A Blockchain-based Solution for Enabling Log-Based resolution of Disputes in Multi-party Transactions

Leonardo Aniello, Roberto Baldoni, and Federico Lombardi

Research Center of Cyber Intelligence and Information Security
Department of Computer, Control, and Management Engineering "Antonio Ruberti"
"La Sapienza" University of Rome
{aniello,baldoni,lombardi}@dis.uniroma1.it

Abstract. We are witnessing an ongoing global trend towards the automation of almost any transaction through the employment of some Internet-based mean. Furthermore, the large spread of cloud computing and the massive emergence of the software as a service (Saas) paradigm have unveiled many opportunities to combine distinct services, provided by different parties, to establish higher level and more advanced services, that can be offered to end users and enterprises. Business-to-business (B2B) integration and third-party authorization (i.e., using standards like OAuth) are examples of processes requiring more parties to interact with each other to deliver some desired functionality.

These kinds of interactions mostly consist of transactions and are usually regulated by some agreement which defines the obligations that involved parties have to comply with. In case one of the parties claims a violation of some clause of such agreement, disputes can occur if the party accused of the infraction refuses to recognize its fault. Moreover, in case of auditing, for convenience reasons a party may deny to have taken part in a given transaction, or may forge historical records related to that transaction.

Solutions based on a trusted third party (TTP) have drawbacks: high overhead due to the involvement of an additional party, possible fees to pay for each transaction, and the risks stemming from having to blindly trust another party. If it were possible to only base on transaction logs to sort disputes out, then it would be feasible to get rid of any TTP and related shortcomings.

In this paper we propose SLAVE, a blockchain-based solution which does not require any TTP. Storing transactions in a public blockchain like Bitcoin's or Ethereum's provides strong guarantees on transactions' integrity, hence they can be actually used as proofs when controversies arise. The solution we propose defines how to embed transaction logs in a public blockchain, so that each involved party can verify the identity of the others while keeping confident the content of transactions.

Keywords: blockchain, log certification, trustworthiness, multi-party transactions

1 Introduction

As Internet-based services are evolving, companies need to integrate their IT infrastructures. Business-to-Business (B2B) integration aims to connect key business processes in an automated and optimized way, so as to deliver sustainable competitive advantage to customers and suppliers. A relevant example regards cloud federations, where multiple private/public IaaS providers share their own resources [2, 8, 10] to cope with load peaks without over-provisioning, by renting out resources otherwise unused. IaaS providers supply these resources temporarily, upon explicit requests by parties in need. Such integrations require multi-party transactions that need to be regulated through some Service Level Agreement (SLA) so that, in case one party claims an SLA violation, she can prove it. Indeed, each party may keep logs of sent requests and received responses, but the other party may ignore requests/responses or deny logs validity.

Current solutions employ a *trusted-third party* (TTP) [3, 7] which is in charge of checking SLA compliance and solve possible disputes. In this way, parties cannot drop or deny any sent request or received response, because the TTP is involved in and logs every interaction (see Fig. 1). The main drawbacks of TTP-based solutions are mainly related to: (i) performance overhead, as required interactions are routed through the TTP, which can be a single point of failure and a performance bottleneck; (ii) additional fees, as the TTP intermediation does not usually come for free and may ask for an initial fee or for per-transaction fees; (iii) the TTP must be trusted and if it behaves dishonestly or colludes with the other parties, there is no chance to prove the injustice.

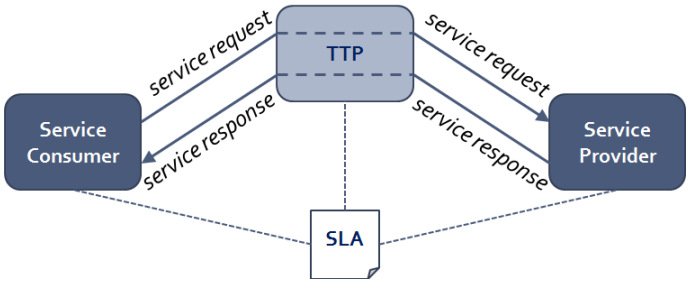

Fig. 1. TTP-based solution

In this paper we propose *SLAVE* (***S**ervice **L**evel **A**greement **VE**rified*), a solution to replace a non-totally trustworthy TTP with an intermediary based on a public blockchain like Bitcoin's [6] or Ethereum [11], such that data sent to a public blockchain cannot be falsified, hence no risk of dishonest behaviour or collusion. Since data in a public blockchain can be seen by everyone, pseudonyms and asymmetric cryptography is used to mask sensitive information.

