



XFY Protocol – Audit Remediation Response

XFY Protocol – Audit Remediation Response

Medium Severity Findings

Finding #01: Inconsistent token unit handling between initial mint and ongoing mint

Finding #02: CCIP Pool burn will revert unless granted burn permissions

Finding #03: CCIP_ADMIN storage is redundant and potentially misleading

Finding #04: Role naming inconsistency

Acknowledged / Low-Risk Findings

Finding #05: Re-entrancy risk in _lockOrBurn

Finding #06: Third-party dependency risks

Centralization Risks

Centralized Privileges & Arbitrary Mint/Burn

Conclusion



XFY Protocol – Audit Remediation Response

To: InterFi Audit Team

Re: Security Audit Report for XFY Smart Contracts (Report Date: December 10, 2025)

Submitted By: XFY Protocol Team

Date: December 14, 2025

Thank you for your thorough security review of the XFY smart contracts. We have carefully reviewed all findings and implemented the necessary changes to address the identified risks. Below is our point-by-point response.



Medium Severity Findings

Finding #01: Inconsistent token unit handling between initial mint and ongoing mint

Status:  Resolved

Action Taken:

- Standardized all token amounts to be expressed in the **smallest unit** (i.e., scaled by `10^decimals`).
- Updated constructor documentation to explicitly state that `initialSupply` must be provided in scaled form (e.g., `1_000_000 * 10**18` for 1M tokens).
- Added consistent NatSpec comments to `mint()` and `burn()` functions reinforcing this convention.
- Documented this requirement prominently in the project README.

Evidence:

- `XFYToken.sol` : [Lines 3–6 (notice), 58–63 (constructor doc)]
 - README: “💡 All mint/burn amounts... are expressed in the smallest token unit...”
-

Finding #02: CCIP Pool burn will revert unless granted burn permissions

Status:  Resolved

Action Taken:

- Confirmed that `XFYTokenPool._lockOrBurn()` requires the pool to hold `CCIP_MINT_BURN_ROLE`.
- Exposed a dedicated function `grantCcipMintBurnRole(address)` for secure role assignment.
- Added explicit deployment instructions in the README requiring this step via governance timelock.

Evidence:

- `XFYToken.sol` : `grantCcipMintBurnRole()` function (lines 98–101)
 - README: “⚠️ Critical Configuration via Timelock” section
-

Finding #03: `CCIP_ADMIN` storage is redundant and potentially misleading

Status:  Not Applicable / Design Improved

Action Taken:

- The codebase does not define a standalone `CCIP_ADMIN` storage variable.
- Instead, we implement `getCCIPAdmin()` as a view function that returns the first member of `DEFAULT_ADMIN_ROLE`, aligning with OpenZeppelin's `AccessControl` pattern and Chainlink CCIP registration requirements.
- This avoids redundancy and ensures consistency with role-based access control.

Evidence:

- `XFYToken.sol` : `getCCIPAdmin()` function (lines 70–78)
-

Finding #04: Role naming inconsistency

Status:  Resolved

Action Taken:

- Corrected the role definition to ensure name-hash alignment:

```
1 bytes32 public constant CCIP_MINT_BURN_ROLE = keccak256("CCIP_MINT_BURN_ROLE");
```

- This eliminates potential confusion in deployment scripts or tooling.

Evidence:

- `XFYToken.sol` : Line 18
-

Acknowledged / Low-Risk Findings

Finding #05: Re-entrancy risk in `_lock0rBurn`

Status:  Acknowledged & Mitigated by Design

Response:

- The `_lock0rBurn` function calls `burn(amount)` on the native `XFYToken`, which

performs a standard internal `_burn` with no external calls, callbacks, or hooks.

- The token is non-upgradeable and used exclusively by the protocol's own pool.
- Risk is negligible under intended usage.

Finding #06: Third-party dependency risks

Status:  Acknowledged

Response:

- We rely on well-audited, industry-standard libraries (OpenZeppelin v5.5+, Chainlink CCIP).
 - The team commits to monitoring security advisories and applying updates as needed.
-



Centralization Risks

Centralized Privileges & Arbitrary Mint/Burn

Status:  Mitigated

Action Taken:

- Introduced a **24-hour TimelockController** (`XFYTimelockController.sol`) for all privileged operations.
- Transferred ownership/admin roles of `XFYToken` and `XFYTokenPool` to the timelock contract at deployment.
- All sensitive actions (role grants, ownership acceptance, cross-chain config) now require:
 - a. Proposal by multisig (proposer),
 - b. 24-hour delay,
 - c. Execution by multisig (executor).
- The `CCIP_MINT_BURN_ROLE` is only granted to the `XFYTokenPool` contract—never to EOAs.

Evidence:

- `XFYTimelockController.sol`

- README: “👤 Governance Security Model” section
-

✅ Conclusion

All medium–severity findings have been **fully resolved**. Centralization risks have been **substantially mitigated** through the implementation of timelocked, multisig–governed administration. The codebase and documentation now reflect security best practices for a production–ready CCIP–compatible token.

We appreciate InterFi’s valuable feedback and believe these changes significantly enhance the protocol’s robustness and trustworthiness.

Sincerely,

The XFY Protocol Team

Security Contact: zane3412x@gmail.com