Preventing catastrophic cryptocurrency attacks

Neha Narula
MIT Digital Currency Initiative
Financial Cryptography 2019

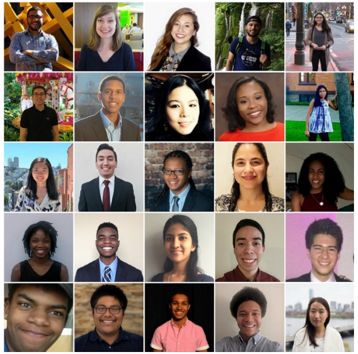
mit
media
lab

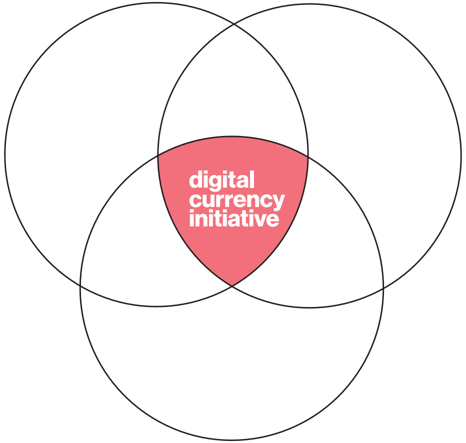
digital
currency
initiative

MIT Digital Currency Initiative

Educators

Build industry capacity by teaching courses and advising students





Researchers

Contribute research and core open-source development addressing scalability, privacy, and security

Conveners

MIT has a history of standards setting, and providing a common platform

We're neutral—no ICOs, most don't hold material amounts of cryptocurrency

digital currency initiative

Cryptocurrency is not ready for billions of users

- Many challenges remain in scalability, interoperability, usability, and privacy
- There is increasing security risk with new, unproven protocols and latent implementation bugs

Current state of cryptocurrency security

- Thousands of cryptocurrencies and codebases
- Varied levels of security experience
- Attackers can easily and anonymously exploit vulnerabilities for financial gain

This talk

- Experience with a disclosure
- Lessons learned
- Open questions

Three vulnerabilities

IOTA	signature forgeries	steal money	$1.2B
BitcoinCash	chain split	double spend	$24B
bitcoin	DoS inflation	halt network create new money	$116B

digital currency initiative

Important note

- All of these bugs were **disclosed** to developers
- As far as we know they were **not exploited**
- The developers all **deployed mitigations** for them
- These vulnerabilities **no longer impact** the security of any of the cryptocurrencies mentioned here

This talk

- Experience with a disclosure
  - A signature forgery attack on IOTA’s multisig
  - Breaking the Curl-P-27 hash function
  - Disclosure
- Lessons learned
- Open questions

