2019

Adapting Financial Technology Standards to Blockchain Platforms

Gabriel Bello

Follow this and additional works at: https://csuepress.columbusstate.edu/theses_dissertations

Part of the Computer Sciences Commons, and the Finance and Financial Management Commons

Recommended Citation

ADAPTING FINANCIAL TECHNOLOGY STANDARDS TO BLOCKCHAIN PLATFORMS

Gabriel Bello
2019

COLUMBUS STATE UNIVERSITY

ADAPTING FINANCIAL TECHNOLOGY STANDARDS TO BLOCKCHAIN PLATFORMS

A THESIS SUBMITTED TO THE

HONORS COLLEGE

IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR HONORS IN THE DEGREE OF

BACHELOR OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

TURNER COLLEGE OF BUSINESS

BY

GABRIEL BELLO

ABSTRACT

Traditional payment systems have standards designed to keep transaction data secure, but blockchain systems are not in scope for such security standards. We compare the Payment Application Data Security Standard's (PA-DSS) applicability towards transaction-supported blockchain platforms to test the standard's applicability. By highlighting the differences in implementation on traditional and decentralized transaction platforms, we critique and adapt the standards to fit the decentralized model. In two case studies, we analyze the QTUM and Ethereum blockchain platforms' industry compliance, as their payment platforms support transactions equivalent to that of applications governed by the PA-DSS. We determine QTUM's and Ethereum's capabilities to properly ensure secure data handling with respect to current security standards. After adapting the PA-DSS and analyzing the QTUM and Ethereum platforms, we revise the new set of standards to create a set of best-practices for ensuring data security on both traditional and blockchain payment systems. We report the security gaps identified on each platform based on the final revision of the standards, presenting a conclusive perspective that neither platform is suitable for business adoption based on the PA-DSS standard's results. Finally, we discuss open research issues.

TABLE OF CONTENTS

Adapting Financial Technology Standards to Blockchain Platforms

Gabriel Bello
Columbus State University
Columbus, GA, USA
gabriel_bello@columbusstate.edu

ABSTRACT

Traditional payment systems have standards designed to keep transaction data secure, but blockchain systems are not in scope for such security standards. We compare the Payment Application Data Security Standard's (PA-DSS) applicability towards transaction-supported blockchain platforms to test the standard's applicability. By highlighting the differences in implementation on traditional and decentralized transaction platforms, we critique and adapt the standards to fit the decentralized model. In two case studies, we analyze the QTUM and Ethereum blockchain platforms' industry compliance, as their payment platforms support transactions equivalent to that of applications governed by the PA-DSS. We determine QTUM's and Ethereum's capabilities to properly ensure secure data handling with respect to current security standards. After adapting the PA-DSS and analyzing the QTUM and Ethereum platforms, we revise the new set of standards to create a set of best-practices for ensuring data security on both traditional and blockchain payment systems. We report the security gaps identified on each platform based on the final revision of the standards, presenting a conclusive perspective that neither platform is suitable for business adoption based on the PA-DSS standard's results. Finally, we discuss open research issues.

CCS CONCEPTS

• **Applied computing** → **Online banking**; **Secure online transactions**; *E-commerce infrastructure*; • **Security and privacy** → *Security services*;

KEYWORDS

Blockchain, Smart contracts, Privacy, Compliance, User data, Security frameworks, Application security, Financial technology, Security standards, Payment card industry

ACM Reference Format:
Gabriel Bello. 2019. Adapting Financial Technology Standards to Blockchain Platforms. In *CSU Honor's College Thesis*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.475/123_4

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CSU Honor's College Thesis, April 2019,
© 2019 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

1 INTRODUCTION AND MOTIVATION

1.1 Motivation

Legal liability in the digital age is increasingly dynamic, both in theory and practice. Contracts have transformed as industries grow more adjusted to technology. Paper and pen have morphed to lines of code, and with it, the legal contracts have been challenged in implementation. The definition of a contract remains the same: an agreement between two parties outlined by the terms and conditions, wherein value is exchanged [24]. The conditions set by this definition allow both parties to protect their interests through a document that binds them to a set of rules. As the world transitions to a more digitalized era, smart contracts have been introduced as a replacement for the arduous, tedious legal agreements of the past.

Smart contracts were first devised by Nick Szabo, who documented the idea of a contract automation in 1997 [19]. To present his idea, Szabo provided an example between a human and a vending machine: The vending machine, depending on the human's input (in coins) would allow or disallow candy to be dispensed from the machine. At the base-level, this is precisely what a smart contract does: assure an exchange of data with anybody who satisfies the constraints set forth by the contract. His rationale envisioned the smart contracts' adoption across the industry, spanning over multitudes of applications. His focus, however, revolved around the profitability and feasibility to implement this technology on a large scale, as any smart contract that is created, in his vision, should have safeguards whose robustness depends on the process performed.

According to Szabo [19], the security concerns for each smart contract should exist within the bounds of the business transaction. Essentially, as in any security system, the controls in place should not outstrip nor fall short of the functionality of the process. For example, the vending machine that Szabo describes also features security controls, such as a lock to open the machine. These mechanisms should not inhibit profitability. In 2002, Szabo developed a second work [20] that elaborates on his previous ideas on smart contract development. This second paper defines a set of guidelines to follow when designing smart contracts, such as monitoring code development to prevent exploitation (e.g. contract breaching), which may be done if a smart contract is not configured or programmed properly. Overall, this second work serves as a reference to create contracts, especially in the scope of auditing. Some of these components, which if not implemented properly may result in security gaps, can be seen in current smart contract implementations.

Current smart contracts are built on blockchain technology. Thus, to understand the functionality and security concerns of smart contracts, it is necessary to understand the focal points of blockchain security. Moreover, all instances of cryptocurrency, beginning with

Satoshi Nakamoto and bitcoin, are built on blockchain technology [18]. Blockchain provides a decentralization of information on a given network. Typical networks, which can be identified as centralized to contrast with blockchain, often have one central point where all the information is stored. The decentralized design of blockchain serves both as a failsafe and a protection against data alteration as the information is distributed and spread over multiple hosts (called nodes). Consequently, there is a copy of all data on each device that cannot be altered. Additionally, account management on blockchain platforms are often rooted in privacy measures to ensure anonymity on this decentralized network. From the perspective of cybersecurity, the challenge in smart contracts lies within the authentication, communication, and execution of the technology. The difference in implementation between centralized and decentralized networks results in a difference in security approach. Decentralized platforms may necessitate additional measures in place to verify a legally binding agreement, ensure privacy on a distributed network, and conduct a secure transaction of goods.

Smart contracts, along with blockchain as a whole, have grown in interest over the past years. Many companies have begun work on fitting the technology into their business model. The scope of smart contracts' potential impact on global solutions is wide. From business optimization to disaster recovery, smart contracts offer alternative methods to common business issues. However, with adoption in the industry, especially within the realms of the Payment Card Industry (PCI), healthcare, and other industries who handle private user data, user privacy protection is mandatory. As such, smart contract platforms that wish to be adapted to enterprise environments must comply or exceed certain security standards, often modeled after industry best practices. Whatever standard an organization chooses to use matters in the case of smart contracts, as often times the immutability of blocks on a blockchain result in tedious efforts to prevent disastrous vulnerabilities. That is, a smart contract with a security flaw must be disabled and replaced, which is a much taller task than rolling out a patch for centralized systems. As the roles of smart contracts and blockchain platforms grow as payment systems, so too does the necessity for proper auditing and compliance to ensure proper data protection. The purpose of this paper is to provide a method by which to measure the security of a smart contract's platform. This is done by categorizing the controls and mechanisms that the platform enforces for security and analyzing them to determine their compliance. To achieve this categorization and compliance analysis, an application security framework, paired with a risk assessment framework, will be used to evaluate the platform. Many organizations utilize the same types of frameworks to audit their security program for strengths and weaknesses; therefore, it is the baseline tool for this research.

This research adapts the Payment Application Data Security Standard (PA-DSS) to meet the needs of blockchain payment systems. As blockchain and smart contract platforms grow in popularity, the necessity for security standards on these platforms increases. Once the standards are reworked, we analyze two smart contract platforms as case studies: QTUM (pronounced *quantum*) and Ethereum. QTUM is a smart contract system that functions across multiple devices. Claiming functionality on mobile, QTUM seeks to bridge the gap between blockchain systems and the business world. As it stands, there is a disconnect between the two for multiple reasons,

namely the spatial complexity required for blockchains on a given host computer. QTUM offers solutions for the adoption of smart contract systems into enterprise environments, and this project seeks to outline the implemented security controls. Ethereum is a popular smart contract supporting blockchain with a large user base. Its structured smart contract language makes it ideal for business adoption. Both platforms have strengths that make them more adaptable to business functions than other platforms. We analyze both to test the revised PA-DSS standards.

