BlockchainWebForms: A Novel Approach to Securing Website Contact Forms

Rohit Mehta

***Abstract*—Blockchain technology provides an interface for machine-to-machine communication that can be conducted without a central body of trust. I study a novel application of blockchain technology and its advantages and limitations by designing and implementing BlockchainWebForms, a WordPress plugin that redirects information entered by users on a contact form to a blockchain instead of to an SQL database. I find that with only a minimal drop in performance, BlockchainWebForms achieve greatly enhanced data security through immutability, robustness, and disintermediation. Real-world uses of BlockchainWebForms face certain limitations, but given the speed at which blockchain technology is developing, it is not unreasonable to think that these limitations will become obsolete in the coming years.**

I. Introduction

As technology has developed into an essential cornerstone of today's society, the need for enhanced, reliable cybersecurity has skyrocketed, manifesting itself in the form of many relatively new developments toward a secure internet, some fueled by the concept of blockchain. New blockchain technologies such as smart contracts present a unique, secure interface for machine-to-machine communication, almost completely eliminating the risk of break-ins and providing a safe platform to record and store information that can be shared without a central administrator or third party. However, it is essential for real-world applications of blockchain technology to address potential trade-offs such as compromises in performance, anonymity, and efficiency. This is where BlockchainWebForms come in.

A. *What are BlockchainWebForms?*

A BlockchainWebForm is a modified version of a Wordpress Contact Form 7 that, instead of submitting user-entered data to an SQL database as usual, initializes and deploys a smart contract that saves the data to a blockchain that only the administrator of the system has the key to [1]. In addition to being saved to a blockchain, the data goes through high-level encryption to further ensure maximum security.

II. The Case for BlockchainWebForms

Traditionally, websites have been hosted on centralized servers and the responsibility of securing data has been left to hosting companies, large corporations, banks, and other organizations. Nowadays, as blockchain technology grows, there are numerous entities racing toward hosting websites in a decentralized fashion. Although advancements are swiftly being made, many questions and concerns loom: replacing traditional hosting with a ledger technology like blockchain is difficult, since websites evolve constantly and blockchain is created for immutable data. Is the added security really worth the tremendous additional costs, slower speed, and faltering performance?

A. *Intermediation*

One of the core features of blockchain is its ability to verify transactions and keep data secure without a central administrator. When data is stored in SQL databases hosted on third party servers, there is a degree of trust that must be placed on them: after all, given the physical tangibility of servers, anybody with sufficient access to them can destroy or corrupt the data. This need to trust third parties with valuable data creates enticing targets for attack, many of which have been exploited [2]. Additionally, since the data is not guarded autonomously, human resources and physical space are needed, both of which cost money. In these instances, blockchain and its disintermediation shine victorious, as there is no longer a need to trust third parties with sensitive data.

B. *Data Tampering*

Because of the vast number of processes that work to provide unique signatures to each block and prevent alteration of any block connected on the chain, any data stored on a blockchain is immutable [3]. Here, blockchains beat out SQL databases: databases can be corrupted, the data stored in them deleted or altered, but blockchains are far more secure [2].

C. *Robustness*

Since blockchain networks are decentralized and involve many nodes across the world constantly referring to one another to ensure consensus, the troubles of ensuring that the database has a backup for critical times or preparing for disasters like hardware failure are erased. Not everything is dependent on the stability of a server: nodes can be added or subtracted at any time without consequences, and no individual node is more important than any other. Although disasters like server crashes cannot be avoided with the BlockchainWebForms model, the most important data—that entered by users—is secure on a blockchain, unaffected.

D. *Performance*

Exclusively considering the troubles of intermediation and robustness found in SQL databases, it seems that a broad, blockchain-oriented approach to hosting data is superior. However, there arise the issues of performance and cost.

Rohit Mehta (e-mail: rohit9mehta@gmail.com)

SQL databases will always perform more efficiently than blockchains, so BlockchainWebForms play to the strengths of both options: apart from the most important data on a website, everything is stored on SQL databases, which ensures performance close to that of websites hosted entirely on SQL databases and excellent security that is characteristic of blockchain technology.

