Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-SIT

Seidlgasse 22 / 9, 1030 Wien
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

Inffeldgasse 16a, 8010 Graz
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

SECURITY ARCHITECTURE FOR DIGITAL TWINS

Version 1.0 vom 19.05.2021
Felix Hörandner – felix.hoerandner@iaik.tugraz.at

The concept of digital twins has gained traction over recent years. A digital twin is a digital representation of a physical object in the cloud, which is continuously synchronized in both directions. Consequently, owners can conveniently review the state of the digital twin's physical object and interact. Furthermore, collecting the data of one or multiple digital twins enables powerful computations. This concept has been applied in various fields, e.g., to monitor and optimize manufacturing processes or to provide precision medicine. However, as digital twins are maintained by cloud services that are not fully trusted, the confidentiality of sensitive digital twin data needs to be protected, which, unfortunately, has been neglected in related work so far.

This work proposes a security architecture and involved processes to provide end-to-end confidentiality for digital twin systems while keeping the concept flexible with regard to the sharing rules. Our concept builds upon key-policy conditional proxy re-encryption, in which ciphertext is associated with attribute sets upon which owners define policies. Owners generate re-encryption keys for such policies to enable the cloud service to translate selected subsets of the digital twin's encrypted data with authorized receivers. We integrate this protection mechanism into the processes to achieve the desired functionality of a digital twin system: to synchronize digital twin data to and from the cloud, to protect communication with external requesters, and to share subsets with processing services that offer computations on the digital twins' data. Finally, our performance evaluation highlights the feasibility and practical efficiency of this concept.

Table of Contents

1. Introduction

Advances in the Internet of Things domain gave rise to a new concept: Digital Twins [Barricelli, Fuller]. A digital twin is a digital representation of a physical object that is continuously kept up to date. This connection also extends in the opposite direction, as changes to the digital twin also trigger actions at the physical counterpart. Digital twins can, for example, be established for machinery in a production line, vehicles in transportation, or also humans in governmental or medical use cases. The digital twins reside in a central location, e.g., a cloud service, which governs the interactions with the twin and physical object. Gartner[1] named the concept of digital twins a top strategic technology trend of 2019.

Benefits: Digital twins enable a number of benefits:

- Data about the individual physical objects within a complex system can be reviewed and changed at a single, convenient-to-access location. For example, supervisors of a manufacturing process do not have to physically or digitally approach the various involved items to review their status or change their operational parameters.
- Interaction with the digital twins is routed through the central infrastructure, which simplifies the management of the data flow and enables to prevent attacks on individual devices. The physical devices only have to connect to the central service, which manages all further interactions. For example, the central service may apply firewall rules or rate-limiting, so attackers cannot repeatedly connect to a device to prevent that device from entering a sleep mode, draining its battery (i.e., denial of sleep attack).
- Powerful computations and simulations become possible as the up-to-date data of one or multiple digital twins is collected in the same location.

Use Cases: The concept of digital twins can be applied for various use cases [Barricelli, Fuller]. The following lines highlight a few examples.

- Qi and Tao [Qi] apply digital twins to monitor a manufacturing process, such that failures are detected, and the system can compute an optimized solution to address the problem.
- Kraft [Kraft] uses digital twins for aircraft components, where the collected sensor data enables simulations and prediction to reduce both the development and maintenance effort.
- Chen et al. [Chen] focus on vehicles in a smart city use case. By establishing digital twins of cars, these cars can be connected with each other, and traffic management can be implemented.
- Liu et al. [Liu] construct digital twins of patients using medical data. Wearables and in-house sensors collect these data. Their goal is to monitor the health status of the elderly and predict issues.

Challenge: Privacy but also Flexibility. The data of digital twins can be of sensitive nature, e.g., reveal business secrets or the users' health status. As these sensitive data are collected in a central service, they are a tempting target for misuse by insiders (e.g., cloud service providers and employees) or external attackers. Security, privacy, and trust have been identified as central challenges for the adoption of the digital twin's concept [Fuller]. Consequently, a rigorous access control scheme has to be enforced to protect the confidentiality of the digital twins' sensitive data. Such access control can be provided by end-to-end encryption mechanisms so that the central service cannot read the data. In a simple approach, the data of a digital twin could be encrypted with public-key encryption for the data owner and each intended receiver that needs read access to review the data or perform computations.