1.2   Purpose of this Research

The QTUM (pronounced *quantum*) platform is the smart contract system that functions across multiple devices. Claiming functionality on mobile, QTUM seeks to bridge the gap between blockchain systems and the business world. As it stands, there is a disconnect between the two for multiple reasons, namely the spatial complexity required for blockchains on a given host computer. QTUM offers solutions for the adoption of smart contract systems into enterprise environments, and this project seeks to outline the implemented security controls. As a result, we will analyze the QTUM platform to determine if it is compliant with the standards set in the application security framework. We also analyze the Ethereum platform, a popular smart contract supporting blockchain with a large user base. Its structured smart contract language makes it ideal for business adoption. This paper is structured as follows:

- Section 1: Introduction and Motivation
- Section 2: Background Information
- Section 3: Related Work
- Section 4: Methodology
- Section 5: Results
- Section 6: Conclusion and Future Work

This research will be a traditional thesis, and the structure follows the standard Association for Computing Machinery format. Many venues are applicable for this work, especially those relating to either security or blockchain. The International Workshop on Emerging Trends in Software Engineering for blockchain hosted its first event in 2017, setting a precedent for blockchain in formalized research. This workshop is also a part of the larger International Conference on Software Engineering, a highly prestigious conference led by the Association for Computing Machinery (ACM). Additionally, there are multiple venues for security research, including Conference for Information Systems Security Education (CISSE) and ACM Southeast. The conference(s) this work is submitted to will depend highly on the emerging results, which will be more concrete as the project advances (Computer Science conferences acceptance and awards are highly results-based, as in many other discipline).

2   BACKGROUND INFORMATION

This section of the work discusses the preliminary technical information necessary to understand the analysis done later in the research. Smart contract anatomy, including qualities such as transaction models and consensus algorithms, requires a discussion of blockchain technology as a whole. This sections briefly outlines the different characteristics a platform may contain, which sets the foundation for critiquing the security mechanisms present on a

given system. This section is intended for a reader to familiarize himself or herself with the technology as it relates to the work documented in this paper.

Blockchain technology has the potential to solve major security principles by its inherent design. For example, data integrity, the confidence a party has in ensuring the information transmitted or accessed has not been altered, is partially solved by blockchain. Because the data is distributed over all nodes on the network, it is only possible to change the stored data if it is changed on all nodes; this is infeasible with current computational capabilities. Data availability is also evident, for all information is stored directly on a node's machine. This results in a scenario where a node has direct access to the blockchain and its data.

When discussing security controls in any capacity, it is essential to first form a foundation around what makes applications, systems, or data secure. The major tenets of security are confidentiality, integrity, and availability; some other tenets include authentication and nonrepudiation. There are many other concepts to consider in security, but the ones mentioned above are the ones necessary to understand the security mechanisms of smart contract platforms.

Confidentiality concerns keeping information private from unauthorized eyes. Common controls to ensure confidentiality include encryption, hash functions, and encoding. These are standard and required in many payment systems today. Integrity keeps information from altered by unauthorized users. Availability ensures that the data is available when needed by authorized parties. Authentication is a method of confirming the identity of a user or system; this ensures that only authorized users can read or alter data. Nonrepudiation is a concept that maintains that a user cannot deny an action that he or she performed.

2.1 Smart Contract Anatomy

Smart contracts operate on blockchain technology, which acts as a distributed ledger for all present information. Blockchain, in essence, is a decentralized platform on which transactions are executed, recorded, and maintained. To explain, the blockchain itself is not stored on a central location; rather, it is stored on every participating node (computer) in a given network. This quality is what differentiates blockchain from other centralized platforms. There is no single point of failure, nor is there any single target for attack. The blockchain is immutable, and it provides a platform on which permanent items can be stored. One such item is a smart contract.

If blockchain is the platform on which transactions are executed, recorded, and maintained, then smart contracts can be described as the mechanisms by which these transactions are automated. Figure 1 depicts two individuals agreeing on a smart contract. In the figure, the contract is stored on the blockchain, ensuring its immutability an authenticity for both parties regardless of party trust. However, within smart contracts, there are multiple avenues for implementation, each offering varying levels of privacy and security.

2.2 Transaction Models on Blockchain

Above all, smart contracts offer automated transactions with reduced overhead when compared to traditional contracts. These

Figure 1: Anatomy of a smart contract [6]
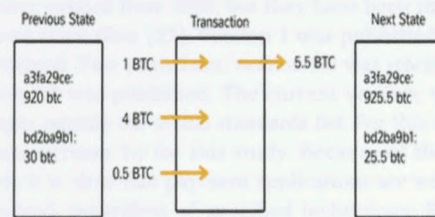
Figure 2: UTXO transaction example [1]

transactions can be carried out in two methods: unspent transaction output or account-based. Both models have seen widespread use in different blockchain technologies, and both have benefits and disadvantages regarding decentralization and privacy.

The Unspent Transaction Output model (hereafter referred to as UTXO) was developed by Satoshi Nakamoto, first seen in the publication for Bitcoin [18]. in 2008. In this work, Satoshi covered a range of topics for implementing blockchain technology, including the UTXO model, proof of work (covered in a later section), and cryptocurrency mining. UTXO works by assigning a unique identifier for each transaction. It should also be noted that this is true for the initial mining of the bitcoin, which returns a determined value. These unique identifiers are the backbone of UTXO, as they are used as inputs and outputs for each transaction that occurs on the blockchain. Figure 2 details UTXO. The users in the transaction are somewhat anonymized, for the accounts used in transactions are not directly linked to personal information. Therefore, UTXO is, by default, primitively private for users on the blockchain. The steps for UTXO are listed below:

- User 1 sends currency. This step is depicted as one arrow and may be interpreted as a single currency or coin sent, but the reality may be multiple transactions of smaller amounts that comprise the required transaction amount (e.g., User 1 may need to send 20 coins for the transaction. He or she may send coins of values 5, 5, and 10 to sum 20. These can be sent, or a single transaction of value 20 can be sent).
- The required transaction amount is sent to User 2 and a unique output value that is different from the input value sent in step 1. This is a critical step in the UTXO model.
- If applicable, User 1 is sent *change* from his or her inputted values. This only applies if User 1 sent more currency than
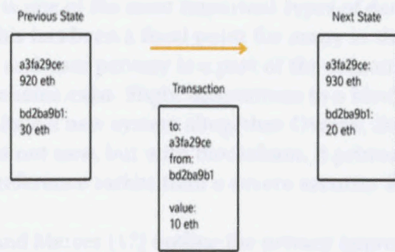
Figure 3: Acount-based model transaction example [1]

required by the transaction. This output is different from the input value(s) (e.g., if User 1 sends 25 currency on a 20-currency transaction, he or she will be refunded 5 currency during this step).

The account-based model resembles what one might see in a traditional banking system. Two users are given accounts, and they are able to conduct transactions, given there is enough money to be sent. This model differs greatly from UTXO, as it seemingly eliminates a portion of the decentralization by forcing a trusted party to maintain accounts on the blockchain. Essentially, there is an established account system that mirrors a digital bank, which limits the privacy of this transaction model. Figure 3 depicts the transaction. The steps can be seen below:

- Account 1 sends currency
- Account 2 receives currency

Each transaction model has benefits and detriments to its design, but each is popular within its own niche. UTXO was the first transaction model, introduced when blockchain was initially proposed [4], and the goal lies in anonymity. UTXO surpasses the account-based model in this sense, for UTXO offers a transaction without basis in accounts. It simply takes the currency much like a vending machine, computing the return value and forwarding the transaction to its intended destination. The downfall here is complexity. With UTXO, transactions rely on more computations to sum the inputs and return *change* if applicable. As a result, there are multiple steps to the transaction that do not exist when using a simpler design. This is where the account-based model succeeds. Account-based systems, which can be seen in many popular blockchains such as Ethereum, utilize a much more standardized method to execute transactions. Not unlike a bank, there exist two parties looking to conduct a transaction. These two parties, much like in modern banking, have accounts wherein their balance is stored. The account-based model utilizes these accounts to verify that there is enough âĂIJmoneyâĂĬ to be sent for the transaction. The benefit here is the simplicity of the model; streamlined by accounts, the transactions rely only on verification of valid funds to execute. However, the anonymity is called into question when accounts can be linked to one another through transactions. Both UTXO and account-based models have found homes in various blockchain technologies, often dependent on the goal of the blockchain.