IOTA

Volkswagen

BOSCH

Invented for life

800M dollar marketcap

Custom hash function called **Curl**

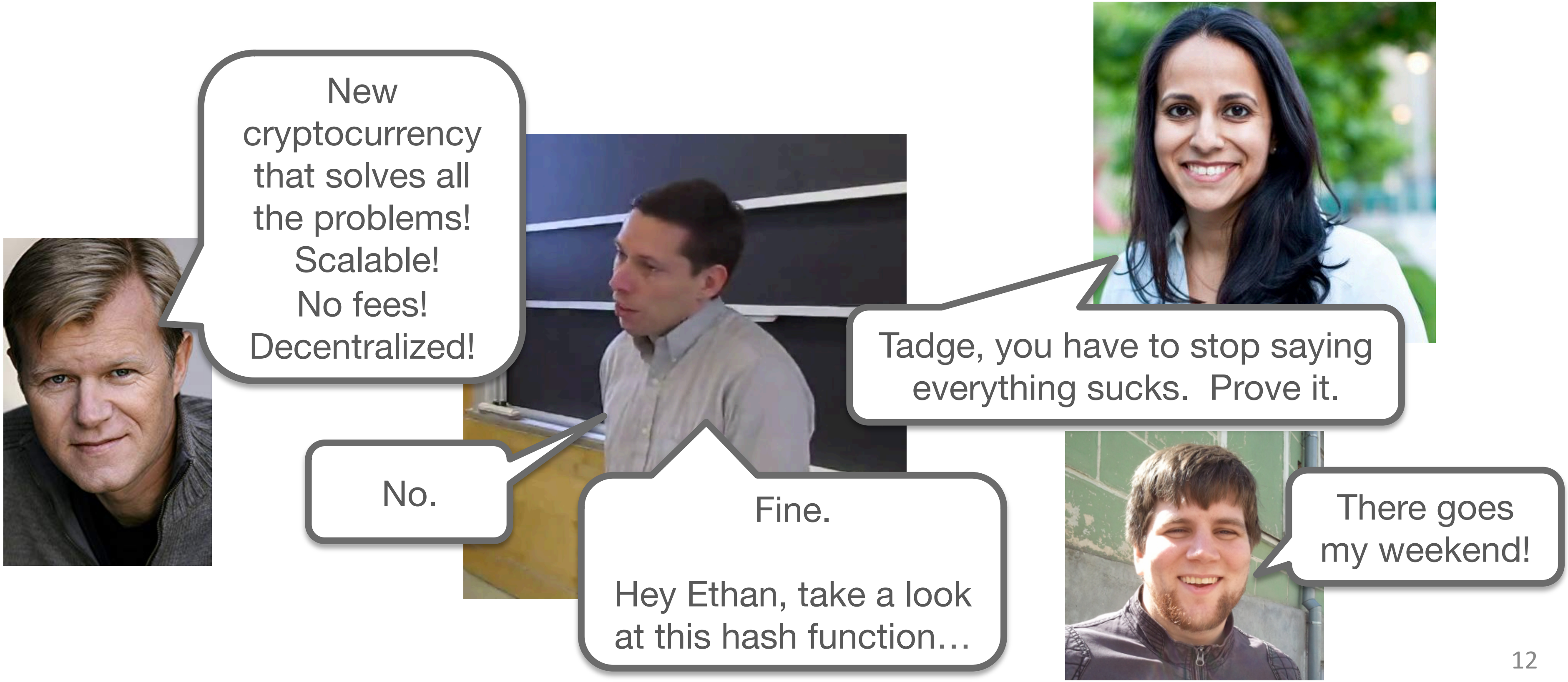
IOTA Background: Terminology

	Bitcoin	**IOTA**
Payment	Transaction	Bundle
Currency	1 Bitcoin ~ $3.9K	1M IOTA ~ $0.30

IOTA Background: Terminology

	Bitcoin	**IOTA**
Payment	Transaction	Bundle
Currency	1 Bitcoin ~ $3.9K	1M IOTA ~ $0.30
Representation	Bits (0, 1) bytes (8 bits)	Trits (-1, 0, 1) trytes (3 trits)

Why did we look at IOTA?



What is our attack?

- Bob and Eve have funds under joint control and wish to spend them
- Bob signs a payment where he gets $2M and Eve gets almost nothing
- Eve forges Bob's signature and instead sends a payment where she gets $2M and Bob gets almost nothing
- Chosen message setting: Eve gets to create the payment Bob signs

Ethan Heilman (Boston University, Arwen, advisor at DAGLabs), Neha Narula (MIT Media Lab)Tadge Dryja (MIT Media Lab), Madars Virza (MIT Media Lab, Zcash), Garrett Tanzer (Harvard University), James Lovejoy (MIT Media Lab, Vertcoin), Michael Colavita (Harvard University)

This talk

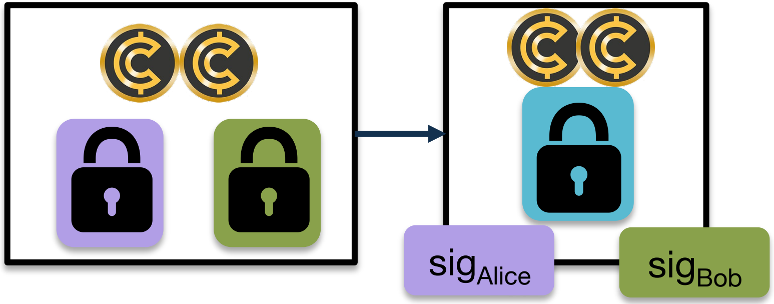
- Experience with a disclosure
  - **A signature forgery attack on IOTA's multisig**
  - Breaking the Curl-P-27 hash function
  - Disclosure
- Lessons learned
- Open questions

What is multisig?

