A Novel Reputation System for Mobile App Stores Using Blockchain

Gonçalo Sousa Mendes
Aptoide, S.A.

Daniel Chen
Aptoide, S.A.

Bruno M. C. Silva
Instituto de Telecomunicações,Universidade da Beira Interior; IADE, Universidade Europeia

Carlos Serrão
ISTAR - Information Sciences and Technologies and Architecture Research Center; ISCTE - Instituto Universitário de Lisboa

João Casal
Aptoide, S.A.

***Abstract*—With thousands of mobile applications submitted to online application stores the mobile application market has experienced a significant growth. This growth is however accompanied by an increase in malware presence which is detected after infecting users or when a it is reported to the store. A possible solution would be to leverage those reports, across all mobile ecosystem, creating a shared reputation system, to provide more accurate feedback to the app stores quality assurance and security teams. To support this sharable reputation management system, we present a scalable blockchain-based solution, that provides the necessary scalability, data privacy and trust requirements, while being cost-effective. This paper also presents a real case study and respective results on performance, scalability, and cost evaluation. case study and respective results on performance, scalability, and cost evaluation.**

MOBILE APPLICATIONS economy is an expanding market. In August 2019, Google Play Store had almost 2.5 million apps in its repositories, Apple App Store had 1.8 million and Aptoide with nearly 1 million, with 21.3 billion downloads in Google Play alone [1], with several other third-party app stores experiencing similar rates [2]. However, malware presence has also increased [3], which exposes users to possible privacy breaches and loss of assets (*e.g.*, baking

Trojans to steal login credentials [3]) as app stores are struggling to keep up with such malware growth [2].

Current centralized models of cybersecurity verification and validation hardly fit the ever-growing global mobile app economy size. Several apps have obfuscated code, that hardens automated code analysis. Studying their behavior during execution is not feasible due to its prohibitive costs and high inefficiency. This shows how a centralized approach to cybersecurity may not be the only answer to fight mobile malware, where each app store has its own methods and do not share the discovered threats.

A new approach to this problem is to relate the multiple actions of millions of users and thousands of developers (**reputation events**), such as ratings, comments, or app uploads, to a degree of trust, and use that found reputation in the cybersecurity processes, to provide trustworthy and relevant decision support to the quality assurance [QA] and security teams. Moreover, such approach could benefit both the developers, as they could switch between different markets while keeping their reputation, and consumers, as it makes it easier for them to trust a new developer, which significantly increases the content quality of all app stores, having a direct impact in their revenue, a practice widely adopted on e-commerce platforms, such as Amazon or e-Bay, among several others [4], [5].

Building a consortium solution for several competing app stores presents several problems of performance, scalability, interoperability, data security and trust between entities [6], [7], [8], [9]) trust between the app stores inside the consortium; 2) scalability, interoperability and performance to manage multiple actions of millions of users and thousands of developers, where several different app stores can contribute, and each one has its own business model and rules; 3) data privacy, as the user's data needs to be securely saved to comply with data regulations; 4) traceability, as it is necessary to know the history of actions of the users; and 5) cost efficiency.

Blockchain is a distributed ledger, in a peer-to-peer network, with unique characteristics that can effectively solve security problems [6]. The new blocks are validated through a consensus protocol, that also guarantees a total order of them. Blockchains can be public, private, or a mixture of both. In public blockchains, everyone can join the network and participate. Every node can mine the new block and rewarded by the issuers of transactions inside that block. On private blockchains, only one entity controls the network and dictates who can join the network, read, and write blocks. In this network, it is not necessary to pay mining fees. In consortium blockchains, the control of the network is shared among all participant entities, where everyone can read and write, but joining requires the approval of all members. As in private blockchains, there are no mining fees.

A possible solution to tackle the mentioned problems consists in using a blockchain, integrated with a cloud system, to support the blockchain with the necessary logic operations and provide an interface for the QA and security teams.

Aptoide, already considered one of the safest Android app store [2] has nonetheless felt the need to improve its security by developing such system. Here, millions of users, through the app stores installed on their smartphones, perform **reputation events** that are disseminated through the blockchain's services. The app stores can then infer threats on its applications through those **reputations' events**, where the threats are removed, creating a mobile Cyber-Physical System (CPS) connecting millions of devices, with blockchain and cloud/interface services.