2.3   Purpose of PA-DSS Standards

The Payment Application Data Security Standard, more commonly referred to as the (PA-DSS), was first published in 2008 as the Payment Application Best Practices (PABS) [23]. It is currently governed by five major global payment brands: American Express, Discover Financial Services, JCB International, MasterCard Worldwide, and Visa Inc. These organizations collaborate to determine the security requirements for third-party payment applications. It should be noted that this standard specifically targets third-party payment applications, for their systems are likely to be distributed to multiple vendors. As such, their system will not be tailored for a single organization; tailored solutions or applications built for a single company are not mandated by this standard. The PA-DSS standards have existed from 2008, but they have been through several iterations since then [23]. Version 1 was published when the PABS was formed. Two years later, version 2.0 was released, and in 2013, version 3.0 was published. The current version, v3.2, holds minor changes outside the actual standards list. For this reason, we choose to use version 3.0 for this study. Because of the scope of this research, it is clear that payment applications are within scope of this standard, regardless of specified technology. Blockchain platforms, in most use cases, can be considered payment systems. Because they are also not typically developed by organizations that may use them, they can also be assumed, in general, to be third-party applications. By definition, they should follow the described standards of PA-DSS. We use these standards as a guideline for best security practices on payment applications, and blockchain platforms fall under the umbrella.

3   RELATED WORK

Blockchain, as with any other digital system, has vulnerabilities and gaps leading to exploitations. Still in its infancy, the technology is constantly evolving to create new and secure ways to operate. Moreover, each blockchain comes with its own set of challenges which depend on the mechanisms they use. For example, Ethereum may face different security concerns than Bitcoin simply because Ethereum chooses to use an account-based transaction model. Many issues, whether severe flaws or privacy preferences have been documented to highlight the evident or perceived shortcomings of each blockchain. They can be separated into three major categories: anonymity of users, privacy of transactions, and software assurance of contracts. Each entails its own set of challenges and solutions, but all are relevant to the comprehensive security of a given blockchain system.

3.1   User security

User security has been a point of discussion on more than blockchain systems for years, and the contention continues onto this technology. Countless debates (and blockchains) are dedicated to proposing the optimal solution to provide users with ample anonymity for transactions on a blockchain. It is important to differentiate user privacy from the other categories, as doing so will help to outline the type of works included in this field. The goal of blockchain, and especially the Unspent Output from Bitcoin Transactions (abbreviated as UTXO) transaction model, is to provide a mechanism for secure transactions between two untrusting parties. As such, user

information is one of the most important types of data that can be disclosed. This has been a focal point for many in the blockchain community, and user privacy is a part of the reason so many different blockchains exist. Slight adaptations to a blockchain sometimes necessitate a new system altogether. Overall, the idea of user anonymity is not new, but with blockchain, it primarily concerns the user's preference rather than a severe security flaw (with exceptions).

Ferrante and Mercer [17] outline the privacy improvements that can be made to the existing Bitcoin platform. The focus of the paper centralizes around the UTXO model and its transaction sources and destinations. Since the basis of UTXO is untrusting, user anonymity is paramount. The researchers detail improvements to the model to prevent the source or destination from knowing who has sent or received the transaction. By doing so, there is no room for de-anonymization. Furthermore, this work overlaps heavily with transaction privacy, wherein multiple solutions are proposed to further increase privacy (these proposals will be discussed in the next section). Conti et al. [7] document blockchain vulnerabilities on its transaction system; however, this work does not mention security and privacy aspects of Bitcoin wallets and user identity. Both works [11] and [5] are the direct sources from which the researchers retrieved their information. Koshy et al. [11] utilized network traffic analysis to heuristically determine the IP addresses that linked with the Bitcoin accounts. This is a severe vulnerability that compromises the anonymity supposed on Bitcoin's platform.

User privacy is essential in a decentralized platform. With the blockchain's intrinsic availability to users, anonymity it is necessary and cannot be overlooked as a luxury. Account privacy works hand-in-hand with transactional security, for each transaction generally links to a particular user or set of users. As smart contracts are more widely considered for adoption, the security and privacy of users and transactions cannot be of question. It must be at least as reliable as modern centralized platforms. Even though these works support progress in smart contract platforms, there is still more work to be done.

3.2 Transaction security

Transaction security is undeniably important when considering overall security of an application or platform. Data in transit must be protected against both passive, active eavesdropping, and unintended alteration. When dealing with smart contracts, users still face similar issues of traditional transaction systems. However, blockchain technology has allowed certain growth from the perspective of absolute privacy during transactions. Such advancements are detailed, and their benefits and drawbacks are outlined.

Ferrante and Mercer describe in their research the benefits of using the UTXO transaction model to optimize transaction privacy, especially between distrusting parties [?]. The intrinsic security features of the UTXO model, which include supposedly unidentifiable accounts linked to unique transactions, by default outstrip the privacy capabilities of the standard account-based model. However, there have been studies that exploit certain aspects of the model to track users. Due to possible vulnerabilities of the UTXO base model, Koshy et al. [11] offer solutions to boost user privacy and

anonymity. The two recommended controls are linkable ring signatures and stealth addresses. Linkable ring signatures allow users to verify that they are a part of a group without revealing exactly which user they are. This group may be a set of public keys, where a user may have to use his or her private key to authenticate. Stealth addresses ensure that a given user's identity (address) is indistinguishable from random, and they also guarantee that only a user can conduct transactions through that account. These modifications to the UTXO model raise the level of privacy for transactions on the blockchain, eliminating the possibility of transaction identification or spoofing. The work done by Sompolinsky and Zohar regarding a GHOST protocol for Bitcoin's transactions centers around mitigating double-spending attacks on the blockchain [23]. In standard environments, malicious users may have the opportunity to access and spend the same currency two or more times before the blockchain realizes the error; this defines a double-spending attack. With the GHOST protocol, the researchers theorize a system where lightweight processing allows the high-rate transactions to execute without this vulnerability. This protocol not only enhances the security features while also presenting interesting follow-up inquiries regarding the scalability of a blockchain platform. In essence, to process transactions on a large scale, the GHOST protocol or an equivalent is necessary to maintain security when scaling.

In the work done by the researchers Shunli et al, proposals for account-based transaction models are used to boost privacy [13]. By implementing homomorphic encryption, a method where encrypted data can be used to perform operations without decrypting the data, information can be altered by authorized users. This eliminates the need to decrypt information to utilize it, which is standard practice in most payment systems. Another feature mentioned was a zero-knowledge (ZK) approach, which relies on hiding information from users unless absolutely necessary to the transaction. These solutions are proposed on the account-based model, which may boost the intrinsic privacy to match that of the UTXO model.

The work done by Andrychowicz et al. [2] pertains to *honest participants* and transactional assurance. While not directly tied to privacy, assurance is still an integral component of security, and these researchers designed a trustless protocol to prevent fraud and exploitation. On a similar note, Zhang et al. [26] proposes an authenticated data feed for smart contracts. This work doesn't refer to smart contract security itself, but rather the data that would inevitably feed into a smart contract platform. The authenticity of this data is just as vital as the data generated on the blockchain itself, and it should be considered relevant to the overall maturity of smart contract security. Gray and Hadju have also contributed to the smart contract security realm with a practical implementation utilizing *Cryptlets* [14–16]. In their work, they contextualize the need for an optimized, trustless mode for transactions and analyze the Ethereum platform, wherein they describe the optimization woes that it has faced in the past. They offer a platform optimization using Cryptlets, a seemingly third-party-reminiscent repository which stores the logic of a smart contract to be executed by a node on the blockchain. Based on a semi-trust or fully-trust model, Cryptlets offer an optimized platform that does not rely on each node to execute the logic of a smart contract, providing a reduced computational overhead for the blockchain as a whole. This is known outside of blockchain technology as cyber-offloading.

Cyber-offloading, particularly on a blockchain, is dangerous to the integrity of decentralized systems, as it relies on a trusted third-party to maintain integrity parallel to the blockchain. However, in a more pragmatic sense, offloading is a reasonable risk for business implementations of smart contract technology. There is incentive for organizations to use a trust or semi-trust model, considering there is some liability and governance surrounding the execution of smart contracts. This may be seen as a transitional middle-ground between centralized and decentralized platforms. In 2016, Hawk was created as a smart contract platform, which boosted security measures for code design on that platform [10]. This adaptive platform automates cryptographic protocols, such as encryption, for smart contracts. Essentially, developers do not need to manually code the cryptography on the contracts as this process is handled by the compiler. This protocol remediates the traditional lack of privacy on most popular decentralized platforms, and it does so without burdensome interference for the users and developers. Since this platform was designed with transactional privacy in mind, it may be difficult to port this technology or protocol to other platforms without a complete redesign. Regardless, the success of this automated cryptographic protocol shows that security can be implemented intuitively and automatically, given the right design.

Transactional security is a focal point for many different works on smart contracts, and the progression towards optimal privacy is undeniable. Many of the researchers referenced, as well as many other smart contract developers, have designed unique, separate platforms for each protocol change. The challenge, at this time, is not solving the issue of transactional privacy on a decentralized network, but rather implementing the solutions already discovered, governing the maintenance of said solutions, and ensuring there are as few vulnerabilities as possible.

