RiSE

Building an Internet of Things We Can Trust

Ben Zorn

Principal Researcher and Research Manager

Research in Software Engineering (RiSE) Group

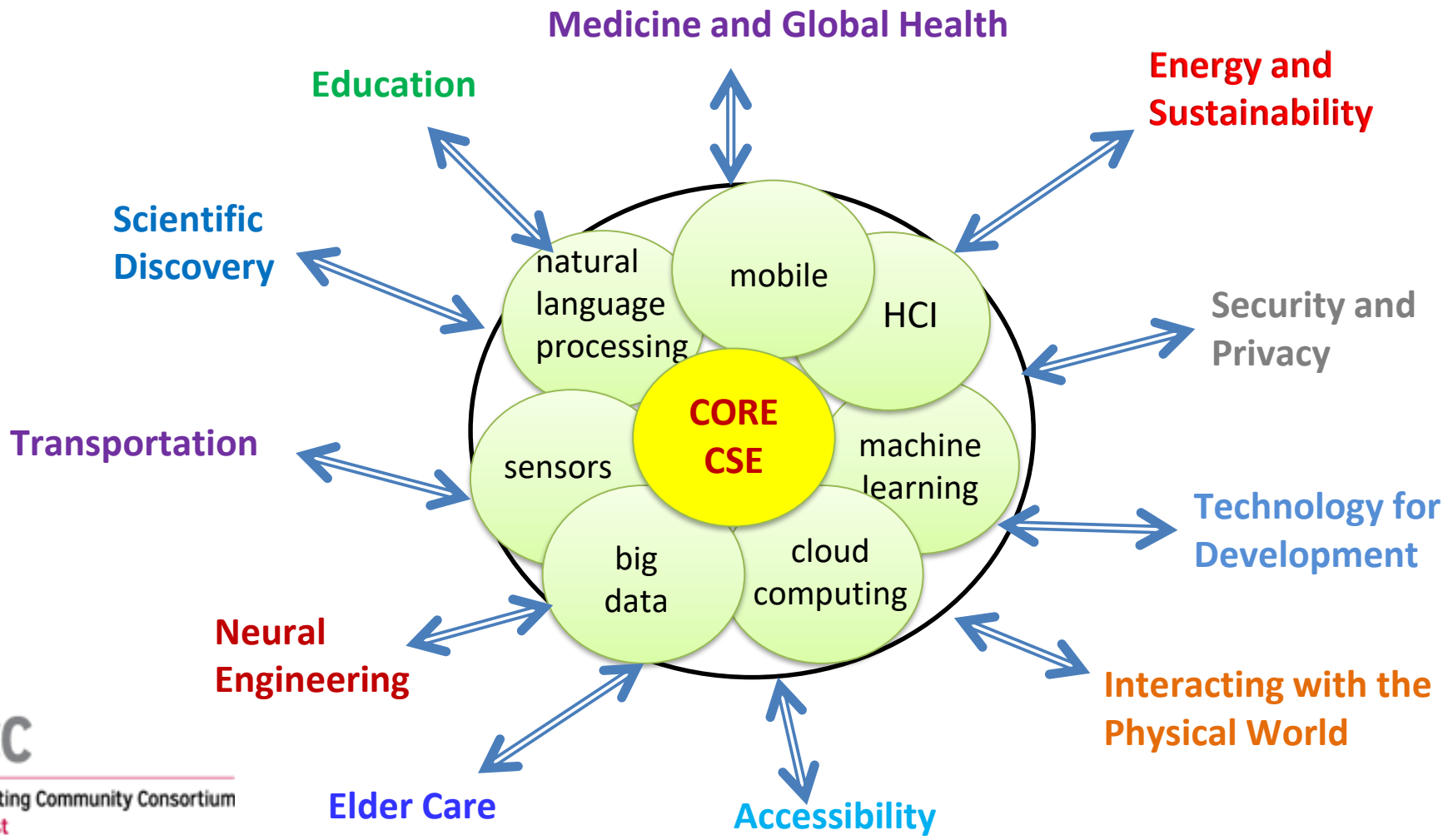
Microsoft Research, Redmond

Microsoft

RiSE – Research in Software Engineering

http://research.microsoft.com/en-us/groups/rise/

Rapidly Expanding World of Computing



Graphic: Lazowska

Zorn, UWMSR 2017

RiSE

Toyota Hires All-Star Team for Robotics and Artificial Intelli

by Jason Si

Towards the social media enabled jet engine

September 26, 2012

Posted by Paul Wallbank at 10:02 am

Add comments

Mother Jones | POLITICS | ENVIRONMENT | CULTURE | PHOTO ESSAYS

Must Reads: $15 an Hour Isn't the Answer | Why You Shouldn't Bet on Fossil Fuels | Look Who's Running Burma

