HashEye

// SMART CONTRACT SECURITY AUDIT

Roll

Security assessment by HashEye · prepared for Blockchain

HASHEYE AUDITED

PROJECT	Roll
CLIENT	Blockchain
CATEGORY	Blockchain
PUBLISHED	July 1, 2022
REPORT ID	research-roll-2022-07-01-1sivrt

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-roll-2022-07-01-1sivrt.

ScrollzkEVMhalo2Circuits SecurityAssessment October12,2023 Preparedfor: HaichenShen Scroll
Preparedby:FilipeCasal,JoeDoyle,OpalWright,andWillSong

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.

TableofContents Abouthasheye1 NoticesandRemarks2 TableofContents3 ExecutiveSummary5 ProjectSummary7 ProjectGoals8 ProjectTargets9 ProjectCoverage10 AutomatedTesting11 CodebaseMaturityEvaluation12 SummaryofFindings14 DetailedFindings17

ExecutiveSummary EngagementOverview
ScrollengagedhasheyetoreviewthesecurityofitszkEVMhalo2circuits.Thescoped
codebaseimplementstheEthereumVirtualMachine(EVM)opcodes,aswellasother
crucialcomponentsofScroll’szkEVM:theStatecircuit,theBytecodecircuit,andthecircuit forthe modexp
precompile.Additionally,weweretaskedwithreviewingchangestothe Keccakcircuitandtothe halo2-lib and
snark-verifier circuits. Ateamoffourconsultantsconductedthereviewfrom April17toJune23,2023,foratotal
of23engineer-weeksofeffort.Ourtestingeffortsfocusedoncircuitsoundnessandthe
correctimplementationoftheEVMsemantics.Withfullaccesstothesourcecodeand
documentation,weperformedstaticanddynamictestingofthezkEVMhalo2circuits,using
automatedandmanualprocesses. ObservationsandImpact Thesecurityreviewrevealedmanyhigh-
impactvulnerabilitiesrelatedtocircuitsoundness.
Duetoincorrect,incomplete,ormissingconstraints,amaliciousprovercouldconvincea
verifierofanEVMexecutionthatdoesnotmatchthecorrectEVMsemantics.Specifically,
anattackercouldcauseopcodestoreturnanunintendedresult(TOB-SCROLL-1, TOB-SCROLL-
3),manipulategascostsofcertainopcodes(TOB-SCROLL-7,TOB-SCROLL-12),
executedifferentopcodesfromwhatwasspecifiedinthebytecode(TOB-SCROLL-13, TOB-SCROLL-
14),orcallcertainopcodesinsituationswheredoingsoshouldnotbe
allowedaccordingtotheEVMspecification(TOB-SCROLL-9).Allofthesecasescouldleadto
statedivergenceandcauselossoffundsifthezkEVMisusedinthecontextofabridge.The
gascostsmanipulationissuescouldalsoenabledenial-of-serviceattacks,andfinding TOB-SCROLL-
9couldenablereentrancyattacksatthecontractlevel. Recommendations
Basedonthecodebasematurityevaluationandfindingsidentifiedduringthesecurity
review,hasheyerecommendsthatScrolltakethefollowingstepsbeforedeployment: •
Remediatethefindingsdisclosedinthisreport.Thesefindingsshouldbe
addressedaspartofadirectremediationoraspartofanyrefactoringthatmay
occurwhenaddressingotherrecommendations. •
InvestinadversarialtestingThesecurityrequirementsofazkEVM
implementationareextremelystrict,andthecomplexityofevaluatingsuchan
implementationisextremelyhigh.Theuseofnondeterministiccomputationina
complexcircuitsuchasthezkEVMissimultaneouslyacrucialoptimization
techniqueandasourceofpotentiallycatastrophicerrors,asdescribedin TOB-SCROLL-
13.Evenerrorsthatseemextremelyminor,suchasTOB-SCROLL-7, hasheye
5ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

canleadtodivergentexecution.Andevenintheabsenceofdivergentexecution
bugs,thecomplexityofexecutingEVMbytecodeprovidesawealthofpotential
implementationmistakes,whereminorerrorscancausestatedivergenceifazkEVM
isimproperlydeployedaspartofacross-chainbridge—forexample,ifonesideof
thebridgeusesthezkEVMcircuitandtheotherusesa go-ethereum
implementationtweakedinaccordancewithScroll’spublicdocumentation.In
addition,anyoftheseerrorscanbeintroducedbyseeminglyinnocuouschanges duringthedevelopmentprocess.
Withalloftheseconsiderationsinmind,webelievethatthisprojectrequiresan
unusuallyhighlevelofassuranceandwillcontinuetoneedactivetestingandreview
throughoutitslifetime.WestronglyencourageScrolltoinvestinanongoing,
adversarialtestingprocess,especiallyfocusedonpotentialmaliciousprover
behavior.WealsoencourageScrolltodevelopaclearwrittenspecificationforeach
specializedargument(e.g.,theRWtable,RLC-basedwords)usedinthefinal
implementation.Finally,investinacontinuousinternalcode-reviewingeffort. •
LeveragetheRusttypesystemtoenforcecompile-timeinvariants.TheRust

typesystemshouldbeusedtoensurethatcertainwitnessvalueshavebeen constrained.Asanexample,certainBooleanoperationsdefinedinthecodebase requiretheirargumentstobeBoolean.However,ensuringthattheargumentsare Booleaniscurrentlyeitherenforcedinanadhocmannerorsometimesnoteven enforced,leadingtopotentialsoundnessissues.DefiningnewRusttypesand signaturesfortheBooleanfunctionsandthe ConstraintBuilder::query_bool() functionwouldallowdeveloperstoknow thatthosevaluesareBoolean-constrained.Thismechanismalsocanimprove efficiencybypreventingpotentialdouble-enforcementofthesameconstraint. Thefollowingtablesprovidethenumberoffindingsbyseverityandcategory. EXPOSUREANALYSIS SeverityCount High8 Medium4 Low2 Informational13 CATEGORYBREAKDOWN CategoryCount Cryptography1 DataValidation25 Patching1 hasheye 6ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

ProjectSummary ContactInformation Thefollowingmanagerswereassociatedwiththisproject: DanGuido,AccountManagerBrookeLanghorne,ProjectManager dan@hasheye.iobrooke.langhorne@hasheye.io Thefollowingengineerswereassociatedwiththisproject: FilipeCasal,ConsultantJoeDoyle,Consultant filipe.casal@hasheye.iojoseph.doyle@hasheye.io OpalWright,ConsultantWillSong,Consultant opal.wright@hasheye.iowill.song@hasheye.io ProjectTimeline Thesignificanteventsandmilestonesoftheprojectarelistedbelow. DateEvent April17,2023Pre-projectkickoffcall April21,2023Statusupdatemeeting#1 April28,2023Statusupdatemeeting#2 May5,2023Statusupdatemeeting#3 May12,2023Statusupdatemeeting#4 May19,2023Statusupdatemeeting#5 May26,2023Statusupdatemeeting#6 June2,2023Statusupdatemeeting#7 June9,2023Statusupdatemeeting#8 June16,2023Statusupdatemeeting#9 June26,2023Deliveryofreportdraftandreportreadoutmeeting October12,2023Deliveryofcomprehensivereportwithfixreviewappendix hasheye 7ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

ProjectGoals TheengagementwasscopedtoprovideasecurityassessmentoftheˆScrollzkEVMhalo2 circuits.Specifically,wesoughttoanswerthefollowingnon-exhaustivelistofquestions: • ArethecircuitsimplementingtheEVMopcodesproperlyenforcingtheEVM semantics? Arethecircuitssoundandcomplete? • AreEVMstatesemanticsproperlyenforced?Arestate-changingopcodesallowed withinastaticcontext? • Arethefixedlookuptablesproperlypopulated? • Arethelookuptablesproperlyconstrainedtopreventmaliciousinsertions? • IstheEVMexecutioncontrolflowcorrectlyenforced?Canamaliciousprover providedifferentexecutiontracesforthesameState-Transactioninput? • Canamaliciousproverconvinceaverifierofadifferentfinalstatethanwhat runningtheEVMwouldreach? • DoestheconstraintbuilderAPIcorrectlyenforcetheintendedconstraints? • Howareopcode-to-opcodetransitionsenforced?Howareopcode-to-error transitionsenforced? • Couldanattackertraceleadtoanerrorstatewhenthecorrectexecutionshouldnot error? Couldtheoppositealsooccur? • Arememorysemanticscorrectlyguaranteedbytheˆ RWtableconstraints? • Dothecodechangesto halo2-lib , snark-verifier ,ortheKeccakcircuit introduceanyvulnerabilities? • Dothecodechangesimprovereadabilityandcodemaintainability? • ForacorrectFiat-Shamirtransformation,arethestatementandallprovermessages includedinthetranscript? hasheye 8ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

ProjectTargets Theengagementinvolvedareviewandtestingofthefollowingtargets. zkevm-circuits Repositoryscroll-tech/zkevm-circuits Version e8bcb23e1f303bd6e0dc52924b0ed85710b8a016 TypesRust,halo2 PlatformNative snark-verifiercodedi Repositoryscroll-tech/snark-verifier Version a3d0a5ab48522bc533686da3ea8400282c91f536 TypesRust,halo2 PlatformNative modexp Repositoryscroll-tech/misc-precompiled-circuit Version 05725ec61d52d29a063395b0a1130467bee0d2f1 TypesRust,halo2 PlatformNative halo2-libcodedi Repositoryscroll-tech/halo2-lib Version a805052→b1d1567 TypesRust,halo2 PlatformNative Bytecodecircuit Repositoryscroll-tech/zkevm-circuits/src/bytecode_circuit Version 44000e55eddaec42da958f2555d9bdeec8b865c2 TypeRust,halo2 PlatformNative hasheye 9ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

ProjectCoverage Thissectionprovidesanoverviewoftheanalysiscoverageofthereview,asdeterminedby ourhigh-levelengagementgoals.Ourapproachesincludedthefollowing: • WemanuallyreviewedthezkEVMcircuit,includingthemathematicalgadgets, utilities,andallEVMopcodegadgets.Wecheckedforcircuitsoundnessissues, opcodeedgecases,underconstrainedwitnessvalues,andcorrectstatetransition enforcement.Ifweidentifiedvulnerablecodepatternsorcodesmells,weused Semgreptoperformvariantanalysisandsearchforotherinstancesofthesame patterns. • WemanuallyreviewedtheRWtablecircuits,especiallyfocusedonlookups performedinexecutiongadgetsinthe zkevm-circuits/src/evm_circuit/execution/ directoryandthestructural constraintsinthe zkevm-circuits/src/state_circuit/ directory. • WemanuallyreviewedtheBytecodetableconsistencycircuit,includingitsPoseidon extendedcolumnvariant.Wecheckedthecorrectenforcementofthespecified constraints.Welookedforpotentialwaystobreakthesoundnessofthecircuit. •

WemanuallyreviewedtheKeccakcircuitcode,particularlythefilesinthe zkevm-circuits/src/keccak_circuit/ directory. • Wemanuallyreviewedthe halo2-lib and snark-verifier diffs,payingspecial attentiontochangesinthe snark-verifier/src/pcs.rs file,aswellasthe snark-verifier/src/pcs/ directory. • Wemanuallyreviewedthe modexp precompilecircuit.Atthetimeoftheaudit,the circuitwasstillbeingdevelopedandwasincompleteintermsoffeaturesand engineeringwork;thecurrentversionofthecircuitdoesnotsupportarbitrary lengthvalueslikeTheEVM modexp precompile,andthereareseveralempty functions,whichimpactthesoundnessandcompletenessofthecircuit. hasheye 10ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

AutomatedTesting hasheyeusesautomatedtechniquestoextensivelytestthesecuritypropertiesof software.Weusebothopen-sourcestaticanalysisandfuzzingutilities,alongwithtools developedin-house,toperformautomatedtestingofsourcecodeandcompiledsoftware. TestHarnessConfiguration Weusedthefollowingtoolsintheautomatedtestingphaseofthisproject: ToolDescriptionPolicy SemgrepAnopen-sourcestaticanalysistoolforfindingbugsand enforcingcodestandardswheneditingorcommittingcode andduringbuildtime AppendixD ClippyAnopen-sourceRustlinterusedtocatchcommonmistakes andunidiomaticRustcode AppendixD AreasofFocus Ourautomatedtestingandverificationfocusedonthefollowing: • Identificationofgeneralcodequalityissuesandunidiomaticcodepatterns • Identificationofdangeroushalo2-specificandScroll’sAPIpatterns TestResults Theresultsofthisfocusedtestingaredetailedbelow. Clippy • zkEVM-circuits:ThezkEMV-circuitscodebasehasseveralinformationalClippy warnings: uninlined_format_args and unnecessarycast warnings. • Modexpprecompilecircuit:The modexp precompilecircuithasseveralClippy warningsthatshouldbeaddressed. WerecommendthatScrolladdaClippyGitHubactiontoallofitsRustrepositories. Semgrep Wepresentsomeoftherulesthatwewrotetofindhalo2-specificandScroll’sAPIpatterns inappendixD. hasheye 11ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

CodebaseMaturityEvaluation hasheyeusesatraffic-lightprotocoltoprovideeachclientwithaclearunderstandingof theareasinwhichitscodebaseismature,immature,orunderdeveloped.Deficiencies identifiedhereoftenstemfromrootcauseswithinthesoftwaredevelopmentlifecyclethat shouldbeaddressedthroughstandardizationmeasures(e.g.,theuseofcommonlibraries, functions,orframeworks)ortrainingandawarenessprograms. CategorySummaryResult ArithmeticWefoundnoissuesrelatedtotheuseofinteger arithmeticinthecodebase. Satisfactory Complexity Management Thecodebaseiswellorganizedandseparatedintofiles andfolders.Specificgadgetimplementationsarealso cleanlyimplementedandseparatetheconstraint generationfromthewitnessgeneration.Thisseparation allowsbetterscrutinyoftheconstraintsimposedineach circuit. Ontheotherhand,theexecutionofthezkEVMdepends onnondeterministicprogrammingpatterns,whichare hardtoreasonaboutandtohaveaglobalunderstanding of.Wefoundtwoinstanceswhereamaliciousprover couldhijacktheexecutionflowoftheexecuting bytecode:TOB-SCROLL-13andTOB-SCROLL-14. Moderate Cryptography andKey Management Wefoundnoissuesrelatedtotheuseofcryptography primitivesinthecodebase. However,thecodebasedependsuponanoutdated versionofthe halo2-ecc library,whichhashadseveral updatesmadetoitscryptographicprimitives,as describedinTOB-SCROLL-4.Allusesof halo2-ecc cryptographicprimitivesshouldbecarefullyreviewed, andScrollshoulduseanup-to-dateandwell-tested versionofthatlibrary. Satisfactory DocumentationThereareseveralsourcesofdocumentationinsteadof onecentralizedandup-to-datedocumentation.Itis commontofindmissingorincompletesectionsof documentation,whichshouldbeaddressed.Wealso Moderate hasheye 12ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

foundthatwhilesomedocumentationhasthegeneral designdescription,itfrequentlylackscorrectness requirements. MemorySafety andError Handling The zkevm-circuits projectusesnounsafeRustcode. Wefoundoneruntimepanicduringwitnessgeneration: TOB-SCROLL-26. Satisfactory Testingand Verification Thecodebaseusesgethtracingtoobtainvaluesfor witnessgeneration.Byusinggethasgroundtruth,tests willmostlyexercisethecompletenessaspectofthe circuits. Aswefoundmanyhigh-severityfindingsrelatedtocircuit soundness(TOB-SCROLL-1,TOB-SCROLL-3, TOB-SCROLL-7,TOB-SCROLL-9,TOB-SCROLL-12, TOB-SCROLL-13,TOB-SCROLL-14),webelieveitis necessarytodevelopanadversarialtestingprocess, especiallyfocusedonmaliciousproverbehavior.Formal methodstocheckfor,suchascircuitdeterminacy,should alsobeconsideredforresearch. Wealsofoundgadgetswithouttests(e.g., binary_number.rs , rlp.rs)andignoredtests(e.g.,in jump.rs).Testsshouldbereenabledandaddedwhere therearenone.Wealsorecommendaddingthetest vectorsforallEIPsthatthecodebaseimplements.Asan example,the SSTORE gasrefundEIPisimplementedin thecodebase,butitstestvectorsarenotpresentinthe codebasetoensureacorrectimplementation. Weak hasheye 13ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

SummaryofFindings Thetablebelowsummarizesthefindingsofthereview,includingtypeandseveritydetails.
IDTitleTypeSeverity 1ModGadgetisunderconstrainedandallows incorrectMULMODoperationstobeproven
DataValidationHigh 2TheRlpU64Gadgetisunderconstrainedwhen is_lt_128isfalse DataValidationHigh
3TheBLOCKHASHopcodeisunderconstrainedand allowsthehashofanyblocktobecomputed DataValidationHigh
4zkevm-circuitscratedependsonanoutdated versionofhalo2-ecc PatchingMedium
5N_BYTESparametersarenotcheckedtoprevent overflow DataValidationInformational
6Differencesinsharedcodebetween zkevm-circuitsandhalo2-lib DataValidationMedium
7UnderconstrainedwarmstatusonCALLopcodes allowsgascostforgery DataValidationHigh
8RWtableconstantsmustmatchexactlywhenthe verificationkeyiscreated DataValidationInformational
9TheCREATEandCREATE2opcodescanbecalled withinastaticcontext DataValidationHigh
10ResponsibleOpcodetableincorrectlyhandles CREATEandCREATE2 DataValidationInformational
11Ellipticcurveparametersomittedfrom Fiat-Shamir CryptographyInformational hasheye
14ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