3.3 Software assurance of contracts

Many researchers have also chosen to target the smart contract code itself, finding various shortcomings with regards to software assurance. These issues are outside the scope of this paper, but the impact of software assurance is evident, and it is necessary to acknowledge its progress. Bartoletti and Pompianu, in their research, analyze the popular smart contract platforms, Ethereum and Bitcoin, for their security implementation [4]. Their focus is on the available code for smart contracts on each platform, analyzing it for patterns in design. The purpose of this paper is to find a recurring issue in the design of smart contracts such that it may be remediated. The empirical process the researchers follow is highly adoptable and may follow the structure of organizations that review smart contract code.

Delmolino et al have also seen the value in software assurance for smart contracts, like any other piece of code [8]. By analyzing public-domain smart contracts, they noticed patterns and recurring issues in the programming practices. As a result, the researchers created open-source platforms to teach programming for smart contracts. This paper largely focuses on the education of safe programming practices, but the core components of smart contracts and their code are programmers. Therefore, the training of such practices is necessary for proper progression in smart contract security.

Similar to the work of Delmolino et al, Luu et al documented their findings of smart contract bugs on the Ethereum platform [12]. Their work is limited to a single platform, albeit the largest smart contract blockchain used at the moment. By describing the pitfalls of smart contracts on Ethereum, they both contextualize the impact of these bugs and find common threads between them. Smart contract bugs often result in lost or stolen cryptocurrencies, sometimes totaling the equivalent of millions of dollars, and this paper describes how each attack or bug executed. Furthermore, the common characteristics of the bugs allow a conclusion to be made around the specific programming practices that lead to each vulnerability.

Buterin's describes Ethereum bugs that have the same principal practices that lead to bugs on the smart contract platform [25]. Bugs like Transaction Ordering Dependence, timestamp dependence, mishandled exceptions, and many others, are pervasive throughout smart contract code. Still in its infancy, smart contract software assurance's state can be seen through the lens of the bugs documented so far. Blockchain and smart contract bugs have stumbled into infamy with their costly mistakes, and each vulnerability seems to uncover more. In a blog post written on Ethereum's website, a list of the most popular bugs can be found with a description of their causes and effects [3]. This work serves as another indication that software assurance is at the forefront of most people's minds when discussing smart contract security.

From the research compiled on software assurance, it is clear that, while there has been significant progress to remediate and prevent bugs from existing, there is still much work to be done. Specifically, the shortcomings of non-Ethereum bug documentation, along with cross-platform bug documentation, limit the progress of new smart contract platforms from prospering. Without formal code review, it is difficult to determine the strength of smart contracts. Many platforms have not had the attention that Ethereum has; as a result, there may be undiscovered bugs present on their platforms. Furthermore, software assurance has received the bulk of focus from the community as a whole, leaving protocol-level security with fewer resources. The community of blockchain and smart contracts has developed a handful of useful protocols, practices, and platforms that boost security. Nevertheless, there is a clear ceiling on smart contracts that must be conquered before adoption in enterprise business environments.

3.4 Limitations of current blockchain platforms

With major popularity in the community, blockchains and smart contracts have been subject to both praise and scrutiny. Even with its revolutionary design to eliminate many of the issues centralized systems face, there are still many shortcomings that must be addressed before the technology is deemed mature by security standards. Aside from the aforementioned security concerns and research, there are still some remaining limitations of smart contract platforms.

Some of the most popular platforms, namely Bitcoin and Ethereum, have succeeded both due to their first-to-market status and simplicity in execution. However, they both fall victim to an issue that serves as a substantial roadblock to large-scale adoption: space on disk. Ethereum, as of 2017, was documented at over 350GB,

with Bitcoin being a very comparable size. Simply put, this spatial requirement is not sustainable for widespread adoption, which inevitably involves mobile devices. There is a balance to be struck between complete access to the blockchain and computational viability on mobile. So far, there are only a handful of platforms, none of them as popular as Ethereum or Bitcoin, that attempt to solve this issue.

Due to the communal nature of blockchain as a whole, smart contracts suffer from the same detriment: lack of standardization. With everyone's hand in the proverbial cookie jar, there is little room to prevent a myriad of unmanageable smart contracts and unsafe protocols. Only recently has there been more effort in the space of academic and professional research to expand on the security controls of smart contract platforms. More work is needed here to advance the viability of smart contracts in enterprise environments. Security and privacy concerns will always be forthcoming and newfound, but the baseline by which smart contracts are standardized must be developed to ensure the proper measures are in place to protect private information.

From the related work, it is clear that the community surrounding blockchain and smart contract technology is very focused on user and transaction privacy. Based on the research already conducted, we find a few proposed solutions to boost user security based on disallowing deanonymization of user addresses, using stealth addresses, and implementing linkable ring signatures. Furthermore, proposed solutions for transaction security show promise as well. The GHOST protocol, along with the ZK approach with homomorphic encryption, both show substantial results in boosting the overall security of transactions on a blockchain. Integrity maintenance on these platforms is paramount, so work done to boost transaction assurance through authenticated data feeds is undoubtedly valuable to a security-focused blockchain. We begin to notice room for concern when researchers drift away from a zero-trust approach for blockchain systems. With Cryptlet's cyber-offloading architecture, maintaining integrity for the centralized storage point is the weak link in the structure. Moving toward a semi-trust or full-trust model can only work with proper governance over the central unit.

With all of the work done with respect to blockchain security, especially the research that uncovered identity linking with supposedly-anonymous pseudonyms on Bitcoin [7], it is interesting to note that no researcher has acknowledged the security and privacy of the ledger itself. Bitcoin, and a number of other platforms, have no record of protecting the transaction data logged on the ledger from viewing. With the de-anonymization of user IDs and addresses, it is a point of concern to keep in mind when discussing the comprehensive security of blockchain platforms.

4 METHODOLOGY

To adapt a set of standards to match the nontraditional mechanisms of blockchain platforms, it is necessary to first establish the baseline standards by which we compare blockchain and centralized platforms. The Payment Application Data Security Standard [22][22] defines the set of guidelines for traditional transaction applications. We analyze each guideline and highlight the shortcomings with respect to their applicability on decentralized systems. Based on

its ability to be directly applicable to decentralized platforms, we categorize each guideline as either *Fully Applicable*, *Partially Applicable*, or *Not Applicable*. If a guideline is deemed *Fully Applicable*, it is able to be applied without alteration to a decentralized system. If a guideline is deemed *Partially Applicable*, it is able to be applied to decentralized platforms with modifications. If a guideline is deemed *Not Applicable*, it cannot be applied to a decentralized system without major alteration (in these cases, alteration would essentially create a new standard).

Rationalizations for the categorization of guidelines are given in the full PA-DSS analysis table. Each guideline was categorized for a specific reason, and the rationalization field dictates why each decision was made. This field may also contain supplemental considerations for *Fully Applicable* guidelines, wherein suggestions are proposed to incorporate decentralized platforms into the scope of the standard. That is, the *Fully Applicable* guidelines may be fully applicable to traditional payment systems, and they may also be applicable to decentralized payment systems, but supplemental information is necessary (and proposed) to cover the scope of decentralized systems.

Once the analysis of the PA-DSS guidelines are complete, we adapt the guidelines to create a fully-enveloping set of standards for centralized and decentralized platforms. It is important to note that this set of standards includes guidelines for both types of platforms, not one or the other. By doing this, we create a comprehensive set of standards that applies to modern payment applications and their underlying technology. The new set of standards is described in a table with several categories: Current Standard, Applicability to blockchain Platforms, Rationalization, and New Standard. These categories serve to identify weaknesses in the current PA-DSS standards as they apply to blockchain. Furthermore, we revise standards, if necessary, to form a comprehensive standard for both types of platforms. After developing the revised PA-DSS standards, we analyze two blockchain and smart contract platforms: QTUM and Ethereum. Both platforms are scrutinized for their adherence to the newly defined standards. The platforms overall security maturity is also taken into consideration, and there is an opportunity for the security measures of the blockchain platforms to warrant the revision of the new standards. The manual analysis of the QTUM and Ethereum platforms includes a high-level review of the platform as it appears to a standard user, a code-level review of user profiles and transaction execution, and a reporting of key security findings. The review of each platform determines whether recent security features in the blockchain community have been adopted; the review also compares the blockchain platforms with the security mechanisms of modern traditional payment applications.

There are many standards developed by the Payment Card Industry to govern the way transaction information is stored, transmitted, and used, but the PA-DSS standard applies directly to the context of new payment systems, such as blockchain. By analyzing this standard, we reimagine transaction data protection in a context where current standards do not match current technology. By creating a structured approach to analyzing the blockchain platforms against the revised PA-DSS standards, we ensure that each platform is adequately analyzed for its implemented security mechanisms.

Ultimately, we achieve a sound result in adapting traditional payment standards to a nontraditional technology that is gaining more popularity by the day.

