HashEye

// SMART CONTRACT SECURITY AUDIT

Scroll ZkEVM Wave 2

Security assessment by HashEye · prepared for Scroll

HASHEYE AUDITED

PROJECT	Scroll ZkEVM Wave 2
CLIENT	Scroll
CATEGORY	Ethereum/EVM
PUBLISHED	August 1, 2023
REPORT ID	research-scroll-zkevm-wave-2-2023-08-01-1mjtlu

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-scroll-zkevm-wave-2-2023-08-01-1mjtlu.

ScrollzkEVMCircuits,Wave2 SecurityAssessment September8,2023 Preparedfor: HaichenShen Scroll
Preparedby:FilipeCasal,JoeDoyle,andMarcIlunga

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. 228ParkAveS#80688 NewYork,NY10003 https://www.hasheye.io info@hasheye.io hasheye

NoticesandRemarks CopyrightandDistribution reportintheUnitedKingdom. Thisreportisconsideredbyhasheyetobepublicinformation;itislicensedtoScroll underthetermsoftheprojectstatementofworkandhasbeenmadepublicatScroll's request.Materialwithinthisreportmaynotbereproducedordistributedinpartorin wholewithouttheexpresswrittenpermissionofhasheye. 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

TableofContents
 hasheye

ExecutiveSummary EngagementOverview
ScrollengagedhasheyetoreviewthesecurityofasetofzkEVMcircuits:theMerkle
Patriciatrie(MPT)circuit,theCopycircuit,word-addressablememoryoptimizations,and theSupercircuit.
AteamofthreeconsultantsconductedthereviewfromJuly17toAugust4,2023,foratotal ofsixengineer-
weeksofeffort.Ourtestingeffortsfocusedonensuringthatthecircuitsare
sound,complete,andfaithfulimplementationsofthespecifications.Withfullaccesstothe
sourcecodeanddocumentation,weperformedstaticanddynamictestingofthe
codebase,usingautomatedandmanualprocesses. ObservationsandImpact
Thesecurityreviewofthecodebaseuncoveredthreehigh-severityissuesrelatedtothe
soundnessofthecircuits:findingTOB-SCROLL2-6affectstheMPTcircuitandallows
attackerstobypasschecks whencreatingorupdatingtheMPTnodecontainingtheaccount
nonceandcodesize;findingTOB-SCROLL2-8allowsattackerstohijacktheEVMcontrol flowafterthe CREATE
opcodeiscalled,allowingthemtoredirectasuccessfulEVM executiontoahaltwithanerrorstate;andfindingTOB-
SCROLL2-9affectsthe OneHot
primitiveandmayallowamaliciousprovertoskipconstraintsintheMPTupdatecircuit.
WefoundseveralinconsistenciesbetweentheMPTcircuitspecificationandits implementation(TOB-SCROLL2-
3,TOB-SCROLL2-4,andTOB-SCROLL2-7). Wealsofoundsomeissuesinvolvingunresolved“TODO”commentsandstub
implementationsusedfortesting,includingtheinformationalfindingTOB-SCROLL2-1and thehigh-
severityfindingTOB-SCROLL2-9. Recommendations
Basedonthecodebasematurityevaluationandfindingsidentifiedduringthesecurity
review,hasheyerecommendsthatScrolltakethefollowingsteps: ●
Remediatethefindingsdisclosedinthisreport.Thesefindingsshouldbe
addressedaspartofadirectremediationoraspartofanyrefactorthatmayoccur
whenaddressingotherrecommendations. ●
Revisethecircuitspecifications.Duringtheengagement,wefoundseveral
inconsistenciesbetweenthespecificationsandthecircuitimplementations.The documentationalsolackshigh-
leveldescriptionsofhowthecircuitswork. Higher-
leveldescriptionswouldgreatlyaiddevelopersandauditorsinquickly
understandingtheinnerworkingsofthecircuitsandallowthemtomoreeasily hasheye
4ScrollzkEVMCircuitsSecurityAssessment PUBLIC

identifypotentialstructuralflaws.Afterthespecificationsarerevisedandclarified,
ensurethatthecircuitimplementationsarefaithfultotheupdatedspecifications. ●
Implementbranch/leafdomainseparationintheMerkletreedesign.The
MerkletreedesignimplementedintheMPTcircuitdoesnotcorrectlyinclude
domainseparationbetweenbranchandleafnodes,whichcouldenableMerkle
proofforgery.Wedidnotincludethisflawasafindinginthisreportbecausewe
founditduringourseparateauditofthe zktrie library,butthefixwillneedtobe implementedinboth zktrie
andtheMPTcircuit. ● Addressthe“TODO”commentsinthecodeandtestwithnon-stub
implementations.Althoughtheyareusefulforrapiddevelopmentandtesting,
“TODO”commentsandtestingwithstubimplementationsallowflawstoremainin
thecodebase.Each“TODO”shouldbetrackedandtriagedinacentralizedissue
tracker(e.g.,GitHubissues),notincodecomments,andshouldbefixedifneeded;
eachmockimplementationshouldbeassociatedwithafullimplementationthatcan
beusedinoccasionalfulltestsuiteruns. ●
ConsiderrefactoringtheMPTupdateverificationcodetoseparateMerkle path-checkinglogicfromproof-
specificlogic.Thecurrentcircuitsimultaneously checksMerklepathsandchecksvariousproof-
specificattributesoftheleaves.This
designmakesthecircuitfairlycomplex,withtwosimultaneousstatemachines
modeledwithinthetableandmanyrepeatedchecks,whichwebelievehas contributedtocopy-
pasteproblemssuchasfindingTOB-SCROLL2-6.Separatingthe
Merklepathchecksfromthespecificchecksusedforspecificupdatetypeswould
improvemodularityandmaymakeanyfuturerequiredmodificationseasiertodo.
Thefollowingtablesprovidethenumberoffindingsbyseverityandcategory. EXPOSUREANALYSIS SeverityCount
High3 Medium0 Low0 Informational7 Undetermined0 CATEGORYBREAKDOWN CategoryCount Cryptography9
Testing1 hasheye 5ScrollzkEVMCircuitsSecurityAssessment PUBLIC

ProjectSummary ContactInformation Thefollowingmanagerswereassociatedwiththisproject:
DanGuido,AccountManagerBrookeLanghorne,ProjectManager dan@hasheye.iobrooke.langhorne@hasheye.io
Thefollowingengineerswereassociatedwiththisproject: FilipeCasal,ConsultantJoeDoyle,Consultant
filipe.casal@hasheye.iojoseph.doyle@hasheye.io MarcIlunga,Consultant marc.ilunga@hasheye.io
ProjectTimeline Thesignificanteventsandmilestonesoftheprojectarelistedbelow. DateEvent
July13,2023Pre-projectkickoffcall July21,2023Statusupdatemeeting#1 July28,2023Statusupdatemeeting#2
August7,2023Deliveryofreportdraft August7,2023Reportreadoutmeeting
September8,2023Deliveryofcomprehensivereport

October10,2023Deliveryofcomprehensivereportwithfixreviewappendix hasheye
6ScrollzkEVMCircuitsSecurityAssessment PUBLIC

ProjectGoals Theengagementwasscopedtoprovideasecurityassessmentofasubsetof Scroll’szkEVM circuits.Specifically,wesoughttoanswerthefollowingnon-exhaustivelistofquestions: • Doesthecodebasefollowbestpracticesfor Rust? • Arethecircuitssoundandcomplete? • AretheMPTcircuitandword-addressablememoryoptimization Rust implementationsfaithfultotheirspecifications? • AretheMPTpath-checkingstatemachinescorrectlydesignedandimplemented? Arethecorrecttransitionsenforced? Isitpossibletoskiprequiredchecksby manipulatingthestatemachines? • Aretheword-addressableoptimizationssound?Hastheassociatedcoderefactoring introducedanyvulnerabilitiesintothecodebase? hasheye 7ScrollzkEVMCircuitsSecurityAssessment PUBLIC