However, the employed cryptographic mechanisms also need to be sufficiently flexible. In systems where multiple actors participate by interacting with the digital twins, reviewing their data, and offering computations on the data, a complex set of end-to-end secured communication channels

[1] https://www.gartner.com/en/documents/3904569/top-10-strategic-technology-trends-for-2019-digital-twin

have to be established. Furthermore, as new actors should participate, or as other actors lose their trust, these end-to-end encryption relationships need to be adapted as well.

Related Work: Related work on digital twins has initially focused on the benefits the concept of digital twins can offer when applied to various use case domains, as surveyed in [Barricelli, Fuller]. Recently, initial work has been introduced to protect the data of the digital twins. For example, Gehrmann and Gunnarsson [Gehrmann] propose an abstract architecture on general security mechanisms in a digital twin system. Also, Dietz et al. [Dietz] focused on integrity protection by applying distributed ledger technology. However, there remains a gap in addressing privacy challenges while ensuring flexibility, as discussed above.

Contribution: In this work, we propose a security architecture for digital twins. This architecture employs key-policy conditional proxy re-encryption to ensure end-to-end confidentiality and enforce fine-grained access control rules defined by data owners on a cryptographic level. Furthermore, our system remains flexible, as proxy re-encryption enables us to only encrypt the data for the data owner while sharing of data is set up by data owners generating re-encryption keys towards authorized receivers. Our work highlights how this advanced cryptographic mechanism can be integrated into the individual actors within a digital twin system, such that 1) external parties can interact through end-to-end secure channels with the intended digital twin and physical object, 2) the owner is able to review all data on their digital twins without the central service (e.g., cloud) learning any data of the digital twin, and 3) sufficiently trusted services might receive subsets of the digital twin data to perform their processing functions (e.g., simulations, predictions, etc.).
The results of this work served as one contribution to our accepted research paper at SECRYPT 2021 [Hörandner].

Outline: Initially, Section 2 provides background information on key-policy conditional proxy re-encryption, which serves as a crucial building block within this work. Section 3 gives an overview of the system model applying digital twins. Section 4 elaborates on the instantiation of the individual processes and the integration of our cryptographic mechanisms. Section 5 presents a performance evaluation. Finally, Section 6 concludes this work.

2. Building Block: Key-Policy Conditional Proxy Re-Encryption

This section recalls key-policy conditional proxy re-encryption, which is an essential building block in the remainder of this work. Key-policy conditional proxy re-encryption is an extension of classical proxy re-encryption.

In **classical proxy re-encryption** [Blaze] (PRE), data encrypted for one entity can be transformed into ciphertext then encrypted for another entity. This re-encryption is done by a proxy, which does not learn the ciphertext's underlying plain message in any intermediate step. However, the proxy requires a re-encryption key. To enable re-encryption, the user for which the data was originally encrypted has to generate a re-encryption key from their private key material and the public key of the intended recipient.

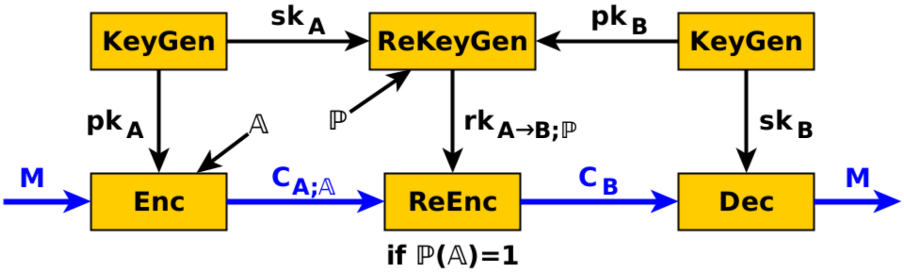

Figure 1: Key-Policy Conditional Proxy Re-Encryption

Key-policy conditional proxy re-encryption [Zhao] (KP-CPRE, shown in Figure 1) extends upon classical proxy re-encryption by introducing attributes and policies over these attributes that govern which ciphertexts can be re-encrypted by a given re-encryption key. During encryption, the sender associates a set of attributes with the ciphertext, which describes the content of the underlying plain message. Also, the user defines a policy over such attributes when generating a re-encryption key. Re-encryption only succeeds if the ciphertext's attributes satisfy the re-encryption key's policy.

