HashEye

// SMART CONTRACT SECURITY AUDIT

Offchain Custom Fee Token

Security assessment by HashEye · prepared for Offchain Labs

HASHEYE AUDITED

PROJECT	Offchain Custom Fee Token
CLIENT	Offchain Labs
CATEGORY	Offchain Labs
PUBLISHED	September 1, 2023
REPORT ID	research-offchain-custom-fee-token-2023-09-01-1sajk9

This report was produced under HashEye's layered review process — **automated detection**, **pattern correlation**, and **senior manual verification** — with every finding signed off by a human reviewer. Full findings detail and on-chain attestation are available on the report page at hasheye.io/audits/research-offchain-custom-fee-token-2023-09-01-1sajk9.

CustomFeeToken SecurityAssessment(SummaryReport) August1,2024 Preparedfor: HarryKalodner,StevenGoldfeder,andEdFelten OffchainLabs Preparedby:GustavoGrieco,TroySargent,andKurtWillis

Abouthasheye Foundedin2012andheadquarteredinNewYork,hasheyeprovidestechnicalsecurity assessmentandadvisoryservicestosomeoftheworld'smosttargetedorganizations.We combinehigh-endsecurityresearchwithareal -worldattackermentalitytoreduceriskand fortifycode.With100+employeesaroundtheglobe,we'vehelpedsecurecriticalsoftware elementsthatsupportbillionsofendusers,includingKubernetesandtheLinuxkernel. Wemaintainanexhaustivelistofpublicationsathttps://github.com/hasheye-io/publications, withlinkstopapers,presentations,publicauditreports,andpodcastappearances. Inrecentyears,hasheyeconsultantshaveshowcasedcutting-edgeresearchthrough presentationsatCanSecWest,HCSS,Devcon,EmpireHacking,GrrCon,LangSec,NorthSec, theO'ReillySecurityConference,PyCon,REcon,SecurityBSides,andSummerCon. Wespecializeinsoftwaretestingandcodereviewprojects,supportingclientorganizations inthetechnology,defense,andfinanceindustries,aswellasgovernmententities.Notable clientsincludeHashiCorp,Google,Microsoft,WesternDigital,andZoom. hasheyealsooperatesacenterofexcellencewithregardtoblockchainsecurity.Notable projectsincludeauditsofAlgorand,BitcoinSV,Chainlink,Compound,Ethereum2.0, MakerDAO,Matic,Uniswap,Web3,andZcash. Tokeepuptodatewithourlatestnewsandannouncements,pleasefollowhasheyeon Twitterandexploreourpublicrepositoriesathttps://github.com/hasheye-io.Toengageus directly,visitour"Contact"pageathttps://www.hasheye.io/contact,oremailusat info@hasheye.io. hasheye,Inc. 497CarrollSt.,Space71,SeventhFloor Brooklyn,NY11215 https://www.hasheye.io info@hasheye.io hasheye 1OffchainLabsSecurityAssessment PUBLIC

NoticesandRemarks CopyrightandDistribution ©2024byhasheye,Inc. Allrightsreserved.hasheyeherebyassertsitsrighttobeidentifiedasthecreatorofthis reportintheUnitedKingdom. Thisreportisconsideredbyhasheyetobepublicinformation;itislicensedtoOffchain Labsunderthetermsoftheprojectstatementofworkandhasbeenmadepublicat OffchainLabs'srequest.Materialwithinthisreportmaynotbereproducedordistributedin partorinwholewithouttheexpresswrittenpermissionofhasheye. ThesolecanonicalsourceforhasheyepublicationsisthehasheyePublicationspage. Reportsaccessedthroughanysourceotherthanthatpagemayhavebeenmodifiedand shouldnotbeconsideredauthentic. TestCoverageDisclaimer Allactivitiesundertakenbyhasheyeinassociationwiththisprojectwereperformedin accordancewithastatementofworkandagreeduponprojectplan. Securityassessmentprojectsaretime-boxedandoftenreliantoninformationthatmaybe providedbyaclient,itsaffiliates,oritspartners.Asaresult,thefindingsdocumentedin thisreportshouldnotbeconsideredacomprehensivelistofsecurityissues,flaws,or defectsinthetargetsystemorcodebase. hasheyeusesautomatedtestingtechniquestorapidlytestthecontrolsandsecurity propertiesofsoftware.Thesetechniquesaugmentourmanualsecurityreviewwork,but eachhasitslimitations:forexample,atoolmaynotgeneratearandomedgecasethat violatesapropertyormaynotfullycompleteitsanalysisduringtheallottedtime.Theiruse isalsolimitedbythetimeandresourceconstraintsofaproject. hasheye 2OffchainLabsSecurityAssessment PUBLIC

