

CAKE: Sharing Slices of Confidential Data on Blockchain

Edoardo Marangone¹ , Michele Spina¹ , Claudio Di Ciccio² , and Ingo Weber³ 

¹ Sapienza University of Rome, Rome, Italy

marangone@di.uniroma1.it; spina.1711821@studenti.uniroma1.it

² Utrecht University, Utrecht, Netherlands

c.diciccio@uu.nl

³ Technical University of Munich, School of CIT, and Fraunhofer Gesellschaft, Munich, Germany

ingo.weber@tum.de

Abstract. Cooperative information systems typically involve various entities in a collaborative process within a distributed environment. Blockchain technology offers a mechanism for automating such processes, even when only partial trust exists among participants. The data stored on the blockchain is replicated across all nodes in the network, ensuring accessibility to all participants. While this aspect facilitates traceability, integrity, and persistence, it poses challenges for adopting public blockchains in enterprise settings due to confidentiality issues. In this paper, we present a software tool named Control Access via Key Encryption (CAKE), designed to ensure data confidentiality in scenarios involving public blockchains. After outlining its core components and functionalities, we showcase the application of CAKE in the context of a real-world cyber-security project within the logistics domain.

Keywords: Cyphertext Policy · Attribute-Based Encryption · Cryptography · Blockchain technology · Smart Contract

1 Introduction

Blockchain technology is increasingly being applied in information systems of diverse enterprise domains due to its capacity to facilitate the establishment and execution of cooperative processes involving multiple parties with limited mutual trust [28,26]. The decentralized structure of public permissionless blockchains ensures that each participant in the network possesses a replicated ledger, thereby allowing for unrestricted accessibility of all data. This transparency, in conjunction with the immutability of data and the non-repudiable nature of transactions, makes blockchains a robust foundation for verifiable and trustworthy interactions.

In scenarios where there is a lack of mutual trust among parties, hiding some data from the majority of users can be advantageous. Indeed, when blockchain technology is discussed, the security and privacy topics are the critical issues and their importance is underlined and well recognized [32,9,6]. A solution to guarantee data secrecy and confidentiality among parties was presented in [20] under the name of Control Access via Key Encryption (CAKE). The parties can securely exchange information using the CAKE architecture, hiding data or parts thereof from others. This paper demonstrates the CAKE tool, illustrating its implementation and the newly introduced features. We used CAKE as a core component of a larger platform designed

and realized in the context of a national cyber-security research and innovation project for international logistics: Blockchain Register for Import-Export (BRIE).⁴ We employ the case study to showcase the maturity and integration of the tool within a real-world setting. At large, our research provides security-minded practitioners with a tool to securely transact confidential data: A whole public blockchain network permanently stores the transactions attesting to the validity and integrity of the data, but only authorized parties can read the actual information in-clear.

In the following, Sect. 2 outlines the CAKE architecture and the core concepts it builds upon. In Sect. 3, we demonstrate our proof-of-concept implementation with the BRIE real-world use case. Section 4 provides implementation details about our tool. Section 5 presents the related work in the literature. Finally, Sect. 6 concludes the paper and draws some avenues for future work.

2 Core concepts and tool architecture

In the following, we outline the key methodologies and techniques underpinning our solution. Equipped with these notions, we describe the core components of CAKE thereafter.

