

RISC V and Security: How, When and Why

Helena Handschuh
Rambus Security Technologies Fellow
RISCV Security Standing Committee Chair

CHES 2019 @ Atlanta 08/26/2019



Rambus
Data • Faster • Safer

Outline

- RISC-V Foundation
- Security Standing Committee Creation and Charter
 - Security Task Group Charters and status update
 - Crypto Extensions TG
 - Trusted Execution Environment TG
 - Taxonomy and related DARPA SSITH activities
 - Speaker Program
- Academic and industry initiatives around RISC-V
- Open problems and research directions



- RISC-V (pronounced “risk-five”) is a free and open ISA enabling a new era of processor innovation through open standard collaboration.
 - Founded in 2015
 - 300+ member organizations and individual members
 - open, collaborative community of software and hardware innovators
 - RISC-V base ISA was born in academia and research (Berkeley)
 - A new level of free, extensible software and hardware freedom on architecture
 - Paving the way for the next 50 years of computing design and innovation.
 - Members of the RISC-V Foundation have access to and participate in the development of the RISC-V ISA specifications and extensions and related HW / SW ecosystem.

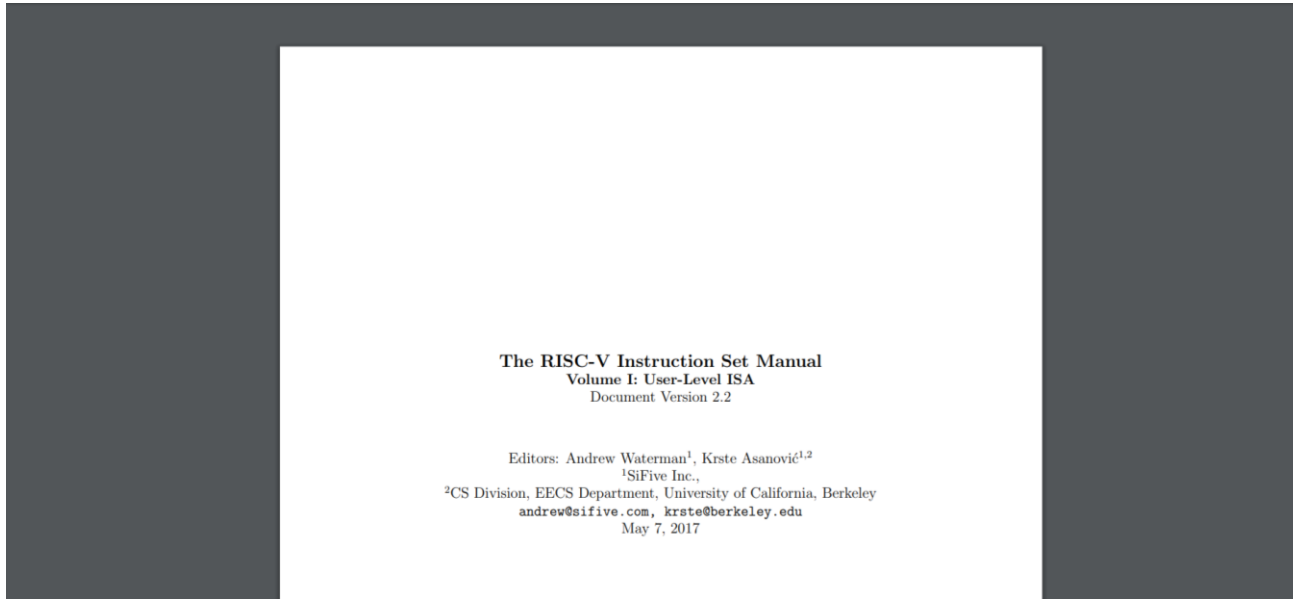
<http://riscv.org>



- RISC-V (pronounced “risk-five”) is a **free** and **open** ISA enabling a new era of processor innovation through open standard collaboration.
 - Founded in 2015
 - 300+ member organizations and individual members
 - **open**, collaborative community of software and hardware innovators
 - RISC-V base ISA was born in academia and research (Berkeley)
 - A new level of **free**, extensible software and hardware **freedom** on architecture
 - Paving the way for the next 50 years of computing design and innovation.
 - Members of the RISC-V Foundation have access to and participate in the development of the RISC-V ISA specifications and extensions and related HW / SW ecosystem.