ProjectTargets Theengagementinvolvedareviewandtestingofthefollowingtargets. mpt-circuit Repositoryhttps://github.com/scroll-tech/mpt-circuit/tree/v0.4 Version 7b56d0b323e92ac11e54213520f6e7db41941cd0 TypesRust,halo2 PlatformNative copy-circuit Repositoryhttps://github.com/scroll-tech/zkevm-circuits Version fc6c8a2972870e62e96cde480b3aa48c0cc1303d TypesRust,halo2 PlatformNative super_circuit.rs Repositoryhttps://github.com/scroll-tech/zkevm-circuits Version fc6c8a2972870e62e96cde480b3aa48c0cc1303d TypesRust,halo2 PlatformNative hasheye 8ScrollzkEVMCircuitsSecurityAssessment PUBLIC

ProjectCoverage Thissectionprovidesanoverviewoftheanalysiscoverageofthereview,asdeterminedby ourhigh-levelengagementgoals.Ourapproachesincludedthefollowing: • TheMPTcircuit:Wemanuallyreviewedthecircuitutilitiesandtheupdate-checking circuitin mpt_update.rs .Ifweidentifiedvulnerablecodepatternsorcodesmells, weusedSemgreptoperformvariantanalysisandtosearchforotherinstancesof thesamepatterns.Wecomparedtheimplementationwiththespecificationand reportedtheidentifieddifferences. • TheCopycircuitandword-addressablememoryoptimizations:Wereviewedthe memoryoptimizationspecificationforsoundnessandmanuallyreviewedtheCopy circuitandassociatedgadgetsagainsttheCopycircuitspecificationdocument. Additionally,wereviewedtherefactoringthatneededtobedoneonseveralzkEVM circuitsduetotheintroducedoptimizations. • TheSupercircuit:Wemanuallyreviewedthecircuitforincorrectcircuit initializationissues. hasheye 9ScrollzkEVMCircuitsSecurityAssessment PUBLIC

AutomatedTesting hasheyeusesautomatedtechniquestoextensivelytestthesecurityproperties of software.Weusebothopen-sourcestaticanalysisandfuzzingutilities,alongwithtools developedinhouse,toperformautomatedtestingofsourcecodeandcompiledsoftware. TestHarnessConfiguration Weusedthefollowingtoolsintheautomatedtestingphaseofthisproject: ToolDescriptionPolicy SemgrepAnopen-sourcestaticanalysistoolforfindingbugsand enforcingcodestandardswheneditingorcommittingcode andduringbuildtime AppendixD.1 cargo-llvm -cov AtoolforgeneratingtestcoveragereportsinRustAppendixD.2 cargo-edit AtoolforquicklyidentifyingoutdatedcratesAppendixD.3 ClippyAnopen-sourceRustlinterusedtocatchcommonmistakes andunidiomaticRustcode AppendixD.4 AreasofFocus Ourautomatedtestingandverificationworkfocusedonidentifyingthefollowing: • Generalcodequalityissuesandunidiomaticcodepatterns • Untestedcoderegionswithcoveragereports • VariantsofmanuallyfoundvulnerableAPIpatterns TestResults Theresultsofthisfocusedtestingaredetailedbelow. Clippy MPTcircuit:RunningClippyinpedanticmodeontheMPTcircuitrevealsseveralcode qualityissuesandunidiomaticcodepatterns,includingthefollowing: semicolon_if_nothing_returned , match_same_arms , needless_pass_by_value , hasheye 10ScrollzkEVMCircuitsSecurityAssessment PUBLIC

inconsistent_struct_constructor , return_self_not_must_use ,and map_unwrap_or . Werecommendroutinely(e.g.,everyminorrelease)runningClippyinpedanticmode.If certainpatternsarecommonlyfound,consideraddingtheserulestothedefaultCIClippy runs. Semgrep Therulesthatwewrotetofindvariantsofmanuallyfoundvulnerable APIpatternsare providedinappendixD. cargo-edit Running cargo-edit with cargoupgrade--incompatible--dry-run findsfive outdatedcratesusedbytheMPTcircuitcodebase: ethers-core , itertools , strum , strum_macros ,and rand_chacha . cargo-llvm-cov ThecoveragereportforthemPTcircuitindicatesseveralcoveragelimitations. hasheye 11ScrollzkEVMCircuitsSecurityAssessment PUBLIC

CodebaseMaturityEvaluation hasheyeusesatraffic-lightprotocoltoprovideeachclientwithaclearunderstandingof theareasinwhichitscodebaseismature,immature,orunderdeveloped.Deficiencies identifiedhereoftenstemfromrootcauseswithinthesoftwaredevelopmentlifecyclethat

shouldbeaddressedthroughstandardizationmeasures(e.g.,theuseofcommonlibraries, functions,orframeworks)ortrainingandawarenessprograms. CategorySummaryResult ArithmeticWefoundnoissuesrelatedtotheuseofinteger arithmeticinthecodebase. Satisfactory Complexity Management Thegeneralstructureoftheimplementationiswell organizedandseparatedintoclearmodules. However,thestatemachine-baseddesignoftheMPT circuitisfairlycomplex,andtheMerklepathverification codeistightlycoupledwiththecodethatvalidatesthe datastoredwithinagivenleaf.Twostatemachines governMerklepathchecking,eachofwhosestate transitionsaredifferentdependingontheprooftype. Eachprooftypealsohassomerepetitivebutslightly changedconstraints,suchastheconstraintsenforcing theexactpaththroughanaccountleaf. Moderate Cryptography andKey Management TheMPTdesignimplementedinthiscircuitisvulnerable toproofforgeryattacks,asdisclosedinapreviousaudit ofthe zktrie library.Theimplementationwillneedto beupdatedsothatithasasecuredesignbeforeitis suitablefordeployment. Weak DocumentationBoththeMPTandtheCopycircuit,aswellasthe word-addressablememoryoptimizations,have associatedspecificationdocuments.However,wefound severaldiscrepanciesbetweenthespecificationsand implementations.Werecommendthatthespecifications berevisedandextendedwithhigh-leveldescriptionsof thelogicbehindeachcircuit.Havingtherevised specificationswillallowdeveloperstohaveaclearer understandingofthecircuits. Moderate hasheye

MemorySafety andError Handling ThezkEVMcircuitsprojectusesnounsafeRustcode.Satisfactory Testingand Verification Weidentifiedsometest-coveragelimitationsintheMPT circuitcodebasebyusingthe cargo-llvm-cov tool. Severalofthefindingsthatweidentifiedinthisreview couldhavebeendetectedwithacomprehensivepositive andnegativetestsuite. Becausewefoundhigh-severityfindingsrelatedtocircuit soundness,webelieveitisnecessarytodevelopan adversarialtestingprocess,especiallyfocusedon maliciousproverbehavior.Formalmethodstocheckfor (e.g.,circuitdeterminacy)shouldalsobeinvestigatedand considered. Weak hasheye

SummaryofFindings Thetablebelowsummarizesthefindingsofthereview,includingtypeandseveritydetails. IDTitleTypeSeverity 1PoseidonLookupisnotimplementedTestingInformational 2IsZeroGadgetdoesnotconstraintheinverse witnesswhenthevalueiszero CryptographyInformational 3TheMPTnonexistenceproofgadgetismissing constraintsspecifiedinthedocumentation CryptographyInformational 4DiscrepanciesbetweentheMPTcircuit specificationandimplementation CryptographyInformational 5RedundantlookupsintheWordRLCcircuitCryptographyInformational 6TheNonceChangedconfigurationcircuitdoesnot constrainthenewvaluenoncevalue CryptographyHigh 7TheCopycircuitdoesnottotallyenforcethetag values CryptographyInformational 8The“invalidcreation”errorhandlingcircuitis unconstrained CryptographyHigh 9TheOneHotprimitiveallowsmorethanonevalue atonce CryptographyHigh 10IntermediatecolumnsarenotexplicitCryptographyInformational hasheye