5   RESULTS

5.1   Applying PA-DSS to blockchain platforms

Financial institutions use Payment Card Industry (PCI) standards to ensure the secure development of their technology environments. Most industries that incorporate technology into the business model have equivalent standards, but the payment card industry heavily relies on standards to determine the security of their technology. The Payment Application Data Security Standard (PA-DSS) focuses on financial technology applications; that is, the software that processes, stores, or otherwise encounters sensitive payment data will follow the standards set by the PA-DSS [22]. The full table of the critiqued and adapted PA-DSS standards are too long to include in this work, but have been made available for reference [9].

The Payment Application Data Security Standard, more commonly referred to as the (PA-DSS), was first published in 2008 as the Payment Application Best Practices (PABS) [21]. It is currently governed by five major global payment brands: American Express, Discover Financial Services, JCB International, MasterCard Worldwide, and Visa Inc. These organizations collaborate to determine the security requirements for third-party payment applications. It should be noted that this standard specifically targets third-party payment applications, for their systems are likely to be distributed to multiple vendors. As such, their system will not be tailored for a single organization; tailored solutions or applications built for a single company are not mandated by this standard. The PA-DSS standards have existed from 2008, but they have been revised through several iterations since then. Version 1 was published when the PABS was formed. Two years later, version 2.0 was released, and in 2013, version 3.0 was published. The current version, v3.2, holds minor changes outside the actual standards list. For this reason, we chose version 3.0 for this study [21].

Because of the scope of this research, it is clear that payment applications are within scope of this standard, regardless of specified technology. Blockchain platforms, in most use cases, can be considered payment systems because they are used to send and receive currency. Because they are also not typically developed by organizations that may use them, they can also be assumed, in general, to be third-party applications. By definition, they should follow the described standards of PA-DSS. We use these standards as a guideline for best security practices on payment applications, and blockchain platforms fall under the umbrella. The PA-DSS is designed for modern, centralized applications, and decentralized platforms have contrasting designs that make the standard only partially useable. Data access, data confidentiality, and the very definition of what sensitive information is all contributing factors to the necessity for revisions to the PA-DSS to include blockchain technology. Both centralized and decentralized systems support transactions, so widespread adoption also requires proper security standards to prevent exploitation.

Upon analysis of the PA-DSS standards, we find that, of the forty (40) standards reviewed, several of the standards do not allow proper adaptation to blockchain platforms. The breakdown of applicability

Table 1: Summary of the applicability of PA-DSS standards in blockchain platforms

Fully applicable	Partially applicable	Not applicable
33	7	3

can be seen in Table 1. While the majority of the standards are still applicable to blockchain platforms, there are significant gaps in the scope of the PA-DSS standards.

A key area for revision is the classification of sensitive data and how to properly manage the data. Five of the standards marked *Partially Applicable* deal with the mandatory secure storage and transmission of sensitive information. With traditional payment systems, cardholder data is clearly defined, and the PA-DSS standards match the definitions. The current standards explicitly list fields that must be securely handled, such as credit-card number (CCN), primary account number (PAN), and PIN numbers. However, due to the anatomy of blockchain transactions, with user IDs as the main form of identification and direction for the transaction, user IDs have a much larger role than most cardholder data fields. User IDs are essentially used as source and destination placeholders for the transaction, and the same is true for the data stored on the ledger. As such, the standards must be revised to account for this difference in core architecture of the payment platform.

Data access and platform logging are two more categorical issues with the current standards. On traditional platforms, data access is, in most situations, restricted only to a select number of business employees; on traditional platforms, this is understandable considering this data is vulnerable to manipulation and/or destruction. Logging is typically implemented in the same scenario to ensure that the organization knows who is accessing the data at any time. However, there is a fundamental change in data access on blockchain platforms: the data is available for all participants to view. As a result, data access cannot be restricted, lest the integrity of the blockchain be compromised. If data access was limited on blockchain platforms, then the trust model that blockchain technology is built on would be destroyed altogether. Access logging is also difficult logistically (and arguably needless) on blockchain platforms. If all participating nodes have access to the ledger, then it is safe to assume every participant can or has viewed the data, secured or not. The three standards associated with data access and logging, consequently, are revised to match the information dynamic present on blockchains.

Vulnerability identification and remediation, including the process of patching, are specified in the PA-DSS standards, but they understandably lack the exceptions necessary for an immutable blockchain. For traditional payment systems, vulnerability scanning and remediation is an arduous and ongoing challenge. Patches to code and platforms are difficult on traditional payment systems, but blockchain platforms impose an entirely unique and complex roadblock to remediation. Blockchain platforms, inherently unchangeable, do not allow the same process of vulnerability remediation. Smart contracts are stored on the blockchain, and the only solution to remediate a vulnerable smart contract is to disallow it from further use and create a new (remediated) code to store on

the blockchain. Moreover, blockchain platforms as a whole have been known to have vulnerabilities; to remediate weaknesses to the entire platform, the blockchain must be forked to create a new, but related, platform to be used. The current PA-DSS standards do not account for such processes, and revisions are necessary to accommodate blockchain systems. Based on the above observations and revisions to the PA-DSS standards, we compiled a list of controls to look for when analyzing the case-study blockchain platforms. Overall, the PA-DSS standards are excellent measures for protecting traditional payment platforms, but they fall short in several aspects concerning blockchain platforms. After revisions, the standards are much more comprehensive and are ready to be applied to the case studies.

5.2   Case studies

The two blockchain platforms, QTUM and Ethereum, are popular and well-developed systems that have been considered for business adoption. These platforms signify advancements from the early stages of blockchain platforms, such as Bitcoin, for they offer supposed enhancements to the weaknesses of Bitcoin-like platforms. Based on the analysis of each platform, we find fundamental shortcomings for both platforms when considering business adoption in the US financial industry. Based on the adapted PA-DSS standards, we identify key points of security weakness that lead to sub-standard platforms.

The Ethereum platform is one of the most popular blockchains with a key advancement from early cryptocurrency platforms: smart contract support. With this advancement, Ethereum establishes its potential for financial technology adoption. Upon analysis, we find that Ethereum has a fundamental design, non-protected transaction information, that undermines the PA-DSS standards. User IDs are publicly available on the blockchain ledger, along with currency transaction data. From this information, it is trivial to link all transactions to a single pseudonym, and Ethereum has a dedicated webpage that allows anyone to see all transactions made by a given user. Furthermore, Ethereum does not adopt most of the privacy-boosting mechanisms that more recent blockchains have developed. As a result, there is little obfuscation between two users conducting a transaction.

The implications of Ethereum's security measures matters most when considering real-world business adoption. A common use case for smart contracts in enterprise is as follows: two businesses automate a contractual subscription to goods or services. In this use case, businesses must know who they are sending money to, and vice versa. Without proper data security, namely on the transactional information, it is possible to make inferences towards inter-business transactions from the ledger alone. The reality is that complete anonymity is not feasible for business adoption; therefore, proper data security is necessary to obfuscate the representation of the transaction on the ledger. This can be achieved via the revised PA-DSS standards.

The PA-DSS standards, when used to analyze Ethereum's platform, show a significant oversight regarding data security. While the oversight may be understandable considering business or enterprise adoption was not foreseen by Ethereum, it does result in the platform's weakness towards large-scale financial adoption. To

Table 2: Summary of violations of the adapted PA-DSS standards in QTUM and Ethereum

Technology	Total violations	Violations
Ethereum	5	2.1, 2.2, 2.3, 11.2, 12.2
QTUM	5	2.1, 2.2, 2.3, 11.2, 12.2

remediate, Ethereum would have to fundamentally alter the way data is stored on the blockchain, encrypting sensitive data before the ledger stores it. As it stands, Ethereum does not meet the revised PA-DSS standards.

QTUM is designed specifically to be a platform ready for lightweight, versatile business deployment, according to its white papers. Its main features, in addition to the blockchain itself, include a lite wallet for mobile use and a transaction-model abstraction layer to allow transactions between UTXO and account-based platforms (e.g. Bitcoin and Ethereum). The business viability seems to be strong, but the PA-DSS standards show that the platform faces similar issues to Ethereum. Static user IDs and transaction data are stored on the ledger without encryption. QTUM, too, has a webpage that allows users to search for specific transactions and list all of a specific user's transactions. However, with QTUM's added functionality of lite wallets, there is more to analyze against the standards. After review, the lite wallet features a robust security system to protect the account and transaction data when it is on mobile devices. Furthermore, we find no weaknesses in the data transmission between lite wallets and the core wallet of the blockchain.

The implications of QTUM are almost identical to the repercussions of Ethereum's platform given a real-world scenario. User data stored on the ledger is not private, and inferences can be made based on repeated transactions on the system. Moreover, peer-to-peer communications do not obfuscate the source or destination addresses, meaning anonymity is only partially achieved. No modern security measures of community-developed blockchains have been adopted to ensure proper privacy between users or transactions. However, the lite wallets show some attention to security, with standard security measures in place to protect local account information when stored. The PA-DSS standards uncover significant issues concerning both QTUM and Ethereum. From the table, we see the exact standards that each platform violated from the adapted PA-DSS standards, along with a total count for the violations. Each platform had 5 violations, all directly related to data security and handling. Table 2 shows a summary of the violations for each platform. Data security is a vital portion of overall payment application security, and both platforms fall short in this regard. More work is needed to ensure that businesses are able to adopt blockchain platforms as payment systems. In their current state, actual usability and transactional functionality is not the challenge; instead, the platforms face the challenge of adhering to financial technology's security standards.