12ThegascostfortheCALLopcodeis underconstrained DataValidationHigh
13Unconstrainedopcodesallownondeterministic execution DataValidationHigh
14Nondeterministicexecutionof ReturnDataCopyGadgetand ErrorReturnDataOutOfBoundGadget
DataValidationHigh 15ManyRWcounterupdatesaremagicnumbersDataValidationInformational
16NativePCSaccumulationdecidersacceptan emptyvector DataValidationMedium
17TheErrorOOGSloadSstoreandtheErrorOOGLog gadgetshaveredundanttablelookups
DataValidationInformational 18TheStatecircuitdoesnotenforcetransaction receiptconstraints
DataValidationInformational 20TheEXPopcodehasanunusedwitnessDataValidationInformational
21Thebn_to_fieldfunctionsilentlytruncatesbig integers DataValidationLow
22Thefield_to_bnfunctiondependson implementation-specificdetailsoftheunderlying field
DataValidationLow 23Thevaluesofthebytecodetabletagcolumnare notconstrainedtobeHEADERorBYTE
DataValidationInformational 24Unconstrainedcolumnsonthebytecode HEADER rows
DataValidationInformational 25decompose_limbdoesnotworkasintendedDataValidationInformational
hasheye 15ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

26ZeromoduluswillcauseapanicDataValidationMedium
27TheConstraintBuilder::conditionAPIisdangerousDataValidationInformational
28TheEXTCODECOPYopcodeimplementationdoes notworkwhentheaccountaddressdoesnot exist
DataValidationInformational hasheye 16ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

DetailedFindings 1.ModGadgetisunderconstrainedandallowsincorrectMULMODoperations tobeproven
Severity:HighDifficulty:Medium Type:DataValidationFindingID:TOB-SCROLL-1 Target: zkevm-
circuits/src/evm_circuit/util/math_gadget/modulo.rs Description The ModGadget
circuitcomputesthemoduloperation, amodn ,withthecaveatthatthe resultshouldbe 0 whenever n is 0
.However,anincorrectconstraintallowsaproofthat that amod0=a .ThiscausesincorrectEVMsemanticsforthe
MULMOD opcode,allowingan attackertoprovethat a*bmod0=a*b .AccordingtotheEVMsemantics,thecorrect
resultis 0 . The ModGadget circuitimplementationusesawitnessvalue, a_or_zero ,thatissupposed
totakethevalueof a when n isnonzeroor 0 when n is 0 .Thecodecommentsindicatethat
thefollowingconstraintensuresthat a_or_zero satisfiesthiscondition,buttheconstraint
alsoallowsthecase a_or_zero=a and n=0 : ///Constraintsforthewordsa,n,r: ///amodn=r,ifn≠0
///r=0,ifn=0 /// ///Weusetheauxiliarya_or_zeroword,whosevalueisconstrainedtobe:
///a_or_zero=aifn≠0,0ifn=0.Thisallowstousetheequation
///k*n+r=a_or_zerotoverifythemodulus,whichholdswithr=0inthe
///caseofn=0.Unliketheusualk*n+r=a,whichforcesr=awhenn=0, ///thisequationassuresthatr<norr=n=0. ...
impl<F:Field>ModGadget<F>{ pub(crate)fnconstruct(cb:&mutConstraintBuilder<F>,words:[&util::Word<F>;
3])→Self{ let(a,n,r)=(words[0],words[1],words[2]); letk=cb.query_word_rlc();
leta_or_zero=cb.query_word_rlc(); letn_is_zero=IsZeroGadget::construct(cb,sum::expr(&n.cells));
leta_or_is_zero=IsZeroGadget::construct(cb,sum::expr(&a_or_zero.cells));
letmul_add_words=MulAddWordsGadget::construct(cb,[&k,n,r, &a_or_zero]);
leteq=IsEqualGadget::construct(cb,a.expr(),a_or_zero.expr()); hasheye
17ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

letlt=LtWordGadget::construct(cb,r,n); //Constraintheauxvariablea_or_zerotobe=aor=0ifn=0:
//(a=a_or_zero)^(n=0&a_or_zero=0) cb.add_constraint("(1-(a=a_or_zero))*(1-(n=0)*
(a_or_zero=0)", (1.expr()-eq.expr())*(1.expr()-n_is_zero.expr()* a_or_is_zero.expr()),);
Figure1.1: evm_circuit/util/math_gadget/modulo.rs#L10-L44 Tocorrectlyconstrainthe a_or_zero
variable,rewritetheconstraintasthefollowing: [1-((n=0)*(a_or_zero=0)+(1-n=0)*
(a_or_zero=a)))]=0 Thisconstraintresultsinthefollowingtruthtable: Figure1.2showshow ModGadget
isusedtoconstraintheresultsofthe MULMOD opcode. Sincetheconstraintsaresatisfiedbysetting
a_reduced=a insteadof a_reduced=0 , when,theresultcanbesettobysetting. □·□<2 256 □·□□ 1 =□ 2 =□
=0, □=□=□·□ //1.k1*n+a_reduced=a letmodword=ModGadget::construct(cb,[&a,&n,&a_reduced]);
//2.a_reduced*b+0=d*2^256+e letmul512_left=MulAddWords512Gadget::construct(cb,

[&a_reduced,&b,&d,&e], None); //3.k2*n+r==d*2^256+e
letmul512_right=MulAddWords512Gadget::construct(cb,[&k,&n,&d,&e],Some(&r)); //(r<n)orn==0
letn_is_zero=IsZeroGadget::construct(cb,sum::expr(&n.cells));
letlt=LtWordGadget::construct(cb,&r,&n); cb.add_constraint("(1-(r<n)-(n==0))", 1.expr()-lt.expr()-
n_is_zero.expr(), hasheye 18ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

); Figure1.2: zkevm-circuits/src/evm_circuit/execution/mulmod.rs#58-73 ExploitScenario
AbridgeusestheScrollzkEVMtotrackthecurrentstateofEthereum.Amaliciousprover
generatesaproofofexecutionforatransactioninvolvingthe MULMOD instructionwith
,andsetstheresulttoasdescribedabove.Theproversubmitsthatproof,the □=0□·□
resultsofwhichwillnotmatchthecorrectEVMsemantics,leadingtostatedivergenceand lossoffunds.
Recommendations Shortterm,fixtheconstraint;extendthe assign functiontoreceivethe a_or_zero
witness.Addtestsforthisfinding.
Longterm,adddeterminacytestingtoanygadgetsthatconstrainnondeterministic witnesses. hasheye
19ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

2.TheRlpU64Gadgetisunderconstrainedwhenis_lt_128isfalse Severity:HighDifficulty:Medium
Type:DataValidationFindingID:TOB-SCROLL-2 Target:z kevm-
circuits/src/evm_circuit/util/math_gadget/rlp.rs Description The RlpU64Gadget
constrainswitnessvaluestomatchtheoutputofacorrectRLP encoding.SincethelengthandvalueoftheRLP-
encodedvaluedependonthevaluebeing lessthan128,the is_lt_128
flagispartofthewitness.Arangecheckensuresthatif is_lt_128
istrue,thenthevalueisactuallybelow128.However,thereisnoconstraint ensuringthat value isabove127when
is_lt_128 isfalse: letis_lt_128=cb.query_bool(); cb.condition(is_lt_128.expr(),|cb|{
cb.range_lookup(value,128); }); Figure2.1: evm_circuit/util/math_gadget/rlp.rs#L67-L70
Thismeansthatamaliciousprovercouldhaveavaluesmallerthan128butset is_lt_128
tofalse,leadingtoanincorrectlengthandRLP-encodedoutput. ExploitScenario
AmaliciousproverinterpretstwobytesinanRLP-serializeddatastructureasavalueless
than128,causinglaterfieldsinthedatastructuretobedeserializedstartingatanincorrect
offset.Theproverthenusesthisincorrectlydeserializeddatastructuretoproveaninvalid
statetransition,leadingtostatedivergenceandpotentiallossoffunds. Recommendations
Shortterm,addaconstrainttoensurethatthevalueisabove127when is_lt_128 is false.
Longterm,addnegativetestsensuringthatmismatchedwitnesses value and is_lt_128
donotsatisfythecircuitconstraints. hasheye 20ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

3.TheBLOCKHASHopcodeisunderconstrainedandallowsthehashofany blocktobecomputed
Severity:HighDifficulty:Medium Type:DataValidationFindingID:TOB-SCROLL-3 Target:z kevm-
circuits/src/evm_circuit/util/math_gadget/rlp.rs Description The BLOCKHASH
opcodereturnsthehashoftheblockidentifiedbythestackargument, block_number
,providedthatitisoneofthe256mostrecentcompleteblocks.However,
theimplementationallowsamaliciousprovertoprovideanonzeroresultevenwhenthe
providedblocknumberisnotamongthe256mostrecentblocks,contradictingtheEVM specification.
Tovalidatethat block_number isamongthe256mostrecentblocks,theimplementation checksthat
current_block_number-block_number<257 ,where current_block_number
issupposedtobetheblocknumberofthecurrentblock. However, current_block_number
isunconstrainedandcouldtakeanyvalue: impl<F:Field>ExecutionGadget<F>forBlockHashGadget<F>{
constNAME:&'staticstr="BLOCKHASH"; constEXECUTION_STATE:ExecutionState=ExecutionState::BLOCKHASH;
fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ letcurrent_block_number=cb.query_cell();
letblock_number=WordByteCapGadget::construct(cb, current_block_number.expr());
cb.stack_pop(block_number.original_word()); //FIXME //cb.block_lookup(
//BlockContextFieldTag::Number.expr(), //cb.curr.state.block_number.expr(),
//current_block_number.expr(), //); letblock_hash=cb.query_word_rlc();
letdiff_lt=LtGadget::construct(cb, current_block_number.expr(),
(NUM_PREV_BLOCK_ALLOWED+1).expr()+block_number.valid_value(), hasheye
21ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Figure3.1: zkevm-circuits/src/evm_circuit/execution/blockhash.rs#L33-L49
Amaliciousprovercouldprovideaninvalid current_block_number andreturnthehash
ofanyblockpresentintheblocklookuptable,independentofitsblocknumber. ExploitScenario
Amaliciousprovergeneratesaproofofexecutionforatransactioninvolvingthe BLOCKHASH
opcodethatresultsinanonzerohashforanolderblock.Theproversubmits
thatproof,theresultsofwhichwillnotmatchthecorrectEVMsemantics,leadingtostate
divergenceandlossoffunds. Recommendations Shortterm,addthemissinglookupconstraintforthe
current_block_number witness.
Longterm,trackandtriage“FIXME”and“TODO”itemsinacentralizedissuetracking

system,suchasGitHubissues.Addfailingtestswhensecurity-relevant“TODO”itemsare identified. hasheye 22ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

4.zkevm-circuitscratedependsonanoutdatedversionofhalo2-ecc Severity:MediumDifficulty:Medium Type:PatchingFindingID:TOB-SCROLL-4 Target:z kevm-circuits/{Cargo.toml,src/tx_circuit/sign_verify.rs} Description The zkevm-circuits cratedependsonthe halo2-ecc libraryinScroll’sforkof halo2-lib ,whichprovides halo2 circuitsforellipticcurveandfinitefieldoperations.As illustratedinfigure4.1,thiscratedependsonthe halo2-ecc-snark-verifier-0323 tag, whichcurrentlypointstocommit d24871338ade7dd56362de517b718ba14f3e7b90 . halo2-base={git="https://github.com/scroll-tech/halo2-lib",branch= "halo2-ecc-snark-verifier-0323",default-features=false, features=["halo2-pse","display"]} halo2-ecc={git="https://github.com/scroll-tech/halo2-lib",branch= "halo2-ecc-snark-verifier-0323",default-features=false, features=["halo2-pse","display"]} Figure4.1: zkevm-circuits/Cargo.toml#32–33 TheScrollforkof halo2-lib iscloselyrelatedtotheupstream halo2-lib library.In particular,the v0.3.0 versionof halo2-ecc (commit c31a30bcaff384b0c3aa7c823dd343f5c85da69e)hasacommonancestorcommitof 4338af81bb2de4f278467e5c484e067c064cc66b withtheScrollversion. Figure4.2:TheGitcommithistoryof halo2-lib withthecommonancestorofthe minimize-diff and upgrade-v0.3.0 brancheshighlighted Theupstreamlibraryhasvariousfixesandimprovementsthatshouldbeincorporated. Somenotableexistingfixesincludethefollowing: • FpChip::assert_equal hashadasoundness-relatedtypofixed(PR#18). hasheye 23ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

• ecdsa_verify_no_pubkey_check nolongerrejectscertainvalidsignatures(PR #36). While FpChip::assert_equal doesnotcurrentlyappeartobeused,the SignVerifyChip circuitusesthe ecdsa_verify_no_pubkey_check function,asshown infigure4.3.
//returnstheverificationresultofecdsasignature //
//WARNING:thiscircuitdoesnotenforcethereturnedvaluetobetrue //makesurethecallerchecksthisresult!
letecdsa_is_valid=ecdsa_verify_no_pubkey_check::<F,Fp,Fq,Secp256k1Affine>(&ecc_chip.field_chip, ctx, &pk_assigned, &integer_r, &integer_s, &msg_hash, 4, 4,); Figure4.3: zkevm-circuits/src/tx_circuit/sign_verify.rs#386–399 SignVerify isthenusedtochecksignaturesonEVMtransactions,asshowninfigure4.4, andbecauseofthepre-patchbehavior,anadversarycangenerateacorrectlysigned transactionthatwillneverthelessfailsignatureverification. #[cfg(feature="enable-sign-verify")] {
letassigned_sig_verifs= self.sign_verify
.assign(&config.sign_verify,layouter,&sign_datas,challenges)?;
self.sign_verify.assert_sig_is_valid(&config.sign_verify, layouter, assigned_sig_verifs.as_slice(),)?; self.assign(config, challenges, layouter, assigned_sig_verifs, Vec::new(), &padding_txs,)?; } Figure4.4: zkevm-circuits/src/tx_circuit.rs#1804–1822 hasheye 24ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

ExploitScenario Anadversarycreatesatransactionwithavalidsignaturethattheoldimplementation wouldrejectandsubmitsittoEthereum.Ethereumacceptsthetransaction,buttheScroll zkEVMisunabletoacceptit,stallingthezkEVMandcreatingadenialofservicethatmay freezeuserfunds. Recommendations Shortterm,reviewthesecurityimplicationsofthisoutdatedversionof halo2-ecc onthe zkEVMcodebase.Then,eitherupdatetoamorerecentversionof halo2-lib that incorporatesupstreamfixesorbackportthosefixestoScroll’sfork.
Longterm,keepalldependenciesuptodatewheneverpossible.Foranydependenciesthat havebeenforkedfromtheupstreamversion,developaplantoportanyupstreamsecurity updatesontothatfork. hasheye 25ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

5.N_BYTESparametersarenotcheckedtopreventoverflow Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-5 Target: z kevm-circuits/src/evm_circuit/util/math_gadget/{constant_division,l t}.rs Description The ConstantDivisionGadget and LtGadget circuitsimplementoperationson multi-byteintegers:divisionbyaconstantvalueandcomparison,respectively.Eachcircuit hasagenericparameter N_BYTES representingthenumberofbytesused.However,each ofthesecircuitshasadditionalimpliedrestrictionson N_BYTES requiredtoprevent unexpectedbehaviorduetooverflowingfieldelements. In ConstantDivisionGadget (showninfigure5.1),the quotient valueisconstrainedto belessthan,buttheexpressionquotient.expr()*denominator.expr() 256 □_□ □□□□ mayoverflowif denominator issufficientlylarge(e.g.,if denominator is 1024 and N_BYTES is 31 .Thecommenthighlightedinfigure5.2providessufficientconditionsto preventoverflow,butthesearenotfullyenforcedeitherinthecircuitorinassertionsat circuitconstructiontime.
letquotient=cb.query_cell_with_type(CellType::storage_for_expr(&numerator));
letremainder=cb.query_cell_with_type(CellType::storage_for_expr(&numerator));
//Requirethatremainder<denominator cb.range_lookup(remainder.expr(),denominator);

//Requirethatquotient<256**N_BYTES //sowecan'thaveanyoverflowwhendoing`quotient*denominator`. letquotient_range_check=RangeCheckGadget::construct(cb,quotient.expr()); //Checkifthedivisionwasdonecorrectly cb.require_equal("numerator-remainder==quotient⋅denominator", numerator-remainder.expr(), quotient.expr()*denominator.expr(),); Figure5.1: zkevm-circuits/src/evm_circuit/util/math_gadget/constant_division.rs#33- 48 hasheye 26ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

///Returns(quotient:numerator/denominator,remainder:numerator%denominator), ///with`numerator`anexpressionand`denominator`aconstant. ///Inputrequirements: ///-`quotient<256**N_BYTES` ///-`quotient*denominator<fieldsize` ///-`remainder<denominator`requiresarangelookuptablefor`denominator` #[derive(Clone,Debug)] pubstructConstantDivisionGadget<F,constN_BYTES:usize>{ Figure5.2: zkevm-circuits/src/evm_circuit/util/math_gadget/constant_division.rs#13- 20 InLtGadget(showninfigure5.3),valuesofN_BYTESabove31willcauselttobean unconstrainedBoolean,sinceamaliciousprovercanset diff totherepresentationof (rhs-lhs) evenif rhs<lhs .Thecommenthighlightedinfigure5.4describessufficient conditionstopreventoverflowwithoutchangestothecircuit,buttheserestrictionsare enforcedonlybya debug_assert! in from_bytes::expr (showninfigure5.5). letlt=cb.query_bool(); letdiff=cb.query_bytes(); letrange=pow_of_two(N_BYTES*8); //Theequationwerequiretohold:`lhs-rhs==diff-(lt*range)`. cb.require_equal("lhs-rhs==diff-(lt⋅range)", lhs-rhs, from_bytes::expr(&diff)-(lt.expr()*range),); Figure5.3: zkevm-circuits/src/evm_circuit/util/math_gadget/lt.rs#37-46
///Returns`1`when`lhs<rhs`,andreturns`0`otherwise. ///lhsandrhs`<256**N_BYTES` ///`N_BYTES`isrequiredtobe`⩽MAX_N_BYTES_INTEGER`topreventoverflow: ///valuesarestoredinasinglefieldelementandtwoofthesareadded ///together. ///Theequationthatisenforcedis`lhs-rhs==diff-(lt*range)`. ///Becauseallvaluesare`⩽256**N_BYTES`and`lt`isboolean,`lt`canonly ///be`1`when`lhs<rhs`. #[derive(Clone,Debug)] pubstructLtGadget<F,constN_BYTES:usize>{ Figure5.4: zkevm-circuits/src/evm_circuit/util/math_gadget/lt.rs#14-23 pub(crate)fnexpr<F:FieldExt,E:Expr<F>>(bytes:&[E])→Expression<F>{ debug_assert!(bytes.len()⩽MAX_N_BYTES_INTEGER, "Toomanybytestocomposeanintegerinfield"); hasheye 27ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Figure5.5: zkevm-circuits/src/evm_circuit/util.rs#528-532 ExploitScenario Adeveloperwhoisunawareoftheseissuesusesthe ConstantDivisionGadget or LtGadget circuitwithvaluesof N_BYTES thataretoolarge,causingpotentially underconstrainedcircuits. Recommendations Shortterm,addexplicitchecksatcircuitconstructiontimetoensurethat N_BYTES is limitedtovaluesthatpreventoverflow. Longterm,considerperformingthesevalidationsatcompiletimewith static_assertions orassertsina const context. hasheye 28ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