This paper presents a cloud-based solution to manage reputation events, integrated with a consortium blockchain, to serve as a distributed ledger to store those events, as the support for the reputation system. This combination allows for the identification of compromised apps, through a dedicated interface for the analysts, where the apps classified as threat by high reputation users are displayed first for review, providing and improvement on the quality assurance to users and mitigating the potential for privacy breaches and data thefts, for all users of the app stores that wish to be in the consortium. This possibility generates a global app economy, with increase security, trust and transparency, where each app store can control how the reputations are calculated, allowing for a better fit on their business model.

The main contributions of this work are:

1. Study and implementation of a cloud solution integrated with a blockchain to manage and store millions of reputations events;

2. Implementation of a real case study to evaluate the performance of the solution developed.

From the obtained evaluation results, it was observed how the combination of a cloud solution with a blockchain could improve this type of system regarding data security, interoperability, and scalability, while being cost-effective.

The remainder of this paper is organized as follows. Section 2 identifies some related work and background to the system proposed in this article. In Section 3 we present the system implementation. Section 4 offers a real case study setup to demonstrate the solution behavior. Finally, Section 5 presents the conclusions and some suggestions for future work.

2. Related Work and Background

Some works have already integrated blockchain in their reputation systems to overcome the problems mentioned above [10], [8]. Some examples follow.

Rep on the Roll [11] is a generalist reputation system for peer-to-peer networks, where blockchain is used for effective communication and information sharing between clients, and whenever behavior traceability is desired. Both trust and reputation of peers are maintained unmodified due to the proprieties of blockchain. The authors compared their work with eBay, that can process average of 23.184 transactions per second, against the modest quantity of 10 in their work. Such a low number comes from the time it takes to mine a block in a bitcoin-based blockchain. This is a performance problem, as well as an opportunity for a denial of service attack, as the authors noted. This performance issue also makes the network hard to adopt and implement in larger scales, since the resources required from each node are costly. The network is also public, which raises some data privacy concerns, as all user's data is accessible on a public network.

SH-BlockCC [8] proposes a cloud solution based on blockchain to increase the security and trust of IoT in smart homes while maintaining availability, scalability, and traceability. The authors demonstrated how their proposed architecture could make IoT more secure and efficient. However, their blockchain is too tailored to their specific problem, without providing clear implementation details, making it very difficult to translate it to other domains.

Ink Protocol [12] is a decentralized third-party software to handle a payment system for e-commerce marketplaces. The goal is to increase trust, and therefore security, by evaluating transactions between the sellers and buyers to create reputations scores. This is very close to our goal. However, they require cryptocurrency, so it is based on Ethereum, a public blockchain. As mentioned, this type of blockchain brings some performance and data privacy issues, as well as the necessary mining fees.

Decentralized Science [13] is a blockchain-based distributed platform for science publications, together with a reputation system of peer reviewers. The platform is used to store the peer review process communications, where the smart contracts validate the system's rules. This system is also based on Ethereum.

Some several other implementations and areas use the blockchain for storage and reputation management, such as, emission trading scheme [14], with a private blockchain, where new participants must pay to adhere; auto-mobile industry [15], where a public blockchain is used, with an external service to provide the necessary security and data privacy; cyberphysical systems [16], a bitcoin-based blockchain, with extra work to offer the required CPS systems data privacy, among others.

None of the presented works can provide a truly scalable solution while ensuring data privacy. A possible solution is to use consortium blockchains, a blockchain that is controlled by a set of trusted entities, not entirely public or entirely private. This approach brings several benefits to the collaborations between organizations besides security, namely scalability (it is easy to add new nodes and other app stores) and data privacy (only authorized app stores can access the data) [17], [7], while it still cost-effective [7], [8], [18], as since there is no need to pay the mining fees, as it is for public blockchains and the maintenance costs can be equally shared between all participant entities. The consortium approach

also adds the necessary trust between entities in the app market, since no one fully controls the network, or can temper the data unnoticeable.

Some works already demonstrate the usefulness of such approach [4], where the authors used a consortium blockchain for asset trading, combined with a common database. Here, the reputation is computed inside the network, on a smart contract. This makes it impossible to adjust to the specific needs of each new entity, a desirable property for the app market ecosystem. Also, the use of a common database demonstrates the need for a balance between a cloud and blockchain.

To implement a consortium blockchain, we selected the **Hyperledger Fabric** [19] [HLF], a blockchain framework under the umbrella of the Hyperledger Greenhouse, hosted by The Linux Foundation. It was chosen due to its strong modularity, widespread adoption, popularity [17], and its proven scalability properties [7], besides providing all the mentioned requirements [19].

