November 25, 2023

Alethea AI - Bonding curve contracts Audit

Contents

1. Findings

1	Consider accumulating fees instead of transfering directly
2	prefer encodeWithSelector over encodeWithSignature
3	receive/fallback functions accept ETH for no apparent reason
4	Transfers.transfer may not use enough gas
5	Merkle root can be arbitrarily changed
6	receive() accepts funds even after \`lastRewardBlock\` has expired
7	fallback usage seems unnecessary
8	fees can exceed 100%

2. Detailed Findings

2.1. Consider accumulating fees instead of transfering directly

Functions such as `ETHShares.__processProtocolFee` and `ETHShares.__processHoldersFeeAndNotify` drip fees to their destination addresses. Depending on the gas constrains of the target network, and the usage of the protocol, it may be worth considering accumulating these fees in the contract, and adding a permissioned ability to withdraw them, thereby reducing the total gas cost of each transaction

2.2. prefer encodeWithSelector over encodeWithSignature

There are several calls to `encodeWithSelector` with a fully hardcoded function signature. When this happens, it is much more efficient to use instead `encodeWithSignature` with the appropriate ABI selector.

This both helps reduce total gas costs, but also reduce the chance for developer mistakes (as any mistyped selector will be caught by the compiler)

example:

```
- abi.encodeWithSignature("owner()")
+ abi.encodeWithSelector(Ownable.owner.selector)
```

2.3. receive/fallback functions accept ETH for no apparent reason

`RewardSystem.sol` include both `receive()` and `fallback()` callbacks which fully accept ETH and silently ignore errors. This happens regardless of the sender or current contract state. This choice is an antipattern, as it removes the security `fallback()` is meant to introduce (i.e.: by default, contracts don't accept mistakenly sent ETH).

It is recommended to include checks that ensure ETH can only be sent under the right circunstances

2.4. Transfers.transfer may not use enough gas

`RewardSystem.claimReward` uses an internal function, `Transfers.transfer`, which limits the ETH transfer to use only 4900 gas. This is not enough to cover most smart contract accounts. For example, transfers to Gnosis Safes vaults usually take nearly 30000 gas to complete.

It should be clarified whether this is a desired trade-off, or alternatively, adjust the limit gas

2.5. Merkle root can be arbitrarily changed

The merkle root of the rewards tree can be changed by the owner at any time, without any sort of timelock, or ensurance that previous tree leafs will be able to claim their past pending amount.

It is not clear whether this is intentional or not, but it is widely regarded as an antipattern.

2.6. receive() accepts funds even after `lastRewardBlock` has expired

The `HoldersRewardsDistributorV1.receive()`, which internally calls `__accept()`, receives ETH from arbitrary users, and uses it as rewards to be distributed.

However, rewards stop accumulating after `lastRewardBlock` is reached. In that scenario, `receive()` will still work, since `__accept()` simply returns instead of reverting, but any ETH sent ends up ignored and locked forever.

It is recommended to explicitly revert in this scenario.

2.7. fallback usage seems unnecessary

The `HoldersRewardsDistributorV1.fallback()` function seems intended to handle calls from function such as `ETHShares.__processsHoldersFeeAndNotify()` where the following logic is defined:

```
bytes memory syncMessage = abi.encode(trader, isBuy, amount);
(bool success, ) = address(holdersFeeDestination).call{value: holdersFee}
(syncMessage);
```

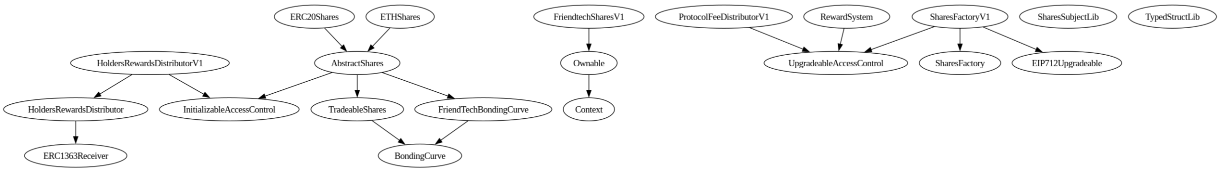
It appears the same could be achieved by defining a proper `processTrade(address, bool, uint256)` in the target contract, instead of manually encoding/decoding the payload.

