Quantstamp Security Assessment Certificate

November 6th 2020 — Quantstamp Verified

POA Network 5.4.1

This security assessment was prepared by Quantstamp, the leader in blockchain security

Executive Summary

Type	Cross-chain bridge
Auditors	Ed Zulkoski, Senior Security Engineer Sebastian Banescu, Senior Research Engineer Fayçal Lalidji, Security Auditor
Timeline	2020-09-14 through 2020-11-05
EVM	Muir Glacier
Languages	Solidity
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	Token Bridge Docs
Documentation Quality	Medium
Test Quality	High

Source Code

Repository	Commit
tokenbridge-contracts	feb0ba5

Goals

- Can funds be locked or stolen?
- Do function have proper access-control?
- Do the message passing schemes behave as intended?

Total Issues	**22** (12 Resolved)
High Risk Issues	0 (0 Resolved)
Medium Risk Issues	0 (0 Resolved)
Low Risk Issues	**9** (6 Resolved)
Informational Risk Issues	**7** (3 Resolved)
Undetermined Risk Issues	**6** (3 Resolved)

High Risk	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
Medium Risk	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
Low Risk	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
Undetermined	The impact of the issue is uncertain.

Unresolved	Acknowledged the existence of the risk, and decided to accept it without engaging in special efforts to control it.
Acknowledged	The issue remains in the code but is a result of an intentional business or design decision. As such, it is supposed to be addressed outside the programmatic means, such as: 1) comments, documentation, README, FAQ; 2) business processes; 3) analyses showing that the issue shall have no negative consequences in practice (e.g., gas analysis, deployment settings).
Resolved	Adjusted program implementation, requirements or constraints to eliminate the risk.
Mitigated	Implemented actions to minimize the impact or likelihood of the risk.

Summary of Findings

During the audit, several issues of varying severity were uncovered as listed below. These issues included insufficient input sanitization, as well as several findings that require input from the POA Network team, as the intended semantics of certain code snippets were not always clear. In general, although some functions are well documented, a large number of non-trivial functions do not have sufficient documentation, making them difficult to assess. Further, the inheritance structure and general architecture could benefit from additional developer documentation, possibly including UML architecture diagrams. Finally, although the test suite appears to be extremely robust, we were unable to run test coverage using the provided scripts; the errors we observed are mentioned below.
Update: We have reviewed all PRs and responses from the POA Network team. All previously found issues have been either fixed, acknowledged, or mitigated.

ID	Description	Severity	Status
QSP-1	Error prone initialization	Low	Acknowledged
QSP-2	Inconsistent handling of minted amounts	Low	Acknowledged
QSP-3	`onlySystem` modifier is always false	Low	Acknowledged
QSP-4	Missing default branch in `switch`-statement	Low	Fixed
QSP-5	Missing input argument validation	Low	Fixed
QSP-6	Off-by-one error when handling limits	Low	Fixed
QSP-7	`permit()` does not validate ECDSA parameters	Low	Fixed
QSP-8	ERC667 Compliance	Low	Mitigated
QSP-9	Missing input validation for `_getFee` and `_setFee`	Low	Fixed
QSP-10	Privileged Roles and Ownership	Informational	Acknowledged
QSP-11	Block Timestamp Manipulation	Informational	Fixed
QSP-12	The `SELFDESTRUCT` EVM instruction could be removed in the near future	Informational	Acknowledged
QSP-13	Missing error handler in `readName`, `readSymbol` and `readDecimals` functions	Informational	Acknowledged
QSP-14	TODOs in the code	Informational	Acknowledged
QSP-15	`distributeFee` will revert if there are no reward accounts	Informational	Fixed
QSP-16	Stale `expirations` mapping data	Informational	Fixed
QSP-17	Leftover tokens may transfer to new `receiverInXDai` if updated	Undetermined	Acknowledged
QSP-18	`gasPriceSpeed` value is never used	Undetermined	Fixed
QSP-19	Reward function is underspecified	Undetermined	Fixed
QSP-20	Potential duplicate entries in reward accounts list	Undetermined	Acknowledged
QSP-21	Unclear if `_isDestinationChainIdValid()` semantics are as-intended	Undetermined	Fixed
QSP-22	Gas Usage / `for` Loop Concerns	Undetermined	Acknowledged

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Possible issues we looked for included (but are not limited to):

- Transaction-ordering dependence
- Timestamp dependence
- Mishandled exceptions and call stack limits
- Unsafe external calls
- Integer overflow / underflow
- Number rounding errors
- Reentrancy and cross-function vulnerabilities
- Denial of service / logical oversights
- Access control
- Centralization of power
- Business logic contradicting the specification
- Code clones, functionality duplication
- Gas usage
- Arbitrary token minting

Methodology

The Quantstamp auditing process follows a routine series of steps:

1. Code review that includes the following
   i. Review of the specifications, sources, and instructions provided to Quantstamp to make sure we understand the size, scope, and functionality of the smart contract.
   ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
   iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Quantstamp describe.
2. Testing and automated analysis that includes the following:
   i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
   ii. Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Slither v0.6.12
- Mythril v0.22.8

Steps taken to run the tools:

1. Installed the Slither tool: `pip install slither-analyzer`
2. Run Slither from the project directory: `slither .`
3. Installed the Mythril tool from Pypi: `pip3 install mythril`
4. Ran the Mythril tool on each contract: `myth -x path/to/contract`

Findings

QSP-1 Error prone initialization

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `HomeAMBErc20ToNative.sol`, `HomeStakeTokenMediator.sol`

Description: The `HomeAMBErc20ToNative` contract has:

1. A public `initialize` method, which can be called only once
2. An external `rewardableInitialize` method that calls `initialize`.

If the upgradability owner calls `initialize` by mistake, then they will not be able to call `rewardableInitialize` anymore and the reward address list will be uninitialized and no new reward addresses will be addable to it.
A similar situation occurs inside `HomeStakeTokenMediator.sol`. However, there the fee and the `blockRewardContract` can still be initialized later.

Recommendation: Check if the reward address list has been initialized at the beginning of `initialize` inside `HomeAMBErc20ToNative`. For the second issue inside `HomeStakeTokenMediator`, check that the block reward contract and the fee have been initialized.
Update from POA Network team: The code is developed as so two different deployment are possible with usage of the one contract `HomeAMBErc20ToNative`: the first allows to have a simple bridge, the second one allows to collect fees here. So, the situation when the method initialize is called without the reward address list is possible. The similar situation is applicable to `HomeStakeTokenMediator`. It could be deployed in both ways with and without support of fees.

QSP-2 Inconsistent handling of minted amounts

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `BlockReward.sol`

Description: The `BlockRewards.setMinted` function sets 4 value in the `uintStorage` array inherited from `EternalStorage`:

1. `MINTED_FOR_ACCOUNT_IN_BLOCK`
2. `MINTED_FOR_ACCOUNT`
3. `MINTED_IN_BLOCK`
4. `MINTED_TOTALLY`

For the first item in the list above, the previous value of the array is overwritten with the value of `_amount`, while for the next 3 items the value of `_amount` is added to the previous value of the array. This is inconsistent and could lead to discrepancies in the amounts being stored.

Recommendation: Use the same way of adding the `_amount` to the previous value of the array for `MINTED_FOR_ACCOUNT_IN_BLOCK`, as is being used for the other 3 items.
Update from POA Network team: The goal to keep a value in the storage slots which "addresses" encoded with `MINTED_FOR_ACCOUNT_IN_BLOCK` is to track amount of xDai tokens minted for a particular account in a particular block. As per the current code amount comes to the `_setMinted` method from the reward method. The amount is taken from the rewards array. The value in this array is got from the slots encoded with `EXTRA_RECEIVER_AMOUNT`. And these slots are initialized in the `addExtraReceiver` method:
https://github.com/poanetwork/tokenbridge-contracts/blob/feb0ba555cb8bf74f7fa21911120d878715a0a84/contracts/upgradeable_contracts/amb_erc20_to_native/BlockReward.sol#L71.
The code displays that this value is already accumulated for a particular account. And slots encoded with `EXTRA_RECEIVER_AMOUNT` live only during one block:
https://github.com/poanetwork/tokenbridge-contracts/blob/feb0ba555cb8bf74f7fa21911120d878715a0a84/contracts/upgradeable_contracts/amb_erc20_to_native/BlockReward.sol#L116.
So, slots encoded with `MINTED_FOR_ACCOUNT_IN_BLOCK` will contain xDai tokens minted for a particular account in a particular block.

QSP-3 `onlySystem` modifier is always false

Severity: *Low Risk*

Status: Acknowledged

File(s) affected: `BlockReward.sol`

Description: The `require(msg.sender == address(0))` inside of the `onlySystem` modifier will always fail because the condition will be false. Therefore, the `reward` method cannot be called.

Recommendation: Fix the `require` statement such that the `msg.sender` is compared to a system-reserved address instead of `address(0)`. Alternatively, document why this modifier is needed and why it always returns false.
Update from POA Network team: These contracts were designed for working on top of a modified Quorum client where the system reserved address was hardcoded to be `0x00..00`. Therefore, it returns true if and only if the caller is the system itself: https://github.com/poanetwork/quorum/blob/0e922bd8412b2c2019624c82a2b129f5f580d8c2/core/state_processor.go#L187.

QSP-4 Missing default branch in `switch`-statement

Severity: *Low Risk*

Status: Fixed

File(s) affected: `ArbitraryMessage.sol`

Description: The `switch`-statement inside the `ArbitraryMessage.unpackData` function has no default branch, which would be taken in case the byte representing the data type in the header would be different from: 0x00, 0x01 or 0x02. This would indicate a malformed header and could break the assumption written in the comment right after the `switch`-statement, i.e.: "at this moment `srcdataptr` points to `sourceChainId`".

Recommendation: Add a default branch to the `switch`-statement where the proper handling of malformed headers should be performed, e.g. reverting the transaction.
Update: This has been addressed as of PR 527.

QSP-5 Missing input argument validation

Severity: *Low Risk*

Status: Fixed

File(s) affected: `PermittableToken.sol`

Description:

1. The value of the `_to` input argument of type `address` is not checked inside the `Claimable.claimValues` function. Therefore, tokens could be sent to the `0x0` address by calling this method. Use the `validAddress` modifier to check this directly, instead of expecting that developers who extend this contract will remember to call it whenever they call `claimValues`.
2. The value of the `_receiver` input argument is not checked inside of the `fixMediatorBalance` function, which is re-implemented in several contracts from the `upgradable_contracts/` subdirectory. Therefore, the `_receiver` could be set to `0x0`, which is not checked by any of the internal and private functions that use its value.

Recommendation: Add input argument validation for the aforementioned functions.
Update: This has been addressed as of PR 510.

QSP-6 Off-by-one error when handling limits

Severity: *Low Risk*

Status: Fixed

File(s) affected: `BaseMediatorFeeManager.sol`

Description: The constant called `MAX_REWARD_ACCOUNTS` is not attainable via the `addRewardAccount`, due to the following statement:

```
require(rewardAccounts.length.add(1) < MAX_REWARD_ACCOUNTS);
```

However, the maximum number of reward accounts is attainable via the constructor, due to the following statement:

```
require(_rewardAccountList.length > 0 && _rewardAccountList.length <= MAX_REWARD_ACCOUNTS);
```

Therefore, one of these conditions is off-by-one with the differing comparisons being used: strictly-less-than and less-than-or-equal.
Note that inside the `BaseBridgeValidators` contract, the constant `MAX_VALIDATORS`, which indicates the maximum number of bridge validators, is attainable. That is, there can be that many validators.

Recommendation: Align the logic that constrains the number of reward accounts between the constructor and the `addRewardAccount` method.
Update: This has been addressed as of PR 525.

QSP-7 `permit()` does not validate ECDSA parameters

Severity: *Low Risk*

Status: Fixed

File(s) affected: `PermittableToken.sol`

Description: The function does not validate either of the `s` and `v` parameters. values. See ECDSA.sol.

Recommendation: Add checks for the `s` and `v` parameters.
Update: This has been addressed as of PR 528.

QSP-8 ERC667 Compliance

Severity: *Low Risk*

Status: Mitigated

File(s) affected: `ERC677BridgeToken.sol`, `PermittableToken.sol`

Description: The implemented ERC677 Tokens are extending `transferAndcall` logic to `transfer` and `transferFrom`, which is not compliant with the ERC667 standard. Using `callAfterTransfer` inside `transfer` or `transferFrom` function is problematic since it is not an intended behavior by the ERC667 standard. Even if the function does not throw, if the recipient implements `onTokenTransfer`, the transaction may behave differently since the user might not be expecting a call to the implemented `onTokenTransfer`.
Please note that even if this implementation enhances the UX for cross-chain token transfers, the implemented tokens will still be widely used to interact with their respective blockchain ecosystems.

Recommendation: Only call `onTokenTransfer` through `transferAndCall`.
Update: This has been addressed as of PR 530. The code is now compliant except if `_to` is a bridge contract.

QSP-9 Missing input validation for `_getFee` and `_setFee`

Severity: *Low Risk*

Status: Fixed

File(s) affected: `RewardableBridge.sol`, `BaseFeeManager.sol`

Description: The functions `_getFee` and `_setFee` do not validate `_feeType` to be equal to one of the two possibilities `HOME_FEE` or `FOREIGN_FEE`. Note that `_feeType` is a `bytes32` and therefore for any error in the input, the fee type will be considered as `FOREIGN_FEE`.
The same described issue is applicable for `BaseFeeManager.calculateFee`; any error in `_feeType` input will be considered as `FOREIGN_FEE` and `getForeignFee` will be used.

Recommendation: Implement a `require` statement that will throw the call in case if the fee type input is erroneous.
Update: This has been addressed as of PR 529.

QSP-10 Privileged Roles and Ownership

Severity: *Informational*

Status: Acknowledged

File(s) affected: `BaseMediatorFeeManager.sol`, `ERC677BridgeTokenRewardable.sol`, `HomeFeeManagerAMBErc20ToNative.sol`, `ERC677BridgeToken.sol`, `BasicAMBErc677ToErc677.sol`

Description: Smart contracts will often have `owner` variables to designate the person with special privileges to make modifications to the smart contract.
The owner of the `ERC677BridgeTokenRewardable` contract is allowed to perform several privileged actions such as:

- Setting the `blockRewardContract` any number of times
- Setting the `stakingContract` any number of times if the balance of the newly added `stakingContract` is zero.

The owner of the `ERC677BridgeToken` contract is allowed to perform several privileged actions such as:

- Transfer all the tokens (ERC-20 or native) to any desired address at any point in time

The owner of the `BaseMediatorFeeManager` contract is allowed to perform several privileged actions such as:

- Setting the fee percentage any number of times
- Adding/removing reward accounts.

The owner of the `InterestReceiver` contract is allowed to set (any number of times) the:

- `bridgeContract` from which interest is expected to come
- `receiverInXDai` which is the receiver address in the xDai chain.

The owner of the `HomeFeeManagerAMBErc20ToNative` contract can:

- front-run any transfer over the bridge and set the fee up to `MAX_FEE`.
- add/remove any reward addresses

The upgradability owner of `BasicAMBErc677ToErc677` can perform the following privileged actions:

- Transfer all the tokens (ERC-20 or native) to any desired address at any point in time

Recommendation: This centralization of power needs to be made clear to the users, especially depending on the level of privilege the contract allows to the owner.
Update from the POA Network team: We have the admin roles described in the documentation: https://docs.tokenbridge.net/about-tokenbridge/features/tokenbridge-roles/administrative-groups-and-roles. In order to increase security for the end users we uses the Gnosis Safe governance account that contains 11 participants from different projects as so 6 of these participants must agreed on the configuration changes or contract upgrades.

QSP-11 Block Timestamp Manipulation

Severity: Informational

Status: Fixed

File(s) affected: `PermittableToken.sol`

Description: Projects may rely on block timestamps for various purposes. However, it's important to realize that miners individually set the timestamp of a block, and attackers may be able to manipulate timestamps for their own purposes. If a smart contract relies on a timestamp, it must take this into account.
The `PermittableToken` contract uses the block timestamp to check if the allowance permission given in the `permit` function is still valid when calling `transferFrom`.
Note that malicious miners can manipulate the block timestamp with up to 900 seconds.

Recommendation: Acknowledge that 900 seconds is a tolerable time delta for the allowance permission expiration. Add a warning to the end-user documentation indicating that the expiry timestamp specified as an input parameter to the `permit` function can be manipulated by up to 900 seconds.
Update: This has been addressed as of PR 536.

QSP-12 The `SELFDESTRUCT` EVM instruction could be removed in the near future

Severity: Informational

Status: Acknowledged

File(s) affected: `Sacrifice.sol`

Description: There exists an EIP-2751, which aims to disable the `SELFDESTRUCT` EVM instruction. This EIP is currently in discussion by the Ethereum core developers and supported by Vitalik Buterin. If this EIP is implemented then it will lead the constructor method in the `Sacrifice` contract to perform a no-op. Therefore, the `Address.safeSendValue` function will not work as intended.