Core concepts. Distributed Ledger Technologies (DLTs) are protocols that facilitate transactional storage, processing, and validation within a decentralized network without the need for central authorities or intermediaries. These transactions come along with cryptographic signatures. The resulting shared transaction log collectively constitutes a ledger accessible to all participants in the network. In a **blockchain**, a specific type of DLT, transactions are organized in blocks, which are linked to form an append-only singly linked list, namely a chain. DLTs, including blockchains, are tamper-resistant thanks to cryptographic techniques such as hashing and decentralized validation of transactions. Public blockchain platforms such as Ethereum [30] and Algorand [5] require the payment of fees for submitting and processing transactions on the platform. These platforms enable the utilization of **smart contracts**, which are programs deployed, stored, and executed directly on-chain [7,33]. Ethereum and Algorand support smart contracts through the Ethereum Virtual Machine (EVM) and the Algorand Virtual Machine (AVM), respectively. These contracts are deployed and invoked via transactions. Their code is stored on the blockchain and executed by the nodes within the distributed system. The results of contract invocations are subject to blockchain consensus, thereby being verified by the network and completely traceable. To reduce the costs associated with invoking smart contracts, external Peer-to-Peer (P2P) systems are employed for storing significant volumes of data [31]. Among the facilitating technologies is the **InterPlanetary File System (IPFS)**. IPFS is a distributed system utilizing a Distributed Hash Table (DHT) to distribute stored files across multiple nodes. It employs hashing to generate a uniquely identifying resource locator for every file. In a conventional blockchain integration, the locator is subsequently transmitted to a smart contract for permanent storage on the blockchain [16]. Notice that such an address is content-based: changing even a single bit in the data entails the modification of the hash, thus the original locator does not match the modified data. **Attribute-Based Encryption (ABE)** is a type of public-key encryption scheme where the ciphertext (i.e., an encrypted plaintext) and its corresponding decryption key are linked via attributes [4,25]. In Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [3,15], a set of such attributes is assigned to potential users. Policies are linked to cipher-

⁴<https://brie.moveax.it/en>, accessed 2024-03-11

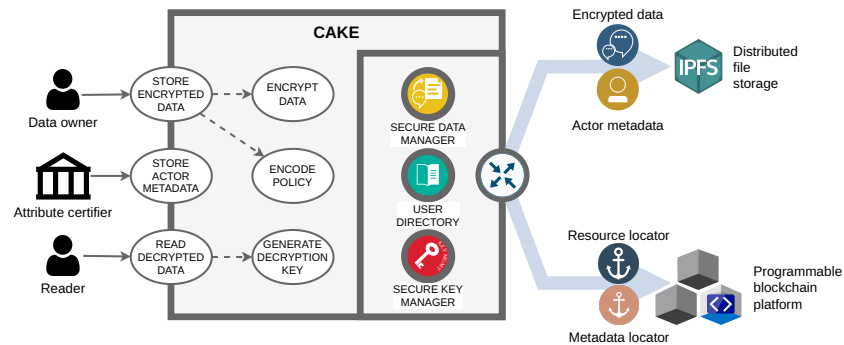


Fig. 1: An overview of the CAKE architecture

texts and articulated as propositional formulae over the attributes. The formulae are evaluated to determine whether a user holds the necessary properties to grant access to the unencrypted data.

Tool architecture. Figure 1 depicts the core traits of CAKE’s architecture. It offers three core functionalities, drawn as use cases in the figure: (i) storing encrypted data, which in turn requires the encryption of the transmitted artifacts via ABE and the encoding of ciphertext policies to control access; (ii) storing actor metadata, mapping ABE attributes to specific users to later determine their suitability to read the stored information; (iii) reading decrypted data, which entails the generation of decryption keys depending on the attributes that the requesting user bears. Those three functionalities are realized by the interplay of three basic components: (i) the Secure Data Manager (SDM), which is responsible for the encryption of data based on the policies and the subsequent storage thereof; (ii) the User Directory (UD), recording the association of users with the attributes they bear in the context of the collaborative process enactment; (iii) the Secure Key Manager (SKM), which generates decryption keys for users who wish to read data in clear based on their attributes. CAKE is interfaced with an IPFS distributed file storage to save the files with encrypted data and with the actor metadata. It resorts to a programmable blockchain platform to record the locators of those files via smart contracts. The *attribute certifier* writes the actor metadata via UD in a file uploaded onto IPFS, the locator of which is later stored on-chain. A *data owner*, namely a process actor that wants to share data with selected users, sends the data in clear and the policies to encrypt it to the SDM. The latter performs the encryption based on the encoded policy, stores the secured data on IPFS, and notarizes the locator thereof on chain. To access the data, a *reader* asks for a decryption key to the SKM, which in turn retrieves the users’ attributes from the UD and uses them to generate the key. If those attributes satisfy the policy originally used to encrypt the data, the reader can access the contents in clear. Note that the transactions stored on-chain do not disclose core information. The hash-based resource locator is stored on chain, but the sender of the transaction is the SDM itself, and the recipient is a smart contract. Thus, even if any network node can fetch the public ledger, it cannot extract any information on the exchanged data, its owner, or the intended readers therefrom.