“Two-person” rule for nuclear launch

Using 2-of-2 multisig for payments

A valid payment requires **k**-of-**n** signatures. Example 2-of-2:



Spending from a multisig address

Why multisig? Added security.

- Attacker has to compromise **both** keys
- Can store keys in isolated locations (cold storage)
- Used by many exchanges

IOTA Background: Signatures

IOTA's signature scheme:

- IOTA builds on Winternitz One-Time Signatures (WOTS)
- IOTA modifies WOTS

  ...to hash messages with Curl-P-27 prior to WOTS

```
IOTA_Sign(sk, m):
    h_m = Curl-P-27(m)
    sig = WOTS_Sign(sk, h_m)
    return sig
```

IOTA Background: Signatures

IOTA's si

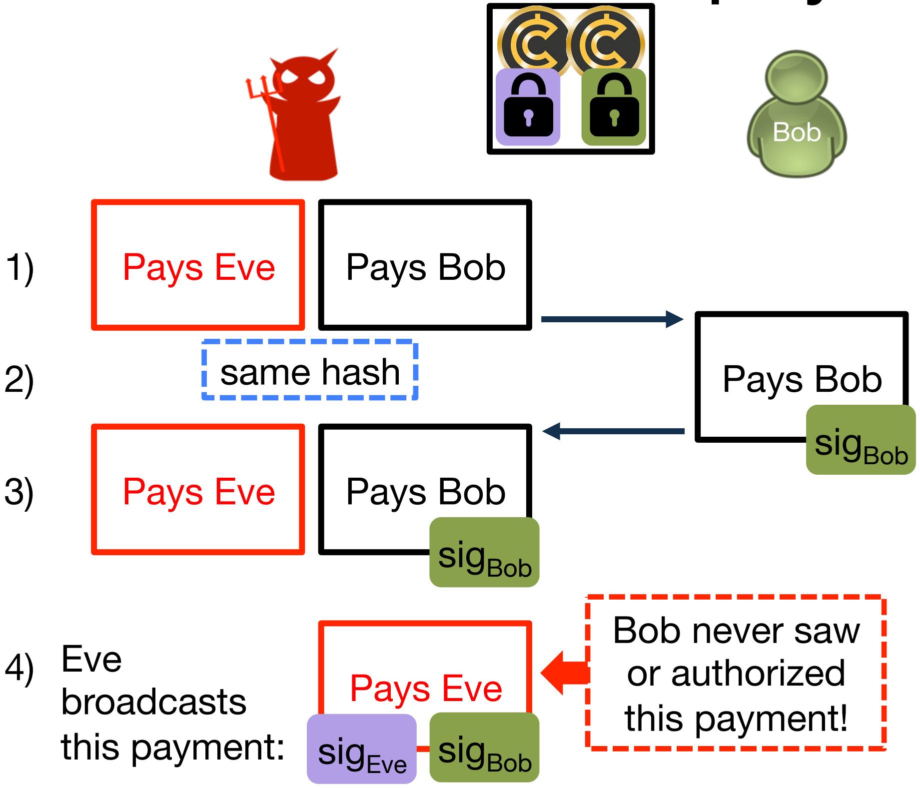
- IOTA b WOTS)
- IOTA r

...to h TS

The signature scheme details don't matter because in IOTA, payments are **hashed** before they are signed

If you can break the hash function, you can forge signatures!

Exploiting colliding bundles: Unauthorized payments



1. Eve creates **two** special bundles which have the **same** hash
2. Eve asks Bob to sign the bundle paying him
3. Eve **copies** Bob's signature from the benign bundle to the evil bundle
4. Eve signs and broadcasts the evil bundle