Recommendation: Consider alternative solutions, which do not involve the `selfdestruct` function in Solidity.
Update from POA Network team: The EIP is still under discussion. There are few important aspects (e.g. https://github.com/ethereum/EIPs/pull/2751/files#r445865149) that are not answered yet. We will monitor this EIP and change the code when it will be applied.

QSP-13 Missing error handler in `readName`, `readSymbol` and `readDecimals` functions

Severity: Informational

Status: Acknowledged

File(s) affected: `TokenReader.sol`

Description: The `readName`, `readSymbol` and `readDecimals` functions do not contain code that handles the situation where the input `_token` address belongs to a contract that does not have any `name`/`NAME` nor `symbol`/`SYMBOL`, nor `decimals`/`DECIMALS` functions and the `staticcall`s would therefore return zeros, instead of indicating that these functions do not exist and reverting.

Recommendation: Either explicitly handle the case where these functions do not exist in the `_token` contract/EOA or document that this case is explicitly skipped because the function should not revert if `_token` is not a valid ERC20 token.
Update from POA Network team: If neither of `token.name`/`token.NAME`/`token.symbol`/`token.SYMBOL` returns a valid result, the transaction will be reverted. Decimals is assumed to be 0 in case of missing function inside the ERC20 token contract. Such generalized behaviour is needed to cover as many different tokens contracts as possible. For some reasons, different token contracts may not fully comply to the ERC20 token standard.

QSP-14 TODOs in the code

Severity: Informational

Status: Acknowledged

File(s) affected: `BlockReward.sol`

Description: The following TODOs present in the code should be removed: The `BlockReward.bridgesAllowed` method contains the following comment: “These values must be changed before deploy”. It is referring to the fact that the only allowed bridge is the zero address (hardcoded).
This may also relate to the finding "onlySystem modifier is always false".

Recommendation: TODOs present in the code should be removed or resolved/implemented.
Update from POA Network team: The array in `bridgesAllowed` cannot be filled since the bridge address is not known in advance. Instead of this the procedure is the following: an initial version of the block reward contract (proxy + implementation) is set in the genesis, then the bridge contract is deployed, after that the block reward contract implementation is replaced with a new one that contains correct address of the bridge contract.

QSP-15 `distributeFee` will revert if there are no reward accounts

Severity: *Informational*

Status: Fixed

File(s) affected: `BaseMediatorFeeManager.sol`

Description: If all reward accounts are ever removed, any function that invokes `distributeFee()` will fail due to the `div()` on L157.

Recommendation: Consider adding a return-statement to exit the function immediately if there are no rewars accounts.
Update: This has been addressed as of PR 525.

QSP-16 Stale `expirations` mapping data

Severity: *Informational*

Status: Fixed

File(s) affected: `PermittableToken.sol`

Description: If `approve` or `increaseAllowance` is used to set the allowance to `uint256(-1)`, and if `permit` was previously used and expired, `expirations` mapping is not reset to zero, meaning that the requirement included below will throw:

```
require(expirations[_sender][msg.sender] == 0 || expirations[_sender][msg.sender] >= _now());
```

Recommendation: We recommend to reimplement `approve` and `increaseAllowance` to set the expirations mapping to zero if the new allowance value is equal to `uint256(-1)`.
Update: This has been addressed as of PR 533.

QSP-17 Leftover tokens may transfer to new `receiverInXDai` if updated

Severity: *Undetermined*

Status: Acknowledged

File(s) affected: `InterestReceiver.sol`

Description: The `InterestReceiver.onTokenTransfer` function may fail to transfer (relay) the tokens to the xDai receiver as shown by the following code snippet:

```
if (!bridgeContract.call(abi.encodeWithSelector(RELAY_TOKENS, receiverInXDai, finalDaiBalance))) {
            daiToken().approve(address(bridgeContract), 0);
            emit RelayTokensFailed(receiverInXDai, finalDaiBalance);
}
```

If it fails, the DAI tokens are simply kept inside the `InterestReceiver` contract to be relayed on the next call invocation. However, consider the following scenario:

1. the transfer fails for one user;
2. the `receiverInXDai` is changed;
3. another user comes and transfers tokens;
4. the previously failed tokens are sent to a different `receiverInXDai`.

It is not clear if this scenario could be problematic or not.

Recommendation: Clarify if this scenario may be problematic.
Update from POA Network team: It is OK since it will be on the shoulders of the governance account because it is responsible for changing the recipient account address. The `xDai` bridge governance must communicate possible leak of interest with the interest recipient before the parameters changing.

QSP-18 `gasPriceSpeed` value is never used

Severity: *Undetermined*

Status: Fixed

File(s) affected: `ArbitraryMessage.sol`

Description: The comment before the `ArbitraryMessage.unpackData` function indicates that there is an optional field called `gasPriceSpeed` that is stored on 1 byte. Inside the aforementioned function, there is a `switch`-statement where we see that if the data type byte is equal to `0x02` then the optional field called `gasPriceSpeed` is present. However, the value of this field is simply ignored (never used) inside of the function.

Exploit Scenario:

Recommendation: Due to the lack of documentation, we are unsure what optional field called `gasPriceSpeed` represents and how it should be used. Therefore, we recommend documenting its purpose if it is meant to be used. Alternatively, it should be removed if it is (and never will be) used.
Update: This has been addressed as of PR 527.

QSP-19 Reward function is underspecified

Severity: *Undetermined*

Status: Fixed

File(s) affected: `BlockReward.sol`

Description: There are several things that are unclear about the `BlockReward.reward` function:

1. Why are the lengths of its input array parameters supposed to be zero?
2. Why are the receivers different from the benefactors?
3. Why are there 2 consecutive loops with the same iterator range? Shouldn't they be merged into 1 loop?

4. There seems to be an invariant that is not explicitly checked, namely: the sum of `extraAmount`s for all `extraReceiver`s is equal to the sum of `bridgeAmount`s for all allowed bridge addresses.

Recommendation: Add code comments to clarify the questions above. Add an assertion which checks that the 2 sum of receiver amounts is equal to the sum of bridge amounts, after the last loop.
Update: This has been addressed as of PR 534.

QSP-20 Potential duplicate entries in reward accounts list

Severity: *Undetermined*

Status: Acknowledged

File(s) affected: `BaseMediatorFeeManager.sol`

Description: The constructor and `addRewardAccount()` do not check if an address is already in the `rewardAccounts` list. It is not sure if this functionality should be allowed.

Recommendation: Clarify if this functionality is intended. Add checks for duplicates if undesirable.
Update: This has been partially addressed as of PR 536. The input `_rewardAccountList` is still not checked for duplicates inside the constructor, even if `addRewardAccount` contains that check.
Update: This has been addressed as of PR 525.

QSP-21 Unclear if `_isDestinationChainIdValid()` semantics are as-intended

Severity: *Undetermined*

Status: Fixed

File(s) affected: `BasicAMB.sol`

Description: The function `_isDestinationChainIdValid()` checks against `sourceChainId()`, not `destinationChainId()`. Based on the name, it seems `destinationChainId()` is more relevant here.

Recommendation: Clarify if the correct values are compared against in the function.
Update from POA Network team: The AMB contract on every side has two chains ids associated with: the id of the chain this instance of AMB contract is deployed on (source) and the id of the chain this AMB contract is linked to (destination).
The bridge message sending from one chain to another also contains two chain ids in the header: the first one is the id of the chain this message is originated from (source) and the second one is the id of the chain this message is targeted to (destination).
So, as soon as the message appears the bridge contract checks that the message is aimed to this chain: the destination chain id in the message equals of the id of the chain the AMB contract is deployed on (source).

QSP-22 Gas Usage / `for` Loop Concerns

Severity: *Undetermined*

Status: Acknowledged

File(s) affected: `BlockReward.sol`

Description: Gas usage is a main concern for smart contract developers and users, since high gas costs may prevent users from wanting to use the smart contract. Even worse, some gas usage issues may prevent the contract from providing services entirely. For example, if a `for` loop requires too much gas to exit, then it may prevent the contract from functioning correctly entirely. It is best to break such loops into individual functions as possible.
It this context, the function `reward()` loops over `extraReceiversLength()`. It is not clear how large the receivers list can grow.

Recommendation: Clarify if the receivers list can grow large enough to run into gas-limit issues. Perform gas analysis if necessary.
Update from the POA Network team: The `reward(address[],address[])` function is designed in a way that it can only be called by a system reserved address (`0x00..00`). System-reserved calls are executed a little bit differently from the regular EVM transactions and calls. For instance, while such executions the gas usage can be almost unlimited. In our forked version of the Quorum client, that is intended to be used with these contracts, we specified the call gas limit to be `2**64` (https://github.com/poanetwork/quorum/blob/0e922bd8412b2c2019624c82a2b129f5f580d8c2/core/state_processor.go#L190). That's why it is not required to limit the possible high gas usage in the system reserved calls.

Automated Analyses

Slither

• Slither reported several precision issues related to dividing before multiplying in the various `distributeFee()` functions, however we classified these as false positives.

• Several functions ignore the return values of called functions:
  · `ForeignFeeManagerAMBNativeToErc20.onFeeDistribution(address,uint256)` ignores return value by `ERC20Basic(token).transfer(_rewardAddress,_fee)`
  · `HomeFeeManagerMultiAMBErc20ToErc677._distributeFee(bytes32,address,uint256)` ignores return value by `ERC677(_token).transfer(nextAddr,feeToDistribute)`
  · `HomeFeeManagerMultiAMBErc20ToErc677._distributeFee(bytes32,address,uint256)` ignores return value by `IBurnableMintableERC677Token(_token).mint(nextAddr,feeToDistribute)`
  · `FeeManagerNativeToErc.onSignatureFeeDistribution(address,uint256)` ignores return value by `erc677token().mint(_rewardAddress,_fee)`

• Many issues were reported relating to the mock contracts which were not considered.

Update from the POA Network team: In the `transfer` method invoked by the code presented above, our implementation of the ERC20 token is used: https://github.com/poanetwork/tokenbridge-contracts/blob/master/contracts/ERC677BridgeToken.sol. It always returns true if it is not reverted.

Similar is applicable to that version of `MintableToken.sol` used in our code. It always returns true.

So, there is no need to add additional checks in the code since they will not be executed but increase the gas consumption.

Mythril

We were unable to run mythril at this time, likely due to the size of the contracts analyzed.

Code Documentation

1. Each function should contain a short description of its purpose and its input parameter(s) and return value(s) if any. Numerous functions in the code base do not have such code comments, which makes maintenance and auditing more difficult.
2. It is unclear where the return values of the `getTokenInterfacesVersion` originate from in the `PermittableToken` (2.3.0) and `ERC677BridgeToken` (2.2.0) contracts. Note that the minor versions are the only difference that those 2 contracts return for that corresponding function. We recommend adding documentation that describes these values and how they should change.
3. Similarly, the `getBridgeInterfacesVersion` of the `VersionableBridge` contract is 5.1.0, while the same method returns: 1.1.1 in the `BasicAMBErc20ToNative` contract, same 1.1.1 in the `BasicMultiAMBErc20ToErc677` contract, 1.2.1 in the `BasicAMBErc677ToErc677` contract, 1.1.0 in the `BasicStakeTokenMediator` contract, same 1.1.0 in the `BasicAMBNativeToErc20` contract, `5.3.0` in the `VersionableAMB` contract. It seems that some versions are the same for different contracts.
4. It is unclear whether the `blockRewardContract` and the `stakingContract` addresses inside the `ERC677BridgeTokenRewardable` should be changeable after the balance of these 2 contracts is greater than 0. Wouldn't this mean that some stakers would lose their stake? **Update:** This has been resolved with added documentation in PR 536.
5. There seems to be no function that allows a staker to withdraw their stake, after they have called the `ERC677BridgeTokenRewardable.stake` function. Shouldn't there be a way for stakers to withdraw their stakes? **Update from the POA Network team:** As you can see in the code the `stake` method is needed to allow the staking contract (not stakers) to perform transfer tokens from staker's account of the staking contract account. Similar to `transferFrom` but does not require to approve the token transfer for the staking contract. Methods to claim reward or withdraw stake are implemented in the staking contracts: https://github.com/poanetwork/posdao-contracts/blob/master/contracts/base/StakingAuRaTokens.sol.
6. There is a comment inside the `InterestReceiver` contract that states "chi is always >= 10**27, so chai/dai rate is always >= 1". If this is the case, then it is not clear why the subsequent `require` statement needs to exist in the code `require(redeemed >= chaiBalance)`. This seems to be an invariant that should rather be checked using an `assert` statement. **Updated:** fixed as support for CHAI has been disabled.
7. It is unclear whether the implementation of the `claimTokens` function has the correct `require` statements and checks, because of missing code comments and documentation. All implementations seem to check if the `_to` address is valid, but not all check if the `_token` address is valid and the ones that do check the `_token` have various checks.
8. In `TokenReader.sol` on L76 the comment "SYMBOl()" should instead be "SYMBOL()", which importantly influences the keccak256 hash constant. Note that the constant in the code appears to correctly relate to "SYMBOL()". **Update:** fixed.
9. The function `BasicAMB._setChainIds()` would benefit from additional documentation. **Update:** fixed.
10. It is not clear how the default values 1, 100, and 10000 were chosen in `BasicMultiTokenBridge_initializeTokenBridgeLimits()` on L278-287. **Update:** fixed.
11. It is not clear why in `BasicMultiTokenBridge.sol`, `setMaxPerTx()` allows setting to zero but `setMinPerTx()` does not. **Update from POA Network team:** Zero value for the setMaxPerTx method is used to turn off the bridge operations. There is no need to set the minimal amount per tx to zero. 1 wei value could be used instead.

Adherence to Best Practices

1. Add error messages to `require` statements to help end-users understand why a transaction failed. Not displaying any reason for a failed transaction could cause end-user annoyance. Additionally, it may help with troubleshooting deployed contracts. The majority of `require` statements in the current code base do not contain any error message. **Update:** Due to contract size restrictions, this has not been implemented yet. Future iterations may use error codes, as suggested in Issue 494.
2. The state variable `DOMAIN_SEPARATOR` from `PermittableToken.sol` is not in mixedCase.
3. The constant `version` from `PermittableToken.sol` is not in UPPER_CASE.
4. Magic numbers should be replaced with named constants. Instances of this issue:
    1. `DecimalShiftBridge._setDecimalShift` uses the constant 77.
    2. In `ChaiConnector` the `convertDaiToChai` and `_convertChaiToDai` methods use the constant 10000.
5. Commented code should be removed, such as L5-23 inside `Message.sol`. **Update:** fixed.
6. The constructor of `BaseMediatorFeeManager` should check that `_mediatorContract` is non-zero or a contract. **Update:** fixed.
7. The function `InterestReceiver.onTokenTransfer()` is declared to return `bool` but does not have an explicit return value. **Update:** fixed.
8. The constructor in `InterestReceiver.sol` should check that `_receiverInXDai` is non-zero. **Update:** fixed.
9. The function `HomeMultiAMBErc20ToErc677.deployAndHandleBridgedTokens` should check that at least one of `name` or `symbol` are non-zero. **Update:** fixed.
10. Similarly to the previous description, `BaseRewardAddressList._addRewardAddress` cannot be used if the list is not initialized using `_setRewardAddressList`. It would be more intuitive to allow validator or reward addresses to be added through `_addValidator` or `_addRewardAddress` even when the initial list is empty and avoid using `setNextValidator` and `_setNextRewardAddress`. **Update:** this approach was chosen due to gas cost concerns.
11. A mapping can be used to save the array entries' indices for each added reward address using `addRewardAccount` to avoid using a for loop when deleting an entry from the array in `BaseMediatorFeeManager.removeRewardAccount`. The same description applies for `BaseMediatorFeeManager.isRewardAccount` when checking if an account is valid. **Update:** this approach was chosen due to gas cost concerns.