The detailed explanation of the above passages goes beyond the scope of this demo paper. More information can be found in the paper describing the CAKE approach [20]. Next, we provide further details about the implementation of CAKE.

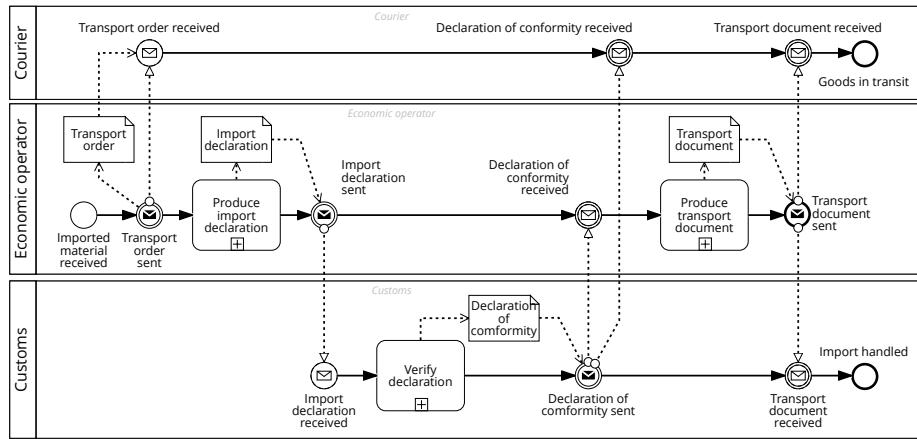


Fig. 2: An excerpt of a process workflow in the BRIE project

3 Demonstration through a real-world case study

BRIE (Blockchain Register for Import-Export)⁴ is a project aimed at the design and realization of a blockchain-based solution for the monitoring and optimization of international logistics processes. The primary goal is to support stakeholders by facilitating the tracking of shipments, the effective management of pertinent documentation, and the establishment of novel synergies to enhance the management, storage, and transit of goods within Europe.

Figure 2 shows a Business Process Model and Notation (BPMN) collaboration diagram [8] illustrating a model fragment of a process handled in the BRIE project. The process actors involved are the *Courier*, the *Economic Operator*, and *Customs*. A new process instance begins when the Economic Operator sends a *transport order* to a Courier. Then, the Economic Operator compiles an *import declaration* for Customs. This document describes the goods, the country of destination, the buyer, and the selected courier for the import. The import declaration is subsequently verified and confirmed by Customs emitting a *declaration of conformity* which can be accessed both by the Economic Operator and the Courier. After this confirmation, the Economic Operator produces a *transport document* with the mode of transport, the Courier, the data and address for goods collection, the expected delivery date, and the delivery address.

Since we utilize ABE, we associate the process actors with users, each having attributes that characterize their role. We assume here that the importing country's Chamber of Commerce and the competent ministerial body acted as attribute certifiers to register the actors involved as licensed operators. In our example, the involved Courier, Economic Operator and Customs are associated with attributes *courier*, *economic_operator*, and *customs*, respectively. We represent the participation in the process identified by number 29837 with an attribute recalling the number itself (29837) for short.

