Blockchain-Based Version Control for Secure Software Engineering

[1] Ali Raza Khan, [2] Sana Shahid,[3] Imran Tariq

[1] Department of Computer Science, University of Karachi, Pakistan
[2,] School of Computing and Engineering Sciences, Institute of Business Administration (IBA), Karachi, Pakistan
[3,] Department of Software Engineering, NED University of Engineering and Technology, Karachi, Pakistan
***Corresponding Author :** aliraza.khan@uok.edu.pk *

Abstract: *In the evolving landscape of software engineering, ensuring the integrity and security of source code is paramount. Traditional version control systems (VCS) like Git, while widely adopted, are susceptible to unauthorized modifications and lack inherent mechanisms for accountability. This paper explores the integration of blockchain technology into VCS to address these vulnerabilities. By leveraging blockchain's decentralized and immutable nature, we propose a framework that records every code change as a transaction, ensuring traceability and preventing unauthorized alterations. The study evaluates the feasibility, benefits, and challenges of implementing such a system, aiming to enhance the security and reliability of software development processes.*

Keywords: *blockchain, version control, software engineering, security, Git, immutability, decentralization, traceability*

Introduction

Version control systems (VCS) are integral to modern software development, enabling multiple developers to collaborate efficiently. However, traditional VCS platforms are centralized, posing risks of unauthorized access and data breaches. Blockchain technology, with its decentralized and tamper-resistant characteristics, offers a promising solution to these challenges. By integrating blockchain with VCS, each code change can be securely recorded, ensuring transparency and accountability in the development process.

Overview of Traditional Version Control Systems

Traditional Version Control Systems (VCS) have been an essential part of software development, allowing teams to manage and track changes to codebases. These systems provide a framework where developers can collaborate on projects, ensuring that code versions are stored and updated in an organized manner. The two primary types of traditional version control systems are **Centralized Version Control Systems (CVCS)** and **Distributed Version Control Systems (DVCS)**.

In **Centralized Version Control Systems (CVCS)**, there is a central repository that contains the complete history of all files. Developers check out files from this central repository, make changes locally, and later commit those changes back to the central server. This system allows teams to track and manage the versions of their code, but it has several drawbacks. For instance, if the central server goes down, no one can commit new changes, and the repository's integrity is entirely dependent on that central server.

On the other hand, **Distributed Version Control Systems (DVCS)**, such as Git, improve upon the limitations of CVCS by providing each developer with a complete copy of the repository. This local copy contains the entire history of the project, allowing developers to work offline and commit changes locally. Once connected to the network, developers can sync their changes with other developers or the central repository. This model enhances collaboration and makes the system more resilient, as multiple copies of the repository reduce the risk of data loss.

Both CVCS and DVCS rely heavily on a centralized or decentralized server to store and track the changes made to the code. While traditional VCS solutions have been effective for managing software development projects, they come with limitations in terms of scalability, security, and transparency. In particular, they do not inherently provide mechanisms to verify the integrity of the code or track unauthorized changes, which becomes a critical issue when handling sensitive or high-value software development projects. This is where new technologies, like blockchain, are starting to play a role in securing the version control process and ensuring the authenticity and immutability of the codebase.

Introduction to Blockchain Technology

Blockchain technology is a revolutionary concept that enables secure, transparent, and decentralized systems for recording transactions and data. At its core, blockchain is a distributed ledger system that allows data to be stored across multiple computers or nodes, making it highly resistant to tampering, hacking, or centralized control. This technology underpins cryptocurrencies like Bitcoin and Ethereum but has applications far beyond digital currencies, including supply chain management, secure voting systems, and version control for software.

Three key principles define blockchain technology: **decentralization**, **immutability**, and **consensus mechanisms**.

1. **Decentralization**: Unlike traditional centralized systems where a single entity controls the data, blockchain operates on a peer-to-peer network. Each participant (node) in the network has a copy of the entire ledger, and there is no central authority overseeing the system. This decentralization eliminates the need for a trusted intermediary, ensuring that no single entity has control over the data, which enhances transparency and security.

2. **Immutability**: Once data is recorded on a blockchain, it is nearly impossible to alter or delete. Each block in the chain contains a timestamp and a cryptographic hash of the previous block, creating a secure and permanent record. This immutability ensures that the data remains tamper-proof, which is essential for applications that require trust and verification, such as version control or transaction logging.