III. The Development of BlockchainWebForms

All experiments and modifications were conducted on website forms present on www.cardboardlearning.io, my WordPress website for an unrelated community project that used the Contact Form 7 plugin for website forms [1]. My objective was to create a new contact form plugin that encrypted entered data and wrote it to a blockchain instead of to an SQL database.

A. *How a Contact Form 7 works*

When a user submits anything on a Contact Form 7, an entry in the WordPress MySQL database is created and the data is stored, accessible by the server administrator with a password. Data submitted through a Contact Form 7 is not encrypted and is intermediated, typically by a third-party hosting service. The system is not particularly robust since it relies on the physical server avoiding damage or downtime, but it is quite efficient.

B. *How a BlockchainWebForm works*

Anything entered and submitted by a user on a BlockchainWebForm goes through multiple layers of encryption. A smart contract is then deployed and the data is written to a blockchain, where the contents are viewable only with an administrator key. The contents are not corruptible because of blockchain's read-only principle: all nodes can read the data (which is now heavily encrypted), but they cannot modify it.

C. *Methods*

To make sure that the transition from a database to a blockchain is seamless, it is necessary to consider both form submission and data retrieval.

1) Handling form submission—Encrypting Data:

a) Why is encryption necessary?: Every node in a blockchain verifies and processes every transaction. Because of this, it has full visibility into the digital signature proving the transaction's origin and the modification the transaction is seeking to make. This feature of blockchain technology proves to be very useful in various facets, but for instances when a blockchain must deal with confidential information, this transparency can create (however minuscule) security risks. To mitigate this potential concern, I heightened the security by encrypting the data entered by the user before deploying a smart contract and writing the data to a blockchain.

b) Methods of encryption utilized in BlockchainWebForms: Before a BlockchainWebForm is activated, a 2048-bit RSA key pair is generated by the website administrator [4]. The public key generated using this asymmetric cryptography technique can be used by anyone to encrypt messages, but only the holder of the paired private key can decrypt messages.

After this encryption system is set up, whenever a form is submitted, a random AES 256-bit key and a random initialization vector are generated. The AES system is an example of symmetric-key encryption in which one key is used for both encryption and decryption. The randomly generated AES 256-bit key is used to encrypt the plain text entered by the user on the BlockchainWebForm. This AES key, in turn, is encrypted with the 2048-bit RSA key pair. The RSA pair is not used to encrypt the user data directly because it is a relatively slow algorithm compared to AES [5]. The initialization vector (IV) adds randomness to the start of the encryption process and helps to produce distinct cipher texts even if the same plain text is encrypted multiple times [6].

I have listed and briefly described the purpose of the key functions in the BlockchainWebForms plugin that contribute to encryption.

importKey

Imports the administrator-generated RSA public key for encryption.

generateIV

Randomly generates a unique 16-byte initialization vector each time a form is submitted. It uses the *crypto.getRandomValues* method from the Web Crypto API to ensure that the random values are cryptographically strong [7].

generateAESKey

Randomly generates a cryptographically secure 256-bit AES key using the *crypto.subtle.generateKey* method from the Web Crypto API [8].

exportAESKey

Exports the AES key created by *generateAESKey* to the "key" variable so it can be used by *encryptString* to encrypt the data.

cryptAESData

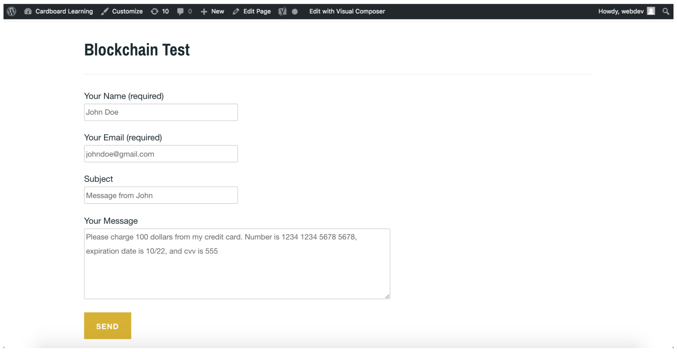
Applies the randomly generated initialization vector to the AES-encrypted data and ensures that a new IV will be generated with every encryption.

encryptString

This is the central function for encryption, and it brings together most of the other functions concerned with encryption. The AES key is encrypted with the RSA key pair, the data entered by the user is encrypted with the AES key, and the IV is applied.