 hasheye 3OffchainLabsSecurityAssessment PUBLIC

ProjectSummary ContactInformation Thefollowingprojectmanagerwasassociatedwiththisproject: MaryO'Brien,ProjectManager mary.obrien@hasheye.io Thefollowingengineeringdirectorwasassociatedwiththisproject: JosselinFeist,EngineeringDirector,Blockchain josselin.feist@hasheye.io Thefollowingconsultantswereassociatedwiththisproject: GustavoGrieco,ConsultantTroySargent,Consultant gustavo.grieco@hasheye.iotroy.sargent@hasheye.io

KurtWillis,Consultant kurt.willis@hasheye.io ProjectTimeline
Thesignificanteventsandmilestonesoftheprojectarelistedbelow. DateEvent September25,2023Pre-
projectkickoffcall October2,2023Deliveryofreportdraft October2,2023Reportreadoutmeeting
August1,2024Deliveryofsummaryreport hasheye 4OffchainLabsSecurityAssessment PUBLIC

ProjectTargets Theengagementinvolvedareviewandtestingofthetargetslistedbelow. nitro-contractsPR#19
Repositoryhttps://github.com/OffchainLabs/nitro-contracts/pull/19 Version PR#19(f23c15c...7dc1aa4)
TypeSolidity PlatformEVM token-bridge-contractsPR#33
Repositoryhttps://github.com/OffchainLabs/token-bridge-contracts/pull/33 Version
PR#33(8cb573a...6396a17) TypeSolidity PlatformEVM token-bridge-contractsPR#34
Repositoryhttps://github.com/OffchainLabs/token-bridge-contracts/pull/34 Version
PR#34(32d00e7...9503d3c) TypeSolidity PlatformEVM hasheye 5OffchainLabsSecurityAssessment PUBLIC

ExecutiveSummary EngagementOverview
OffchainLabsengagedhasheyetoreviewthesecurityoftheCustomFeeToken
implementedinthePRsdetailedintheProjectTargetssection.Thesechangesallownew
rollupstobedeployedusingaspecificERC20implementationthatwillbeusedtopay transactionfees.
AteamofthreeconsultantsconductedthereviewfromSeptember21,2023toSeptember
29,2023,foratotalofthreeengineer-weeksofeffort.Withfullaccesstosourcecodeand
documentation,weperformedamanualreviewofthecodebase. ObservationsandImpact
OffchainLabsaddedafeaturetoallowrollupownerstoselectaspecificERC20tokenon
theparentchainthatwillbeusedtopayforthetransactionfees,completelyreplacingthe
useofETHinthechildchain.Thisnewfeaturerequireschangesinthetokenbridgeand
ArbOS,includingnewcodetocorrectlydeploythetokenbridgeinthechain.
WefocusedonthechangesineachPR,butwehavenotperformedafullreviewofthe
repositoriesinvolved.Wealsoworkedundertheassumptionthatrollupownerswill
carefullyreviewtheavailabledocumentationbeforedeployment,inordertoavoidknown
issueswithcertaintypesoftokens(e.g.,rebasingtokens). Thisreviewuncoveredtwohigh-
severityissuesrelatedtotheassumptionsabouthow ERC20shouldbehave(TOB-ARB-CFT-
001)andhowfundscanflowintothetokenbridge contracts(TOB-ARB-CFT-002). Recommendations
WerecommendthatOffchainLabsfixthereportedissuesandensurethatthetokenbridge
documentationisuptodatebeforedeploymenttoensurethatrollupownersusesuitable ERC20asfeetokens.
hasheye 6OffchainLabsSecurityAssessment PUBLIC

SummaryofFindings Thetablebelowsummarizesthefindingsofthereview,includingtypeandseveritydetails.
IDTitleTypeSeverity 1DoubleentrypointorDeFiintegratedERC20 tokensshouldnotbeused AccessControlsHigh
2TokenbridgewillreceiveandlocketherUndefined Behavior High 3Cross-chainmessageout-of-orderexecution
couldaffectcorrecttokenbridgedeployment Undefined Behavior Medium hasheye
7OffchainLabsSecurityAssessment PUBLIC

