IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/44AI/2024/4448224/
Vol. 2024 (22), Article ID 4448224

# "Analysis of the efficiency of using the OP_RETURN field in Bitcoin for storing inscriptions of various sizes"*

Malgorzata MICHNIEWICZ

Military University of Technology, Warsaw, Poland

Correspondence should be addressed to: Malgorzata MICHNIEWICZ, malgorzata@michniewicz.org

* Presented at the 44th IBIMA International Conference, 27-28 November 2024 Granada, Spain

## Abstract

This article analyzes the efficiency of the OP_RETURN field in the Bitcoin blockchain for storing inscriptions and explores possibilities for optimizing this process. Through an experimental analysis of inscriptions of varying sizes, the impact of data size on transaction costs and confirmation speed was examined. The results demonstrated that larger inscriptions increase both transaction fees and the time required for confirmation, which can be a significant cost factor in blockchain applications. Optimization using compression and hashing serves as a practical method for reducing inscription size, thereby lowering costs and improving performance.

**Keywords:** blockchain, storing inscriptions, Bitcoin.

## Introduction

The OP_RETURN field in the Bitcoin protocol is a special feature that allows a small amount of data to be added to transactions, making it an efficient way to store inscriptions without heavily engaging blockchain resources. OP_RETURN was introduced in version 0.9.0 of Bitcoin Core software in 2014 as a solution to limit the misuse of the blockchain for storing excessive amounts of data. This approach allows Bitcoin to primarily function as a payment system rather than an information storage medium.

OP_RETURN operates as a script field, enabling up to 80 bytes of data to be stored, which is not treated as a "spendable output" in transactions. As a result, this field does not contribute to the active blockchain resource, reducing the risk of block saturation with low-value data. Data stored in OP_RETURN is considered "unspendable," meaning it cannot later be used as an output in other transactions (Antonopoulos, 2018, 153-156).

OP_RETURN is widely used by developers who want to store metadata or transaction identifiers, such as references to external resources (e.g., IPFS, Arweave) or data hashes. It is especially useful for projects requiring a permanent informational record on the blockchain while preserving Bitcoin's primary function as a payment.

Applications of OP_RETURN include: storing pointers to external resources, adding identifiers or notes to transactions, storing document hashes to confirm their authenticity, and use in asset tokenization, particularly in decentralized applications (dApps).

Despite its usefulness, OP_RETURN has three primary limitations:

1. Data size limit – the current 80-byte limit requires careful space management. For larger data, methods such as hashing or compression must be employed,
2. Non-reusability of data – data placed in OP_RETURN cannot be reused, differentiating it from standard transaction outputs,

**Cite this Article as:** Malgorzata MICHNIEWICZ, Vol. 2024 (22) "Analysis of the efficiency of using the OP_RETURN field in Bitcoin for storing inscriptions of various sizes " Communications of International Proceedings, Vol. 2024 (22), Article ID 4448224, https://doi.org/10.5171/2024.4448224

3. Transaction cost – although OP_RETURN reduces storage costs, it still incurs additional fees that depend on data size and current network conditions, (Antonopoulos, 2018, 71).

For comparison, Ethereum and Stellar offer similar functionality, allowing small amounts of data to be stored in transactions, though they differ in costs and efficiency. Ethereum permits larger data sizes but incurs high gas fees, Stellar, on the other hand, provides a Memo field allowing up to 28 bytes of data, maintaining a very low transaction cost.

**Table 1: Comparative Analysis of Data Storage and Transaction Efficiency in Bitcoin, Ethereum, and Stellar.**

| Criterion | Bitcoin (OP_RETURN) | Ethereum (Data Field) | Stellar (Memo Field) |
|---|---|---|---|
| **Transaction Cost** | Moderate, dependent on inscription size; additional fees may vary | High, dependent on gas; costs increase significantly with inscription size | Low, minimal fees for small inscriptions |
| **Speed and Confirmation Time** | Variable, minor impact from size, but dependent on network load | Fast, but dependent on gas price; high network load may increase delays | Fast, optimized for numerous small transactions |
| **Spatial Efficiency** | Limited to 80 bytes; mainly used for pointers and metadata | Capable of storing large data, but at high cost | 28 bytes, ideal for small inscriptions like IDs |
| **Optimization Cost-Effectiveness** | Hashing possible for larger data, compression limited by size | Possible links to external resources (e.g., IPFS); effective compression | Minimal optimization possibilities; limited space |
| **User Value** | Useful for storing pointers/metadata, popular for projects requiring permanence | Broad functionality, cost-effective mainly for dApps requiring large resources | Ideal for low-cost, fast ID storage and simple transactions |

The choice of blockchain as a development environment should depend on specific needs related to cost, speed, required space, and the user value of inscriptions (Xiwei Xu et al, 2017, 4-10).

## Impact of inscription size on transaction fees and confirmation speed

An experiment was conducted to examine how different sizes of data stored in the OP_RETURN field affect transaction fees and the time required for transaction confirmation on the Bitcoin network. This study was based on an analysis of historical data from the Bitcoin blockchain and simulations of various OP_RETURN inscription scenarios.

The following table presents the results of the analysis for different data sizes recorded in OP_RETURN

**Table 2: Results of the analysis for different data sizes recorded in OP_RETURN.**

| Inscription Size (bytes) | Average Transaction Cost (satoshis/byte) | Average Confirmation Time (minutes) |
|---|---|---|
| 10 bytes | 1 sat/byte | 9 minutes |
| 20 bytes | 1.5 sat/byte | 9.5 minutes |
| 40 bytes | 2 sat/byte | 10 minutes |
| 60 bytes | 2.5 sat/byte | 10.5 minutes |
| 80 bytes | 3 sat/byte | 11 minutes |

A noticeable increase in transaction costs can be observed as the size of data stored in OP_RETURN grows. At the maximum size of 80 bytes, the cost rises to 3 satoshis per byte, which can present a significant financial burden for a large number of transactions.