3. System Model

This section first introduces the actors involved in our system model before outlining the main processes between those actors. Figure 2 illustrates the system model.

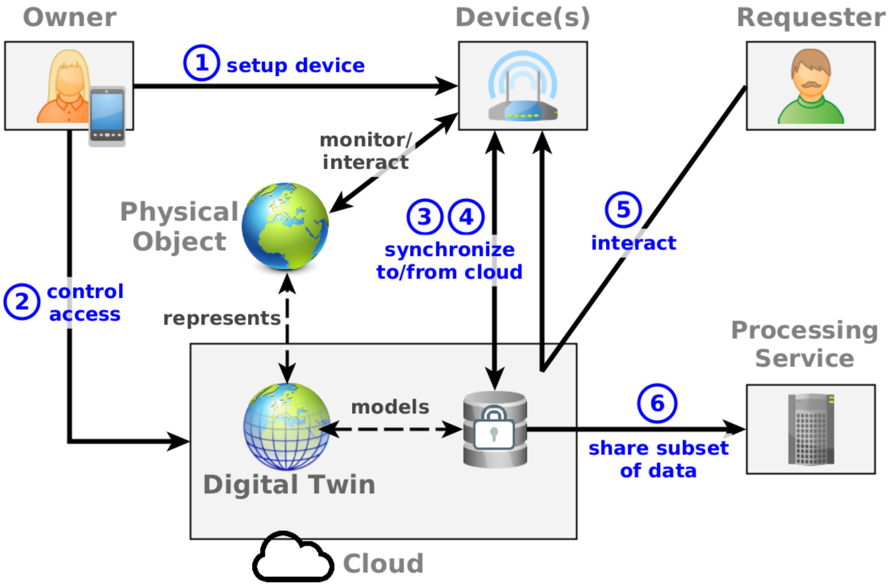

Figure 2: System Model

Actors:

- **Owners** own one or multiple physical objects as well as devices.
- **Physical objects** are of interest in various use cases. They need to be represented in the digital domain as digital twins.
- **Digital twins** are the reflection of physical objects. Changes to the digital twin also influence the physical object.
- The **cloud** (or a service in the cloud) serves as a central location to collect and share the digital twins' data.
- **Devices** form the technical link between physical objects and digital twins. They monitor the physical object and transmit changes to the digital twin in the cloud but also interact with the physical object upon changes applied to the digital twin in the cloud.
- **Requesters** want to interact with the physical object. These requesters may be other physical objects (or rather their digital twins) or other external entities.
- **Processing services** offer computations upon subsets of the digital twins' data. For example, these computations may be simulations or predictions.
- Of course, multiple actors can be instantiated by the same entity when applied to a use case.

Main Processes:

1. **Setup Device**: Initially, the owner has to set up their devices (1) to enable the devices to monitor and interact with their physical objects and (2) to establish the connection with a digital twin in the associated cloud service.
2. **Control Access**: Next, the owner defines access control rules, which determine who is able to read and write the digital twins' data.

3. **Synchronize to Cloud**: After their configuration, the devices monitor their physical objects. Once they observe a change in the objects, they communicate this change to the digital twin in the cloud service.
4. **Synchronize from Cloud**: If the digital twin's data in the cloud is modified, this change is also forwarded to the respective device. The device may then use this new data to interact with the physical object.
5. **Interact**: Requesters may wish to interact with a physical object. Therefore, they direct their request to the digital twin in the cloud. This digital twin then forwards the request to the device associated with the physical object at an appropriate time.
6. **Processing / Share Subset**: To enable computation on the digital twins' data, the cloud may forward a suitable subset to the processing service. Of course, the data is shared according to the users' policies established in Process 2 (Control Access).