DetailedFindings 1.DoubleentrypointorDeFiintegratedERC20tokensshouldnotbeused
Severity:HighDifficulty:High Type:AccessControlsFindingID:TOB-ARB-CFT-001 Target:
src/bridge/ERC20Bridge.sol Description
TheuseofERC20tokenswithtwoormoreentrypointscanallowanattacktodrainthe bridge.
TheuseofERC20tokensforpayingfeesintherollup/bridgerequiresanumberofchecks
andrestrictionstoavoidlossoffunds.Oneofthesechecksisimplementedinthebridge whenawithdrawisexecuted:
function_executeLowLevelCall(addressto, uint256value, bytesmemorydata
)internaloverridereturns(boolsuccess,bytesmemoryreturnData){
//wedon'tallowoutgoingcallstonativetokencontractbecauseitcould
//resultinlossofnativetokenswhichareescrowedbyERC20Bridge if(to==nativeToken){
gvladikamarkedthisconversationasresolved. revertCallTargetNotAllowed(nativeToken); }
//firstreleasenativetoken IERC20(nativeToken).safeTransfer(to,value); success=true; ...
Figure1.1:Headerofthe _executeLowLevelCall functionin src/bridge/ERC20Bridge.sol
Usersarenotallowedtodirectlycallthenativetokenaddress;otherwise,theycould
transferfundsout.However,thischeckwillnotbesufficientifthetokenhasmorethanone
entrypoint(e.g.,whentwodifferentaddressescanbeusedtoexecuteERC20operations, suchastransfer).
hasheye 8OffchainLabsSecurityAssessment PUBLIC

AnotherproblematictypeofERC20istightlyintegratedinDeFiapplications.Forinstance,
theLUSDERC20tokencontainsthefollowingfunction:
functionburn(address_account,uint256_amount)externaloverride{ _requireCallerIsBOorTroveMorSP();
_burn(_account,_amount); } Figure1.2:BurnfunctionfromtheLUSDtoken
Thistokencanbemintedorburnedthroughamanagercontract(whichisdifferentfrom
thetokencontractitself),therebybypassingtheabovecheck.Inparticular,thisDeFiallows
LUSDtokenownerstoopen,close,orrepayvaults,soallofthebridgeERC20LUSDcould
beeasilymanipulatedusingthelow-levelcallbackwithoutrequiringallowancestobeset up. ExploitScenario
Ausercreatesarollupthatusesadoubleentrypointtokensforfees,allowinganyuserto drainthebridgecontract.

Recommendations Shortterm,clearlydocumentthislimitationtomakesureofthispotentialsecurityissue. Longterm,reviewtheassumptionsrequiredbyERC20tokensinordertobeintegratedin eachcomponent. References • Medium-severitybuginBalancerLabs hasheye 9OffchainLabsSecurityAssessment PUBLIC

2.Tokenbridgewillreceiveandlockether Severity:HighDifficulty:Medium Type:UndefinedBehaviorFindingID:TOB-ARB-CFT-002 Target: tokenbridge/ethereum/gateway/L1OrbitERC20Gateway.sol Description Thetokenbridge'sentrypointfordepositscanreceiveether,butthetokenbridgecannot retrieveitinanyway. UsersdepositERC20tokensusingthe outboundTransfer* functionsfromthetoken bridge.Anexampleisshownbelow: functionoutboundTransferCustomRefund(address_l1Token, address_refundTo, address_to, uint256_amount, uint256_maxGas, uint256_gasPriceBid, bytescalldata_data)publicpayableoverridereturns(bytesmemoryres){ ... Figure1.2:Headerofthe outboundTransferCustomRefund functionin src/tokenbridge/ethereum/gateway/L1OrbitERC20Gateway.sol Thisfunctionwilltriggerthecreationofaretryableticket,soitneedsfundstopayfeesand gas.ThesefeescanbepaidusingetherorsomespecificERC20,butindifferenttoken bridgedeploymentsthatsharethesameinterface.Inthelattercase,theentryfunction shouldnotreceiveethereventhoughitispayable. ExploitScenario Auseraccidentallyprovidesethertoatokenbridgeassociatedwitharollupthatusesa customERC20tokenfee.Theetherwillbelockedinthetokenbridge. Recommendations Shortterm,addaconditionthatchecksthevalueprovidedintothe outboundTransfer function,andhavethefunctionrevertifthevalueispositive. Longterm,reviewhowfundsflowfromtheuserto/fromdifferentcomponents,andensure thattherearenosituationswheretokenscanbetrapped. hasheye 10OffchainLabsSecurityAssessment PUBLIC