<http://riscv.org>

“The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 2.2”, Editors Andrew Waterman and Krste Asanovi´c, RISC-V Foundation, May 2017.



- Creative Commons Attribution 4.0 International License.
- This document is a derivative of “The RISC-V Instruction Set Manual, Volume I: User-Level ISA Version 2.1” released under the following license: c 2010–2017 **Andrew Waterman, Yunsup Lee, David Patterson, Krste Asanovi´c**. Creative Commons Attribution 4.0 International License.

RISCV Base Instruction Set Architecture and its Extensions

3 / 145

Preface

This is version 2.2 of the document describing the RISC-V user-level architecture. The document contains the following versions of the RISC-V ISA modules:

Base	Version	Frozen?
RV32I	2.0	Y
RV32E	1.9	N
RV64I	2.0	Y
RV128I	1.7	N
Extension	Version	Frozen?
M	2.0	Y
A	2.0	Y
F	2.0	Y
D	2.0	Y
Q	2.0	Y
L	0.0	N
C	2.0	Y
B	0.0	N
J	0.0	N
T	0.0	N
P	0.1	N
V	0.2	N
N	1.1	N

- Base ISA:
 - 32 bit
 - 32 bit (Embedded)
 - 64 bit
 - 128 bit
- Extensions:
 - M: Multiplication/division
 - A: Atomic instructions
 - F: Single Precision Floating Point
 - D: 2P Floating Point
 - Q: 4P Floating Point
 - L: Decimal Floating Point
 - C: Compressed Instructions
 - B: Bit Manipulation
 - ...
 - V: Vectors Extensions
 - ...

“The RISC-V Instruction Set Manual, Volume I: **User-Level ISA**, Document Version **20190608-Base-Ratified**”, Editors Andrew Waterman and Krste Asanović, RISC-V Foundation, March 2019. Creative Commons Attribution 4.0 International License.

The RISC-V Instruction Set Manual
Volume I: Unprivileged ISA
Document Version 20190608-Base-Ratified

Editors: Andrew Waterman¹, Krste Asanović^{1,2}

¹SiFive Inc.,

²CS Division, EECS Department, University of California, Berkeley
andrew@sifive.com, krste@berkeley.edu

June 8, 2019



145 → 236 pages; shows **Ratified** parts; additional extensions

Page: 4 of 236 50%

Preface

This document describes the RISC-V unprivileged architecture.

The RVWMO memory model has been ratified at this time. The ISA modules marked **Ratified**, have been ratified at this time. The modules marked *Frozen* are not expected to change significantly before being put up for ratification. The modules marked *Draft* are expected to change before ratification.

The document contains the following versions of the RISC-V ISA modules:

Base	Version	Status
RVWMO	2.0	Ratified
RV32I	2.1	Ratified
RV64I	2.1	Ratified
RV32E	1.9	Draft
RV128I	1.7	Draft

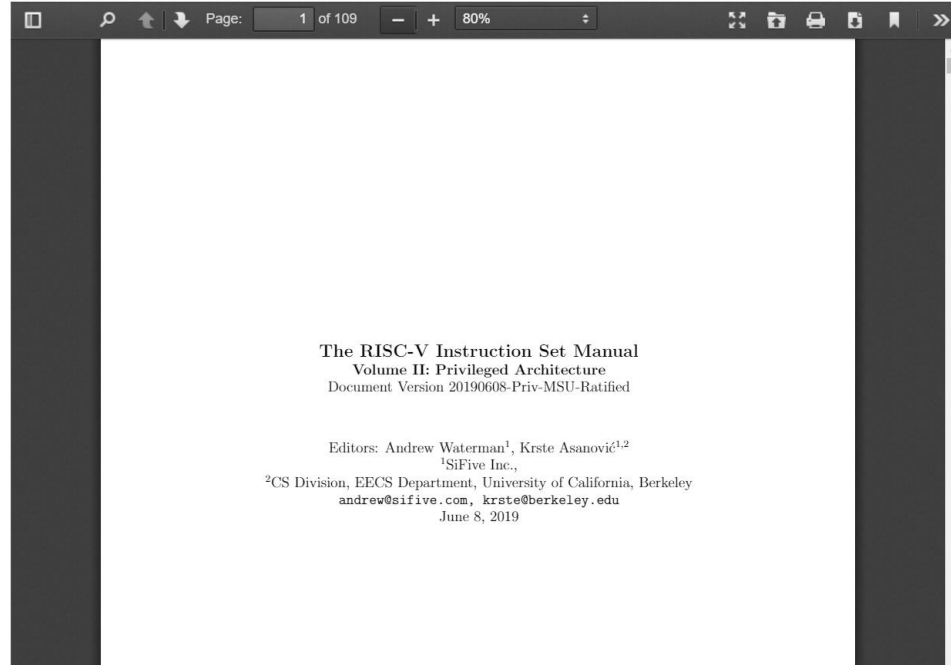
Extension	Version	Status
ZifenceI	2.0	Ratified
Zicsr	2.0	Ratified
M	2.0	Ratified
A	2.0	Frozen
F	2.2	Ratified
D	2.2	Ratified
Q	2.2	Ratified
C	2.0	Ratified
Ziso	0.1	Frozen
COUNTERS	2.0	Draft
L	0.0	Draft
B	0.0	Draft
J	0.0	Draft
T	0.0	Draft
P	0.2	Draft
V	0.7	Draft
N	1.1	Draft
Zam	0.1	Draft

