CL) private key is split three-ways. To fully decrypt ciphertexts, all key parts have to be used. Since the majority of decryption calculations are outsourced to a cloud server, no expensive bilinear pairings have to be done on the client's side. Furthermore, the server's key-parts have to be retrieved on each decryption request. This ensures that key revocation is enforced instantaneously, by simply invalidating and updating user key-parts. The split private keys also prevent key escrow, since no actor has knowledge of all three key-parts throughout the entire decryption process.

In this section, we first give an introduction to involved actors of the implemented ABE systems with outsourced decryption. Next, we discuss security definitions and required algorithms.

2.1 Actors

The architecture of this ABE system identifies multiple actors, as outlined in this section. Each actor plays a significant role in the environment of this ABE system.

– *Data Owner:* A Data Owner (DO) produces information, which is encrypted using a random secret and an access policy. This initial ciphertext is subsequently sent to the Re-Encryption Server (RS).

- *Re-Encryption Server:* The main responsibility of the Re-Encryption Server (RS) is to re-encrypt initial ciphertexts. The re-encryption process incorporates the attribute groups of a system into the ciphertext. The RS is furthermore involved in the generation of the system parameters.
- *Key Authority:* The Key Authority (KA) is in charge of key management in the system. It creates and updates client key-parts in conjunction with the RS, but remains the sole authority to grant or revoke attributes from key-parts mentioned above. Any modifications to the system parameters, such as additional attributes throughout the systems lifetime, are exclusively performed by the KA.
  Decryption Server: The Decryption Server (DS) is capable of partially decrypting ciphertexts, obtained by a client. For this operation, the DS requires the key-parts from KA and RS. This partial decryption performs all expensive bilinear pairing operations, while leaving a simple El-Gamal encrypted ciphertext for the CL to decrypt.
- *Client:* A Client (CL) is an endpoint, which wants to access encrypted data. Since a valid CL key-part is necessary for this operation, the client needs to participate in the update process of key-parts. If the key-parts associated with a CL fulfil the access policy of a ciphertext, it is possible to obtain the plaintext with the help from the DS.

2.2 Definitions

Lin et al. [19] based their protocols on bilinear pairings of Type I. However, Chatterjee et al. [8] claim that Type I pairings are expected to be slower compared to Type II or Type III pairings. Hence, we chose a Type III pairing, specifically the *Ate Pairing over Barreto-Naehrig curves*. We will be using the notion of additive groups instead of multiplicative groups for the two paired groups $\mathbb{G}_1$ and $\mathbb{G}_2$, for illustration purposes. The target group remains a multiplicative group.

Bilinear Maps. Given two additive cyclic groups $\mathbb{G}_1$, $\mathbb{G}_2$ with generators g_1, g_2 respectively and a multiplicative group $\mathbb{G}_T$. Let the pairing map be:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T \tag{1}$$

The prime order of $\mathbb{G}_1$ is $p \in \mathbb{P}$. The hash functions of this system are defined as:

$$\mathbb{H}_0 : \{0,1\}^* \to \mathbb{G}_1 \tag{2a}$$
$$\mathbb{H}_1 : \mathbb{G}_1 \to \mathbb{Z}_p^* \tag{2b}$$
$$\mathbb{H}_T : \mathbb{G}_T \to \mathbb{Z}_p^* \tag{2c}$$

The system defines attributes, that may be used to build ciphertext policies. This is denoted by the set S. All authorised clients are contained in the set $\mathcal{U}$. Each attribute is associated with an attribute group G, consisting of system clients that hold this attribute. The set of all attribute groups is specified as $\mathcal{G}$. The implemented ABE scheme consists of the following algorithms.

System Setup. The setup of an ABE system consists of three algorithms. They generate the system parameters *params* which are required to be publicly available to all system actors.

