San Jose State University

SJSU ScholarWorks

Faculty Research, Scholarly, and Creative Activity

7-8-2022

Cloud-centric blockchain public key infrastructure for big data applications

Brian Tuan Khieu
San Jose State University

Melody Moh
San Jose State University, melody.moh@sjsu.edu

Follow this and additional works at: https://scholarworks.sjsu.edu/faculty_rsca

Recommended Citation

Brian Tuan Khieu and Melody Moh. "Cloud-centric blockchain public key infrastructure for big data applications" *Research Anthology on Convergence of Blockchain, Internet of Things, and Security* (2022): 314-329. https://doi.org/10.4018/978-1-6684-7132-6.ch019

This Contribution to a Book is brought to you for free and open access by SJSU ScholarWorks. It has been accepted for inclusion in Faculty Research, Scholarly, and Creative Activity by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

Chapter 19
Cloud-Centric Blockchain Public Key Infrastructure for Big Data Applications

Brian Tuan Khieu
San Jose State University, USA

Melody Moh
https://orcid.org/0000-0002-8313-6645
San Jose State University, USA

ABSTRACT

A cloud-based public key infrastructure (PKI) utilizing blockchain technology is proposed. Big data ecosystems have scalable and resilient needs that current PKI cannot satisfy. Enhancements include using blockchains to establish persistent access to certificate data and certificate revocation lists, decoupling of data from certificate authority, and hosting it on a cloud provider to tap into its traffic security measures. Instead of holding data within the transaction data fields, certificate data and status were embedded into smart contracts. The tests revealed a significant performance increase over that of both traditional and the version that stored data within blocks. The proposed method reduced the mining data size, and lowered the mining time to 6.6% of the time used for the block data storage method. Also, the mining gas cost per certificate was consequently cut by 87%. In summary, completely decoupling the certificate authority portion of a PKI and storing certificate data inside smart contracts yields a sizable performance boost while decreasing the attack surface.

INTRODUCTION

Verification of one's identity continues to be the cornerstone upon which any interactions or transactions between two parties lie. One key method of verifying one's identity is through using public and private keys, which are cryptographically related strings that can be used to lock and unlock files. If a public key is used to lock a file, only its corresponding private key can be used to unlock it and vice-versa. People

DOI: 10.4018/978-1-6684-7132-6.ch019

Copyright © 2023, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

could use a public key to lock or encrypt a file, and they would be sure that the only person who could unlock it would be whoever held the matching private key. However, the issue after the establishment of public and private keys was identifying whether or not someone's private key and persona were appropriately matched. Malicious actors could claim to be another party and attempt to associate their own public key with the false persona in an attempt to redirect and steal sensitive information. Thus, public key infrastructure (PKI) was born in order to properly associate online identities with the correct public keys so that any online communication could be trusted to involve the correct parties.

However, with the pervasiveness and expansion of the Internet of Things, there comes new challenges for securing and authenticating the heavy flow of data generated by IoT devices. PKI's age has shown, and it has been unable to keep up with the demands of the IoT and Big Data era (Claeys, Rousseau, & Tourancheau, 2017). Big Data ecosystems require solutions that are scalable and resilient, two attributes that fail to be applied to traditional PKIs. Thus, in order to further secure the internet, a new method for identity verification over the web needs to be realized. The main issue with the currently outdated PKI lies with the Certificate Authority (CA) portion of the PKI (Doukas, Maglogiannis, Koufi, Malamateniou, & Vassilacopoulos, 2012; Gupta & Garg, 2015). CAs are the authorizing parties within a PKI; they validate and associate online personas with public keys by distributing and revoking digital certificates. These digital certificates act as ID cards for anyone that communicates over the internet, and they give a degree of assurance that the party one is communicating with is actually who they say they are. As of now, these CAs are the main points of failure within a PKI system; once any one CA is compromised, the whole PKI crumbles (Zhou, Cao, Dong, &Vasilakos, 2017). Furthermore, it is currently extremely difficult for a traditional CA to revoke an old identity. However, there are newer iterations of PKI that attempt to overcome these shortcomings; one of which is called Web of Trust (WoT). Another promising new solution marries the traditional PKI system with that of cloud and blockchain technology to overcome the weaknesses of the past (Tewari, Hughes, Weber, & Barry, 2018). Both of these systems are new ways of verifying identities that can pave the way towards a safer and more secure internet.

In this chapter, we explore the current state of PKI and its new incarnations that attempt to address the limitations of traditional systems. By doing so, we aim to answer the following questions: "How can new technologies such as blockchain be leveraged to improve traditional PKIs" and "What are the pros and cons of using one new solution over another". This chapter is extended from a conference paper which reported preliminary results (Khieu & Moh, 2019).

This chapter is organized in the following manner. First, it will establish background information surrounding the project and then cover research related to this area with a specific focus on other implementations of different PKIs. Afterwards, the methodology and reasoning behind our solution to the issues with the PKI model will be detailed. Subsequently, the chapter will cover the test results and performance comparisons between the different PKI models including our solution. Finally, the chapter will conclude with a summary and areas for future work.

RESEARCH OBJECTIVE

The objective of this research is to test and implement a Cloud-based blockchain PKI system, CBPKI, to provide Big Data applications with a scalable and persistent identity management system. In addition, the goal is to determine whether such a system can outperform traditional PKI models using metrics such as complete revocation time.