Placing collisions to pay different amounts

Payee	Value
Alice:	100
Eve:	**1**
Carol:	100
Bob:	**2541865828330**

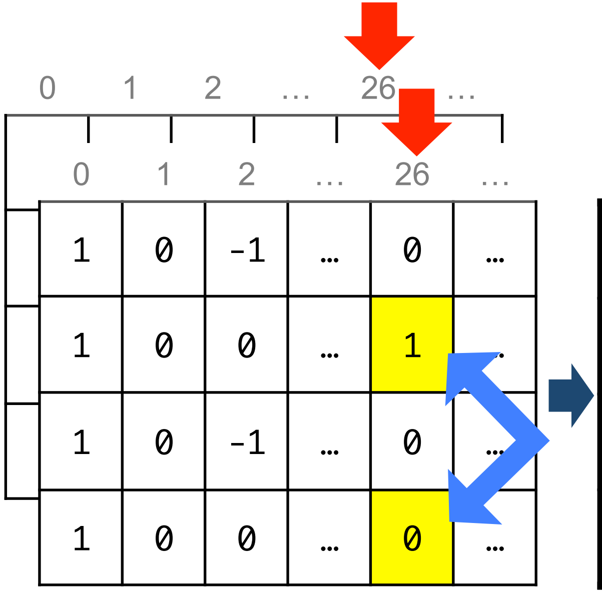
0	1	2	…	26	…
1	0	-1	…	0	…
1	0	0	…	0	…
1	0	-1	…	0	…
1	0	0	…	1	…

- Target Value fields for differing trits
- Create two colliding bundles which differ in 26th trit of two message blocks

Placing collisions to pay different amounts

Payee	Value
Alice:	100
Eve:	**1**
Carol:	100
Bob:	**2541865828330**

Bundle Bob sees



Payee	Value
Alice:	100
Eve:	**2541865828330**
Carol:	100
Bob:	**1**

Bundle Eve broadcasts

- Target Value fields for differing trits
- Create two colliding bundles which differ in 26th trit of two message blocks
- **Limitations: Can only play this trick in specific places**

This talk

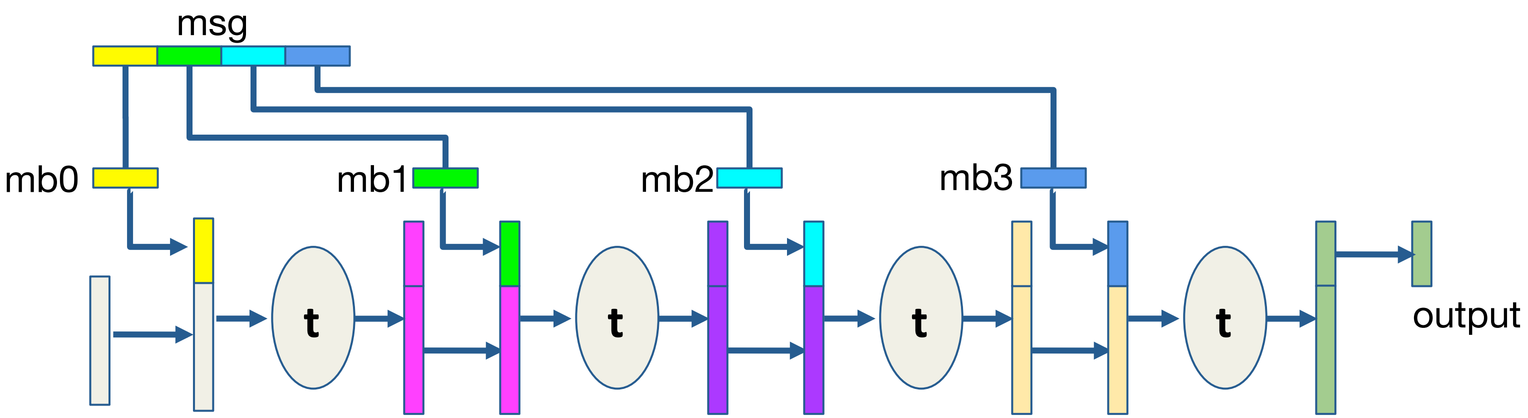
- Experience with a disclosure
  - A signature forgery attack on IOTA’s multisig
  - **Breaking the Curl-P-27 hash function**
  - Disclosure process
- Lessons learned
- Open questions

Curl-P-27: A Cryptographic Hash Function

To forge signatures we need to find
colliding msgs for Curl-P-27:

Curl-P-27(`-1,0,1,1...,-1`) == Curl-P-27(`0,1,0,0,0...,0`)

Curl-P-27 uses a sponge-like construction



Security depends on the transform function t

The transformation function in Curl-P-27 is just the repeated application of a permutation + a simple S-Box