- *BaseSetup(λ, S)* $\rightarrow$ *$params_{base}$:* On input the security parameter λ and a set S of attributes, the algorithm outputs the base system parameters.
- *KeyAuthoritySetup($params_{base}$)* $\rightarrow$ *$params_{KA}$:* On input the base system parameters $params_{base}$, the algorithm selects a random element $q \in \mathbb{Z}_p^*$ as its secret parameter and outputs the KA parameters.
- *ReEncryptionSetup($params_{base}$)* $\rightarrow$ *$params_{RS}$:* On input $params_{base}$, the algorithms selects a random element $\alpha \in \mathbb{Z}_p^*$ as its secret parameter and outputs the public parameters $params_{RS}$.

After the RS registers its public parameter with the KA, the final system parameters are created and distributed to all actors in the system:

$$params = \{params_{base},\ params_{KA},\ params_{RS}\} \tag{3}$$

Key Creation and Update. In this system, the private key of a client is split 3-ways, where KA, RS and the CL each possess a part of the key. The process for generating these key parts is based on two algorithms:

- *KeyCreation(params, S)* $\rightarrow$ *IK_{KA}:* The key creation algorithm takes as input the base system parameters *params* and the CL's set of attributes S. It outputs the initial key IK_{KA}.
- *KeyUpdate(params, IK_{KA})* $\rightarrow$ *(PK_{KA}, PK_{RS}, PK_{CL}):* The key update algorithm takes as input the base system parameters *params* and the initial key IK_{KA}. It outputs the corresponding keys for KA, RS and CL.

Encryption & Decryption. Encryption & decryption consists of four algorithms, discussed in the following:

- *Encryption(params, $\mathbb{A}$, $\mathcal{M}$)* $\rightarrow$ *CT_{init}:* On input the base system parameters *params*, an access structure $\mathbb{A}$ and a plaintext $\mathcal{M}$ the algorithm outputs an initial ciphertext CT_{init}.
- *Re-Encryption(params, CT_{init})* $\rightarrow$ CT: On input the base system parameters *params* and the initial ciphertext CT_{init}, the re-encryption algorithm calculates and outputs the final ciphertext CT.
- *Partial-Decryption(CT, PK_{KA}, PK_{RS})* $\rightarrow$ $\mathcal{M}_{part}$: On input the final ciphertext CT and the keys PK_{KA} and PK_{RS}, the algorithm outputs the partially decrypted message $\mathcal{M}_{part}$.
- *Decryption($\mathcal{M}_{part}$, PK_{CL})* $\rightarrow$ $\mathcal{M}$: On input the partially decrypted message $\mathcal{M}_{part}$ and the client's key PK_{CL} the algorithm outputs the plaintext.

3 Evaluation

In ABE the main factors, influencing the performance of cryptographic operations are the number of used attributes and the security parameter. Hence, we evaluate

how different security levels and the number of attributes affect the system's performance. As our goal is to evaluate applicability in the IIoT, we first test all cryptographic operations on a cloud-instance. Next, we measure the execution time of encryption and decryption operations on a Raspberry Pi 3 Model B+, representing a low-performance IIoT device.

3.1 Test-Setup

To provide consistent test-results we performed a series of benchmark tests, repeating each test 100 times. Each iteration was run on a single core. We encrypted a random ciphertext of 16 bytes using AES with a 256-bit key. The key is then encrypted via ABE. We used the median to eliminate outliers that could occur due to initialisation operations our scheduling. We relied on the IAIK Provider for the Java^{TM} Cryptography Extension (IAIK-JCE) [15] and the IAIK ECCelerate^{TM} [14] for cryptographic functionality, such as bilinear pairings.

Settings. Since heavy operations are outsourced to a cloud server, we used two Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz running a 64-bit Linux Kernel to act as cloud and decryption server. To measure the performance of resource-constrained devices, we used a Raspberry Pi 3 Model B+.