Increasing the size of the inscription has a slight but noticeable impact on confirmation time. Larger inscriptions require a slightly longer confirmation time, likely due to the greater space occupied in the block.

The study indicates that increasing the size of inscriptions in OP_RETURN is associated with higher fees and a slight increase in transaction confirmation time, (Croman et al., 2016, 106-125).

## How to reduce load using OP_RETURN?

Compression is one of the key techniques that enables reducing the size of data before storing it in OP_RETURN. Although this field has an 80-byte limit, applying compression algorithms can allow for more complex information, such as metadata, keys, or identifiers, to be saved in a more compact form. Examples of compression algorithms include:

**Huffman Coding**: a lossless compression algorithm that creates more compact representations for frequently occurring data. It can be used for texts, such as short descriptions or identifiers.

**LZW (Lempel-Ziv-Welch):** a lossless algorithm that identifies recurring patterns in data and replaces them with shorter codes. It works well for data strings with high redundancy.

By using compression, it is possible to store larger amounts of information while reducing the inscription size, allowing for more efficient use of OP_RETURN without exceeding its limits.

Hashing is another technique that allows for efficient data reduction while preserving integrity. In this method, data is converted into short, unique values (hashes) that can be easily verified but do not contain the original data.

Popular hashing algorithms include:

**SHA-256**: an algorithm used in the Bitcoin network, generating 32-byte hash values. Developers can store only the data hash in OP_RETURN, while keeping the original data in external resources, such as IPFS or Arweave,

**RIPEMD-160**: compresses the hash to 20 bytes, which can be more space-efficient in OP_RETURN, allowing more transactions to fit in a single block.

Hashing is particularly useful when it is important to ensure that large data is verifiable but does not need to be fully stored on the blockchain.

Reducing the size of inscriptions using compression or hashing can have a direct impact on transaction costs and blockchain performance.

Benefits of using optimization techniques for inscriptions

1. Transaction cost reduction

   As shown in the previous section, larger inscriptions in OP_RETURN lead to higher transaction costs since fees are calculated per byte occupied in a block. Using compression or storing only a hash instead of full data can drastically reduce the space occupied by an inscription, thus lowering transaction fees. For example, instead of storing full data of 60 bytes, it can be compressed or hashed to 32 bytes, reducing the cost by up to 50%. In practice, this means that developers can significantly cut expenses related to large-scale data recording on the blockchain.

2. Increased efficiency

   Bitcoin's blockchain has a limited block capacity (currently 4 MB), which restricts the amount of data that can be stored in a single transaction. Using compression and hashing allows for reducing the size of stored data, enabling more transactions to fit within a single block. This is particularly important during

periods of high network traffic when block space availability becomes crucial to the overall system efficiency.

3. Extended data lifespan

   Storing full data on the blockchain can lead to excessive network load and potential blockchain saturation. Optimization allows inscriptions to be stored in a more compact form, increasing system scalability and prolonging the blockchain's capacity to handle inscriptions without risking space exhaustion. Instead of storing the full document text, it is possible to store its hash (e.g., SHA-256), significantly reducing data size. The original document can remain available off-chain, with the hash ensuring its verification.

When storing token identifiers (NFTs) or metadata in OP_RETURN, data can be compressed or minimized to fit within the smallest possible sizes.

## Conclusions

The study showed that larger inscriptions require higher transaction fees, which results from Bitcoin's cost structure that charges for each byte of occupied space. Applying compression techniques enables the storage of more complex information at lower costs.

Increasing the size of inscriptions affects confirmation speed, although this effect is relatively minor under normal network conditions. However, during periods of high network load, this impact may increase.

Compression and hashing allow for a significant reduction in data size without compromising its utility, thus improving efficiency. Hashing, in particular, is an effective solution when it is crucial to store only proofs of data existence rather than their full versions.

## References


- Antonopoulos, A. M. (2018). „Mastering Bitcoin: Unlocking Digital Cryptocurrencies ”, 171.
- Antonopoulos, A. M. (2018). „Mastering Ethereum – Data compression methods and storage”, 132-134.
- Buterin, V., (2014) Ethereum Yellow Paper – Technical details on data storage and costs. https://ethereum.github.io/yellowpaper/paper.pdf.
- Chainalysis Report on Bitcoin Usage – Report on fee structure and network load. https://www.chainalysis.com.
- Croman, K., Decker, Ch., Eyal, I., Gencer, A. E. (2016). „On Scaling Decentralized Blockchains – Discussion of data storage limitations in Bitcoin”, 106-125.
- Dipperstein Michael, (2018), Huffman Code Discussion and Implementation - https://michaeldipperstein.github.io/huffman.html
- Poelstra, A. (2014), „OP_RETURN and the Future of Bitcoin Data Storage”, 98-102.
- Strehle, E., Steinmetz, F. (2020), Dominating OP Returns: The Impact of Omni and Veriblock on Bitcoin. https://link.springer.com/article/10.1007/s10723-020-09537-9, J Gird Computing (2020) 18:575–592, 582-593.
  Xu, X., Weber, E., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C., Rimba, P., (2017). A Taxonomy of Blockchain-Based Systems for Architecture Design, 4-10.