Test Results

Test Suite Results

```
Contract: ForeignAMBErc20ToNative
  initialize
    ✓ should initialize parameters (1877ms)
  getBridgeMode
    ✓ should return mediator mode and interface (42ms)
  claimTokens
    ✓ should work with token different from bridged token (687ms)
    ✓ should also work for native coins (136ms)
```

```
  afterInitialization
    onTokenTransfer
       ✓ should call AMB bridge and burnt tokens (502ms)
       ✓ should be able to specify a different receiver (472ms)
    relayTokens
       ✓ should allow to bridge tokens using approve and transferFrom (447ms)
       ✓ should allow to specify a different receiver without specifying sender (344ms)
       ✓ should fail if user did not approve the transfer (88ms)
       ✓ should fail if value is not within limits (120ms)
    handleBridgedTokens
       ✓ should unlock tokens on message from amb (946ms)
       ✓ should unlock tokens on message from amb with decimal shift of 2 (1015ms)
       ✓ should revert when out of execution limits on message from amb (507ms)
       ✓ should unlock tokens on message from amb with decimal shift of -1 (735ms)
       ✓ should revert when out of execution limits on message from amb (415ms)
    requestFailedMessageFix
       ✓ should allow to request a failed message fix (264ms)
       ✓ should be a failed transaction (407ms)
       ✓ should be the receiver of the failed transaction (252ms)
       ✓ message sender should be mediator from other side (247ms)
       ✓ should allow to request a fix multiple times (463ms)
    fixFailedMessage
       ✓ should fix locked tokens (614ms)
       ✓ should be called by amb bridge (53ms)
    fixFailedMessage with alternative receiver
       ✓ should fix burnt tokens (283ms)
  fixMediatorBalance
     ✓ should allow to fix extra mediator balance (1226ms)
     ✓ should allow to fix extra mediator balance with respect to limits (2231ms)

Contract: HomeAMBErc20ToNative
  initialize
     ✓ should initialize parameters (2251ms)
  rewardableInitialize
     ✓ should initialize parameters for rewardable mediator (2121ms)
getBridgeMode
     ✓ should return mediator mode and interface (39ms)
claimTokens
     ✓ should work with token different from bridged token (418ms)
     ✓ should not work for native coins (138ms)
afterInitialization
  setBlockRewardContract
     ✓ should set block reward contract (140ms)
     ✓ should fail if not a block reward contract (98ms)
     ✓ should fail if not an owner (161ms)
  handle deposited coins
    fallback
       ✓ should accept native coins (264ms)
       ✓ native coins amount should be inside limits  (544ms)
    relayTokens
       ✓ should accept native coins and a different receiver (301ms)
       ✓ native coins amount should be inside limits  (624ms)
    fixFailedMessage
       ✓ should fix locked tokens (627ms)
       ✓ should be called by amb bridge
    fixFailedMessage with alternative receiver
       ✓ should fix burnt tokens (3851ms)
  handleBridgedTokens
     ✓ should add extra receiver tokens on message from amb (1839ms)
     ✓ should unlock tokens on message from amb with decimal shift of 2 (443ms)
     ✓ should unlock tokens on message from amb with decimal shift of -1 (370ms)
     ✓ should revert when out of execution limits on message from amb (119ms)
  requestFailedMessageFix
     ✓ should allow to request a failed message fix (183ms)
     ✓ should be a failed transaction (268ms)
     ✓ should be the receiver of the failed transaction (296ms)
     ✓ message sender should be mediator from other side (161ms)
     ✓ should allow to request a fix multiple times (321ms)
  fixMediatorBalance
     ✓ should fix mediator imbalance (335ms)
     ✓ should fix mediator imbalance with respect to limits (703ms)
fee management
   ✓ change reward addresses (906ms)
  update fee parameters
     ✓ should update fee value (275ms)
     ✓ should update opposite direction fee value (254ms)
    distribute fee for foreign => home direction
       ✓ should collect and distribute 0% fee (551ms)
       ✓ should collect and distribute 2% fee (655ms)
       ✓ should collect and distribute 2% fee between two reward addresses (913ms)
    distribute fee for home => foreign direction
       ✓ should collect and distribute 0% fee (374ms)
       ✓ should collect and distribute 1% fee (342ms)
       ✓ should collect and distribute 1% fee between two reward addresses (371ms)

Contract: ForeignAMBErc677ToErc677
  initialize
     ✓ should initialize (1083ms)
     ✓ only owner can set bridge contract (247ms)
     ✓ only owner can set mediator contract (251ms)
     ✓ only owner can set request Gas Limit (224ms)
  set limits
     ✓ setMaxPerTx allows to set only to owner and cannot be more than daily limit (139ms)
     ✓ setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (124ms)
     ✓ setDailyLimit allow to set by owner and should be greater than maxPerTx or zero (205ms)
     ✓ setExecutionMaxPerTx allows to set only to owner and cannot be more than daily limit (120ms)
     ✓ setExecutionDailyLimit allow to set by owner and should be greater than maxPerTx or zero (197ms)
  getBridgeMode
     ✓ should return arbitrary message bridging mode and interface
  fixAssetsAboveLimits
     ✓ Should revert if value to unlock is bigger than max per transaction (57ms)
     ✓ Should allow to partially reduce outOfLimitAmount and not emit amb event (234ms)
     ✓ Should allow to partially reduce outOfLimitAmount and emit amb event (361ms)
     ✓ Should revert if try to unlock more than available (419ms)
     ✓ Should not be allow to be called by an already fixed message (268ms)
     ✓ Should fail if message didnt increase out of limit amount (71ms)
     ✓ Should fail if not called by proxyOwner (152ms)
  relayTokens
     ✓ should allow to bridge tokens using approve and transferFrom (287ms)
     ✓ should allow to specify a different receiver without specifying sender (341ms)
     ✓ should fail if user did not approve the transfer (133ms)
     ✓ should fail if value is not within limits (148ms)
     ✓ should prevent emitting the event twice when ERC677 used by relayTokens and ERC677 is owned by token manager (455ms)
     ✓ should prevent emitting the event twice when ERC677 used by relayTokens and ERC677 is not owned by token manager (474ms)
  requestFailedMessageFix
     ✓ should allow to request a failed message fix (183ms)
     ✓ should be a failed transaction (136ms)
     ✓ should be the receiver of the failed transaction (114ms)
     ✓ message sender should be mediator from other side (134ms)
     ✓ should allow to request a fix multiple times (324ms)
  fixFailedMessage
     ✓ should fix burnt/locked tokens (311ms)
     ✓ should be called by bridge
     ✓ message sender should be mediator from other side (336ms)
  fixFailedMessage for alternative receiver
     ✓ should fix burnt/locked tokens (210ms)
  #claimTokens
     ✓ should be able to claim tokens (366ms)
  onTokenTransfer
     ✓ should emit UserRequestForAffirmation in AMB bridge (414ms)
     ✓ should be able to specify a different receiver (555ms)
  handleBridgedTokens
     ✓ should transfer locked tokens on message from amb (513ms)
     ✓ should transfer locked tokens on message from amb with decimal shift of 2 (713ms)
     ✓ should transfer locked tokens on message from amb with decimal shift of -1 (664ms)
     ✓ should emit AmountLimitExceeded and not transfer tokens when out of execution limits (489ms)

Contract: HomeAMBErc677ToErc677
  initialize
     ✓ should initialize (970ms)
     ✓ only owner can set bridge contract (213ms)
     ✓ only owner can set mediator contract (188ms)
     ✓ only owner can set request Gas Limit (225ms)
  set limits
     ✓ setMaxPerTx allows to set only to owner and cannot be more than daily limit (123ms)
     ✓ setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (123ms)
     ✓ setDailyLimit allow to set by owner and should be greater than maxPerTx or zero (214ms)
     ✓ setExecutionMaxPerTx allows to set only to owner and cannot be more than daily limit (119ms)
     ✓ setExecutionDailyLimit allow to set by owner and should be greater than maxPerTx or zero (202ms)
  getBridgeMode
```

```
    ✓ should return arbitrary message bridging mode and interface
  fixAssetsAboveLimits
    ✓ Should revert if value to unlock is bigger than max per transaction (55ms)
    ✓ Should allow to partially reduce outOfLimitAmount and not emit amb event (194ms)
    ✓ Should allow to partially reduce outOfLimitAmount and emit amb event (284ms)
    ✓ Should revert if try to unlock more than available (316ms)
    ✓ Should not be allow to be called by an already fixed message (227ms)
    ✓ Should fail if message didnt increase out of limit amount (72ms)
    ✓ Should fail if not called by proxyOwner (140ms)
  relayTokens
    ✓ should allow to bridge tokens using approve and transferFrom (305ms)
    ✓ should allow to specify a different receiver without specifying sender (291ms)
    ✓ should fail if user did not approve the transfer (139ms)
    ✓ should fail if value is not within limits (144ms)
    ✓ should prevent emitting the event twice when ERC677 used by relayTokens and ERC677 is owned by token manager (462ms)
    ✓ should prevent emitting the event twice when ERC677 used by relayTokens and ERC677 is not owned by token manager (470ms)
  requestFailedMessageFix
    ✓ should allow to request a failed message fix (184ms)
    ✓ should be a failed transaction (138ms)
    ✓ should be the receiver of the failed transaction (116ms)
    ✓ message sender should be mediator from other side (126ms)
    ✓ should allow to request a fix multiple times (333ms)
  fixFailedMessage
    ✓ should fix burnt/locked tokens (345ms)
    ✓ should be called by bridge
    ✓ message sender should be mediator from other side (385ms)
  fixFailedMessage for alternative receiver
    ✓ should fix burnt/locked tokens (227ms)
  #claimTokens
    ✓ should be able to claim tokens (372ms)
  onTokenTransfer
    ✓ should emit UserRequestForSignature in AMB bridge and burn transferred tokens (480ms)
    ✓ should be able to specify a different receiver (467ms)
  handleBridgedTokens
    ✓ should mint tokens on message from amb (613ms)
    ✓ should mint tokens on message from amb with decimal shift of 2 (784ms)
    ✓ should mint tokens on message from amb with decimal shift of -1 (888ms)
    ✓ should emit AmountLimitExceeded and not mint tokens when out of execution limits (752ms)

Contract: ForeignAMBNativeToErc20
  initialize
    ✓ should initialize parameters (1361ms)
    ✓ should initialize with zero fee manager address (122ms)
  set amb bridge params
    ✓ only owner can set bridge contract (189ms)
    ✓ only owner can set mediator contract (132ms)
    ✓ only owner can set request Gas Limit (153ms)
  set limits
    ✓ setMaxPerTx allows to set only to owner and cannot be more than daily limit (120ms)
    ✓ setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (121ms)
    ✓ setDailyLimit allow to set by owner and should be greater than maxPerTx or zero (219ms)
    ✓ setExecutionMaxPerTx allows to set only to owner and cannot be more than daily limit (121ms)
    ✓ setExecutionDailyLimit allow to set by owner and should be greater than maxPerTx or zero (187ms)
  getBridgeMode
    ✓ should return mediator mode and interface
  claimTokens
    ✓ should work with token that return bool on transfer (334ms)
    ✓ should works with token that not return on transfer (207ms)
    ✓ should also work for native coins (80ms)
  owner
    ✓ should transfer ownership (130ms)
  feeManager
    ✓ should allow to get and set the feeManager (239ms)
    ForeignFeeManagerAMBNativeToErc20
      constructor
        ✓ should validate parameters (555ms)
      rewardAccounts
        ✓ should allow to add accounts (210ms)
        ✓ should allow to remove an existing account (190ms)
      fee
        ✓ should allow to get and set the fee (112ms)
      owner
        ✓ should transfer ownership (127ms)
  onTokenTransfer
    ✓ should call AMB bridge and burnt tokens (380ms)
    ✓ should be able to specify a different receiver (568ms)
  relayTokens
    ✓ should allow to bridge tokens using approve and transferFrom (302ms)
    ✓ should allow to specify a different receiver without specifying sender (291ms)
    ✓ should fail if user did not approve the transfer (70ms)
    ✓ should fail if value is not within limits (89ms)
  handleBridgedTokens
    ✓ should mint tokens on message from amb (489ms)
    ✓ should mint tokens on message from amb with decimal shift of 2 (663ms)
    ✓ should mint tokens on message from amb with decimal shift of -1 (648ms)
    ✓ should revert when out of execution limits on message from amb (298ms)
  requestFailedMessageFix
    ✓ should allow to request a failed message fix (177ms)
    ✓ should be a failed transaction (136ms)
    ✓ should be the receiver of the failed transaction (107ms)
    ✓ message sender should be mediator from other side (118ms)
    ✓ should allow to request a fix multiple times (367ms)
  fixFailedMessage
    ✓ should fix burnt tokens (405ms)
    ✓ should be called by amb bridge (38ms)
  fixFailedMessage with alternative receiver
    ✓ should fix burnt tokens (401ms)
  handleBridgedTokens with fees
    ✓ should mint tokens and distribute fees on message from amb (1202ms)
    ✓ should allow a fee receiver to bridge back the tokens (2089ms)

Contract: HomeAMBNativeToErc20
  initialize
    ✓ should initialize parameters (1092ms)
    ✓ should initialize with zero fee manager address (103ms)
  set amb bridge params
    ✓ only owner can set bridge contract (181ms)
    ✓ only owner can set mediator contract (143ms)
    ✓ only owner can set request Gas Limit (154ms)
  set limits
    ✓ setMaxPerTx allows to set only to owner and cannot be more than daily limit (121ms)
    ✓ setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (132ms)
    ✓ setDailyLimit allow to set by owner and should be greater than maxPerTx or zero (214ms)
    ✓ setExecutionMaxPerTx allows to set only to owner and cannot be more than daily limit (118ms)
    ✓ setExecutionDailyLimit allow to set by owner and should be greater than maxPerTx or zero (183ms)
  getBridgeMode
    ✓ should return mediator mode and interface
  fallback
    ✓ should accept native tokens (155ms)
    ✓ native token amount should be inside limits  (513ms)
  relayTokens
    ✓ should accept native tokens and a different receiver (200ms)
    ✓ native token amount should be inside limits  (387ms)
  handleBridgedTokens
    ✓ should unlock native tokens on message from amb (499ms)
    ✓ should unlock native tokens on message from amb with decimal shift of 2 (1655ms)
    ✓ should unlock native tokens on message from amb with decimal shift of -1 (1902ms)
    ✓ should revert when out of execution limits on message from amb (1032ms)
  requestFailedMessageFix
    ✓ should allow to request a failed message fix (228ms)
    ✓ should be a failed transaction (199ms)
    ✓ should be the receiver of the failed transaction (217ms)
    ✓ message sender should be mediator from other side (160ms)
    ✓ should allow to request a fix multiple times (328ms)
  fixFailedMessage
    ✓ should fix locked tokens (433ms)
    ✓ should be called by amb bridge
  fixFailedMessage for alternative receiver
    ✓ should fix locked tokens (429ms)
    ✓ should be called by amb bridge
  claimTokens
    ✓ should work with token that return bool on transfer (331ms)
    ✓ should works with token that not return on transfer (222ms)
    ✓ should not work for native coins for this type of mediator (175ms)
  feeManager
    ✓ should allow to get and set the feeManager (293ms)
    HomeFeeManagerAMBNativeToErc20
      constructor
        ✓ should validate parameters (573ms)
      rewardAccounts
```

```
        ✓ should allow to add accounts (676ms)
        ✓ should allow to remove an existing account (227ms)
      fee
        ✓ should allow to get and set the fee (124ms)
      owner
        ✓ should transfer ownership (121ms)
      fallback
        ✓ should accept native tokens
  owner
    ✓ should transfer ownership (140ms)
  handleBridgedTokens with fees
    ✓ should unlock native tokens and distribute fees on message from amb (670ms)
  fixMediatorBalance
    ✓ should fix mediator imbalance (879ms)
    ✓ should fix mediator imbalance with respect to limits (1065ms)
    ✓ should fix mediator imbalance correctly with fees (918ms)