3.Cross-chainmessageout-of-orderexecutioncouldaectcorrecttoken bridgedeployment Severity:MediumDifficulty:High Type:UndefinedBehaviorFindingID:TOB-ARB-CFT-002 Target: tokenbridge/ethereum/L1AtomicTokenBridgeCreator.sol Description Out-of-orderexecutionofoutboxtransactionsonL1andretryableticketsonL2canleadto unexpectedresultswhenatokenbridgeiscreated.Thisissuereliesonthespecificordering ofretryabletickets. Thetokenbridgecreationrequiresretryableticketstobesubmittedandexecutedina certainorder: /** *@noticeDeployandinitializetokenbridge,bothL1andL2sides,aspartof asingleTX. *@devThisisasingleentrypointofL1tokenbridgecreator.Functiondeploys L1sideoftokenbridgeandthenuses *2retryableticketstodeployL2side.1stretryabledeploysL2factory. Andthen'retryablesender'contract *iscalledtoissue2ndretryablewhichdeploysandinitstherestofthe contracts.L2chainisdetermined *by`inbox`parameter. * *Tokenbridgecanbedeployedonlyonceforcertaininbox.Anyfurther callsto`createTokenBridge`willrevert *becauseL1saltsarealreadyusedatthatpointandL1contractsare alreadydeployedatcanonicaladdresses *forthatinbox. */ functioncreateTokenBridge(addressinbox, addressrollupOwner, uint256maxGasForContracts, uint256gasPriceBid)externalpayable{ ... Figure3.1:Headerofthe createTokenBridge functionin L1AtomicTokenBridgeCreator.sol hasheye 11OffchainLabsSecurityAssessment PUBLIC

However,amalicioususercanleveragetheout-of-orderexecutionofretryableticketsto breaktheassumptionsofthetokenbridgecreatorandproduceafaileddeployment. ExploitScenario Alicestartsthedeploymentofacanonicaltokenbridgeforanewrollup.Evenoticesthis deploymentandspamstherollupbridgewithtransactionstoincreasetheL2gascost,and theticketsarenotauto-redeemed.Later,Evecantriggertheticketsoutofordertoproduce abrokendeployment.Alicewillnotbeabletoredeploy,andnocanonicaldeploymentof thetokenbridgecanbeused. Recommendations Shortterm,considermigratingpartofthedeploymentstepstoL2andrequireasingle retryabletickettobeexecuted. Longterm,reviewallpossiblewaysinwhichtheout-of-orderexecutionofretryabletickets mayaffecteachcomponentanddocument. hasheye 12OffchainLabsSecurityAssessment PUBLIC

A.VulnerabilityCategories Thefollowingtablesdescribethevulnerabilitycategories,severitylevels,anddifficulty levelsusedinthisdocument. VulnerabilityCategories CategoryDescription AccessControlsInsufficientauthorizationorassessmentofrights AuditingandLoggingInsufficientauditingofactionsorloggingofproblems AuthenticationImproperidentificationofusers ConfigurationMisconfiguredservers,devices,orsoftwarecomponents CryptographyAbreachofsystemconfidentialityorintegrity DataExposureExposureofsensitiveinformation DataValidationImproperrelianceonthestructureorvaluesofdata DenialofServiceAsystemfailurewithanavailabilityimpact ErrorReportingInsecureorinsufficientreportingoferrorconditions PatchingUseofanoutdatedsoftwarepackageorlibrary SessionManagementImproperidentificationofauthenticatedusers TestingInsufficienttestmethodologyortestcoverage TimingRaceconditionsorotherorder-of-

operationsflaws UndefinedBehaviorUndefinedbehaviortriggeredwithinthesystem hasheye 13OffchainLabsSecurityAssessment PUBLIC