2.8. fees can exceed 100%

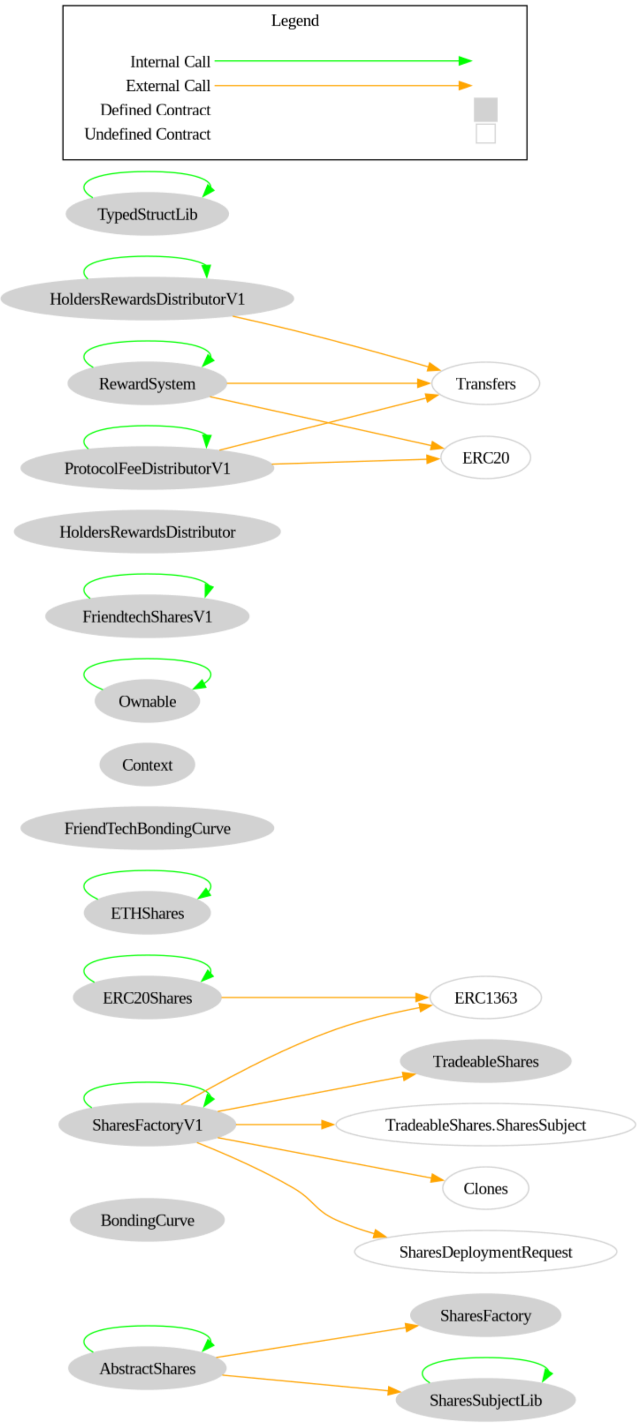
`SharesFactoryV1.setProtocolFeeDestination` and other similar fee-setting functions accept any value greater than 0.

At no point is it verified whether the cumulative value of all fees exceeds 100%. Exceeding such value, either intentionally or by mistake, could result in the protocol griefing its own users since, for example, it would cause all calls to `ETHShares.sellSharesTo()` to revert due to overflow.

3. Inheritance Graph



4. Dependency Graph



5. Checklist

5.1.1.1. bonding_curves/HoldersRewardsDistributor.sol

1	reviewed	`event SharesTraded(...)`
2	reviewed	`event FeeReceived(...)`
3	reviewed	`event RewardClaimed(...)`
4	reviewed	`function getPaymentToken(...)`
5	reviewed	`function sharesBought(...)`
6	reviewed	`function sharesSold(...)`
7	reviewed	`function accept(...)`
8	reviewed	`function claimTheReward(...)`
9	reviewed	`function pendingReward(...)`

5.1.1.2. bonding_curves/TypedStructLib.sol

1	reviewed	`function hashType(...)`
2	reviewed	`function hashType(...)`
3	reviewed	`function hashStruct(...)`
4	reviewed	`function hashStruct(...)`