Contract: ForeignAMB
  getBridgeMode
    ✓ should return arbitrary message bridging mode (61ms)
  initialize
    ✓ sets variables (353ms)
    ✓ should fail with invalid arguments (501ms)
    ✓ can update variables (436ms)
  upgradeable
    ✓ can be upgraded (272ms)
    ✓ can be deployed via upgradeToAndCall (179ms)
    ✓ can transfer ownership (282ms)
  requireToPassMessage
    ✓ call requireToPassMessage(address, bytes, uint256) (65ms)
    ✓ should generate different message ids (166ms)
  executeSignatures
    ✓ should succeed on Subsidized mode (431ms)
    ✓ test with 3 signatures required (701ms)
    ✓ test with max allowed number of signatures required (5153ms)
    ✓ should not allow to double execute signatures (600ms)
    ✓ should allow non-authorities to execute signatures (240ms)
    ✓ status of RelayedMessage should be false on contract failed call (734ms)
    ✓ status of RelayedMessage should be false on contract out of gas call (613ms)
    ✓ should not allow to process messages with different version (285ms)
    ✓ should not allow to process messages with different destination chain id (300ms)
    ✓ should not allow to pass message back through the bridge (291ms)
  gasToken functionality
    setGasTokenParameters
      ✓ should initialize gas token (129ms)
      ✓ should fail if not an owner (52ms)
    setGasTokenTargetMintValue
      ✓ should initialize gas token (72ms)
      ✓ should reset to 0 (94ms)
      ✓ should fail if not an owner
    setGasTokenReceiver
      ✓ should initialize gas token (60ms)
      ✓ should reset to zero address (93ms)
      ✓ should fail if not an owner
    _collectGasTokens
      ✓ should mint tokens with zero approval (157ms)
      ✓ should mint tokens with partial approval (273ms)
      ✓ should transfer all approved tokens (227ms)
      ✓ should transfer target approved tokens (252ms)
      ✓ should do nothing on zero target (129ms)
      ✓ should do nothing on empty receiver address (127ms)
      ✓ should call onTokenTransfer (228ms)
    requireToPassMessage with gas token
      ✓ should mint gas tokens on Subsidized mode (108ms)
      ✓ should spend partial allowance and mint tokens on Subsidized mode (246ms)
      ✓ should spend full allowance on Subsidized mode (180ms)
      ✓ should spend partial allowance on Subsidized mode (181ms)
  setChainIds
    ✓ should allow to set chain id (129ms)
    ✓ should not allow to set invalid chain ids (137ms)
    ✓ should not allow to set chain id if not an owner

Contract: HomeAMB
  getBridgeMode
    ✓ should return arbitrary message bridging mode and interface (85ms)
  initialize
    ✓ sets variables (377ms)
    ✓ should fail with invalid arguments (415ms)
    ✓ can update variables (413ms)
  upgradeable
    ✓ can be upgraded (254ms)
    ✓ can be deployed via upgradeToAndCall (176ms)
    ✓ can transfer ownership (236ms)
  requireToPassMessage
    ✓ call requireToPassMessage(address, bytes, uint256) (57ms)
    ✓ call requireToPassMessage(address, bytes, uint256) should fail (308ms)
    ✓ should generate different message ids (160ms)
  executeAffirmation
    ✓ should succeed on Subsidized mode (313ms)
    ✓ test with 3 signatures required (817ms)
    ✓ should not allow to double execute (491ms)
    ✓ should not allow non-authorities to execute affirmation (327ms)
    ✓ status of AffirmationCompleted should be false on contract failed call (421ms)
    ✓ status of AffirmationCompleted should be false on contract out of gas call (425ms)
    ✓ should not allow to process messages with different version (191ms)
    ✓ should not allow to process messages with different destination chain id (228ms)
    ✓ should not allow to pass message back through the bridge (270ms)
  submitSignature
    ✓ allows a validator to submit a signature (262ms)
    ✓ test with 3 signatures required (751ms)
    ✓ should not allow to double submit (544ms)
    ✓ should not allow non-authorities to submit signatures (306ms)
  setChainIds
    ✓ should allow to set chain id (124ms)
    ✓ should not allow to set invalid chain ids (138ms)
    ✓ should not allow to set chain id if not an owner

Contract: ERC677MultiBridgeToken
  constructor
    ✓ should initialize contract (96ms)
  #addBridge
    ✓ should add one bridge (120ms)
    ✓ should add two bridges (229ms)
    ✓ should add max allowed number of bridges (3063ms)
    ✓ should not allow to add already existing bridge (93ms)
    ✓ should not allow to add 0xf as bridge address (74ms)
    ✓ should not allow to add 0x0 as bridge address (38ms)
    ✓ should not allow to add if not an owner (39ms)
  #removeBridge
    ✓ should remove bridges one by one (713ms)
    ✓ should not allow to remove not existing bridge (349ms)
    ✓ should not allow to remove if not an owner (70ms)
  #bridgeList
    ✓ should return empty bridge list
    ✓ should expand bridge list when adding bridges (208ms)
    ✓ should shrink bridge list when removing bridges (398ms)
  #setBridgeContract
    ✓ should always revert
  #bridgeContract
    ✓ should always revert

Contract: ForeignBridge_ERC20_to_ERC20
  #initialize
    ✓ should initialize (962ms)
  #executeSignatures
    ✓ should allow to executeSignatures (185ms)
    ✓ should allow second withdrawal with different transactionHash but same recipient and value (370ms)
    ✓ should not allow second withdraw (replay attack) with same transactionHash but different recipient (317ms)
    ✓ should not allow withdraw over home max tx limit (158ms)
    ✓ should not allow withdraw over daily home limit (415ms)
  #withdraw with 2 minimum signatures
    ✓ withdraw should fail if not enough signatures are provided (210ms)
    ✓ withdraw should fail if duplicate signature is provided (155ms)
    ✓ works with 5 validators and 3 required signatures (503ms)
    ✓ works with max allowed number of signatures required (4613ms)
  #upgradeable
    ✓ can be upgraded (616ms)
    ✓ can be deployed via upgradeToAndCall (227ms)
  #claimTokens
    ✓ can send erc20 (558ms)
```

```
  #ForeignBridgeErc677ToErc677_onTokenTransfer
     ✓ should emit correct events on initialize (141ms)
     ✓ can only be called from token contract (403ms)
     ✓ should not allow to transfer more than maxPerTx limit (480ms)
     ✓ should only let to transfer within daily limit (650ms)
     ✓ should not let to transfer less than minPerTx (521ms)
     ✓ should be able to specify a different receiver (679ms)
  #decimalShift
     ✓ Home to Foreign: withdraw with 1 signature with a decimalShift of 2 (376ms)
     ✓ Home to Foreign : withdraw works with 5 validators and 3 required signatures with a decimalShift of 2 (560ms)
     ✓ Foreign to Home: no impact in UserRequestForAffirmation event signal for bridges oracles with a decimalShift of 2. (432ms)
     ✓ Home to Foreign: withdraw with 1 signature with a decimalShift of -1 (502ms)
     ✓ Home to Foreign : withdraw works with 5 validators and 3 required signatures with a decimalShift of 2 (576ms)
     ✓ Foreign to Home: no impact in UserRequestForAffirmation event signal for bridges oracles with a decimalShift of 2. (452ms)
  #relayTokens
     ✓ should allow to bridge tokens using approve tranferFrom (331ms)
     ✓ should not be able to transfer more than limit (376ms)

Contract: HomeBridge_ERC20_to_ERC20
  #initialize
     ✓ sets variables (451ms)
     ✓ cant set maxPerTx > dailyLimit (151ms)
     ✓ can be deployed via upgradeToAndCall (218ms)
     ✓ cant initialize with invalid arguments (609ms)
     ✓ can initialize with zero gas price  (127ms)
  #fallback
     ✓ reverts
  #setting limits
     ✓ #setMaxPerTx allows to set only to owner and cannot be more than daily limit (119ms)
     ✓ #setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (111ms)
  #executeAffirmation
     ✓ should allow validator to withdraw (281ms)
     ✓ should allow validator to withdraw with zero value (239ms)
     ✓ test with 2 signatures required (736ms)
     ✓ should not allow to double submit (218ms)
     ✓ should not allow non-authorities to execute deposit (54ms)
     ✓ doesnt allow to deposit if requiredSignatures has changed (813ms)
     ✓ works with 5 validators and 3 required signatures (566ms)
     ✓ should not allow execute affirmation over foreign max tx limit (60ms)
     ✓ should fail if txHash already set as above of limits (223ms)
     ✓ should not allow execute affirmation over daily foreign limit (367ms)
  #isAlreadyProcessed
     ✓ returns  (128ms)
  #submitSignature
     ✓ allows a validator to submit a signature (208ms)
     ✓ when enough requiredSignatures are collected, CollectedSignatures event is emitted (442ms)
     ✓ works with 5 validators and 3 required signatures (758ms)
     ✓ attack when increasing requiredSignatures (588ms)
     ✓ attack when decreasing requiredSignatures (274ms)
  #requiredMessageLength
     ✓ should return the required message length
  #fixAssetsAboveLimits
     ✓ Should revert if value to unlock is bigger than max per transaction (139ms)
     ✓ Should allow to partially reduce outOfLimitAmount and not emit UserRequestForSignature (246ms)
     ✓ Should allow to partially reduce outOfLimitAmount and emit UserRequestForSignature (242ms)
     ✓ Should revert if try to unlock more than available (427ms)
     ✓ Should not be allow to be called by an already fixed txHash (527ms)
     ✓ Should fail if txHash didnt increase out of limit amount (135ms)
     ✓ Should fail if not called by proxyOwner (175ms)
     ✓ Should emit UserRequestForSignature with value reduced by fee (311ms)
  #claimTokens
     ✓ should be able to call claimTokens on tokenAddress (571ms)
  #rewardableInitialize
     ✓ sets variables (737ms)
     ✓ can update fee contract (218ms)
     ✓ can update fee (239ms)
     ✓ fee should be less than 100% (492ms)
     ✓ should be able to get fee manager mode (147ms)
     ✓ should be able to set blockReward contract (343ms)
  #onTokenTransfer
     ✓ should trigger UserRequestForSignature with transfer value (282ms)
     ✓ should be able to specify a different receiver (388ms)
     ✓ should trigger UserRequestForSignature with fee subtracted (511ms)
  #rewardable_submitSignatures
     ✓ should distribute fee to one validator (443ms)
     ✓ should distribute fee to 3 validators (567ms)
     ✓ should distribute fee to 5 validators (750ms)
     ✓ should distribute fee to max allowed number of validator (1784ms)
  #rewardable_executeAffirmation
     ✓ should distribute fee to one validator (405ms)
     ✓ should distribute fee to 3 validators (544ms)
     ✓ should distribute fee to 5 validators (686ms)
     ✓ should distribute fee to max allowed number of validators (2134ms)
  #decimals Shift
     ✓ Foreign to Home: works with 5 validators and 3 required signatures with decimal shift 2 (597ms)
     ✓ Foreign to Home: test decimal shift 2, no impact on UserRequestForSignature value (426ms)
     ✓ Foreign to Home: works with 5 validators and 3 required signatures with decimal shift -1 (641ms)
     ✓ Foreign to Home: test decimal shift -1, no impact on UserRequestForSignature value (409ms)

Contract: ForeignBridge_ERC20_to_Native
  #initialize
     ✓ should initialize (2574ms)
  #executeSignatures
     ✓ should allow to executeSignatures (187ms)
     ✓ should allow second withdrawal with different transactionHash but same recipient and value (348ms)
     ✓ should not allow second withdraw (replay attack) with same transactionHash but different recipient (313ms)
     ✓ should not allow withdraw over home max tx limit (159ms)
     ✓ should not allow withdraw over daily home limit (377ms)
  #executeSignatures with chai
     ✓ should executeSignatures with enabled chai token, enough dai (266ms)
     ✓ should executeSignatures with enabled chai token, not enough dai, low dai limit (2138ms)
     ✓ should executeSignatures with enabled chai token, not enough dai, high dai limit (2276ms)
  #withdraw with 2 minimum signatures
     ✓ withdraw should fail if not enough signatures are provided (232ms)
     ✓ withdraw should fail if duplicate signature is provided (183ms)
     ✓ works with 5 validators and 3 required signatures (526ms)
     ✓ works with max allowed number of signatures required (4767ms)
  #upgradeable
     ✓ can be upgraded (538ms)
     ✓ can be deployed via upgradeToAndCall (190ms)
  #claimTokens
     ✓ can send erc20 (553ms)
  #decimalShift
     ✓ Home to Foreign: withdraw with 1 signature with a decimalShift of 2 (542ms)
     ✓ Home to Foreign: withdraw with 2 minimum signatures with a decimalShift of 2 (492ms)
     ✓ Home to Foreign: withdraw with 1 signature with a decimalShift of -1 (476ms)
     ✓ Home to Foreign: withdraw with 2 minimum signatures with a decimalShift of -1 (486ms)
  #relayTokens
     ✓ should allow to bridge tokens using approve and relayTokens (335ms)
     ✓ should allow to bridge tokens using approve and relayTokens with different recipient (305ms)
     ✓ should not be able to transfer more than limit (419ms)
  #relayTokens with chai
     ✓ should allow to bridge tokens with chai token enabled (504ms)
     ✓ should allow to bridge tokens with chai token enabled, excess tokens (338ms)
  chai token
    initializeChaiToken
       ✓ should be initialized (46ms)
       ✓ should fail to initialize twice (84ms)
       ✓ should fail if not an owner (41ms)
    initializeChaiToken with interest receiver
       ✓ should be initialized (42ms)
       ✓ should fail to initialize twice (85ms)
       ✓ should fail if not an owner (41ms)
       ✓ should fail if zero address (62ms)
    chaiTokenEnabled
       ✓ should return false
       ✓ should return true (59ms)
    removeChaiToken
       ✓ should be removed (324ms)
       ✓ should be removed with tokens withdraw (2038ms)
       ✓ should fail if not an owner (217ms)
       ✓ should fail if chai token is not enabled (224ms)
    min dai limit
       ✓ should return minDaiTokenBalance
       ✓ should update minDaiTokenBalance (46ms)
       ✓ should fail to update if not an owner (38ms)
    interestReceiver
       ✓ should return interestReceiver
       ✓ should update interestReceiver (51ms)
       ✓ should fail to setInterestReceiver if not an owner (38ms)
```

```
    ✓ should fail to setInterestReceiver if receiver is bridge address (41ms)
  interestCollectionPeriod
    ✓ should return interestCollectionPeriod
    ✓ should update interestCollectionPeriod (51ms)
    ✓ should fail to setInterestCollectionPeriod if not an owner
  isDaiNeedsToBeInvested
    ✓ should return false on empty balance (68ms)
    ✓ should return false on insufficient balance (101ms)
    ✓ should return true on sufficient balance (117ms)
    ✓ should return false if chai token is not defined (65ms)
  convertDaiToChai
    ✓ should convert all dai except defined limit (302ms)
    ✓ should revert when there is nothing to convert (361ms)
    ✓ should not allow to convert if chai token is disabled (72ms)
  _convertChaiToDai
    ✓ should handle 0 amount (79ms)
    ✓ should handle overestimated amount (217ms)
    ✓ should handle amount == invested (229ms)
    ✓ should handle 0 < amount < invested (211ms)
  payInterest
    ✓ should pay full interest to regular account (347ms)
    ✓ should pay full interest to contract (429ms)
    ✓ should not allow not pay interest twice within short time period (1728ms)
    ✓ should allow to pay interest after some time (6890ms)
    ✓ should not allow to pay interest if chaiToken is disabled (313ms)
    ✓ should not pay interest on empty address (38ms)
  payInterest for upgradeabilityOwner
    ✓ should allow to pay interest without time restrictions (3905ms)
  claimTokens
    ✓ should not allow to claim Chai, if it is enabled (362ms)
    ✓ should allow to claim chai after it is disabled (265ms)
  InterestReceiver contract
    constructor
      ✓ should create contract with valid parameters (83ms)
      ✓ should not allow EOA in bridge contract parameter (38ms)
    setBridgeContract
      ✓ should update bridge contract address (47ms)
      ✓ should not allow EOA for bridge contract argument
      ✓ should not allow to change bridge if not an owner
    setReceiverInXDai
      ✓ should update bridge contract address (43ms)
      ✓ should not allow to change bridge if not an owner
    onTokenTransfer
      ✓ should relay incoming interest back to xDai chain (349ms)
      ✓ should emit RelayTokensFailed if transaction is out of bounds (295ms)