In ABE, policies are linked with ciphertexts and articulated as propositional formulae over attributes. They serve to ascertain whether a user is authorized for access. Table 1 contains the encryption policies associated with the aforementioned documents. The one related to the import declaration, e.g., is expressed as $(29837 \text{ and } ((\text{economic_operator}) \text{ or } (\text{customs})))$ as it is meant to be accessed by the Economic Operator and Customs involved in case 29837. CAKE encrypts every document with the corresponding policy to let only the

Table 1: Documents exchanged in Fig. 2 for process instance 29837

Document	Sender	Recipients	Policy
Transport order	Economic Operator	Courier	(29837 and ((economic_operator) or (courier)))
Import declaration	Economic Operator	Customs	(29837 and ((economic_operator) or (customs)))
Declaration of conformity	Customs	Economic Operator; Courier	(29837 and ((customs) or (economic_operator) or (courier)))
Transport document	Economic Operator	Courier; Customs	(29837 and ((economic_operator) or (customs) or (courier)))

intended actors read it. Notice that if the writer of a document wants to be able to decrypt the shared document later on, they need to include themselves in the set of authorized readers. Once the document is encrypted, it is uploaded on IPFS, and the resulting resource locator (e.g., `QmTnDqWf [. . .] i9wZUGYp`), is stored on the blockchain alongside a unique message ID.

To access the import declaration later, Customs must ask for a decryption key and obtain the document in clear. In this example, the attributes of Customs satisfy the policy used to encrypt the import declaration, so they can obtain the document and read its decrypted content. On the contrary, the Courier has attributes that do not satisfy the policy, so they cannot access that document's content. After the Customs agency verifies the conformity of the declaration, the process can progress. As for the transport document, all the three players considered in this example can read the document. Their attributes satisfy the ciphertext policy, so their key is apt for decryption.

The above scenario was used during the final review meeting of the BRIE project and involved a larger integrated platform for logistics data collection and exchange. Next, we provide an overview of the implementation of CAKE and its integration with the BRIE platform.

4 Implementation

We implemented CAKE and its communication channels in Python. The CAKE components expose their interfaces as APIs to a bundled service provider to ease communication and integration with other systems. The communication infrastructure is thus external to the blockchain and IPFS and relies on the Secure Sockets Layer (SSL) protocol. We used this protocol to mitigate the risk of packet sniffing by potential malicious third parties aiming to intercept the transmitted data. Moreover, the communication from the data owner to the Secure Data Manager and from the reader to the Secure Key Manager is preceded by an initial authentication phase via a preliminary handshake. Without this security measure, any malicious peer could submit requests on behalf of the authentic reader, having obtained their address and conjecturing a file to which access might be granted. CAKE allows for the encryption of different types of documents. It is possible to handle a single text file, with the option of applying different policies to different parts thereof. Alternatively, multiple text or binary documents can be uploaded, each being associated with a separate policy.

The source code of CAKE is openly available at <https://github.com/apwbs/CAKE>. In the code repository, we provide two implementations of CAKE, distributed within Docker containers: one for the EVM and one for the AVM. The smart contracts we employ are encoded in Solidity for the EVM and in PyTeal for the AVM. They are deployed on the Sepolia testnet⁵ and the Algorand testnet,⁶ respectively. The AVM-based version of CAKE was used for the BRIE project described in Sect. 3. To integrate our tool with the BRIE platform, we developed

⁵<https://sepolia.etherscan.io/>, accessed 2024-03-11

⁶<https://app.dappflow.org/dashboard/home>, accessed 2024-03-11

a plug-in named Secrecy and Privacy Enhancer for Ciphred Knowledge (SPECK, available at github.com/MichaelPlug/SPECK), including a collection of scripts to automatically retrieve information from a shared data repository and interact with the APIs of CAKE.

Next, we provide a comparative summary of research endeavors that relate to CAKE.

5 Related work

Numerous research endeavors have focused on automating collaborative processes utilizing blockchain technology. Weber et al. [28] introduce a method leveraging this technology to facilitate the conduction of business among parties in the absence of mutual trust. Their work demonstrates how actors can mutually agree on executed behaviors without relying on a central enforcement authority. López Pintado et al. [17] introduce Caterpillar, a process execution engine based on Ethereum. Caterpillar enables users to generate process instances and monitor their progress. Madsen et al. [18] investigate the execution of distributed declarative workflows, particularly in situations involving collaboration among adversarial entities. Corradini et al. [6] introduce ChorChain, a tool that executes and monitors process choreographies on the Ethereum blockchain platform. These studies enhance the fusion of blockchain and process management, unlocking security and traceability opportunities. However, they lack mechanisms to ensure fine-grained access control over data stored on a public platform. In contrast, our work addresses this aspect in a collaborative business process scenario.