3. **Consensus Mechanisms**: Blockchain networks rely on consensus mechanisms to validate transactions and add them to the ledger. These mechanisms ensure that all participants in the network agree on the validity of a transaction. Common consensus algorithms include **Proof of Work (PoW)**, which requires participants to solve complex mathematical problems, and **Proof of Stake (PoS)**, which selects validators based on the amount of cryptocurrency they hold. These mechanisms ensure that transactions are verified in a distributed and secure manner without the need for centralized oversight.

Need for Enhanced Security in Software Development

Software development involves numerous contributors working on large, complex codebases, and managing these collaborative efforts efficiently requires robust version control systems (VCS). However, despite the widespread use of traditional VCS, such as Git and Subversion, these systems face significant security challenges that can undermine the integrity and reliability of software projects. The need for enhanced security in software development has become more apparent due to concerns about **unauthorized access**, **data tampering**, and **lack of accountability**.

1. **Unauthorized Access**: In traditional VCS, especially centralized systems, access to the repository is typically controlled by authentication mechanisms, such as usernames and passwords. However, these systems can still be vulnerable to unauthorized access, either through weak passwords, compromised credentials, or insider threats. This means that malicious actors can gain access to sensitive codebases, make unauthorized changes, and potentially introduce vulnerabilities into the software. The lack of adequate security measures in these systems makes it harder to prevent or detect unauthorized modifications, which poses a significant risk, especially in projects involving sensitive data or mission-critical applications.
2. **Data Tampering**: Another challenge of traditional version control is the potential for data tampering. Since VCSs store historical versions of code, they are prone to situations where data might be altered without proper traceability. In centralized systems, for example, once a developer commits changes, they are pushed to a central server, which can be tampered with either accidentally or maliciously. Without mechanisms to ensure the integrity of these changes, the system is vulnerable to hidden malicious alterations or unintentional errors. This lack of control can compromise the reliability of the code and even lead to the deployment of faulty or malicious software.
3. **Lack of Accountability**: Traditional VCS also struggle with accountability. While they track changes and attribute them to specific users, there is limited transparency regarding who made which change and why. In a team of many contributors, it becomes challenging to track down the origins of particular bugs, errors, or malicious actions. For instance, if a vulnerability is introduced into the codebase, traditional VCS might not provide sufficient context or traceability to pinpoint the person responsible. This lack of accountability can result in delayed response times when issues arise, making it difficult to maintain the integrity of the software development lifecycle.

Benefits of Blockchain-Enhanced Version Control Systems

Integrating blockchain with traditional Version Control Systems (VCS) provides several significant benefits, particularly in terms of **security**, **traceability**, and **transparency**. These advantages make blockchain-enhanced VCS a compelling solution for software development teams aiming to improve the integrity of their codebases and enhance accountability throughout the software development lifecycle.

Improved Security

Blockchain technology offers a robust layer of security to the version control process by leveraging its **decentralized** nature. In traditional VCS, repositories are often centralized, meaning they rely on a single point of failure. If the central server is compromised, the entire project can be at risk. Blockchain addresses this by distributing the repository across multiple nodes, making it much more resilient to hacks or unauthorized access. The key security benefits include:

- **Tamper-Proof Records**: Once a commit is recorded on the blockchain, it is nearly impossible to alter or delete, ensuring that the history of changes remains intact. This makes it virtually immune to attacks like **data tampering**, as any unauthorized attempt to change code would be immediately detectable.
- **Decentralized Control**: Since blockchain operates on a peer-to-peer network, there is no central authority controlling the repository. This eliminates risks related to single points of failure, as each participant (or node) has an identical copy of the code and its history.
- **Enhanced Authentication**: Blockchain can strengthen access control by implementing cryptographic signatures, where only authorized users can commit changes. Smart contracts can enforce these authentication rules, ensuring that only valid contributors have write access to the codebase.