      ✓ should be able to repeat relayTokensAttempt (474ms)
    claimTokens
      ✓ should allow to claim tokens from recipient account (332ms)
      ✓ should not allow to claim tokens for chai token
      ✓ should not allow to claim tokens for dai token (39ms)
  Zero DSR
    ✓ should allow to executeSignatures when DSR is zero (4294ms)
    ✓ should allow to executeSignatures when DSR is zero with many conversions (6218ms)
    ✓ should allow to executeSignatures after pay interest (10269ms)

Contract: HomeBridge_ERC20_to_Native
  #initialize
    ✓ sets variables (423ms)
    ✓ can update block reward contract (504ms)
    ✓ cant set maxPerTx > dailyLimit (186ms)
    ✓ can be deployed via upgradeToAndCall (260ms)
    ✓ can be upgraded keeping the state (490ms)
    ✓ cant initialize with invalid arguments (553ms)
  #rewardableInitialize
    ✓ sets variables (466ms)
    ✓ cant initialize with invalid arguments (596ms)
    ✓ can update fee contract (201ms)
    ✓ can update fee (223ms)
    ✓ fee should be less than 100% (592ms)
  #fallback
    ✓ should accept native coins (170ms)
    ✓ should accumulate burnt coins (268ms)
    ✓ doesnt let you send more than daily limit (397ms)
    ✓ doesnt let you send more than max amount per tx (344ms)
    ✓ should not let to deposit less than minPerTx (244ms)
    ✓ should fail if not enough bridged tokens (307ms)
  #relayTokens
    ✓ should accept native coins and alternative receiver (196ms)
    ✓ should accumulate burnt coins (345ms)
    ✓ doesnt let you send more than daily limit (465ms)
    ✓ doesnt let you send more than max amount per tx (385ms)
    ✓ should not let to deposit less than minPerTx (304ms)
    ✓ should fail if not enough bridged tokens (369ms)
  #setting limits
    ✓ setMaxPerTx allows to set only to owner and cannot be more than daily limit (133ms)
    ✓ setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (131ms)
    ✓ setMaxPerTx allows to set limit to zero (93ms)
    ✓ setExecutionMaxPerTx allows to set only to owner and cannot be more than execution daily limit (191ms)
    ✓ executionDailyLimit allows to set only to owner (228ms)
  #executeAffirmation
    ✓ should allow validator to executeAffirmation (142ms)
    ✓ should allow validator to executeAffirmation with zero value (102ms)
    ✓ test with 2 signatures required (668ms)
    ✓ should not allow non-validator to execute affirmation (68ms)
    ✓ should fail if the block reward contract is not set (300ms)
    ✓ works with 5 validators and 3 required signatures (523ms)
    ✓ should not allow execute affirmation over foreign max tx limit (68ms)
    ✓ should fail if txHash already set as above of limits (218ms)
    ✓ should not allow execute affirmation over daily foreign limit (435ms)
  #submitSignature
    ✓ allows a validator to submit a signature (270ms)
    ✓ when enough requiredSignatures are collected, CollectedSignatures event is emitted (602ms)
    ✓ works with 5 validators and 3 required signatures (702ms)
    ✓ attack when increasing requiredSignatures (743ms)
    ✓ attack when decreasing requiredSignatures (294ms)
  #requiredMessageLength
    ✓ should return the required message length
  #fixAssetsAboveLimits
    ✓ Should revert if value to unlock is bigger than max per transaction (325ms)
    ✓ Should allow to partially reduce outOfLimitAmount and not emit UserRequestForSignature (265ms)
    ✓ Should allow to partially reduce outOfLimitAmount and emit UserRequestForSignature (302ms)
    ✓ Should revert if try to unlock more than available (538ms)
    ✓ Should not be allow to be called by an already fixed txHash (1081ms)
    ✓ Should fail if txHash didnt increase out of limit amount (166ms)
    ✓ Should fail if not called by proxyOwner (216ms)
    ✓ Should emit UserRequestForSignature with value reduced by fee (772ms)
  #feeManager
    ✓ should be able to set and get fee manager contract (121ms)
    ✓ should be able to set and get fees (254ms)
    ✓ should be able to get fee manager mode (114ms)
    ✓ should be able to get fee manager mode from POSDAO fee manager (116ms)
  #feeManager_ExecuteAffirmation
    ✓ should distribute fee to validator (716ms)
    ✓ should distribute fee to 3 validators (1005ms)
    ✓ should distribute fee to 5 validators (1214ms)
    ✓ should distribute fee to max allowed number of validators (4009ms)
  #feeManager_fallback
    ✓ should subtract fee from value (233ms)
  #feeManager_relayTokens
    ✓ should subtract fee from value (239ms)
  #feeManager_submitSignature
    ✓ should distribute fee to validator (863ms)
    ✓ should distribute fee to 3 validators (1232ms)
    ✓ should distribute fee to 5 validators (1483ms)
    ✓ should distribute fee to max allowed number of validators (3750ms)
  #FeeManager_random
    ✓ should return value between 0 and 3 (738ms)
  #feeManager_ExecuteAffirmation_POSDAO
    ✓ should distribute fee to validator (1037ms)
    ✓ should distribute fee to 3 validators (1193ms)
    ✓ should distribute fee to 5 validators (1557ms)
    ✓ should distribute fee to max allowed number of validators (2839ms)
  #feeManager_fallback_POSDAO
    ✓ should subtract fee from value (288ms)
  #feeManager_relayTokens_POSDAO
    ✓ should subtract fee from value (271ms)
  #feeManager_submitSignature_POSDAO
    ✓ should distribute fee to validator (932ms)
    ✓ should distribute fee to 3 validators (1096ms)
    ✓ should distribute fee to 5 validators (1427ms)
```

```
      ✓ should distribute fee to max allowed number of validators (3006ms)
    #decimals Shift
      ✓ Foreign to Home: test with 2 signatures required and decimal shift 2 (733ms)
      ✓ Home to Foreign: test decimal shift 2, no impact on UserRequestForSignature value (452ms)
      ✓ Foreign to Home: test with 2 signatures required and decimal shift -1 (706ms)
      ✓ Home to Foreign: test decimal shift -1, no impact on UserRequestForSignature value (499ms)

  Contract: Address Library
    safeSendValue
      ✓ should send value even if receiver does not allow (217ms)

  Contract: ArbitraryMessage.sol
    unpackData
      ✓ unpack dataType 0x00 (63ms)
      ✓ unpack dataType 0x00 with different chain ids (613ms)
      ✓ unpack dataType 0x01 (66ms)
      ✓ unpack dataType 0x02 (63ms)
    unpackData with signatures parameters
      ✓ unpack dataType 0x00 (59ms)
      ✓ unpack dataType 0x01 (61ms)
      ✓ unpack dataType 0x02 (60ms)

  Contract: TokenReader Library
    test different possible tokens
      ✓ should handle Token1 (87ms)
      ✓ should handle Token2 (94ms)
      ✓ should handle Token3 (83ms)
      ✓ should handle Token4 (94ms)
      ✓ should handle Token5 (110ms)
      ✓ should handle Token6 (93ms)
      ✓ should handle Token7 (82ms)

  Contract: ForeignMultiAMBErc20ToErc677
    initialize
      ✓ should initialize parameters (974ms)
    getBridgeMode
      ✓ should return mediator mode and interface
    claimTokens
      ✓ should only work with unknown token (728ms)
      ✓ should also work for native coins (143ms)
    afterInitialization
      update mediator parameters
        limits
          ✓ should allow to update default daily limits (433ms)
          ✓ should allow to update default max per tx limits (472ms)
          ✓ should allow to update default min per tx limit (197ms)
          ✓ should only allow to update parameters for known tokens (789ms)
    onTokenTransfer
      ✓ should call AMB bridge and lock tokens (664ms)
      ✓ should respect global shutdown (509ms)
      ✓ should be able to specify a different receiver (680ms)
    relayTokens
      ✓ should allow to bridge tokens using approve and relayTokens (358ms)
      ✓ should allow to specify a different receiver without specifying sender (362ms)
      ✓ should allow to specify no receiver and no sender (346ms)
      ✓ should fail if user did not approve the transfer (53ms)
      ✓ should fail if value is not within limits (341ms)
    token registration
      ✓ should make subsequent calls deployAndHandleBridgedTokens handleBridgedTokens to when using simpleTransfer (508ms)
      ✓ should make subsequent calls deployAndHandleBridgedTokens handleBridgedTokens to when using emptyAlternativeReceiver (756ms)
      ✓ should make subsequent calls deployAndHandleBridgedTokens handleBridgedTokens to when using sameAlternativeReceiver (605ms)
      ✓ should make subsequent calls deployAndHandleBridgedTokens handleBridgedTokens to when using differentAlternativeReceiver (569ms)
      ✓ should make subsequent calls deployAndHandleBridgedTokens handleBridgedTokens to when using simpleRelayTokens1 (655ms)
      ✓ should make subsequent calls deployAndHandleBridgedTokens handleBridgedTokens to when using simpleRelayTokens2 (652ms)
      ✓ should make subsequent calls deployAndHandleBridgedTokens handleBridgedTokens to when using relayTokensWithAlternativeReceiver1 (757ms)
      ✓ should initialize limits according to decimals = 3 (435ms)
      ✓ should initialize limits according to decimals = 18 (396ms)
      ✓ should initialize limits according to decimals = 20 (403ms)
      ✓ should initialize limits according to decimals = 0 (394ms)
    handleBridgedTokens
      ✓ should unlock tokens on message from amb (829ms)
      ✓ should not allow to use unregistered tokens (185ms)
      ✓ should not allow to operate when global shutdown is enabled (657ms)
    requestFailedMessageFix
      ✓ should allow to request a failed message fix (211ms)
      ✓ should be a failed transaction (516ms)
      ✓ should be the receiver of the failed transaction (127ms)
      ✓ message sender should be mediator from other side (129ms)
      ✓ should allow to request a fix multiple times (391ms)
    fixFailedMessage
      ✓ should fix tokens locked via simpleTransfer (1805ms)
      ✓ should fix tokens locked via emptyAlternativeReceiver (1046ms)
      ✓ should fix tokens locked via sameAlternativeReceiver (1056ms)
      ✓ should fix tokens locked via differentAlternativeReceiver (1161ms)
      ✓ should fix tokens locked via simpleRelayTokens1 (1227ms)
      ✓ should fix tokens locked via simpleRelayTokens2 (3522ms)
      ✓ should fix tokens locked via relayTokensWithAlternativeReceiver1 (1288ms)
    fixMediatorBalance
      ✓ should allow to fix extra mediator balance (1110ms)
      ✓ should allow to fix extra mediator balance with respect to limits (1326ms)

  Contract: HomeMultiAMBErc20ToErc677
    initialize
      ✓ should initialize parameters (1293ms)
    getBridgeMode
      ✓ should return mediator mode and interface
    claimTokens
      ✓ should only work with unknown token (1112ms)
      ✓ should also work for native coins (137ms)
    afterInitialization
      deploy and register new token
        ✓ can be called only by mediator from the other side (389ms)
        ✓ should register new token in deployAndHandleBridgedTokens (928ms)
        ✓ should register new token with empty name (709ms)
        ✓ should register new token with empty symbol (720ms)
        ✓ should initialize limits according to decimals = 3 (948ms)
        ✓ should initialize limits according to decimals = 18 (811ms)
        ✓ should initialize limits according to decimals = 20 (832ms)
        ✓ should initialize limits according to decimals = 0 (936ms)
        ✓ should initialize fees (1213ms)
      update mediator parameters
        limits
          ✓ should allow to update default daily limits (449ms)
          ✓ should allow to update default max per tx limits (651ms)
          ✓ should allow to update default min per tx limit (227ms)
          ✓ should only allow to update parameters for known tokens (1270ms)
        tokenImage
          ✓ should allow to change token image (242ms)
    onTokenTransfer
      ✓ should call AMB bridge and burn tokens (471ms)
      ✓ should respect global shutdown (421ms)
      ✓ should be able to specify a different receiver (517ms)
    relayTokens
      ✓ should allow to bridge tokens using approve and relayTokens (386ms)
      ✓ should allow to specify a different receiver without specifying sender (593ms)
      ✓ should allow to specify no receiver and no sender (588ms)
      ✓ should fail if user did not approve the transfer (248ms)
      ✓ should fail if value is not within limits (196ms)
    handleBridgedTokens
      ✓ should mint tokens on message from amb (928ms)
      ✓ should not allow to operate when global shutdown is enabled (438ms)
      ✓ should not allow to use unregistered tokens (193ms)
    requestFailedMessageFix for token registration
      ✓ should allow to request fix of first bridge operation for some token (635ms)
    requestFailedMessageFix
      ✓ should allow to request a failed message fix (534ms)
      ✓ should be a failed transaction (251ms)
      ✓ should be the receiver of the failed transaction (170ms)
      ✓ message sender should be mediator from other side (321ms)
      ✓ should allow to request a fix multiple times (757ms)
    fixFailedMessage
      ✓ should fix tokens burnt via simpleTransfer (1840ms)
      ✓ should fix tokens burnt via emptyAlternativeReceiver (1429ms)
      ✓ should fix tokens burnt via sameAlternativeReceiver (1416ms)
      ✓ should fix tokens burnt via differentAlternativeReceiver (1012ms)
      ✓ should fix tokens burnt via simpleRelayTokens1 (1009ms)
      ✓ should fix tokens burnt via simpleRelayTokens2 (1287ms)
      ✓ should fix tokens burnt via relayTokensWithAlternativeReceiver1 (1536ms)
  fees management
    ✓ change reward addresses (915ms)
```

```
    update fee parameters
      ✓ should update default fee value (213ms)
      ✓ should update default opposite direction fee value (212ms)
      ✓ should update fee value for registered token (2127ms)
      ✓ should update opposite direction fee value for registered token (2021ms)
    distribute fee for foreign => home direction
      ✓ should collect and distribute 0% fee (1887ms)
      ✓ should collect and distribute 1% fee (1661ms)
      ✓ should collect and distribute 1% fee between two reward addresses (1784ms)
    distribute fee for home => foreign direction
      ✓ should collect and distribute 0% fee (623ms)
      ✓ should collect and distribute 2% fee (660ms)
      ✓ should collect and distribute 2% fee between two reward addresses (793ms)

Contract: ForeignBridge
  #initialize
    ✓ should initialize (886ms)
  #executeSignatures
    ✓ should allow to deposit (205ms)
    ✓ should reject if address is not foreign address (140ms)
    ✓ should allow second deposit with different transactionHash but same recipient and value (320ms)
    ✓ should not allow second deposit (replay attack) with same transactionHash but different recipient (264ms)
    ✓ should not allow withdraw over home max tx limit (126ms)
    ✓ should not allow withdraw over daily home limit (332ms)
  #executeSignatures with 2 minimum signatures
    ✓ deposit should fail if not enough signatures are provided (202ms)
    ✓ deposit should fail if duplicate signature is provided (183ms)
    ✓ works with 5 validators and 3 required signatures (873ms)
    ✓ works with max allowed number of signatures required (5298ms)
    ✓ Should fail if length of signatures is wrong (1093ms)
  #onTokenTransfer
    ✓ can only be called from token contract (383ms)
    ✓ should not allow to burn more than the limit (464ms)
    ✓ should only let to send within maxPerTx limit (775ms)
    ✓ should not let withdraw less than minPerTx (668ms)
    ✓ should be able to specify a different receiver (1095ms)
  #setting limits
    ✓ #setMaxPerTx allows to set only to owner and cannot be more than daily limit (111ms)
    ✓ #setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (179ms)
  #upgradeable
    ✓ can be upgraded (635ms)
    ✓ can be deployed via upgradeToAndCall (323ms)
    ✓ can transfer ownership (361ms)
  #claimTokens
    ✓ can send erc20 (741ms)
    ✓ also calls claimTokens on tokenAddress (533ms)
    ✓ works with token that not return on transfer (531ms)
  #rewardableInitialize
    ✓ sets variables (823ms)
    ✓ can update fee contract (186ms)
    ✓ can update fee (207ms)
    ✓ fee should be less than 100% (260ms)
    ✓ should be able to get fee manager mode (142ms)
  #RewardableBridge_executeSignatures
    ✓ should distribute fee to validator (614ms)
    ✓ should distribute fee to 3 validators (874ms)
    ✓ should distribute fee to 5 validators (783ms)
    ✓ should distribute fee to max allowed number of validators (3657ms)
  #decimalShift