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

AES S-Box

	-1	0	1
-1	1	1	-1
0	0	-1	1
1	-1	0	0

Curl-P-27 S-Box

Curl-P-27: Reducing collision resistance

Choose a random message

`-1011110101...-1`

Flip a trit

`-1011010101...-1`

If we flip the 26th trit the probability of a collision is:

$>1/(2^{42.40})$

If we are clever about choosing the message this increases to $>1/2^{22.87}$ = **1 out of 7.6 million**

In cryptographic terms this is **23-bit collision resistance**

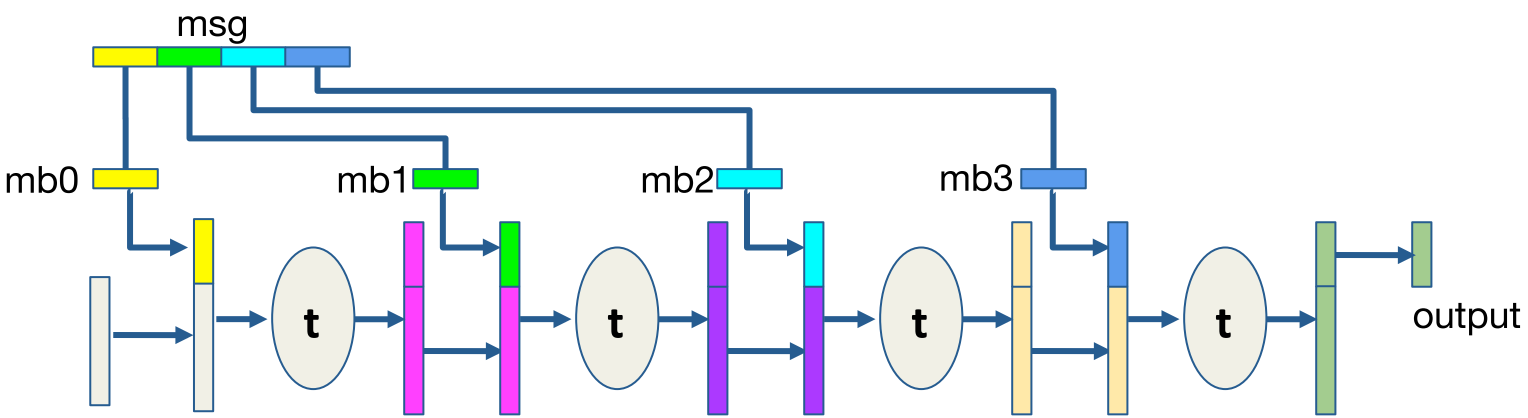
IOTA bundles: unconstrained tag field

As the likelihood of a collision is at least 1 out of 7.6 million we need to try many messages (bundles) before we are successful

address	tag		value
DKSDJFLS...R	99999...999 DJKLQ…JKAJF		22000000...
QWEWEABZ...9	99999...999 QIERP…LKQCB		00000010...
ABEPCMQQ...Z	99999...999 PLKEU…VBNTY		00050000...