5.1.1.3. bonding_curves/SharesFactory.sol

1	reviewed	`enum ImplementationType`
2	reviewed	`struct SharesDeploymentRequest`
3	reviewed	`event ProtocolFeeUpdated(...)`
4	reviewed	`event SharesContractRegistered(...)`
5	reviewed	`event NonceUsed(...)`
6	reviewed	`function getSharesImplAddress(...)`
7	reviewed	`function getDistributorImplAddress(...)`
8	reviewed	`function getProtocolFeeDestination(...)`
9	reviewed	`function getProtocolFeePercent(...)`
10	reviewed	`function getHoldersFeePercent(...)`
11	reviewed	`function getSubjectFeePercent(...)`
12	reviewed	`function setProtocolFeeDestination(...)`
13	reviewed	`function setProtocolFeePercent(...)`
14	reviewed	`function setHoldersFeePercent(...)`
15	reviewed	`function setSubjectFeePercent(...)`
16	reviewed	`function setProtocolFee(...)`
17	reviewed	`function deploySharesContractPaused(...)`
18	reviewed	`function deploySharesContract(...)`
19	reviewed	`function deploySharesContractAndBuy(...)`
20	reviewed	`function mintSubjectAndDeployShares(...)`
21	reviewed	`function executeDeploymentRequest(...)`
22	reviewed	`function getNonce(...)`
23	reviewed	`function rewindNonce(...)`

24	reviewed	`function lookupSharesContract(...)`
25	reviewed	`function registerSharesContract(...)`
26	reviewed	`function notifySubjectUpdated(...)`

5.1.1.4. bonding_curves/FriendtechSharesV1.txt

1	reviewed	`function _msgSender(...)`
2	reviewed	`function _msgData(...)`
3	reviewed	`address private _owner`
4	reviewed	`event OwnershipTransferred(...)`
5	reviewed	`constructor(...)`
6	reviewed	`modifier onlyOwner(...)`
7	reviewed	`function owner(...)`
8	reviewed	`function _checkOwner(...)`
9	reviewed	`function renounceOwnership(...)`
10	reviewed	`function transferOwnership(...)`
11	reviewed	`function _transferOwnership(...)`
12	reviewed	`address public protocolFeeDestination`
13	reviewed	`uint256 public protocolFeePercent`
14	reviewed	`uint256 public subjectFeePercent`
15	reviewed	`event Trade(...)`
16	reviewed	`mapping(address => mapping(address => uint256)) public sharesBalance`
17	reviewed	`mapping(address => uint256) public sharesSupply`
18	reviewed	`function setFeeDestination(...)`
19	reviewed	`function setProtocolFeePercent(...)`
20	reviewed	`function setSubjectFeePercent(...)`
21	reviewed	`function getPrice(...)`
22	reviewed	`function getBuyPrice(...)`
23	reviewed	`function getSellPrice(...)`
24	reviewed	`function getBuyPriceAfterFee(...)`
25	reviewed	`function getSellPriceAfterFee(...)`
26	reviewed	`function buyShares(...)`
27	reviewed	`function sellShares(...)`

5.1.1.5. bonding_curves/ERC20Shares.sol

1	reviewed	`ERC1363 private /*immutable*/ paymentToken`	
2	reviewed	`constructor(...)`	
3	reviewed	`function postConstruct(...)`	
4	reviewed	`function getPaymentToken(...)`	
5	reviewed	`function buyShares(...)`	
6	reviewed	`function sellShares(...)`	
7	reviewed	`function buySharesTo(...)`	fees calculated but not transfered
8	reviewed	`function __buySharesTo(...)`	

9	reviewed	function sellSharesTo(...)
10	reviewed	function __processProtocolFee(...)
11	reviewed	function __processHoldersFeeAndNotify(...)
12	reviewed	function __processSubjectFee(...)