2) Handling form submission—Deploying the Contract: JavaScript code is used to trigger the BlockchainWebForms plugin when a contact form is submitted. Upon the clicking of the "Send" button shown in Figure 1, a smart contract object, *FormTst*, is initiated. Simultaneously, the default propagation of Contact Form 7 transferring the entered data to an SQL database is halted, or "timed out," with a *setTimeout* command. Next, the BlockchainWebForms plugin takes over, checks for errors, and processes the data. Eventually, it deploys the smart contract and sends the data to a blockchain.

The central function of the plugin, *sendBCFormData*, contributes to deploying the smart contract. and First, it sets *web3.eth.defaultAccount* (referenced from the standard web3 JavaScript API, which enables access to the Ethereum blockchain) as the default field from where the transaction is sent [9]. Next, a contract method (*sendData*) is called



Blockchain Test

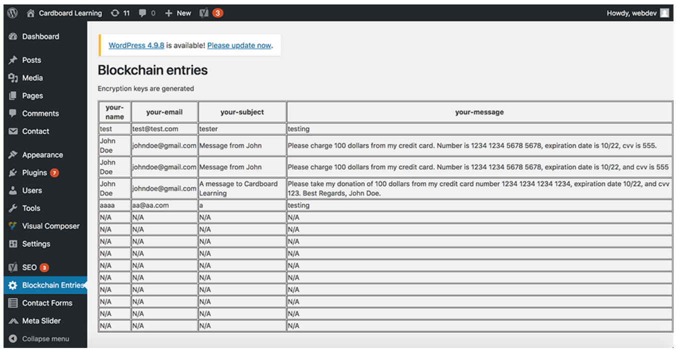
Your Name (required)
John Doe

Your Email (required)
johndoe@gmail.com

Subject
Message from John

Your Message
Please charge 100 dollars from my credit card. Number is 1234 1234 5678 5678, expiration date is 10/22, and cvv is 555

Fig. 1. BlockchainWebForms user interface



WordPress 4.9.8 is available! Please update now.

Blockchain entries

Encryption keys are generated

your-name	your-email	your-subject	your-message
test	test@test.com	tester	testing
John Doe	johndoe@gmail.com	Message from John	Please charge 100 dollars from my credit card. Number is 1234 1234 5678 5678, expiration date is 10/22, cvv is 555.
John Doe	johndoe@gmail.com	Message from John	Please charge 100 dollars from my credit card. Number is 1234 1234 5678 5678, expiration date is 10/22, and cvv is 555
John Doe	johndoe@gmail.com	A message to Cardboard Learning	Please take my donation of 100 dollars from my credit card number 1234 1234 1234 1234, expiration date 10/22, and cvv 123. Best Regards, John Doe.
aaaa	aa@aa.com	a	testing
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A
N/A	N/A	N/A	N/A

Fig. 2. Decrypted blockchain entries in the WordPress administrator dashboard

and subsequently used to retrieve the current gas price (gas is Ethereum's primary unit of measure) and to determine the amount of gas required to complete the transaction [10]. Finally, the data is sent to a blockchain through *sendData*.

3) Handling data retrieval: The process of data retrieval with BlockchainWebForms varies greatly with that of Contact Forms 7.

a) How Contact Forms 7 handle data retrieval: The user's responses to the four parameters of the contact form (name, email, subject, and message) can be easily retrieved from the SQL database and accessed through the server portal by an administrator in unencrypted, plain-text form.

b) How BlockchainWebForms handle data retrieval: The BlockchainWebForms plugin creates an interface for the administrator to view entries. I have listed and briefly described the purpose of the key functions in the BlockchainWebForms plugin that contribute to retrieving and displaying data from the blockchain.

generateKeys

Randomly generates a 2048-bit RSA key pair upon the clicking of a button in the administrator dashboard.

importAESKey

Imports the AES key generated for decrypting the data.

importRSAKey

Imports the RSA private key generated by the administrator for decryption.

decryptRSA

Decrypts the AES key which was encrypted using the RSA private key.

decryptAES

Decrypts the data using the AES key.