It has three distinct phases: **the execute phase**, that starts when a client submits a new transaction to the endorsement peers, that run the chaincode (equivalent to smart-contracts) creating the read-write sets [RW]. Defined by the endorsement policy, these peers are the only ones that can simulate new transactions. This set is then returned to the client, that aggregates the necessary RW and sends them to the ordering service, starting the **order phase**. Here, through a consensus protocol a new block is created, defining the total order of transactions, and sent to all peers. In the **validate phase**, the peers verify the latest transactions and update their world state. This world state is a snapshot of the most recent state of the blockchain.

To improve data security and compliance with data regulations, HLF provides data privacy. In this mechanism, data is written in a separated database, only accessible to the authorized peers, and only the hash of transaction is saved on the blocks. This allows deleting personal and sensitive data, as only the hash is permanently kept. Fabric also contains the Membership Service Provider [MSP], a service responsible for defining the roles of the entities inside the network and validating the certificates generated by the Fabric Certificate Authority, that is abstracted through this service.

It is through those services that Fabric provides the necessary data privacy and security to the blockchain solution.

3. Scalable blockchain-based solution for reputation management

The proposed system architecture is composed of two major components: the blockchain, where all the **reputation events** are stored; and the cloud infrastructure, which is responsible for reputation management and app cybersecurity processes. The cloud infrastructure supports an interface that provides the QA and security teams with the necessary dashboards for their analysis, creating an intelligent decision support system.

This cloud is only a complementary module, provided as Software as a Service (SaaS), since the blockchain is enough to store the **reputation events**. However, it is through the cloud that It is possible to deliver a dedicated interface to the QA and security teams to analyze the apps considered dangerous by the reports provided by the users, thus improving the cybersecurity processes of app stores (*e.g.*, more efficiency in reviewing apps).

The cloud is responsible for: 1) gathering events from Aptoide app store (in our case), process them, (to create the **reputation events**) and send them to the blockchain; 2) collecting the events from the blockchain, from all app stores, and compute the user's reputations and app threat levels; 3) orchestrate the two modules and produce the necessary data for the interface, namely what apps have the higher priority in being analyzed, considering the reputations of users and their feedback.

The task division between the cloud and blockchain is crucial, as not only it determines the efficiency of our solution, but also its accessibility to other app stores. The blockchain saves the **reputation events**, and not the reputation itself, increasing the interoperability, which allows for each store to model those following their business model. By proving an external API, this interoperability is improved, as it dismisses the complicated knowledge to communicate directly with the blockchain. This API has four operations: 1) insert event: to add new **reputation events**; 2) get event: get a specific event; 3) get by date: fetch all events that were added to the blockchain on

a particular day; and 4) get by user: returns all events from a user.

To guarantee that the blockchain is app store agnostic, a reputation event as generic as possible was defined, reducing the complexity of the smart contract (chaincode in our blockchain implementation), making it easier for any app store to send and receive **reputation events**: **event_id**: The Id of the event, generated by each app store; **store_name**; Field that contains the app store name; **user_email**: The email of the user; **user_type**: Type of the user - developer or consumer; **action**: Action performed by the user; value_of_action: the value of the action itself (e.g., the flag given); **destination_email**: Who receives the action; **added_on**: Timestamp from when the event was added to the blockchain (in coordinated universal time), added upon insertion in blockchain; occurred_on: Timestamp from when the event occurred in the app store. To differentiate between different app stores, each app store in the consortium will have a unique Id to place at the beginning of the Id field.

The default HLF Certificate Authority was used to generate X509 certificates. As for the consensus algorithm, we used Raft. This protocol works in a leader-follower fashion and is crash fault-tolerant (CFT) and can be distributed in different servers and organizations, decentralizing the control of the consensus algorithm. Raft is the first fully consensus protocol integrated in Fabric (implemented in V1.4.1, April, 2019), therefore providing fast integration and a higher performance, when compared to its ancestor, a junction of Kafka and ZooKeeper.

All Fabric components run in containers and communicate through Remote Procedure Calls (RPC) over a secure and authenticated channel using Transport Layer Security (TLS) protocol. The chaincode was written in Go, and our endorsement policy stated that at least one peer of each organization (randomly chosen by Fabric) has to endorse the transaction.

The version of the HLF used is the 1.4.1. For the implementation and our proof-of-concept, two organizations were used for deployment, the real app store, Aptoide, and a dummy app store, named app store X, to simulate the dissemination of information in the blockchain.