5.1.1.6. bonding_curves/AbstractShares.sol

1	reviewed	SharesSubject private sharesSubject
2	reviewed	address private protocolFeeDestination
3	reviewed	uint64 private /*immutable*/ protocolFeePercent
4	reviewed	HoldersRewardsDistributor private /*immutable*/ holdersFeeDestination
5	reviewed	uint64 private /*immutable*/ holdersFeePercent
6	reviewed	uint64 private /*immutable*/ subjectFeePercent
7	reviewed	uint256 internal sharesSupply
8	reviewed	mapping(address => uint256) internal sharesBalances
9	reviewed	event SharesSubjectUpdated(...)
10	reviewed	event ProtocolFeeDestinationUpdated(...)
11	reviewed	event HoldersFeeDisabled(...)
12	reviewed	uint32 public constant ROLE_PROTOCOL_FEE_MANAGER
13	reviewed	uint32 public constant ROLE_HOLDERS_FEE_MANAGER
14	reviewed	uint32 public constant ROLE_SHARES_SUBJECT_MANAGER
15	reviewed	function _postConstruct(...)
16	reviewed	function getSharesSubject(...)
17	reviewed	function updateSharesSubject(...)
18	reviewed	function updateSharesSubject(...)
19	reviewed	function getProtocolFeeDestination(...)
20	reviewed	function updateProtocolFeeDestination(...)
21	reviewed	function getProtocolFeePercent(...)
22	reviewed	function getProtocolFeeInfo(...)
23	reviewed	function getHoldersFeeDestination(...)
24	reviewed	function disableHoldersFee(...)
25	reviewed	function getHoldersFeePercent(...)
26	reviewed	function getHoldersFeeInfo(...)
27	reviewed	function getSubjectFeeInfo(...)
28	reviewed	function getSubjectFeePercent(...)
29	reviewed	function getSharesIssuer(...)
30	reviewed	function getSharesBalance(...)
31	reviewed	function getSharesSupply(...)
32	reviewed	function getBuyPrice(...)
33	reviewed	function getSellPrice(...)
34	reviewed	function getBuyPriceAfterFee(...)
35	reviewed	function getSellPriceAfterFee(...)
36	reviewed	function getBuyPrice(...)
37	reviewed	function getSellPrice(...)

38	reviewed	function getBuyPriceAfterFee(...)	
39	reviewed	function getSellPriceAfterFee(...)	

5.1.1.7. bonding_curves/ProtocolFeeDistributorV1.sol

1	reviewed	struct RecipientDetails	
2	reviewed	RecipientDetails[] private recipients	
3	reviewed	ERC20 private /*immutable*/ paymentToken	
4	reviewed	uint8 public MAX_RECIPIENTS_ALLOWED	
5	reviewed	uint32 public constant ROLE_RECIPIENT_LIST_MANAGER	
6	reviewed	uint32 public constant ROLE_DISTRIBUTION_MANAGER	
7	reviewed	event ETHReceived(...)	
8	reviewed	event ETHSent(...)	
9	reviewed	event ERC20Sent(...)	
10	reviewed	event RecipientsListUpdated(...)	
11	reviewed	function postConstruct(...)	
12	reviewed	function getPaymentToken(...)	
13	reviewed	receive(...)	
14	reviewed	function distributeETH(...)	may fail if recipients are too large
15	reviewed	Transfers.transfer(...)	
16	reviewed	emit ETHSent(...)	
17	reviewed	function distributeERC20(...)	
18	reviewed	require(...)	
19	reviewed	emit ERC20Sent(...)	
20	reviewed	function updateRecipientsList(...)	
21	reviewed	function getRecipientsLength(...)	
22	reviewed	function getRecipient(...)	
23	reviewed	function getRecipients(...)	

5.1.1.8. bonding_curves/ETHShares.sol

1	reviewed	constructor(...)	
2	reviewed	function postConstruct(...)	
3	reviewed	function buyShares(...)	
4	reviewed	function sellShares(...)	
5	reviewed	function buySharesTo(...)	
6	reviewed	function __buySharesTo(...)	
7	reviewed	function sellSharesTo(...)	
8	reviewed	function __processProtocolFee(...)	1
9	reviewed	function __processHoldersFeeAndNotify(...)	
10	reviewed	function __processSubjectFee(...)	