Attributes. To demonstrate the impact of attributes, we perform our tests for a varying range of attributes. However, while Ambrosin et al. [2] claim that 30 attributes are a "range that represents a reasonable choice in real scenarios", Green et al. [11] show that policies can become highly complex in typical use cases. Hence, we tested our implementation for up to $n = 100$ attributes. To reflect this in our experiments, we generated policies in the form $(A_1, (A_2, (..., (A_{n-1}, A_n, AND), AND), AND), AND)$.

Security. To measure the impact of the security level, we evaluated our system for six security levels and associated curves, as recommended in [7]. As shown in [16], the security of pairing-based cryptography depends on the prime order n of the basepoint $P \in E(\mathbb{F}_q)$ and the embedding degree k. We evaluate our implementation with $n = 2^{160}$ to $n = 2^{512}$ and according k.

3.2 Performance

As argued by Wang et al. [23], ABE has a considerable performance overhead. This may lead to long execution times, especially on resource-constrained devices. This section provides an overview of the execution time of the different operations of our implementation, to show the applicability of our approach in an IIoT scenario.

Cloud-Server. Figure 1 depicts the execution time of different ABE operations using varying security levels and policies with up to 100 attributes. The examined prime order n ranges from 160-bit to 512-bit. This range comprises the recommended sizes for legacy (continued use or already deployed), near-term (at least ten years) and long-term (thirty to fifty years) use cases, as shown in Table 1.

Table 1. Key size recommendations published by ECRYPT-CSA [9] and NIST [5].

	Symmetric	Factoring Modulus	Elliptic Curve
Legacy	80	1 024	160-223
Near Term	128	3 072	256-383
Long Term	256	15 360	512+

The first row shows the timings corresponding to systems with legacy security requirements with prime order of 160-bit and 192-bit. The near-term scenarios are depicted in the three subsequent diagrams, with prime order 224-bit, 256-bit, and 384-bit. The long-term scenario is shown in the last diagram, with prime order 512-bit.

As expected, the execution time of all operations, except for the decryption operation, grows linearly with the number of attributes. As discussed, decryption is constant due to it being a simple El-Gamal operation. Thus, the execution time of the decryption step is independent of the number of attributes and only depends on the security level. Even with prime order 512-bit, the decryption operation in our comparison takes less than 30ms on our cloud server. While encryption and key-generation have a close to linear growth rate, both are still quite lightweight. For legacy and near-term applications, these operations take just a few tenths of a second, even when using up to 100 attributes. Only when aiming for long-term security, the required time to encrypt or generate keys while using policies with 100 attributes can take slightly longer than half a second. The most expensive operations in our scheme are the partial-decryption and the re-encryption operations. The execution time of partial-decryption exceeds five seconds in the long-term scenario when using a large number of attributes. However, in the more traditional near-term scenario using a limited number of attributes, these two operations still perform well by providing the results in less than a second on our test server.

IoT-Device. In our implementation, the encryption and decryption operations will typically be executed on an IIoT device. To depict such a resource-constrained device, we have chosen a Raspberry Pi 3 Model B+ for evaluation. Table 2 illustrates the execution times of the two operations. We have focused on prime order 256-bit, as such a security level provides near-term security. The encryption operation grows linearly in the amount of time. When using policies with up to 20 attributes, the operation is completed in less than a second and close to 4 seconds for up to 100 attributes. As the decryption operation is a constant time operation, it takes less than two-tenths of a second, irrespective of the number of attributes. Therefore, we propose to rely on OABE in applications which only require near-term security or for systems which do not rely on real-time computing.