3.1. Cloud Implementation The proposed cloud solution, integrated with the blockchain, serves the propose of supporting the above-mentioned intelligent decision support system. It is microservice oriented, built on top of two main modules: Event Tracker and Data Collection (**Figure 1**).

The Event Tracker is responsible for fetching the events from the app store, analyze each one individually and re-routing them to the blockchain (left side of Figure 1). If the event is a flag (Aptoide allows users to flag its app), it is stored in a local cache, waiting for the analysis of the QA reviewer to assess its truthfulness. That review creates a new event (QA review) that is used to evaluate all flags given to that app to determine their accuracy, propagating that information to the blockchain, increasing the reliability, since only truthful events are stored in the blockchain, which later reduces the logic needed in computing the user's reputation. If the event is none of the above, it is adequately treated and sent to the blockchain. To increase the performance of cybersecurity processes and notify the QA team of apps that require immediate action, the flags given by high reputation users and uploads by low rated developers, are also treated separately, to update the app threat level and the list of apps submitted by low rated developers, respectively. This app threat level represents the severity deemed by the system for each app, increasing or decreasing its urgency in being reviewed, making the system reacting faster to what may be the biggest threats to the end-users.

The Data Collection (right side of Figure 1) is responsible for collecting the events from the blockchain, including from other stores' as well. During this phase, the **reputation events** are processed, through an internal defined reputation system, and the reputation of the users involved is updated. If the user is new on Aptoide, we check to see if he has already done other actions on other app stores, updating his reputation accordingly. Then, the system updates the threat level for each app with the updated reputations.

Each module is executed once per day by a scheduler, being that the Event Tracker always starts first, at the beginning of the day (after mid-night in UTC) by fetching the events from the previous day and sending them to the

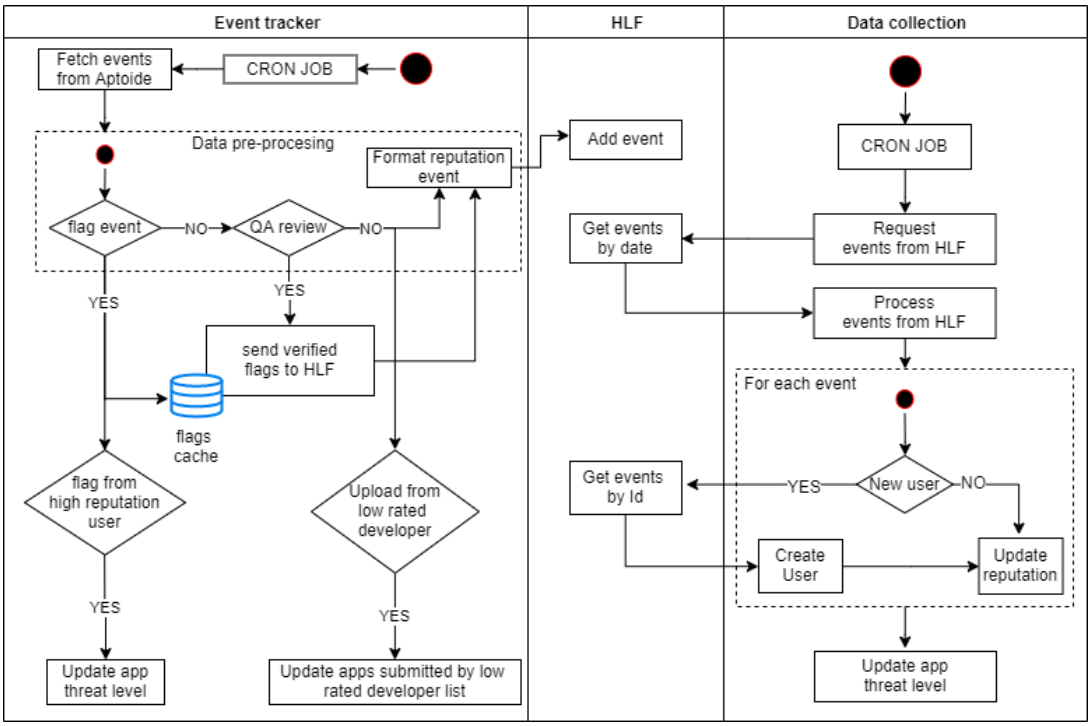

Figure 1. Flow chart of the implemented cybersecurity process. It depicts the flow of the two main off-chain modules and their interactions with the blockchain. It also demonstrates how the reputations are used to update the app threat levels.

blockchain through the *add_event* functionality. However, if deemed necessary, by an increase in flag actions, for example, it can be executed more frequently, further decreasing the reaction time. The Data Collection then fetches all events, from all app stores, from the previous day using the *get_event_by_date*. This query is done on the date from which the events were added to the blockchain to avoid missing events added by other app stores, that occurred before the previous day, but were all added on that last day.