4. Our Concept

This section introduces our security architecture for digital twins. Figure 3 provides a detailed view of steps and data flows in the individual phases. In the remainder of this section, we first describe the main aspects of our general approach before elaborating on the cryptographic operations performed at the involved actors for each process.

4.1. General Approach

Key-Policy Conditional PRE for End-to-End Encryption and Flexibility: Devices encrypt their data (about the physical object) for their owner before uploading it to the cloud service to build the digital twin. The owner can then download and decrypt these data, e.g., to review the current state of the digital twin and its physical object. Owners also control access to the digital twin: They generate re-encryption keys from their own private key towards the public key of a user-authorized receiver. With such a re-encryption key, the digital twin's data can be re-encrypted into ciphertext that can be decrypted by the selected receiver. Basically, proxy re-encryption decouples the encryption operation from later granting read access. Devices only need to know their owner to encrypt the ciphertext for this owner, while the owner can select authorized receivers at a later point in time. Consequently, proxy re-encryption offers flexibility to integrate further receivers and to react to changing trust relationships over time.

Attributes and Policies: Attributes and policies govern which subsets of data can be shared with others. Devices derive a set of attributes, e.g., based on the data's content or type, which they attach to ciphertexts during encryption. Based on these attributes, the owners define access policies as trees of logic gates (AND, OR). Within key-policy conditional PRE, the user generates re-encryption keys with respect to such access policies. Given a re-encryption key, the cloud service can re-encrypt the user's ciphertext if and only if the ciphertext's attributes satisfy the re-encryption key's policy.

Synchronized Data: Devices may model data about a physical object in various ways. We have chosen the representation as a state machine, as popularly adopted in related work. The device observes state changes in the physical object. By applying these state changes to the current state, a new and updated state is created. State changes may trigger interaction with the observed digital twin. Additionally, state changes may also originate from different sources, e.g., from interaction with external entities or from modifications at the digital twin that are synchronized to the device. These state changes should be meaningful on their own, e.g., to capture different aspects of a physical object so that it makes sense to share subsets of state changes.

Authenticity of State Data: Our system requires a mechanism to control who may create state changes, as anyone could encrypt data for the user. We employ digital signatures for this purpose. Naturally, devices may change their own state. Devices use their own signature and verification keys to sign their state changes before encrypting and verify the signature of state changes obtained from the cloud after decryption. Additionally, owners can also grant write permissions to others by signing a write token, which contains the other party's verification key. When writing a state change, the other party signs the state change with their own signing key. Both the writer-signed state change as well as the corresponding write token are synchronized to the device, where the write token is verified first before the contained writer's verification key is used to check the state change.

Improving Performance: Key-policy conditional PRE is used in a hybrid encryption setting, where the content is encrypted with a newly-generated symmetric key, while the symmetric key is protected via PRE. By re-using the symmetric key of data with the same attribute sets for a limited time frame, these symmetric keys have to be encrypted and decrypted less frequently, which reduces the required computational effort.

Interaction and Request Filtering: External entities, i.e., requesters, may want to interact with the owner's devices and their physical objects. Key-policy conditional PRE can also be applied to flexibly protect the confidentiality of the communication channel between requester and device. Requesters encrypt their request for the owner before sending the request to the cloud service. The cloud service may filter requests, e.g., via rate-limiting, to protect the devices and their resources. With a re-encryption key issued by the owner, the cloud re-encrypts the request for the device, which is then able to decrypt and process the request. A symmetric key embedded in the request may be used to protect the follow-up communication with the requester, e.g., to transmit responses securely.

Processing on Subsets: As the digital twins' data is encrypted, the cloud cannot perform computations on the state data. Instead, owners may place their trust in other parties (i.e., processing services), which are sufficiently trusted to learn an owner-specified subset of state data in order to perform computations on the plain data. Owners generate re-encryption keys for policies over the ciphertexts' attribute sets. With such re-encryption keys, the cloud can share subsets of state data with the processing services, which are then able to decrypt and process the data.