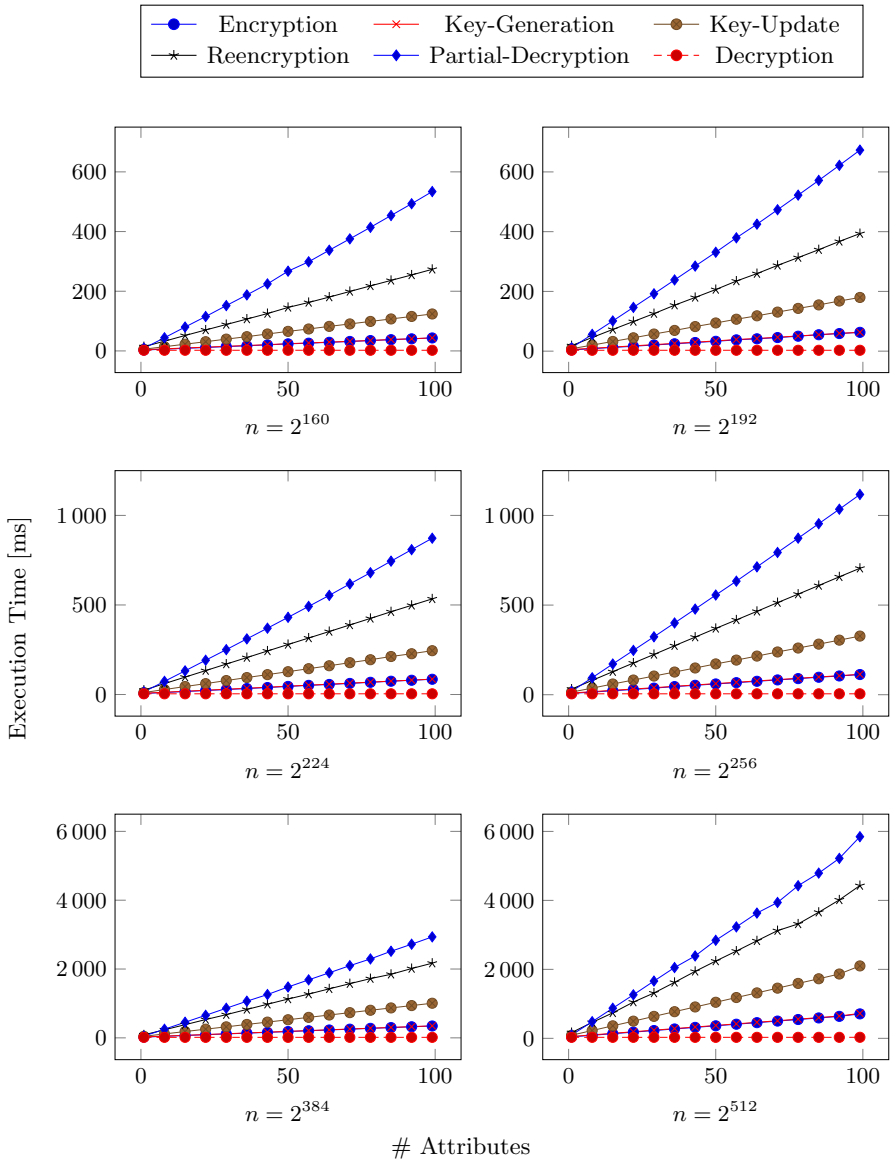

Fig. 1. Execution Time of ABE operations for different prime orders. Tests were executed on an Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz.

Table 2. Execution Time of ABE En- and Decryption on Raspberry Pi 3 Model B+

Security Level	Attributes	Median
256	1	231 ms
256	20	968 ms
256	40	1 738 ms
256	60	2 580 ms
256	80	3 306 ms
256	100	4 085 ms

(a) Execution Time of ABE Encryption

Security Level	Attributes	Median
256	1	155 ms
256	20	158 ms
256	40	154 ms
256	60	158 ms
256	80	157 ms
256	100	157 ms

(b) Execution Time of ABE Decryption

3.3 Data Overhead

ABE introduces additional data overhead, as access structures need to be embedded in the final ciphertext. Hence, we evaluate the additional payload of the resulting ABE ciphertext. Figure 2 illustrates this circumstance. As expected, the ciphertext size grows with the number of used attributes and the security level. As encrypted data is a constant size AES key, data overhead is independent of the underlying plaintext.