4. Evaluation Results

The evaluation consisted of several tests. First on the HLF configuration, to benchmark and test the performance, scalability, and fault tolerance. Then, off-chain solution, to evaluate the capacity of the system in managing several thousands of events. All the tests were performed on a Dell R210-2 remote server, with an Intel Quad-Core Xeon E3-1270v2, 16Gb DDR3 of RAM and 2Tb SATA2 for the hard drive, running Ubuntu 16.04 LTS.

4.1. Hyperledger Fabric Evaluation

Three different steps in the evaluation [18], [20] were defined: first, on the performance, by measuring the impact of changing, first the block size, then the number of transactions on each block; second, on the scalability, by adding more peers to the organizations; third, on fault tolerance. For each test, two metrics are presented, the latency (time of each operation to complete) and throughput (transactions per second). Moreover, the tests do not include the authentication of users, one of the most cumbersome steps, as that is processed outside of HLF itself. The experiments were performed by Hyperledger Caliper, a benchmark tool for hyperledger blockchains.

In this scenario three functionalities were tested: a write operation (add_event), a read operation (get_by_id) and a read operation that returns several events (get_by_date). For each event, 1000 different transactions were sent, at a rate of 100 per second. The fields in the events were generated randomly, aside from the Id, that was incremental, starting at 1, and the date, that was chosen randomly for a set of 10 different

dates. The maximum number of transactions to be sent per second was 100. The initial configuration contained two peers in each organization and three raft nodes. Figure 2 contains the results of the tests, where each plot contains the results for all three operations.

Aside from the blocksize, we used the initial configuration provided by the HLF. Figure 2A and 2B. Both the latency and TPS are somewhat stable through the trials, with a small peak in TPS for the add_event operation on 2Mb. This cap is explained by the limitation on the number of transactions. Therefore, we increasingly changed this limitation to observe its behavior. From Figure 2C and 2D it is observable how incrementing the number of transactions improves the performance. Latency wise, it is not observable further increases after 75 transactions per block. As for the TPS, it reaches its peak at 100 transactions per block, with approximately 70 TPS, for write operations. It is here that the network reaches its saturation and its no capable of processing more transactions.

The scalability of the system was tested by iteratively, adding peers to each organization. The initial configuration is the one with the best results (Figure 2E and 2F). Those results are not expected, as having more peers, means more availability. However, an observation on the resource consumption explains this performance, as the tests are run in a single server, that rapidly reaches its limit, and the Caliper must wait for all the peers to commit the new block

Finally, the system was tested regarding its fault tolerance (Figure 2G and 2H). From the results, it is possible to conclude that adding fault tolerance does not create performance issues. This is expected, as the only communication is done to the leader, that simultaneously propagates the message and waits for a majority of responses. Therefore, as long the number of nodes does not consume all server resources, it is not expected to observe changes in the times reported. For the following tests, the number of raft nodes was set to 3.

The configuration found is useful for our setup but may not be the best for others. However, these tests show how versatile and scalable the HLF is, where it is possible to add or remove peers, in any location, due to its modularity.

The configuration found is useful for our setup but may not be the best for others. However, these tests show how versatile and scalable the HLF is, where it is possible to add or remove peers, in any location, due to its modularity.

4.2. Cloud evaluation

After finding the best configuration for HLF, the whole system was tested, by generating 10,000 **reputation events** (a number way above the average daily in Aptoide), for 1000 users (that were previously inserted in the database), corresponding to a fictional day. Using the above-presented flow shown in Figure 1, the results presented here will be of: pre-processing of data, sending events to the blockchain, requesting those events and updating the reputations and apps threat level. Note that, despite all component being in the same network, the requests are not done in localhost, as to approximate the results with a real-world scenario, without adding too much latency.