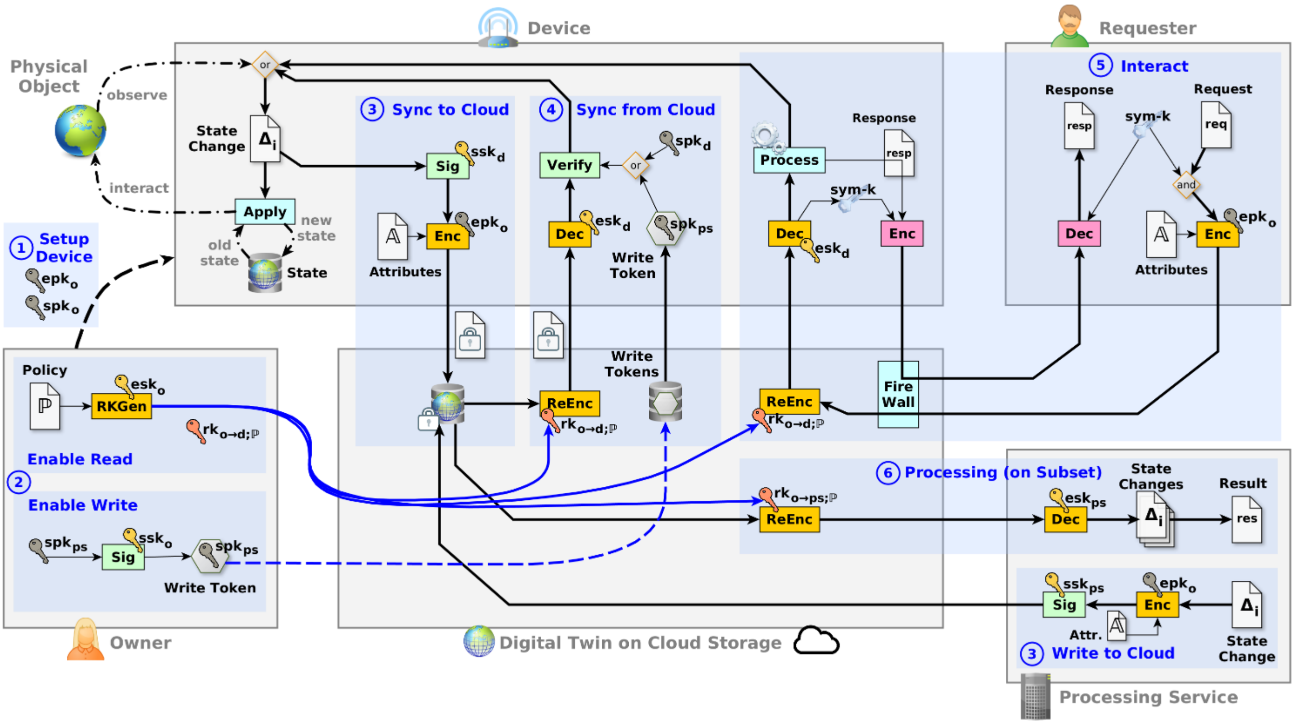

Figure 3: Process Phases of Our Concept

4.2. Process Descriptions

This section describes the individual phases of Figure 3 by elaborating on the aspects of the general approach presented above. For a more formal treatment, we refer the interested reader to the protocol definitions in the publication [Hörandner].

4.2.1. Setup Device

In the setup phase, the owner shares their public encryption key and their public verification key with the device. Also, the owner configures the device with an initial state, the id of the linked digital twin, and connectivity information to access the digital twin at the cloud storage. The device generates its own PRE key pair as well as a signature key pair. The owner proceeds to run "Control Access" (described in the next section) to share 1) the digital twin's state data and 2) external requests for the digital twin.

4.2.2. Control Access

To grant read access, the owner generates a re-encryption key from the owner's private key to the receiver's public key (e.g., the device's public key) for a suitable policy, limiting the amount of data that can be re-encrypted. To grant write permissions, the owner issues a write token. That is, the owner signs the tuple of the writer's public verification key and the digital twin's id. The re-encryption key and write tokens are installed at the cloud service.