DetailedFindings 1.PoseidonLookupisnotimplemented Severity:InformationalDifficulty:N/A Type:TestingFindingID:TOB-SCROLL2-1 Target: src/gadgets/poseidon.rs Description PoseidonhashingisperformedwithintheMPTcircuitbyperforminglookupsintoa Poseidontableviathe PoseidonLookup trait,showninfigure1.1. ///Lookuprepresentstheposeidontableinzkevmcircuit pubtraitPoseidonLookup{ fnlookup_columns(&self)→(FixedColumn,[AdviceColumn;5]){ let(fixed,adv)=self.lookup_columns_generic(); (FixedColumn(fixed),adv.map(AdviceColumn)) } fnlookup_columns_generic(&self)→(Column<Fixed>,[Column<Advice>;5]){ let(fixed,adv)=self.lookup_columns(); (fixed.0,adv.map(|col|col.0)) } } Figure1.1: src/gadgets/poseidon.rs#11-21 Thistraitisnotimplementedbyanytypesexceptthetesting-only PoseidonTable shown infigure1.2,whichdoesnotconstrainitscolumnsatall. #[cfg(test)] #[derive(Clone,Copy)] pubstructPoseidonTable{ q_enable:FixedColumn, left:AdviceColumn, right:AdviceColumn, hash:AdviceColumn, control:AdviceColumn, head_mark:AdviceColumn, } #[cfg(test)] implPoseidonTable{ pubfnconfigure<F:FieldExt>(cs:&mutConstraintSystem<F>)→Self{ hasheye

let[hash,left,right,control,head_mark]= [0;5].map(|_|AdviceColumn(cs.advice_column())); Self{ left, right, hash, control, head_mark, q_enable:FixedColumn(cs.fixed_column()), } } Figure1.2: src/gadgets/poseidon.rs#56-80 Therestofthecodebasetreatsthistraitasablack-boximplementation,sothisdoesnot seemtocausecorrectnessproblemselsewhere.However,itdoeslimitone’sabilitytotest somenegativecases,anditmakesthetestcoveragerelyonthecorrectnessofthe PoseidonTable struct’switnessgeneration. Recommendations Shortterm,createaconcreteimplementationofthe PoseidonLookup traittoenablefull testingoftheMPTcircuit. Longterm,ensurethatallpartsoftheMPTcircuitaretestedwithbothpositiveand negativetests. hasheye

2.IsZeroGadgetdoesnotconstraintheinversewitnesswhenthevalueiszero
Severity:InformationalDifficulty:N/A Type:CryptographyFindingID:TOB-SCROLL2-2 Target:
src/gadgets/is_zero.rs Description The IsZeroGadget implementationallowsforanarbitrary
inverse_or_zero witness valuewhenthe value parameteris 0 . Thegadgetreturns 1 when value is 0
;otherwise,itreturns 0 .Theimplementationrelies ontheexistenceofaninverseforwhen value
isnonzeroandoncorrectlyconstraining that value*(1-value*inverse_or_zero)==0 . However,when value is
0 ,theconstraintisimmediatelysatisfied,regardlessofthevalue ofthe inverse_or_zero
witness.Thisallowsanarbitraryvaluetobeprovidedforthat witnessvalue. pubfnconfigure<F:FieldExt>(
cs:&mutConstraintSystem<F>, cb:&mutConstraintBuilder<F>,
value:AdviceColumn,//TODO:makethisaqueryonceQueryis clonable/copyable.....)→Self{
letinverse_or_zero=AdviceColumn(cs.advice_column()); cb.assert_zero(
"valueis0orinverse_or_zeroisinverseofvalue", value.current()*(Query::one()-value.current()*
inverse_or_zero.current()),); Self{ value, inverse_or_zero, } } Figure2.1: mpt-
circuit/src/gadgets/is_zero.rs#48-62 Recommendations
Shortterm,ensurethatthecircuitisdeterministicbyconstraining inverse_or_zero to equal 0 when value
is 0 . hasheye 17ScrollzkEVMCircuitsSecurityAssessment PUBLIC

Longterm,documentwhichcircuitshavenondeterministicwitnesses;overtime,constrain
themsothatallcircuitshavedeterministicwitnesses. hasheye 18ScrollzkEVMCircuitsSecurityAssessment
PUBLIC

3.TheMPTnonexistenceproofgadgetismissingconstraintsspecifiedinthe documentation
Severity:InformationalDifficulty:N/A Type:CryptographyFindingID:TOB-SCROLL2-3 Target:
src/gadgets/mpt_update/nonexistence_proof.rs Description
Thegadgetforcheckingtheconsistencyofnonexistenceproofsismissingseveral
constraintsrelatedtotype2nonexistenceproofs.
Thecircuitspecificationincludesconstraintsforthenonexistenceofpathproofsthatare
notincludedintheimplementation.Thiscausesthewitnessvaluestobeunconstrainedin
somecases.Forexample,thefollowingconstraintsarespecified: • other_key_hash shouldequal 0 when key
doesnotequal other_key . • other_leaf_data_hash shouldequalthehashoftheemptynode(pointerby
other_key). Neitheroftheseconstraintsisenforcedintheimplementation:thisisbecausethe
implementationhasnoexplicitconstraintsimposedforthetype2nonexistenceproofs.
Figure3.1showsthatthecircuitconstrainsthesevaluesonlyfortype1proofs. pubfnconfigure<F:FieldExt>(
cb:&mutConstraintBuilder<F>, value:SecondPhaseAdviceColumn, key:AdviceColumn,
other_key:AdviceColumn, key_equals_other_key:IsZeroGadget, hash:AdviceColumn,
hash_is_zero:IsZeroGadget, other_key_hash:AdviceColumn, other_leaf_data_hash:AdviceColumn,
poseidon:&implPoseidonLookup,){ cb.assert_zero("valueis0foremptynode",value.current());
cb.assert_equal("key_minus_other_key=key-otherkey", key_equals_other_key.value.current(),
key.current()-other_key.current(),); cb.assert_equal(hasheye
19ScrollzkEVMCircuitsSecurityAssessment PUBLIC

"hash_is_zeroinput==hash", hash_is_zero.value.current(), hash.current(),);
letis_type_1=!key_equals_other_key.current(); letis_type_2=hash_is_zero.current(); cb.assert_equal(
"Emptyaccountiseithertype1xortype2", Query::one(),
Query::from(is_type_1.clone())+Query::from(is_type_2),); cb.condition(is_type_1,|cb|{
cb.poseidon_lookup("other_key_hash==h(1,other_key)",
[Query::one(),other_key.current(),other_key_hash.current()], poseidon,); cb.poseidon_lookup(
"hash==h(key_hash,other_leaf_data_hash)", [other_key_hash.current(),
other_leaf_data_hash.current(), hash.current(),], poseidon,); }); Figure3.1: mpt-
circuit/src/gadgets/mpt_update/nonexistence_proof.rs#7-54
TheScrollteamhasstatedthatthisisaspecificationerrorandthatthemissingconstraints
donotimpactthesoundnessofthecircuit. Recommendations
Shortterm,updatethespecificationtoremovethedescriptionoftheseconstraints;ensure
thatthedocumentationiskeptupdated.
Longterm,addpositiveandnegativetestsforbothtypesofnonexistenceproofs. hasheye
20ScrollzkEVMCircuitsSecurityAssessment PUBLIC