The changes in this version of the document include:

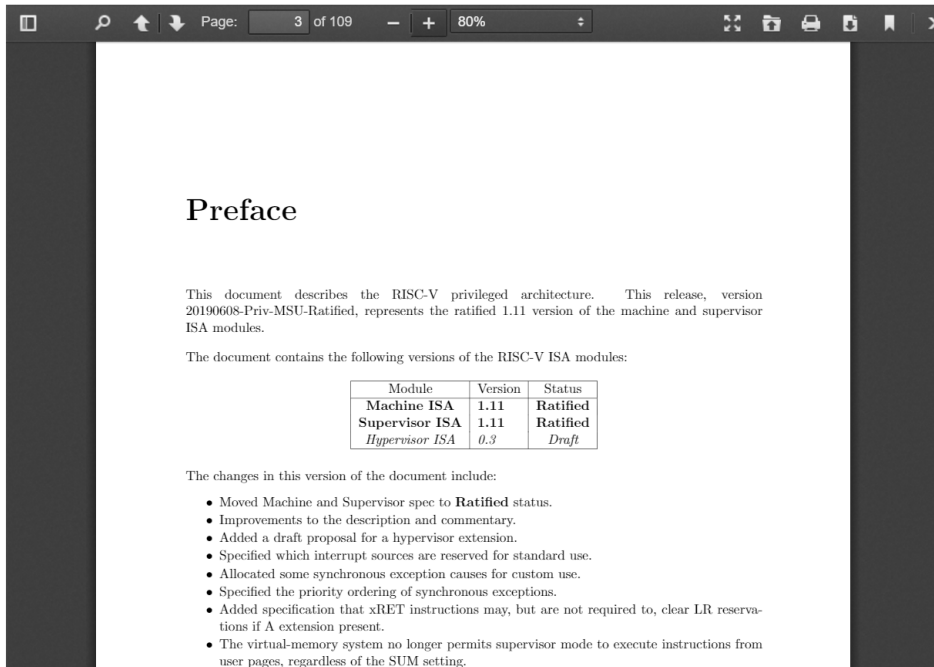
- Moved description to **Ratified** for the ISA modules ratified by the board in early 2019.

i

“The RISC-V Instruction Set Manual, Volume II: **Privileged Architecture**, Document Version **20190608-Priv-MSU-Ratified**”, Editors Andrew Waterman and Krste Asanović, RISC-V Foundation, June 2019.



Defines Machine, Supervisor and Hypervisor modes



Page: 3 of 109 80%

Preface

This document describes the RISC-V privileged architecture. This release, version 20190608-Priv-MSU-Ratified, represents the ratified 1.11 version of the machine and supervisor ISA modules.