6.Dierencesinsharedcodebetweenzkevm-circuitsandhaIo2-lib Severity:MediumDifficulty:High Type:DataValidationFindingID:TOB-SCROLL-6 Target:Severalfiles Description Thecodebasecontainscodethatisalsopresentinthe halo2-lib codebase(notthrougha dependency)anditdoesnotmatchinallcases.Forexample,theseveral constraint_builder functionsusethe debug_assert!() macroforimportant validations,whichwillnotperformthosechecksinreleasemode. pub(crate)fncondition<R>(&mutself, condition:Expression<F>, constraint:implFnOnce(&mutSelf)→R,)->R{ debug_assert!(self.condition.is_none(), "Nestedconditionisnotsupported"); self.condition=Some(condition); letret=constraint(self); self.condition=None; ret } Figure6.1: evm_circuit/util/constraint_builder.rs#L216-L229 pub(crate)fnvalidate_degree(&self,degree:usize,name:&'staticstr){ ifself.max_degree>0{ debug_assert!(degree⩽self.max_degree, "Expression{}degreetoohigh:{}>{}", name, degree, self.max_degree,); } } Figure6.2: evm_circuit/util/constraint_builder.rs#L246-L256 pub(crate)fnvalidate_degree(&self,degree:usize,name:&'staticstr){ hasheye 29ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

//WeneedtosubtractIMPLICIT_DEGREEfromMAX_DEGREEbecauseallexpressions //willbemultipliedbystateselectorandq_step/q_step_first //selector. debug_assert!(degree⩽MAX_DEGREE-IMPLICIT_DEGREE, "Expression{}degreetoohigh:{}>{}", name, degree, MAX_DEGREE-IMPLICIT_DEGREE,); } Figure6.3: evm_circuit/util/constraint_builder.rs#L1370-L1381 Thecodebasealsoincludesthe log2_ceil functionin zkevm-circuits/src/util.rs , whichmiscomputesitsresultonazeroinput—thebehaviorhasbeenfixedinPR#37for halo2-lib . Recommendations Shortterm,fixtheissuesincommonwiththe halo2-lib codebase.Also,checkallusesof debug_assert throughoutthecodebaseandensurethattheyarenotusedtovalidate criticalinvariants,astheywillnotruninreleasemode.

Longterm,minimizeduplicatecodebyrefactoringtheconstraintbuildercodebase. hasheye 30ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

7.UnderconstrainedwarmstatusonCALLopcodesallowsgascostforgery Severity:HighDifficulty:Medium Type:DataValidationFindingID:TOB-SCROLL-7 Target:z kevm-circuits/src/evm_circuit/execution/callop.rs Description Anunderconstrainedvariableinthe CallOpGadget allowsanattackertoprovethe executionofatransactionwithincorrectgascostsbysettinganaddressascoldwhenit shouldbecomewarm. The CallOpGadget implementsthe CALL , CALLCODE , DELEGATECALL ,and STATICCALL EVMopcodes.Thegascostoftheseopcodesdependsonwhetherthecalleeaddressis warm.Additionally,theimplementationoftheseopcodesmustmaketheaddresswarmso thatfuturecallstothesameaddresscostlessgas.However,thevariablethatcontrolsthe address’snewwarmstatusisnotconstrainedandisreferencedonlyinthewritetotheRW table: //Addcalleetoaccesslist letis_warm=cb.query_bool(); letis_warm_prev=cb.query_bool(); cb.account_access_list_write(tx_id.expr(), call_gadget.callee_address_expr(), is_warm.expr(), is_warm_prev.expr(), Some(&mutreversion_info),); Figure7.1: zkevm-circuits/src/evm_circuit/execution/callop.rs#L129-L138 Thismeansthatamaliciousprovercanmakethe is_warm variableequal false ,causinga calledaddresstoactuallybecomecoldduringtheexecutionofa CALL ,insteadofwarmas intheEVMspecification. AconstraintontheRWtable,requiringthattheinitialvalueoftheaccesslistelementsis alwaysfalse,preventsanotherpossiblescenariowherethe is_warm_prev valuecouldbe definedaswarmeventhoughtheaddresshadnotbeenaccessedbefore. ExploitScenario Amaliciousprovergeneratesaproofofexecutionforatransactioninvolvingtwo CALL opcodestothesameaddressthatresultsindifferentgascostsfromtheEVMspecification: hasheye 31ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

inthefirst CALL opcodeexecution,theproversetstheaddressascoldinsteadofwarm, causingthewronggascalculationforthesecondcall.Theprovesubmitsthatproof,the resultsofwhichwillnotmatchthecorrectEVMsemantics,leadingtostatedivergenceand lossoffunds. Recommendations Shortterm,addconstraintstoensurethatthecalleeaddressbecomeswarmonthe CALL opcodes,byconstraining is_warm tobetrue. hasheye 32ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

8.RWtableconstantsmustmatchexactlywhentheverificationkeyis created Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-8 Target:RWtable Description NearlyallruntimestateofEVMprogramexecutionistrackedandvalidatedinalookup tablereferredtoasthe“RWtable.”Thistableenforcescorrectinitializationandcoherency ofreadandwriteoperationsforaddressablepartsofthestate,includingthestack, memory,andaccountstorage,aswellasinputsandoutputssuchasthetransactionaccess listandthetransactionlog.Figure8.1showsthestoragetypescombinedinthistable: pubenumRwTableTag{ ///Start(usedforpadding) Start=1, ///Stackoperation Stack, ///Memoryoperation Memory, ///AccountStorageoperation AccountStorage, ///TxAccessListAccountoperation TxAccessListAccount, ///TxAccessListAccountStorageoperation TxAccessListAccountStorage, ///TxRefundoperation TxRefund, ///Accountoperation Account, ///CallContextoperation CallContext, ///TxLogoperation TxLog, ///TxReceiptoperation TxReceipt, } Figure8.1: zkevm-circuits/src/table.rs#354-377 ThezkEVMcircuitenforcescorrectmemoryoperationresultsforEVMopcodesby performinglookupsintothistable,asshowninfigure8.2.Callssuchas memory_lookup aretranslatedinto Lookup::Rw values(showninfigure8.3),whicharethenfurther hasheye 33ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

translatedintomulti-columnlookupsintotheRWtable,asillustratedinfigure8.4.Note thatinlookups,thefirstcolumn,correspondingtothefixedcolumn q_enable ,isalways setto1. cb.condition(is_mstore8.expr(),|cb|{ cb.memory_lookup(1.expr(), from_bytes::expr(&address.cells), value.cells[0].expr(), None,); }); Figure8.2:Amemorylookup (zkevm-circuits/src/evm_circuit/execution/memory.rs#73-80) Rw{ ///Counterforhowmuchread-writehavebeendone,whichstandsfor ///thesequentialtimestamp. counter:Expression<F>, ///Abooleanvaluetospecifyiftheaccessrecordisareadorwrite. is_write:Expression<F>, ///Tagtospecifywhichread-writedatatoaccess,seeRwTableTagfor ///alltags. tag:Expression<F>, ///Valuescorrespondingtothetag. values:RwValues<F>, }, Figure8.3: Lookup::Rw (zkevm-circuits/src/evm_circuit/table.rs#197-208) Self::Rw{ counter, is_write, tag, values, }⇒{ vec![1.expr(), counter.clone(), is_write.clone(), tag.clone(), values.id.clone(), values.address.clone(), values.field_tag.clone(), values.storage_key.clone(), values.value.clone(), values.value_prev.clone(), values.aux1.clone(), values.aux2.clone(),] hasheye 34ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

} Figure8.4:Conversiontolookupcolumns,with q_enable highlighted (zkevm-circuits/src/evm_circuit/table.rs#321-341)

However,theselookupsenforceonlytheexistenceofsuchrows,andforcorrectexecution, itisvitalthatthereadsandwritespresentinthetablearethefollowing: • Coherentwiththeexternalstate:Thefirstreadofanydataiscorrectlyinitialized, andthelastwriteofexternallyvisibledata(e.g.,storage)isreflectedintheEthereum statecommitment. • Coherentwitheachother:Valuesinreadoperationsmatchthemostrecentwritten valueortheinitialvalue. • Coherentwiththeexecutiontrace:EveryentryintheRWtablecorrespondsto exactlyonememory-access-generatingstepintheexecutiontrace.Equivalently, thereareno“extra”entriesinthetable. TheseglobalconstraintsontheRWtableareenforcedthroughthreemajorchecks. First,theRWtableislexicographicallyorderedwithrespecttoitscolumns.Several constraintsareusedtoenforcelexicographicordering.Anillustrativeexampleisshownin figure8.5.Notethattheconstraintisconditionalonthefixedcolumn selector .Allother lexicographicorderingconstraintsarealsoconditionalonthisfixedcolumn,soanyrows where selector==0 arenotrequiredtobeordered. meta.create_gate("limb_differenceisnotzero",|meta|{ letselector=meta.query_fixed(selector,Rotation::cur()); letlimb_difference=meta.query_advice(limb_difference,Rotation::cur()); letlimb_difference_inverse= meta.query_advice(limb_difference_inverse,Rotation::cur()); vec![selector*(1.expr()-limb_difference*limb_difference_inverse)] }); Figure8.5:Alexicographicorderingconstraint (zkevm-circuits/src/state_circuit/lexicographic_ordering.rs#128-134) Second,alargecollectionofstructuralpropertiesonthesortedtableareenforced.Figures 8.6and8.7showexamplesofsuchconstraints.Whentherowsaresorted,alloperations involvingthesameaddressorstorageidentifieraregroupedtogether,sortedinincreasing orderbythefinaltwocolumns,whichrepresentthevalue rw_counter inbig-endian.Note thatincreasingvaluesof rw_counter aretreatedas“happeninglaterintime,”asshownin thehighlightedportionoffigure8.6,whichenforcesthatreadsdonotchangethevalueby requiringthat value==value_prev inreadentries. hasheye 35ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Unlikeinotherpartsofthetable,theconstraintsappliedto Start rows(illustratedin figure8.7)usethe rw_counter fieldsforadifferentpurpose.Every Start rowwhere lexicographic_ordering_selector==1 isrequiredtoexactlyincrease rw_counter by 1 . Start rowsdonotrepresentmemoryoperations,andthuscanbethoughtofas padding. //Whenallthekeysinthecurrentrowandpreviousrowareequal. self.condition(q.not_first_access.clone(),|cb|{ cb.require_zero("non-firstaccessreadsdon'tchangevalue", q.is_read()*(q.rw_table.value.clone()-q.rw_table.value_prev.clone()),); cb.require_zero("initialvaluedoesn'tchangeinanaccessgroup", q.initial_value.clone()-q.initial_value_prev(),); }); Figure8.6:AstructuralconstraintontheRWtable (zkevm-circuits/src/state_circuit/constraint_builder.rs#177-187) self.require_zero("field_tagis0forStart",q.field_tag()); self.require_zero("addressis0forStart",q.rw_table.address.clone()); self.require_zero("idis0forStart",q.id()); self.require_zero("storage_keyis0forStart",q.rw_table.storage_key.clone()); //1.1.rw_counterincreasesby1foreverynon-firstrow self.require_zero("rw_counterincreasesby1foreverynon-firstrow", q.lexicographic_ordering_selector.clone()* (q.rw_counter_change()-1.expr()),); Figure8.7:Someconstraintsappliedto Start rows (zkevm-circuits/src/state_circuit/constraint_builder.rs#192-200) Third,arunningcountofRWlookupsistrackedinthe rw_counter fieldof StepState (showninfigure8.8)foreachstep.Whenexecutionreachesthe EndBlock state,two additionallookupsareperformed,showninfigure8.9.Theselookupsensurethatthereare max_rws-step.rw_counter paddingrowsintheRWtable,andaredesignedtocheck thatthereareatmost step.rw_counter non-paddingrowsinthetable. pub(crate)structStepState<F>{ ///Theexecutionstateselectorforthestep pub(crate)execution_state:DynamicSelectorHalf<F>, ///TheRead/Writecounter pub(crate)rw_counter:Cell<F>, Figure8.8: StepState andits rw_counter field (zkevm-circuits/src/evm_circuit/step.rs#456-460) hasheye 36ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

//3.Verifyrw_countercountstothesamenumberofmeaningfulrowsin //rw_tabletoensurethereisnomaliciousinsertion. //Verifythatthereareatmosttotal_rwsmeaningfulentriesintherw_table cb.rw_table_start_lookup(1.expr()); cb.rw_table_start_lookup(max_rws.expr()-total_rws.expr()); //SinceeverylookupdoneintheEVMcircuitmustsucceedanduses //auniquerw_counter,weknowthatatleastthereare //total_rwsmeaningfulentriesintherw_table. //Weconcludethatthenumberofmeaningfulentriesintherw_table //istotal_rws. Figure8.9:ConstraintsensuringthattheRWtablehasbeenpaddedto max_rws rows (zkevm-circuits/src/evm_circuit/execution/end_block.rs#78-87) Thesechecksaresufficienttoguaranteerwtablecorrectness,assumingthefollowing: 1.The rw_counter fieldof StepState correctlytrackshowmanydistinct,non-Start

RWlookupsareperformedintheexecutiontrace. 2. lexicographic_ordering_selector==1 whenever rw_table.q_enable==1 . 3. rw_table.q_enable isasequenceofall 1 sfollowedbyall 0 s. 4.Thereareatmost max_rws rowswhere rw_table.q_enable==1 . Ifanyoftheserequirementsisfalse,amaliciousprovercanproveerroneousexecution tracesbymanipulatingtheRWtableinsomeway: 1.Ifthe rw_counter overcountsthenumberofdistinctRWlookups,arow representingamaliciousmemorywritecanbeinserted. 2.If lexicographic_ordering_selector==0 inanycellwhere rw_table.q_enable==1 ,theprovermaybypassnearlyallstructuralproperty checksbypartitioningtheRWtableintotwo“versions,”onestartingatthebeginning ofthetableandonestartingatthatunrestrictedrow. 3.If rw_table.q_enable is 1 ,then 0 ,then 1 ,themiddlerowwillnotcorrespondto anyRWlookup,andthusmaybesettoamaliciousmemorywrite. 4.If rw_table.q_enable is 1 formorethanthan max_rws rows,amaliciousmemory writecanbeinserted. Thesefourassumptionsareallpropertiesoffixedrows,constants,orthecircuititself,so theydonotneedtobeconstrainedinthecircuit.However,theyarenotcurrentlyexplicitly enforcedatcircuit-constructiontime,soifanyofthemisviolatedwhengeneratingthe zkEVMverificationkey,thiswillgoundetectedandwouldleadtoglobalcircuit unsoundness. hasheye 37ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Unfortunately,thefirstisequivalentto“thereareno rw_counter -relatedbugsintheEVM circuit,”soitisdifficulttoenforce.However,therelationshipsbetween lexicographic_ordering_selector , rw_table.q_enable ,and max_rws canand shouldbecheckedautomaticallywithassertions. ExploitScenario Anincorrectversionofthezkevmcircuitisusedtogenerateaverificationkeythatfailsto enforceoneoftheassumptionsabove.Amaliciousproverthencraftsan RWtablethat leadstoincorrectexecutionofatransaction,causingstatedivergenceandpotentiallossof funds. Recommendations Shortterm,add assert!(...) callstoenforcecorrectcorrespondencebetween rw_table.q_enable , lexicographic_ordering_selector ,and max_rws . Longterm,reviewanddocumentassumptionsmadeaboutallcircuitconstants.When possible,usetechniquessuchasassertionstocheckthese assumptionsat circuit-constructiontime. hasheye 38ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

9.TheCREATEandCREATE2opcodescanbecalledwithinastaticcontext Severity:HighDifficulty:Medium Type:DataValidationFindingID:TOB-SCROLL-9 Target:z kevm-circuits/src/evm_circuit/execution/create.rs Description The CREATE and CREATE2 opcodesaremissingaconstraintthatpreventsthemfrombeing calledinthecontextofastaticcall.Thisallowsforastate-changingoperationthatisnot allowedbytheEVMspecification. Inthecontextofa STATICCALL ,thestatecannotbemodified.Asaresult,state-changing opcodeslike CREATE , CREATE2 , LOGX , SSTORE ,and CALL areforbiddenwhentheargument valuediffersfrom0,accordingtotheEVMspecification.However,thecurrent implementationofthe CREATE and CREATE2 opcodesdoesnothaveachecktoensurethat thecallingcontexthaspermissiontochangethestate.Bycontrast,theother implementationsofstate-changingopcodeshavethefollowingcheck: //constrainnotinstaticcall letis_static=cb.call_context(None,CallContextFieldTag::IsStatic); cb.require_zero("is_staticisfalse",is_static.expr()); Figure9.1: zkevm-circuits/src/evm_circuit/execution/sstore.rs#L57-L59 Withoutthisvalidationinplace,amaliciousprovercouldgenerateaproofofexecutionfor atransactioninvolvingthe CREATE opcodewithinthecontextofa STATICCALL ,leadingto statedivergence. Notethatthe SELFDESTRUCT opcodeisdisabled,butisalsosubjecttothenon-static constraintaccordingtotheEthereumYellowPaper.Thisshouldbetakenintoaccountifthe opcodeisimplementedinthefuture. ExploitScenario Alicedeploysaconstant-functionautomatedmarketmaker(AMM)smartcontract AliceMM totheScrollzkEVM.IneachAMMtransaction, AliceMM receivesfundsintokentypeA(or B),thencalculatestheexchangerate,thensendsfundsintokentypeB(orA).Tocalculate theexchangerate, AliceMM callsBob’s ComplicatedMath contract.Aliceknowsabout reentrancyattacksandiscarefultocall ComplicatedMath onlywith STATICCALL . However,Bobhasdeployedamaliciousversionof ComplicatedMath thatuses CREATE to call AliceMM inareentrantfashion.Bobcalls AliceMM withamalicioustransactionthat hasheye 39ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

manipulatestheexchangerate,thendrainsthecontractoftokenAinexchangeforatiny amountoftokenB,resultinginlossoffunds. Recommendations Shortterm,addtheconstrainttovalidatethattheexecutioncontextdoesnotallow state-changingoperations. Longterm,addtestsforthe CREATE , CREATE2 , LOGX , SSTORE ,and CALL opcodeswhen calledwithina STATICCALL . hasheye 40ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