4.2.3. Synchronization to the Cloud

Initially, the device observes a state change in the physical object and applies this state change to the current state to obtain a new state. To synchronize this state change with the cloud, the device signs the change with its signature key. Then, the device encrypts the signed state change with the owner's public encryption key for a suitable attribute set, e.g., derived from the content or type of the state change. This encryption process includes generating a symmetric key, using the symmetric key to encrypt the content, and using the PRE-key to encrypt the symmetric key. If desired, the symmetric key can be re-used for a time frame. In this case, the device remembers and re-uses the symmetric key while skipping the PRE-encryption of this key. Finally, the device uploads the signed and encrypted change to the cloud service to update the digital twin. If other entities write to the digital twin, they perform these operations and use their own signature keys to sign the state changes.

4.2.4. Synchronization from the Cloud

We assume the owner has performed "Control Access" such that the device is able to read the digital twin's data (i.e., state changes). Upon receiving an encrypted state change that did not originate from the device, the cloud service re-encrypts the state change for the device. That is, the encrypted symmetric key of the state change is re-encrypted for the device, while the symmetrically encrypted content remains untouched. If the symmetric keys are re-used for a time frame, each involved symmetric key only has to be made available for a device once.

The device obtains the re-encrypted state change and decrypts it. Next, it verifies whether the state can be accepted by verifying the signature on the data. The signature needs to be verifiable with a) the device's verification key, b) the owner's verification key, or c) with the verification key of a write token that has been issued by the owner for the respective identifier of the digital twin. If the signature can be verified successfully, the state change is applied to the current state to reach a new state, which might trigger interaction with the physical object.

4.2.5. Interaction

To enable interaction with a digital twin (or rather the device maintaining the digital twin's physical object), we assume the owner has performed "Control Access" such that a re-encryption key from the owner towards the device for requests to the digital twin's identifier has been installed at the cloud service. Initially, the requester obtains the identity of the digital twin and the public encryption key of the twin's owner out of band (e.g., by scanning a QR code, receiving a wireless announcement, or engaging with discovery services). The requester encrypts the request for the owner's public encryption key and associates an attribute set, which specifies a) the data is a request rather than a state change, b) the digital twin's identifier, and c) any additional suitable attributes. To protect follow-up communication, the request may also contain a symmetric key, or the symmetric key of the applied hybrid encryption may be re-used.

The cloud receives the encrypted request and may apply filtering logic to protect the system's resources, e.g., prevent attackers from periodically pinging a device with irrelevant requests that prevent the device from going into a sleep mode. Next, the cloud applies the re-encryption key obtained from the owner to transform the request for the related device.

The device decrypts and processes the re-encrypted request. If the request led to a state change, this change is applied, which might trigger to interaction with the physical object. Also, such a state change would again be synchronized to the cloud as described previously.

4.2.6. Processing

The processing of subsets of the digital twin's data must be authorized by the digital twin's owner in a first step. Therefore, we assume the owner has performed "Control Access" such that a re-encryption key from the owner to the processing service is installed at the cloud service. According to the policy of the re-encryption key, the cloud may transform a subset of the owner's data (of their digital twins) into ciphertext for the processing service. This re-encrypted ciphertext can then be decrypted by the processing service with its own private decryption key. The decrypted data then enables computation. The owners may only share subsets with different processing services, which are trusted to keep the plain text private. Therefore, owners are not confronted with the risk that one central entity learns the whole sensitive digital twin data, as the cloud service would in a plain-text approach. Instead, owners can make selective decisions on disclosing data while still allowing computations on their digital twin data.

5. Evaluation

This section presents the performance evaluation of our concept to highlight its feasibility. Our concept relies on various cryptographic schemes. We set AES as symmetric encryption scheme and ECDSA as digital signature scheme, while we build upon the key-policy conditional proxy re-encryption scheme by Zhao et al. [Zhao]. As the performance characteristics of AES and ECDSA are well established, this section focuses on key-policy conditional proxy re-encryption.

KP C-PRE Implementation: The implementation of Zhao et al.'s scheme [Zhao] uses parameters to achieve 128-bit security according to the recommendation of NIST [NIST]. This implementation builds upon the RELIC toolkit [RELIC], which provides the mathematical foundation, i.e., functions to operate on elliptic curves to evaluate bilinear mappings. Only a single thread is used within the implementation.