The document contains the following versions of the RISC-V ISA modules:

Module	Version	Status
Machine ISA	1.11	Ratified
Supervisor ISA	1.11	Ratified
<i>Hypervisor ISA</i>	<i>0.3</i>	<i>Draft</i>

The changes in this version of the document include:

- Moved Machine and Supervisor spec to **Ratified** status.
- Improvements to the description and commentary.
- Added a draft proposal for a hypervisor extension.
- Specified which interrupt sources are reserved for standard use.
- Allocated some synchronous exception causes for custom use.
- Specified the priority ordering of synchronous exceptions.
- Added specification that xRET instructions may, but are not required to, clear LR reservations if A extension present.
- The virtual-memory system no longer permits supervisor mode to execute instructions from user pages, regardless of the SUM setting.

RISC V Cores / SoCs

- 65 cores available here:
- <https://riscv.org/risc-v-cores/>
- Note that none of these cores/SoCs have passed the **in-development** RISC-V compliance suite.

WEBSITE	Type: Cores Supplier: Darklife User spec: most of RV32I License: BSD Primary Language: Verilog Bit Processor: 32	Core: E31 ISA: RV32IMAC OS Capability: RTOS Bit Processor: 32 DevKit: HiFive1 Availability: public since 2016Q4
D25F Type: Cores Supplier: Andes Priv. spec: 1.11 User spec: RV32GCP + Andes V5 ext. License: Andes Commercial License Primary Language: Verilog Bit Processor: 32	GITHUB	DATASHEET
WEBSITE	freedom Type: Cores Supplier: SiFive Priv. spec: 1.11-draft User spec: 2.3-draft License: BSD Primary Language: Chisel	Freedom U540 Type: SoCs Supplier: SiFive Core: U54 (4 cores), E51 (1 management core) ISA: RV64GC (application cores), RV64IMAC (management core) OS Capability: Linux Bit Processor: 64 DevKit: HiFive Unleashed development board Availability: public since 2018Q1
FE310-G002 Type: SoCs Supplier: SiFive Core: E31 ISA: RV32IMAC OS Capability: RTOS	GITHUB	PRODUCT PAGE
	GAP8 Type: SoCs Supplier: GreenWaves	

RISC-V Software Ecosystem Overview

- [Simulators](#)
- [Object toolchain](#)
- [Debugging](#)
- [C compilers and libraries](#)
- [Boot loaders and monitors](#)
- [OS and OS kernels](#)
- [Compilers and runtimes for other languages](#)
- [IDEs](#)
- [..... Security \(!\)](#)

Security

Name	Links	License	Maintainers
Hex Five Security	SDK	Proprietary	Hex Five Security Inc.
Keystone Enclave	Website, Repositories	BSD 3-clause	Keystone Team

January 2018... the o..s... moment



January 2018... the oops(!) moment

Cache-timing side-channels



Branch prediction



Speculative execution

Creation of the RISC-V Security Standing Committee

- July 2, 2018
- [“RISC-V Foundation Announces Security Standing Committee, Calls Industry To Join In Efforts”](#)
 - **“Security is one of the fundamental issues in our connected world.** The RISC-V community is committed to pushing the industry forward through innovative approaches and new thinking to **address existing and emerging threats”** (Helena)
 - **“It is an exciting time to witness the advent of a new compute platform that has formal methods at its foundation for processor correctness and security,”** ...“RISC-V is a simple, free and open ISA that is an ideal vehicle for research in **formally assured security and secure hardware development** for everything from consumer devices to national security applications.” (Joe Kiniry)

Security Standing Committee

chair: Helena Handschuh, Rambus

vice-chair: Joe Kiniry, Galois

website: <https://lists.riscv.org>

meetings roughly every other week, alternating between “Speaker Program” and “Business Meeting”

- Security Standing Committee Charter:
 - Promote RISC-V as an ideal vehicle for the **security community**
 - Liaise with other internal RISC V committees and with external security committees
 - Create an information repository on new attack trends, threats and countermeasures
 - Identify **top 10 open challenges** in security for the RISC-V community to address
 - Propose security committees (Marketing or Technical) to tackle specific security topics
 - **Recruit security talent** to the RISC-V ecosystem (e.g., into committees)
 - Develop consensus around **best security practices** for IoT devices and embedded systems

Cryptographic Extensions Task Group

Chair: Richard Newell, Microchip, Vice-chair: Derek Atkins, SecureRF

- **Charter:**

- propose ISA extensions to the vector extensions for the standardized and secure execution of popular cryptography algorithms.
- To ensure that processor implementers are able to support a wide range of performance and security levels the committee will create a base and an extended specification.
- The base will be comprised of low-cost instructions that are useful for the acceleration of common algorithms.
- The extended specification will include greater functionality, reserve encodings for more algorithms, and will facilitate improved security of execution and higher performance.
- The scope will include symmetric and asymmetric cryptographic algorithms and related primitives such as message digests. The committee will also make ISA proposals regarding the use of random bits and secure key management.