ENVIRONMENT

→ *Climate Change, Climate Desk, Corporations, Food and Ag, Tech, Top Stories*

Monsanto Is Using Big Data to Take Over the World

The GMO giant wants to help you beat climate change...with your phone.

—*By* ***Tim McDonnell*** | Wed Nov. 19, 2014 6:00 AM EST

Implication:
Every company is a software company

Microsoft

Zorn, UWMSR 2017

Yes, this is a computer too

Amazon Echo

Smart Meter

Every object is a computer

Ring.com

Nest

Microsoft

Zorn, UWMSR 2017

IoT Devices Cause Unintended Consequences

https://www.us-cert.gov/ncas/alerts/TA16-288A

US-CERT

UNITED STATES COMPUTER EMERGENCY READINESS TEAM

Mirai malware used to create 380,000 node device botnet

ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

RISK ASSESSMENT —

Record-breaking DDoS reportedly delivered by >145k hacked cameras

Once unthinkable, 1 terabit attacks may soon be the new normal.

DAN GOODIN - 9/28/2016, 5:50 PM

Botnet was leveraged to deliver massive DDOS attack on KrebsOnSecurity

http://arstechnica.com/security/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever/

Did I Mention Companies Cheat?

MIT Technology Review

A View from **David Zax**

Many Cars Have a Million Lines of Co

Who gets to write it?

December 3, 2012

MIT Technology Review

You can hide a lot
in 100M lines of code

ndal

e that was

How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles

Moritz Contag*, Guo Li†, Andre Pawlowski*, Felix Domke‡,
Kirill Levchenko†, Thorsten Holz*, and Stefan Savage†

* Ruhr-Universität Bochum, Germany, {moritz.contag, andre.pawlowski, thorsten.holz}@rub.de
† University of California, San Diego, {gul027, klevchen, savage}@cs.ucsd.edu
‡ tmbinc@elitedvb.net

https://cseweb.ucsd.edu/~klevchen/diesel-sp17.pdf

Takeaway: Creating a trustworthy IoT is hard…
and we don’t even do the easy stuff well.