4.DiscrepanciesbetweentheMPTcircuitspecificationandimplementation
Severity:InformationalDifficulty:N/A Type:CryptographyFindingID:TOB-SCROLL2-4 Target:Severalfiles
Description TheMPTcircuitimplementationisnotfaithfultothecircuitspecificationinmanyareasand
doesnotcontaincommentsfortheconstraintsthatareeithermissingfromthe
implementationorthatdivergefromthoseinthespecification.
Theallowedsegmenttransitionsdependontheprooftype.Forthe NonceChanged proof
type,thespecificationstatesthatthe Start segmenttypecantransitionto Start andthat the AccountLeaf0
segmenttypealsocantransitionto Start .However,neitherofthese pathsisallowedintheimplementation.
MPTProofType::NonceChanged |MPTProofType::BalanceChanged |MPTProofType::CodeSizeExists

|MPTProofType::CodeHashExists⇒[(SegmentType::Start, vec![SegmentType::AccountTrie,//mpthas>1account SegmentType::AccountLeaf0,//mpthas⩽1account],), (SegmentType::AccountTrie, vec![SegmentType::AccountTrie, SegmentType::AccountLeaf0, SegmentType::Start,//emptyaccountproof],), (SegmentType::AccountLeaf0,vec! [SegmentType::AccountLeaf1]), (SegmentType::AccountLeaf1,vec![SegmentType::AccountLeaf2]), (SegmentType::AccountLeaf2,vec![SegmentType::AccountLeaf3]), (SegmentType::AccountLeaf3,vec! [SegmentType::Start]), Figure4.1: mpt-circuit/src/gadgets/mpt_update/segment.rs#20-42 hasheye 21ScrollzkEVMCircuitsSecurityAssessment PUBLIC

Figure4.2:PartoftheMPTspecification(spec/mpt-proof.md#L318-L328) Thetransitionsallowedforthe PoseidonCodeHashExists prooftypealsodonotmatch: thespecificationstatesthatithasthesametransitionsasthe NonceChanged prooftype, buttheimplementationhasdifferenttransitions. Thekeydepthdirectionchecksalsodonotmatchthespecification.Thespecificationstates thatthe depth parametershouldbeusedbuttheimplementationuses depth-1 . cb.condition(is_trie.clone(),|cb|{ cb.add_lookup("directioniscorrectforkeyanddepth", [key.current(),depth.current()-1,direction.current()], key_bit.lookup(),); cb.assert_equal("depthincreasesby1intriesegments", depth.current(), depth.previous()+1,); cb.condition(path_type.current_matches(&[PathType::Common]),|cb|{ cb.add_lookup("directioniscorrectforother_keyanddepth", [other_key.current(), depth.current()-1, Figure4.3: mpt-circuit/src/gadgets/mpt_update.rs#188-205 hasheye 22ScrollzkEVMCircuitsSecurityAssessment PUBLIC

Figure4.4:PartoftheMPTspecification(spec/mpt-proof.md#L279-L282) Finally,thespecificationstatesthatwhenasegmenttypeisanon-trietype,thevalueof key shouldbeconstrainedto 0 ,butthisconstraintisomittedfromtheimplementation. cb.condition(!is_trie,|cb|{ cb.assert_zero("depthis0innon-triesegments",depth.current()); }); Figure4.5: mpt-circuit/src/gadgets/mpt_update.rs#212-214 Figure4.6:PartoftheMPTspecification(spec/mpt-proof.md#L284-L286) Recommendations Shortterm,reviewthespecificationandensureitsconsistency.Matchtheimplementation withthespecification,anddocumentpossibleoptimizationsthatremoveconstraints, detailingwhytheydonotcausesoundnessissues. Longterm,includebothpositiveandnegativetestsforalledgecasesinthespecification. hasheye 23ScrollzkEVMCircuitsSecurityAssessment PUBLIC

5.RedundantlookupsintheWordRLCcircuit Severity:InformationalDifficulty:N/A Type:CryptographyFindingID:TOB-SCROLL2-5 Target: src/gadgets/mpt_update/word_rlc.rs Description TheWordRLCcircuithastworedundantlookupsintothe BytesLookup table. TheWordRLCcircuitcombinestherandomlinearcombination(RLC)forthelowerand upper16bytesofawordintoasingleRLCvalue.Forthis,itchecksthatthelowerand upperwordsegmentsare16bytesbylookingintothe BytesLookup table,anditchecks thattheirRLCsarecorrectlycomputedbylookingintothe RlcLookup table.However,the lookupintothe RlcLookup tablewillalsoensurethattheloweranduppersegmentsofthe wordhavethecorrect16bytes,makingthefirsttwolookupsredundant. pubfnconfigure<F:FieldExt>(cb:&mutConstraintBuilder<F>, [word_hash,high,low]:[AdviceColumn;3], [rlc_word,rlc_high,rlc_low]: [SecondPhaseAdviceColumn;3], poseidon:&implPoseidonLookup, bytes:&implBytesLookup, rlc:&implRlcLookup, randomness:Query<F>,){ cb.add_lookup("old_highis16bytes", [high.current(),Query::from(15)], bytes.lookup(),); cb.add_lookup("old_lowis16bytes", [low.current(),Query::from(15)], bytes.lookup(),); cb.poseidon_lookup("word_hash=poseidon(high,low)", [high.current(),low.current(),word_hash.current()], poseidon,); cb.add_lookup("rlc_high=rlc(high)andhighis16bytes", [high.current(),Query::from(15),rlc_high.current()], hasheye 24ScrollzkEVMCircuitsSecurityAssessment PUBLIC

rlc.lookup(),); cb.add_lookup("rlc_low=rlc(low)andlowis16bytes", [low.current(),Query::from(15),rlc_low.current()], rlc.lookup(), Figure5.1: mpt-circuit/src/gadgets/mpt_update/word_rlc.rs#16-49 Althoughthe WordRLC::configure functionreceivestwodifferentlookupobjects, bytes and rlc ,theyareinstantiatedwiththesameconcretelookup: letmpt_update=MptUpdateConfig::configure(cs, &mutcb, poseidon, &key_bit, &byte_representation, &byte_representation, &rlc_randomness, &canonical_representation,); Figure5.2: mpt-circuit/src/mpt.rs#60-69 Wealsonotethatthelabelsrefertotheupperandlowerbytesas old_high and old_low insteadofjust high and low . Recommendations Shortterm,determinewhetherboththe BytesLookup and RlcLookup tablesareneeded forthiscircuit,andrefactorthecircuitaccordingly,removingtheredundantconstraints. Longterm,reviewthecodebaseforduplicatedorredundantconstraintsusingmanualand automatedmethods. hasheye 25ScrollzkEVMCircuitsSecurityAssessment PUBLIC

6.TheNonceChangedconfigurationcircuitdoesnotconstrainthenewvalue noncevalue Severity:HighDifficulty:Low Type:CryptographyFindingID:TOB-SCROLL2-6 Target: src/gadgets/mpt_update.rs Description The NonceChanged configurationcircuitdoesnotconstrainthe config.new_value parametertobe8bytes.Instead,thereisaduplicatedconstraintfor config.old_value : SegmentType::AccountLeaf3⇒{ cb.assert_zero("directionis0",config.direction.current()); letold_code_size=(config.old_hash.current()-config.old_value.current()) *Query::Constant(F::from(1<<32).square().invert().unwrap()); letnew_code_size= (config.new_hash.current()-config.new_value.current()) *Query::Constant(F::from(1<<32).square().invert().unwrap()); cb.condition(config.path_type.current_matches(&[PathType::Common]), |cb|{ cb.add_lookup("oldnonceis8bytes", [config.old_value.current(),Query::from(7)], bytes.lookup(),); cb.add_lookup("newnonceis8bytes", [config.old_value.current(),Query::from(7)], bytes.lookup(),); Figure6.1: mpt-circuit/src/gadgets/mpt_update.rs#1209-1228 Thismeansthatamaliciousprovercouldupdatethe Account nodewithavalueof arbitrarylengthforthe Nonce and Codesize parameters. Thesameconstraint(withacorrectlabelbutincorrectvalue)isusedinthe ExtensionNew pathtype: cb.condition(config.path_type.current_matches(&[PathType::ExtensionNew]), |cb|{ cb.add_lookup(hasheye 26ScrollzkEVMCircuitsSecurityAssessment PUBLIC

