FYEO

Aureus Ox

Security Review Update: Mar 25, 2025

Reviewer: FYEO Security Team

Aureus Ox Security Review Update

New security issues, 1

After the development team implemented the latest updates, FYEO conducted a review of the modifications. The primary goal of this evaluation was to ensure the continued robustness of the wallet's security features, safeguarding user funds and maintaining the overall integrity of the wallet.

General Updates:

The wallet update involves adding XRPL (XRP Ledger) functionality and transaction signing to the iOS application. Currently, the application uses Web3Auth and its SDKs for Ethereum (EVM-based) transaction signing, and this update extends that functionality to XRPL transactions. The update includes the creation of an XRPL TSS (Threshold Signature Scheme) account, which is modeled after the existing Ethereum TSS account object but adapted for XRPL-specific encoding and transaction formatting requirements. A key aspect of the update is implementing custom encoding formats needed for XRPL, particularly for signature and transaction formatting, to ensure compatibility with the XRPL ecosystem.

The account manager, which is an existing component used for EVM-based transaction signing, is updated to support XRPL transactions by adding a generic function that takes an XRPL transaction instead of an EVM-based transaction. Since both XRPL and EVM use SECP256K1 signatures, the signing process remains similar, with the primary difference being the formatting adjustments required for XRPL.

A new object called XRPLTSS account has been created to handle XRPL transaction signing. This object takes a transaction, converts it into a hash to be signed, and then signs half of the hash value, as required by XRPL. Public key store functionality has also been introduced to handle the derivation of XRPL addresses. Additionally, a configuration utility has been implemented to manage URLs and strings specific to the XRP Ledger.

Specific Security Changes:

`AccountManager`: Manages user accounts via Web3Auth and tKey-MPC, focusing on threshold key reconstruction, secure factor management, and TSS account recovery for both Ethereum and XRPL chains.

`XRPLAccountProtocol`: Defines interfaces for XRPL TSS account operations, including signing transactions, TSS key bootstrapping, and handling XRPL-specific cryptographic flows.

`XRPLTssAccount`: Implements XRPL TSS account logic for signing and constructing transactions via TSS clients. Integrates tightly with the TSS infrastructure and ensures valid signature creation and verification for XRPL.

`XRPLPublicKeystore`: Provides functionality to validate XRPL public keys, compress them, and derive XRPL account addresses, enforcing strict secp256k1 compliance.

`XRPLClient`: Establishes websocket connections to XRPL networks, offering simple access to devnet, testnet, and mainnet environments.

The system uses a clear separation of concerns between account management, TSS signing, XRPL transaction handling, and public key management. It enforces strict cryptographic validation and offers multi-factor recovery options through device shares and mnemonic-based factors. The use of Web3Auth and tKey-MPC demonstrates adherence to modern decentralized identity and key management practices.

Incomplete test coverage

Finding ID: FYEO-AUREUS-OX-01
Severity: Informational
Status: Remediated

Description

The current test coverage is not complete and covers only basic positive scenarios.

Proof of Issue

File name: OxenFlow/OxenFlowTests/XRPL/TestXRPLPublicKeystore.swift

Tests are written correctly, basic checks are present and cover important scenarios: - positive scenario with a valid point on secp256k1. - case when the point lies on the wrong curve (secp256r1 instead of secp256k1) - case with incorrect x value (shortened value)

But a case can also be added where a point is on secp256k1 but is invalid, i.e., it fails the `y^2 ≡ x^3 + 7 (mod p)` check.

Proof of Issue

File name: OxenFlow/OxenFlowTests/XRPL/Tss/TestXRPLTssAccount.swift

The tests only cover positive scenarios, but it is worth adding negative scenarios and additional checks to the existing tests: - case where `derEncodeSignature()` on other `r` and `s` gives a different result than the hardcoded value from mock. this confirms that the encoding is dynamic, not static - a check that the `signingPubKey` and `txnSignature` fields are indeed set in `xrplTssAccount.constructSignedTransaction()` should be added to the existing `testTssAccount()` test - check that `constructSignedTransaction()` throws an error if `sign()` fails

Severity and Impact Summary

Insufficient test coverage can cause bugs in the code to go undetected, especially in critical parts such as address generation (XRPLPublicKeystore) and transaction signing (XRPLTssAccount).

Recommendation

Improve the test coverage.

Commit Hash Reference:

For transparency and reference, the security review was conducted on the specific commit hash for the Aureus Ox iOS repository. The commit hash for the reviewed versions is as follows: 3c7868c042d32aed754f42e5089bb2afb2ccfbd1

Commit hash of remedation: b69ee462f8cadeb92db7cbf35a48a50207de7337

Conclusion:

In conclusion, the security aspects of the Aureus Ox iOS wallet remain robust and unaffected by the recent updates. Users can confidently interact with the wallet, assured that their funds are well-protected. The commitment to security exhibited by the development team is commendable, and we appreciate the ongoing efforts to prioritize the safeguarding of user assets.