Hiding Historical Data on Permissionless Blockchain

Zihan Wu

School of computer science, Jiangsu University, 212013,China

优胜奖

Introduction

Arbitrary data insertion into blockchain has been extensively used as a public bulletin board due to its immutability. Although this data insertion can be beneficial in some use cases, researches show that even specific blockchains like Bitcoin already contain harmful and potentially illegal documents, images, and links. Despite there is a lot of attention to blockchain technology, the proposed solution for dealing with these kinds of data insertion is far from feasible. In this paper, we provide a historical data hiding scheme to mitigate the influence of arbitrary data insertion. We describe a 'burn after reading' mechanism for UTXO-based cryptocurrencies and present a transaction data pruning algorithm for locally erasing inserted data. Additionally, our scheme wouldn't affect the liveness and persistence and reduce the local storage size of the chain when applied to the existing blockchain network.

Hiding Blockchain Data

1. Modification on the Transaction

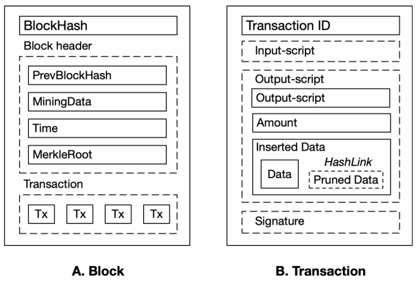


Figure 1. Transaction Modification

We denote by t_D and t_C the plaintext and ciphertext of the inserted data. Before issuing a transaction in the blockchain network, the nodes first generate the document encryption key $dk \leftarrow \{0,1\}^{\lambda}$ randomly. Then the nodes execute $t_C \leftarrow Enc(dk, t_D)$ to encrypt the inserted data. It uses asymmetric encryption algorithm (e.g AES) with the secret key dk to encrypt the plaintext data t_D to obtain the ciphertext t_C. The ciphertext t_C would be store in the third-party storage (e.g. IPFS, Inter-Planetary-File-System) and get the hash link $hLink_{t_C}$ to the storage location. Besides, to make sure that the plaintext and the encrypted data are identical, the blockchain nodes need to calculate the hash of t_D. So the inserted data should be $t_D^{'} := \{hLink_{t_C}, hash(t_D), t_D\}$.

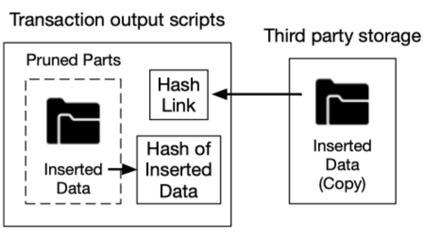


Figure 2. System Model

2. Transaction Data Pruning

The following interfaces enable the nodes to access the network and other users:

- $\{\mathcal{C}', \perp\} \leftarrow \Gamma.updateChain$: return a longer and valid chain $\mathcal{C}'$ in the network (if it exists), otherwise returns $\perp$.
- $\{0,1\} \leftarrow \Gamma.validateChain(\mathcal{C})$: The chain validity check takes as input a chain $\mathcal{C}$ and returns 1 iff the chain is valid according to a public set of rules.
- $\{0,1\} \leftarrow \Gamma.validateBlock(\mathcal{B})$: The block validity check takes as input a block B and returns 1 iff the block is valid according to a public set of rules.
- $\Gamma.broadcast(x)$: takes as input some data x and broadcasts it to all the nodes of the blockchain network.

In the proposed scheme, the inserted data would only be displayed in a period. Then, when the display time expires, the blockchain network would execute the transaction data pruning process to erase these inserted data. We define an extra interface to realize the pruning function:

- $\{^{n-q\rceil}\mathcal{C}, \mathcal{C}'^{\lceil q}\} \leftarrow \Gamma.transactionPrune$: return q rightmost blocks of $\mathcal{C}'$ with pruned transaction data and q leftmost blocks of $\mathcal{C}$ without any modification on the transaction data.

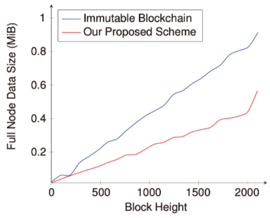
Algorithm 1: transactionPrune

Input: Chain $C = (B_1, \ldots, B_n)$ of length n, the length of block need to excute the process q

Output: Chain consist of blocks with pruned transactions $\{^{n-q\rceil}\mathcal{C}, \mathcal{C}'^{\lceil q}\}$

Check $^{n-q\rceil}\mathcal{C}$;
foreach $Tx_{i,j} \in B_i \in \mathcal{C}'^{\lceil q}$ **do**
  if $P : \langle Tx_{i,j}; t_D \rangle = ture$ **then**
    Prune t_D from $t'_D := \{hLink_{t_C}, hash(t_D), t_D\}$;
    Calculate $Tx_{i,j,saved\ hash}$ with the pruned part;
  else
    Continue
End.

Experiment



We show the disk size required for full nodes to store the blockchain data compared to the immutable blockchain. The experiment was conducted on blocks of different sizes rather than on blockchains of different sizes.

Conclusion

We have presented a scheme to hide historical data by pruning the inserted data in the transaction output scripts in a permissionless blockchain (e.g Bitcoin). As we have discussed, there are multiple reasons why the users in the blockchain prefer a redactable blockchain to an immutable one. The proof-of-concept shows that our scheme is feasible, as the implementation of our design requires a minor modification to the current blockchain protocol. Moreover, the overhead caused by the pruned process is negligible.

Reference

1. S. Nakamoto and A. Bitcoin, "A peer-to-peer electronic cash system," Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf, 2008.
2. M. E. Peck, "Blockchains: How they work and why they'll change the world," IEEE spectrum, vol. 54, no. 10, pp. 26–35, 2017.
3. B. Gipp, J. Kosti, and C. Breitinger, "Securing video integrity using decentralized trusted timestamping on the bitcoin blockchain," in Mediterranean Conference on Information Systems (MCIS), Association For Information Systems, 2016.
4. M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in 2014 IEEE Symposium on Security and Privacy, pp. 443–458, IEEE, 2014.
5. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in 2016 IEEE symposium on security and privacy (SP), pp. 839–858, IEEE, 2016.
6. G. Tziakouris, "Cryptocurrencies—a forensic challenge or opportunity for law enforcement? an interpol perspective," IEEE Security & Privacy, vol. 16, no. 4, pp. 92–94, 2018.
7. G. Ateniese, B. Magri, D. Venturi, and E. Andrade, "Redactable blockchain–or–rewriting history in bitcoin and friends," in 2017 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 111–126, IEEE, 2017.
8. D. Deuber, B. Magri, and S. A. K. Thyagarajan, "Redactable blockchain in the permissionless setting," in 2019 IEEE Symposium on Security and Privacy (SP), pp. 124–138, IEEE, 2019.
9. M. Florian, S. Henningsen, S. Beaucamp, and B. Scheuermann, "Erasing data from blockchain nodes," in 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pp. 367–376, IEEE, 2019.
10. J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 281–310, Springer, 2015.
11. F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2084–2123, 2016.
12. R. Matzutt, J. Hiller, M. Henze, J. H. Ziegeldorf, D. Müllmann, O. Hohlfeld, and K. Wehrle, "A quantitative analysis of the impact of arbitrary blockchain content on bitcoin," in International Conference on Financial Cryptography and Data Security, pp. 420–438, Springer, 2018.

For further information

Contact：Zihan Wu
Phone：18351961505
Email：zihan.wu042@qq.com