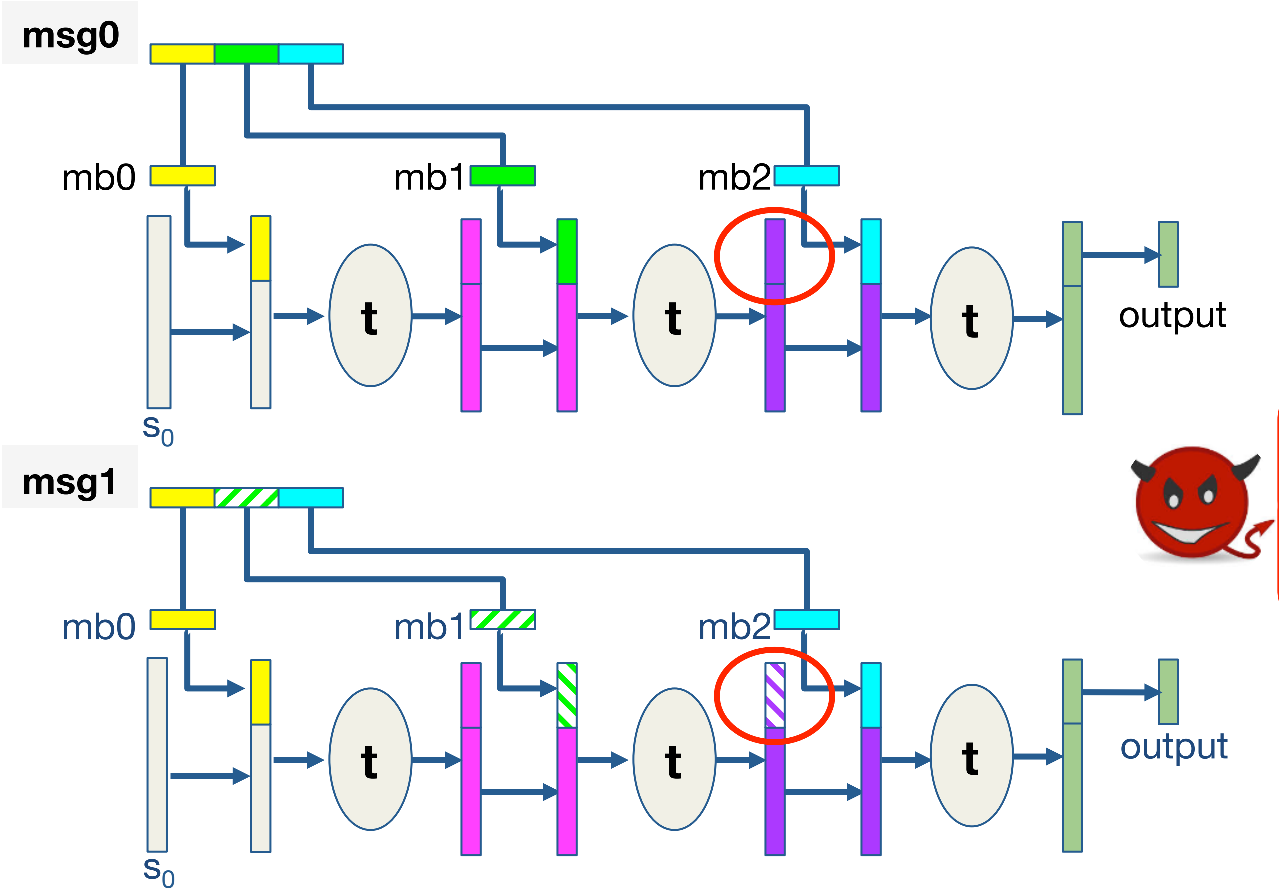
We can change the 81-trit tag field in IOTA bundles
Tags have no impact on transaction validity

Curl-P-27 modifies sponge to overwrite



Differences are erased as new message blocks overwrite the first third of the state

How do we create collisions in Curl-P-27?



Plan: ensure all the diffs are in first 3rd of the state

```
neha@ben:$
```

github.com/mit-dci/tangled-curl

```
0$ bash  1$ bash                                     Aug 2018 17:40:37
```

This talk

- Experience with a disclosure
  - A signature forgery attack on IOTA’s multisig
  - Breaking the Curl-P-27 hash function
  - **Disclosure**
- Lessons learned
- Open questions

IOTA fixes our signature forgery vulnerability

In July 2017 we disclosed this to the IOTA devs
...in response the IOTA devs replaced Curl-P-27 with Kerl

Kerl is used in IOTA for the following tasks:

Functionality	Curl-P-27	Curl-P-81	Kerl
Address generation			V^
Signature generation			V
Signature verification			V
Essence calculation (bundleHash)			V
Proof of Work		V	
Transaction Hash		V	
Milestone verification	V		

Curl-P-N: N number of rounds

https://github.com/iotaledger/kerl

IOTA claims this was a backdoor

"[..] *Curl-P was indeed deployed in the open-source IOTA protocol code as a copy-protection mechanism to prevent bad actors cloning the protocol and using it for nefarious purposes. Once the practical collisions were uncovered, its purpose as a copy-protection mechanism was of course rendered obsolete*"

In response to Ethan's question *"Did we discover a copy-protection backdoor in IOTA?"*
they write: *"The answer to the first question is of course, yes, as we have explained above."*

Read IOTA's full statement at blog.iota.org/11fdccc9eb6d

Troika:
a ternary hash function

Reference document

Version 1.0.1

December 21, 2018

A new hash function appears

- In December 2018 IOTA announced the creation of a new ternary hash function Troika designed by Cybercrypt A/S
- €200,000 prize pool to break round-reduced variants

"Currently IOTA uses the relatively hardware intensive NIST standard SHA-3/Keccak for crucial operations for maximal security."