PUBLIC KEY INFRASTRUCTURE

A public and private key are two mathematically and cryptographically linked strings of characters. The two of them together form a key pair; an operation conducted using one can be reversed using the other half of the key pair. In practice, encrypting a document with a public key ensures that the only feasible method of decrypting said document is by using the corresponding private key. As their names imply, a public key is the public portion of a key pair while the private key is the private portion. Key pairs help ensure the privacy and integrity of online communications.

A public key infrastructure is a system that authenticates devices and handles digital certificates; digital certificates are essentially virtual ID cards designated by the PKI to associate an identity with a public key. By relying on a public key infrastructure, users trust that any party with a digital certificate distributed by the PKI is accurate. Once a certificate is retired or deemed to be compromised, the public key infrastructure will revoke that certificate to protect other users from communicating with anything that tries to use the revoked certificate. Currently, PKIs revolve around a Certificate authority (CA) to administer and revoke certificates using the X.509 standard.

Certificate authorities certify and issue these certificates to requesting users. Since they are central to the security provided by PKIs, they often must endure more attacks. Once a certificate authority is compromised, the integrity and authenticity of every certificate within the ecosystem comes under question.

WEB OF TRUST

As Chen, Le and Wei (2009) explain, WoT is a new system that attempts to validate one's identity and corresponding public key in a different way from how traditional PKI systems do. In order to avoid the single point of failure of PKIs, it mimics human society and its network of authenticating statements by the individuals; trust between parties is generated through past interactions with other parties. Certificate Authorities are not needed or used by WoT systems since they disperse the role of authenticator and validator to all of the user in the system. The basis for WoT is comprised of three rules: When two parties interact, they generate "trust information" (Chen et al., 2009). Secondly, every party knows what information belongs to them. And finally, every party must provide "trust" feedback to everyone else. These tenets form a model in which users are able to make decisions on whether to trust others.

In the research conducted by Bakar, Ismail, Ahmad and Manan (2010), it is stressed that WoT relies on a decentralized nature of authenticating users and they push for a trust scoring process for WoTs. The researchers postulate that this decentralization property allows the systems to overcome the past issues of traditional PKIs. With users providing trust scores for each other, there is no longer a need for a CA to dispense certificates. Instead, each user acts as a fraction of a CA, and only when enough pieces are put together is something allowed to be verified and used. Thus, according to the research by Bakar et al. (2010), the single point of failure in PKIs, the CA, is completely removed as a factor in WoT systems. However, this avoidance of a need for a CA comes at a cost as stated by the two papers referenced above.

While WoT systems lack the weak spot that traditional PKIs possess, the study conducted by Alexopoulos, Daubert, Mühlhäuser, and Habib (2017) criticizes the WoT model and conclude that the decentralized nature of the model negatively impacts its ease of setup and limits its applicable areas. By its very nature, WoT systems require further assistance in the setup phase than other PKIs (Alexopoulos et al., 2017). Not only does it require a sizable amount of users to be effective, it also needs enough of a history of interactions to generate meaningful trust scores. Furthermore, this process is not automatable, because the system is based on activity from real users. WoT complicates the initial setup, because it requires a number of real people to manually validate other real people until it can successfully run on its own. In addition to the effects on its setup, WoT cannot be applied to every use case due to its decentralized nature (Bakar et al., 2010). For example, systems that require a strict hierarchy of users, e.g. government agencies, do not have time for nor wish for the trust building process. If a system cannot handle the initial setup requirement of having enough users validate both each other and enough new users, then WoT should not be applied to it. In the case of Big Data applications and ecosystems, the Web of Trust model fails to properly meet the needs of the two. The setup requirements are too burdensome, and while WoT can eventually scale properly, the need for cross-verification would be another bottleneck on the ecosystem,

BLOCKCHAIN PKI

Blockchain technology is a relatively new invention; it revolves around the usage of a public immutable ledger called a blockchain. Multiple parties on a network encode transactions into this ledger after going conducting a verification process. In a blockchain network, every party holds essentially the same power to verify new transactions that wish to be recorded onto the public ledger. The whole system is decentralized, and the users act as a self-policing force to ensure the integrity of every transaction that is embedded into the public ledger.

Within a blockchain network, miners, nodes that hold a current version of the ledger, compete with one another in an attempt to be the first to mine a new block on the ledger. They do so to receive transaction fees for their services rendered. Once a block has been mined, the data will be publicly accessible on the blockchain network.

Closely associated with blockchains, smart contracts are digital contracts typically hosted on an electronic public ledger. It comprises of an agreement between two parties and facilitates the completion of said agreement. The smart contract allows money to be deposited within it for holding and future disbursement upon successful completion of the aforementioned agreement between both parties. If a party does not abide by the terms, the money the contract holds will automatically refund the held deposit and self-terminate. Smart contracts are self-executing and can be used in a large variety of applications from legal processes to residential leases.