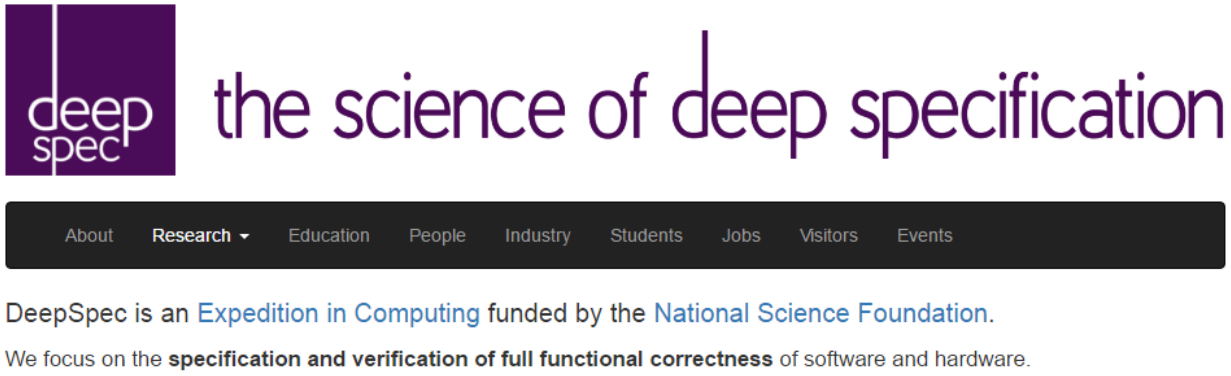
The Cathedral and the Skyscraper

Heroic effort, amazing engineering, one of a kind…

Stronger materials, reusable components, mathematical analysis…

Trust…. But **Verify**
Two Expeditions: DeepSpec and Everest

Scalable reasoning meets software verification at scale



http://deepspec.org/

$10M NSF Expedition in Computing
Awarded 2016

https://project-everest.github.io/

Zorn, UWMSR 2017

Weaknesses in Critical Infrastructure Components: HTTPS

2014

ROBERT MCMILLAN BUSINESS 04.11.14 6:30 AM

HOW HEARTBLEED BROKE THE INTERNET — AND WHY IT CAN HAPPEN AGAIN

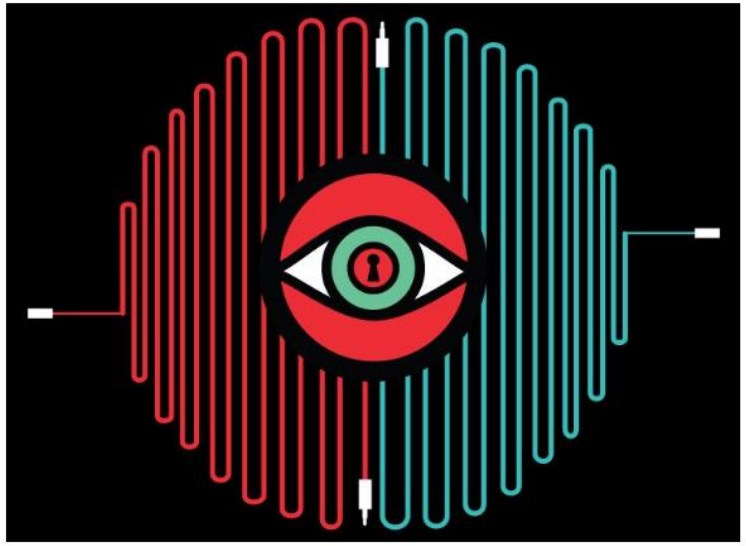


Illustration: Ross Patton/WIRED

https://www.wired.com/2014/04/heartbleedslesson/

The DROWN Attack

Paper | Q&A

DROWN is a serious vulnerability that affects HTTPS and other services that rely on SSL and TLS, some of the essential cryptographic protocols for Internet security. These protocols allow everyone on the Internet to browse the web, use email, shop online, and send instant messages without third-parties being able to read the communication.

DROWN allows attackers to break the encryption and read or steal sensitive communications, including passwords, credit card numbers, trade secrets, or financial data. Our measurements indicate 33% of all HTTPS servers are vulnerable to the attack.