Another research area within our domain concerns the privacy and integrity of data stored on-chain. Hawk [14] is a decentralized system that leverages user-defined private smart contracts to execute cryptographic techniques autonomously. In contrast, our approach eliminates the need for custom smart contract encoding, as it relies on on-chain policies for message encryption. Rahulamathavan et al. [24] introduce a novel privacy-preserving blockchain architecture tailored for Internet of Things (IoT) applications, utilizing Attribute-Based Encryption (ABE) techniques. While we also utilize ABE, we aim to augment existing software architectures. In contrast, their model alters the blockchain protocol itself. Benhamouda et al. [2] propose a solution enabling a public blockchain to function as a repository for confidential data. In their system, a secret is initially stored on the blockchain, followed by the specification of conditions for its release, with the secret disclosed only if these conditions are satisfied. In contrast, our approach involves the utilization of shared secrets among components. However, it does not entail utilizing the blockchain as a storage system for secret data or disclosing the secret. Differently from these methodologies, our approach addresses the challenge of regulated data access within a multi-party process scenario. This scenario involves the exchange of multiple information artifacts, where various actors can read specific segments of messages based on access policies.

Wang et al. [27] propose an electronic health record framework integrating Attribute-Based Encryption (ABE), Identity-Based Encryption (IBE), and Identity-Based Signature (IBS) mechanisms with blockchain technology. Unlike the CAKE model, this system design empowers hospitals with patient data ownership while patients delineate access policies. In our architecture, no central authority is intended to manage the data except the data owners, who, in healthcare processes, would be the patients. Pournaghi et al. [23] propose a framework named MedSBA that leverages blockchain technology and Attribute-Based Encryption. The distinction in their architecture lies in using two private blockchains. Instead, we consider only a public blockchain scenario.

6 Conclusion and future developments

In this paper we presented CAKE, a tool integrating public blockchain platforms, Attribute-Based Encryption (ABE) and the InterPlanetary File System (IPFS) for controlled data access within multi-party processes. IPFS serves as a tamper-proof repository for storing information artifacts, access policies, and actor metadata. Smart contracts manage user attributes, determine access permissions for process participants, and establish connections to IPFS files for notarization. Thereby, CAKE offers the ability to define precise specifications of access privileges, while ensuring data integrity, immutability, non-repudiation, and ultimately facilitating auditability. The maturity and integration of the tool is testified by its adoption in the context of a real-world cybersecurity project (BRIE)⁴ in the area of international logistics. Testing the adoption of our tool in further industry settings, thereby gathering feedback, extracting practical implications and devising theory from on-field experience [10,29] represents a future, highly interesting endeavor. Nevertheless, our solution exhibits limitations that we aim to address in future work, too. To begin with, whenever a data owner wishes to withdraw access to data from a specific reader, the only possibility is modifying the policy and re-encrypting the messages. However, the data previously uploaded on IPFS remains accessible. We are considering InterPlanetary Name System (IPNS) to overcome this limitation. Recently, we introduced an alternative approach to blockchain-based secure data sharing in cooperative settings, which divides the tasks of the attribute certification and key forging among multiple computing nodes in a distributed fashion [19]. The full distribution of computing (and the additional overhead it entails) was deemed as unnecessary in the BRIE setting among the stakeholders, due to the involvement of authoritative bodies for the attribution of user metadata and keys in the project. It is in our plans to implement the latter solution on multiple blockchain platforms and reach a level of maturity that is akin to CAKE in order to conduct comparative analyses on the applicability and trade-offs of the two approaches. Also, we plan to incorporate oracles in our solution to enable smart contracts' validation of off-chain data [1,22]. This integration empowers system designers to set the balance between complete transparency in the decision-making process and access control. Achieving this equilibrium entails strategically managing the storage of data both on-chain and off-chain, as discussed in [11]. Finally, combining our solution with techniques for process analytics based on blockchain data [21,13,12] paves the path for future research avenues.