Tewari et al. (2018) emphasized the viability of integrating current iterations of both PKIs and blockchains. In order to shore up the current weakness of PKIs, the group implemented a blockchain PKI management framework by developing a hybrid x.509 certificate that included details such as the blockchain name and hashing algorithm used. The system relied upon a Restful service as a communication medium; users could request certificates or certificate revocation lists (CRL) from the framework. Upon receiving a request for a new certificate, the framework would generate a hybrid certificate, send the requesting user a copy of it, and finally embed the certificate into a smart contract for storage in the Ethereum blockchain (Tewari et al., 2018). This approach fused many aspects of both blockchain technology and traditional PKIs; this lends itself an increased ease of usage and compatibility due to the usage of pre-existing technologies.

Along similar lines, Alexopoulos, Daubert, Mühlhäuser, and Habib (2017) highlighted the importance of establishing perpetual access to the new PKI framework in order to avoid certain attacks. In their study, they also utilized blockchain technologies to circumvent the issues with CAs. The authors note that the CAs of old could be blocked by Denial of Service (DOS) attacks that would prevent the distribution of CRLs (Alexopoulos et al., 2017). Since internet entities rely on checking these CRLs for whether a given certificate is still valid, the denial of access to them would allow faulty or compromised certificates to be accepted. Because of the decentralized and immutable nature of blockchain ledgers, they can be used to easily distribute the lists of certificates a Certificate Authority has created as well as its CRL. This makes DOS attacks ineffective versus this new PKI model; there will also be ensured access to the PKI due to the distributed nature of the blockchain (Tewari et al., 2018; Alexopoulos et al., 2017). However, these benefits of integrated blockchain PKIs come at the cost of limitations created through integration; one such limit is that CRLs can often exceed the size of a block which poses an issue for access to said CRLs.

Conversely to the previous approach by the two groups, Chen, Yao, Yuan, He, Ji, and Du (2018) decided to completely overhaul the PKI model and developed a service that heavily utilizes blockchain technology and concepts. CertChain is a new PKI scheme that aims to overcome the issues with past implementations of blockchain based PKIs. Specifically, past implementations possessed three issues: Centralized nodes could overpower other nodes and thus control a larger share of the blockchain. When checking the history of operations on a certificate, the entire blockchain would need to be traversed in order to find said info. And finally, block size limits would often break up CRLs whose size could reach 76MB. Unfortunately, for each of these issues, Tewari et al. and Alexopolous et al. chose to leave these issues as future concerns and did not implement safe guards versus them (Tewari et al., 2018; Alexopoulos et al., 2017).

Throughout their study, Chen et al. (2018) develop solutions for the problems with the integrated approach taken by Tewari et al. (2018) and Alexopolous et al. (2017) in order to create a more robust blockchain PKI. J. Chen et al. addresses the first issue of centralization through dispersing the trust within the system by use of a distributed dependability-rank based protocol (Chen et al., 2018). Essentially, an incentive mechanism was put in place to determine whether or not a CA would be made leader of a block like that of a centralized node in typical blockchains. Each CA would be given a dependability rank that would move up or down depending on the CA's good or bad behavior. On the other hand, the approach Tewari et al. and Alexopolous et al. took relies on the infeasibility of aggregating that much control over the network of nodes (Tewari et al., 2018; Alexopoulos et al., 2017). Rather than build in a safeguard like how J. Chen et al. does, they prefer to pay the sacrifice of provable security for the convenience of integration (Tewari et al., 2018; Alexopoulos et al., 2017; Chen et al., 2018). Regarding the second issue of traversal, this was solved through the proposal and development of a new data structure called CertOper while this issue goes unaddressed by Tewari et al. and Alexopolous et al. (Tewari et al., 2018; Alexopoulos et al., 2017). This data structure would be stored in a block and allow for operations such as efficient query and forward traceability (Chen et al. 2018). And finally, the issue of block size limits for CRLs was solved through the usage of Dual counting bloom filter, a method that efficiently stores data space wise and eliminates false positives that may come up during queries (Chen et al., 2018). And in regards to the existing integrated versions of blockchain PKIs, they also fail to address this issue (Tewari et al., 2018; Alexopoulos et al., 2017). CertChain builds upon the idea of blockchain based PKIs by overhauling previous hybridized systems in order to overcome the inherent issues with the fusion approach. Unfortunately, while it is faster and more robust in its security, it will be quite expensive to implement and maintain. The second problem, the issue of traversal, places an unnecessary bottleneck on the verification process of a certificate; this hindrance prevents blockchain PKIs from fully addressing the needs of Big Data applications and ecosystems.

REAL WORLD STATUS OF PKI

Current traditional PKIs are unable to keep up with the demands of new applications; in one survey, over 60% of respondents stated that their current PKI system was unable to handle new apps regardless of what the software was based in (Grimm 2016). Some companies have already taken steps towards basing their PKI in the cloud; others use newer versions such as the Web of Trust (WoT) approach. However, the WoT version generally lacks speed due to its manual nature of authenticating new users even though it is quite strong in its security.