Cryptographic Extensions Task Group

Chair: Richard Newell, Microchip, Vice-chair: Derek Atkins, SecureRF

- Approach based on vector extensions
- AES instructions
 - 128, 192, 256; done
- SHA-2 instructions
 - SHA-256 and SHA-512; almost done
- Need to convert AES and SHA-2 into formal specs now...
- Prototyping Public Key Crypto algorithms
 - Long integer arithmetic
 - Implementation proof of concept
- Future directions:
 - More light-weight approach: could recommend subset of vector extensions only
 - XCrypto (Bristol): proposed scalar instructions, rotates, etc. to have SW run faster
 - Paris Telecom also interested in same type of research

Trusted Execution Environment Task Group

Chair: Joe Xie, Nvidia Vice-chair: Nick Kossifidis, Forth

Charter:

- To define an **architecture specification** to support trusted execution environment for RISC-V processors
- To provide necessary **implementation guidelines and/or recommendations** to assist hardware developers to realize the specification
- To enable the development of **necessary components, such as compiler, simulation model, hardware, and software components** to support the specification

Trusted Execution Environment Task Group

Chair: Joe Xie, Nvidia Vice-chair: Nick Kossifidis, Forth

- HW:
 - PMP Physical Memory Protection based on Privilege spec 1.12
 - IO PMP proposal 0.1
 - Next: Control Flow Integrity (CFI) ext.
- SW:
 - Secure Monitor architecture
 - Secure boot architecture: signature verification + optional extensions for key management, certs, revocation, attestation
 - TEE APIs: OS-TA, App-TA, TA-TA, TA-SecMon, Attestation of a TA, TEE/TA Mgmt.

Taxonomy and related DARPA SSITH activities

SSC Vice-Chair: Joe Kiniry, Galois

- “Lando” : a formal specification language for HW design with 4 sublanguages:
 - A system spec language
 - Architecture language
 - Product line engineering language
 - Security property specification language
- Domain Model for specifying security properties.
 - Ex: formalization of the NIST CWEs related to buffer/memory errors

Taxonomy and related DARPA SSITH activities

SSC Vice-Chair: Joe Kiniry, Galois

- BESSPIN: a tool suite for formal reasoning
 - GRIFT: subsystem of tool suite already contributed to RISC-V Formal TG
- Platform specs and security-enriched ISA:
 - Secure voting machine platform spec includes security properties/guarantees
 - Built on RISC-V; demonstrated @ Defcon this month
 - 6 other platform specs based on RISC-V SoCs
 - Rocket, Boom, Piccolo, Flute, Bassoon, Riscy

SSC Speaker Program

- Gernot Heiser, Data61 on Timing Attacks and Augmented ISA
- Dayeol Lee, Berkeley on the Keystone project (TEE framework)
- Jose Renau, Esperanto on Timing Attack Mitigation Ideas
- Jon Geater, Thales on insights into Trustzone and TEEs
- Nicole Fern, Tortuga Logic on Security-Oriented Verification Tools
- Daniel Genkin on Foreshadow
- Stefan Mangard, IAIK Graz on ISA extensions (SCA, CFI, secure memory access)
- Ben Marshall, Bristol on Xcrypto ISA extensions
- **Your name here**

Other ongoing security initiatives: security contests

- Hack@Dac2018 (independent from RISC-V Foundation)
 - HW bug hunting;
 - Systematic bug construction for bug hunting
 - Organized on RISC-V processors
 - Some “native” bugs were also found in some RISC-V processors
 - Results available online

Other ongoing security initiatives: security contests

<https://riscv.org/2019/07/risc-v-softcpu-core-contest/>

- Thales and Microchip announced a hackathon on July 15th:
 - Soft core (verilog) running on a Microchip FPGA; opensource submissions
 - Contest rules on the website riscv.org; propose security countermeasures
 - Based on Zephyr; can make limited changes to the compiler
 - **September 15th** deadline for submissions
- Need to protect against 5 very classical attacks:
 - Corrupting a function pointer on the heap
 - Buffer overflow on the stack to corrupt longjump buffer
 - Buffer overflow on the stack to change the return address
 - Corrupting a function pointer on the stack
 - Corrupting a C structure holding a function pointer