10.ResponsibleOpcodetableincorrectlyhandlesCREATEandCREATE2 Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-10 Target:z kevm-circuits/src/evm_circuit/step.rs Description The ResponsibleOpcode tableisusedtoattributedifferentexecutionstatestoparticular

setsofopcodes.Formanyopcodes,thistableistheprimarysourceoftruthforwhichstate theytransitionto.The SameContextGadget (showninfigure10.1)enforcesthatexecuting opcodescorrectlyusethecorrespondingstate.Forexample,itenforcesthatthe ADD opcodeusesthe ADD_SUB state. cb.add_lookup("Responsibleopcodelookup", Lookup::Fixed{ tag:FixedTableTag::ResponsibleOpcode.expr(), values:[cb.execution_state().as_u64().expr(), opcode.expr(), 0.expr(),], },); Figure10.1: zkevm-circuits/src/evm_circuit/util/common_gadget.rs#48-58 Thistableispopulatedviathe ExecutionState::responsible_opcodes method,which alsoisusedforreportingexecutionstatistics.Thismethoddoesnothandlethe CREATE2 state,andincorrectlyreportsboth CREATE and CREATE2 astheresponsibleopcodesfor the CREATE state,asshowninfigure10.2: Self::CREATE⇒vec![OpcodeId::CREATE,OpcodeId::CREATE2], Figure10.2: zkevm-circuits/src/evm_circuit/step.rs#304 Sincethe CREATE and CREATE2 opcodesconstraintheexecutionstateinawaythatdoes notuse SameContextGadget ,thisdoesnotcauseanysoundnessissues.However,ifa similarerrorweremadeforanotheropcodeorstateinthetable,theresultingcircuitmay beeitherincompleteorunderconstrained. hasheye 41ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Recommendations Shortterm,fixthedatainthistablebycorrectlymappingthe CREATE and CREATE2 states tothe CREATE and CREATE2 opcodes,respectively. Longterm,developteststocheckttheconsistencyoftheopcodetableagainstthe executionbehavior. hasheye 42ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

11.EllipticcurveparametersomittedfromFiat-Shamir Severity:InformationalDifficulty:N/A Type:CryptographyFindingID:TOB-SCROLL-11 Target:Severalfiles Description TheFiat-Shamircodeinthe snark-verifier patchdoesnotincorporatetheellipticcurve parametersintothetranscript.Pointsareincorporatedintothetranscriptusingonlythe x and y coordinates,withnoreferencetotheassociatedcurve,andwearenotabletofind anyinstanceswherethecurveparametersareexplicitlyaddedtoaFiat-Shamirtranscript. fncommon_ec_point(&mutself,ec_point:&C)→Result<(),Error>{ letcoordinates= Option:: <Coordinates<C>>::from(ec_point.coordinates()).ok_or_else(||{ Error::Transcript(io::ErrorKind::Other, "Cannotwritepointsatinfinitytothetranscript".to_string(),) })?; [coordinates.x(),coordinates.y()].map(|coordinate|{ self.buf.extend(coordinate.to_repr().as_ref().iter().rev().cloned()); }); Ok(()) } Figure11.1: snark-verifier/src/system/halo2/transcript/evm.rs#L173-L187 Non-interactiveproofsmustcommitexactlytothestatementbeingprovenbeforeany challengesaregenerated.Ifaprovercanequivocateaboutattributesofthestatement(e.g., whichellipticcurvethepointsaresupposedtobeon),aproofforonestatementmaybe passedoffasaproofforanother,asintheFrozenHeartclassofvulnerabilities.(Notethat theFrozenHeartPlonKvulnerabilitydiscussedinthelinkedarticleisnotunder considerationhere;itisonlyanillustrationofFiat-Shamirvulnerabilities.) The snark-verifier codeisintendedtobecurve-agnostic,soaproofgeneratedusing onecurvemaybeverifiedusingadifferentellipticcurvethatsharesonlythepointspresent inthetranscript,leadingtoidenticalchallengevaluesbutadifferentstatement. Ingeneral,twodifferentellipticcurvescanshareonlyalimitednumberofpoints,sothe existingcodemayimplicitlycommittothecurvebeingused.However,wehavenot hasheye 43ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

determinedtheexactthreshold,andadetailedsecurityproofshouldbedoneifthat propertyisreliedupon. IntheScrollzkEVMsystem,theproverandverifieruseafixedsetofcurveparameters,soit isnotpossibletoconvinceScrollsoftwaretoacceptaproofusinganothercurve. Recommendations Shortterm,includethecurveparametersatthebeginningoftheFiat-Shamirtranscript. Longterm,alwaysconsiderincludingallpublicparametersofthesystemintheFiat-Shamir transformations. hasheye 44ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

12.ThegascostfortheCALLopcodeisunderconstrained Severity:HighDifficulty:Medium Type:DataValidationFindingID:TOB-SCROLL-12 Target:z kevm-circuits/src/evm_circuit/execution/callop.rs Description Thegascostofthe CALL -likeopcodes(CALL , CALLCODE , DELEGATECALL ,and STATICCALL) isnotconstrained,allowingamaliciousprovertospendasmuchgasasdesiredincertain conditions.Thisallowsfreegas CALL operationsiftheprovesetsthisvaluetozero,orit cancausethetransactionexecutiontoterminateaftertheexecutionofthecurrentopcode bydefiningahighgascost.Bothoptionscouldcauseastatedivergencefromanexecution followingtheEVMspecification. Figure12.1showsthecodethatgetsthewitnesscell step_gas_cost andthenusesit unconstrainedtosetthegascostofthecurrentopcode.Thishappenswhenthecall precheckconditionsarevalid(i.e.,thecalldepthisvalid,andthecallerbalanceisenoughto transferthecallvalue),andthecalledaddresshasnoassociatedcode: letstep_gas_cost=cb.query_cell();

letmemory_expansion=call_gadget.memory_expansion.clone(); cb.condition(and::expr([no_callee_code.expr(), not::expr(is_precompile.expr()), is_precheck_ok.expr(),]), |cb|{ //Savecaller'scallstate forfield_tagin[CallContextFieldTag::LastCalleeId, CallContextFieldTag::LastCalleeReturnDataOffset, CallContextFieldTag::LastCalleeReturnDataLength,] { cb.call_context_lookup(true.expr(),None,field_tag,0.expr()); } },); cb.condition(and::expr([is_precompile.expr(),is_precheck_ok.expr()]), |cb|{ //Savecaller'scallstate hasheye 45ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

for(field_tag,value)in[(CallContextFieldTag::LastCalleeId,callee_call_id.expr()), (CallContextFieldTag::LastCalleeReturnDataOffset,0.expr()), (CallContextFieldTag::LastCalleeReturnDataLength, return_data_len.expr(),),]{ cb.call_context_lookup(true.expr(),None,field_tag,value); } },); cb.condition(and::expr([call_gadget.is_empty_code_hash.expr(),is_precheck_ok.expr()]), |cb|{ //ForCALLCODEopcode,ithasanextrastackpop`value`andoneaccount read //forcallerbalance(+2). // //ForDELEGATECALLopcode,ithastwoextracallcontextlookupsfor current //calleraddressandvalue(+2). // //NoextralookupsforSTATICCALLopcode. lettransfer_rwc_delta= is_call.expr()*not::expr(transfer.value_is_zero.expr())*2.expr(); letrw_counter_delta=21.expr() +is_call.expr()*1.expr() +transfer_rwc_delta.clone() +is_callcode.expr() +is_delegatecall.expr()*2.expr() +precompile_memory_writes; cb.require_step_state_transition(StepStateTransition{ rw_counter:Delta(rw_counter_delta), program_counter:Delta(1.expr()), stack_pointer:Delta(stack_pointer_delta.expr()), gas_left:Delta(- step_gas_cost.expr()), Figure12.1: zkevm-circuits/src/evm_circuit/execution/callop.rs#L255-L314 ExploitScenario Amaliciousprovergeneratesandsubmitsaproofofexecutionforatransactioninvolvinga CALL toanaddresswithemptycodethatwouldnormallyexhaustthetransaction’sgas.By definingthegascostaszero,thetransactionsucceeds.However,thisexecutiondoesnot matchthecorrectEVMsemantics,leadingtostatedivergenceandlossoffunds. Recommendations Shortterm,addconstraintstocorrectlycomputethegascostforthecallopcodes. hasheye 46ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Longterm,addnegativetestsensuringthatEVMtracesgascostsdonotsatisfythecircuit constraints. hasheye 47ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

13.Unconstrainedopcodesallownondeterministicexecution Severity:HighDifficulty:Medium Type:DataValidationFindingID:TOB-SCROLL-13 Target: zkevm-circuits/src/evm_circuit/execution/{return_revert.rs, error_code_store.rs,error_invalid_creation_code.rs, error_precompile_failed.rs} Description Severalopcodesaremissingconstraintsthatensurethecorrectcorrespondencebetween executionstateandopcode,allowingamaliciousprovertohijackthetransaction execution. TheScrollzkEVMcircuitchecksthecorrectexecutionofatransactionbyverifyinga prover-generatedexecutiontrace.Thisexecutiontraceconsistsofaseriesofstates,each representedbyaconstructorofthe ExecutionStateenum .Eachstatecorrespondstoan “executiongadget”inthecircuit,whichcheckspreconditionsandenforcescorrectupdates toEVMdatastructuressuchasmemoryandstorage. IntheScrollcodebase,thecorrespondencebetweentheexecutionstateandopcodeis enforcedentirelybythesegadgets.Mostexecutiongadgetsusea SameContextGadget (showninfigure13.1)tocheckthatthecurrent (executionstate,opcode) pair appearsinthe ResponsibleOpcode table.Executiongadgetsthatdonotuse SameContextGadget mustcheckthatthecurrentopcodeiscorrectforthecurrentstate throughothermeans.Forexample, ErrorOOGSloadSstoreGadget ,showninfigure13.2, usesa PairSelectGadget toenforcethat,whentheexecutionstateis ErrorOutOfGasSloadSstore ,theopcodemustbeeither SSTORE or SLOAD . cb.add_lookup("Responsibleopcodelookup", Lookup::Fixed{ tag:FixedTableTag::ResponsibleOpcode.expr(), values:[cb.execution_state().as_u64().expr(), opcode.expr(), 0.expr(),], },); Figure13.1: zkevm-circuits/src/evm_circuit/util/common_gadget.rs#48-58 hasheye 48ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

impl<F:Field>ExecutionGadget<F>forErrorOOGSloadSstoreGadget<F>{ constNAME:&'staticstr="ErrorOutOfGasSloadSstore"; constEXECUTION_STATE:ExecutionState=ExecutionState::ErrorOutOfGasSloadSstore; fnconfigure(cb:&mutConstraintBuilder<F>)->Self{ letopcode=cb.query_cell(); letis_sstore=PairSelectGadget::construct(cb, opcode.expr(), OpcodeId::SSTORE.expr(), OpcodeId::SLOAD.expr(),); Figure13.2: zkevm-circuits/src/evm_circuit/execution/error_oog_sload_sstore.rs#48-61 Becausecheckingtheopcode/statecorrespondenceistheresponsibilityofeachexecution gadget,ifanyexecutiongadgetfailstoproperlyconstraintheopcode,amaliciousprover canreplaceanotherexecutionstepwiththatgadget’sexecutionstate.

Inthesimplestcase,thiscanleadtostatedivergence,but,ingeneral,amaliciousprover mayhavealargeamountofcontrolovertheresultingstate. Forexample,the ReturnRevertGadget ,showninfigure13.3,doesnotenforcethatthe opcodeiseither RETURN or REVERT .Amaliciousprovercanreplaceanyexecutionstate witha RETURN_REVERT state,causingtheexecutiontohaltatanarbitrarypoint,and potentiallyreturningdatadependingonthevaluescurrentlyonthestackandinmemory.If thetransactioncreatesacontract,amaliciousprovercanreplacethecodebeingdeployed withvaluesavailableinmemoryatotherpointsintheinitcode’sexecution.

```
impl<F:Field>ExecutionGadget<F>forReturnRevertGadget<F>{ constNAME:&'staticstr="RETURN_REVERT";
constEXECUTION_STATE:ExecutionState=ExecutionState::RETURN_REVERT;
fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ letopcode=cb.query_cell();
cb.opcode_lookup(opcode.expr(),1.expr()); letoffset=cb.query_cell_phase2();
letlength=cb.query_word_rlc(); cb.stack_pop(offset.expr()); cb.stack_pop(length.expr());
letrange=MemoryAddressGadget::construct(cb,offset,length);
letis_success=cb.call_context(None,CallContextFieldTag::IsSuccess);
cb.require_boolean("is_successisboolean",is_success.expr()); //cb.require_equal( hasheye
49ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

//"ifis_success,opcodeisRETURN.ifnot,opcodeisREVERT", //opcode.expr(),
//is_success.expr()*OpcodeId::RETURN.expr()
//+not::expr(is_success.expr())*OpcodeId::REVERT.expr(), //); Figure13.3: zkevm-
circuits/src/evm_circuit/execution/return_revert.rs#55-77
```

Intotal,wefoundfourgadgetsthatdonotconstraintheopcodetomatchthecurrent executionstate: • ErrorCodeStoreGadget (execution/error_code_store.rs#41-87) • ErrorPrecompileFailedGadget (execution/error_precompile_failed.rs#38-85) ○ Additionally,the ErrorPrecompileFailedGadget failstocheckthatthe calledaddressisaprecompilecontractandismissingacorrecttransition enforcementusingthe CommonErrorGadget . • ErrorInvalidCreationCodeGadget (execution/error_invalid_creation_code.rs#L35-L73) • ReturnRevertGadget (execution/return_revert.rs#L60-L293) ExploitScenario SupposeabridgebetweentwoblockchainsusestheScrollzkEVMtobridgeassetsbetween them.Alicecraftsatransactionwhich,whenanopcodesuchasan ADD isinsteadexecuted asa RETURN ,willerroneouslywithdrawfundsfromthebridge.Shegeneratesamalicious executiontraceandsubmitsazkEVMprooftothebridge,whichallowshertodrainthe bridgeoffunds. Recommendations Shortterm,addthemissingopcodechecksto ErrorCodeStoreGadget , ErrorPrecompileFailedGadget , ErrorInvalidCreationCodeGadget ,and ReturnRevertGadget . Longterm,considerredesigningthewaythatopcodesmaptostates.Thecurrentdesign meansthatanyexecutiongadgetthatfailstoconstraintheopcodewillcause nondeterministicexecution.If,instead,eachexecutiongadgethasan enable input,and theEVMcircuitdeterministicallyselectswhichgadget(s)have enable==1 ,an underconstrainedexecutiongadgetcanaffectonlythebehaviorofopcodesthatare supposedtousethatgadget. hasheye 50ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

14.NondeterministicexecutionofReturnDataCopyGadgetand ErrorReturnDataOutOfBoundGadget Severity:HighDifficulty:High Type:DataValidationFindingID:TOB-SCROLL-14 Target: zkevm-circuits/src/evm_circuit/execution/returndatacopy.rs Description Thegadgetthatimplementsthesuccessfulexecutionofthe RETURNDATACOPY opcodeis underconstrained,allowingamaliciousprovertosuccessfullyexecutetheopcodewhenit isinanerrorconditionforparticularopcodeinputs.Thisallowstheprovertocausestate divergencefromacorrectEVMexecution.