Acknowledgements. The work of E. Marangone and C. Di Ciccio was funded by projects PINPOINT (B87G22000450001), under the PRIN MUR program, and BRIE (Cyber 4.0).

References

1. Basile, D., Goretti, V., Di Ciccio, C., Kirrane, S.: Enhancing blockchain-based processes with decentralized oracles. In: BPM Blockchain and RPA Forum. pp. 102–118 (2021)
2. Benhamouda, F., Gentry, C., Gorbunov, S., Halevi, S., Krawczyk, H., Lin, C., Rabin, T., Reyzin, L.: Can a public blockchain keep a secret? In: TCC (2020)
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: SP. pp. 321–334 (2007)
4. Chase, M.: Multi-authority attribute based encryption. In: TCC. pp. 515–534 (2007)
5. Chen, J., Micali, S.: Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci.* **777**, 155–183 (2019)
6. Corradini, F., Marcelletti, A., Morichetta, A., et al.: Engineering trustable and auditable choreography-based systems using blockchain. *ACM Trans. Manage. Inf. Syst.* **13**(3) (2022)

7. Dannen, C.: *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Apress (2017)
8. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: *Fundamentals of Business Process Management*, Second Edition. Springer (2018)
9. Feng, Q., He, D., Zeadally, S., Khan, M.K., Kumar, N.: A survey on privacy protection in blockchain system. *Journal of Network and Computer Applications* **126**, 45–58 (2019)
10. Ghaisas, S., Rose, P., Daneva, M., Sikkel, K., Wieringa, R.J.: Generalizing by similarity: Lessons learnt from industrial case studies. In: *CESI*. pp. 37–42 (2013)
11. Haarmann, S., Batoulis, K., Nikaj, A., Weske, M.: Executing collaborative decisions confidentially on blockchains. In: *BPM (Blockchain and CEE Forum)*. pp. 119–135 (2019)
12. Hobeck, R., Weber, I.: Towards object-centric process mining for blockchain applications. In: *BPM (Blockchain and RPA Forum)*. pp. 51–65 (2023)
13. Klinkmüller, C., Ponomarev, A., Tran, A.B., Weber, I., van der Aalst, W.M.P.: Mining blockchain processes: Extracting process mining data from blockchain applications. In: *BPM Blockchain and CEE forum*. pp. 71–86 (2019)
14. Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: *SP*. pp. 839–858 (2016)
15. Liu, Z., Jiang, Z.L., Wang, X., et al.: Multi-authority ciphertext policy attribute-based encryption scheme on ideal lattices. *ISPA/IUCC/BDCloud/SocialCom/SustainCom* pp. 1003–1008 (2018)
16. López-Pintado, O., Dumas, M., García-Bañuelos, L., Weber, I.: Controlled flexibility in blockchain-based collaborative business processes. *Inf. Syst.* **104**, 101622 (2022)
17. López-Pintado, O., García-Bañuelos, L., Dumas, M., et al.: Caterpillar: A business process execution engine on the Ethereum blockchain. *Softw., Pract. Exper.* **49**(7), 1162–1193 (2019)
18. Madsen, M.F., Gaub, M., Høgnason, T., et al.: Collaboration among adversaries: Distributed workflow execution on a blockchain. In: *FAB*. pp. 8–15 (2018)
19. Marangone, E., Di Ciccio, C., Friolo, D., Nemmi, E.N., Venturi, D., Weber, I.: MARTSIA: Enabling data confidentiality for blockchain-based process execution. In: *EDOC*. pp. 1–17 (2023)
20. Marangone, E., Di Ciccio, C., Weber, I.: Fine-grained data access control for collaborative process execution on blockchain. In: *BPM Blockchain and RPA Forum*. pp. 51–67 (2022)
21. Mühlberger, R., Bachhofner, S., Di Ciccio, C., other: Extracting event logs for process mining from data stored on the blockchain. In: *BPM Workshops*. pp. 690–703 (2019)
22. Mühlberger, R., Bachhofner, S., Ferrer, E.C., et al.: Foundational oracle patterns: Connecting blockchain to the off-chain world. In: *BPM 2020 Blockchain and RPA Forum*. pp. 35–51 (2020)
23. Pournaghi, S., Bayat, M., Farjami, Y.: MedSBA: A novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption. *JAIHC* **11** (11) 2020
24. Rahulamathavan, Y., Phan, R.C.W., Rajarajan, M., Misra, S., Kondo, A.: Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. In: *ANTS*. pp. 1–6 (2017)
25. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: *EUROCRYPT*. p. 457–473 (2005)
26. Stiehle, F., Weber, I.: Blockchain for business process enactment: A taxonomy and systematic literature review. In: *BPM Blockchain and RPA Forum*. pp. 5–20 (2022)
27. Wang, H., Song, Y.: Secure cloud-based ehr system using attribute-based cryptosystem and blockchain. *J. Med. Syst.* **42**(8), 152 (2018)
28. Weber, I., Xu, X., Riveret, R., Governatori, G., Ponomarev, A., Mendling, J.: Untrusted business process monitoring and execution using blockchain. In: *BPM*. pp. 329–347 (2016)
29. Wieringa, R., Daneva, M.: Six strategies for generalizing software engineering theories. *Science of Computer Programming* **101**, 136–152 (2015)
30. Wood, G.: *Ethereum: A secure decentralised generalised transaction ledger* (2014)
31. Xu, X., Weber, I., Staples, M.: *Architecture for Blockchain Applications*. Springer (2019)
32. Zhang, R., Xue, R., Liu, L.: Security and privacy on blockchain. *ACM Comput. Surv.* **52**(3) (2019)
33. Zheng, Z., Xie, S., Dai, H.N., Chen, W., Chen, X., Weng, J., Imran, M.: An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **105**, 475–491 (2020)