Academic and national lab initiatives

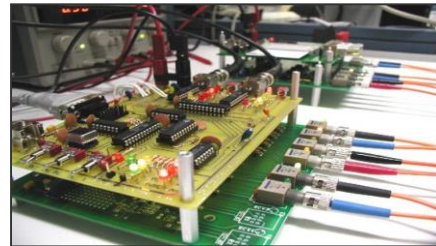
- RISC-V extensions for security
 - IAIK Graz: side-channel attacks, control flow integrity, secure memory access
 - U Bristol: XCrypto ISA extensions
 - Telecom ParisTech expressed interest
 - ...
- CEA Leti: secure processor with authenticated/encrypted resources
 - Busses, memories, datapath, instruction path
 - See RISC-V Zurich workshop talk

Secure Enclaves initiatives based on RISC-V

- Sanctum, MIT and Keystone, Berkeley
- OpenTitan, Google
- Multizone TEE API, HexFive
- CryptoManager Root of Trust, Rambus
 - see demo next door at coffee/lunch
 - Joel Wittenauer invited talk at FDTC 2019
- DPA-resistant RISC-V CPU, Rambus
 - Mike Hutter, Elke DeMulder, Samatha Gummalla
 - RISC-V Summit 2018, DAC 2019 invited talk, Lorentz WS 2019 (next month)

Open problems: How to mitigate micro-architectural flaws?

- Problem statement:



- Side-channel Information leakage:

- Power attacks (1999), electro-magnetic attacks (1999), differential fault attacks (1997),...

- More recently discovered cross-layer exploits:

- Spectre (2018), Meltdown (2018), Foreshadow (2019)
- Spoiler, TLBleed, CacheBleed, RowHammer, CLKScrew
- More to come...



- How to address: at ISA level? Platform spec level?
- How to do better than proprietary micro-architectures?

Open problems: Cache timing side-channels mitigation

Some existing proposals:

- *Augmented ISA* – Gernot Heiser Data61
 - (cache) Flush instructions, memory partitioning instructions
- *Timing attack mitigations* – Jose Renau Esperanto
 - Security classification tags
- *Speculative Taint Tracking (STT)* – Chris Fletcher Univ Illinois
 - Tainted registers and corresponding update policies
 - Secure Enclaves Workshop, Berkeley 2019

Security Certification/Assurance for RISC-V based systems?

- RISC-V Formal specs and Compliance test suite: “functional spec compliance”
- How to certify for security when most security aspects are micro-architecture and implementation related and not ISA related?
- What (if any) security levels should be defined?
- SESIP (NXP, Brightsight, GlobalPlatform)?
- PSA (ARM)?
- Open-source versus Security Certification?
 - Loosing points in CC JHAS/JIL table?
 - Open-source versus obscurity?
- Formal verification of HW security properties
 - Lando, Besspin, Grift
- HW verification tools or SW source code analysis which trace and enforce security properties
 - Not just RISC-V-related
 - Tortuga Logic Radix, FortifyIQ TracerIQ, Secure-IC Virtualyzr, Riscure True Code



RISCV and Post-quantum Crypto?

- Address at vector extension level?
- Specific MatrixVectorMul? InnerProduct?
- Is this necessary? useful? sufficient?
- Any specific PQ extensions for lattices, codes, supersingular isogenies?

Type	Algorithm	Key Strength Classic (bits)	Key Strength Quantum (bits)	Quantum Attack
Asymmetric	RSA 2048	112	0	Shor's Algorithm
	RSA 3072	128		
	ECC256	128		
	ECC 521	256		
Symmetric	AES128	128	64	Grover's Algorithm
	AES 256	256	128	

Conclusion

- Open source approach is great
- Many new opportunities
- Thriving RISC-V ecosystem

- How to address security in the RISC-V world is still a challenging question
 - Most serious security issues result from micro-arch flaws
 - Many good ideas and initiatives already
 - Still many open problems to work on



Call to action!





Thank you!

<https://riscv.org>

Helena.Handschuh@cryptography.com