6   CONCLUSION AND FUTURE WORK

Blockchain technology and smart contracts are peaking in popularity, and many businesses are considering the adoption of such

technology. Smart contracts allow the automation of many tasks on a blockchain, including the automation of payments themselves. Smart contracts, consequently, have garnered attention from businesses for reducing the overhead of traditional contracts. However, this new technology has many shortcomings, some of which are not easily remediated.

The communal nature of blockchain development leads to a lack of accountability in the products. There is no governance or standardization of the platform development. With businesses seeking out blockchain platforms, there is a need for a structured methodology to fully analyze the capabilities and security of these new, decentralized payment systems. Considering the financial technology industry as a gold-standard for rigorous auditing, we adopt the Payment Application Data Security Standards (PA-DSS) to apply to blockchain platforms. By revising the standards to meet the requirements of both parties, blockchain and financial technology organizations, we solidify the methodology used to critique modern smart contract platforms.

Through two case studies, QTUM and Ethereum, we report key weaknesses in the foundations of the blockchain platforms. Data security is the main issue, for neither system offers adequate user privacy concerning transaction information. Transaction data are openly available through the ledger, and privacy as a whole is compromised as a result. There are critical alterations, including proper data protection, to be made to each platform for PA-DSS compliance. Fundamentally, smart contract platforms offer tremendous business potential, but the lack of security governance is overwhelming. With sensitive transaction information present on these immutable platforms, data security is essential and should not be overlooked. Even with the modern security-focusing blockchains, there has been little effort to standardize the solutions and provide a template for a comprehensive blockchain security solution. This work sets the precedence for continuing security practices into decentralized payment systems.

From the results of the case studies and the adapted PA-DSS standards, there is a clear set of next steps for contributors of the Ethereum and QTUM platforms. Security changes can be made to the existing platforms, or a new platform can be developed to both implement a smart contract transaction platform and securely store data. Developing such a system would be time-consuming, but it would provide the necessary security measures to ensure proper data handling at the transaction level. Moreover, analyzing more smart contract platforms would prove useful in creating a survey of common security practice (or malpractice) in the blockchain community. The Payment Card Industry is a well-known, established field where security standards are heavily enforced, but there are many other technology industries that would benefit from the same type of standards adaptation. The Healthcare industry, for example, has several data privacy laws to follow with respect to health information. There have been discussions and research surrounding the adoption of blockchain models to healthcare, but the extent at which the security measures have been thought out is unclear. There is, at the least, room for consolidation of information.

Data privacy and security is the main focus of this work, but there are other aspects that lie outside the scope of this work. Blockchain solves one of data protection's most notable issues: data integrity. That is, data on the blockchain is immutable. There have been many

experiments and new blockchains that propose new methods to enhance the blockchains speed or usability. However, a comprehensive analysis of such innovations may prove helpful in determining the best approach for designing payment systems for particular purposes or audiences.

REFERENCES

[1] Alyssa Hertig 2018. How Ethereum Works. (2018). Retrieved January 21, 2019 from https://www.coindesk.com/information/how-ethereum-works/

[2] Marcin Andrychowicz, Stefan Dziembowski, Daniel Malinowski, and Lukasz Mazurek. 2014. Secure multiparty computations on bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*. IEEE, 443–458.

[3] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. 2017. A survey of attacks on ethereum smart contracts (sok). In *Principles of Security and Trust*. Springer, 164–186.

[4] Massimo Bartoletti and Livio Pompianu. 2017. An empirical analysis of smart contracts: platforms, applications, and design patterns. In *International Conference on Financial Cryptography and Data Security*. Springer, 494–509.

[5] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. 2014. Deanonymisation of clients in Bitcoin P2P network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 15–29.

[6] Blockgeeks 2018. Smart Contracts: The Blockchain Technology That Will Replace Lawyers. (2018). Retrieved Jan 21, 2018 from https://blockgeeks.com/guides/smart-contracts/

[7] Mauro Conti, Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. 2018. A survey on security and privacy issues of bitcoin. *IEEE Communications Surveys & Tutorials* 20, 4 (2018), 3416–3452.

[8] Kevin Delmolino, Mitchell Arnett, Ahmed Kosba, Andrew Miller, and Elaine Shi. 2016. Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In *International Conference on Financial Cryptography and Data Security*. Springer, 79–94.

[9] Gabriel Bello and Alfredo J. Perez 2018. Adapted PA-DSS Standards. (2018). Retrieved Dec 1, 2018 from https://tinyurl.com/yabykwf8

[10] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. 2016. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE symposium on security and privacy (SP)*. IEEE, 839–858.

[11] Philip Koshy, Diana Koshy, and Patrick McDaniel. 2014. An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security*. Springer, 469–485.

[12] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. 2016. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 254–269.

[13] Shunli Ma, Yi Deng, Debiao He, Jiang Zhang, and Xiang Xie. 2017. An efficient NIZK scheme for privacy-preserving transactions over account-model blockchain. *IACR Cryptol. ePrint Arch., Tech. Rep* (2017), 1239.

[14] Marley Gray and Craig Hajduk 2017. Anatomy of a Smart Contract. (2017). Retrieved Dec 1, 2018 from https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/AnatomyofASmartContract.md

[15] Marley Gray and Craig Hajduk 2017. Anatomy of a Smart Contract 2. (2017). Retrieved Dec 1, 2018 from https://azure.microsoft.com/en-us/blog/scanatomy-2

[16] Marley Gray and Craig Hajduk 2017. Cryptlets Deep Dive. (2017). Retrieved Dec 1, 2018 from https://github.com/Azure/azure-blockchain-projects/blob/master/bletchley/CryptletsDeepDive.md

[17] Matthew Di Ferrante and Rebekah Mercer 2017. Towards Blockchain Transaction Privacy. (2017). Retrieved Dec 1, 2018 from https://www.clearmatics.com/wp-content/uploads/2017/06/IEEE-Presentation.pdf

[18] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[19] Nick Szabo 1997. The Idea of Smart Contracts. (1997). Retrieved Dec 1, 2018 from http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html

[20] Nick Szabo 2002. A Formal Language for Analyzing Contracts. (2002). Retrieved Dec 1, 2018 from http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/contractlanguage.html

[21] PCI Security Standards Council 2008. Payment Application Data Security Standard: Frequently Asked Questions. (2008). https://www.pcisecuritystandards.org/pdfs/pci_pa-dss_faqs.pdf

[22] PCI Security Standards Council 2013. Payment Card Industry (PCI) Payment Application Data Security Standard-Requirements and Security Assessment Procedures version 3.0. (2013). https://www.pcisecuritystandards.org/minisite/en/docs/PA-DSS_v3.pdf

[23] Yonatan Sompolinsky and Aviv Zohar. 2015. Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security*. Springer, 507–527.

[24] U.S. Small Business Administration 2016. Contract Law - How to Create a Legally Binding Contract. (2016). Retrieved Dec 1, 2018 from https://www.sba.gov/blogs/contract-law-how-create-legally-binding-contract

[25] Vitalik Buterin 2016. Thinking About Smart Contract Security. (2016). Retrieved Dec 1, 2018 from https://blog.ethereum.org/2016/06/19/thinking-smart-contract-security/

[26] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. 2016. Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM, 270–282.

Standard Number	Current Standard	Applicability to Blockchain Platforms	Rationalization	New Standard
1.1.1	After authorization, do not store the full contents of any track from the magnetic stripe (that is on the back of a card, in a chip or elsewhere).	FULLY APPLICABLE	Depending on the transaction method used on the decentralized application, card data may still need protection. Magnetic stripes, CV codes, or other sensitive card data will be protected if present. If no card data is used (e.g. non-card-based transaction system such as UTXO), this standard is no longer applicable.	After authorization, do not store the full contents of any track from the magnetic stripe (that is on the back of a card, in a chip or elsewhere).
1.1.2	After authorization, do not store the card-validation value or code (three-digit or four-digit number printed on the front or back of a payment card) used to verify card-not-present transactions.	FULLY APPLICABLE	Depending on the transaction method used on the decentralized application, card data may still need protection. Magnetic stripes, CV codes, or other sensitive card data will be protected if present. If no card data is used (e.g. non-card-based transaction system such as UTXO), this standard is no longer applicable.	After authorization, do not store the card-validation value or code (three-digit or four-digit number printed on the front or back of a payment card) used to verify card-not-present transactions.
1.1.3	After authorization, do not store the personal identification number (PIN) or the encrypted PIN block.	FULLY APPLICABLE	Depending on the transaction method used on the decentralized application, card data may still need protection. Magnetic stripes, CV codes, or other sensitive card data will be protected if present. If no card data is used (e.g. non-card-based transaction system such as UTXO), this standard is no longer applicable.	After authorization, do not store the personal identification number (PIN) or the encrypted PIN block.
1.1.4	Securely delete any magnetic stripe data, card validation values or codes, and PINs or PIN block data stored by previous versions of the payment application, in accordance with industry-accepted standards for secure deletion, as defined, for example by the list of approved products maintained by the National Security Agency, or by other State or	FULLY APPLICABLE	Depending on the transaction method used on the decentralized application, card data may still need to be securely deleted.	Securely delete any magnetic stripe data, card validation values or codes, and PINs or PIN block data stored by previous versions of the payment application, in accordance with industry-accepted standards for secure deletion, as defined, for example by the list of approved products maintained by the National Security Agency, or by other State or