https://drownattack.com/

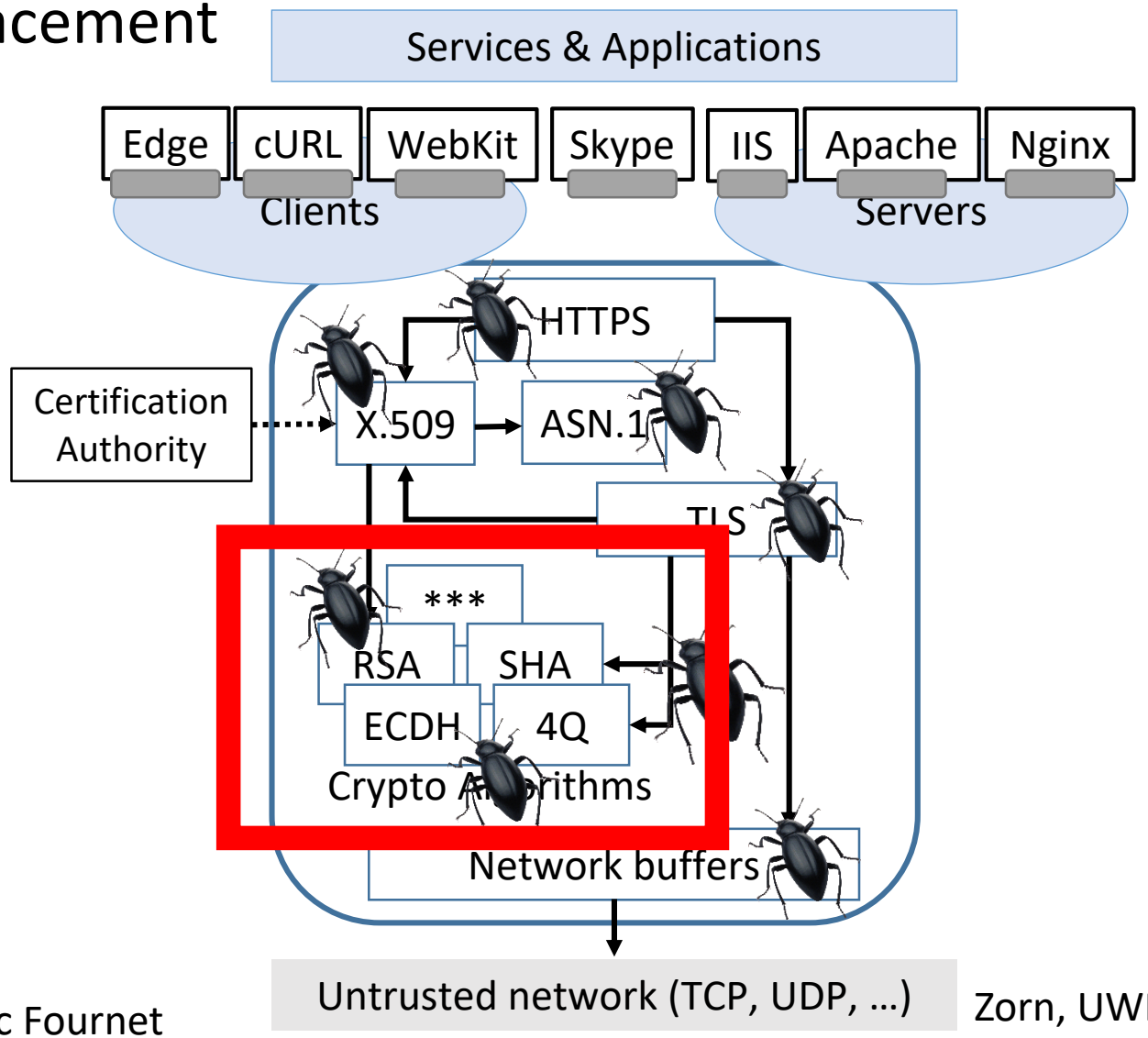
2016

Everest Expedition: Microsoft Research

Goal: drop-in verified HTTPS replacement

Challenges:

- scalability of verification
- performance
- usable tool chain



Slide courtesy of Cedric Fournet

Zorn, UWMSR 2017

Subgoal: Verified low-level crypto

Slides courtesy of Nikhil Swamy

Why verify crypto implementations?

Bugs are real, and potentially devastating!
3 bugs in OpenSSL's Poly1305 this year!

"These produce wrong results. The first example does so only on 32 bit, the other three also on 64 bit."

"I believe this affects both the SSE2 and AVX2 code. It does seem to be dependent on this input pattern."

"I'm probably going to write something to generate random inputs and stress all your other poly1305 code paths against a reference implementation."

Poly1305 functions of openssl.

These produce wrong results. The first example d
the other three also on 64 bit.

Hi folks,

You know the drill. See the attached poly1305_test2.c.

```
$ OPENSSL_ia32cap=0 ./poly1305_test2
PASS
```

Microsoft

Zorn, UWMSR 2017

Efficient crypto requires customizations

- Poly1305: Uses the prime field with $p = 2^{130} - 5$
  - Need 130 bits to represent a number
  - Efficient implementations require custom bignum libraries to delay carries
  - On X86: use 5 32-bit words, but using only **26 bits in each word**
  - On X64: use 3 64-bit words, but using only **44 bits in each word**
- Curve25519: Uses the prime field with $p = 2^{255} - 19$
  - On X64: use 5 64-bit words, but using only **51 bits per word**
- OpenSSL has 12 unverified bignum libraries optimized for each case

Everest goal: generic, efficient bignum libraries

Performance of Everest's High Assurance Crypto Library

Algorithm	Spec (F* loc)	Code+Proofs (Low* loc)	C Code (C loc)	Verification (s)
Salsa20	70	651	372	280
Chacha20	70	691	243	336
Chacha20-Vec	100	1656	355	614
SHA-256	96	622	313	798
SHA-512	120	737	357	1565
HMAC	38	215	28	512
Bignum-lib	-	1508	-	264
Poly1305	45	3208	451	915
X25519-lib	-	3849	-	768
Curve25519	73	1901	798	246
Ed25519	148	7219	2479	2118
AEAD	41	309	100	606
SecretBox	-	171	132	62
Box	-	188	270	43
Total	801	22,926	7,225	9127