The literature notes that WoT systems are currently used in several applications when they fit the use case, but blockchain based PKIs lack the same amount of adoption. Bakar et al. notes that while WoT is not applicable to every situation, it lends itself well to systems that are also decentralized in nature and can tolerate the manual addition and validation of new users. Currently, WoT is used in Peer to Peer (P2P) file-sharing networks and is embedded in some email clients (Bakar et al., 2010). Despite its manual nature, WoT has been adopted by some niche applications, but this manual nature and large setup requirements hold it back from being properly applied to Big Data applications. Blockchain based PKIs have not been as widely adopted primarily due to the lack of commercial offerings. This new iteration of a PKI has not fully left the research and development phase of its life cycle. However, it has further reaching consequences than that of WoT systems; it possesses more viability in replacing traditional PKIs than WoT systems do (Alexopoulos et al., 2017). And while there may not be any commercially available blockchain based PKIs available, Tewari et al. stresses that they can be implemented if one is willing to bear both the cost of development as well as that of encoding certificates into a blockchain (Tewari et al., 2018).

SECURITY OF CLOUD AND IOT

Past work in the area of cloud computing and IoT note the inherent security issues with both and attempt to rectify them in a variety of manners. In regards to IoT, these problems result from the computing limitations of the devices which prevents full-fledged security solutions from being applied to them. Stergiou, Psannis, Kim, and Gupta (2018) surveyed the literature surrounding cloud and IoT technology and identified limitations such as that regarding the security of a cloud service provider to customer relationship. The customer must surrender any potentially sensitive and damaging information to the provider by use of the service and trust that the provider has undertaken proper measures to protect said information. With the integration of cloud and IoT, issues arise regarding the storage location of sensitive data and lack of trust in service-level agreements. The group developed a new hybridized Advanced Encryption Standard (AES) platform composed of both IoT and cloud technologies to rectify some of these limitations (Stergiou et al., 2018). Similarly, Tewari and Gupta addressed the security issues regarding IoT resulting from the limited resources of its devices (2016). Their approach differed by use of an ultra-lightweight solution consisting of only bitwise operations for mutual authenitication between RFID tags and readers. Due to lack of computationally intensive operations, this method would be more readily usuable directly by the IoT sensors themselves.

Furthermore, the literature highlights the need for a proper management system in IoT and Big Data ecosystems and proposes several solutions. In one paper, the authors proposed that all data from IoT devices in a smart building be relayed to a cloud server with the additional ability to control such sensors remotely (Plageras, Psannis, Stergiou, Wang, & Gupta, 2017). This system aimed to provide proper monitoring of sensors and optimization of energy efficiency within a cloud environment. Contrastingly, another approach took techniques applied to Wireless Sensor Networks (WSN) to develop a new solution, an Efficient Algorithm for Media-based Surveillance Systems (EAMSuS) to provide privacy and security for IoT networks (Memos, Psannis, Ishibashi, Kim, & Gupta, 2017). WSN techniques were paired with High Efficiency Video Encoding (HEVC) compression and one-time pads in order to cover the privacy and security weaknesses of a one-to-one adaptation of WSN techniques to IoT ecosystems.

SUMMARY OF THE LITERATURE

The literature identifies that researchers have developed two key methods for solving the issues with traditional PKIs. D. Chen et al. and Bakar et al. developed and tested WoT systems; these systems avoid incorporating the CAs, the single point of failure of past PKIs, by offloading the validation and addition of new users through a community based trust model (Chen et al., 2009; Bakar et al., 2010). Alternatively, Tewari et al. (2018) developed a blockchain based PKI that demonstrated its practicality and compatibility with current technologies while also improving and protecting CAs. Chen et al. (2018) took the scheme further and developed a new system that overcomes the issues plaguing existing blockchain based PKI models. In summary, while PKIs have become outdated, there are two models that improve upon and may ultimately replace traditional PKIs to pave the way for a safer, more secure internet. However, there are still areas that require more development and research to increase the viability of replacing traditional PKIs. And even then, both solutions do not completely satisfy the needs of current Big Data applications in terms of scalability and persistence.

Additionally, the literature notes the inherent weaknesses in the security of IoT environments due to the computational limitations of the sensor devices. Stergiou et al. developed a hybrid cloud and IoT AES system to address the weaknesses of both cloud and IoT systems with the strengths of the other (2018). Conversely, another approach minimized the need for heavy computation through simplifying authentication down to bitwise operations (Tewari & Gupta, 2017). Several management systems were also proposed for IoT and Big Data environments. One revolved around the use of a cloud server to provide remote monitoring and control of sensors (Plageras et al., 2017). Another proposed the usage of WSN techniques paired with HEVC and one-time pads for proper application to IoT networks (Memos et al., 2017).

RELATED RESEARCH

Several other public key infrastructure models also utilize blockchain technology. These other models do closely resemble our proposed model, but we have made significant departures from their approaches.

X509Cloud, the model proposed by Tewari et al. (2018) emphasized the storage, retrieval, and revocation of certificates. In order to circumvent the associated maintenance costs of verifying identities, the model aims for mutual authentication between users and the organizations. The framework connects a cloud service to a Bitcoin inspired blockchain protocol that is used to store in newly created certificates. This approach differs from CBPKI in that instead of storing the entire certificate itself within a blockchain, mine holds all relevant certificate data within a smart contract.