Our tests reveal that the ciphertext, generated by a client is approximately 1kB for 160-bit prime order and 100 attributes. In applications with near-term security, the ciphertext size can increase up to 40kB when 100 attributes are used. For 512-bit security, ciphertext size can even grow to 60kB for 100 attributes. In contrast, we compared these results with encryption in RSA. With RSA ciphertexts getting as large as 2kB for a 15360-bit modulus, at most.

We conclude that while 60kB, does not seem as large overhead, given today's infrastructure, it can have a significant influence on the network. Especially in IIoT environments, with typically a large amount of small messages in short times, this additional overhead may be larger than the actual message itself. Hence, ABE not only introduces computational overhead but can also put stress on the network itself.

3.4 Comparison: RSA and ECC

In order to get a better understanding of OABE compares to established encryption algorithms, we have conducted a comparison with RSA and ECC. The comparison can be found in Figure 3. The security level corresponds to the equivalent symmetric security level. While ABE offers a bigger feature set than both RSA and ECC, and, therefore, cannot directly be compared, it still gives

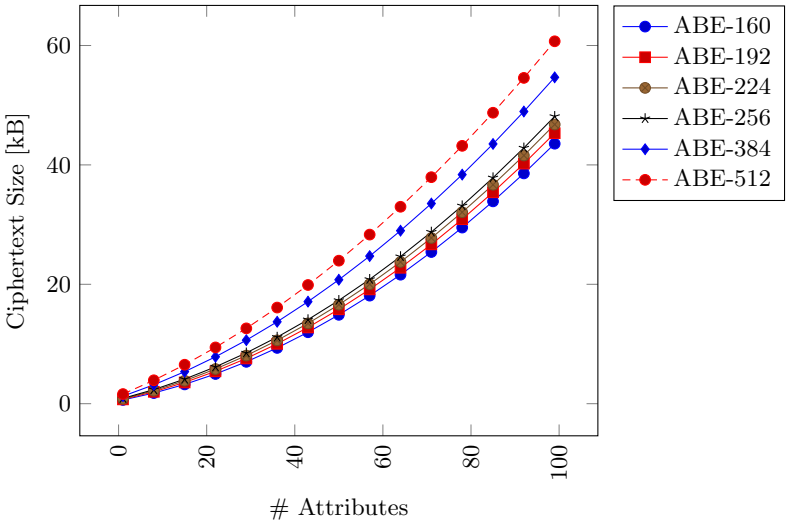

Fig. 2. Ciphertext size of ABE with prime order 160-bit to 512-bit. Tests were executed on an Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz.

an intuition of the overhead ABE introduces. For this experiment, all ABE operations have been executed using a policy with a single attribute. For better illustration, a logarithmic scale is used on the y-axis. The experiment compares all steps needed to generate keys, encrypt, and decrypt ciphertexts. We found that generating a private key for RSA can take a highly variable amount of time, from 2 to 35 minutes. We attribute this to the fact that two large random prime numbers have to be generated. Therefore, RSA key generation has been omitted from the Figure. As shown in the figure, ECC operations are nearly instant, taking less than a millisecond, even when operating at the highest security level. This is because ECC does not have a direct form of encryption. Instead, it can be used as a key agreement protocol. Hence, the timings for ECC encryption and decryption operation merely represent the timing for the corresponding AES operation. While encryption in RSA is also rather fast, decryption performance in RSA for near-term applications is already comparable to the operations in our ABE scheme. When using even higher RSA key sizes decryption starts to take considerably longer than partial-decryption and re-encryption in our ABE scheme.