Table 1: HACL* code size and verification times

- Several complete TLS ciphersuites
- *Verification can scale up!*

Algorithm	HACL*	OpenSSL
SHA-256	13.43	16.11
SHA-512	8.09	10.34
Salsa20	6.26	-
ChaCha20	6.37 (ref) 2.87 (vec)	7.84
Poly1305	2.19	2.16
Curve25519	154,580	358,764
Ed25519 sign	63.80	-
Ed25519 verify	57.42	-
AEAD	8.56 (ref) 5.05 (vec)	8.55

cycles/ECDH

- With performance as good as or better than hand-written C

Everest Impact on the TLS 1.3 Standard

Everest verification efforts led to many of their proposals being included in the standard:

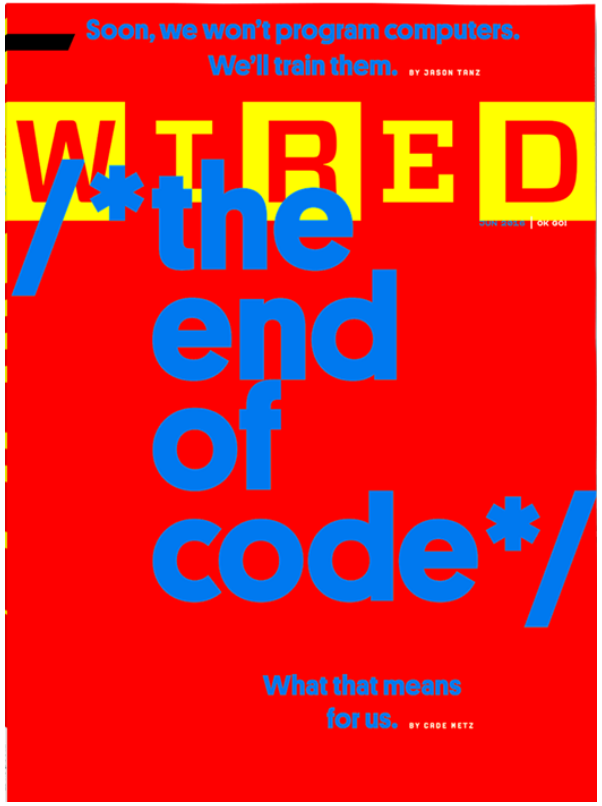
#4	log-based key separation extended session hashes (fixing attacks we found on 1.2)
#11	stream terminators (eventually fixing an attack)
#14	downgrade resilience
#15	session ticket format
#17	simplified key schedule pre-shared-key 0RTT
#18	PSK binding (fixing an attack)

Takeaway 4: Verification at scale is real and already having impact

Challenges Beyond Verifying Code

- Is code obsolete?
  - How do we test ML/AI models?
  - What's the equivalent of code coverage?
  - What math is needed to "verify" an ML model?

- Program analysis focuses on code.
  What tools do we have to debug data?
  - Data wrangling is a hot topic, about getting data fast
  - What debugging tools exist for finding bad data?



Zorn, UWMSR 2017

Addressing Human Understanding

- We need to retrain everyone to think about smart things differently
- Developer and engineers
  - Design for adversaries – every object is smart and can be hacked
  - Design for failsafe operation – like a limit switch on centrifuge
- End users
  - What is the minimum level of knowledge to be IoT-savvy?
  - Can we really assume people know nothing? What are best practices in smart object hygiene?

IoT is Happening Now

- Smart objects will replace dumb objects
- The software embedded in this objects will be written in the next 5 years, but will have implications for the next 50 years
- Our lives will depend on these objects…
- We need languages, tools, and processes to make these objects safe and secure
- **Verification** at scale is **possible**, **necessary**, and is **already having impact**

Thank you!

Research in Software Engineering (RiSE) at Microsoft Research

https://www.microsoft.com/en-us/research/group/research-in-software-engineering-rise/

CCC Computing in the Physical World Task Force

http://cra.org/ccc/task-forces/computing-in-the-physical-world/

Follow me: @benzorn https://twitter.com/benzorn