A paper authored by Alexopolous et al. (2017) analyzed the merits of integrating open distributed ledgers (ODLs). Alexopolous et al. (2017) developed a formally defined trust management model for use with ODLs; these ODLs are the ledgers that blockchain technologies have implemented and center around. The paper also provided an analysis of common attacks versus typical trust management systems and detailed how the use of ODLs assisted in mitigating or even preventing the harm. This mathematical model was used in part as a blueprint for the implementation of CBPKI. It also inspired further modifications in an effort to circumvent the common denial-of-Service attack.

Conversely, to the previous approach by the two groups, Chen et al. (2018) overhauled the PKI model and developed a service that heavily utilizes blockchain technology and concepts. The proposed model, CertChain, does not simply embed certificates inside of a blockchain. Instead, CertChain uses a newly created data structure, CertOper, to aid in both the storage of certificates within a blockchain and traversal along a blockchain. Also, it modified certificate authorities into miners belonging to the CertChain blockchain network. CBPKI does not go as far as CertChain does in terms of using a newly developed blockchain system specifically designed to fix the power centralization and block traversal issues. Rather, CBPKI uses the pre-existing blockchain Ethereum to store certificate data. However, CertChain directly inspired the use of smart contracts to store certificate data to address the traversal issue, and the usage of smart contracts is addressed in section 10.1.1.

These projects inspired CBPKI, but they differ from it in some significant aspects. While CBPKI utilizes blockchain technology to store data, it does not store the certificate itself; rather, it holds certificate data such its validity and expiration date within a blockchain. In addition, our proposed model altered the certificate authority portion of a PKI by hosting it as a stateless web app on a cloud provider.

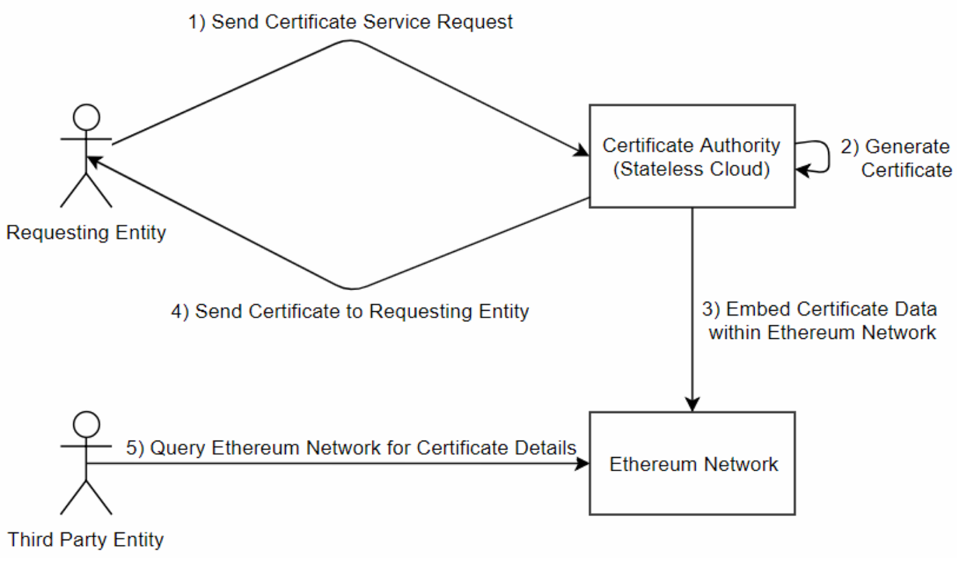
PROPOSED SOLUTION AND METHODOLOGY

The following sections detail our newly proposed model for a PKI system that utilizes blockchain and cloud technologies, and they are organized as follows. Firstly, the model itself and the enhancements are explained in the next section. Afterwards, implementation details are covered, and the final section discusses the differences CBPKI possesses compare to models from related works.

Model

CBPKI consists of a stateless certificate authority that stores certificate data on the Ethereum network, a blockchain network that allows unlimited processing potential for smart contracts. The certificate authority is implemented as a Restful API; upon receiving a certificate service request posted to it, the CA generates a new certificate. Shortly following, the new certificate is embedded into a blockchain whose address is listed on the certificate itself. A python script is used to check for the certificate's appearance within the blockchain. Upon revocation of a certificate, the certificate is similarly embedded within a different blockchain for further verification purposes. Figure 1 below displays the overall flow of the generation and verification of a certificate using CBPKI.

Figure 1. CBPKI process of certificate generation and verification


1) Send Certificate Service Request
Certificate Authority (Stateless Cloud)
2) Generate Certificate
Requesting Entity
4) Send Certificate to Requesting Entity
3) Embed Certificate Data within Ethereum Network
5) Query Ethereum Network for Certificate Details
Ethereum Network
Third Party Entity

Enhancements

One specific enhancements over past PKI models is the transformation of the certificate authority into a stateless web service hosted on the cloud. The conversion of the certificate authority into a stateless protocol hosted on a cloud platform significantly reduces the size of the viable attack surface. Since coveted data is no longer stored within the CA itself, it also lowers the value of targeting the certificate authority for attacks. In addition, stateless web services are more conducive to relying on the protections offered by cloud platforms. For example, Amazon offers a web traffic monitoring service named AWS Shield to secure stateless web services. In addition, hosting the CA on a cloud platform with auto-scaling mitigates the common Denial-of-Service attack; with auto-scaling, more resources are automatically provisioned to the certificate authority so it can handle the flood of requests during a DOS attack.