"[...] we [...] started tackling the hardware side with new thinking in computational processing. A next generation of microprocessor architecture based on ternary logic for ultimate efficiency in IoT is the result. (A deep dive blog post on trinary's superiority over binary will come soon)."

Read IOTA's full statements at blog.iota.org/678e741315e8 and blog.iota.org/615d2df79001

This talk

- Experience with a disclosure
  - A signature forgery attack on IOTA’s multisig
  - Breaking the Curl-P-27 hash function
  - Disclosure process
- **Lessons learned**
- Open questions

Lessons learned
(for disclosers)

- Expect wildly different types of responses
- Be prepared to obtain legal representation
- Consider disclosing anonymously

BOSTON UNIVERSITY **School of Law** Technology Law Clinic

Lessons learned (for cryptocurrencies)

- Have a responsible disclosure policy
  - Contact address, GPG keys
- Support anonymous communication

Other reasons to disclose anonymously

- Potential to exploit vulnerability and make a lot of money
- Also potential to cause others to *lose* a lot of money
- If a vulnerability is exploited, you become a suspect and target

Cryptocurrencies should consider commensurate bounties!

Bitcoin ABC

`SIGHASH_BUG = 0x20`

Responsible disclosure in the era of cryptocurrencies

My experience disclosing a critical Bitcoin Cash vulnerability

Cory Fields
Aug 9, 2018 · 8 min read

On April 25, 2018, I anonymously and privately disclosed a critical vulnerability in Bitcoin Cash, one of the world's most valuable cryptocurrencies—not to be confused with Bitcoin. A successful exploit of this vulnerability could have been so disruptive that transacting Bitcoin Cash safely would no longer be possible, completely undermining the utility (and thus the value) of the currency itself. Instead, the vulnerability was fixed without incident, and publicly disclosed on May 7, 2018.

- There was no disclosure policy
- It was hard to find contact information for developers
- It was hard to contact them anonymously
- It was hard to confirm receipt

all since fixed!

medium.com/mit-media-lab-digital-currency-initiative/48a99b85aad4

Lessons learned
(for cryptocurrencies)

- Have a responsible disclosure policy
  - Contact address, GPG keys
- Support anonymous communication
- Forge relationships with researchers and related implementations

Roger Ver
Aug 9, 2018

Thank you for the responsible disclosure Cory. It is appreciated by myself and others in the community.

158

Paulo Falcao
Aug 9, 2018

It’s people like you that makes me believe in crypto. Well done!

126

Alex Martell
Aug 9, 2018

Amazing story, amazing and laudable ethos. Every BCH holder owes you big time.

And probably every BTC holder too :)

Thank you

42

Next vulnerability in bitcoin-core was disclosed by a Bitcoin Cash developer (u/awemany)

This talk

- Experience with a disclosure
  - An signature forgery attack on IOTA’s multisig
  - Breaking the Curl-P-27 hash function
  - Disclosure process
- Lessons learned
- **Open questions**

Open questions
(for everyone)

- How do we coordinate disclosures across multiple cryptocurrencies?
- How should developers communicate the vulnerability and its mitigation across the cryptocurrency’s ecosystem?

Bitcoin Core

CVE-2018-17144 Full Disclosure

Full disclosure

CVE-2018-17144, a fix for which was released on September 18th in Bitcoin Core versions 0.16.3 and 0.17.0rc4, includes both a Denial of Service component and a critical inflation vulnerability. It was originally reported to several developers working on Bitcoin Core, as well as projects supporting other cryptocurrencies, including ABC and Unlimited on September 17th as a Denial of Service bug only, however we quickly determined that the issue was also an inflation vulnerability with the same root cause and fix.

bitcoincore.org/en/2018/09/20/notice

1. Hide a fix for the inflation bug inside a fix for the DoS bug
2. Tell everyone about the DoS bug and fix to get them to upgrade as fast as possible

This effectively dropped a 0-day on many coins derived from bitcoin-core

Open questions (for everyone)

- How do we coordinate disclosures across multiple cryptocurrencies?
- How should developers communicate the vulnerability and its mitigation across the cryptocurrency's ecosystem?
- Who should one even disclose to?
- Should the discloser or developers move vulnerable funds?
- How can we prevent vulnerabilities in the first place?

