📝 XFY Protocol – Audit Remediation Response

📝 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:

```
solidity

编辑


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 `_lockOrBurn`

Status: ✅ **Acknowledged & Mitigated by Design**

Response:

- The `_lockOrBurn` 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:

- Implemented a governance timelock (`XFYTImelockController.sol`) with a **7-day minimum delay** for all privileged operations, as designed from inception.
- Ownership and admin roles of both `XFYToken` and `XFYTokenPool` are transferred to the timelock contract at deployment.
- All sensitive actions — including granting `CCIP_MINT_BURN_ROLE`, accepting pool ownership, and configuring cross-chain parameters — require:

  a. Proposal by a multisig wallet (holding `PROPOSER_ROLE`),
  b. **7-day minimum delay**,
  c. Execution by an authorized executor (typically the same multisig).
- The `CCIP_MINT_BURN_ROLE` is granted exclusively to the `XFYTokenPool` contract — never to externally owned accounts (EOAs).

This 7-day delay provides a robust window for public review and emergency response, aligning with security best practices for cross-chain token protocols.

Evidence:

- `XFYTImelockController.sol` (deployed with `minDelay = 7 * 24 * 3600`)
- README: "🔐 Governance Security Model" section

✅ Conclusion

All medium-severity findings have been fully resolved. Centralization risks have been substantially mitigated through the implementation of a **7-day timelocked governance model as originally designed**. 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, transparency, and trustworthiness.

Sincerely,
The XFY Protocol Team
Security Contact: zane3412x@gmail.com