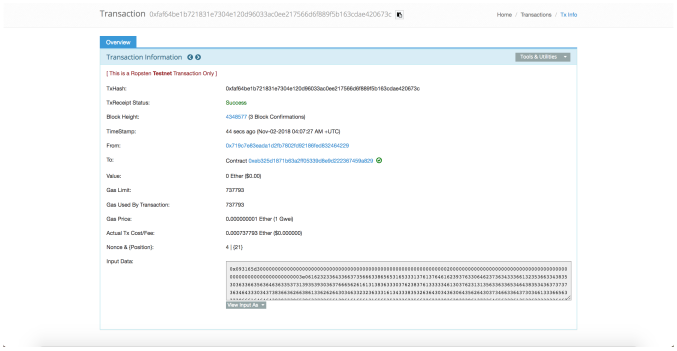
showEntries

This is the central function for displaying the retrieved data. First, a contract method called *getDataAtIndex* gets the data from the blockchain and sorts it by parameters (name, email, subject, message). Next, the data is decrypted and displayed under a "Blockchain Entries" table in the WordPress administrator dashboard as shown in Figure 2.

D. Results

BlockchainWebForms present a successful, effective method of further securing data entered by a user in a contact form without significantly impacting performance. Let's revisit the four issues with traditional websites and blockchain hosted websites that BlockchainWebForms address.

1) Intermediation: BlockchainWebForms eliminate the danger of third parties potentially accessing confidential or sensitive data, since all user-entered data is written to a blockchain instead of stored on a database. Additionally, the data entered is heavily encrypted so that even the nodes of the blockchain network may not view or interpret the data. For example, take the message displayed in Figure 1: nodes interacting with the transaction of that data only see what is shown under "Input Data" in Figure 3.



Transaction 0xfaf64be1b721831e7304e120d96033ac0ee217566d6f889f5b163cdae420673c

Transaction Information

[This is a Ropsten **Testnet** Transaction Only]

TxHash: 0xfaf64be1b721831e7304e120d96033ac0ee217566d6f889f5b163cdae420673c

TxReceipt Status: Success

Block Height: 4348577 (3 Block Confirmations)

TimeStamp: 44 secs ago (Nov-02-2018 04:07:27 AM +UTC)

From: 0x719c7e83eada1d2fb7802fd92186fed832464229

To: Contract 0xeb325d1871b63a2ff05339d8e9d222367459a829 ✓

Value: 0 Ether ($0.00)

Gas Limit: 737793

Gas Used By Transaction: 737793

Gas Price: 0.000000001 Ether (1 Gwei)

Actual Tx Cost/Fee: 0.000737793 Ether ($0.000000)

Nonce & {Position}: 4 | {21}

Input Data:

```
0x093165d300000000000000000000000000000000000000000000000000000000000000200000000000000000000000000000000000000
000000000000000000000003e061623233643366373566633865653165333137613764616239376330646237363433366132353663343835
3036336635636463633537313935393036376665626161313836333037623837613333346130376231313563363365346438353436373737
3634643330343738366362663861336262643034633232363331613433383532636430343630643562643037346633643730346133366563
```

Fig. 3. Information from a test transaction that encrypted and stored data on a blockchain

2) Data Tampering: The utilization of blockchain to secure data entered in contact forms means that the dangers of data deletion or manipulation are eliminated, given the immutability it grants [3]. Referring back to Figure 3, the data is viewable (albeit in a heavily encrypted form), but it is not alterable by anyone, not even the administrator.

3) Robustness: Robustness, like immutability, can be attributed to the nature of blockchain technology. Since various nodes participate in the transaction, a failure of one of the nodes will not jeopardize the data or its security like a server failure or hijack would for an SQL database.