3.5 Use-Case Description: Siemens MindSphere

Our experiments reveal that ABE can indeed be used in resource-constrained environments with near and long-term security requirements. Hence, we deployed our scheme to the Siemens MindSphere IoT. We evaluated, how sensitive (sensor) data, can be efficiently shared with multiple users while reducing overhead for IIoT devices. We found that encryption operations can be performed on resource-

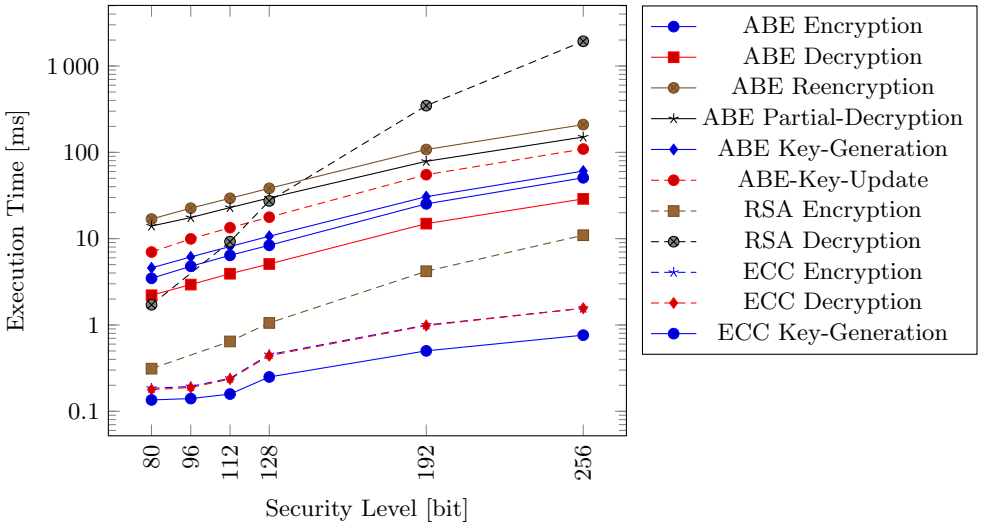

Fig. 3. Comparison of ABE with RSA, and ECC. For ABE, policies with a single attribute have been used. Tests were executed on an Intel(R) Xeon(R) CPU E5-2699 v4 @ 2.20GHz.

constrained devices, as long as there are no real-time computing requirements. Additionally, we discovered that attribute revocation, respectively key-update, which is essential in professional environments, does at most take two seconds, even for a high prime order. Hence, ABE can even provide a speed-up in revocation, in comparison to RSA.

4 Related Work

The calculation of pairings constitutes the most expensive part of ABE schemes. This is why one approach is to offload CPU intensive parts to dedicated servers, as can be seen in the work of Green et al. [11]. In their approach, any ABE ciphertext satisfied by that user's attributes is translated into a (constant-size) El Gamal-style ciphertext, regardless of the number of attributes. Hence, the resource-constrained device only has to decrypt the El Gamal-style ciphertext. Lin et al. [19] presented another approach, which offloads decryption operations. Their system is based on Hur's [13] architecture but adds another actor, which partially decrypts ciphertexts for users. A scheme proposed by Zhang et al. [24] allows outsourcing key generation, decryption, and encryption. They use two different non-colluding servers for key generation. They evaluate their system and compare it against plain CP-ABE. However, they do not assess their approach on more resource-constrained devices, such as smartphones or IoT devices.

Wang et al. [23] have conducted a performance evaluation of an ABE implementation on a smartphone and a laptop using an Intel CPU. In their study,

all operations were executed solely on the client device, without the help of a dedicated server. They compare different performance metrics while using up to 30 attributes. They examine three different security levels, ranging from an 80 to 128-bit security level. Encryption and decryption operations on the smartphone lead to execution times of multiple seconds, even when using the lowest examined security level. While this scenario on the utilised laptop resulted in a reasonable encryption and decryption time of less than a second, raising the security level also increased the execution time substantially. Ambrosin et al. evaluate the feasibility of ABE on different IoT devices [2] as well as smartphones [3], using both Intel and ARM CPUs. Similar to Wang et al. [23], their evaluation setup also only harnessed the computing power of the device. They also report execution times in the seconds for security levels of 80-bit to 128-bit. In contrast, this paper evaluates the feasibility of ABE with outsourced decryption using up to 100 attributes and security levels from 80-bit to 512-bit.