The second enhancement made by CBPKI is the use of smart contracts as a storage device on the blockchain network. In general, the use of blockchains in PKI systems allows persistent certificate revocation list access and further circumvention of DOS attacks. Using smart contracts to store certificate data instead of using block transaction data fields yields the additional benefits of lowered mining times and operating costs. Also, CBPKI removes the need for CRLs due to the storage of certificate validity within the smart contracts. This allows for direct verification of a certificate as opposed to traversing a blockchain for a certificate's status thus making our solution better fit the needs of Big Data applications.

Implementation

As noted earlier, CBPKI builds on the works of Tewari et al. (2018), Alexopolous et al. (2017), and Chen et al. (2018). Three different PKIs were built for testing and comparison purposes: a traditional PKI and two Cloud Based PKIs utilizing blockchain technology. Both the traditional PKI and the Cloud PKIs used a remote CA in order to standardize the experiment. The overarching approach is as follows:

1. Implement a Restful API using Python and Django to act as a CA. Allow for the traditional CA to service queries for its certificate revocation lists (CRL).
2. Connect the Certificate Authority to the Ethereum Test Net Ropsten. Associate a cryptocurrency wallet with the CA in order to pay for the associated cost of embedding a certificate
3. For one of the blockchain PKIs, hash a X.509 certificate and embed it into the transaction data field of a block and send it to be mined in Ropsten. Similarly for the new approach, set up a smart contract in which to embed the hash of an X.509 certificate and send it to be mined in Ropsten. In both cases, the X.509 certificate requires the address of where it is stored within Ethereum.
4. Implement certificate verification methods for the Blockchain PKIs. This involves searching for the hash of a certificate within a blockchain or pulling certificate information from the smart contract.

Distinct Features

The modifications to the approach created by Tewari et al. (2018) and Alexopolous et al. (2017) center around the hosting of the CA on the cloud as well as the embedding of certificate data in smart contracts. Both research groups utilized a Restful API implementation of a CA in order to accept and service requests for certificates. However, they did not host the CA in the Cloud; conversely, CBPKI's hosting of the CA within a cloud service such as Amazon Web Services (AWS) allows for some added security benefits. For one, AWS offers web traffic monitoring and filtering of requests to a web app. In addition, Denial of Service attacks are mitigated since AWS with its auto-scaling feature will simply continue to provision more resources in order to meet the increased demand. This change allows for CBPKI to scale properly with any changes in demand by Big Data applications.

CBPKI also drew upon the hybridized certificate implementation from Tewari et al. (2018) and Alexopolous et al. (2017); the modified certificates possess information regarding where pertinent certificate or CA information is stored in the blockchain network. In addition to the hosting of the CA on a cloud service, our proposed solution differs from the two referenced papers' approach in how certificate data is stored within the blockchain network. Instead of storing the revocation list data within a block's transaction data field, our new approach stores said data within smart contracts. This has the added benefit of lower gas cost to be mined as well as quicker mining speed as opposed to the block approach. Also of note is that this method addresses the traversal issue with the approach of Tewari et al. (2018) and Alexopolous et al. (2017) as put forth by Chen et al. (2018). Instead of having to traverse the blockchain in search of a revoked certificate, the certificate itself would contain the direct address of where to access the smart contract that contains all of the relevant data regarding said certificate. Since smart contracts can update its data fields, the CA can simply update the status of a certificate to revoked; this removes the need for CRLs and eliminates the traversal issue certificate data will be directly accessible. Note that in order to use this PKI system, one needs to simply implement a quick verification script that matches the hash of the certificate on hand with that stored within the smart contract. This use of smart contracts helps our solution better fit the needs of Big Data applications in both terms of scalability and persistence. For one, it avoids the need for a traversal along the blockchain in order to verify the certificate. And secondly, there should be minimal persistent access issues to the data required for verification since said data is being held in a distributed network with no single point of failure.

PERFORMANCE AND RESULTS

This section consists of the performance and results from the experiments conducted on the CBPKI model. Three different models were used for testing, and these models consisted of one traditional PKI and two variants of the CBPKI model. The section is organized as follows. First, the experimental settings subsection details the environment and testing methods used for each model. Then, the results subsection summarizes the outcomes of the tests. And finally, the last subsection evaluates each model's level of security compared to one another.

Experimental Settings

The experiments conducted on our proposed work revolved around access times and the costs associated with operating different models. Our proposed model utilized many different software tools, and two different variants were used alongside one another for the tests. As mentioned earlier, three models in total were subjected to the same conditions, and their performance was evaluated based on three metrics relating to speed, time, and cost where applicable.