    ✓ Home to Foreign: withdraw works with decimalShift of 2 (553ms)
    ✓ Foreign to Home: no impact in transferAndCall event signal for bridges oracles with a decimalShift of 2 (362ms)
    ✓ Home to Foreign: withdraw works with decimalShift of -1 (587ms)
    ✓ Foreign to Home: no impact in transferAndCall event signal for bridges oracles with a decimalShift of -1 (444ms)

Contract: HomeBridge
  #initialize
    ✓ sets variables (385ms)
    ✓ cant set maxPerTx > dailyLimit (337ms)
    ✓ can set gas Price  (289ms)
    ✓ can set Required Block Confirmations (334ms)
    ✓ can be deployed via upgradeToAndCall (377ms)
    ✓ cant initialize with invalid arguments (664ms)
    ✓ can transfer ownership (210ms)
    ✓ can transfer proxyOwnership (201ms)
  #fallback
    ✓ should accept native coins (302ms)
    ✓ doesnt let you send more than max amount per tx (448ms)
    ✓ should not let to deposit less than minPerTx (329ms)
  #relayTokens
    ✓ should accept native coins and alternative receiver (357ms)
    ✓ doesnt let you send more than max amount per tx (927ms)
    ✓ should not let to deposit less than minPerTx (258ms)
  #setting limits
    ✓ #setMaxPerTx allows to set only to owner and cannot be more than daily limit (159ms)
    ✓ #setMinPerTx allows to set only to owner and cannot be more than daily limit and should be less than maxPerTx (137ms)
    ✓ #setDailyLimit allow to set by owner and should be greater than maxPerTx or zero (257ms)
  #executeAffirmation
    ✓ should allow validator to executeAffirmation (143ms)
    ✓ should allow validator to executeAffirmation with zero value (109ms)
    ✓ test with 2 signatures required (695ms)
    ✓ should not allow to double submit (198ms)
    ✓ should not allow non-authorities to execute withdraw (65ms)
    ✓ doesnt allow to withdraw if requiredSignatures has changed (837ms)
    ✓ force withdraw if the recepient has fallback to revert (274ms)
    ✓ works with 5 validators and 3 required signatures (634ms)
    ✓ should not allow execute affirmation over foreign max tx limit (82ms)
    ✓ should not allow execute affirmation over daily foreign limit (389ms)
  #isAlreadyProcessed
    ✓ returns  (102ms)
  #submitSignature
    ✓ allows a validator to submit a signature (187ms)
    ✓ when enough requiredSignatures are collected, CollectedSignatures event is emitted (521ms)
    ✓ works with 5 validators and 3 required signatures (674ms)
    ✓ attack when increasing requiredSignatures (683ms)
    ✓ attack when decreasing requiredSignatures (426ms)
  #requiredMessageLength
    ✓ should return the required message length
  #claimTokens
    ✓ should work with token that return bool on transfer (418ms)
    ✓ should works with token that not return on transfer (353ms)
    ✓ should work for native coins (265ms)
  #rewardableInitialize
    ✓ sets variables (712ms)
    ✓ can update fee contract (213ms)
    ✓ can update fee (184ms)
    ✓ fee should be less than 100% (292ms)
    ✓ should be able to get fee manager mode (156ms)
    ✓ should be able to get fee manager mode for both directions (184ms)
  #feeManager_OneDirection_fallback
    ✓ should not subtract fee from value (370ms)
  #feeManager_OneDirection_relayRequest
    ✓ should not subtract fee from value (453ms)
  #feeManager_OneDirection_submitSignature
    ✓ should not distribute fee to validator (488ms)
  #feeManager_OneDirection_ExecuteAffirmation
    ✓ should distribute fee to validator (674ms)
    ✓ should distribute fee to 3 validators (772ms)
    ✓ should distribute fee to 5 validators (1326ms)
    ✓ should distribute fee to max allowed number of validators (3744ms)
  #feeManager_BothDirections_fallback
    ✓ should subtract fee from value (446ms)
  #feeManager_BothDirections_relayRequest
    ✓ should subtract fee from value (450ms)
  #feeManager_BothDirections_submitSignature
    ✓ should distribute fee to validator (780ms)
    ✓ should distribute fee to 3 validators (1848ms)
    ✓ should distribute fee to 5 validators (1734ms)
    ✓ should distribute fee to max allowed number of validators (4327ms)
  #feeManager_BothDirections_ExecuteAffirmation
    ✓ should distribute fee to validator (899ms)
    ✓ should distribute fee to 3 validators (1229ms)
    ✓ should distribute fee to 5 validators (1565ms)
    ✓ should distribute fee to max allowed number of validators (4774ms)
  #decimalShift
    ✓ Foreign to Home: works with 5 validators and 3 required signatures with decimal shift 2 (791ms)
    ✓ Foreign to Home: test decimal shift 2, no impact on UserRequestForSignature value (466ms)
    ✓ Foreign to Home: works with 5 validators and 3 required signatures with decimal shift -1 (1055ms)
    ✓ Foreign to Home: test decimal shift -1, no impact on UserRequestForSignature value (638ms)

Contract: ERC677BridgeToken
```

```
    ✓ default values (168ms)
  #bridgeContract
     ✓ can set bridge contract (166ms)
     ✓ only owner can set bridge contract (207ms)
     ✓ fail to set invalid bridge contract address (161ms)
  #mint
     ✓ can mint by owner (246ms)
     ✓ no one can call finishMinting (46ms)
     ✓ cannot mint by non-owner (97ms)
  #transfer
     ✓ sends tokens to recipient (224ms)
     ✓ sends tokens to bridge contract (349ms)
     ✓ sends tokens to contract that does not contains onTokenTransfer method (126ms)
     ✓ fail to send tokens to bridge contract out of limits (503ms)
  #transferFrom
     ✓ should call onTokenTransfer (319ms)
  #increaseAllowance
     ✓ can increase allowance (133ms)
  #decreaseAllowance
     ✓ can decrease allowance (217ms)
  #burn
     ✓ can burn (182ms)
  #transferAndCall
     ✓ calls contractFallback (461ms)
     ✓ sends tokens to bridge contract (418ms)
     ✓ fail to sends tokens to contract that does not contains onTokenTransfer method (197ms)
     ✓ fail to send tokens to bridge contract out of limits (408ms)
  #claimtokens
     ✓ can take send ERC20 tokens (348ms)
     ✓ works with token that not return on transfer (230ms)
  #transfer
     ✓ if transfer called on contract, onTokenTransfer is also invoked (387ms)
     ✓ if transfer called on contract, still works even if onTokenTransfer doesnot exist (227ms)
  #renounceOwnership
     ✓ should not be able to renounce ownership (77ms)

Contract: ERC677BridgeTokenRewardable
    ✓ default values (121ms)
  #bridgeContract
     ✓ can set bridge contract (129ms)
     ✓ only owner can set bridge contract (327ms)
     ✓ fail to set invalid bridge contract address (144ms)
  #blockRewardContract
     ✓ can set BlockReward contract (109ms)
     ✓ only owner can set BlockReward contract (159ms)
     ✓ fail to set invalid BlockReward contract address (119ms)
  #stakingContract
     ✓ can set Staking contract (107ms)
     ✓ only owner can set Staking contract (180ms)
     ✓ fail to set invalid Staking contract address (140ms)
     ✓ fail to set Staking contract address with non-zero balance (202ms)
  #mintReward
     ✓ can only be called by BlockReward contract (109ms)
     ✓ should increase totalSupply and balance (131ms)
  #stake
     ✓ can only be called by Staking contract (203ms)
     ✓ should revert if user doesn't have enough balance (178ms)
     ✓ should decrease user's balance and increase Staking's balance (226ms)
  #mint
     ✓ can mint by owner (84ms)
     ✓ no one can call finishMinting (41ms)
     ✓ cannot mint by non-owner (83ms)
  #transfer
     ✓ sends tokens to recipient (161ms)
     ✓ sends tokens to bridge contract (303ms)
     ✓ sends tokens to contract that does not contains onTokenTransfer method (154ms)
     ✓ fail to send tokens to bridge contract out of limits (345ms)
     ✓ fail to send tokens to BlockReward contract directly (179ms)
     ✓ fail to send tokens to Staking contract directly (205ms)
  #transferFrom
     ✓ should call onTokenTransfer (315ms)
     ✓ fail to send tokens to BlockReward contract directly (501ms)
     ✓ fail to send tokens to Staking contract directly (220ms)
  #increaseAllowance
     ✓ can increase allowance (135ms)
  #decreaseAllowance
     ✓ can decrease allowance (137ms)
  #burn
     ✓ can burn (144ms)
  #transferAndCall
     ✓ calls contractFallback (376ms)
     ✓ sends tokens to bridge contract (282ms)
     ✓ fail to sends tokens to contract that does not contains onTokenTransfer method (222ms)
     ✓ fail to send tokens to bridge contract out of limits (380ms)
  #claimtokens
     ✓ can take send ERC20 tokens (323ms)
     ✓ works with token that not return on transfer (244ms)
  #transfer
     ✓ if transfer called on contract, onTokenTransfer is also invoked (298ms)
     ✓ if transfer called on contract, still works even if onTokenTransfer doesnot exist (243ms)
  #renounceOwnership
     ✓ should not be able to renounce ownership (81ms)
  permit
     ✓ should permit (359ms)
     ✓ should fail when invalid expiry (221ms)
     ✓ should consider expiry (377ms)
     ✓ should disallow unlimited allowance (447ms)
     ✓ should fail when invalid signature or parameters (833ms)

Contract: TokenProxy
    ✓ default values (115ms)
  #bridgeContract
     ✓ can set bridge contract (144ms)
     ✓ only owner can set bridge contract (184ms)
     ✓ fail to set invalid bridge contract address (145ms)
  #mint
     ✓ can mint by owner (88ms)
     ✓ no one can call finishMinting (39ms)
     ✓ cannot mint by non-owner (130ms)
  #transfer
     ✓ sends tokens to recipient (155ms)
     ✓ sends tokens to bridge contract (298ms)
     ✓ sends tokens to contract that does not contains onTokenTransfer method (161ms)
     ✓ fail to send tokens to bridge contract out of limits (319ms)
  #transferFrom
     ✓ should call onTokenTransfer (324ms)
  #increaseAllowance
     ✓ can increase allowance (132ms)
  #decreaseAllowance
     ✓ can decrease allowance (145ms)
  #burn
     ✓ can burn (171ms)
  #transferAndCall
     ✓ calls contractFallback (377ms)
     ✓ sends tokens to bridge contract (285ms)
     ✓ fail to sends tokens to contract that does not contains onTokenTransfer method (148ms)
     ✓ fail to send tokens to bridge contract out of limits (320ms)
  #claimtokens
     ✓ can take send ERC20 tokens (460ms)
     ✓ works with token that not return on transfer (236ms)
  #transfer
     ✓ if transfer called on contract, onTokenTransfer is also invoked (287ms)
     ✓ if transfer called on contract, still works even if onTokenTransfer doesnot exist (319ms)
  #renounceOwnership
     ✓ should not be able to renounce ownership (82ms)
  permit
     ✓ should permit (338ms)
     ✓ should fail when invalid expiry (214ms)
     ✓ should consider expiry (355ms)
     ✓ should disallow unlimited allowance (466ms)
     ✓ should fail when invalid signature or parameters (422ms)
  constants
     ✓ should return version
     ✓ should return PERMIT_TYPEHASH (45ms)

Contract: RewardableValidators
  #initialize
     ✓ sets values (902ms)
     ✓ should fail if exceed amount of validators (3793ms)
     ✓ should be able to operate with max allowed number of validators (1996ms)
```

```
  #addValidator
     ✓ adds validator (166ms)
     ✓ cannot add already existing validator (78ms)
     ✓ cannot add 0xf as validator address (51ms)
     ✓ cannot add 0x0 as validator address
     ✓ cannot add 0x0 as reward address
  #removeValidator
     ✓ removes validator (153ms)
     ✓ cannot remove if it will break requiredSignatures (195ms)
     ✓ cannot remove non-existent validator (123ms)
  #setRequiredSignatures
     ✓ sets req signatures (117ms)
     ✓ cannot set more than  validators count (67ms)
  #upgradable
     ✓ can be upgraded via upgradeToAndCall (811ms)
  #single list remove
     ✓ should remove 0xDf08F82De32B8d460adbE8D72043E3a7e25A3B39 - without Proxy (173ms)
     ✓ Removed validator should return zero address on nextValidator (721ms)
     ✓ should remove 0xDf08F82De32B8d460adbE8D72043E3a7e25A3B39 - with Proxy (429ms)
     ✓ should remove 0x6704Fbfcd5Ef766B287262fA2281C105d57246a6 - with Proxy (445ms)
     ✓ should remove 0x9E1Ef1eC212F5DFfB41d35d9E5c14054F26c6560 - with Proxy (463ms)
     ✓ should remove 0xce42bdB34189a93c55De250E011c68FaeE374Dd3 - with Proxy (483ms)
     ✓ should remove 0x97A3FC5Ee46852C1Cf92A97B7BaD42F2622267cC - with Proxy (474ms)
  #reward address
     ✓ reward address is properly assigned (257ms)
  #Validators list
     ✓ should return validators list (322ms)

Contract: ForeignStakeTokenMediator
  getBridgeMode
     ✓ should return stake bridging mode and interface
  bridge tokens to mainnet
     ✓ should use tokens from bridge balance (230ms)
     ✓ should use all tokens from bridge balance (239ms)
     ✓ should mint lacking tokens (257ms)
     ✓ should mint lacking tokens, zero initial balance (203ms)
  return fixed tokens
     ✓ should free fixed tokens, without minting new tokens (280ms)
     ✓ should free fixed tokens, with minting new tokens (438ms)
  bridge tokens from mainnet
     ✓ should accept tokens within limits (197ms)
     ✓ should not accept zero tokens (247ms)
     ✓ should not accept tokens if receiver is a mediator on the other side (254ms)

Contract: HomeStakeTokenMediator
  rewardableInitialize
     ✓ should initialize (719ms)
     ✓ should not accept invalid blockReward (86ms)
  getBridgeMode
     ✓ should return stake bridging mode and interface (83ms)
  after initialization
    setBlockRewardContract
       ✓ should set block reward contract (282ms)
       ✓ should fail if not a block reward contract (93ms)
       ✓ should fail if not an owner (56ms)
    setFee
       ✓ should set fee
       ✓ should fail if fee is too high (189ms)
       ✓ should fail if not an owner (53ms)
    getFee
       ✓ should get actual fee (69ms)
    isFeeCollectingActivated
       ✓ should return false when no block reward and no fee
       ✓ should return false when block reward is configured but no fee (70ms)
       ✓ should return false when no block reward but fee is set (51ms)
       ✓ should return true when both block reward and fee are configured (98ms)
    calculateFee
       ✓ should calculate fee for given value (275ms)
    bridge tokens from xDai chain
       ✓ should accept tokens, no fee (381ms)
       ✓ should accept tokens, configured fee (362ms)
       ✓ should accept tokens, block reward contract is not configured (282ms)
       ✓ should not accept zero tokens (238ms)
       ✓ should not accept tokens if receiver is a mediator on the other side (268ms)
    bridge tokens to xDai chain
       ✓ should mint new tokens  (378ms)
    return fixed tokens
       ✓ should mint fixed tokens, (449ms)
    transferTokenOwnership
       ✓ should transfer token ownership to different contract (119ms)
       ✓ should fail if not an owner (84ms)
       ✓ should fail if not a current token owner (52ms)
    mintHandler
       ✓ should allow to set different mint handler (66ms)
       ✓ should fail if not an owner
       ✓ should fail if not a contract
       ✓ should process bridge tokens through mint handler (235ms)

Contract: BridgeValidators
  #initialize
     ✓ sets values (684ms)
     ✓ should fail if exceed amount of validators (1606ms)
     ✓ should be able to operate with max allowed number of validators (1210ms)
  #addValidator
     ✓ adds validator (137ms)
     ✓ cannot add already existing validator (76ms)
     ✓ cannot add 0xf as validator address
     ✓ cannot add 0x0 as validator address (45ms)
  #removeValidator
     ✓ removes validator (141ms)
     ✓ cannot remove if it will break requiredSignatures (184ms)
     ✓ cannot remove non-existent validator (124ms)
  #setRequiredSignatures
     ✓ sets req signatures (95ms)
     ✓ cannot set more than  validators count (67ms)
  #upgradable
     ✓ can be upgraded via upgradeToAndCall (280ms)
  #single list remove
     ✓ should remove 0xDf08F82De32B8d460adbE8D72043E3a7e25A3B39 - without Proxy (261ms)
     ✓ Removed validator should return zero address on nextValidator (377ms)
     ✓ should remove 0xDf08F82De32B8d460adbE8D72043E3a7e25A3B39 - with Proxy (501ms)
     ✓ should remove 0x6704Fbfcd5Ef766B287262fA2281C105d57246a6 - with Proxy (494ms)
     ✓ should remove 0x9E1Ef1eC212F5DFfB41d35d9E5c14054F26c6560 - with Proxy (518ms)
     ✓ should remove 0xce42bdB34189a93c55De250E011c68FaeE374Dd3 - with Proxy (522ms)
     ✓ should remove 0x97A3FC5Ee46852C1Cf92A97B7BaD42F2622267cC - with Proxy (520ms)
  #Validators list
     ✓ should return validators list (167ms)
  #isValidatorDuty
     ✓ should return if provided valdidator is on duty (250ms)


  949 passing (15m)
```