This document is a pre-print copy of the manuscript
([Marangone et al. 2024](#))
published by Springer (available at link.springer.com).

The final version of the paper is identified by DOI: [10.1007/978-3-031-61000-4_16](https://doi.org/10.1007/978-3-031-61000-4_16)

References

Marangone, Edoardo, Michele Spina, Claudio Di Ciccio, and Ingo Weber (2024). “CAKE: Sharing Slices of Confidential Data on Blockchain”. In: *CAiSE Forum*. Ed. by Shareeful Islam and Arnon Sturm. Vol. 520. Lecture Notes in Business Information Processing. Springer, pp. 138–147. ISBN: 978-3-031-60999-2. DOI: [10.1007/978-3-031-61000-4_16](https://doi.org/10.1007/978-3-031-61000-4_16).

BibTeX

```
@InProceedings{ Marangone.etal/CAiSEForum2024:CAKE,
  author      = {Marangone, Edoardo and Spina, Michele and Di Ciccio,
                Claudio and Weber, Ingo},
  booktitle   = {CAiSE Forum},
  title       = {{CAKE:} Sharing Slices of Confidential Data on
                Blockchain},
  year        = {2024},
  pages       = {138--147},
  crossref    = {CAiSE2024Forum},
  doi         = {10.1007/978-3-031-61000-4_16},
  keywords    = {Cyphertext Policy; Attribute-Based Encryption;
                Cryptography; Blockchain technology; Smart Contract}
}
@Proceedings{ CAiSE2024Forum,
  title       = {Intelligent Information Systems -- CAiSE Forum 2024,
                Limassol, Cyprus, June 3-7, 2024, Proceedings},
  year        = {2024},
  editor      = {Shareeful Islam and Arnon Sturm},
  isbn        = {978-3-031-60999-2},
  publisher   = {Springer},
  series      = {Lecture Notes in Business Information Processing},
  volume      = {520}
}
```