Implementation and Resources

The following software resources are required for the implementation of our project: Python Programming language, Python Packages Cryptography, Hashlib, and Web3.py, Heroku, Django web framework, Django API TastyPie, Solidity Smart Contract programming language, X.509 certificates, Ethereum Test Net Ropsten, MetaMask, Ethereum IDE Remix, Etherscan, and the Infura Ethereum API. Django and the API Tastypie were used to implement the CA as a Restful API; these two technologies were available at no cost. Django web apps can be hosted by use of Heroku, a cloud platform as a service, and Heroku could also be used at no cost by use of its free tier. The Solidity smart contract programming language is available at solidity.readthedocs.io. In addition, the Python programming language is also available at python.org, and the X.509 certificates being used can be imported by installing the pyca/cryptography package for python with pip. Also, the Hashlib python package, which is used to hash certificates, and Web3.py, which handles connections to the Ethereum network, are available through pip. Ethereum Test Net Ropsten is a test blockchain network for the cryptocurrency Ethereum. MetaMask is a free Ethereum wallet which is used to pay for the gas required to mine blocks or smart contracts. Ethereum IDE Remix is a free IDE for Ethereum smart contracts used to create and deploy said smart contracts. Also, Etherscan is a website that can monitor transactions, smart contracts, and wallets. Finally, Infura is a blockchain API that allows a connection between the Ethereum network and a python script. Below in Table 1 is a compilation of all software tools used along with their associated costs.

Table 1. Software tools used and associated costs

Software Tool	Cost
Python Programing Language	Free
Python Package: Cryptography	Free
Python Package: Hashlib	Free
Python Package:Web3.py	Free
Heroku Cloud Platform	Free Tier Used
Django Web Framework	Free
Django API: TastyPie	Free
X.509 Certificates	Free
Ethereum Test Net Ropsten	Free
Ethereum IDE Remix	Free
Etherscan	Free
Infura Blockchain API	Free
Solidity Smart Contract Programming Language	Free

Experimental Settings

Every test run used the same settings amongst each model; fifty runs were conducted for each model type. Across different models, the most similar conditions as possible to one another were used; each PKI model would be loaded with the same initial dataset of a 2 MB large certificate history list. This history list consisted of details regarding old certificates the CA had distributed in the past. Each model was also paired with a certificate revocation list that was 1MB big; this CRL comprised of half of the distributed certificates within the certificate history list. The traditional model stored the both the history and revocation lists within cloud platform. Instead, the CBPKI block storage version held the data from both lists within two separate blockchains while the smart contract version used smart contracts as storage devices.

Experimental Settings

The metrics used for judgment of both the traditional PKI and the CBPKIs will be revocation status access time. Certificate revocation status time within this chapter is defined as the time it takes for a given certificate's status to be verified. In addition, the two CBPKIs will have additional metrics regarding mining time of certificate data and gas costs of mining certificate data in their different storage methods. Mining time is the time it takes for the data storage method to be mined and thus publicly accessible on the blockchain network. Mining gas costs are the amount of gas or money required in order to pay miners to service the mining request.

PKI Models

During the experiments, three PKI models were used; these models were the traditional version, CBPKI block storage version, and the CBPKI smart contract version. The traditional version is hosted on a cloud service to limit variables, but it still holds all certificate data and certificate revocation lists together with its certificate authority. The two CBPKI models instead use a stateless CA hosted in the cloud while holding certificate data within the Ethereum blockchain network. However, they differ distinctly in how they store said data. The block storage version stores the certificate itself within the transaction data field while the smart contract version merely stores a hash of the certificate along with key certificate data inside smart contracts.

Results

Overall, there is a notable improvement in each area for the Cloud PKI implementation using smart contracts to store certificate data. While both CBPKIs allow for faster CRL retrieval than that of the traditional PKI, Table 2 shows that the smart contract version is faster than the block storage version. This is mostly likely due to how the block storage version requires a traversal across the blockchain in order to find the specific certificate while the smart contract version simply pulls the relevant validity data directly from the smart contract.

Table 2. Certificate revocation status access times (ms)

Model	Mean
Traditional	208.52
Block Storage	142.97
Smart Contract	129.34

Regarding the mining times, there is a significant speedup when using smart contracts as displayed in Table 3. Smart contracts are smaller and thus do not need as many resources to complete mining when compared to blocks. According to the monitoring done by Etherscan, most of the time used for mining the block was spent waiting to be serviced. Mining times are highly dependent on network congestion which helps explain the variance between different run times.

Table 3. Mining timings per certificate (ms)

Model	Mean
Block Storage	325.60
Smart Contract	21.36

The final category of tests centered on the mining gas cost to embed certificate data into the Ethereum network. As the Table 4 shows, the average cost of mining a smart contract is greatly reduced in comparison to that of an entire block. This average reduction of about $5 gives significant savings in the operational costs of a PKI utilizing blockchain technology. However, it pales in comparison to the traditional PKI since there is a negligible cost associated with storing a certificate in a database.

Table 4. Mining gas cost per certificate ($)

Model	Mean
Block Storage	5.97
Smart Contract	0.79

Security Analysis and Qualitative Comparison

The various public key infrastructure models covered in this chapter possess clear benefits and tradeoffs. Traditional models retain the discussed failings of certificate authorities; their large attack surface size makes them vulnerable and a target for infiltration and disruption. The weaknesses of these traditional models are well known which thus makes them quite susceptible to any attacks launched by malicious actors. However, they are cheap to operate in terms of issuing certificates, and apart from hosting and electricity costs, there is a negligible cost per certificate issuance. The CBPKIs both decrease this attack surface size significantly, and by making the CA stateless, they are able to piggyback on a cloud platform's security measures. But as noted in table 5 below, this comes at a significant operating cost.