5 Conclusions

We implemented a purely Java and Kotlin based CP-ABE system with outsourced decryption. We presented performance data regarding execution time and data overhead. To prove the feasibility of ABE in the IIoT we conducted our experience on both resource-constrained devices and high-performance cloud instances. Additionally, we successfully deployed our implementation in the Siemens MindSphere cloud to achieve fine-grained access control.

As expected, our results show that the performance of our ABE implementation scales linearly with the number of used attributes and security level. However, as most of the work is done by cloud servers, clients are left with only lightweight operations. As a result, we achieve constant decryption time and encryption time scaling linearly with the number of attributes and security parameters. Hence, we achieve fast decryption, even on devices with limited computing power. Our experiments further reveal that data overhead of ABE can be up to 60kB. Thus ABE can introduce an additional burden on the network, in systems with a large number of small messages in a short time.

Summarising, we showed that ABE, while introducing some additional overhead, can indeed be successfully deployed in the IIoT to provide fine-grained access control. In the future we will further explore how our ABE implementation can be integrated into existing IIoT infrastructure and Identity and Access Management (IAM) solutions. We believe that fine-grained access control is crucial for the success of IIoT.

Acknowledgments

This research was conducted in cooperation with Graz University of Technology, the Institute for Applied Information Processing and Communications (IAIK), Know-Center GmbH and Siemens AG Austria, as part of the Siemens MindSphere research project.

Bibliography

[1] Agrawal, S., Chase, M.: FAME: Fast Attribute-based Message Encryption. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 665–682, CCS '17, ACM, New York, NY, USA (2017), ISBN 978-1-4503-4946-8, https://doi.org/10.1145/3133956.3134014

[2] Ambrosin, M., Anzanpour, A., Conti, M., Dargahi, T., Moosavi, S.R., Rahmani, A.M., Liljeberg, P.: On the Feasibility of Attribute-Based Encryption on Internet of Things Devices. IEEE Micro **36**(6), 25–35 (nov 2016), ISSN 0272-1732, https://doi.org/10.1109/MM.2016.101

[3] Ambrosin, M., Conti, M., Dargahi, T.: On the Feasibility of Attribute-Based Encryption on Smartphone Devices. In: Proceedings of the 2015 Workshop on IoT Challenges in Mobile and Industrial Systems, pp. 49–54, IoT-Sys '15, ACM, New York, NY, USA (2015), ISBN 978-1-4503-3502-7, https://doi.org/10.1145/2753476.2753482

[4] Atzori, L., Iera, A., Morabito, G.: The Internet of Things: A survey. Computer Networks **54**(15), 2787–2805 (2010), ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2010.05.010

[5] Barker, E.: Recommendation for Key Management Part 1: General. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD (jan 2016), https://doi.org/10.6028/NIST.SP.800-57pt1r4

[6] Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: 2007 IEEE Symposium on Security and Privacy (SP '07), pp. 321–334 (may 2007), https://doi.org/10.1109/SP.2007.11

[7] Brown, D.R.L.: SEC 2 : Recommended Elliptic Curve Domain Parameters. Tech. Rep. Standards for Efficient Cryptography, Certicom Research (2010)

[8] Chatterjee, S., Hankerson, D., Menezes, A.: On the Efficiency and Security of Pairing-Based Protocols in the Type 1 and Type 4 Settings. In: Hasan, M.A., Helleseth, T. (eds.) Arithmetic of Finite Fields, pp. 114–134, Springer Berlin Heidelberg, Berlin, Heidelberg (2010), ISBN 978-3-642-13797-6, https://doi.org/10.1007/978-3-642-13797-6_9

[9] ECRYPT – CSA: D5.4 Algorithms, Key Size and Protocols Report (2018). Tech. rep., H2020-ICT-2014 – Project 645421 (2018)