4) Performance: Multiple tests regarding performance were conducted using two reputable website performance tools, Pingdom Tools (tools.pingdom.com) and GTMetrix (gtmetrix.com). Pingdom's website monitoring tools analyze the front-end performance of any website and return a Performance Grade along with numerous other test results, and

GTMetrix's performance reports include a "PageSpeed" score along with other details [11]. I ran three tests on each performance tool at different times of the day from the same location, ensuring that all conditions (internet, computer performance, etc) were held near constant, and that no other aspect of the web page (cardboardlearning.io/blockchain-test) was altered. Below are tables displaying a summary of the results I received when BlockchainWebForms were activated and when they were deactivated. For both tables, an asterisk (*) signifies that the BlockchainWebForms plugin was activated.

TABLE I
PINGDOM

	Trial 1	Trial 2	Trial 3	Average
Performance Grade	82 (B)	82 (B)	82 (B)	82 (B)
Performance Grade*	79 (C)	80 (C)	79 (C)	79 (C)
Page size	146.9 KB	146.9 KB	146.9 KB	146.9 KB
Page size*	190.5 KB	192.1 KB	190.5 KB	191.0 KB
Load time	1.11 s	726 ms	1.19 s	1.01 s
Load time*	1.19 s	958 ms	916 ms	1.02 s
Requests	20	20	20	20
Requests*	26	26	26	26

TABLE II
GTMETRIX

	Trial 1	Trial 2	Trial 3	Average
PageSpeed Score	97 (A)	97 (A)	97 (A)	97 (A)
PageSpeed Score*	95 (A)	95 (A)	95 (A)	95 (A)
YSlow Score	75 (C)	75 (C)	75 (C)	75 (C)
YSlow Score*	72 (C)	72 (C)	72 (C)	72 (C)
Load Time	0.9 s	1.4 s	0.9 s	1.1 s
Load Time*	1.3 s	1.4 s	1.3 s	1.3 s
Page size	154 KB	154 KB	154 KB	154 KB
Page size*	200 KB	200 KB	200 KB	200 KB
Requests	19	19	19	19
Requests*	25	25	25	25

The differences of about two or three between the average Pingdom Performance Grades, GTMetrix PageSpeed Scores, and GTMetrix YSlow Scores show the slight drop in performance when BlockchainWebForms were enabled on the web page. However, the loading times were nearly identical as measured by both Pingdom and GTMetrix, the page sizes varied by about 46 KB, and the number of requests varied by six. Overall, although BlockchainWebForms do slightly affect the performance of the web page, most users will find that the significant security boost BlockchainWebForms grant makes up for it.

IV. Advantages and Shortcomings: An Evaluation of BlockchainWebForms

A. Security

Data entered in traditional contact forms is stored on an SQL database, where it can be mutated, manipulated, and even deleted [2]. BlockchainWebForms, on the other hand, submit data directly to a blockchain through deploying a smart contract and thus greatly improve security.

B. Performance

As shown earlier, the integration of blockchain with a contact form does affect website performance due to the computationally complex consensus mechanisms that a blockchain network must carry out to ensure security, but only minimally.

With this drawback to blockchain technology in mind, I developed BlockchainWebForms so that their performance would resemble that of a traditional Contact Form 7 but their security would match that of a blockchain hosted website. As blockchain technologies develop, however, there is room for improvement.

BlockchainWebForms use Ethereum's currency, ether, to process transactions. Ethereum is currently in the process of switching from proof of work, the proof consensus mechanism made famous by the rise of Bitcoin, to proof of stake, a relatively new mechanism that consumes much less energy [12]. With this change and with future developments in proof mechanisms, I expect blockchain performance only to improve, especially given the relatively unexplored nature of the new technology.

C. Other Inconveniences

1) The need for a browser extension: A BlockchainWebForm is an Ethereum-enabled dApp (decentralized application) in the form of a WordPress plugin, so it requires an extension to help it function correctly when loaded on a browser. MetaMask is one such extension: it allows users to run Ethereum dApps in a browser without running a full Ethereum node by enabling the connection between the plugin and the web3.js library, thus allowing the plugin to interact with an Ethereum provider [13]. Although installing and setting up the MetaMask extension is far from a hassle, the need for it can be a limitation, especially if BlockchainWebForms are to be administered on widely popular websites.