Using blockchain technology as a storage device for certificates and certificate data does not come for free. Amongst the two CBPKI models, the smart contract variant outperformed the corresponding block storage version in every metric used. The smart contract version has the additional benefit of not requiring CRLs which has resulted in measurable performance boosts over the block storage version. Table 6 highlights the inherent tradeoffs between the two approaches. It is important to note that the smart contract version does not necessarily possess immutability. Depending on the implementation, the smart contract data fields could be subject to malicious alteration. While the code behind the smart contract itself is immutable, the data it holds does not possess this attribute. Thus, even though the block storage method may be more costly and slower to use than the smart contract CBPKI model, the block storage model is more secure since its records are immutable. Another thing of note is that one must be careful in programming a smart contract; since the code itself is immutable, a bugged smart contract can run forever on the network.

Table 5. Qualitative comparison of PKI models

PKI Model	Certificate Issuance Cost	CA Attack Surface Size
Traditional	Low	High
CBPKI Block Storage	High	Low
CBPKI Smart Contract	Medium	Low

Table 6. Qualitative comparison of CBPKI variants

PKI Model	CRL Size	Immutable Records
Block Storage	High	Yes
Smart Contract	Low	No

CONCLUSION

Traditional PKIs contain issues regarding its certificate authorities that inhibit their ability to properly meet the demands of Big Data ecosystems. These issues can be addressed by usage of integrating current PKIs with existing blockchain and cloud technologies. Offloading the certificate authority to the cloud allows the PKI to tap into existing security measures against common attacks such as Denial of Service. In addition, storing certificate data in the blockchain enables persistent CRL and certificate data access while avoiding potential caching issues and DOS attacks. The aforementioned qualities provide an advantage for our CBPKI model over past models in fitting the needs, scalability and availability, of Big Data applications and ecosystems.

The results of the conducted tests reflect a significant performance increase of blockchain PKIs over traditional PKIs. Furthermore, the proposed solution of storing certificate data in smart contracts also outperformed the block storage version in terms of CRL access time, mining speed, and mining cost. However, this mining cost is a large source of the operational costs associated with blockchain PKIs, something that traditional PKIs don't possess.

Areas for further research are based on the study conducted by Chen et al. (2018). As noted earlier, the group of researchers highlighted three specific issues regarding the approach taken by Tewari et al. (2018) and Alexopolous et al. (2017). Additional study could be conducted in trying to resolve these issues without having to completely overhaul the PKI system as done by Chen et al. (2018). In addition, since there is a significant operating cost associated with storing certificate data on a blockchain network, further study in the reduction of this cost is also warranted.

Additional areas for future work involve the cloud portion of the CBPKI model. Routing requests for certificates through a cloud service opens up a plethora of possibilities for the application of cloud and alternative services. One such possibility is to further merge the CBPKI model with a machine learning model to automatically verify and revoke faulty certificates, potentially before the certificate has even been issued.

REFERENCES

Alexopoulos, N., Daubert, J., Mühlhäuser, M., & Habib, S. M. (2017). Beyond the Hype: On Using Blockchains in Trust Management for Authentication. 2017 IEEE Trustcom/BigDataSE/ICESS.

Bakar, A. A., Ismail, R., Ahmad, A. R., & Manan, J. A. (2010). Trust Formation Based on Subjective Logic and PGP Web-of-Trust for Information Sharing in Mobile Ad Hoc Networks. *2010 IEEE Second International Conference on Social Computing*. 10.1109/SocialCom.2010.149

Chen, D., Le, J., & Wei, J. (2009). A Peer-to-Peer Access Control Management Based on Web of Trust. *2009 International Conference on Future Computer and Communication*. 10.1109/ICFCC.2009.77

Chen, J., Yao, S., Yuan, Q., He, K., Ji, S., & Du, R. (2018). CertChain: Public and Efficient Certificate Audit Based on Blockchain for TLS Connections. *IEEE INFOCOM 2018*.

Claeys, T., Rousseau, F., & Tourancheau, B. (2017). Securing Complex IoT Platforms with Token Based Access Control and Authenticated Key Establishment. *2017 International Workshop on Secure Internet of Things (SIoT)*. 10.1109/SIoT.2017.00006

Doukas, C., Maglogiannis, I., Koufi, V., Malamateniou, F., & Vassilacopoulos, G. (2012). Enabling data protection through PKI encryption in IoT m-Health devices. *2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)*.

Grimm, J. (2016). PKI: Crumbling under the pressure. *Network Security*, *2016*(5), 5–7. doi:10.1016/S1353-4858(16)30046-0

Gupta, R., & Garg, R. (2015). Mobile Applications Modelling and Security Handling in Cloud-Centric Internet of Things. *2015 Second International Conference on Advances in Computing and Communication Engineering*.

Tewari, H., Hughes, A., Weber, S., & Barry, T. (2018). A blockchain-based PKI management framework. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*.

Zhou, J., Cao, Z., Dong, X., & Vasilakos, A. V. (2017, January). Security and Privacy for Cloud-Based IoT:Challenges. *IEEE Communications Magazine*, *55*(1), 26–33. doi:10.1109/MCOM.2017.1600363CM

This research was previously published in Security, Privacy, and Forensics Issues in Big Data; pages 125-140, copyright year 2020 by Information Science Reference (an imprint of IGI Global).