1. DO NOT RETAIN FULL TRACK DATA, CARD VERIFICATION CODE OR VALUE (CAV2, CID, CVC2, CVV2), OR PIN BLOCK DATA

	National standards or regulations.			National standards or regulations.
1.1.5	Securely delete any sensitive authentication data (pre-authorization data) used for debugging or troubleshooting purposes from log files, debugging files, and other data sources received from customers, to ensure that magnetic stripe data, card validation codes or values, and PINS or PIN block data are not stored on software vendor systems.	FULLY APPLICABLE	Depending on the transaction method used on the decentralized application, authentication data may still need to be securely deleted.	Securely delete any sensitive authentication data (pre-authorization data) used for debugging or troubleshooting purposes from log files, debugging files, and other data sources received from customers, to ensure that magnetic stripe data, card validation codes or values, and PINS or PIN block data are not stored on software vendor systems.
2.1	Software vendor must provide guidance to customers regarding purging of cardholder data after expiration of customer-defined retention period.	PARTIALLY APPLICABLE	Decentralized applications [smart contracts] differ from the decentralized platform [e.g. Ethereum] itself. Data retention standards apply normally to applications that may store cardholder data. They do not apply to the data stored on the blockchain itself, as the data should exist immutably. We know the data stored on the blockchain for cryptocurrency accounts; we can only assume the type of data stored on the blockchain with traditional transactions. That data, however, should be stored securely.	Software vendor must provide guidance to customers regarding purging of cardholder data after expiration of customer-defined retention period. Retention period of customer data is dependent on transactional method. Blockchain ledgers will securely store transactional data permanently regardless of transaction method.
2.2	Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed).	PARTIALLY APPLICABLE	This standard is specific to cardholder data. When cardholder data is used, this standard should be followed. CCN is similar to account ID on decentralized platforms. There is an equivalence to be drawn between the two types of data, and both should be equally secured.	Mask PAN when displayed (the first six and last four digits are the maximum number of digits to be displayed). On blockchain platforms, UTXO and Account-Based transaction models require user ID for transactions. User ID or any information determining a transaction's source or destination should be masked when displayed.
2.3	Render PAN, at a minimum, unreadable anywhere it is stored.	PARTIALLY APPLICABLE	The equivalent to PAN in a decentralized platform must be properly stored. PAN and user ID can be viewed equivalently, and they should be secured equally.	Render PAN, at a minimum, unreadable anywhere it is stored. If on a blockchain platform with blockchain-specific transaction models, render user ID unreadable by unauthorized users when stored.
2.4	If disk encryption is used (rather than file- or column-	FULLY APPLICABLE	This standard is based in a central management	If disk encryption is used (rather than file- or column-

2. PROTECT STORED CARDHOLDER DATA

		level database encryption), logical access must be managed independently of native operating system access control mechanisms.		principle. There is no change for a central management application. However, the decentralized platform may not follow the exact same principles of what data to encrypt.	level database encryption), logical access must be managed independently of native operating system access control mechanisms.
	2.5	Payment application must protect encryption keys used for encryption of cardholder data against disclosure and misuse.	FULLY APPLICABLE	Wallet information may also warrant encryption during transactions.	Payment application must protect encryption keys used for encryption of cardholder data against disclosure and misuse.
	2.6	Payment application must implement key management processes and procedures for keys used for encryption of cardholder data.	FULLY APPLICABLE	Any information deemed necessary to encrypt must have equivalent, secure key management protocols.	Payment application must implement key management processes and procedures for keys used for encryption of cardholder data.
	2.7	Securely delete any cryptographic key material or cryptogram stored by previous versions of the payment application, in accordance with industry-accepted standards for secure deletion.	FULLY APPLICABLE	Any information deemed necessary to encrypt must have equivalent, secure key management protocols.	Securely delete any cryptographic key material or cryptogram stored by previous versions of the payment application, in accordance with industry-accepted standards for secure deletion.
3. PROVIDE SECURE AUTHENTICATION FEATURES	3.1	The "out of the box" installation of the payment application in place at the completion of the installation process, must facilitate use of unique usernames and secure authentication for all administrative access and for all access to cardholder data.	PARTIALLY APPLICABLE	Smart contracts placed on the blockchain are immutable unless proper access is obtained to allow destruction of the contract. This access should be restricted to authorized users. Any repository storing cardholder data (or equivalent sensitive information) should follow the standards already established unless on the blockchain itself. Access to protected data on the blockchain should also be managed.	The "out of the box" installation of the payment application in place at the completion of the installation process, must facilitate use of unique usernames and secure authentication for all privileged and administrative access and for all access to cardholder data.
	3.2	Access to PCs, servers, and databases with payment applications must require a unique username and secure authentication.	FULLY APPLICABLE	The core principles of secure login and authentication do not change for decentralized applications or platforms.	Access to PCs, servers, and databases with payment applications must require a unique username and secure authentication.
	3.3	Encrypt payment application passwords during transmission and storage, using strong cryptography based on approved standards.	FULLY APPLICABLE	The core principles of secure login and authentication do not change for decentralized applications or platforms.	Encrypt payment application passwords during transmission and storage, using strong cryptography based on approved standards.
4. LOG PAYMENT APPLICATION	4.1	At the completion of the installation process, the "out of the box" default installation of the payment application must log all user access.	NOT APPLICABLE	The nature of the blockchain ledger makes it unfeasible to logistically log individual events of access to the ledger.	At the completion of the installation process, the "out of the box" default installation of the payment application must log all user access. On a blockchain platform, user access to the ledger will not be logged, as it is available to all blockchain nodes.

	4.2	Payment application must implement an automated audit trail to track and monitor access.	NOT APPLICABLE	The nature of the blockchain ledger makes it unfeasible to logistically log or audit individual events of access to the ledger.	Payment application must implement an automated audit trail to track and monitor access. On a blockchain platform, user access to the ledger will not be monitored or logged for auditing, as it is available to all blockchain nodes.
5. DEVELOP SECURE PAYMENT APPLICATIONS	5.1	Develop all payment applications based on industry best practices and incorporate information security throughout the software development life cycle.	FULLY APPLICABLE	Secure coding and communication practices are comparable for decentralized applications. All practices should be followed to protect against security threats.	Develop all payment applications based on industry best practices and incorporate information security throughout the software development life cycle.
	5.2	Develop all web payment applications (internal and external, and including web administrative access to product) based on secure coding guidelines.	FULLY APPLICABLE	Secure coding and communication practices are comparable for decentralized applications. All practices should be followed to protect against security threats.	Develop all web payment applications (internal and external, and including web administrative access to product) based on secure coding guidelines.
	5.3	Software vendor must follow change control procedures for all product software configuration changes.	FULLY APPLICABLE	Secure coding and communication practices are comparable for decentralized applications. All practices should be followed to protect against security threats.	Software vendor must follow change control procedures for all product software configuration changes.
	5.4	The payment application must not use or require use of unnecessary and insecure services and protocols.	FULLY APPLICABLE	Secure coding and communication practices are comparable for decentralized applications. All practices should be followed to protect against security threats.	The payment application must not use or require use of unnecessary and insecure services and protocols.
6. PROTECT WIRELESS TRANSMISSIONS	6.1	For payment applications using wireless technology, the wireless technology must be implemented securely.	FULLY APPLICABLE	Wireless transmission security standards are not dependent on decentralized applications. The standards are the same.	For payment applications using wireless technology, the wireless technology must be implemented securely.
	6.2	For payment applications using wireless technology, payment application must facilitate use of encrypted transmissions by using WiFi protected access (WPA or WPA2) technology, IPSEC VPN or SSL/TLS.	FULLY APPLICABLE	Wireless transmission security standards are not dependent on decentralized applications. The standards are the same.	For payment applications using wireless technology, payment application must facilitate use of encrypted transmissions by using WiFi protected access (WPA or WPA2) technology, IPSEC VPN or SSL/TLS.
	6.3	Provide instructions for customers about secure use of wireless technology.	FULLY APPLICABLE	Wireless transmission security standards are not dependent on decentralized applications. The standards are the same.	Provide instructions for customers about secure use of wireless technology.