Pre-processing data and sending it to the blockchain is continuous, however, run times are presented separately to facilitate the analysis of each module. The full process, for the 10000 events, took approximately 11 hours. The initial flow of the cloud takes the most time, having one hour in pre-processing and almost 10 in sending them to the blockchain. The process of preparing and analyzing the data is extremely fast, demonstrating its ability to quickly react to new threats while preparing the events. Sending the events to the blockchain is where the system spends most of its time, averaging 3 seconds per event, which includes the latency of the requests, authentication, done for every new request, HLF internal flow and the response, all of which becomes the bottleneck of our system. Given the added layer provided by the blockchain, with the extra steps of security and replication, and the fact that our implementation is a proof-of-concept that does not consider parallelization strategies, these times are not a surprise and confirms that the system can be improved to leverage the capabilities of HLF and deliver higher performances.

To request the 10,000 events from the blockchain, it only took 3 seconds, which is a fast response, considering the number of events and the times to send them. Then, it took almost 18

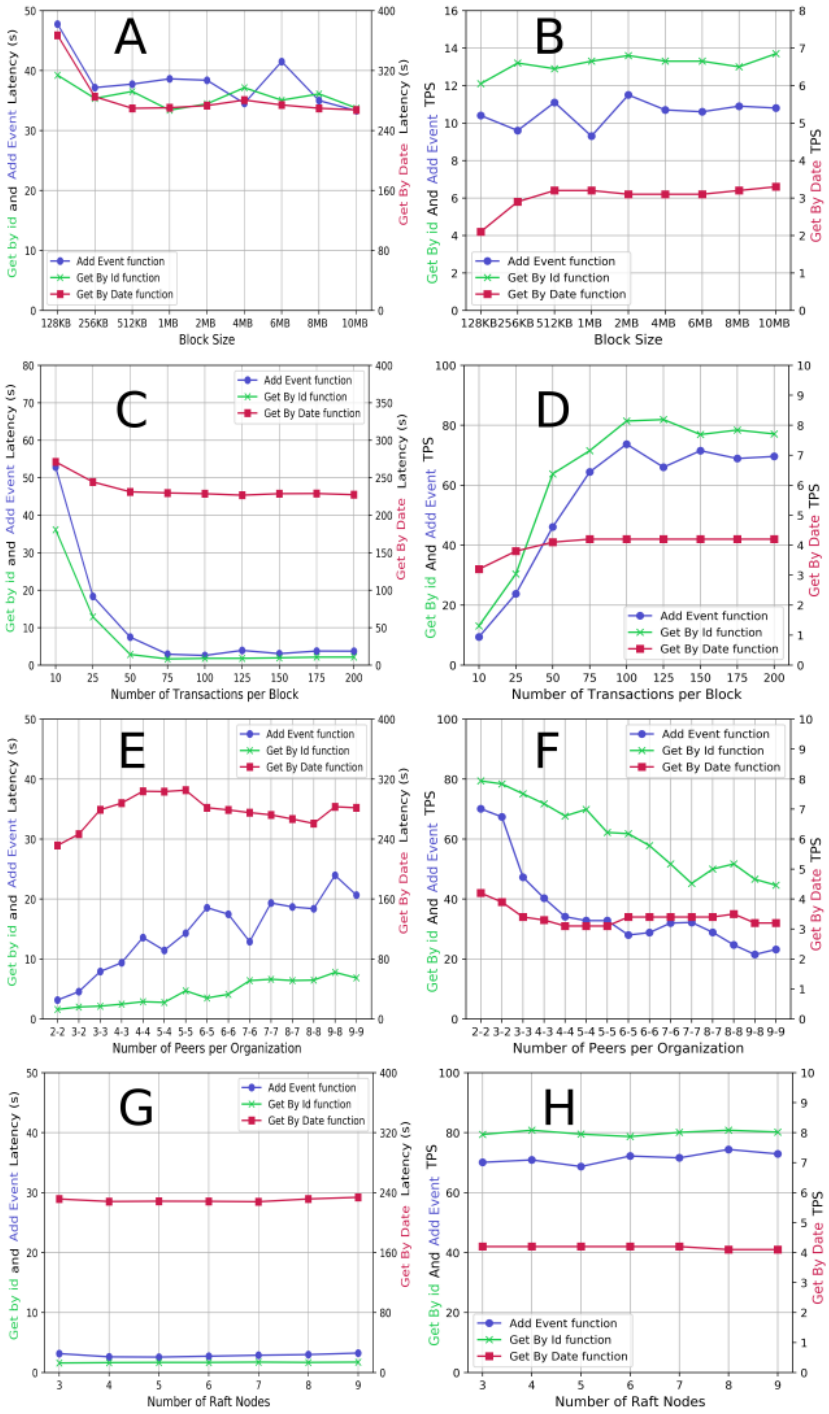

Figure 2. Tests on the HLF configuration, with the Throughput and Latency, for each analysis performed. A/B results while varying the block size, 2Mb was chosen as a safe threshold; C/D: results while changing the number of messages per block. 100 messages were selected. E/F: results on the variation on the number of peers. The initial configuration has the best performance. G/H: results on the variation on the number of raft nodes. No changes on the performance were detected.