SeverityLevels SeverityDescription InformationalTheissuedoesnotposeanimmediateriskbutisrelevanttosecuritybest practices. UndeterminedTheextentoftheriskwasnotdeterminedduringthisengagement. LowTheriskissmallorisnotonetheclienthasindicatedisimportant. MediumUserinformationisatrisk;exploitationcouldposereputational,legal,or moderatefinancialrisks. HighTheflawcouldaffectnumeroususersandhaveseriousreputational,legal, orfinancialimplications. DifficultyLevels DifficultyDescription UndeterminedThedifficultyofexploitationwasnotdeterminedduringthisengagement. LowTheflawiswellknown;publictoolsforitsexploitationexistorcanbe scripted. MediumAnattackermustwriteanexploitorwillneedin-depthknowledgeofthe system. HighAnattackermusthaveprivilegedaccesstothesystem,mayneedtoknow complextechnicaldetails,ormustdiscoverotherweaknessestoexploitthis issue. hasheye 14OffchainLabsSecurityAssessment PUBLIC

B.DevelopmentPractices Inthissection,weprovidebestpracticesregardingcodecomplexitymanagement. • Whendesigningsmartcontractswiththepurposeofreuse,trytominimize inheritancewheneverpossible,asitoftencanleadtothecreationofmultiplelevels ofindirectionandmaketheexecutionflowveryhardtofollow.Whilesomeamount ofinheritanceisexpected,itcaneasilybeabused. Anexampleofacomplexinheritancestructurecanbeseenwhentracingthe internalfunctioncallswhencreatingdepositsto L1OrbitERC20Gateway : • L1GatewayRouter.outboundTransferCustomRefund • super(L1ERC20Gateway).outboundTransferCustomRefund • super(L1ArbitrumGateway).outboundTransferCustomRefund • L1ArbitrumGateway._parseUserEncodedData (overloadedin L1OrbitERC20Gateway) • L1ArbitrumGateway.calculateL2TokenAddress (overloadedin L1ERC20Gateway) • L1ArbitrumGateway.getOutboundCalldata (overloadedin L1ERC20Gateway) • L1ArbitrumGateway._initiateDeposit (overloadedin L1OrbitERC20Gateway) • L1ArbitrumMessenger.sendTxToL2CustomRefund • L1ArbitrumMessenger._createRetryable (overloadedin L1OrbitERC20Gateway) AsshowninfigureB.1,thecallsjumpbackandforthbetweenfourcontractsusing the super keywordandfunctionoverloading.Abstractionisusefulforseparating concernsandreducingcodeduplication;however,itshouldnotbeoverused.Too muchabstractioncanmakefollowingexecutiontracesdifficultandintroduce significantmentaloverhead. hasheye 15OffchainLabsSecurityAssessment PUBLIC

FigureB.1:ArbitrumGateway’scomplexinheritancestructure hasheye 16OffchainLabsSecurityAssessment PUBLIC

• Whenoverloadingfunctions,aimtoplaceoptionalparametersattheendifpossible. ThismakesitclearerwhatparametersareoptionalandreducescognitiveloadBy notshiftingpositionsofotherparameterstoomuch. // ... functionregisterTokenToL2(address_l2Address, uint256_maxGas, uint256_gasPriceBid, uint256_maxSubmissionCost, uint256_feeAmount)externalreturns(uint256){ return registerTokenToL2(_l2Address, _maxGas, _gasPriceBid, _maxSubmissionCost, msg.sender, _feeAmount); } // ... functionregisterTokenToL2(address_l2Address, uint256_maxGas, uint256_gasPriceBid, uint256_maxSubmissionCost, address_creditBackAddress, uint256_feeAmount)publicreturns(uint256){ return _registerTokenToL2(_l2Address, _maxGas, _gasPriceBid, _maxSubmissionCost, _creditBackAddress, _feeAmount); } FigureB.2:Overloading registerTokenToL2 in L1OrbitCustomGateway • Aimtokeepthesameorderwhenpassingonfunctionparameters. functioncreateOutboundTxCustomRefund(address_refundTo, address_from, uint256,/*_tokenAmount*/ hasheye 17OffchainLabsSecurityAssessment PUBLIC