Section	#	Requirement	Applicability	Blockchain Notes	Requirement
7. TEST PAYMENT APPLICATIONS TO ADDRESS VULNERABILITIES AND MAINTAIN PAYMENT APPLICATION UPDATES	7.1	Software vendors must establish a process to identify newly discovered security vulnerabilities (e.g., subscribe to alert services freely available on the Internet) and to test their payment applications for vulnerabilities.	FULLY APPLICABLE	Vulnerability control security standards are similar to decentralized applications (smart contracts). However, the immutability of smart contracts on the blockchain necessitates immediate action to remediate the contract. This entails creating new contract and disabling the old, unsecure contract.	Software vendors must establish a process to identify newly discovered security vulnerabilities (e.g., subscribe to alert services freely available on the Internet) and to test their payment applications for vulnerabilities.
	7.2	Software vendors must establish a process for timely development and deployment of security patches and upgrades, which includes delivery of updates and patches in a secure manner with a known chain-of-trust, and maintenance of the integrity of patch and update code during delivery and deployment.	PARTIALLY APPLICABLE	Vulnerability control security standards are similar to decentralized applications (smart contracts). However, the immutability of smart contracts on the blockchain necessitates immediate action to remediate the contract. This entails creating new contract and disabling the old, unsecure contract.	Software vendors must establish a process for timely development and deployment of security patches and upgrades, which includes delivery of updates and patches in a secure manner with a known chain-of-trust, and maintenance of the integrity of patch and update code during delivery and deployment. On blockchain platforms, proper steps must be taken to ensure the security of the system. The disabling of vulnerable smart contracts or the forking of the blockchain itself may be necessary.
8. FACILITATE SECURE NETWROK IMPLEMENTATION	8.1	The payment application must be able to be implemented into a secure network environment. Application must not interfere with use of devices, applications, or configurations required for PCI DSS compliance	FULLY APPLICABLE	Secure network environments will still be necessary on decentralized platforms.	The payment application must be able to be implemented into a secure network environment. Application must not interfere with use of devices, applications, or configurations required for PCI DSS compliance
	8.2	The payment application must only use or require use of necessary and secure services, protocols, daemons, components, and dependent software and hardware, including those provided by third parties, for any functionality of the payment application.	FULLY APPLICABLE	Secure network environments will still be necessary on decentralized platforms.	The payment application must only use or require use of necessary and secure services, protocols, daemons, components, and dependent software and hardware, including those provided by third parties, for any functionality of the payment application.
	8.3	The payment application must not require use of services or protocols that preclude the use of or interfere with normal operation of two-factor authentication technologies for secure remote access to network resources.	FULLY APPLICABLE	Secure network environments will still be necessary on decentralized platforms.	The payment application must not require use of services or protocols that preclude the use of or interfere with normal operation of two-factor authentication technologies for secure remote access to network resources.

9. CARD DATA STORAGE	9.1	The payment application must be developed such that the database server and web server are not required to be on the same server, nor is the database server required to be in the DMZ with the web server.	PARTIALLY APPLICABLE	Depending on implementation, data storage may differ for cardholder data and decentralized-account data.	The payment application must be developed such that the database server and web server are not required to be on the same server, nor is the database server required to be in the DMZ with the web server.
10. SECURE REMOTE ACCESS TO APPLICATION	10.1	If payment application updates are delivered via remote access into customers' systems, software vendors must tell customers to turn on modem only when needed for downloads from vendor, and to turn off immediately after download completes. Alternatively, if delivered via VPN or other high-speed connection, software vendors must advise customers to properly configure a firewall or a personal firewall product to secure "always-on" connections.	NOT APPLICABLE	Decentralized platforms and applications do not "update" in the same manner as centralized systems. Platforms "fork" and applications (smart contracts) are destroyed to prevent further use when "updated."	If payment application updates are delivered via remote access into customers' systems, software vendors must tell customers to turn on modem only when needed for downloads from vendor, and to turn off immediately after download completes. Alternatively, if delivered via VPN or other high-speed connection, software vendors must advise customers to properly configure a firewall or a personal firewall product to secure "always-on" connections.
	10.2	Any remote access into the payment application must be performed securely.	FULLY APPLICABLE	Remote access should be governed in the same manner as the standards dictate.	Any remote access into the payment application must be performed securely.
11. ENCRYPT SENSITIVE TRAFFIC	11.1	The payment application must not interfere with use of a two-factor authentication mechanism. The payment application must allow for technologies such as RADIUS or TACACS with tokens, or VPN with individual certificates.	FULLY APPLICABLE	Two-factor authentication is still applicable.	The payment application must not interfere with use of a two-factor authentication mechanism. The payment application must allow for technologies such as RADIUS or TACACS with tokens, or VPN with individual certificates.
	11.2	If the payment application facilitates sending of PANs by end-user messaging technologies, the payment application must provide a solution that renders the PAN unreadable or implements strong cryptography or specify the use of strong cryptography to encrypt the PANs.	PARTIALLY APPLICABLE	The scope of data protection must be expanded to include blockchain-specific transaction models.	If the payment application facilitates sending of PANs by end-user messaging technologies, the payment application must provide a solution that renders the PAN unreadable or implements strong cryptography or specify the use of strong cryptography to encrypt the PANs. On blockchain platforms and transaction models, all sensitive transactional data will be protected through cryptography.

Section	#	Requirement	Applicability	Blockchain Notes	Adapted Requirement
12. ENCRYPT ALL NON-CONSOLE ADMINISTRATIVE ACCESS	12.1	If the payment application sends, or facilitates sending, cardholder data over public networks, the payment application must support use of strong cryptography and security protocols such as secure sockets layer (SSL) / transport layer security (TLS) and, internet protocol security (IPSEC) to safeguard sensitive cardholder data during transmission over open, public networks.	FULLY APPLICABLE	Encryption is fully applicable on blockchain platforms and should be used.	If the payment application sends, or facilitates sending, cardholder data over public networks, the payment application must support use of strong cryptography and security protocols such as secure sockets layer (SSL) / transport layer security (TLS) and, internet protocol security (IPSEC) to safeguard sensitive cardholder data during transmission over open, public networks.
	12.2	The payment application must never send unencrypted PANs by end-user messaging technologies (for example, e-mail, instant messaging, chat).	PARTIALLY APPLICABLE	Data Loss Prevention is fully applicable on blockchain platforms and should be used. The scope should be expanded to include blockchain-specific transaction models.	The payment application must never send unencrypted PANs by end-user messaging technologies (for example, e-mail, instant messaging, chat). On blockchain platforms and transaction models, no sensitive transactional data should be unencrypted when sent over end-user messaging technology.
13. MAINTAIN IMPLEMENT	13.1	Instruct customers to encrypt all non-console administrative access using technologies such as SSH, VPN, or SSL/TLS for web-based management and other non-console administrative access.	FULLY APPLICABLE	Standard communication vectors should be treated in the same manner as the standard dictates.	Instruct customers to encrypt all non-console administrative access using technologies such as SSH, VPN, or SSL/TLS for web-based management and other non-console administrative access.
14. ASSIGN PA-DSS RESPONSIBILITIES FOR PERSONNEL AND MAINTAIN	14.1	Develop, maintain, and disseminate a PA-DSS Implementation Guide(s) for customers, resellers, and integrators.	FULLY APPLICABLE	n/a	Develop, maintain, and disseminate a PA-DSS Implementation Guide(s) for customers, resellers, and integrators.
	14.2	Develop and implement training and communication programs to ensure payment application resellers and integrators know how to implement the payment application and related systems and networks.	FULLY APPLICABLE	n/a	Develop and implement training and communication programs to ensure payment application resellers and integrators know how to implement the payment application and related systems and networks.
	14.3	Develop and implement training and communication programs for payment application integrators and resellers.	FULLY APPLICABLE	n/a	Develop and implement training and communication programs for payment application integrators and resellers.

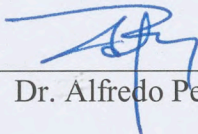# ADAPTING FINANCIAL TECHNOLOGY STANDARDS TO BLOCKCHAIN PLATFORMS

By

Gabriel Bello
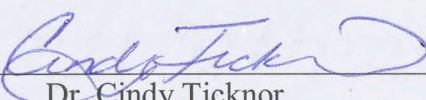
A Thesis Submitted to the

HONORS COLLEGE

In Partial Fulfillment of the Requirements
for Honors in the Degree of

BACHELOR OF SCIENCE
COMPUTER SCIENCE
TURNER COLLEGE OF BUSINESS

Thesis Advisor _____ Date 2/4/2019
                        Dr. Alfredo Perez

Committee Member _____ Date 2/4/2019
                        Dr. Yesem Peker

Committee Member _____ Date 2/4/2019
                        Dr. Lydia Ray

Honors College Dean _____ Date 5/30/2019
                        Dr. Cindy Ticknor