[10] Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98, CCS '06, ACM, New York, NY, USA (2006), ISBN 1-59593-518-5, https://doi.org/10.1145/1180405.1180418

[11] Green, M., Hohenberger, S., Waters, B.: Outsourcing the Decryption of ABE Ciphertexts. In: Proceedings of the 20th USENIX Conference on Security, p. 34, SEC'11, USENIX Association, Berkeley, CA, USA (2011)

[12] Henning Kargermann, Wolfgang Wahlster, J.H.: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Tech. Rep. April (2013)

[13] Hur, J.: Improving Security and Efficiency in Attribute-Based Data Sharing. IEEE Transactions on Knowledge and Data Engineering **25**(10), 2271–2282 (oct 2013), ISSN 1041-4347, https://doi.org/10.1109/TKDE.2011.78

[14] Institute for Applied Information Processing and Communications (IAIK): IAIK ECCelerate Library (2018), URL `https://jce.iaik.tugraz.at`

[15] Institute for Applied Information Processing and Communications (IAIK): IAIK-JCE (2018), URL `https://jce.iaik.tugraz.at`

[16] Koblitz, N., Menezes, A.: Pairing-Based Cryptography at High Security Levels. In: Smart, N.P. (ed.) Cryptography and Coding, pp. 13–36, Springer Berlin Heidelberg, Berlin, Heidelberg (2005), ISBN 978-3-540-32418-8, https://doi.org/10.1007/11586821_2

[17] Li, J., Chen, X., Li, J., Jia, C., Ma, J., Lou, W.: Fine-Grained Access Control System Based on Outsourced Attribute-Based Encryption. In: Crampton, J., Jajodia, S., Mayes, K. (eds.) Computer Security – ESORICS 2013, pp. 592–609, Springer Berlin Heidelberg, Berlin, Heidelberg (2013), ISBN 978-3-642-40203-6, https://doi.org/10.1007/978-3-642-40203-6_33

[18] Li, J., Huang, X., Li, J., Chen, X., Xiang, Y.: Securely Outsourcing Attribute-Based Encryption with Checkability. IEEE Transactions on Parallel and Distributed Systems **25**(8), 2201–2210 (aug 2014), ISSN 1045-9219, https://doi.org/10.1109/TPDS.2013.271

[19] Lin, G., Hong, H., Sun, Z.: A Collaborative Key Management Protocol in Ciphertext Policy Attribute-Based Encryption for Cloud Data Sharing. IEEE Access **5**, 9464–9475 (2017), https://doi.org/10.1109/ACCESS.2017.2707126

[20] Sadeghi, A.R., Wachsmann, C., Waidner, M.: Security and Privacy Challenges in Industrial Internet of Things. In: Proceedings of the 52Nd Annual Design Automation Conference, pp. 54:1—-54:6, DAC '15, ACM, New York, NY, USA (2015), ISBN 978-1-4503-3520-1, https://doi.org/10.1145/2744769.2747942

[21] Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) Advances in Cryptology – EUROCRYPT 2005, pp. 457–473, Springer Berlin Heidelberg, Berlin, Heidelberg (2005), ISBN 978-3-540-32055-5, https://doi.org/10.1007/11426639_27

[22] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakley, G.R., Chaum, D. (eds.) Advances in Cryptology, pp. 47–53, Springer Berlin Heidelberg, Berlin, Heidelberg (1985), ISBN 978-3-540-39568-3, https://doi.org/10.1007/3-540-39568-7_5

[23] Wang, X., Zhang, J., Schooler, E.M., Ion, M.: Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT. 2014 IEEE International Conference on Communications, ICC 2014 pp. 725–730 (2014), https://doi.org/10.1109/ICC.2014.6883405

[24] Zhang, R., Ma, H., Lu, Y.: Fine-grained access control system based on fully outsourced attribute-based encryption. Journal of Systems and Software **125**, 344–353 (2017), ISSN 0164-1212, https://doi.org/10.1016/j.jss.2016.12.018