uint256_maxGas, uint256_gasPriceBid, uint256_maxSubmissionCost, bytesmemory_outboundCalldata)internalvirtualreturns(uint256){ //WemakethisfunctionvirtualsinceoutboundTransferlogicisthesamefor manygateways //butsometimes(ieweth)youconstructtheoutgoingmessagedifferently. //msg.valueissent,but0issettotheL2callvalue //theethsentisusedtopayforthetx'sgas return sendTxToL2CustomRefund(inbox, counterpartGateway, _refundTo, _from, msg.value,//weforwardtheL1callvaluetotheinbox 0,//l2callvalue0bydefault L2GasParams({ _maxSubmissionCost:_maxSubmissionCost, _maxGas:_maxGas, _gasPriceBid:_gasPriceBid }), _outboundCalldata); } FigureB.3: L2GasParams arenotinorderwithfunctionparameters.(L1ArbitrumGateway) • Aimtobeconsistentwithfunctionvariablenameswhenpossible,ordeclare temporaryvariablesandaddcommentsexplainingvariablenameswitching. function_initiateDeposit(address_refundTo, address_from, uint256,//_amount,thisinfoisalreadycontainedin_data uint256_maxGas, uint256_gasPriceBid, uint256_maxSubmissionCost, uint256tokenTotalFeeAmount, bytesmemory_data)internaloverridereturns(uint256){ return sendTxToL2CustomRefund(inbox, counterpartGateway, _refundTo, _from, tokenTotalFeeAmount, 0, L2GasParams({ hasheye 18OffchainLabsSecurityAssessment PUBLIC

_maxSubmissionCost:_maxSubmissionCost, _maxGas:_maxGas, _gasPriceBid:_gasPriceBid }), _data); }
FigureB.4: tokenTotalFeeAmount ismappedto _l1CallValue .(L1OrbitERC20Gateway)
functionsendTxToL2CustomRefund(address_inbox, address_to, address_refundTo, address_user,
uint256_l1CallValue, uint256_l2CallValue, L2GasParamsmemory_l2GasParams, bytesmemory_data
)internalreturns(uint256){ //... } FigureB.5: tokenTotalFeeAmount ismappedto _l1CallValue .(
L1ArbitrumMessenger) • Usenamedparametersortemporarilydeclarenamedvariableswhenpassingin
unnamedconstantsasfunctionparameters. function_initiateDeposit(address_refundTo, address_from,
uint256,//_amount,thisinfoisalreadycontainedin_data uint256_maxGas, uint256_gasPriceBid,
uint256_maxSubmissionCost, uint256tokenTotalFeeAmount, bytesmemory_data
)internaloverridereturns(uint256){ return sendTxToL2CustomRefund(inbox, counterpartGateway,
_refundTo, _from, tokenTotalFeeAmount, 0, L2GasParams({ _maxSubmissionCost:_maxSubmissionCost,
_maxGas:_maxGas, _gasPriceBid:_gasPriceBid }), hasheye 19OffchainLabsSecurityAssessment PUBLIC

_data); } FigureB.6: tokenTotalFeeAmount ismappedto _l1CallValue .(L1OrbitERC20Gateway)
function_initiateDeposit(address_refundTo, address_from,
uint256,//_amount,thisinfoisalreadycontainedin_data uint256_maxGas, uint256_gasPriceBid,
uint256_maxSubmissionCost, uint256tokenTotalFeeAmount, bytesmemory_data
)internaloverridereturns(uint256){ //The`_l2CallValue`issetto`0`whenbridgingERC20tokens.
uint256_l2CallValue=0; return sendTxToL2CustomRefund(inbox, counterpartGateway, _refundTo, _from,
tokenTotalFeeAmount, _l2CallValue, L2GasParams({ _maxSubmissionCost:_maxSubmissionCost,
_maxGas:_maxGas, _gasPriceBid:_gasPriceBid }), _data); } FigureB.7: tokenTotalFeeAmount ismappedto
_l1CallValue .(L1OrbitERC20Gateway) •
Beconsistentwithafunctionnamingconventionofprependinganunderscore“_”
forinternalfunctions.Thiscanhelpdetectwhichfunctionsareimportantforaccess controlchecks.
functioninboundEscrowTransfer(address_l1Token, address_dest, uint256_amount)internalvirtual{
//thismethodisvirtualsincedifferentsubclassescanhandleescrow differently
IERC20(_l1Token).safeTransfer(_dest,_amount); } hasheye 20OffchainLabsSecurityAssessment PUBLIC

/** *@devOnlyexcessgasisrefundedtothe_refundToaccount,l2callvalueis alwaysreturnedtothe_toaccount
*/ functioncreateOutboundTxCustomRefund(address_refundTo, address_from, uint256,/*_tokenAmount*/
uint256_maxGas, uint256_gasPriceBid, uint256_maxSubmissionCost, bytesmemory_outboundCalldata
)internalvirtualreturns(uint256){ FigureB.8: inboundEscrowTransfer and createOutboundTxCustomRefund
donotfollow theconventionseenforinternalfunctions.(L1ArbitrumGateway) hasheye
21OffchainLabsSecurityAssessment PUBLIC