5.1.1.9. bonding_curves/TradeableShares.sol

1	reviewed	`struct SharesSubject`
2	reviewed	`event Trade(...)`
3	reviewed	`function getSharesSubject(...)`
4	reviewed	`function getProtocolFeeDestination(...)`
5	reviewed	`function getProtocolFeePercent(...)`
6	reviewed	`function getProtocolFeeInfo(...)`
7	reviewed	`function getHoldersFeeDestination(...)`
8	reviewed	`function getHoldersFeePercent(...)`
9	reviewed	`function getHoldersFeeInfo(...)`
10	reviewed	`function getSubjectFeeInfo(...)`
11	reviewed	`function getSubjectFeePercent(...)`
12	reviewed	`function getSharesIssuer(...)`
13	reviewed	`function getSharesBalance(...)`
14	reviewed	`function getSharesSupply(...)`
15	reviewed	`function getBuyPrice(...)`
16	reviewed	`function getSellPrice(...)`
17	reviewed	`function getBuyPriceAfterFee(...)`
18	reviewed	`function getSellPriceAfterFee(...)`
19	reviewed	`function getBuyPrice(...)`
20	reviewed	`function getSellPrice(...)`
21	reviewed	`function getBuyPriceAfterFee(...)`
22	reviewed	`function getSellPriceAfterFee(...)`
23	reviewed	`function buyShares(...)`
24	reviewed	`function buySharesTo(...)`
25	reviewed	`function sellShares(...)`
26	reviewed	`function sellSharesTo(...)`

5.1.1.10. bonding_curves/FriendTechBondingCurve.sol

1	reviewed	`function getPrice(...)`

5.1.1.11. bonding_curves/SharesSubjectLib.sol

1	reviewed	`function getSharesIssuer(...)`	2
2	reviewed	`    if(...)`	
3	reviewed	`    if(...)`	
4	reviewed	`function getCollectionOwner(...)`	2
5	reviewed	`    abi.encodeWithSignature(...)`	
6	reviewed	`    return abi.decode(...)`	
7	reviewed	`function getSharesKey(...)`	
8	reviewed	`function equals(...)`	
9	reviewed	`function isZero(...)`	
10	reviewed	`function isCallable(...)`	

5.1.1.12. bonding_curves/RewardSystem.sol

1	reviewed	`bytes32 public root`	
2	reviewed	`mapping(address => uint256) public claimedReward`	
3	reviewed	`bool public rewardSystemType`	
4	reviewed	`ERC20 public erc20RewardToken`	
5	reviewed	`uint256 totalClaimedReward`	
6	reviewed	`uint32 public constant ROLE_DATA_ROOT_MANAGER`	
7	reviewed	`uint32 public constant FEATURE_CLAIM_ACTIVE`	
8	reviewed	`event RootChanged(...)`	
9	reviewed	`event EthRewardClaimed(...)`	
10	reviewed	`event ERC20RewardClaimed(...)`	
11	reviewed	`function postConstruct(...)`	
12	reviewed	`receive(...)`	3
13	reviewed	`fallback(...)`	3
14	reviewed	`function claimReward(...)`	4
15	reviewed	`function setInputDataRoot(...)`	5
16	reviewed	`function isClaimValid(...)`	

5.1.1.13. bonding_curves/README.md

5.1.1.14. bonding_curves/HoldersRewardsDistributorV1.sol

1	reviewed	`struct UserInfo`	
2	reviewed	`address private /*immutable*/ paymentToken`	
3	reviewed	`address public sharesContractAddress`	
4	reviewed	`uint256 public lastRewardBlock`	
5	reviewed	`uint256 public accRewardPerShare`	
6	reviewed	`uint256 public totalShares`	
7	reviewed	`mapping(address => UserInfo) public userInfo`	
8	reviewed	`constructor(...)`	
9	reviewed	`function postConstruct(...)`	
10	reviewed	`function initializeSharesContractAddressIfRequired(...)`	
11	reviewed	`function getPaymentToken(...)`	
12	reviewed	`function __sharesBought(...)`	
13	reviewed	`      userDetail.unclaimedAmount +`	
14	reviewed	`function __sharesSold(...)`	
15	reviewed	`function __accept(...)`	6
16	reviewed	`function claimTheReward(...)`	
17	reviewed	`function pendingReward(...)`	
18	reviewed	`function onTransferReceived(...)`	
19	reviewed	`receive(...)`	
20	reviewed	`fallback(...)`	7
21	reviewed	`function __parseTrade(...)`	

5.1.1.15. bonding_curves/BondingCurve.sol

1	reviewed	function getPrice(...)