Figure14.1showstheerrorgadgetimplementationthatconstrainsthetracetohaveat leastonetrueerrorconditionforthe RETURNDATACOPY opcode.Theseconstraintscheck overflowconditionsonthestackvaluesandtheirsum.

```
//Checkif`data_offset`isUint64overflow.
letdata_offset_larger_u64=sum::expr(&data_offset.cells[N_BYTES_U64..]);
letis_data_offset_within_u64=IsZeroGadget::construct(cb,data_offset_larger_u64);
//Checkif`remainder_end`isUint64overflow. letsum=AddWordsGadget::construct(cb,
[data_offset,size],remainder_end.clone()); letis_end_u256_overflow=sum.carry().as_ref().unwrap();
letremainder_end_larger_u64=sum::expr(&remainder_end.cells[N_BYTES_U64..]);
letis_remainder_end_within_u64=IsZeroGadget::construct(cb, remainder_end_larger_u64);
//checkif`remainder_end`exceedsreturndatalength.
letis_remainder_end_exceed_len=LtGadget::construct( cb, return_data_length.expr(),
from_bytes::expr(&remainder_end.cells[..N_BYTES_U64]), );
//Needtocheckif`data_offset+size`isU256overflowvia`AddWordsGadget` carry.If
//yes,itshouldbealsoanerrorofreturndataoutofbound. cb.require_equal(
"Anyof[data_offset>u64::MAX,data_offset+size>U256::MAX,remainder_end>
```

u64::MAX,remainder_end>return_data_length]occurs", or::expr([//data_offset>u64::MAX hasheye

not::expr(is_data_offset_within_u64.expr()), //data_offset+size>U256::MAX
is_end_u256_overflow.expr(), //remainder_end>u64::MAX
not::expr(is_remainder_end_within_u64.expr()), //remainder_end>return_data_length
is_remainder_end_exceed_len.expr(),]), 1.expr(),); Figure14.1:
evm_circuit/execution/error_return_data_oo_bound.rs#L68-L101
Onthesuccessfulexecutionpath,theseconditionsarenotcheckedtobefalse.Infact,if
data_offset=WORD_CELL_MAX , size=0 ,and return_data_size<2 32 ,the ReturnDataCopyGadget
constraintsaresatisfied.Thiscaseisanerrorstatebecause data_offset islargerthan u64::MAX .
//3.contraintsforcopy:copyoverflowcheck //i.e.,offset+size⩽return_data_size
letin_bound_check=RangeCheckGadget::construct(cb, return_data_size.expr() -
(from_bytes::expr(&data_offset.cells)+from_bytes::expr(&size.cells)),); //4.memorycopy
//Constructmemoryaddressinthedestination(memory)towhichwecopymemory.
letdst_memory_addr=MemoryAddressGadget::construct(cb,dest_offset,size);
//Calculatethenextmemorysizeandthegascostforthismemory
//access.Thisalsoaccountsforthedynamicgasrequiredtocopybytesto //memory.
letmemory_expansion=MemoryExpansionGadget::construct(cb, [dst_memory_addr.address()]);
letmemory_copier_gas=MemoryCopierGasGadget::construct(cb, dst_memory_addr.length(),
memory_expansion.gas_cost(),); letcopy_rwc_inc=cb.query_cell();
cb.condition(dst_memory_addr.has_length(),|cb|{ cb.copy_table_lookup(last_callee_id.expr(),
CopyDataType::Memory.expr(), cb.curr.state.call_id.expr(), CopyDataType::Memory.expr(),
return_data_offset.expr()+from_bytes::expr(&data_offset.cells),
return_data_offset.expr()+return_data_size.expr(), dst_memory_addr.offset(),
dst_memory_addr.length(), hasheye

0.expr(),//forRETURNDATACOPYrlc_accis0 copy_rwc_inc.expr(),); });
cb.condition(not::expr(dst_memory_addr.has_length()),|cb|{ cb.require_zero(
"ifnobytestocopy,copytablerwcinc==0", copy_rwc_inc.expr(),); }); Figure14.2:
evm_circuit/execution/returndatacopy.rs#L99-L141
Insum,theprovercoulddecidewhethertheexecutionwouldcorrectlyhaltwiththe
ErrorReturnDataOutOfBoundGadget errororifitwouldsuccessfullyexecutethe RETURNDATACOPY opcode.
ExploitScenario Amaliciousprovergeneratesandsubmitsaproofofexecutionforatransactioninvolvinga
RETURNDATACOPY withparticulararguments.Duetothemissingvalidationsonthe
successfulexecutionstate,theprovercouldchoosetosuccessfullyexecutetheopcode,or
halttheexecution,leadingtostatedivergenceandlossoffunds. Recommendations
Shortterm,addconstraintstoensurethatthesuccessfulexecutionstateisdisjointfrom
theerrorexecutionstate.
Longterm,investigateothererrorstatesandtheirassociatedopcodeimplementationsto
guaranteethattheirexecutionstateisdisjointandcannotbechosenbyamaliciousprover. hasheye

15.ManyRWcounterupdatesaremagicnumbers Severity:InformationalDifficulty:N/A
Type:DataValidationFindingID:TOB-SCROLL-15 Target: zkevm-circuits/src/evm_circuit/execution/
Description ThezkEVMcircuitchecksmemoryreadandwriteoperationsinatransaction'sexecution
tracebyperforminglookupsintotheRWtable.Withinthecircuit,updatestothestate
variable,whichtracksthetotalnumberofread/writeoperations,arefrequentlyspecified
withamanualcount.Thatmanualprocessiserror-proneanddifficulttocheck,anditcan
bereplacedwithacalculatedvalueinallcaseswehaveseen. Theread-
writeconsistencychecksinthezkEVMcircuitrequiretheoverallblocktohavea
correctcountofthetotalnumberoflookupsintotheRWtable.Ifthatcountisincorrect,a
maliciousprovercaninsertextraneouswriteoperationsintothetableandchoosean
arbitraryresultforanymemoryread(seeTOB-SCROLL-8foradetailedexplanation).Each
executiongadgetisindividuallyresponsibleforcreatinga StepStateTransition that
enforcesthecorrectupdateofthe rw_counter fieldof StepState .Forexample,figure 15.1showsthe
StepStateTransition forthe AddSubGadget .TherearethreeRW lookupscausedbythe stack_pop()
calls,andthusthe rw_counter_field issetto Delta(3) ,representinganincreasebythree.
//ADD:Popaandbfromthestack,pushconthestack //SUB:Popcandbfromthestack,pushaonthestack
cb.stack_pop(select::expr(is_sub.expr().0,c.expr(),a.expr())); cb.stack_pop(b.expr());
cb.stack_push(select::expr(is_sub.expr().0,a.expr(),c.expr())); //Statetransition
letstep_state_transition=StepStateTransition{ rw_counter:Delta(3.expr()),
program_counter:Delta(1.expr()), stack_pointer:Delta(1.expr()), gas_left:Delta(-
OpcodeId::ADD.constant_gas_cost().expr()), ..StepStateTransition::default() };
letsame_context=SameContextGadget::construct(cb,opcode,step_state_transition); Figure15.1:The

rw_counter updatein AddSubGadget (zkevm-circuits/src/evm_circuit/execution/add_sub.rs#51-65)
hasheye 54ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

However,manyexecutiongadgetshavemuchmorecomplicated rw_counter updates,
whicharedifficulttocheckforcorrectness.
Toillustratethiscomplexity,considerfigures15.2and15.3,whichshow ErrorInvalidOpcodeGadget and
ErrorWriteProtectionGadget .Eachofthemuses the CommonErrorGadget
,whichhasanRWcounterdeltaasthethirdparameter. However, ErrorInvalidOpcodeGadget
doesnotseemtocontainanyRWlookupsatall, butprovidesthevalue 2 ,while ErrorWriteProtectionGadget
seemstohaveeitherone orfourRWlookupsdependingonthevalueof is_call ,yetprovidesa 0 to
CommonErrorGadget . fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ letopcode=cb.query_cell();
cb.add_lookup("Responsibleopcodelookup", Lookup::Fixed{
tag:FixedTableTag::ResponsibleOpcode.expr(), values:[Self::EXECUTION_STATE.as_u64().expr(),
opcode.expr(), 0.expr(),], },);
letcommon_error_gadget=CommonErrorGadget::construct(cb,opcode.expr(),2.expr()); Figure15.2:
ErrorInvalidOpcodeGadget (zkevm-circuits/src/evm_circuit/execution/error_invalid_opcode.rs#27-41)
fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... //Lookupvaluesfromstackifopcodeiscall
//Precondition:Ifthere'saStackUnderflowCALL,ishandledbeforethiserror
cb.condition(is_call.expr(),|cb|{ cb.stack_pop(gas_word.expr());
cb.stack_pop(code_address_word.expr()); cb.stack_pop(value.expr());
//cb.require_zero("valueofcallisnotzero", //is_value_zero.expr()); });
//currentcallcontextisreadonly
cb.call_context_lookup(false.expr(),None,CallContextFieldTag::IsStatic,1.expr());
//constrainnotrootcallasatleastonepreviousstaticcallpreset. cb.require_zero(
"ErrorWriteProtectiononlyhappeninginternalcall", hasheye
55ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

cb.curr.state.is_root.expr(),);
letcommon_error_gadget=CommonErrorGadget::construct(cb,opcode.expr(),0.expr()); Figure15.3:
ErrorWriteProtectionGadget (evm_circuit/execution/error_write_protection.rs#33-80) In
ErrorInvalidOpcodeGadget ,thereareinfacttwototalRWlookups;however, unintuitively,theyoccurinside
CommonErrorGadget itself,asshowninfigure15.4.Thus, anycallerof CommonErrorGadget
effectivelymustaddtwotothevalueof rw_counter_delta .
pub(crate)fnconstruct_with_lastcallee_return_data(cb:&mutConstraintBuilder<F>,
opcode:Expression<F>, rw_counter_delta:Expression<F>, return_data_offset:Expression<F>,
return_data_length:Expression<F>,)→Self{ cb.opcode_lookup(opcode.expr(),1.expr());
letrw_counter_end_of_reversion=cb.query_cell(); //currentcallmustbefailed.
cb.call_context_lookup(false.expr(),None,CallContextFieldTag::IsSuccess,0.expr());
cb.call_context_lookup(false.expr(), None, CallContextFieldTag::RwCounterEndOfReversion,
rw_counter_end_of_reversion.expr(),); Figure15.4:TwoRWlookupsinside CommonErrorGadget (zkevm-
circuits/src/evm_circuit/util/common_gadget.rs#1019-1038) Thecaseof ErrorWriteProtectionGadget
issomewhatmorecomplexbutillustratesa usefulalternativetomanuallycountinglookups. CommonErrorGadget
iscalledwithan rw_counter_delta valueof 0 .Onemightexpectthatthisisincorrect;itshouldcountthe
twolookupsinside CommonErrorGadget ,plusoneunconditionallookupandthree
conditionallookupsoutside.However,uponcloserinspection, CommonErrorGadget only uses
rw_counter_delta atallwhen curr.state.is_root istrue. ErrorWriteProtectionGadget
canoccuronlyinsideofastaticcall,andtherootcallofa transactionisneverastaticcall—
therefore,thatcaseisneveractiveandthevalueof rw_counter_delta canbesettoadummyvalue,inthiscase, 0
.Instead, RestoreContextGadget handlestheRWcounterupdate,basingitonacallto
ConstraintBuilder::rw_counter_offset ,asshowninfigure15.5.Since hasheye
56ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

rw_counter_offset isupdatedautomaticallyineachcallto ConstraintBuilder::rw_lookup
,thatcountiscorrectbyconstruction. letrw_counter_offset=cb.rw_counter_offset()
+subsequent_rw_lookups +not::expr(is_success.expr())*cb.curr.state.reversible_write_counter.expr();
//Dostepstatetransition cb.require_step_state_transition(StepStateTransition{
rw_counter:Delta(rw_counter_offset), call_id:To(caller_id.expr()),
is_root:To(caller_is_root.expr()), is_create:To(caller_is_create.expr()),
code_hash:To(caller_code_hash.expr()), program_counter:To(caller_program_counter.expr()),
stack_pointer:To(caller_stack_pointer.expr()), gas_left:To(gas_left),
memory_word_size:To(caller_memory_word_size.expr()),
reversible_write_counter:To(reversible_write_counter), log_id:Same, }); Figure15.5:The rw_counter
updatein RestoreContextGadget (zkevm-circuits/src/evm_circuit/util/common_gadget.rs#185-202)
Ingeneral,theprocessofcountingthenumberofRWlookupsinanygivengadgetisboth
subtleandtediouswhendonemanually,butcasesthatuse ConstraintBuilder::rw_counter_offset
todeterminethatoffsetareeffectively trivialwhenchecking forcorrectness. CreateGadget

hasverycomplexlogic,including threedifferentconditionalcallsto require_step_state_transition andalarge numberofRWlookups.However,each StepStateTransition computesitsRWcounter updateautomatically,asillustratedinfigure15.6. cb.condition(not::expr(is_precheck_ok.expr()),|cb|{ //Savecaller'scallstate forfield_tagin[CallContextFieldTag::LastCalleeId, CallContextFieldTag::LastCalleeReturnDataOffset, CallContextFieldTag::LastCalleeReturnDataLength,] { cb.call_context_lookup(true.expr(),None,field_tag,0.expr()); } cb.require_step_state_transition(StepStateTransition{ rw_counter:Delta(cb.rw_counter_offset()), program_counter:Delta(1.expr()), stack_pointer:Delta(2.expr()+IS_CREATE2.expr()), memory_word_size:To(memory_expansion.next_memory_word_size()), //-(Reversible)WriteTxAccessListAccount(ContractAddress) hasheye 57ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

reversible_write_counter:Delta(1.expr()), gas_left:Delta(-gas_cost.expr()), ..StepStateTransition::default() }); }); Figure15.6:Onepossible rw_counter updatein CreateGadget (zkevm-circuits/src/evm_circuit/execution/create.rs#337-357) Bycontrast, CallOpGadget hasthreedifferent StepStateTransition s,eachofwhich hasamanuallyconstructedRWcounteroffset.Thisincludesthe StepStateTransition showninfigure15.7,whichcountsagrandtotalof41RWlookupsplusfourconditional offsets,allofwhichneedtobeverifiedtobecorrect. lettransfer_rwc_delta= is_call.expr()*not::expr(transfer.value_is_zero.expr())*2.expr(); letrw_counter_delta=41.expr() +is_call.expr()*1.expr() +transfer_rwc_delta.clone() +is_callcode.expr() +is_delegatecall.expr()*2.expr(); cb.require_step_state_transition(StepStateTransition{ rw_counter:Delta(rw_counter_delta), call_id:To(callee_call_id.expr()), is_root:To(false.expr()), is_create:To(false.expr()), code_hash:To(call_gadget.phase2_callee_code_hash.expr()), gas_left:To(callee_gas_left), //ForCALLopcode,`transfer`invocationhastwoaccountwriteifvalueisnot //zero. reversible_write_counter:To(transfer_rwc_delta), ..StepStateTransition::new_context() }); Figure15.7:One rw_counter updatein CallOpGadget (zkevm-circuits/src/evm_circuit/execution/callop.rs#440-458) Recommendations Shortterm,replacemagic-numberRWcounterupdateswithcomputedvalues,suchas thoseprovidedby ConstraintBuilder::rw_counter_offset . Longterm,considerredesigningtheAPIforbuilding StepStateTransition s.SinceallRW lookupsareperformedviathe ConstraintBuilder API,updatestosimplecounter-style statevariables,suchasthestackpointerandtheRWcounter,cantypicallybecomputed ratherthanmanuallyspecified.Iftheeasy-to-calculatefieldsarealwayscomputed,that corecomputationcanbecheckedforcorrectness;asaresult,alluseswillbecorrectby construction. hasheye 58ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

16.NativePCSaccumulationdecidersacceptanemptyvector Severity:MediumDifficulty:Low Type:DataValidationFindingID:TOB-SCROLL-16 Target: snark-verifier/src/pcs/{kzg,ipa}/decider.rs Description BoththeKZGandIPAnative decide_all implementationsacceptanemptyvectorof accumulators.Thiscanallowanattackertobypassverificationbysubmittinganempty vector. fndecide_all(dk:&Self::DecidingKey, accumulators:Vec<IpaAccumulator<C,NativeLoader>>,)→bool{ !accumulators .into_iter() .any(|accumulator|!Self::decide(dk,accumulator)) } Figure16.1: snark-verifier/src/pcs/kzg/decider.rs#L54-L69 ThisfunctioncontrastswiththeEVMloaderimplementationthatassertsthatthe accumulatorvectorisnon-empty: fndecide_all(dk:&Self::DecidingKey, mutaccumulators:Vec<KzgAccumulator<M::G1Affine,Rc<EvmLoader>>>,)→Result<(),Error>{ assert! (!accumulators.is_empty()); Figure16.2: snark-verifier/src/pcs/kzg/decider.rs#L120-L124 ExploitScenario Anattackerisabletocontroltheargumentsto decide_all andpassesanemptyvector, causingtheverificationfunctiontoacceptaninvalidproof. Recommendations Shortterm,addanassertionthatvalidatesthatthevectorisnon-empty. Longterm,addnegativetestsforverificationandvalidationfunctions,ensuringthatwrong orinvalidargumentsarenotaccepted. hasheye 59ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

17.TheErrorOOGSloadSstoreandtheErrorOOGLoggadgetshaveredundant tablelookups Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-17 Target: zkevm-circuits/src/evm_circuit/execution/{error_oog_sload_sstore, error_oog_log}.rs Description Boththe ErrorOOGSloadSstore andthe ErrorOOGLog gadgetsdoanRWtablelookupto checkwhetherthecurrentcalliswithinastaticcontext.However,thelookupresultisnot usedinanysubsequentconstraint,makingthelookupredundant. //constrainnotinstaticcall letis_static_call=cb.call_context(None,CallContextFieldTag::IsStatic); //cb.require_zero("is_static_callisfalseinLOGN",is_static_call.expr()); Figure17.1: evm_circuit/execution/error_oog_log.rs#L53-L55 Thecommented-outconstraintwouldprovideaclearstatedistinctionbetweenthe ErrorOOGLogGadget errorcaseandthe ErrorWriteProtectionGadget ,preventingan attackerfromarbitrarilychoosingoneoftheerrorstatesatwill. Asfarasweknow,inthiscase,thesetwodifferenterrorexecutionstatesdonottranslate

todivergingEVMstates;thus,thisfinding’sseverityisinformational. Recommendations Shortterm,decidewhethertoremovetheRWtablelookuportouncommentthe non-staticenvironmentconstraintinboththe ErrorOOGSloadSstore and ErrorOOGLog gadgets.Investigateallcommented-outconstraintsandremovethemfromthecodebase, orenablethemiftheyarenecessary. hasheye 60ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

18.TheStatecircuitdoesnotenforcetransactionreceiptconstraints Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-18 Target: zkevm-circuits/src/state_circuit/constraint_builder.rs Description Theimplementationofthe State circuitdoesnotenforcetransactionreceiptconstraints. Currently,thesehaveanunsatisfiableconstraint(1==0),andthefunctionthatimplements them, build_tx_receipt_constraints ,isnotcalledinthe ConstraintBuilder::build function. fnbuild_tx_receipt_constraints(&mutself,q:&Queries<F>){ //TODO:implementTxReceiptconstraints self.require_equal("TxReceiptrowsnotimplemented",1.expr(),0.expr()); self.require_equal("state_rootisunchangedforTxReceipt", q.state_root(), q.state_root_prev(),); self.require_zero("value_prev_columnis0forTxReceipt", q.value_prev_column(),); } Figure18.1: state_circuit/constraint_builder.rs#L511-L524 Recommendations Shortterm,implementthetransactionreceiptconstraintsandaddthemtotheconstraint builderbuildfunction. Longterm,enablethe dead_code compilerlintbyremovingthe #![allow(dead_code)] linein zkevm-circuits/src/lib.rs andfixallwarnings. hasheye 61ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

20.TheEXPopcodehasanunusedwitness Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-20 Target: zkevm-circuits/src/evm_circuit/execution/exp.rs Description The EXP opcodedefinesawitnessthatisusedonlyinaconstraintrequiringitsvaluetobe zero.Theconstraintlabelsuggeststhatitwasusedtovalidatethe base_sq witnessvalue atsomepointinthecodedevelopment,butthisisnowdoneintheexponentiationtable circuit. letzero_rlc=cb.query_word_rlc(); cb.require_zero("base*base+c==base^2(c==0)", sum::expr(&zero_rlc.cells),); Figure20.1: evm_circuit/execution/exp.rs#L93-L97 Recommendations Shortterm,removethe zero_rlc variableanditsconstraintfromthe EXP opcodegadget. hasheye 62ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