minutes to process all those events and update the reputations and apps threat level, showing how fast the off-chain solution can collect the events and process them, reacting fast to changes in the users' reputations, reflecting those same changes in the cyber security processes, as the app's threat level.

5. CONCLUSION

This paper presents a blockchain-based solution to manage **reputation events**, through Hyperledger Fabric, named Trustchain. This system provides the necessary data privacy and security, but also guarantees scalability, without a significant loss of performance, while being cost-effective. This work includes an experimental analysis and evaluation on the proposed system using a real app store, Aptoide, together with an explanation of how the solution was constructed and implemented to achieve fast response times regarding mobile cybersecurity. It was demonstrated how a modular blockchain, through Fabric, is adaptable and can easily scale. Both results validate the solution, attesting that it does execute as intended in reacting to new threats and sharing, in a secure manner, the events that lead to those reactions. However, it does need some parallelization strategies to achieve its full potential better. Future work includes scaling, with more parallelization strategies and spread the peers through different servers and test that implementation, using the baseline provided here, to achieve the necessary performance seen in the market, with millions of apps and users

ACKNOWLEDGEMENT

This work is part of the TrustChain project, co-funded by Programa Operacional Regional de Lisboa (Portugal 2020 / EU), in the context of the Portuguese Sistema de Incentivos à I&DT Empresarial (project ID LISBOA-01-0247-FEDER-038315). This work is funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/EEA/50008/2020.

References

1. J. Clement. Mobile app usage - statistics & facts. https://www.statista.com/topics/1002/mobile-app-usage/, August 2019. Accessed on: 23DEC2019.
2. Yuta Ishii, Takuya Watanabe, Fumihiro Kanei, Yuta Takata, Eitaro Shioji, Mitsuaki Akiyama, Takeshi Yagi, Bo Sun, and Tatsuya Mori. Understanding the security management of global third-party android marketplaces. In *Proceedings of the 2nd ACM SIGSOFT International Workshop on App Market Analytics - WAMA 2017*. ACM Press, 2017.
3. McAfee. Mcafee mobile threat report q1, 2019. Technical report, 2019.
4. Haihui Huang, Jing Cai, and Shaoci Xie. Implementing an asset trading system based on blockchain and game theory. In *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, oct 2019.
5. Mareike Möhlmann, Timm Teubner, and Antje Graul. Leveraging trust on sharing economy platforms: reputation systems, blockchain technology and cryptocurrencies. In *Handbook of the Sharing Economy*, pages 290–302. Edward Elgar Publishing, 2019.
6. Jian Chen, Zhihan Lv, and Houbing Song. Design of personnel big data management system based on blockchain. *Future Generation Computer Systems*, 101:1122–1129, dec 2019.
7. Elli Androulaki, Christian Cachin, Angelo De Caro, and Eleftherios Kokoris-Kogias. Channels: Horizontal scaling and confidentiality on permissioned blockchains. In *Computer Security*, pages 111–131. Springer International Publishing, 2018.
8. Saurabh Singh, In-Ho Ra, Weizhi Meng, Maninder Kaur, and Gi H. Cho. SH-BlockCC: A secure and efficient internet of things smart home architecture based on cloud computing and blockchain technology. *International Journal of Distributed Sensor Networks*, 15(4):155014771984415, apr 2019.
9. Elena Karafiloski and Anastas Mishev. Blockchain solutions for big data challenges: A literature review. In *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*. IEEE, jul 2017.
10. Huaqun Wang, Qihua Wang, and Debiao He. Blockchain-based private provable data possession. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2019.
11. Richard Dennis and Gareth Owenson. Rep on the roll: A peer to peer reputation system based on a rolling blockchain. *International Journal for Digital Society*, 7(1), mar 2016.
12. Ink. Decentralized reputation and payments for peer-to-peer marketplaces. Technical report, 2018.
13. Antonio Tenorio-Fornés, Viktor Jacynycz, David Llop-Vila, Antonio Sánchez-Ruiz, and Samer Hassan. To-

wards a decentralized process for scientific publication and peer review using blockchain and IPFS. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*. Hawaii International Conference on System Sciences, 2019.