Paper structure. § 2 introduces an overview of blockchain technology and its properties, § 3 presents the proposed solution, finally § 4 concludes discussing future work.

2 Background on Blockchain

The blockchain is a technology initially conceived to manage in a secure fashion the transactions of Bitcoin [6] in a trustless p2p network. It is a public ledger replicated among all nodes participating the network. It is composed by consecutive chained blocks, each one containing a set of transactions, a hash referencing the previous block, and a special number called *proof-of-work* (PoW), i.e. a number such that the hash of the entire block is lower than a *target* number. This target is tuned so that participating nodes will find a solution (i.e., the PoW) within a certain time with high probability. For Bitcoin's blockchain this time is 10 minutes, while for Ethereum's is about 15 seconds. Computing the PoW requires high computational power, and it is considered nearly impossible for a single node to find a solution for a block in a reasonable time [5]. Nodes responsible to collect transactions and creating the chain by computing the PoW are called *miners*. Miner's incentive consists in a reward for each mined block. Once a block has been solved (i.e. mined), the miner broadcasts it to the network. Each node controls the block validity before chaining it to the previous block. Forks are possible as multiple miners may mine a different block and propose them in the same time to the network. Usually forks are solved during time by employing the rule of always accepting the longest chain, hence after some mined blocks the network will converge to a unique chain. The blockchain is indeed considered an eventual consistent database. Branches cannot be precomputed off-line as mining each block needs the hash of the previous one. This gives to public blockchains strong data integrity guarantees. Indeed, an attacker willing to tamper with data stored in the blockchain should have the majority of the computational power of the entire network. Indeed, to forge a value in a block she should compute again the PoW of every following blocks faster than the rest of the network, so as to propose a longer chain. Assuming a majority of hash power controlled by honest miners, the probability of a fork of depth n is $\mathcal{O}(2^{-n})$ [1]. This gives users high confidence that simply waiting for a small number of nodes to be added (e.g., 6 blocks in Bitcoin) will ensure their transactions become tamper-proof. Because of its decentralisation and data integrity properties, blockchain has been investigated for different purposes, e.g., for *smart contracts* with Ethereum [11], as an alternative to typical Remote Data Auditing solutions [9], and to ensure integrity of cloud storage [4].

3 Proposed Solution

In this section we present SLAVE, a solution to enable log-based resolution of disputes in multi-party transactions. SLAVE employs a public blockchain to store requests/responses. Both provider and consumer participate in the mining

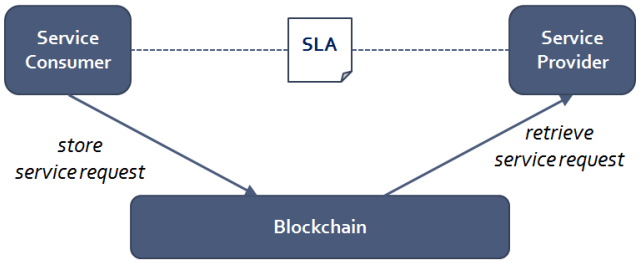

Fig. 2. Interaction between a service consumer and a service provider in SLAVE. Requests and responses are stored in the blockchain, they are the logs to be used for dispute resolution

process to detect requests and responses directed to them (see Fig. 2). Storing requests and responses in a public blockchain provides strong integrity guarantees, thus they cam be then used in case of disputes. As data in a public blockchain can be accessed by everyone, there is the need to mask sensitive information, which in this case are the identities of involved parties and the content of transactions.