21.Thebn_to_fieldfunctionsilentlytruncatesbigintegers Severity:LowDifficulty:Low Type:DataValidationFindingID:TOB-SCROLL-21 Target: misc-precompiled-circuit/src/utils/mod.rs Description The bn_to_field functionconvertsarbitrarylengthintegersintoafieldelement. However,ifthebyterepresentationoftheintegersislargerthan64bytes,thebiginteger byteswillbesilentlytruncated.Thismeansthatanytwointegerswiththesame512least significantbitswillleadtothesamefieldelement. pubfnbn_to_field<F:FieldExt>(bn:&BigUint)→F{ letmutbytes=bn.to_bytes_le(); bytes.resize(64,0); F::from_bytes_wide(&bytes.try_into().unwrap()) } Figure21.1: src/utils/mod.rs#L10-L15 Instead,thefunctionshouldcheckwhetherthebigintegerfitsintothefieldcapacityby usingthe F::capacity constant.Thiswouldguaranteeafaithfulrepresentationofthebig integerintothefieldelementandasuccessfulreconversionbacktothe BigUint type. Notethatthe from_bytes_wide functionwillalsoreducetheelementmodulothefield ordersothatitisrepresentedasafieldelement. ExploitScenario Anattackerprovidestwobigintegerswiththesame512leastsignificantbitstothe bn_to_field function,causingittoreturnthesameelement.Whentheseelementsare usedinfutureoperations,theywillleadtothesameresult,eventhoughtheywere different. Recommendations Shortterm,adddocumentationtothefunctionexplainingtheintendedbehavior;add checksthatvalidatethatthebigintegerisrepresentableinthechosenfield. hasheye 63ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

22.Thefield_to_bnfunctiondependsonimplementation-specificdetailsof theunderlyingfield Severity:LowDifficulty:High Type:DataValidationFindingID:TOB-SCROLL-22 Target: misc-precompiled-circuit/src/utils/mod.rs Description Theimplementationofthe field_to_bn functioncallsthe to_repr functiononthevalue oftheinput f whenconstructingthelittle-endianbinaryrepresentationoftheinput f . pubfnfield_to_bn<F:FieldExt>(f:&F)→BigUint{ letbytes=f.to_repr(); BigUint::from_bytes_le(bytes.as_ref()) } Figure22.1:Theimplementationof field_to_bn expects to_repr toreturnalittle-endian representationofthevalueof f . (src/utils/mod.rs#L5-L8) However,accordingtothedocumentationofthe PrimeField trait,theendianness returnedby PrimeField::to_repr isimplementation-dependentandmaybedifferent dependingontheunderlyingfield. ///Convertsanelementoftheprimefieldintothestandardbyterepresentationfor ///thisfield. /// ///Theendiannessofthebyterepresentationisimplementation-specific.Generic ///encodingsoffieldelementsshouldbetreatedasopaque. fnto_repr(&self)→Self::Repr; Figure22.2:Thevaluereturnedby to_repr isimplementation-dependentandshouldbe treatedasopaquebytheuser. ExploitScenario The field_to_bn functionisreusedwithascalarfield F thatusesadifferentinternal representationoftheelementsof F

.Theresultingbigintegermightnotcorrespondtothe samefieldelement. Recommendations Shortterm,implementafunctionthatassuredlyreturnsalittle-endianrepresentationof thefieldelement. hasheye 64ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Longterm,reviewtheuseofthird-partyAPIstoensurethatthecodebasedoesnotdepend ontheinternalrepresentationofdata. hasheye 65ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

23.Thevaluesofthebytecodetabletagcolumnarenotconstrainedtobe HEADERorBYTE Severity:InformationalDifficulty:High Type:DataValidationFindingID:TOB-SCROLL-23 Target: zkevm-circuits/src/bytecode_circuit/circuit.rs Description Thebytecodetablehasacolumnthatindicatesthe TAG ofeachrow.Currently,the TAG cellsareassignedonlya HEADER ora BYTE value.However,thecircuitdoesnotconstrain the TAG valueofeachrowtoacceptonlythesevalues.Thismissingconstraintdoesnot causeadirectsoundnessissuebecauseofotherindirectconstraintsandhowthebytecode circuitisimplemented,butfuturecoderefactoringscouldcausetheissuetobecome exploitable. Thebytecodetablecontainsthesetofbytecodesthatareexecutedinablock.Foreach bytecode,thetablecontainsa HEADER row,followedby BYTE rowscorrespondingtoeach byteofthebytecode,andafinal HEADER row. Thecircuitimposesconstraintsforeachtypeofrow(e.g.,figure23.1showshowa HEADER rowisconstrainedtohaveanindexcolumnvalueof 0),andfortransitionsbetweentwo rows(e.g.,transitioningfroma HEADER rowtoa BYTE row,thelengthcolumnmuststaythe same). //Whenis_header→ //assertcur.index==0 //assertcur.value==cur.length meta.create_gate("Headerrow",|meta|{ letmutcb=BaseConstraintBuilder::default(); cb.require_zero("cur.index==0", meta.query_advice(bytecode_table.index,Rotation::cur()),); cb.require_equal("cur.value==cur.length", meta.query_advice(bytecode_table.value,Rotation::cur()), meta.query_advice(length,Rotation::cur()),); cb.gate(and::expr(vec![hasheye 66ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

meta.query_fixed(q_enable,Rotation::cur()), not::expr(meta.query_fixed(q_last,Rotation::cur())), is_header(meta),])) }); Figure23.1: zkevm-circuits/src/bytecode_circuit/circuit.rs#L178-L200 Tocheckwhetherarowisa HEADER ora BYTE row,theimplementationperformssteps thatrelyonparticularassumptions: • Itimplicitlyassumesthatthe enum valuecorrespondingto HEADER is 0 andtheone to BYTE is 1 .Thisassumptioncanbebrokeninthefutureifadeveloperaddsan extra enum fieldonthe BytecodeFieldTagenum . • ItusesBooleanoperators and::expr , not::expr onthebytecodetagvalues. Theseoperatorsmustoperateonlyon Booleanvalues;otherwise,theywillreturnan unexpectedvalue. • Itgatestheconstraintswithconjunctionsresultingfromthe and::expr operator:if alookupisguardedbytheconjunctionofnon-Booleanvalues,thevaluethatis lookedupwillbeascaledversionoftheintendedvalue. letis_byte_to_byte=|meta:&mutVirtualCells<F>|{ and::expr(vec![meta.query_advice(bytecode_table.tag,Rotation::cur()), meta.query_advice(bytecode_table.tag,Rotation::next()),]) }; letis_header=|meta:&mutVirtualCells<F>|{ not::expr(meta.query_advice(bytecode_table.tag,Rotation::cur())) }; letis_byte= |meta:&mutVirtualCells<F>|meta.query_advice(bytecode_table.tag, Rotation::cur()); Figure23.2: zkevm-circuits/src/bytecode_circuit/circuit.rs#L125-L137 Thesoundnessofallthesestepsandimplementationdetailsrelyonthebytecodetagvalue beingBoolean.However,theimplementationdoesnothaveaconstraintvalidatingthatthe TAG valuesare,infact,Boolean. Ifamaliciousproverweretoprovideanon-Booleanvalue,sincethe not::expr and and::expr functionsoperateundertheassumptionthattheirinputvaluesareBoolean, thecircuitwillhavesoundnessissues. hasheye 67ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Oneavenueofexploitationisonthe push_data_size_table_lookup onthebytecode table:thislookupisgatedonthe is_byte(meta) constraint,causingittobeimplicitly scaledtoadifferentlookupifthe is_byte(meta) resultisnon-Boolean.Thiswouldallow amaliciousprovertoobtainthewrong push_data_size fromthe push_data_size_table_lookup tableandprovideanincorrectbytecodewithrespectto the is_code column.Inotherwords,thedatapushedina PUSH* opcodecouldbemarked ascode,thenallowingtheEVMexecutiontofollowanexecutionflowincompatiblewith EVMsemantics. meta.lookup_any("push_data_size_table_lookup(cur.value,cur.push_data_size)", |meta|{ letenable=and::expr(vec![meta.query_fixed(q_enable,Rotation::cur()), not::expr(meta.query_fixed(q_last,Rotation::cur())), is_byte(meta),]); letlookup_columns=vec! [value,push_data_size]; letmutconstraints=vec![]; foriin0..PUSH_TABLE_WIDTH{ constraints.push((enable.clone()*meta.query_advice(lookup_columns[i], Rotation::cur()), meta.query_fixed(push_table[i],Rotation::cur()),)) } constraints },); Figure23.3: zkevm-circuits/src/bytecode_circuit/circuit.rs#L220-L241 However,arowwithanon-Boolean TAG actuallysatisfiesboththe is_header(meta) and is_byte(meta) constraints.Thus,foranattackertobesuccessful,theywouldhaveto satisfyanunsatisfiablesetofconstraintsonthe index column: • is_header :requires cur.index==0 • is_header_to_byte :requires next.index==0 • is_byte_to_byte :requires next.index==cur.index+1

Thereexistsanotheravenueofexploitingthemissingconstraintthatrequiresthe TAG valuetoequal BYTE or HEADER .Ifamaliciousproverwereabletoinjectrowsinthetable withadifferent TAG value,theywouldbeabletodisablethe BYTE-TO-BYTE , BYTE-TO-HEADER , HEADER-TO-HEADER ,and HEADER-TO-BYTE transitionconstraints. hasheye 68ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Asanexample,ifbetweentwo HEADER rowsthereexistedarowdifferentfrom HEADER or BYTE ,the HEADER-TO-HEADER transitiongatewouldalwaysbefalse.Thisexploitscenario isunexploitableforthesamereasonasthepreviousexploit.However,whilecorrectly enforcingthe is_header , is_header_to_byte , is_byte_to_byte , is_header_to_header , is_byte ,and is_byte_to_header transitionconstraintsto acceptonlythe HEADER and BYTE valueswouldpreventthe push_data_size_table_lookup exploit,itwouldnotpreventtherow-to-rowtransition constraintsfrombeingbroken.Foracompletefix,itisnecessarytoconstrainthe TAG value tobeoneof HEADER or BYTE . ExploitScenario Anticipatingafutureadditiontothe TAGenum ,adeveloperdecidestoreimplementthe is_header , is_byte ,andtransitionselectorsbyrequiringthatthe TAG cellvalueequals thedesiredenumvalue.Iftheyomitthecheckthatthe TAG valuemustberestrictedtothe enum ’svalueset,amaliciousproverwouldbeabletobreakthetransitionconstraintsby insertingarowwithatagvaluedifferentfrom HEADER or BYTE . Recommendations Shortterm,requirethatthe TAG columnisBooleanintheconstraintsystem;makethe BytecodeFieldTag valuesexplicitly 0 and 1 bydefiningthe enum asfollows: pubenumBytecodeFieldTag{ ///Headerfield Header=0, ///Bytefield Byte=1, } Figure23.4:Explicit enum definition Documentwhichconstraintsneedtobechangedincasethe BytecodeFieldTagenum is extended. Longterm,addstrictertypestotheBooleanfunctionsin gadgets/src/util.rs .These functions,astheirdocumentationstates,shouldoperateonlyonBooleanvalues.Enforcing thisinthetypesystemwouldallowcaseswherethisassumptionisviolatedtobefound andwouldpreventpotentialsoundnessissues. hasheye 69ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

24.UnconstrainedcolumnsonthebytecodeHEADERrows Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-24 Target: zkevm-circuits/src/bytecode_circuit/{circuit/to_poseidon_hash.rs, circuit.rs} Description Thebytecodetable HEADER rowshavetwounconstrainedcolumns, is_code and field_input ,onthePoseidonbytecodeextendedcolumns.Thelackofconstraintson thesecolumnsdoesnotseemtoposeanysoundnessissue,butconstrainingthese columnswouldserveasdefense-in-depth,preventingthecircuit’sflexibilityfromallowinga maliciousprovertoexploitasoundnessissueifavulnerabilityisintroducedinthefuture. Figure24.1showsthe HEADER rowconstraints,andnoconstraintrelatedtothe is_code column. meta.create_gate("Headerrow",|meta|{ letmutcb=BaseConstraintBuilder::default(); cb.require_zero("cur.index==0", meta.query_advice(bytecode_table.index,Rotation::cur()),); cb.require_equal("cur.value==cur.length", meta.query_advice(bytecode_table.value,Rotation::cur()), meta.query_advice(length,Rotation::cur()),); cb.gate(and::expr(vec![meta.query_fixed(q_enable,Rotation::cur()), not::expr(meta.query_fixed(q_last,Rotation::cur())), is_header(meta),])) }); Figure24.1: zkevm-circuits/src/bytecode_circuit/circuit.rs#L179-L201 Recommendations Shortterm,addconstraintsforthe is_code andthe field_input rowsinthe HEADER rowsofthebytecodetable. hasheye 70ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

Longterm,documentalltableconstraintsandensurethateachtypeofrowconstrainsall columns. hasheye 71ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

25.decompose_limbdoesnotworkasintended Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-25 Target: misc-precompiled-circuit/src/circuits/modexp.rs Description The for loopwithin decompose_limb requires bool_limbs tocontainatleast 31 elementstobecorrectlyindexedfrom 0 to 30 .Furthermore,if limbsize islargeenough, thenthe truncate operationdoesnotgrow bool_limbs tothecorrectsize,as to_radix_le producesaminimal Vec withoutanytrailingzeroes. letmutbool_limbs=field_to_bn(&limb.value).to_radix_le(2); bool_limbs.truncate(limbsize); bool_limbs.reverse(); letmutv=F::zero(); foriin0..27{ letl0=F::from_u128(bool_limbs[i]asu128); letl1=F::from_u128(bool_limbs[i+1]asu128); letl2=F::from_u128(bool_limbs[i+2]asu128); letl3=F::from_u128(bool_limbs[i+3]asu128); Figure25.1: misc-precompiled-circuit/src/circuits/modexp.rs#L514-L522 Additionally,theBooleanlimbsarenotproperlyconstrainedtobeBoolean,butthisis mentionedina“TODO”comment. Overall,itcanbeconcludedthatthe decompose_limb needsfurtherdevelopment,butits intendedpurposeandusagewithin mod_exp isclear. Recommendations Shortterm,correctlyimplement decompose_limb .Thiswillallowforpropertestingofthe mod_exp circuit. hasheye 72ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

26.Zeromoduluswillcauseapanic Severity:MediumDifficulty:Low Type:DataValidationFindingID:TOB-SCROLL-26 Target: misc-precompiled-circuit/src/circuits/modexp.rs Description AccordingtoEVMspecifications,ifthemodulusiszero,thentheresultof mod_exp iszero

regardlessoftheinput.Thecurrent mod_exp codereliesonsuccessivecallsto mod_mult withthepassed-inmodulus,butthe mod_mult functioncomputesaquotientthatwill panic. letbn_quotient=bn_mult.clone().div(bn_modulus.clone());//div_rem Figure26.1: misc-precompiled-circuit/src/circuits/modexp.rs#L470 Thisresultsindifferingbehaviorbetweenthescroll mod_exp precompileandthestandard EVMprecompile,whichmaycausesomeexistingsystemsthatdependonthisbehaviorto notworkasintended. Recommendations Shortterm,correctlyhandlethezeromoduluscaseof mod_exp .Addteststothe mod_exp circuit,includingsomethatexerciseitsedgecases:thezeroexponentcaseandthezero moduluscase. hasheye 73ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

27.TheConstraintBuilder::conditionAPIisdangerous Severity:InformationalDifficulty:High Type:DataValidationFindingID:TOB-SCROLL-27 Target:Severalfiles Description The ConstraintBuilder implementsseveralusefulwaysofconstructingconstraints.One caseiswhenconstraintsshouldbeaddedandconditionedbyaparticularvalue.Ifthe valueistrue,theconstraintsmustbesatisfied;otherwise,theydonotneedtobesatisfied. However,aproblemarisesifadeveloperforgetstoconsiderthatanew ConstraintBuilder functioniscalledwithinthecontextofa condition . Allfunctionsinthe ConstraintBuilder APImustconsiderthecasewheretheyarebeing calledfrominsideaconditionedscope.Ifthesefunctionsaddconstraintsorchangevalues irrespectiveoftheconditionvalue,theywillleadtounintendedresults. Asanexample,the opcode_lookup functionupdatesthe program_counter_offset regardlessofthecurrentconditionvalue. pub(crate)fnopcode_lookup(&mutself,opcode:Expression<F>,is_code: Expression<F>){ self.opcode_lookup_at(self.curr.state.program_counter.expr()+self.program_counter_offset.expr(), opcode, is_code,); self.program_counter_offset+=1; } Figure27.1: evm_circuit/util/constraint_builder.rs#608-615 Whenusedinaconditioncontext,the program_counter_offset willbeincremented irrespectiveofthe condition value: constNAME:&'staticstr="STOP"; constEXECUTION_STATE:ExecutionState=ExecutionState::STOP; fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ letcode_length=cb.query_cell(); cb.bytecode_length(cb.curr.state.code_hash.expr(),code_length.expr()); hasheye 74ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

letis_within_range= LtGadget::construct(cb,cb.curr.state.program_counter.expr(), code_length.expr()); letopcode=cb.query_cell(); cb.condition(is_within_range.expr(),|cb|{ cb.opcode_lookup(opcode.expr(),1.expr()); }); Figure27.2: src/evm_circuit/execution/stop.rs#33-45 Theprovidedargumenttothe ConstraintBuilder::condition functionmustalsobe ensuredtobeBoolean.Certainfunctionsassumethis,andtheywouldhaveunexpected resultsotherwise: pub(crate)fnstack_pop(&mutself,value:Expression<F>){ self.stack_lookup(false.expr(),self.stack_pointer_offset.clone(),value); self.stack_pointer_offset=self.stack_pointer_offset.clone()+ self.condition_expr(); } pub(crate)fnstack_push(&mutself,value:Expression<F>){ self.stack_pointer_offset=self.stack_pointer_offset.clone()- self.condition_expr(); self.stack_lookup(true.expr(),self.stack_pointer_offset.expr(),value); } Figure27.3: evm_circuit/util/constraint_builder.rs#1160-1169 The ConstraintBuilder::gate functionisanotherdangerouspatternthatshouldbe reconsideredanddocumented.Initscurrentstate,thefunctionclonesallconstraintsand gatesthemwiththeprovidedselector,returningthesenewgatedconstraints.Itdoesnot changethecurrentconstraints,whichmightbeaninterpretationthatanewdeveloper mighthaveaboutthefunction.Wehavenotseenincorrectusageofthisparticularpattern. Onewayofatleastensuringthatthereturningsetofconstraintsisusedisbyaddingthe #[must_use] attributetothefunction. Recommendations Shortterm,redesignthe ConstraintBuilder API,especiallywithrespecttothe condition function.AddnewRusttypestoensurethattheconditionexpressionis Boolean.Addthe #[must_use] attributetothe ConstraintBuilder::gate function. hasheye 75ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