14. Khamila Nurul Khaqqi, Janusz J. Sikorski, Kunn Hadinoto, and Markus Kraft. Incorporating seller/buyer reputation-based system in blockchain-enabled emission trading application. *Applied Energy*, 209:8–19, jan 2018.
15. Zhaojun Lu, Qian Wang, Gang Qu, and Zhenglin Liu. BARS: A blockchain-based anonymous reputation system for trust management in VANETs. In *2018 17th IEEE TrustCom/ 12th IEEE BigDataSE*. IEEE, aug 2018.
16. Yanqi Zhao, Yannan Li, Qilin Mu, Bo Yang, and Yong Yu. Secure pub-sub: Blockchain-based fair payment with reputation for reliable cyber physical systems. *IEEE Access*, 6:12295–12303, 2018.
17. Omar Dib, Kei-Léo Brousmiche, Antoine Durand, Eric Thea, and Elyes Hamida. Consortium blockchains: Overview, applications and challenges. September 2018.
18. Tien Tuan Anh Dinh, Rui Liu, Meihui Zhang, Gang Chen, Beng Chin Ooi, and Ji Wang. Untangling blockchain: A data processing view of blockchain systems. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1366–1385, jul 2018.
19. Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, Srinivasan Muralidharan, Chet Murthy, Binh Nguyen, Manish Sethi, Gari Singh, Keith Smith, Alessandro Sorniotti, Chrysoula Stathakopoulou, Marko Vukolić, Sharon Cocco, and Jason Yellick. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, pages 30:1–30:15, New York, NY, USA, 2018. ACM.
20. Arati Baliga, Nitesh Solanki, Shubham Verekar, Amol Pednekar, Pandurang Kamat, and Siddhartha Chatterjee. Performance characterization of hyperledger fabric. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, jun 2018.

Gonçalo S. Mendes is a researcher at Aptoide S.A, Lisbon, Portugal. He holds a MsC in Machine Learning, from Faculdade de Ciências e Tecnologias, da Universidade Nova de Lisboa. His researches interests are artificial intelligence, cloud development and high-performance computing. Contact him at goncalo.mendes@aptoide.com.

Daniel Chen is a researcher at Aptoide S.A, Lisbon, Portugal. He holds a BsC in Computer Science, from Iscte - Instituto Universitário de Lisboa. His researches interests are cloud development and mobile applications. Contact him at daniel.chen@aptoide.com.

Bruno M. C. Silva is an Assistant Professor and Head of the Technology at IADE. He is a researcher at Instituto de Telecomunicações, Universidade da Beira Interior, is also a member of the Centro de Investigação em Cidades Inteligentes do Instituto Politécnico de Tomar. Moreover, he is a member of many international TPCs and participated in several international conferences organization. He authors or co-authors several international conference Journal publications. His research areas include: Delay Tolerant Networks; Vehicular Networks; Mobile Computing; but especially: e-Health; Mobile Health; Internet of Things; and Ambient Assisted Living. Contact him at bruno.silva@it.ubi.pt.

Carlos Serrão is an Assistant Professor at Iscte - Instituto Universitário de Lisboa, in the Information Sciences and Technologies Department. Interested in the research areas of "Distributed Applications and Systems", "Information Security", "Mobile and Web Information Security" integrates the SSE (Software Systems Engineering) research group of ISTAR-IUL. Holds a PhD in Computer Architecture and Distributed Systems, from Universitat Politecnica de Catalunya (Barcelona, Spain). Participated in multiple national and international projects. Author and co-author of multiple research articles and communications in international events. Member of the ACM and OWASP and founding member of the AP2SI (Associação Portuguesa para a Promoção da Segurança de Informação). Entrepreneur in some IS/IT startups. Contact him at carlos.serrao@iscte-iul.pt.

João Casal is the head of R&D at Aptoide S.A, Lisbon, Portugal. He is an experienced applied R&D Manager with participation in over 10 projects in the last 8 years. Holds an MSc in Systems Engineering from Universidade do Minho and several professional certifications in Project Management. His research interests encompass several tech trends that relate with mobile and ubiquitous computing, from cybersecurity to artificial intelligence, IoT and blockchain. Casal co-authored over 20 peer-reviewed papers in these areas. Contact him at joaocasal@gmail.com.