Identities are masked through the usage of pseudonyms. Each party has as many disjoint sets of pseudonyms as the parties it has to interact with, so that each pseudonym is used only to interact with a specific party, which is the only party to know the real identity behind such pseudonym. Each pseudonym is a public key, and the corresponding private key is kept secret by the party itself. We use the notation pk and sk to indicate public and private (i.e., secret) keys, respectively, and the notation $\{m\}_k$ to indicate the encryption of m with a key k. For each pair of parties A and B that want to interact through SLAVE, a preliminary handshake phase is required, where A generates a set $\{\langle pk_i^{A,B}, sk_i^{A,B}\rangle\}$ of public/private key pairs to communicate with B, and sends the set $\{pk_i^{A,B}\}$ of generated public keys (i.e., the pseudonyms) to B through a secure channel. Vice versa, B generates a set $\{\langle pk_i^{B,A}, sk_i^{B,A}\rangle\}$ of public/private key pairs to communicate with A, and sends the set $\{pk_i^{B,A}\}$ of generated public keys (i.e., the pseudonyms) to A through a secure channel.

Once the handshake phase is completed, A and B can start exchanging transactions using the SLAVE solution. Let T be a transaction from A to B. Let N_T be a nonce computed by A for T to prevent replay attacks. Let $sign(m, sk)$ be the signature computed on (a digest of) message m using the private key sk, used in this case by A to prove the authenticity of its transaction T. The information to be stored in the blockchain also have to include what pseudonyms $pk_i^{A,B}$ and $pk_j^{B,A}$ have been used by A. The former is put in encrypted form, while the latter is kept in clear to let B recognising that the transaction is directed to her and understanding what private key to use to decipher all the data of the transaction. Overall, the tuple to be stored in the blockchain has the following format: $\langle \{\langle T, N_T\rangle\}_{pk_i^{A,B}}, \quad sign(\langle T, N_T\rangle, sk_i^{A,B}), \quad \{pk_i^{A,B}\}_{pk_i^{B,A}}, \quad pk_i^{B,A}\rangle$.

4 Conclusion

In this paper we propose SLAVE, a solution to enable log-based resolution of disputes in multi-party transactions, which replaces the usage of a TTP with a pubic blockchain. SLAVE allows to overcome the limitations of possible malicious behaviours of a TTP, including the risk of collusion with other parties. SLAVE also improves service availability with respect to TTP-based solutions, as thousands of miners supports the blockchain functioning. As the blockchain provides high latency, the performance bottleneck is still a problem and a possible solution to investigate can be to batch messages to increase the throughput or adopt different architectural solution, as proposed in [4]. As an interesting future, we plan to investigate the real fees of adopting such a blockchain-based solution, and compare these costs to those of current TTP-based settings.

Acknowledgment

This work has been supported by the European Commission's H2020 Programme under the SUNFISH project, grant N.644666.

References

1. J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies. In *IEEE Symposium on Security and Privacy*, 2015.
2. ENISA. Security Framework for Governmental Clouds, 2015.
3. A. M. Froomkin. The essential role of trusted third parties in electronic commerce. *Or. L. Rev.*, 75:49, 1996.
4. E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone. Blockchain-based database to ensure data integrity in cloud computing environments. In *Proceedings of the 1st Italian Conference on Cybersecurity*, 2017.
5. J. Garay, A. Kiayias, and N. Leonardos. *The Bitcoin Backbone Protocol: Analysis and Applications*. Springer Berlin Heidelberg, 2015.
6. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
7. J. W. Palmer, J. P. Bailey, and S. Faraj. The role of intermediaries in the development of trust on the www: The use and prominence of trusted third parties and privacy statements. *Journal of Computer-Mediated Communication*, 5(3), 2000.
8. F. P. Schiavo, V. Sassone, L. Nicoletti, and A. Margheri (Eds.). FaaS: Federation-as-a-Service, 2016. Available at `https://arxiv.org/abs/1612.03937`.
9. M. Sookhak, A. Gani, H. Talebian, A. Akhunzada, S. U. Khan, R. Buyya, and A. Y. Zomaya. Remote Data Auditing in Cloud Computing Environments: A Survey, Taxonomy, and Open Issues. *ACM Comput. Surv.*, 47(4), May 2015.
10. B. Suzic, B. Prünster, D. Ziegler, A. Marsalek, and A. Reiter. Balancing Utility and Security: Securing Cloud Federations of Public Entities. In *C&TC*, volume 10033 of *LNCS*, pages 943–961. Springer, 2016.
11. G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014.