28.TheEXTCODECOPYopcodeimplementationdoesnotworkwhenthe accountaddressdoesnotexist Severity:InformationalDifficulty:N/A Type:DataValidationFindingID:TOB-SCROLL-28 Target: zkevm-circuits/src/evm_circuit/execution/extcodecopy.rs Description Thecurrentimplementationofthe EXTCODECOPY opcodedoesnotconsiderthecase wheretheaccountaddressdoesnotexist.Thisisdocumentedinacodecomment,sothe Scrollteamshouldbeawareofit. //TODO:Ifexternal_addressdoesn'texist,wewillgetcode_hash=0.With //thisvalue,thebytecode_lengthlookupwillnotwork,andthecopy //fromcode_hash=0willnotwork.WeshoulduseEMPTY_HASHwhen //code_hash=0. cb.bytecode_length(code_hash.expr(),code_size.expr()); Figure28.1: zkevm-circuits/src/evm_circuit/execution/extcodecopy.rs#84-88 Recommendations Shortterm,implementthemissingfunctionality.Addteststoensureitscorrectness. Longterm,lookforall“TODO”itemsinthecodebaseandtriagethemintoanorganized

issuetracker.Addresstheseitemsintermsofpriority. hasheye
76ScrollzkEVMhalo2CircuitsSecurityAssessment 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
77ScrollzkEVMhalo2CircuitsSecurityAssessment 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
78ScrollzkEVMhalo2CircuitsSecurityAssessment 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
79ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

C.CodeQualityFindings WeidentifiedthefollowingcodequalityissuesthroughmanualandautomaticcodereviewS.
• Useconstantsinsteadofhard-codedvalues.Insteadof32,usethe N_BYTES_WORD constant. foridxin0..32{
cb.add_lookup("Bitwiselookup", Lookup::Fixed{ tag:tag.clone(), values:[a.cells[idx].expr(),
b.cells[idx].expr(), c.cells[idx].expr(),], },); } FigureC.1: zkevm-
circuits/src/evm_circuit/execution/bitwise.rs#L47-L59 • Use Transition::Same insteadof
Delta(0.expr()) .The require_step_state_transition functionwillperformanextraadditionwhen handling
Delta(0.expr()) .Also,thelinereferencingthatfieldcanberemoved, sincethetype’sdefaultis Same
.Thisissueispresentinseveralfiles: zkevm-circuits/src/evm_circuit/execution/{balance.rs,
calldataload.rs,extcodehash.rs,extcodesize.rs,is_zero.rs, not.rs} . //Statetransition
letstep_state_transition=StepStateTransition{ rw_counter:Delta(2.expr()),

program_counter:Delta(1.expr()), stack_pointer:Delta(0.expr()), FigureC.2: zkevm-circuits/src/evm_circuit/execution/not.rs#L49-L53 • Removetheunusedfunction generate_lagrange_base_polynomial .The functionispresentat zkevm-circuits/src/evm_circuit/util/math_gadget.rs butitisnotusedin thecodebase. • Addextracheckswhendoingarithmeticontheopcode.Severalgadgetsdo arithmeticontheopcodetoextractarelevantvaluewhenmultiplerelatedopcodes areadjacentinvalue.Thispatterniserror-prone,andthegadgetsdonotaddchecks hasheye 80ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

toensuretheopcodeisinthecorrectrange.Extracheckssshouldbeincludedto preventmisuse,andtheseoperationsshouldbefactoredoutintoacommon moduletoaidreadability. letblockctx_tag=BlockContextFieldTag::Coinbase.expr() +(opcode.expr()- OpcodeId::COINBASE.as_u64().expr()); FigureC.3: evm_circuit/execution/block_ctx.rs#35-36 letswap_offset=opcode.expr()-(OpcodeId::SWAP1.as_u64()-1).expr(); FigureC.4: zkevm-circuits/src/evm_circuit/execution/swap.rs#35 letnum_additional_pushed=opcode.expr()- OpcodeId::PUSH1.as_u64().expr(); FigureC.5: zkevm-circuits/src/evm_circuit/execution/push.rs#85 lettopic_count=opcode.expr()-OpcodeId::LOG0.as_u8().expr(); FigureC.6: zkevm-circuits/src/evm_circuit/execution/logs.rs#102 letdup_offset=opcode.expr()-OpcodeId::DUP1.expr(); FigureC.7: zkevm-circuits/src/evm_circuit/execution/dup.rs#35 lettag= FixedTableTag::BitwiseAnd.expr()+(opcode.expr()- OpcodeId::AND.as_u64().expr()); FigureC.8: zkevm-circuits/src/evm_circuit/execution/bitwise.rs#45-46 • Thefollowingcommentisincorrect.The SSTORE gasrefundconstraintshave codecommentsdescribingeachconstraint.However,thecommentforthe delete_slot caseiswrong: //(value_prev≠value)&&(original_value≠value)&&(value== //Word::from(0)) letdelete_slot= not::expr(prev_eq_value.clone())*not::expr(original_is_zero.clone())* value_is_zero; FigureC.9: evm_circuit/execution/sstore.rs#L285-L288 • Theblanketmatchcasein require_step_state_transition couldleadto under-constrainedtransitions.The Transition::Any caseof require_step_state_transition ishandledimplicitlybyablanketmatch.Ifany newcaseisaddedtothe Transitionenum ,thedefaultbehaviorwillbetoleave thatfieldunconstrained.If,instead, Transition::Any isexplicitlyhandled,the Rustcompilerwillgenerateanincompletematcherror. hasheye 81ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

matchstep_state_transition.$name{ Transition::Same⇒self.require_equal(concat! ("Statetransition(same)constraintof",stringify!($name)), self.next.state.$name.expr(), self.curr.state.$name.expr(),), Transition::Delta(delta)⇒self.require_equal(concat! ("Statetransition(delta)constraintof",stringify!($name)), self.next.state.$name.expr(), self.curr.state.$name.expr()+delta,), Transition::To(to)⇒self.require_equal(concat! ("Statetransition(to)constraintof",stringify!($name)), self.next.state.$name.expr(), to,), _⇒{} } FigureC.10: evm_circuit/util/constraint_builder.rs#538-555 • Somecommentsappeartobecopy-pastedandrefertoothermodules. Commentsfor ErrorOOGAccountAccessGadget and ErrorInvalidCreationCodeGadget referto ErrorOOGExpGadget and ErrorCodeStoreGadget ,respectively: ///Gadgettoimplementthecorrespondingoutofgaserrorsfor ///[`OpcodeId::EXP`]. #[derive(Clone,Debug)] pub(crate)structErrorOOGAccountAccessGadget<F>{ FigureC.11: evm_circuit/execution/error_oog_account_access.rs#21-24 ///Gadgetforcodestoreoogandmaxcodesizeexceed #[derive(Clone,Debug)] pub(crate)structErrorInvalidCreationCodeGadget<F>{ FigureC.12: evm_circuit/execution/error_invalid_creation_code.rs#20-22 • Thereisaredundantexpressionidentifiercomputationin store_expression . The store_expression functioncomputestheexpressionidentifiertwiceincase theexpressionisnotalreadystored:onceinthe find_stored_expression and againintheconstructionofthe StoredExpression . pub(crate)fnstore_expression(&mutself, name:&str, expr:Expression<F>, cell_type:CellType,)- >Expression<F>{ hasheye 82ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

//Checkifwealreadystoredtheexpressionsomewhere letstored_expression=self.find_stored_expression(&expr,cell_type); matchstored_expression{ Some(stored_expression)⇒{ debug_assert!(!matches!(cell_type,CellType::Lookup(_)), "Thesamelookupisdonemultipletimes",); stored_expression.cell.expr() } None⇒{ //Evenifwe'rebuildingexpressionsforthenextstep, //theseintermediatevaluesneedtobestoredinthecurrent step. letin_next_step=self.in_next_step; self.in_next_step=false; letcell=self.query_cell_with_type(cell_type); self.in_next_step=in_next_step; //Requirethestoredvaluetoequalthevalueoftheexpression letname=format!("{}(storedexpression)",name); self.push_constraint(Box::leak(name.clone().into_boxed_str()), cell.expr()-expr.clone(),); self.stored_expressions.push(StoredExpression{ name, cell:cell.clone(), cell_type, expr_id:expr.identifier(), expr, }); cell.expr() } } } pub(crate)fnfind_stored_expression(&self, expr:&Expression<F>, cell_type:CellType,)→Option<&StoredExpression<F>>{ letexpr_id=expr.identifier(); self.stored_expressions .iter() .find(|&e|e.cell_type==cell_type&&e.expr_id==expr_id) } FigureC.13:

evm_circuit/util/constraint_builder.rs#L1493-L1546 hasheye
83ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

• Use query_cell_phase2() insteadof query_cell_with_type(CellType::StoragePhase2) .
letphase2_callee_code_hash= cb.query_cell_with_type(CellType::StoragePhase2); FigureC.14: zkevm-
circuits/src/evm_circuit/util/common_gadget.rs#L711 • Useconstantsinsteadofhard-codedvalues.
Ok(BaseFieldEccChip::<C>::variable_base_msm::<C>(self, ctx, &points, &scalars,
C::Scalar::NUM_BITSasusize, 4,//empiricallyclumpfactorof4seemstobebest)) } fnfixed_base_msm(
&mutself, ctx:&mutSelf::Context, pairs:&[(implDeref<Target=Self::AssignedScalar>,C)],)-
>Result<Self::AssignedEcPoint,Error>{ let(scalars,points):(Vec<_>,Vec<_>)=pairs .iter()
.filter_map(|(scalar,point)|{ ifpoint.is_identity().into(){ None }else{ Some((vec!
[scalar.deref().clone()],*point)) } }) .unzip(); Ok(BaseFieldEccChip::<C>::fixed_base_msm::<C>(
self, ctx, &points, &scalars, C::Scalar::NUM_BITSasusize, 0, 4,)) } FigureC.15: snark-
verifier/src/loader/halo2/shim.rs#L371-L406 • Thereareunnecessarytypehintsin origin.rs and
gasprice.rs .Thefiles containseveralunnecessarytypehints,suchas ::<N_BYTES_ACCOUNT_ADDRESS> , 2u64
, 1u64 ,and -1i32 . hasheye 84ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ letorigin=cb.query_word_rlc::
<N_BYTES_ACCOUNT_ADDRESS>(); //Lookupincall_ctxtheTxId
lettx_id=cb.call_context(None,CallContextFieldTag::TxId); //Lookuprw_table-
>call_contextwithtxoriginaddress cb.tx_context_lookup(tx_id.expr(),
TxContextFieldTag::CallerAddress, None,//Nonebecauseunrelatedtocalldata
from_bytes::expr(&origin.cells),); //Pushthevaluetothestack cb.stack_push(origin.expr());
//Statetransition letopcode=cb.query_cell(); letstep_state_transition=StepStateTransition{
rw_counter:Delta(2u64.expr()), program_counter:Delta(1u64.expr()),
stack_pointer:Delta((-1i32).expr()), FigureC.16: evm_circuit/execution/origin.rs#L32-L53 •
UnifytheconstraintbuilderAPIs.Thereareseveralrepeatedfunctionsinthe ConstraintBuilder and
BaseConstraintBuilder APIsin util/constraint_builder.rs . •
Therearefunctionallyidenticalfunctions.The get_num_rows_required_no_padding and
get_min_num_rows_required functionscomputethesamenumberofrowsinaslightlydifferentway.
pubfnget_num_rows_required_no_padding(block:&Block<F>)→usize{
//Startat1sowecanbesurethereisanunused`next`rowavailable letmutnum_rows=1;
fortransactionin&block.txs{ forstepin&transaction.steps{
num_rows+=step.execution_state.get_step_height(); } } num_rows+=1;//EndBlock num_rows } //...
pubfnget_min_num_rows_required(block:&Block<F>)→usize{ letmutnum_rows=0;
fortransactionin&block.txs{ forstepin&transaction.steps{ hasheye
85ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

num_rows+=step.execution_state.get_step_height(); } }
//ItmusthaveonerowforEndBlockandatleastoneunusedone num_rows+2 } FigureC.17: zkevm-
circuits/src/evm_circuit.rs#L210-L242 • Considerrenamingthe offset_add functionto set_offset .The
offset_add functionsetsthe offset astheargumentinsteadofaddingtheargumenttothe offset
,asthenamesuggests. ///Incrementthesteprwoperationoffsetby`offset`.
pub(crate)fnoffset_add(&mutself,offset:usize){ self.offset=offset } FigureC.18: zkevm-
circuits/src/evm_circuit/util.rs#L659-L662 • Thereareincorrectcommentsin halo2-ecc
.Severalellipticcurvefunctions incorrectlysaythattheyassume P.y isreduced,whentheyinsteadrequire
Q.x to bereduced.Thesecommentshavebeenupdatedinversion v0.3.0 ofthe upstream halo2-lib .
///Foroptimizationreasons,weassumethatifyouareusingthiswith
`is_strict=true`,thenyouhavealreadycalled`chip.enforce_less_than_p`on both`P.x`and`P.y`
pubfnec_add_unequal<F:PrimeField,FC:FieldChip<F>>(FigureC.19: halo2-lib/halo2-
ecc/src/ecc/mod.rs#55-56 ///Foroptimizationreasons,weassumethatifyouareusingthiswith
`is_strict=true`,thenyouhavealreadycalled`chip.enforce_less_than_p`on both`P.x`and`P.y`
pubfnec_sub_unequal<F:PrimeField,FC:FieldChip<F>>(FigureC.20: halo2-lib/halo2-
ecc/src/ecc/mod.rs#97-98 • Thereareoutdateddocumentationcommentsin halo2-ecc .The is_soft_zero and
is_soft_nonzero methodsof FieldChip haveoutdated
commentsthatdonotreflecttheimplementation.Thesecommentshavebeen updatedinversion v0.3.0
oftheupstream halo2-lib . //Assumesthewitnessforais0
//Constrainsthattheunderlyingbigintegeris0and<p. //Forfieldextensions,checkscoordinate-wise.
fnis_soft_zero(&self,ctx:&mutContext<F>,a:&Self::FieldPoint)→ hasheye
86ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

AssignedValue<F>; //Constrainsthattheunderlyingbigintegerisin[1,p-1].
//Forfieldextensions,checkscoordinate-wise.
fnis_soft_nonzero(&self,ctx:&mutContext<F>,a:&Self::FieldPoint)→ AssignedValue<F>; FigureC.21:
halo2-lib/halo2-ecc/src/fields/mod.rs#115-122 • Allowthe dead_code lintandfixallissues.The
dead_code lintiscurrently disabled;itshouldbeenabledtoallowdeveloperstoquicklydetectunused

functionsandvariables. • Reusethe CmpWordsGadget inthe ComparatorGadget .The ComparatorGadget implementationshouldreusethe CmpWordsGadget insteadofhavingthesame constraintsduplicatedonbothgadgets. • Simplifyexpressionimplementation.Addacommentexplainingthededuction andsimplifytheexpression. lettotal_rws=not::expr(is_empty_block.expr()) * (cb.curr.state.rw_counter.clone().expr()-1.expr()+1.expr()); FigureC.22: evm_circuit/execution/end_block.rs#L44-L45 • Thelogical and operatorisusedwithanon-Booleanvalue. cb.require_zero("value==0whenis_pad==1forread", and::expr([meta.query_advice(is_pad,Rotation::cur()), meta.query_advice(value,Rotation::cur()),]),); FigureC.23: zkevm-circuits/src/copy_circuit.rs#L322-L328 hasheye 87ScrollzkEVMhalo2CircuitsSecurityAssessment 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 ImproperOpcodeEnforcementRule The ExecutionGadget::configure implementationsmustcheckthattheopcodebeing executedmatchestheexecutionstatethemachineisin.Byusingthe SameContextGadget ,theimplementationimplicitlyenforcesthecorrectopcodeto executionstateconstraint. ThefollowingSemgreprulefinds configure functionsthatdonotproperlyenforce opcodeconstraintsbyfilteringthemostcommonwaysthatthisisvalidated.Itresultsin12 findings,someofwhicharethetruepositiveissuesreportedinfindingTOB-SCROLL-13,as wellassomefalsepositiveresultsthatcanbedismissed. rules: -id:improper-opcode-enforcement message:"configurefunctionwithoutproperopcodeenforcement" languages:[rust] severity:ERROR patterns: -pattern:| fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... } -pattern-not:| fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... let$V=SameContextGadget::construct(...); ... } -pattern-not:| fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... let$V=BlockCtxGadget::construct(...); ... } hasheye 88ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