"newnonceis8bytes", [config.old_value.current(),Query::from(7)], bytes.lookup(),); Figure6.2: mpt-circuit/src/gadgets/mpt_update.rs#1241-1248 ExploitScenario Amaliciousproverusesthe NonceChanged prooftoupdatethenoncewithalargerthan expectedvalue. Recommendations Shortterm,enforcetheconstraintforthe config.new_value witness. Longterm,addpositiveandnegativetestingoftheedgecasespresentinthespecification. Forboththe Common and ExtensionNew pathtypes,thereshouldbeanegativetestthat failsbecauseitchangesthenewnoncetoavaluelargerthan8bytes.Useautomated testingtoolslikeSemgreptofindredundantandduplicateconstraints,asthesecould indicatethataconstraintisincorrect. hasheye 27ScrollzkEVMCircuitsSecurityAssessment PUBLIC

7.TheCopycircuitdoesnottotallyenforcethetagvalues Severity:InformationalDifficulty:N/A Type:CryptographyFindingID:TOB-SCROLL2-7 Target: src/copy_circuit/copy_gadgets.rs Description TheCopytableincludesatagcolumnthatindicatesthetypeofdataforthatparticularrow. However,theCopycircuittagvalidationfunctiondoesnottotallyensurethatthetag matchesoneofthepredefinedtagvalues. Theimplementationusesthe copy_gadgets::constrain_tag functiontobindthe is_precompiled , is_tx_calldata , is_bytecode , is_memory ,and is_tx_log witnessestotheactualtagvalue.However,thecodedoesnotensurethatexactlyoneof theseBooleanvaluesis true . #[allow(clippy::too_many_arguments)] pubfnconstrain_tag<F:Field>(meta:&mutConstraintSystem<F>, q_enable:Column<Fixed>, tag:BinaryNumberConfig<CopyDataType,4>, is_precompiled:Column<Advice>, is_tx_calldata:Column<Advice>, is_bytecode:Column<Advice>, is_memory:Column<Advice>, is_tx_log:Column<Advice>,){ meta.create_gate("decodetag",|meta|{ letenabled=meta.query_fixed(q_enable,CURRENT); letis_precompile=meta.query_advice(is_precompiled,CURRENT); letis_tx_calldata=meta.query_advice(is_tx_calldata,CURRENT); letis_bytecode=meta.query_advice(is_bytecode,CURRENT); letis_memory=meta.query_advice(is_memory,CURRENT); letis_tx_log=meta.query_advice(is_tx_log,CURRENT); letprecompiles=sum::expr([tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Ecrecover), CURRENT,)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Sha256), CURRENT)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Ripemd160), CURRENT, hasheye 28ScrollzkEVMCircuitsSecurityAssessment PUBLIC

)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Identity), CURRENT)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Modexp), CURRENT)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Bn128Add), CURRENT)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Bn128Mul), CURRENT)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Bn128Pairing), CURRENT,)(meta), tag.value_equals(CopyDataType::Precompile(PrecompileCalls::Blake2F), CURRENT)(meta),]); vec![//Matchbooleanindicatorstotheirrespectivetagvalues. enabled.expr()*(is_precompile-precompiles), enabled.expr() *(is_tx_calldata-tag.value_equals(CopyDataType::TxCalldata, CURRENT)(meta)), enabled.expr() *(is_bytecode-tag.value_equals(CopyDataType::Bytecode, CURRENT)(meta)), enabled.expr()*(is_memory-tag.value_equals(CopyDataType::Memory, CURRENT)(meta)), enabled.expr()* (is_tx_log-tag.value_equals(CopyDataType::TxLog, CURRENT)(meta)),] }); } Figure7.1: copy_circuit/copy_gadgets.rs#13-62 Infact,thetagvaluecouldequal CopyDataType::RlcAcc ,asinthe SHA3 gadget.The CopyDataType::Padding valueisalsonotcurrentlymatched. Inthecurrentstateofthecodebase,thisissuedoesnotappeartocauseanysoundness issuesbecausethelookupsintotheCopytableeitheruseastaticallysetsourceand destinationtagor,asinthecaseofprecompiles,thevalueiscorrectlyboundedanddoes

notposeanavenueofattackforamaliciousprover.
WealsoobservethattheCopycircuitspecificationmentionsawitnessvalueforthe is_rlc_acc
case,butthisisnotreflectedinthecode. Recommendations
Shortterm,ensurethatthetagcolumnisfullyconstrained.Reviewthecircuitspecification
andmatchtheimplementationwiththespecification,documentingpossibleoptimizations
thatremoveconstraintsanddetailingwhytheydonotcausesoundnessissues. hasheye
29ScrollzkEVMCircuitsSecurityAssessment PUBLIC

Longterm,includenegativetestsforanunintendedtagvalue. hasheye
30ScrollzkEVMCircuitsSecurityAssessment PUBLIC

8.The“invalidcreation”errorhandlingcircuitisunconstrained Severity:HighDifficulty:Medium
Type:CryptographyFindingID:TOB-SCROLL2-8 Target:
evm_circuit/execution/error_invalid_creation_code.rs Description
The“invalidcreation”errorhandlingcircuitdoesnotconstrainthefirstbyteoftheactual memorytobe 0xef
asintended.Thisallowsamaliciousprovertoredirecttheevm executiontoahaltafterthe CREATE
opcodeiscalled,regardlessofthememoryvalue. The ErrorInvalidCreationCodeGadget
circuitwasupdatedtoaccommodatethe memoryaddressingoptimizations.However,indoingso,the first_byte
witnessvalue thatwasboundtothememory’sfirstbyteisnolongerboundtoit.Therefore,amalicious
provercanalwayssatisfythecircuitconstraints,evenifitheyarenotinanerrorstateafter the CREATE
opcodeiscalled. fnconfigure(cb:&mutEVMConstraintBuilder<F>)→Self{ letopcode=cb.query_cell();
letfirst_byte=cb.query_cell(); //letaddress=cb.query_word_rlc(); letoffset=cb.query_word_rlc();
letlength=cb.query_word_rlc(); letvalue_left=cb.query_word_rlc(); cb.stack_pop(offset.expr());
cb.stack_pop(length.expr()); cb.require_true("is_createistrue",cb.curr.state.is_create.expr());
letaddress_word=MemoryWordAddress::construct(cb,offset.clone()); //lookupmemoryforfirstword
cb.memory_lookup(0.expr(), address_word.addr_left(), value_left.expr(), value_left.expr(), None,
); //letfirst_byte=value_left.cells[address_word.shift()]; //constrainfirstbyteis0xef
letis_first_byte_invalid=IsEqualGadget::construct(cb,first_byte.expr(), hasheye
31ScrollzkEVMCircuitsSecurityAssessment PUBLIC

0xef.expr()); cb.require_true("is_first_byte_invalidistrue", is_first_byte_invalid.expr(),);
Figure8.1: evm_circuit/execution/error_invalid_creation_code.rs#36-67 ExploitScenario
Amaliciousprovergeneratestwodifferentproofsforthesametransaction,oneleadingto
theerrorstate,andtheothersuccessfullyexecutingthe CREATE opcode.Distributingthese
proofstotwoendsofabridgeleadstostatedivergenceandalossoffunds. Recommendations Shortterm,bindthe
first_byte witnessvaluetothememoryvalue;ensurethatthe successful CREATE
endstatechecksthatthefirstbyteisdifferentfrom 0xef .
Longterm,investigatewaystogeneratemaliciousttracesthatcouldbeaddedtothetest
suite;everytimeanewsoundnessissueisfound,createsuchamaliciousttraceandaddit tothetestsuite. hasheye
32ScrollzkEVMCircuitsSecurityAssessment PUBLIC