Methodology: In the performance benchmark, we measure the execution time of the involved cryptographic algorithms. Initially, the benchmark generates two key pairs (via KeyGen) before generating a re-encryption key from the first to the second key pair for a given policy (via RKGen). Then, the benchmark creates a random 128-bit AES key and encrypts that AES key for the first PRE key pair with an attribute set (via Enc). With the re-encryption key, the ciphertext for the first key pair is transformed into a ciphertext for the second key pair (via ReEnc). Finally, this translated ciphertext is decrypted with the private key of the second key pair.

We have performed the performance benchmarks for various sizes of the policy and the attribute sets. The used policies are shaped as binary trees of AND/OR nodes with attributes in the leaf nodes (i.e., the policy's size is the sum of nodes). These leaf nodes form the attribute sets.

The benchmarks were run 100 times to collect an average execution time and the standard deviation for each algorithm. We have evaluated the performance on three platforms that are relevant in digital twin deployments: (1) a PC with AMD Ryzen 5600X CPU, (2) a OnePlus 6T mobile phone, and (3) a Raspberry Pi 4B as an IoT device.

Table 1: Performance Measurements in Milliseconds of the Key-Policy Conditional PRE Implementation

Policy Size	Attrs. Size	KeyGen	RKGen	Enc	ReEnc	Dec
			PC (AMD Ryzen 5600X)			
3	**2**	0.39 ± 0.01	2.04 ± 0.04	2.62 ± 0.05	4.22 ± 0.01	2.36 ± 0.04
7	**4**	0.39 ± 0.01	3.54 ± 0.06	3.00 ± 0.06	4.86 ± 0.01	2.36 ± 0.04
15	**8**	0.39 ± 0.01	6.52 ± 0.07	3.75 ± 0.06	6.13 ± 0.01	2.36 ± 0.03
31	**16**	0.39 ± 0.01	12.52 ± 0.11	5.27 ± 0.11	8.71 ± 0.01	2.37 ± 0.03
			Phone (OnePlus 6T)			
3	**2**	6.93 ± 0.16	37.35 ± 0.68	46.96 ± 0.84	76.85 ± 0.21	41.32 ± 0.73
7	**4**	6.93 ± 0.14	65.34 ± 1.09	54.79 ± 1.08	90.22 ± 0.18	41.29 ± 0.61
15	**8**	6.93 ± 0.15	121.46 ± 1.41	70.11 ± 1.29	117.06 ± 0.74	41.38 ± 0.64
31	**16**	6.91 ± 0.14	233.54 ± 1.94	100.72 ± 1.73	170.56 ± 0.25	41.30 ± 0.63
			IoT Device (Raspberry Pi 4B)			
3	**2**	12.81 ± 0.25	70.27 ± 1.28	88.41 ± 1.48	143.72 ± 0.40	78.50 ± 1.30
7	**4**	12.81 ± 0.28	123.03 ± 1.52	102.48 ± 1.74	167.70 ± 0.14	78.36 ± 1.26
15	**8**	12.75 ± 0.27	228.59 ± 2.29	131.16 ± 2.16	215.82 ± 0.15	78.18 ± 1.28
31	**16**	12.74 ± 0.29	439.65 ± 2.90	187.72 ± 3.16	312.26 ± 0.48	78.36 ± 1.18

Results: Table 1 presents the results of our benchmarks of the individual cryptographic algorithms of the key-policy conditional PRE implementation for different sizes of the used policies and attributes on a PC, mobile phone, and IoT device. Encryption and decryption might be performed on low-power IoT devices, which take <188ms and <79ms, respectively, even for large attribute sets. Re-Encryption is performed on more powerful machines by the cloud service. In our measurements, a desktop PC takes <9ms for such re-encryption operations. These operations only have to be performed once per time frame if the symmetric key is re-used. While key generation requires little execution time even on our IoT device (<13ms), the time to generate a re-encryption key grows with the size of the policy. Nevertheless, <234ms is reasonable if the owner uses a phone to occasionally manage sharing permissions by generating re-encryption keys for large policies. Overall, the low execution times on our relevant platforms demonstrate the feasibility of our concept.