-pattern-not:| fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... <...cb.require_equal(...,opcode.expr(),...)...>; ... } -pattern-not:| fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... <...cb.require_in_set(...,opcode.expr(),...)...>; ... } -pattern-not:| fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... <...CommonCallGadget::construct(...)...>; ... } -pattern-not:| fnconfigure(cb:&mutConstraintBuilder<F>)→Self{ ... cb.add_lookup($LABEL,Lookup::Fixed{tag: FixedTableTag::ResponsibleOpcode.expr(),values:[..., opcode.expr(), ...],},...); } FigureD.2:The improper-opcode-enforcement.yml rule OpcodeLookupwithinConditionRule Thisruleaimstosearchforthe opcode_lookup functioncalledwithina condition context,insearchofinstancesofTOB-SCROLL-27. rules: -id:opcode-lookup-in-condition message:"CBAPIcallsdonottakeconditionintoaccount" languages:[rust] severity:ERROR patterns: -pattern-either: -pattern:$YY.opcode_lookup(...) -pattern-inside:| <...$XX.condition($COND,...)...> FigureD.3:The opcode-lookup-in-condition.yml rule GateUsageOutsideofacreate_gateContextRule Thisruleaimstosearchforusesofthe gate issuedescribedinTOB-SCROLL-27. hasheye 89ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

rules: -id:gate-usage message:".gate()cannotbeusedoutsidethecreate_gate()function" languages:[rust] severity:ERROR patterns: -pattern:$OBJ.gate(...) -pattern-not-inside:| $META.create_gate($LABEL,|$CB|{ ... }); FigureD.4:The gate-usage.yml rule D.2.Clippy TheRustlinterClippycanbeinstalledusing rustup byrunningthecommand rustup componentaddclippy .Invoking cargoclippy intherootdirectoryoftheprojectruns thetool.RunningClippywith cargoclippy--workspace---Wclippy::pedantic will analyzethecodebasewithadditionallinters. hasheye 90ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

E.FixReviewResults Whenundertakingafixreview,hasheyereviewsthefixesimplementedforissues identifiedintheoriginalreport.Thisworkinvolvesareviewofspecificareasofthesource codeandsystemconfiguration,notcomprehensiveanalysisofthesystem. FromSeptember25toOctober2,2023,hasheyereviewedthefixesandmitigations implementedbytheScrollteamfortheissuesidentifiedinthisreport.Wereviewedeach fixtodetermineitseffectivenessinresolvingtheassociatedissue. ScrollprovidedPRswithfixesforallhigh-severity,medium-severity,andlow-severity findingsexceptforthelow-severityfindingTOB-SCROLL-21.ScrollalsoprovidedPRswith fixesforseveraloftheinformational-severityfindings. Insummary,ofthe29issuesdescribedinthisreport,Scrollhasresolved16issuesandhas partiallyresolvedfourissues.Scrollhasindicatedthatitdoesnotintendtoaddresstwo

issues,whichhavebeenlabeledasunresolved.NofixPRswereprovidedfortheremaining sixissues,sotheirfixstatusesareundetermined.Foradditionalinformation,pleasesee theDetailedFixReviewResultsbelow. IDTitleStatus 1ModGadgetisunderconstrainedandallowsincorrectMULMOD operationstobeproven Resolved 2TheRlpU64Gadgetisunderconstrainedwhenis_lt_128isfalseResolved 3TheBLOCKHASHopcodeisunderconstrainedandallowsthehashof anyblocktobecomputed Resolved 4zkevm-circuitscratedependsonanoutdatedversionofhalo2-eccPartially Resolved 5N_BYTESparametersarenotcheckedtopreventoverflowPartially Resolved 6Differencesinsharedcodebetweenzkevm-circuitsandhalo2-libResolved hasheye 91ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

7UnderconstrainedwarmstatusonCALLopcodesallowsgascost forgery Resolved 8RWtableconstantsmustmatchexactlywhentheverificationkeyis created Undetermined 9TheCREATEandCREATE2opcodescanbecalledwithinastatic context Resolved 10ResponsibleOpcodetableincorrectlyhandlesCREATEandCREATE2Resolved 11EllipticcurveparametersomittedfromFiat-ShamirUnresolved 12ThegascostfortheCALLopcodeisunderconstrainedResolved 13Unconstrainedopcodesallownondeterministicexecution Partially Resolved 14NondeterministicexecutionofReturnDataCopyGadgetand ErrorReturnDataOutOfBoundGadget Resolved 15ManyRWcounterupdatesaremagicnumbersUndetermined 16NativePCSaccumulationdecidersacceptanemptyvectorResolved 17TheErrorOOGSloadSstoreandtheErrorOOGLoggadgetshave redundanttablelookups Undetermined 18TheStatecircuitdoesnotenforcetransactionreceiptconstraintsUndetermined 20TheEXPopcodehasanunusedwitnessResolved 21Thebn_to_fieldfunctionsilentlytruncatesbigintegersUnresolved hasheye 92ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

22Thefield_to_bnfunctiondependsonimplementation-specificdetails oftheunderlyingfield Resolved 23ThevaluesofthebytecodetabletagcolumnarenotconstrainedtobeHEADERorBYTE Resolved 24UnconstrainedcolumnsonthebytecodeHEADERrowsUndetermined 25decompose_limbdoesnotworkasintendedResolved 26ZeromoduluswillcauseapanicResolved 27TheConstraintBuilder::conditionAPIisdangerousUndetermined 28TheEXTCODECOPYopcodeimplementationdoesnotworkwhenthe accountaddressdoesnotexist Resolved hasheye 93ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

DetailedFixReviewResults TOB-SCROLL-1:TheModGadgetisunderconstrainedandallowsincorrectMULMOD operationstobeproven ResolvedinPR#512.Theconstraintsfor a_or_zero havebeenreplacedwitha select call thatcorrectlyforces a_or_zero tobe 0 when n is 0 . TOB-SCROLL-2:TheRlpU64Gadgetisunderconstrainedwhenis_lt_128isfalse ResolvedinPR#615.Thenewconstraintsforce is_lt_128 tobe 1 inthecaseofazero value.If is_lt_128 is 1 ,thecircuitrange-checkstheoriginalvalue.If is_lt_128 is 0 ,the circuitrange-checksavalue v ,definedasfollows:if byte[0] isthemostsignificantbyte, v equals byte[0]-128 ,andifnot, v equals 0 .If byte[0] isthemostsignificantbyte,this sufficestocheckthat byte[0] isintherange [128,256) .If byte[0] isnotthemost significantbyte,otherlogicforcestheretobeanon-zerolimbafterthefirstone,which means value≡256*x+y forsome x>0 , y≥0 .Thus, is_lt_128 is 1 onlyif value isin [0,128) . TOB-SCROLL-3:TheBLOCKHASHopcodeisunderconstrainedandallowsthehashof anyblocktobecomputed ResolvedinPR#512.Thecommented-outlookupfor current_block_number hasbeen uncommented.TheScrollteamshouldevaluatewhetheritisbettertoperformthislookup ortoinsteaddirectlyconstrain current_block_number toequal cb.curr.state.block_number . TOB-SCROLL-4:zkevm-circuitscratedependsonanoutdatedversionofhalo2-ecc PartiallyResolvedasofcommit7fe99fe4e3de14801f4d66f75bd35307de39b0a8in zkevm-circuits andcommit 705881779304003 61c731659b15b2ab3f29f7784 in halo2-lib .The zkevm-circuits nowcratedependsonthe v0.1.5 tagofthe scroll-tech/halo2-lib repository,whichincludesafixfortheECDSAimplementation. However,werecommendatleastupdatingtotheupstreamversion0.3.0,whichincludes manychanges,includingadifferentimplementationofthe scalar_multiply function usedbytheECDSAimplementation. Theupdatedcommitfor halo2-lib containsallupstreamchangeswehighlighted,but doesnotseemtobeuptodatewiththeupstreamversionofthelibrary.Notethatwedid notperformafullsecurityassessmentofthiscommit,sowedonotknowwhichissuesmay stillbepresentinit. TOB-SCROLL-5:N_BYTESparametersarenotcheckedtopreventoverflow PartiallyresolvedinPR#512. assert!() callshavebeenaddedtoconstrainthe N_BYTES parameter.Theexpressionin constant_division.rs ,showninfigureE.1,mayoverflow whencompiledwithoutoverflowchecksandmayincorrectlyallowextremelylargevalues of N_BYTES .Notethatthe zkevm-circuits repositoryconfiguresitsreleasebuildto enableoverflowchecks. hasheye 94ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

assert!(N_BYTES*8+64-denominator.leading_zeros()asusize ⩽ MAX_N_BYTES_INTEGER*8);
FigureE.1:Theexpressionthatmayoverflowwhencompiledwithoutoverflowchecks TOB-SCROLL-6:Differencesinsharedcodebetweenzkevm-circuitsandhalo2-lib ResolvedinPR#709andPR#1001.Various debug_assert!() callshavebeenreplaced with assert!() calls,andtheincorrect log2_ceil functionhasbeenfixed. TOB-SCROLL-7:UnderconstrainedwarmstatusonCALLopcodesallowsgascost forgery ResolvedinPR#512,withsomeadditionalfixesaddedinPR#676.PR#512addsa constraintto CallOpGadget forcing is_warm tobetrue,buttheinitialvalueof is_warm is notdirectlyconstrained.Theinitialvalueforallaccesslistreadsisfalse,ashighlightedin figureE.2,whiletheEthereumYellowPaperstatesthatitshouldbetrueforprecompile addresses,asshowninfigureE.3: fnbuild_tx_access_list_account_constraints(&mutself,q:&Queries<F>){ self.require_zero("field_tagis0forTxAccessListAccount",q.field_tag()); self.require_zero("storage_keyis0forTxAccessListAccount", q.rw_table.storage_key.clone(),); self.require_boolean("TxAccessListAccountvalueisboolean",q.value()); self.require_zero("initialTxAccessListAccountvalueisfalse", q.initial_value(),); self.require_equal("state_rootisunchangedforTxAccessListAccount", q.state_root(), q.state_root_prev(),); self.condition(q.not_first_access.clone(),|cb|{ cb.require_equal("valuecolumnatRotation::prev()equalsvalue_prevatRotation::cur()", q.rw_table.value_prev.clone(), q.value_prev_column(),); }); } FigureE.2:Basicconstraintsfortheaccesslist hasheye 95ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

FigureE.3:AnexcerptfromtheEthereumYellowPaperspecifyingtheinitialaccesslist TheinitialprecompileaccessvalueshavebeenfixedinPR#676byaddinganaccesslist writeforeachprecompiletothe BeginTx state. TOB-SCROLL-8:RWtableconstantsmustmatchexactlywhentheverificationkeyis created Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-SCROLL-9:TheCREATEandCREATE2opcodescanbecalledwithinastatic context ResolvedinPR#512.The CreateGadget::configure methodnowperformsa call_context lookuptoretrievethe IsStatic fieldandthenconstrainsthatresulttobe zero. TOB-SCROLL-10:ResponsibleOpcodetableincorrectlyhandlesCREATEandCREATE2 ResolvedinPR#512.Theresponsible-opcodetablehasbeenupdatedtomap ExecutionState::CREATE to OpcodeId::CREATE and ExecutionState::CREATE2 to OpcodeId::CREATE2 . TOB-SCROLL-11:EllipticcurveparametersomittedfromFiat-Shamir Unresolved.TheScrollteamhasindicatedthatitdoesnotintendtofixthisissue. TOB-SCROLL-12:ThegascostfortheCALLopcodeisunderconstrained ResolvedinPR#774.Previously,whencallinganemptyaddressoraprecompile,the unconstrainedcell step_gas_cost wasusedtodeterminethegascostofthe CALL opcode.Now,inallcases,the gas_left fieldof StepStateTransition valuesisderived fromthefullyconstrainedcells callee_gas_left , gas_cost ,and call_gadget.has_value .Wedidnotevaluatewhethertheoverallgascalculationis correctinthisfixreview,butitisconstrained. TOB-SCROLL-13:Unconstrainedopcodesallownondeterministicexecution PartiallyresolvedinPR#633andPR#736.InPR#633,allmentionedgadgetsnowconstrain theopcode.Scrollshouldinspectandtest ErrorPrecompileFailed asdevelopment hasheye 96ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

continuestoensurethatonlyprecompilecallscantriggeritandonlywhentheyfail.PR #736addsconstraintstothe ReturnRevertGadget componenttoensurethat REVERT opcodesareaccompaniedbyareversionintheRWtable.Wewerenotabletodetermine fromthesediffswhetheritispossibletohaveareversionwhileexecutinga RETURN opcode.Scrollshouldinspectandtestthiscomponenttoensurethatamaliciousprover cannotmanipulatethebehaviorofthe RETURN opcode. TOB-SCROLL-14:NondeterministicexecutionofReturnDataCopyGadgetand ErrorReturnDataOutOfBoundGadget ResolvedinPR#661.Theoverflow-relatedvalidationlogicof ReturnDataCopyGadget and ErrorReturnDataOutOfBoundGadget hasbeenfactoredoutintoacommoncomponent, CommonReturnDataCopyGadget .Thiscommongadgetforcesthevalidationtosucceedor failbasedonthe is_overflow parameter.Notethat is_overflow issetto 1.expr() in onecaseand false.expr() intheother.Whiletheseexpressionsdobehavecorrectly,we recommendthattheScrollteammakethesesymmetrical;thatis,either 1.expr() and 0.expr() ,or true.expr() and false.expr() . TOB-SCROLL-15:ManyRWcounterupdatesaremagicnumbers Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-SCROLL-16:NativePCSaccumulationdecidersacceptanemptyvector ResolvedinPR#17.Thecodenowpanicswithanassertionfailureifpassedanempty vector. TOB-SCROLL-17:TheErrorOOGSloadSstoreandtheErrorOOGLoggadgetshave redundanttablelookups Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-SCROLL-18:TheStatecircuitdoesnotenforcetransactionreceiptconstraints Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-

SCROLL-20:TheEXPopcodehasanunusedwitness ResolvedinPR#838.The zero_rlc fieldhasbeenremoved. TOB-SCROLL-21:Thebn_to_fieldfunctionsilentlytruncatesbigintegers Unresolved.TheScrollteamhasindicatedthatitdoesnotintendtofixthisissue. TOB-SCROLL-22:Thefield_to_bnfunctiondependsonimplementation-specificdetails oftheunderlyingfield hasheye 97ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

ResolvedinPR#15.Atesthasbeenaddedtoconfirmthatthedatarepresentationmatches theexpectationsoftheimplementation.Scrollshouldensurethatthistestrunsbeforeany deployment. TOB-SCROLL-23:Thevaluesofthebytecodetabletagcolumnarenotconstrainedto beHEADERorBYTE ResolvedinPR#681.ABooleanconstrainthasbeenaddedtothe tag column. TOB-SCROLL-24:UnconstrainedcolumnsonthebytecodeHEADERrows Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-SCROLL-25:decompose_limbdoesnotworkasintended Resolvedasofcommit 483feb2e4554fcab58878d7c8e6a6f8be792e2f2 .The decompose_limb methodhasbeenmorecompletelyimplementedandappearstonow iteratecorrectlythroughthelimbs.Wedidnotfullyreviewthecorrectnessofthis implementation,andtheredonotappeartobeanydirecttestsforthisimplementation. Scrollshouldaddteststoensureitscorrectbehavior. TOB-SCROLL-26:Zeromoduluswillcauseapanic PartiallyresolvedinPR#4,thenfullyresolvedinPR#12.The mod_mult methodhasbeen modifiedtousethe number_is_zero methodtoreturn 0 whenthe modulus parameteris 0 .However,theoriginalfixtothe number_is_zero method,showninfigureE.4, introducesanewerrorbyincorrectlychecking number.limbs[0] threetimes. //return0ifnotzero,1ifzerofornumber pubfnnumber_is_zero(&self, region:&mutRegion<F>, range_check_chip:&mutRangeCheckChip<F>, offset:&mutusize, number:&Number<F>,)->Result<Limb<F>,Error>{ letzero=F::zero(); letthree=F::from(3u64); //limb0_zerois0ifnotzero,1ifzero letlimb0_zero= self.config .eq_constant(region,range_check_chip,offset,&number.limbs[0], &zero)?; letlimb1_zero= self.config .eq_constant(region,range_check_chip,offset,&number.limbs[0], &zero)?; letlimb2_zero= hasheye 98ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

self.config .eq_constant(region,range_check_chip,offset,&number.limbs[0], &zero)?; FigureE.4:Theincorrectindexinginto number.limbs (misc-precompiled-circuit/src/circuits/modexp.rs#525-545) TheScrollteamhasfixedthisnewlyintroducedissueinPR#12,andthispreventsazero moduluscausingapanicbyreplacingitwiththevalue1whendividing. Althoughthisparticularissuehasbeenresolved,theScrollteamshouldinspectandtest thiscomponentcarefully,asithashadseveralcorrectnessproblems. Inparticular,werecommendthatScrollinvestigatewhetherthesemethodsbehave correctlyinthepresenceofmalformedvaluesofthe Number type.The Number type,shown infigureE.5,appearstocontaina“CRT-style”representationofalargenumber,similarto the CRTInteger typein halo2-ecc .Thefirstthreeentriesofits limbs arraycontainthe “truncation”part,andthefourthentrycontainsthe“native”part.Thisrepresentationisnot documentedinthecode,andwehavenotdeterminedwhetheritispossibletocreatea malformed Number valuewithinthe modexp circuit. #[derive(Clone,Debug)] pubstructNumber<F:FieldExt>{ limbs: [Limb<F>;4], } FigureE.5:The Number type (misc-precompiled-circuit/src/circuits/modexp.rs#23-26) TOB-SCROLL-27:TheConstraintBuilder::conditionAPIisdangerous Undetermined.Nofixwasprovidedforthisissue,sowedonotknowwhetherthisissue hasbeenaddressed. TOB-SCROLL-28:TheEXTCODECOPYopcodeimplementationdoesnotworkwhenthe accountaddressdoesnotexist ResolvedinPR#846.Conditionalconstraintshavebeenaddedtoexplicitlyhandleaccounts withnocode,asindicatedbyazerovalueintheir CodeHash field.Whenanaccounthasno code,thecodesizeisforcedtobezero,andwhenanaccounthascode,thecodesizeis determinedbya bytecode_lookup .Wedidnotfindanytestsforthisbehavior.We recommendthattheScrollteamaddteststoensurethatthisbehavioriscorrectand remainscorrectduringongoingdevelopment. hasheye 99ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC

F.FixReviewStatusCategories Thefollowingtabledescribesthestatusesusedtoindicatewhetheranissuehasbeen sufficientlyaddressed. FixStatus StatusDescription UndeterminedThestatusoftheissuewasnotdeterminedduringthisengagement. UnresolvedTheissuepersistsandhasnotbeenresolved. PartiallyResolvedTheissuepersistsbuthasbeenpartiallyresolved. ResolvedTheissuehasbeensufficientlyresolved. hasheye 100ScrollzkEVMhalo2CircuitsSecurityAssessment PUBLIC