9.TheOneHotprimitiveallowsmorethanonevalueatonce Severity:HighDifficulty:Low
Type:CryptographyFindingID:TOB-SCROLL2-9 Target: constraint_builder/binary_column.rs Description
The OneHot primitiveuses BinaryQuery valuesaswitnessvalues.However,despitetheir
name,thesevaluesarenotconstrainedtobeBooleanvalues,allowingamaliciousprover
tochoosemorethanone“hot”valueinthedatastructure. impl<T:IntoEnumIterator+Hash+Eq>OneHot<T>{
pubfnconfigure<F:FieldExt>(cs:&mutConstraintSystem<F>, cb:&mutConstraintBuilder<F>,)→Self{
letmutcolumns=HashMap::new(); forvariantinSelf::nonfirst_variants(){
columns.insert(variant,cb.binary_columns::<1>(cs)[0]); } letconfig=Self{columns}; cb.assert(
"sumofbinarycolumnsinOneHotis0or1", config.sum(0).or(!config.sum(0)),); config } Figure9.1: mpt-
circuit/src/gadgets/one_hot.rs#14-30 Thereasonthe BinaryQuery
valuesarenotconstrainedtobeBooleanisbecausethe BinaryColumn
configurationdoesnotconstraintheadvicevaluestobeBoolean,andthe
configurationissimplyatypewrapperaroundthe Column<Advice> type.Thisprovidesno
guaranteestotheusersofthisAPI,whomightassumethatthesevaluesareguaranteedto beBoolean.
pubfnconfigure<F:FieldExt>(cs:&mutConstraintSystem<F>, _cb:&mutConstraintBuilder<F>,)→Self{
letadvice_column=cs.advice_column(); //TODO:constraintobebinaryhere... //cb.add_constraint()
Self(advice_column) hasheye 33ScrollzkEVMCircuitsSecurityAssessment PUBLIC

} Figure9.2: mpt-circuit/src/constraint_builder/binary_column.rs#29-37 The OneHot
primitiveisusedtoimplementtheMerklepath-checkingstatemachine,
includingcriticalpropertiessuchasrequiringthe key and other_key columnstoremain
unchangedalongagivenMerklepathcalculation,asshowninfigure9.3. cb.condition(
!segment_type.current_matches(&[SegmentType::Start,SegmentType::AccountLeaf3]), |cb|{
cb.assert_equal("keycanonlychangeonStartorAccountLeaf3rows", key.current(), key.previous(),);
cb.assert_equal("other_keycanonlychangeonStartorAccountLeaf3rows", other_key.current(),
other_key.previous(),); },); Figure9.3: mpt-circuit/src/gadgets/mpt_update.rs#170-184

Wedidnotdevelopaproof-of-conceptexploitforthepath-checkingtable,soitmaybethe casethattheconstraintinfigure9.3isnotexploitableduetootherconstraints.However,if atanypointitispossibletomatchboth SegmentType::Start andsomeothersegment type(suchasbysettingone OneHot cellto 1 andanotherto -1),amaliciousproverwould beabletochangethekeypartwaythroughandforgeMerkleupdates. ExploitScenario Amaliciousproverusesthe OneHot soundnessissuetobypassconstraints,ensuringthat the key and other_key columnsremainunchangedalongagivenMerklepathcalculation. ThisallowstheattackertosuccessfullyforgeMPTupdateproofsthatupdateanarbitrary key. Recommendations Shortterm,addconstraintsthatensurethattheadvicevaluesfromthesecolumnsare Boolean. Longterm,addpositiveandnegativetestsensuringthattheseconstraintbuildersoperate accordingtotheirexpectations. hasheye 34ScrollzkEVMCircuitsSecurityAssessment PUBLIC

10.Intermediatecolumnsarenotexplicit Severity:InformationalDifficulty:N/A Type:CryptographyFindingID:TOB-SCROLL2-10 Target: src/mpt_update.rs Description TheMPTupdatecircuitincludestwoarraysof“intermediatevalue”columns,asshownin figure10.1. intermediate_values:[AdviceColumn;10],//canbe4? second_phase_intermediate_values: [SecondPhaseAdviceColumn;10],//4? Figure10.1: mpt-circuit/src/gadgets/mpt_update.rs#65-66 Thesecolumnsareusedasgeneral-usecellsforvaluesthatareonlyconditionallyneeded inagivenrow,reducingthetotalnumberofcolumnsneeded.Forexample,figure10.2 showsthat intermediate_values[0] isusedforthe address valueinrowsthatmatch SegmentType::Start ,butasshowninfigure10.3,rowsrepresentingthe SegmentType::AccountLeaf3 stateofaKeccakcode-hashproofusethatsameslotfor the old_high value. letaddress=self.intermediate_values[0].current()*is_start(); Figure10.2: mpt-circuit/src/gadgets/mpt_update.rs#78 SegmentType::AccountLeaf3⇒{ cb.assert_equal("directionis1",config.direction.current(),Query::one()); let[old_high,old_low,new_high,new_low,..]=config.intermediate_values; Figure10.3: mpt-circuit/src/gadgets/mpt_update.rs#1632-1635 Insomecases,cellsof intermediate_values areusedstartingfromtheendofthe intermediate_values column,suchasthe other_key_hash and other_leaf_data_hash valuesin PathType::ExtensionOld rows,asillustratedin figure10.4. let[..,key_equals_other_key,new_hash_is_zero]=config.is_zero_gadgets; let[..,other_key_hash,other_leaf_data_hash]=config.intermediate_values; nonexistence_proof::configure(cb, hasheye 35ScrollzkEVMCircuitsSecurityAssessment PUBLIC

config.new_value, config.key, config.other_key, key_equals_other_key, config.new_hash, new_hash_is_zero, other_key_hash, other_leaf_data_hash, poseidon,); Figure10.4: mpt-circuit/src/gadgets/mpt_update.rs#1036-1049 Althoughwedidnotfindanymistakessuchasmisusedcolumns,thispatternisadhocand error-prone,andevaluatingthecorrectnessofthispatternrequirescheckingevery individualuseof intermediate_values . Recommendations Shortterm,documenttheassignmentofall intermediate_values columnsineach relevantcase. Longterm,considerusingRusttypestoexpressthedifferentusesofthevarious intermediate_values columns.Forexample,onecoulddefinean IntermediateValues enum,withcaseslike StartRow{address:&AdviceColumn} and ExtensionOld{other_key_hash:&AdviceColumn,other_leaf_data_hash: &AdviceColumn} ,andasinglefunction fn parse_intermediate_values(segment_type:SegmentType,path_type: PathType,columns:&[AdviceColumn;10])→IntermediateValues .Then,the correctassignmentanduseof intermediate_values columnscanbeauditedonlyby checking parse_intermediate_values . hasheye 36ScrollzkEVMCircuitsSecurityAssessment 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 37ScrollzkEVMCircuitsSecurityAssessment 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
38ScrollzkEVMCircuitsSecurityAssessment PUBLIC

B.CodeMaturityCategories
Thefollowingtablesdescribethecodematuritycategoriesandratingcriteriausedinthis document.
CodeMaturityCategories CategoryDescription
ArithmeticTheproperuseofmathematicaloperationsandsemantics Complexity Management
Thepresenceofclearstructuresdesignedtomanagesystemcomplexity, includingtheseparationofsystemlogicintoclearlydefinedfunctions Cryptographyand KeyManagement
Thesafeuseofcryptographicprimitivesandfunctions,alongwiththe presenceofrobustmechanismsforkeygenerationanddistribution
DocumentationThepresenceofcomprehensiveandreadablecodebasedocumentation MemorySafety andErrorHandling Thepresenceofmemorysafetyandrobusterror-handlingmechanisms Testingand Verification
Thepresenceofrobusttestingprocedures(e.g.,unittests,integration tests,andverificationmethods)andsufficienttestcoverage RatingCriteria RatingDescription
StrongNoissueswerefound,andthesystemexceedsindustrystandards.
SatisfactoryMinorissueswerefound,butthesystemiscompliantwithbestpractices.
ModerateSomeissuesthatmayaffectsystemsafetywerefound.
WeakManyissuesthataffectsystemsafetywerefound.
MissingArequiredcomponentismissing,significantlyaffectingsystemsafety.
NotApplicableThecategoryisnotapplicabletothisreview.
NotConsideredThecategorywasnotconsideredinthisreview. Further Investigation Required
Furtherinvestigationisrequiredtoreachameaningfulconclusion. hasheye
39ScrollzkEVMCircuitsSecurityAssessment PUBLIC