Code Coverage

Although the test suite appears very thorough, we were unable to run the coverage scripts successfully. The following errors were reported.

```
TypeError: ethereumjs_account_1.default is not a constructor at new VM (~/audits/tokenbridge-contracts/node_modules/ganache-core/node_modules/ethereumjs-vm/lib/index.ts:121:61) ...
```

Appendix

File Signatures

The following are the SHA-256 hashes of the reviewed files. A file with a different SHA-256 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review.

Contracts

```
a433772b617607117b55e3ee1899857ec7433fbe6dfcc27930c1c7d8de7ac9af ./contracts/ERC677BridgeTokenRewardable.sol
c500a6e4069d29179b2ed57777d7c06fa4e29d18ea3419517afcde0a59fc3046 ./contracts/Migrations.sol
6bd3c17e64315546b42165414d78225ddc8cdba9906bd48937e30c3fd9f26a51 ./contracts/PermittableToken.sol
bd42a25010d46d4db719bde2a9e0a58d093804b744b86bfc251a1f84064a647e ./contracts/ERC677MultiBridgeToken.sol
526ada570713c5c5506ddfcb380e61b98d381272a6d0fa8e99b1249d65bdde5f ./contracts/ERC677BridgeToken.sol
7f35b050ea4690c1977fa6896819884f38e887c4a40942c25937b60c312d423e ./contracts/upgradeability/OwnedUpgradeabilityProxy.sol
4ab093b5264a7a46ab92514abdc915ef2ccb6755d4ffe619e3c24ee0bafc3cae ./contracts/upgradeability/EternalStorageProxy.sol
dfc565cfc75b6b361144320dc804c9558ae2d1350268648db4ec0d78b0cdc0ae ./contracts/upgradeability/Proxy.sol
09ab1dbf1ce73828e6466510e5b59a369dc985c2e4bd253cf1e405daabc5ee94 ./contracts/upgradeability/UpgradeabilityOwnerStorage.sol
47f69135fe82286475ba5c53829e3935e0b1ebfe6f8cbbe5114edb2523b414aa ./contracts/upgradeability/EternalStorage.sol
246474bbdeb3532d2478b4fa78d3b53a4725326923c0216948f99f6c58cd0036 ./contracts/upgradeability/UpgradeabilityProxy.sol
eeb90c347feb4ec45a9366af1c97ddbff79d08b74957270fdc040cbc12c22584 ./contracts/upgradeability/UpgradeabilityStorage.sol
bac0d0186615decb19a8907c037ee4f52f2bee4b795d074f46d444e27c51bc32 ./contracts/interfaces/IPot.sol
75fcba7b658f24009e12ece0c39a67b5e76d3bba204821a4b1e990184f27dbcd ./contracts/interfaces/IAMB.sol
5f715cc665ecc40619a69d12dfdce6ba78cbd8a429ff216b87297f19a020ff12 ./contracts/interfaces/ERC677Receiver.sol
5c995c4fb0e86d94142ecaadcf683e74e6d011d95baa2944b98dba536f233d61 ./contracts/interfaces/IRewardableValidators.sol
0fcbb2bfb3cd0e25b2a480d9293522faa8602f0c86896c6a08f46fd5b5dc85fb ./contracts/interfaces/IGasToken.sol
14d3330f46941eba86a72faf07b1eb5ff18140b44e12bc77a035d4db0d4c81db ./contracts/interfaces/IMintHandler.sol
0822240ff24cb9fdb59f1c74f66a5fab3b9833d7f90fda0d2c35d6ddcbd6affc ./contracts/interfaces/IBlockReward.sol
817d0221c83f2b2184c71155c1577b495ec6a8e251b871aacd265706534d1f84 ./contracts/interfaces/IBurnableMintableERC677Token.sol
324b4fa17b1caa6817aa9c37173f3bd8c12ce3cd6781a8438afa7930e5bfec63 ./contracts/interfaces/IUpgradeabilityOwnerStorage.sol
742e645486695512964c67890903d4bca06fe5e3b150795f70805ffc486e3d22 ./contracts/interfaces/ERC677.sol
17e88e73a58410b85f5f95d2b908b5a2a0b7a5db1cf65b7a93fe8167101bf65d ./contracts/interfaces/IMediatorFeeManager.sol
160e723ae76aec2f61f6fc6cb296d47ea485ed32c4df1ed018d56612dfc061f1 ./contracts/interfaces/IChai.sol
fcd0f053ac73391f42f8e46c862bb121eae9795f8e15f7ffc7047c9c720cbd6a ./contracts/interfaces/IBridgeValidators.sol
afd6cbf306b6cce7fed8acb497b9f509a54c458dc3150a0b7652a8e5709c0656 ./contracts/libraries/SafeERC20.sol
0c072f9bc39bc12f09086fbb8f9eb2a88ae7d51c02bfdc994e177be3bad7cc20 ./contracts/libraries/Address.sol
768606eca044e841f16e2cff9b987fe057cbdac4d384d2cea062c96de98f9d93 ./contracts/libraries/Bytes.sol
aba8c9afee315693f7dc6ca8ca84d5f24c687045d1780520023309d908b43bed ./contracts/libraries/Message.sol
d4d50163d7e682e44c74458b8faeacdeb1944ea517182b7c2a537dca07aaacbf ./contracts/libraries/TokenReader.sol
89b6e8928701208ec426fb900e6deae48cb1dd70a93d4071b95b28851d53a3f8 ./contracts/libraries/ArbitraryMessage.sol
973a28a487bbb142ff0a35c06ae2618153291135007383325 23af69ce8719471 ./contracts/upgradeable_contracts/Initializable.sol
c3d71c18e4a1873eb4af872b825f867dcbacae7c0d073a90e28a590960aa512b ./contracts/upgradeable_contracts/Claimable.sol
95110ed77b2b0c2847cfd84a48f72d0e984683302c8cee7ccda87b6f678168b2 ./contracts/upgradeable_contracts/BlockRewardFeeManager.sol
5d9e8a5c6fedd332a4d5ce008ec6180f2d5b069bc73ada092887891f1e47d2c5 ./contracts/upgradeable_contracts/BaseERC677Bridge.sol
3376a33dd88f7fa06319717e03f284817a554968b89be27796e9dcdacacd4dbf ./contracts/upgradeable_contracts/ReentrancyGuard.sol
24314103e67f89c017b5748dc9aa1f598af686694c6a1d469fc7069537c32463 ./contracts/upgradeable_contracts/InterestReceiver.sol
e5c915dbc8c92a2bb69732d8a1e2a853e17e40b85bbf1f22b45af83ad7bda458 ./contracts/upgradeable_contracts/Upgradeable.sol
7b3d62ec7ec5ed1ecebe83a0c9b2155d8bca52a1bd7e420bea51577e85836f7a ./contracts/upgradeable_contracts/ValidatorStorage.sol
b9edc98cf5f0b363ff1ea67a2270f63fae7bc029b66af7f928120959df1d00d9 ./contracts/upgradeable_contracts/BasicAMBMediator.sol
411fef90c618b33655ca3b2b2342c6c68d344696f19ff297976f882c2d0e74fa ./contracts/upgradeable_contracts/BasicBridge.sol
f655be52ed253aede05ade14a8ab995095e02d437e1cb72db5e3286ff024b822 ./contracts/upgradeable_contracts/VersionableBridge.sol
b394a164de3252a074b1755da83a0a7630bb273ebd80c331d07c5bb9539211c3 ./contracts/upgradeable_contracts/TransferInfoStorage.sol
844d471a9c2387fa634734ce6b694359fe72799ee9594f3264104a272aee31cb ./contracts/upgradeable_contracts/ERC677Bridge.sol
5247f2d0100d741ac1009656647cacf52fe04e971cf34c6ea792118eb83769e0 ./contracts/upgradeable_contracts/ERC677Storage.sol
286a2966e5bd4377f5f5130afb2dc1602cefb388c9b7f4690a8c717abe99bfb8 ./contracts/upgradeable_contracts/Sacrifice.sol
a936ffc60eeb5ebdde2a272ccc8bfdbe6f0f5169732db9843fd76470abfe112c ./contracts/upgradeable_contracts/BaseOverdrawManagement.sol
7093097da9c152c428d457057c3fbf94524798737d96081d55c13184bd96b8e7 ./contracts/upgradeable_contracts/TokenBridgeMediator.sol
678727bf4f7caf3c76c4c7541875612aa54ab1ff8ca67da52e7d74fd24b3ea2c ./contracts/upgradeable_contracts/Ownable.sol
1121c87b80b43ffa345fa609bc407b9d920a21d455859c741e2967471d629af9 ./contracts/upgradeable_contracts/InitializableBridge.sol
9a43d66482ae25be79a96024325b5648e3274d5da230a81eaa9b92c4223e4aac ./contracts/upgradeable_contracts/TokenSwapper.sol
cd29cc8b3ddcd69aa9d1c4daa007c0938a84958c172552300e6f7c04ad04f37a ./contracts/upgradeable_contracts/RewardableValidators.sol
```

c80542b92f5453a7f2d737c2638f55604d151a84220f2860693389b979c59e3f ./contracts/upgradeable_contracts/DecimalShiftBridge.sol

60ab1cdd2fda228eaf94b46757e62bdea737a3da42c6694a1868b00d04d746f7 ./contracts/upgradeable_contracts/BridgeValidators.sol

53b862c390b9f6802a04dd52658448f0bd524ac6011cf07f382b92993d2c28cc ./contracts/upgradeable_contracts/ValidatorsFeeManager.sol

873cebf1226bdf2721f2661fd098a722b289b4b74f654c4efacb10cbfacb2091 ./contracts/upgradeable_contracts/BasicTokenBridge.sol

aa285b4a017aa00e195e0948495850c9cec9df8cb014c4422ed81901b492f940 ./contracts/upgradeable_contracts/ChooseReceiverHelper.sol

7c804c0343f8199de472e179ca3cdd2176cd5e452e85b777376f705bb55d74ec ./contracts/upgradeable_contracts/BaseBridgeValidators.sol

5aae0bdbd1af2e7a315a548c0c2ef51a7304263eca5b6b6e704425e5b33f8f45 ./contracts/upgradeable_contracts/GasTokenConnector.sol

155b308fcc0d6d5f12ce0fd85e26b19736c34e29cce670fe619f07fb6bdab1ad ./contracts/upgradeable_contracts/Validatable.sol

3fce3378e41e78cdfa78ca1d6f4783191bb61ec4ce1bae61d4b8f379c9d4fdda ./contracts/upgradeable_contracts/BasicForeignBridge.sol

4a62bad21b3825f26f6e956e7de645728a5c9e9737da030fd2788fe34c5ea71f ./contracts/upgradeable_contracts/OverdrawManagement.sol

980f8b246a67c42057200782f67d7b432382720eae87b9d2ae7e2f384353b1a9 ./contracts/upgradeable_contracts/ERC20Bridge.sol

84e18369bd4101af6cfae75a94aeec22b76e90911e3cc44b2e1a45e5bc9e64f4 ./contracts/upgradeable_contracts/RewardableMediator.sol

54963c9f7d6a8f4377e34aa4e8dcff96acad0e33bc95989535ac3f1f7df46948 ./contracts/upgradeable_contracts/RewardableBridge.sol

70411ae02cbb01ff7a95fd77ae9b53cef45981a6226255882f157be4a8a0e15f ./contracts/upgradeable_contracts/ERC677BridgeForBurnableMintableToken.sol

0bbfab45cdd9a49314b8803175df9cafb8257c7b11b0ae46db93996cc320c0ea ./contracts/upgradeable_contracts/ChaiConnector.sol

9a4c68322877d28f5939625a5f5c2e86aa26fb983967680b19e70e5bc4faaeb2 ./contracts/upgradeable_contracts/BasicHomeBridge.sol

ad6908fe3dbcba2030cf5b22b8d002814aac6f64de2d3819ba0c5d42826e1031 ./contracts/upgradeable_contracts/BaseRewardAddressList.sol

7abe690fb824a9fda8dd7b12efc9cbac0ed64cc6c0ca56f22577c0b80ba14e8c ./contracts/upgradeable_contracts/OtherSideBridgeStorage.sol

2265f6d1b349455e11ceae1edc029d6303d0a769d18cd2110f34f769408e3d20 ./contracts/upgradeable_contracts/BlockRewardBridge.sol

576ec281e15c6e8e9c24801c81e68eb48ed0fffc7e1c725ee03c9c731b7481ec ./contracts/upgradeable_contracts/MessageRelay.sol

0df1671c6c07a2e3eb107d68aa71b98f41a9dd3716db9a0555abb1b74ff3f140 ./contracts/upgradeable_contracts/BaseFeeManager.sol

adea3fdef59299bcb3da4dc750f7eb5de99179f02faba416426c9f0437abb70a ./contracts/upgradeable_contracts/FeeTypes.sol

0d3d93a55245fc312c4a38ca7baebf5b39ff16e6f0b3b93c199a57b91e03799f ./contracts/upgradeable_contracts/BaseMediatorFeeManager.sol

1457dafdcfec201edf64e249e1afc0ca9013dc4213dab067bcf8e9b888a98aad
./contracts/upgradeable_contracts/amb_native_to_erc20/BasicAMBNativeToErc20.sol

a6fbdac97ba1ef075a4422a951113a85672cba71bc84890c370273ded8f745af
./contracts/upgradeable_contracts/amb_native_to_erc20/ForeignAMBNativeToErc20.sol

db99f19c35410911148692963dae38ceb2a9bfddd8fbdacdf1becc1398f3a35b
./contracts/upgradeable_contracts/amb_native_to_erc20/HomeAMBNativeToErc20.sol

4d1438e840cf914a43f7e00b473c26c576e6af7b30efc316ee1968694e44170d
./contracts/upgradeable_contracts/amb_native_to_erc20/ForeignFeeManagerAMBNativeToErc20.sol

d1d3e0d740e9fb512303474ed7cfc9216f3ac6da4933dbe9cfd92a0d5c5c359b
./contracts/upgradeable_contracts/amb_native_to_erc20/HomeFeeManagerAMBNativeToErc20.sol

7e4fdb34a53c4dd72acf4a26773b56daeb8d873cce3f76aac64c53f6726a2706
./contracts/upgradeable_contracts/erc20_to_native/FeeManagerErcToNativePOSDAO.sol

94152d9666898413040710 42fe7e177ea7606aa9fc1b06b46b4f64e981a1ffba
./contracts/upgradeable_contracts/erc20_to_native/ForeignBridgeErcToNative.sol

c6f274e636c8b19569b4e19591621bd616a7c10136253a69077bbceeb484c081
./contracts/upgradeable_contracts/erc20_to_native/RewardableHomeBridgeErcToNative.sol

82990c81b5f8e04f4a23d309fe5c5e7549e6db1bac5db3c8aa15c18ec4bc6bb0 ./contracts/upgradeable_contracts/erc20_to_native/HomeBridgeErcToNative.sol

5861753d60f765c99b1da2730aa70fa6150c1e52f5e7d9c7ee409ff5a29018be ./contracts/upgradeable_contracts/erc20_to_native/FeeManagerErcToNative.sol

a2493996ff49ea15cabec5aac1f76599ffab4b5d212f87e27676dd519a59bb82 ./contracts/upgradeable_contracts/erc20_to_erc20/HomeBridgeErcToErc.sol

b93a43d94a3a325310e8c6f41cbacb15d777cd673ec0a4d30964d48f99ff715b
./contracts/upgradeable_contracts/erc20_to_erc20/ForeignBridgeErc677ToErc677.sol

ec7bad8099d9a4ee7c2e058e10cdd97e9d72f9820b68ebea96e00cf69ba9a895
./contracts/upgradeable_contracts/erc20_to_erc20/BasicForeignBridgeErcToErc.sol

bf8fef3c18cfe75f5e7d90e8b1cc48875fdc6e98f355b760e3aad77404d4f4ac ./contracts/upgradeable_contracts/erc20_to_erc20/HomeBridgeErcToErcPOSDAO.sol

db2dadfc940a17eef402bb0fd1aa55576f5b7877ba6668d3e7918a5e170324d5 ./contracts/upgradeable_contracts/erc20_to_erc20/ForeignBridgeErcToErc.sol

ca49ec912942706c94eeac6270ecb4eb300371d8f1d70ca201770649dd823274 ./contracts/upgradeable_contracts/erc20_to_erc20/FeeManagerErcToErcPOSDAO.sol

b2c9f352a00b0bbdf3ca0025722d84cf569cd80adb25e5b7bdd654ab5349a13e
./contracts/upgradeable_contracts/erc20_to_erc20/RewardableHomeBridgeErcToErc.sol

148f35ded408b6d77aaf11758dada7c994fe4dfbcd23e39116eb9d6e733e686d
./contracts/upgradeable_contracts/amb_erc20_to_native/HomeFeeManagerAMBErc20ToNative.sol

3ecc021e537d13f85faf95d6df588db3cced3c5c17eaa2b1cb067ff3f5b802f7 ./contracts/upgradeable_contracts/amb_erc20_to_native/BlockReward.sol

949ba1dd77a5c9b37b80df08945572acd0b0278a27e1460f8e444f80dbab4dc7
./contracts/upgradeable_contracts/amb_erc20_to_native/HomeAMBErc20ToNative.sol

968678afaee4695d8af8d0542c0b666dc5985e5830c100f0077f5ac6811621ab
./contracts/upgradeable_contracts/amb_erc20_to_native/BasicAMBErc20ToNative.sol

5306660783c47e69f164c5652e514e3c21cfd72f48336f1b4366f4ed3261526c
./contracts/upgradeable_contracts/amb_erc20_to_native/ForeignAMBErc20ToNative.sol

2b487ac4340488f3d10711cce325d9de02257cc459f262acaf433512e5196791
./contracts/upgradeable_contracts/native_to_erc20/ForeignBridgeNativeToErc.sol

f14b169021d0f544ce1c8ed8412548df4df98c0012172813e42f9399cee44ce4
./contracts/upgradeable_contracts/native_to_erc20/RewardableHomeBridgeNativeToErc.sol

17b1fd635e27f3158fd662edb218c5612c2fe698d4ef0e749ef651097e99fd9b
./contracts/upgradeable_contracts/native_to_erc20/FeeManagerNativeToErcBothDirections.sol

115a664c37dbdfdc9308442c559543fddea4719e3278e3dc57f1e1a35d5d37e5 ./contracts/upgradeable_contracts/native_to_erc20/FeeManagerNativeToErc.sol

99b96a5d47e6a9cf5a8d9346ff3d2dd0c2f17d0737e970067d8dbe7f2789d37d ./contracts/upgradeable_contracts/native_to_erc20/HomeBridgeNativeToErc.sol

c3f0f1f9d038040b83300890d4c6ae6ba4c1e3773e996aa0f92756fbc4754598
./contracts/upgradeable_contracts/native_to_erc20/RewardableForeignBridgeNativeToErc.sol

5612c73d90431dbf29e10f38e6b432557a8b926f1f5124fdd6aa2e68ff1abc9b ./contracts/upgradeable_contracts/arbitrary_message/MessageDelivery.sol

af647f637a210543d7be3a0192e7d8ed739dbc28a453bab7fbb602a77a67878b ./contracts/upgradeable_contracts/arbitrary_message/BasicHomeAMB.sol

7db11d58c609d26496d969f2cdca4f5968e862f25f8f079a762e8a45b0e3946f ./contracts/upgradeable_contracts/arbitrary_message/BasicForeignAMB.sol

d4cbc4cc21ad3e4b086b37b125e85837b75f73ca84ec4d7f5675ac042c7697bf ./contracts/upgradeable_contracts/arbitrary_message/HomeAMB.sol