2) Transaction times: With traditional contact forms, data entered by a user is transferred almost instantly to the database for the admin to view. With BlockchainWebForms, however, the increase in amount of processing required to transfer data means that transaction times depend on the amount of data processed. The slightly longer transaction times, however, should not be a significant problem for most applications of BlockchainWebForms unless prompt, real-time data retrieval is desired by the administrator. This limitation does not impact users entering data on the form.

3) Conclusion: BlockchainWebForms fulfill their purpose of enhancing the security of data entered in contact forms while keeping website performance at relatively stable levels. As blockchain technology develops, newer proof mechanisms or encryption methods may be incorporated into BlockchainWebForms to further improve them. For example, zero-knowledge proof mechanisms are generating excitement in the blockchain community lately because of their ability to restrict knowledge of the sender, recipient, and quantity of data from all groups involved in a transaction [14]. As of now, however, BlockchainWebForms are structured to play to blockchain technology's current strengths and weaknesses. They are ready to be applied to real-world websites and are, I believe, a step toward the future of blockchain dApps and their integration with existing technologies.

References

[1] T. Miyoshi, "Contact Form 7." [Online]. Available: https://contactform7.com/

[2] R. Millman, "Hackers now hit MySQL databases with ransomware," February 2017. [Online]. Available: https://www.scmagazineuk.com/hackers-hit-mysql-databases-ransomware/article/1475201

[3] [Online]. Available: https://developer.ibm.com/tutorials/cl-blockchain-basics-intro-bluemix-trs/

[4] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Association for Computing Machinery Journal*, 1978. [Online]. Available: 10.21236/ada606588

[5] F. Sultana, B. Choudhury, S. M. S, and J. Mungara, "A Study on Data Encryption Using AES and RSA," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 4, 2017. [Online]. Available: 10.15680/IJIRCCE.2017

[6] K. T. Huang, J. Chiu, and S. Shen, "A Novel Structure with Dynamic Operation Mode for Symmetric-Key Block Ciphers," *International Journal of Network Security & Its Applications*, vol. 5, no. 1, pp. 19–20, 2013. [Online]. Available: 10.5121/ijnsa.2013.5102

[7] "Crypto.getRandomValues ()," March 2018. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Crypto/getRandomValues

[8] " SubtleCrypto.generateKey ()," March 2018. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/generateKey

[9] "Web3.eth," 0. [Online]. Available: https://github.com/ethereum/wiki/wiki/JavaScript-API

[10] V. Buterin, "A Next Generation Smart Contract & Decentralized Appli-cation Platform [White paper]," 2014. [Online]. Available: https://www.weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[11] T. Jankov, "Improving Page Load Performance: Pingdom, YSlow and GTmetrix," August 2018. [Online]. Available: https://www.sitepoint.com/improving-page-load-performance-pingdom-yslow-gtmetrix/

[12] A. Asgaonkar, G. Piliouras, N. Rush, and V. Zamfir, "Introducing the "Minimal CBC Casper" Family of Consensus Protocols [White paper]," 2018. [Online]. Available: https://github.com/cbc-casper/cbc-casper-paper/blob/master/cbc-casper-paper-draft.pdf

[13] Bruno, "What is MetaMask and How to Send and Receive Ether with it?" January 2018. [Online]. Available: https://bitfalls.com/2018/02/16/metamask-send-receive-ether/

[14] L. Schor, "On Zero-Knowledge Proofs in Blockchains – Argon Group – Medium," March 2018. [Online]. Available: https://medium.com/@argongroup/on-zero-knowledge-proofs-in-blockchains-14c48cfd1dd1

Rohit Mehta is a high school senior at Maria Carrillo High School (MCHS) in Santa Rosa, California. His research interests include blockchain optimization and artificially intelligent autonomous systems. He will graduate from MCHS in 2019 and plans to major in Computer Science as an undergraduate. Contact him at rohit9mehta@gmail.com.