C.CodeQualityFindings Weidentifiedthefollowingcodequalityissuesthroughmanualandautomaticcodereview.
• Allowthe dead_code lintandfixallissues.The dead_code lintiscurrently disabled;itshouldbeenabledtoallowdeveloperstoquicklydetectunused functionsandvariables. • Severalconstraintshavemeaninglesslabels.Constraintlabelsareusefulfor explainingtheintentionbehindconstraints;usingmeaninglesslabelshinderscode readability. region .assign_fixed(||"asdfasdfawe", self.0, FigureC.1: mpt-circuit/src/constraint_builder/column.rs#52-55 cb.assert_equal("???????",rlc.current(),byte.current()); });
cb.condition(!index_is_zero.current(),|cb|{ cb.assert_equal("valuecanonlychangewhenindex=0", value.current(), value.previous(),); cb.assert_equal("differences_are_zero_so_far=difference==0&& differences_are_zero_so_far.previous()whenindex≠0", differences_are_zero_so_far.current().into(), differences_are_zero_so_far .previous() .and(difference_is_zero.previous()) .into(),);
cb.assert_equal("???", FigureC.2: mpt-circuit/src/gadgets/canonical_representation.rs#72-89 • Thereareduplicateandunusedfunctionsinthecodebase.The types.rs and util.rs fileshaveseveralduplicatefunctions: fr , hash , storage_key_hash , split_word , hi_lo ,and Bit . • Thefollowingconstraintlabelwasincorrectlycopy-pasted. cb.assert_equal(hasheye
40ScrollzkEVMCircuitsSecurityAssessment PUBLIC

"old_valuedoesnotchange", new_value.current(), new_value.previous(),); FigureC.3: mpt-circuit/src/gadgets/mpt_update.rs#163-167 • Thereareredundantconstraintsinemptystorage/accountconstraints.The configure_empty_storage and configure_empty_account functionsrequire the new_value and old_value fieldstobe 0 butalsoconstrainthemtobeequal. cb.assert_zero("oldvalueis0foremptystorage", config.old_value.current(),); cb.assert_zero("newvalueis0foremptystorage", config.new_value.current(),); ... cb.assert_equal("oldvalue=newvalueforemptyaccountproof", config.old_value.current(), config.new_value.current(),); FigureC.4: mpt-circuit/src/gadgets/mpt_update.rs#1785-1811
cb.assert_zero("oldvalueis0",config.old_value.current());
cb.assert_zero("newvalueis0",config.new_value.current()); ... cb.assert_equal("oldvalue=newvalueforemptyaccountproof", config.old_value.current(), config.new_value.current(),);
FigureC.5: mpt-circuit/src/gadgets/mpt_update.rs#1869-1893 • Thefollowingconstraintlabelisimprecise.Thelabelshouldread rwc_inc_left[1]==rwc_inc_left[0]-rwc_diff,or0attheend tomatch thecode.

//Decrementrwc_inc_leftforthenextrow,whenanRWoperationhappens.
letrwc_diff=is_rw_type.expr()*is_word_end.expr();
letnew_value=meta.query_advice(rwc_inc_left,CURRENT)-rwc_diff; //Attheend,itmustreach0.
letupdate_or_finish=select::expr(not::expr(is_last.expr()),
meta.query_advice(rwc_inc_left,NEXT_ROW), 0.expr(), hasheye 41ScrollzkEVMCircuitsSecurityAssessment PUBLIC

); cb.require_equal("rwc_inc_left[2]==rwc_inc_left[0]-rwc_diff,or0attheend", new_value, update_or_finish,); FigureC.6: src/copy_circuit/copy_gadgets.rs#524-537 • The IsZeroGadgetassign functiondoesnotassignthewitnessvalue. pubfnassign<F:FieldExt,T:Copy+TryInto<F>>(&self, region:&mutRegion<'_,F>, offset:usize, value:T,)where <TasTryInto<F>>::Error:Debug, { self.inverse_or_zero.assign(region, offset, value.try_into().unwrap().invert().unwrap_or(F::zero()),); } //TODO:getridofassignmethodinfavorofit. pubfnassign_value_and_inverse<F:FieldExt,T:Copy+TryInto<F>>(&self, region:&mutRegion<'_,F>, offset:usize, value:T,)where <TasTryInto<F>>::Error:Debug, { self.value.assign(region,offset,value); self.assign(region,offset,value); } FigureC.7: mpt-circuit/src/gadgets/is_zero.rs#20-46 • The OneHotassign functionshouldensurethatnomorethanoneitemis assigned. pubfnassign<F:FieldExt>(&self,region:&mutRegion<'_,F>,offset:usize, value:T){ ifletSome(c)=self.columns.get(&value){ c.assign(region,offset,true) } } FigureC.8: mpt-circuit/src/gadgets/one_hot.rs#31-36 hasheye 42ScrollzkEVMCircuitsSecurityAssessment PUBLIC

• Thereisaredundantconditioninthe configure_empty_account function. Thefunctioncouldjustmatch SegmentType::Start . SegmentType::Start|SegmentType::AccountTrie⇒{ letis_final_segment= config.segment_type.next_matches(&[SegmentType::Start]); cb.condition(is_final_segment,|cb|{ FigureC.9: mpt-circuit/src/gadgets/mpt_update.rs#1878-1880 • Thereisaredundant .and() callinthe MptUpdateConfigconfigure function. The cb.every_row_selector() functionreturnsthefirstconditioninthe conditionstack,sothisconditionisequivalentto is_start . cb.condition(is_start.clone().and(cb.every_row_selector()),|cb|{ FigureC.10: mpt-circuit/src/gadgets/mpt_update.rs#124 • The rw_counter constraintscouldbeconsolidated.Constraining rw_counter requiresconstrainingthetagto Memory or TxLog andconstrainingthe Padding to 0 .Theseconstraintsareimplementedindifferentlocationsinthecodebase,making thecodehardertounderstand. letis_rw_type=meta.query_advice(is_memory,CURRENT)+is_tx_log.expr(); FigureC.11: src/copy_circuit.rs#340-341 //Decrementrwc_inc_leftforthenextrow,whenanRWoperationhappens. letrwc_diff=is_rw_type.expr()*is_word_end.expr(); FigureC.12: copy_circuit/copy_gadgets.rs#L525 • Thefollowingconstraintlabelisimprecise.Thelabelshouldread assign real_bytes_left{} . //real_bytes_left region.assign_advice(||format!("assignbytes_left{}",*offset), self.copy_table.real_bytes_left, *offset, ||Value::known(F::zero()),)?; FigureC.13: src/copy_circuit.rs#776-782 hasheye 43ScrollzkEVMCircuitsSecurityAssessment PUBLIC

