Fireblocks

Secure Multi-Party Computation Framework

Multi-party computation (MPC) is trusted by leading organizations to secure digital asset wallets from external attackers and insider threats. MPC creates a truly secure environment for storing, transferring, and issuing digital assets by removing a single point of compromise throughout the key lifecycle.

At Fireblocks, we are committed to advancing the application of MPC to secure digital assets, leveraging our Secure MPC framework. As pioneers in the field, we established this security framework and are sharing it with fellow cryptographers to help other MPC vendors improve their products and increase security in the digital asset industry.

A PROACTIVE APPROACH TO SECURING MPC

1. Utilize MPC as part of a multi-layer security philosophy.

As we've seen over the years, the best defense against cybercriminals is a multi-layered security approach that can provide redundancy in the event that one of the security controls fails. There is no silver bullet for security, and organizations should layer MPC and hardware defenses to protect all attack surfaces and eliminate the reliance on a single security technology.

2. Use the latest peer-reviewed MPC algorithms.

Stay up-to-date on MPC and academic advances in theoretical and applied cryptography to ensure you use the most secure MPC protocols and implementations. Examine known vulnerabilities in the underlying algorithms to understand the nature and impact presented in technical reports and publications. Assess the research method's rigor, the algorithms' soundness, and relevance in real-world scenarios. Engage with the authors for an open discussion and the wider community to better understand the vulnerabilities.

3. Be thorough when following technical instructions in academic papers

While technical papers are valuable resources and act as a blueprint for how to build your product, engineers must pay special attention to cryptographic nuances while implementing a peer-reviewed algorithm. Missing steps could create vulnerabilities in your MPC implementation. Always perform thorough evaluations and rigorous testing before implementing a new MPC algorithm in your product. Ensure your implementation enforces the same guarantees the researchers specified in their technical paper (e.g., instruction on Lindell17 paper to halt on execution) and confirm your protocol follows all the steps described (e.g., fully implementing Zero-Knowledge Proofs in BitGo). Establish communication channels between the cryptographic and engineering teams to ensure a shared understanding, collaboration, efficiency, and continuous improvement.

4. Be proactive with security audits.

Conduct internal and third-party audits. For an internal audit, your internal cryptography team can work with engineers to challenge their code and assumptions and participate in design reviews. The cryptography team should conduct code reviews to verify algorithms and protocols are correctly implemented and spot nuances potentially missed during the development process while ensuring security, correctness, and optimal performance.

For third-party audits, invite security auditing firms to review your code and the underlying cryptography and conduct them regularly or whenever significant changes occur to the system. Third-party audits often provide an objective perspective, validate claims, or challenge basic assumptions that your team might miss, ensuring security and credibility.

5. Set up a public bug bounty program.

Setting up a bug bounty program can be a valuable and proactive approach to improving the security of your code as it promotes continuous improvement. It incentivizes security researchers to review your code, identify threats and report findings to you. Be sure to make your policy clear and reward for high-severity vulnerabilities.

6. Take your implementation open-source and referenceable.

Releasing your MPC implementation as an open-source project so customers and researchers can inspect it has multiple benefits. Open-sourcing code promotes transparency and attracts a community of researchers, developers, and users to share knowledge and advance software security collectively. Individuals and teams can inspect your code and choose to push fixes and suggestions to the open-source project directly. Open source allows anyone (customers and cryptographers) to inspect and find bugs or issues themselves and report back to the community and contribute to your bug bounty program.

7. Conduct active exploration to proactively detect misuse.

Apply active exploration to investigate and analyze protocol behaviors and security processes thoroughly. The goal is to understand how the protocol behaves in certain situations and proactively monitor the system to observe misuse. When a team discovers a vulnerability in an MPC protocol, it is essential to have an incident response strategy or communication plan in place. Establish a regular and proactive communication channel to alert internal security operations and customers about critical vulnerabilities and real-time incident updates. Promptly inform customers about significant security events impacting their systems and data to foster trust and risk management.

Security for MPC technology requires a proactive approach. Following these recommendations can build a safer future for digital technologies. Learn more at **Fireblocks.com**