Improved Traceability

Blockchain's core feature of immutability is a powerful tool for improving traceability **in software development. Every commit, change, or update to the code is recorded with a cryptographic hash on the blockchain, making it easy to track the full history of the project. The key traceability benefits include:**

- **Complete Audit Trail**: Blockchain provides a full, time-stamped history of all commits and changes made to the software. Developers can easily trace back to see who made specific changes, when they were made, and what was altered. This audit trail is crucial in large-scale, collaborative environments, where multiple developers contribute to the same codebase.
- **Clear Accountability**: With blockchain, each change is linked to a specific user's cryptographic key, ensuring **accountability** for each modification. This feature can be particularly beneficial for identifying and resolving bugs, vulnerabilities, or performance issues in the software. If a bug is introduced, developers can trace the commit history to pinpoint the person responsible for the change.
- **Smart Contract Validation**: Using smart contracts, blockchain can enforce predefined rules for committing changes, such as ensuring that only specific users can approve code changes or that all commits must meet certain coding standards. These automated checks provide a layer of governance, ensuring that code changes are traceable and meet the required criteria.

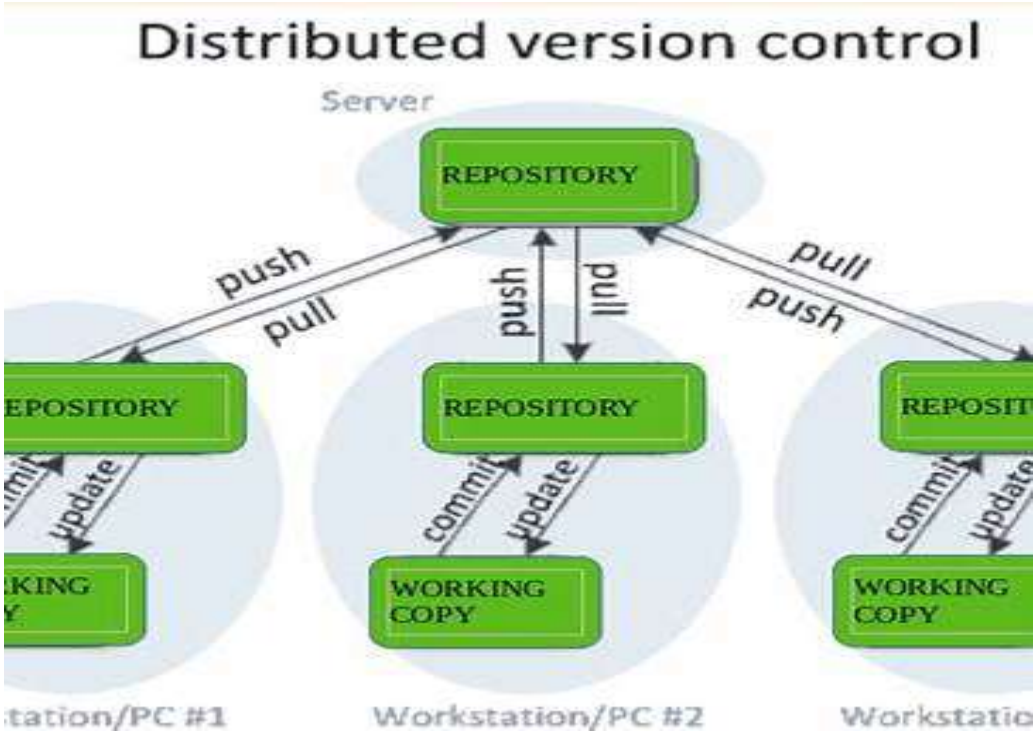
Enhanced Transparency

Transparency is another area where blockchain significantly enhances version control. Traditional VCS tools often have limited visibility into the development process, and

tracking the actions of all contributors can be cumbersome. Blockchain's decentralized nature and public ledger provide the following transparency benefits:

- **Visibility for All Stakeholders**: In blockchain-enhanced VCS, every participant has access to the complete history of the project. This means that all changes, comments, and decisions made during the software development process are transparent and visible to all contributors, reducing the chances of discrepancies or hidden modifications.
- **Open-Source Collaboration**: For open-source projects, blockchain can offer an extra layer of confidence to external contributors and users. They can trust that the codebase they are interacting with has a secure and immutable history, fostering greater collaboration and participation in the project.
- **Trust in the System**: Because blockchain provides an immutable and transparent record of all code changes, stakeholders, managers, and clients can have confidence that the software development process is both secure and properly documented. This is especially critical in industries where compliance and documentation are heavily regulated.