D.AutomatedAnalysisToolConfiguration
Aspartofthisassessment,weusedthetoolsdescribedbelowtoperformautomated testingofthecodebase.
D.1.Semgrep WeusedthestaticanalyzerSemgreptosearchforriskyAPIpatternsandweaknessesinthe sourcecoderepository.Forthispurpose,wewroterulesspecificallytargetingthe ConstraintBuilder APIsandthe ExecutionGadget trait. semgrep--metrics=off--sarif--config=custom_rule_path.yml FigureD.1:TheinvocationcommandusedtorunSemgrepforeachcustomrule DuplicateConstraints Thepresenceofduplicateconstraints,withpotentiallydifferentlabels,indicateseithera redundantconstraintthatcanberemovedoranintendedconstraintthatwasnotcorrectly updated.ThispatternwaswrittentofindvariantsoffindingTOB-SCROLL2-6,butnoother instancesofitwerefoundinthecodebase.However,thispatternshouldbeaddedasa CI/CDSemgrepruletopreventasimilarissuefromrecurringinthecodebase. rules: -id:repeated-constraints message:"Foundredundantorincorrectlyupdatedconstraint" languages:[rust] severity:ERROR patterns: -pattern:| cb.$FUNC($LABEL1,$LEFT,$RIGHT); ... cb.$FUNC($LABEL2,$LEFT,$RIGHT); FigureD.2:The repeated-constraints Semgreprule ConstraintswithRepeatedLabels Thepresenceofarepeatedlabelcouldindicateacopy-pastedlabelthatshouldbe updated. rules: -id:constraints-with-repeated-labels message:"Foundconstraintswiththesamelabel" languages:[rust] severity:ERROR patterns: -pattern:| hasheye 44ScrollzkEVMCircuitsSecurityAssessment PUBLIC

cb.$FUNC("$LABEL",...); ... cb.$FUNC("$LABEL",...); FigureD.3:The repeated-labels Semgreprule D.2.cargollvm-cov cargo-llvm-cov generatesRustcodecoveragereports.Weusedthe cargollvm-cov --open commandintheMPTcodebasetogeneratethecoveragereportpresentedinthe AutomatedTestingsection. D.3.cargoedit cargo-edit allowsdeveloperstoquicklyfindoutdatedRustcrates.Thetoolcanbe installedwiththe cargoinstallcargo-edit commandandthe cargoupgrade --incompatible--dry-run commandcanbeusedtofindoutdatedcrates. D.4.Clippy TheRustlinterClippycanbeinstalledusing rustup byrunningthecommand rustup componentaddclippy .Invoking cargoclippy--workspace---Wclippy::pedantic intherootdirectoryoftheprojectrunsthetoolwiththepedanticruleset. cargoclippy--workspace---

Wclippy::pedantic FigureD.4:TheinvocationcommandusedtorunClippyinthecodebase hasheye
45ScrollzkEVMCircuitsSecurityAssessment PUBLIC

E.FixReviewResults Whenundertakingafixreview,hasheyereviewsthefixesimplementedforissues
identifiedintheoriginalreport.Thisworkinvolvesareviewofspecificareasofthesource
codeandsystemconfiguration,notcomprehensiveanalysisofthesystem.
FromSeptember25toSeptember29,2023,hasheyereviewedthefixesandmitigations
implementedbytheScrollteamfortheissuesidentifiedinthisreport.Wereviewedeach
fixtodetermineitseffectivenessinresolvingtheassociatedissue. ScrollprovidedPRswithfixesforallhigh-
severityfindingsandforthe informational-severityfindingTOB-SCROLL2-
7.Scrolldidnotsubmitfixesfortheremaining informational-severityfindings.
Insummary,ofthe10issuesdescribedinthisreport,Scrollhasresolvedthreeissuesand
haspartiallyresolvedoneissue.Scrollindicatedthatitdoesnotintendtofixfinding TOB-SCROLL2-
2,soitsstatusisunresolved.NofixPRswereprovidedfortheremainingfive
issues,sotheirfixstatusesareundetermined.Foradditionalinformation,pleaseseethe
DetailedFixReviewResultsbelow. IDTitleStatus 1PoseidonLookupisnotimplementedUndetermined
2IsZeroGadgetdoesnotconstraintheinversewitnesswhenthe valueiszero Unresolved
3TheMPTnonexistenceproofgadgetismissingconstraintsspecified inthedocumentation Undetermined
4DiscrepanciesbetweentheMPTcircuitspecificationand implementation Undetermined
5RedundantlookupsintheWordRLCcircuitUndetermined
6TheNonceChangedconfigurationcircuitdoesnotconstrainthe newvaluenoncevalue Resolved hasheye
46ScrollzkEVMCircuitsSecurityAssessment PUBLIC

7TheCopycircuitdoesnottotallyenforcethetagvaluesPartially Resolved
8The“invalidcreation”errorhandlingcircuitisunconstrainedResolved
9TheOneHotprimitiveallowsmorethanonevalueatonceResolved
10IntermediatecolumnsarenotexplicitUndetermined hasheye 47ScrollzkEVMCircuitsSecurityAssessment
PUBLIC

DetailedFixReviewResults TOB-SCROLL2-1:PoseidonLookupisnotimplemented
Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-
SCROLL2-2:IsZeroGadgetdoesnotconstraintheinversewitnesswhenthevalue iszero
Unresolved.TheScrollteamhasindicatedthatitdoesnotintendtofixthisissue. TOB-SCROLL2-
3:TheMPTnonexistenceproofgadgetismissingconstraintsspecified inthedocumentation
Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-
SCROLL2-4:DiscrepanciesbetweentheMPTcircuitspecificationand implementation
Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-
SCROLL2-5:RedundantlookupsintheWordRLCcircuit
Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-
SCROLL2-6:TheNonceChangedconfigurationcircuitdoesnotconstrainthenew valuenoncevalue
ResolvedinPR#73.Thetwoconditionalconstraintsthatreferredto old_value insteadof new_value
havebeenfactoredoutintoasingleunconditionalconstraintreferringto new_value ,andthe ExtensionOld
casehasbeenremovedentirelyinfavorofablanket assertionforbiddingthatcaseof configure_nonce . TOB-
SCROLL2-7:TheCopycircuitdoesnottotallyenforcethetagvalues
PartiallyresolvedinPR#809.Severalcasesofthe CopyDataType taghavebeenremoved,
andanassertionhasbeenaddedtodocumentthatthe Padding tagvalueisinternalonly.
Wedidnotfullyevaluatewhetherthispreventstagvaluesoutsidetheexpectedrange. TOB-SCROLL2-
8:The“invalidcreation”errorhandlingcircuitisunconstrained ResolvedinPR#751.The MemoryMask
gadgetisusedtoextractthecorrectbytefromthe wordat offset andtoconstrainittoequal 0xef . TOB-
SCROLL2-9:TheOneHotprimitiveallowsmorethanonevalueatonce ResolvedinPR#69.Eachbinarycolumnvalue v
isconstrainedbyenforcing 1-v.or(!v) ==0
.Itisnotimmediatelyobviousthatthisconstraintsuffices,butitisequivalentto bythefollowingreasoning: ∅
(1-∅)=0 hasheye 48ScrollzkEVMCircuitsSecurityAssessment PUBLIC

1-v.or(!v)==1-!((!v).and(!!v)==1-(1-((!v)*(!!v)))... ...==((1-v)*(1-(1-v)))==(1-v)*v
Inaddition,anotherrelatedissuewiththe OneHot primitive,whichwasnotdiscovered
duringtheinitialreviewengagement,wasfixedinPR#68.Therelatedproblemwasatypo causing OneHot::previous
toreturntheresultofthecurrentrowratherthanthe
previousrow.Itsexploitscenariowouldbeeffectivelythesameastheonedescribedin findingTOB-SCROLL2-
9.TheScrollteamfixedthisbyreplacingthevalue BinaryColumn::current with BinaryColumn::previous .
TOB-SCROLL2-10:Intermediatecolumnsarenotexplicit
Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. hasheye
49ScrollzkEVMCircuitsSecurityAssessment PUBLIC

F.FixReviewStatusCategories
Thefollowingtabledescribesthestatusesusedtoindicatewhetheranissuehasbeen sufficientlyaddressed.
FixStatus StatusDescription UndeterminedThestatusoftheissuewasnotdeterminedduringthisengagement.

UnresolvedTheissuepersistsandhasnotbeenresolved.
PartiallyResolvedTheissuepersistsbuthasbeenpartiallyresolved.
ResolvedTheissuehasbeensufficientlyresolved. hasheye 50ScrollzkEVMCircuitsSecurityAssessment PUBLIC