5.1.1.16. bonding_curves/SharesFactoryV1.sol

1	reviewed	ERC1363 private /* immutable */ paymentToken
2	reviewed	address private protocolFeeDestination
3	reviewed	uint64 private protocolFeePercent
4	reviewed	uint64 private holdersFeePercent
5	reviewed	uint64 private subjectFeePercent
6	reviewed	address private sharesOwnerAddress
7	reviewed	mapping(bytes32 => TradeableShares) private shares
8	reviewed	mapping(address => TradeableShares.SharesSubject) private subjects
9	reviewed	mapping(ImplementationType => address) private sharesImplementations
10	reviewed	mapping(ImplementationType => address) private distributorsImplementations
11	reviewed	mapping(address => uint256) private nonces
12	reviewed	uint32 public constant FEATURE_SHARES_DEPLOYMENT_ENABLED
13	reviewed	uint32 public constant FEATURE_ALLOW_PAUSED_DEPLOYMENTS
14	reviewed	uint32 public constant FEATURE_ALLOW_EXCLUSIVE_BUY
15	reviewed	uint32 public constant ROLE_PROTOCOL_FEE_MANAGER
16	reviewed	uint32 public constant ROLE_HOLDERS_FEE_MANAGER
17	reviewed	uint32 public constant ROLE_SUBJECT_FEE_MANAGER
18	reviewed	uint32 public constant ROLE_SHARES_REGISTRAR
19	reviewed	uint32 public constant ROLE_FACTORY_DEPLOYMENT_MANAGER
20	reviewed	event SharesOwnerAddressUpdated(...)
21	reviewed	event SharesImplAddressUpdated(...)
22	reviewed	event DistributorImplAddressUpdated(...)
23	reviewed	function postConstruct(...)
24	reviewed	function getPaymentToken(...)
25	reviewed	function getSharesOwnerAddress(...)
26	reviewed	function setSharesOwnerAddress(...)
27	reviewed	function getSharesImplAddress(...)
28	reviewed	function setSharesImplAddress(...)
29	reviewed	function getDistributorImplAddress(...)
30	reviewed	function setDistributorImplAddress(...)
31	reviewed	function getProtocolFeeDestination(...)
32	reviewed	function getProtocolFeePercent(...)
33	reviewed	function getHoldersFeePercent(...)
34	reviewed	function getSubjectFeePercent(...)
35	reviewed	function setProtocolFeeDestination(...)
36	reviewed	function setProtocolFeePercent(...)
37	reviewed	function setHoldersFeePercent(...)
38	reviewed	function setSubjectFeePercent(...)

39	reviewed	function setProtocolFee(...)	8
40	reviewed	function deploySharesContractPaused(...)	
41	reviewed	function deploySharesContract(...)	
42	reviewed	function deploySharesContractAndBuy(...)	
43	reviewed	function mintSubjectAndDeployShares(...)	
44	reviewed	function __mintSubjectAndDeployShares(...)	
45	reviewed	sharesOwnerAddress,	
46	reviewed	address(...)	
47	reviewed	_implementationType	
48	reviewed	function __initSharesContract(...)	
49	reviewed	sharesOwnerAddress,	
50	reviewed	_sharesSubject,	
51	reviewed	protocolFeeDestination,	
52	reviewed	protocolFeePercent,	
53	reviewed	_distributorContract,	
54	reviewed	_holdersFeePercent,	
55	reviewed	subjectFeePercent,	
56	reviewed	_amount,	
57	reviewed	_beneficiary	
58	reviewed	uint256 toPay	
59	reviewed	require(...)	
60	reviewed	require(...)	
61	reviewed	sharesOwnerAddress,	
62	reviewed	_sharesSubject,	
63	reviewed	protocolFeeDestination,	
64	reviewed	protocolFeePercent,	
65	reviewed	_distributorContract,	
66	reviewed	_holdersFeePercent,	
67	reviewed	subjectFeePercent,	
68	reviewed	_amount,	
69	reviewed	_beneficiary,	
70	reviewed	paymentToken	
71	reviewed	require(...)	
72	reviewed	function executeDeploymentRequest(...)	
73	reviewed	function __useNonce(...)	
74	reviewed	function getNonce(...)	
75	reviewed	function rewindNonce(...)	
76	reviewed	function lookupSharesContract(...)	
77	reviewed	function registerSharesContract(...)	
78	reviewed	function notifySubjectUpdated(...)	
79	reviewed	function __registerSharesContract(...)	
80	reviewed	function determineImplementationType(...)	