6. Conclusion

This work has proposed a security architecture and processes to protect the data of digital twins without sacrificing their benefits or requiring extensive maintenance effort to accommodate changes in data sharing relationships. Key-policy conditional proxy re-encryption is a key enabler that lies at the core of our concept. Devices only have to encrypt data for their owner, while the owner defines fine-grained access control rules that are enforced on a cryptographic level at a later point by generating a re-encryption key. This access control mechanism not only specifies which entity may decrypt the owner's digital twin data but also enables to limit the shared data set according to attributes attached to the ciphertexts that satisfy the policies of the re-encryption keys. Additionally, key-policy conditional proxy re-encryption can also be applied to (1) protect the communication with external requesters even if it is routed through the cloud service and (2) share subsets of digital twin data with processing services that perform computations on the owners' data, given the owners' permission. Our performance evaluation demonstrates the feasibility of the concept while considering devices with different computation resources, ranging from powerful desktop PCs to more constrained IoT devices, such as a Raspberry Pi 4B.

7. Bibliography

[Hörandner] Hörandner, Felix and Bernd Prünster (2021). "Armored Twins: Flexible Privacy Protection for Digital Twins through Conditional Proxy Re-Encryption and Multi-Party Computation". SECRYPT. In Press. 2021

[Barricelli] Barricelli, B. R., Casiraghi, E., and Fogli, D. (2019). "A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications". IEEE Access, 7, pp. 167653–167671.

[Fuller] Fuller, A., Fan, Z., Day, C., and Barlow, C. (2020). "Digital Twin: Enabling Technologies, Challenges and Open Research". IEEE Access, 8, pp. 108952–108971.

[Qi] Qi, Q. and Tao, F. (2018). "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison". IEEE Access, 6, pp. 3585–3593.

[Kraft] Kraft, E. M. (2016). "he Air Force Digital Thread/Digital Twin - Life Cycle Integration and Use of Computational and Experimental Knowledge". 54th AIAA Aerospace Sciences Meeting, https://arc.aiaa.org/doi/pdf/10.2514/6.2016-0897

[Chen] Chen, X., Kang, E., Shiraishi, S., Preciado, V. M., and Jiang, Z. (2018). "Digital Behavioral Twins for Safe Connected Cars". In: MoDELS. ACM, pp. 144–153

[Liu] Liu, Y., Zhang, L., Yang, Y., Zhou, L., Ren, L., Wang, F., Liu, R., Pang, Z., and Deen, M. J. (2019). "A Novel Cloud-Based Framework for the Elderly Health-care Services Using Digital Twin". IEEE Access, 7, pp. 49088–49101.

[Gehrmann] Gehrmann, C. and Gunnarsson, M. (2020). "A Digital Twin Based Industrial Automation and Control System Security Architecture". IEEE Trans. Ind. Informatics, 16(1), pp. 669–680.

[Dietz] Dietz, M., Putz, B., and Pernul, G. (2019). "A Distributed Ledger Approach to Digital Twin Secure Data Sharing". In: DBSec. Vol. 11559. LNCS. Springer, pp. 281–300.

[Blaze] Blaze, M., Bleumer, G., and Strauss, M. (1998). "Divertible Protocols and Atomic Proxy Cryptography". In: EUROCRYPT. Vol. 1403. LNCS. Springer, pp. 127–144.

[Zhao] Zhao, J., Feng, D., and Zhang, Z. (2010). "Attribute-Based Conditional Proxy Re-Encryption with Chosen-Ciphertext Security". In: GLOBECOM. IEEE, pp. 1–6.

[NIST] National Institute of Standards & Technology (2016). SP 800-57. Recommendation for Key Management, Part 1: General (Rev 4). Tech. rep. NIST.

[RELIC] Aranha, D. F. and Gouvêa, C. P. L. (2021). RELIC is an Efficient LIbrary for Cryptography. https://github.com/relic-toolkit/relic. Accessed: 2021-02-12.