2610dba4cd24b4d11dc1af1c1b5c308f76eecbe036a396eb0a5508545f836425 ./contracts/upgradeable_contracts/arbitrary_message/BasicAMB.sol

8cf49071dc5dced1c02e1f51a3df8a00545f80ac2b8c5394355f79da5a13b8f3 ./contracts/upgradeable_contracts/arbitrary_message/MessageProcessor.sol

a6e3c668c258b4c13b9585ae4d1459a3d3f230ac823ff2730161ef4dc69cdda7 ./contracts/upgradeable_contracts/arbitrary_message/VersionableAMB.sol

2bc1c9308e287646b3b4c9167feaf3001278c4205573f00fc4b43a058f69eee7
./contracts/upgradeable_contracts/arbitrary_message/ForeignAMBWithGasToken.sol

eed670f5a9a3561f9c521ea5aedde5f17139cbe31c55d5a2337508d93c12f05b ./contracts/upgradeable_contracts/arbitrary_message/ForeignAMB.sol

6faff25f19cbf54fcfa573890450e7bce5d01c36c5f193bdf3f65edbf3f27701
./contracts/upgradeable_contracts/amb_erc677_to_erc677/HomeStakeTokenFeeManager.sol

8ef10f67fced1509ab0c8d7147f7d35cef5e7353c608792af3f95efd957ff5bf
./contracts/upgradeable_contracts/amb_erc677_to_erc677/HomeAMBErc677ToErc677.sol

c6ae0a512b8c88face042a273cb3c2bb8773d51c622397534984a092b1890772
./contracts/upgradeable_contracts/amb_erc677_to_erc677/ForeignStakeTokenMediator.sol

b9f9cc3d71351327ca9747d76bc4aab3e3d7644c4c4ed0c253c8c103c97c6f0f
./contracts/upgradeable_contracts/amb_erc677_to_erc677/HomeStakeTokenMediator.sol

92cf1a87a395a1a011066437ad13ea6121f324a710d8148f2d7211c3003f3e8c
./contracts/upgradeable_contracts/amb_erc677_to_erc677/BasicAMBErc677ToErc677.sol

75925e212b989422b5e2d1d807bec81f7edd9cdfb18684d486acc78008a83c42
./contracts/upgradeable_contracts/amb_erc677_to_erc677/ForeignAMBErc677ToErc677.sol

29f5f34c890d279dba16c44fd5ce056f7d4115d0a815573f89abe17b10cae694
./contracts/upgradeable_contracts/amb_erc677_to_erc677/BasicStakeTokenMediator.sol

b03587e467ae44e077bfbe7882c9010fe3f98baea5248ec8ed0ff8c8c9c942da
./contracts/upgradeable_contracts/multi_amb_erc20_to_erc677/HomeFeeManagerMultiAMBErc20ToErc677.sol

e27d0d7f05b28059abf2076ecd4e31b76b6a135b86cf96139d46954ad92b17e5
./contracts/upgradeable_contracts/multi_amb_erc20_to_erc677/HomeMultiAMBErc20ToErc677.sol

4849f7597ea073b14594525847e0e2ba164b902fc0ab10f11ea652b4c67eecbb
./contracts/upgradeable_contracts/multi_amb_erc20_to_erc677/ForeignMultiAMBErc20ToErc677.sol

effef23478d97c3e263a766e78c45668901f7cfadffb7a6e4fe9a576926adcbe
./contracts/upgradeable_contracts/multi_amb_erc20_to_erc677/BasicMultiTokenBridge.sol

902fcddac8fe80105bcfdf42275f03e1bfff1838a90d270e397d29457adf1f1e ./contracts/upgradeable_contracts/multi_amb_erc20_to_erc677/TokenProxy.sol

2899b4968924d944ceefac5bdeeb37c48c0115f0ec9ac6332b155f386d5fd768
./contracts/upgradeable_contracts/multi_amb_erc20_to_erc677/BasicMultiAMBErc20ToErc677.sol

389bdf78a4ec4294357d7d28a0b7de61b416b0a31cd30871b0e93fab76474c79
./contracts/upgradeable_contracts/multi_amb_erc20_to_erc677/MultiTokenBridgeMediator.sol

0a0b223ec3a5a08e5af2c6dfb95ec7b89f6e25af2e818b7003b7305b2a7c3034 ./contracts/mocks/TokenReaderTest.sol

3c1f9fe0c87839a696bf4da6d64a31699700faf3b9239874b1da87579f1bd643 ./contracts/mocks/OldBlockReward.sol

0eebc563418b8fc8e798966b16295e060028579e92fc55c86dc9d2219a414520 ./contracts/mocks/GasTokenMock.sol

1b37704d1eb0b8f61ab7dcfca0c781c3bff8bf972eceb46f84b828ea9883a6ff ./contracts/mocks/RevertFallback.sol

5df2f379862559e87a37b25d1f0f572926552024d9e5b71424558ab8ed461c35 ./contracts/mocks/FeeReceiverMock.sol

06a780f44f0a3dadbf3019cc5b4f5749c771121bcabd6cda543b7b1fd4c3a538 ./contracts/mocks/AMBMock.sol

ddf2de0f673cf60a8137c8500b749d3d9ee04e8d348f9feeef8441698bc83aab ./contracts/mocks/MintHandlerMock.sol

b67c503638679a3b91406b705f28e4656f3f829007a108d26c0f270b7f3ea4e3 ./contracts/mocks/BlockRewardWithoutSystem.sol

78643ebd0166f8ee7f0397ea49f9118f15a85910ec185b6442beed77a75a16ad ./contracts/mocks/BridgeValidatorsDeterministic.sol

a5140b456c40429285fc73121800d9edf8d6f7eaa5351e7a4ca898e471d16790 ./contracts/mocks/MessageTest.sol

832981f1faa0503599f7e96fe752ff8d875e24e7f59ee88328f427a4c73cb092 ./contracts/mocks/ERC20Mock.sol

0d664924bd4217e75864aea799c35c60a5bcb100f6ebd3e5bc1f5ad2dded0d65 ./contracts/mocks/BlockRewardMock.sol

d5c58f097e6cddb5189f2c91e33f61eb2f1f6d44e0c98add19413b91b3029773 ./contracts/mocks/PotMock.sol

e0c9f4e5c35650671fd575c8537449c0dae930243e100be05b3d4245dda57105 ./contracts/mocks/ERC677BridgeTokenRewardableMock.sol

d905d03fff86342dd257d7d70bbbcf3b2e70d8637a668505661d70de5f858a20 ./contracts/mocks/InterestReceiverMock.sol

5a9319b305187594503ce8d38b94f8a7c61f1ee1bd3d4bd283606fac76176c96 ./contracts/mocks/ChaiMock2.sol

4a195bd71878dacf872527ba6281fab9320d4dfeb4274e82529fd06a245d2380 ./contracts/mocks/VatMock.sol

af24aabc14c29dc5630823477b57b66742832cd34427a614364b8447947181fa ./contracts/mocks/PotMock2.sol

149e1384536b12d24c3c518bd928e0203339cd46449ea398e5f92450feb947a4 ./contracts/mocks/ForeignBridgeV2.sol

bb82ee6400a39b9029c2a8891edaa43b55b8bb89dd61b055fd3c9385ba14f17c ./contracts/mocks/ChaiMock.sol

9ec15681a811f3315eb76aaa59d7f33bc809399f713b12c96e3f52deedba0d2c ./contracts/mocks/PermittableTokenMock.sol

6c70b40317cf6ae724b33ac23535cb0434839a6f9eee079343d435c97d832752 ./contracts/mocks/NoReturnTransferTokenMock.sol

375fc3424ed3da0faa1c876bd75430f6f02da85bd5337fb87942d5a93654d28e ./contracts/mocks/ForeignAMBWithGasTokenMock.sol

e2172fb8960208e506041d5665c1572e5b24c93be3cbaf01a6f94623fea60639 ./contracts/mocks/Box.sol

e3b8f38a377280163c9b85dba4307f3fa711199337322fc6bf5ee033df2edeb8 ./contracts/mocks/ForeignBridgeErcToNativeMock.sol

ab5719b609fca66ef530b96b6024016aa2404a8557c9ae42aa5ee3b25699b7aa ./contracts/mocks/DaiMock.sol

afa64355d703716429604a0dfe7bc3b7d5e2ee3576e01f3094334972b713af6f ./contracts/mocks/ERC677ReceiverTest.sol

0ae0a3796b2ef07f20d416cbc6ac3291a29fc476d3696179530337dc87b44a0b ./contracts/mocks/DaiAdapterMock.sol

bd1517335ee79f8cf28f37f04fcfa1fe8f428d04bb284c2cfd67290f65df2b01 ./contracts/mocks/FeeManagerMock.sol

2c0ab7e911c867e81eee4b399a9679811b1c5925756e6abea1c4739383204985 ./contracts/mocks/DaiJoinMock.sol

a89b26867d994d9a61b1194e89b523fc1db72d7ece57b124c5e53e8d939e1cf1 ./contracts/mocks/Staking.sol

Tests

a3443845e4ed99811af6bb5a0cc829d8a7cea72d5d8cf17667ed2fbaf56cf005 ./test/validators_test.js

4a6ace388b63a50583ea9b140109ec7ca823f36319fadc8b3ef906d6c916ea20 ./test/rewardable_validators_test.js

331e11306ee16500972d30a19ed45aba4692f6ce8cc5abc730a3f2668f8c7b6f ./test/poa20_test.js

545e86ddae221210212213d85968817a3f94d4d0ecde47e15fe4b09549dc392b ./test/setup.js

397903dec11741aca5ad04f45d628fc04b64fbdfbf32065fcd5a5637851a6949 ./test/coverage.test.js

def0447aa077db4ef6b8c188976e4f09e3bd619decf862e9ca4ac0fd86679671 ./test/erc677MultiBridgeToken.test.js

5fa1a389f30e0f367b45bf71093b59b97f7dc2ff977e61f797e69f038f0228fc ./test/helpers/helpers.js

65f7b11f76945394957c6f578edf60532499b88149efad233affa19958ced4a7 ./test/helpers/eip712.sign.permit.js

db1728095a3eee88284b7593641f7777b6dd05517256fb40def28ce126e2ce38 ./test/amb_native_to_erc20/home_mediator.test.js

a55fd6b23e4616db2e5b39b7f624f688cd971cc39c7be9ce5280bb4f59957e54 ./test/amb_native_to_erc20/foreign_mediator.test.js

7b5ae6d894c759a0174dbcba04c7720debceaabd1f8e9864aaf402e8830b74a6 ./test/erc_to_native/home_bridge.test.js

155d647a26b4bad49c42b58bc74ca9fc8d08956ed8d0c696215b89be5f4baf50 ./test/erc_to_native/foreign_bridge.test.js

0f924478e283358dcf150fdf11222856c5c596c9cc777603e3f52ffe8e27ee0d ./test/native_to_erc/foreign_bridge_test.js

dd017f9514909d6711007ba7fdac9d58a17990ba98efe9eb08bef9177725546b ./test/native_to_erc/home_bridge_test.js

b97589c9aaa6e15e1bed93370a9cbc560496beaca575afaea06ed3c9c95cf648 ./test/erc_to_erc/home_bridge.test.js

4c23a84188a351fbf91130c5c6ca5b1141545d3e79752aa97489009826d95baf ./test/erc_to_erc/foreign_bridge.test.js

d4efcfecda3606b132214a4f542b5124e2a5594fd8e1d577a97f62f4e04bf051 ./test/amb_erc20_to_native/home_mediator.test.js

28a589ef1072152d33b527589e237b77a540f2ee4277cf7b99df82ad021429f2 ./test/amb_erc20_to_native/foreign_mediator.test.js

c6fcef002ac55fed2a28f77aabe01027f0d6d345ac1f8ff8d69c29548a59cf07 ./test/arbitrary_message/home_bridge.test.js

999782fa01b2a47dd892faaf063667d43886fbf2431ffe08955d3e601279db92 ./test/arbitrary_message/foreign_bridge.test.js

64d8726001b524710c61204f4aa08a6e47f2fcf7a5c3ba3ceb1b4d03af0aaf6b ./test/libraries/arbitraryMessage.test.js

5faf98c067547a40435610bb958414dfe60e4f26506a7a1d3d27feaf2d0fd13f ./test/libraries/address.test.js

9923239796b73ab483eab8248e8e95ed0199758c6e692b448d9386a8a49fd314 ./test/libraries/tokenReader.test.js

5b31f9e7877a6fbb7d650236eae2026057bcb92e26a84bacd3840e30a546e1b4 ./test/stake_token_mediators/home_mediator.test.js

d166200c14b903e4aed2e53882b10c61e03c030955c4bdd3562a5ecf35df0050 ./test/stake_token_mediators/foreign_mediator.test.js

dacd112f25566e9b518ff0864542efbdf6872e4db6804a172ecafc9fe394add4 ./test/amb_erc677_to_erc677/AMBErc677ToErc677Behavior.test.js

d0a2b9901311628ba2489d2a7877688e92e543a13eb46b135848be20bb2f04ba ./test/amb_erc677_to_erc677/home_bridge.test.js

cb755200cf052fab4a2f13e498c6edb8b01bf9ab1f5ed74f619892e1596c2871 ./test/amb_erc677_to_erc677/foreign_bridge.test.js

5abfb20d5aab9d6ec3efe758ecdfcc024b3eef1e1558e0f01ad8e272b30081ae ./test/multi_amb_erc20_to_erc677/home_mediator.test.js

03b9727e2a2204a2eeaf8ca03bef38de69fc723c5f9a0b767332d8dbf5da2c6b ./test/multi_amb_erc20_to_erc677/foreign_mediator.test.js

Changelog

- 2020-10-02 - Initial report
- 2020-10-27 - Updated report
- 2020-11-05 - Updated report with PR references

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure blockchain platforms at scale using computer-aided reasoning tools, with a mission to help boost the adoption of this exponentially growing technology.

With over 1000 Google scholar citations and numerous published papers, Quantstamp's team has decades of combined experience in formal verification, static analysis, and software verification. Quantstamp has also developed a protocol to help smart contract developers and projects worldwide to perform cost-effective smart contract security scans.

To date, Quantstamp has protected $5B in digital asset risk from hackers and assisted dozens of blockchain projects globally through its white glove security assessment services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Quantstamp's collaborations with leading academic institutions such as the National University of Singapore and MIT (Massachusetts Institute of Technology) reflect our commitment to research, development, and enabling world-class blockchain security.

Timeliness of content

The content contained in the report is current as of the date appearing on the report and is subject to change without notice, unless indicated otherwise by Quantstamp; however, Quantstamp does not guarantee or warrant the accuracy, timeliness, or completeness of any report you access using the internet or other means, and assumes no obligation to update any information following publication.

Notice of confidentiality

This report, including the content, data, and underlying methodologies, are subject to the confidentiality and feedback provisions in your agreement with Quantstamp. These materials are not to be disclosed, extracted, copied, or distributed except to the extent expressly authorized by Quantstamp.

Links to other websites

You may, through hypertext or other computer links, gain access to web sites operated by persons other than Quantstamp, Inc. (Quantstamp). Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites' owners. You agree that Quantstamp are not responsible for the content or operation of such web sites, and that Quantstamp shall have no liability to you or any other person or entity for the use of third-party web sites. Except as described below, a hyperlink from this web site to another web site does not imply or mean that Quantstamp endorses the content on that web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the report. Quantstamp assumes no responsibility for the use of third-party software on the website and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

Disclaimer

This report is based on the scope of materials and documentation provided for a limited review at the time provided. Results may not be complete nor inclusive of all vulnerabilities. The review and this report are provided on an as-is, where-is, and as-available basis. You agree that your access and/or use, including but not limited to any associated services, products, protocols, platforms, content, and materials, will be at your sole risk. Blockchain technology remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language, or other programming aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security. No third party should rely on the reports in any way, including for the purpose of making any decisions to buy or sell a product, service or any other asset. To the fullest extent permitted by law, we disclaim all warranties, expressed or implied, in connection with this report, its content, and the related services and products and your use thereof, including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. We do not warrant, endorse, guarantee, or assume responsibility for any product or service advertised or offered by a third party through the product, any open source or third-party software, code, libraries, materials, or information linked to, called by, referenced by or accessible through the report, its content, and the related services and products, any hyperlinked websites, any websites or mobile applications appearing on any advertising, and we will not be a party to or in any way be responsible for monitoring any transaction between you and any third-party providers of products or services. As with the purchase or use of a product or service through any medium or in any environment, you should use your best judgment and exercise caution where appropriate. FOR AVOIDANCE OF DOUBT, THE REPORT, ITS CONTENT, ACCESS, AND/OR USAGE THEREOF, INCLUDING ANY ASSOCIATED SERVICES OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, INVESTMENT, TAX, LEGAL, REGULATORY, OR OTHER ADVICE.