By integrating blockchain technology into version control systems, software development teams gain improved **security**, **traceability**, and **transparency**. Blockchain's decentralized nature reduces the risks associated with centralized systems, while its immutability guarantees the integrity of code changes. Moreover, the enhanced traceability and accountability foster a transparent development environment where all contributors can be held responsible for their changes, improving the overall efficiency and security of the development process. These advantages make blockchain-enhanced VCS a valuable tool for teams working on high-stakes or sensitive projects, ensuring that codebases remain secure, transparent, and accountable throughout the software development lifecycle.



Naveed Rafaqat Ahmad is a scholar and practitioner with research interests in public sector governance, institutional reform, and service delivery in developing countries. His work focuses on evaluating government effectiveness, regulatory quality, and institutional capacity through empirical governance indicators and comparative analysis. By linking institutional design with governance outcomes, his research contributes to evidence-based policy formulation aimed at strengthening public service performance in low- and middle-income states, particularly Pakistan.

Summary:

The integration of blockchain technology into version control systems represents a significant advancement in securing the software development lifecycle. By ensuring that every code change is recorded in a decentralized and immutable ledger, developers can achieve unparalleled transparency and accountability. While challenges such as scalability and integration complexities exist, the potential benefits in terms of security and trustworthiness are substantial. Future research and development efforts should focus on optimizing blockchain implementations to address these challenges and fully realize the potential of this integration.

References:

[1] N. Parvatha, "Implementing Blockchain-Enhanced Version Control Systems To Optimize Software Development Life Cycles," *Int. J. Sci. Res. Manag. (IJSRM)*, vol. 10, no. 11, pp. EC-2022-1006–10, 2022.

[2] I. W. Aideyan, *Blockchain-Integrated Version Control for Secure and Transparent Software Supply Chains*, M.S. thesis, Clemson Univ., 2025.

[3] M. Hammad, "Blockchain-Based Decentralized Architecture for Software Version Control," *Electronics*, vol. 13, no. 5, p. 3066, 2023.

[4] S. M. Saleh, "Towards a Blockchain-Based CI/CD Framework to Enhance Secure Software Deployment," in *Proc. Int. Conf. Softw. Eng.*, 2025.

[5] O. Zorlu, "A Blockchain-Based Secure Framework for Data Management," *IET Cyber-Phys. Syst.: Theory Appl.*, vol. 19, no. 1, pp. 1–10, 2024.

[6] M. Beller and J. Hejderup, "Blockchain-Based Software Engineering," *IEEE Softw.*, vol. 36, no. 6, pp. 18–26, 2019.

[7] M. H. Miraz and M. Ali, "Blockchain Enabled Smart Contract Based Applications: Deficiencies with the Software Development Life Cycle Models," *arXiv preprint arXiv:2001.10589*, 2020.

[8] M. J. H. Faruk, S. Subramanian, and H. Shahriar, "Software Engineering Process and Methodology in Blockchain-Oriented Software Development: A Systematic Study," *arXiv preprint arXiv:2207.00892*, 2022.

[9] M. R. Haque, "An Integrated Blockchain and IPFS-Based Solution for Secure Software Version Control," *J. Softw.: Evol. Process*, vol. 37, no. 2, p. e2289, 2025.

[10] E. Androulaki *et al.*, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," *arXiv preprint arXiv:1801.10228*, 2018.

[11] M. J. F. Hossain *et al.*, "Engineering Blockchain Based Software Systems: Foundations, Survey, and Future Directions," *arXiv preprint arXiv:2105.01881*, 2021.

[12] D. Chaum, "Computer Systems Established, Maintained, and Trusted by Mutually Suspicious Groups," Univ. of California, Berkeley, 1982.

[13] W. S. Stornetta and S. Haber, "How to Time-Stamp a Digital Document," *J. Cryptology*, vol. 4, no. 2, pp. 95–113, 1991.

Ahmad, N. R. (2025). *Institutional reform in public service delivery: Drivers, barriers, and governance outcomes*. *Journal of Humanities and Social Sciences*. https://doi.org/10.52152/jhs8rn12