Maybe security doesn’t matter?

Price seems to be totally uncorrelated with vulnerabilities and attacks!

- Fixing exploits inspires confidence in developer teams
- The cryptocurrency market is currently small and irrational (it might not stay that way)
- Network attacks so far have been relatively small and those attacked are able to absorb the losses (it might not stay that way)

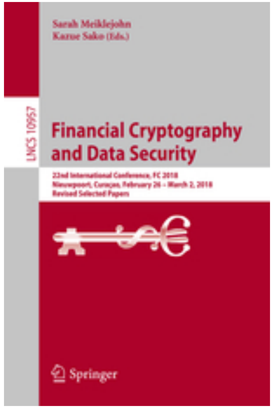
Cryptocurrency security is a public good

- A really bad attack could affect many coins and businesses
- Many bad attacks could reduce trust in cryptocurrencies and set us back years

Cryptocurrency security working group

1. Identify and circulate best practices
2. Write tests, run monitoring and security tools
3. Research to move to safer programming languages and on formal verification

Where to keep up with research?

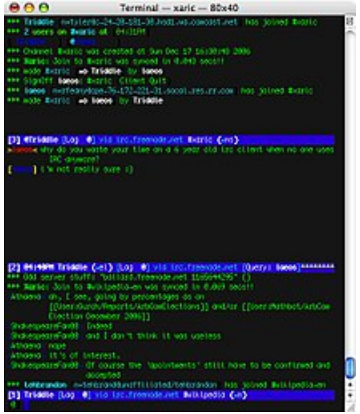


arXiv.org

research

INTERNATIONAL ASSOCIATION FOR CRYPTOLOGIC RESEARCH

iacr



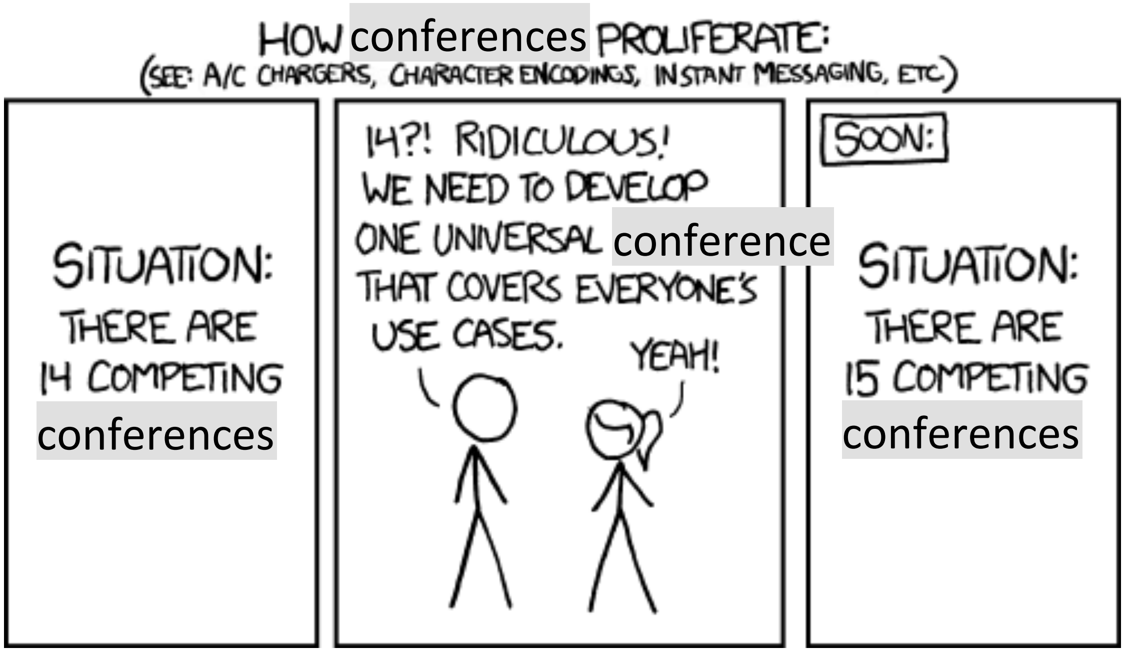
Telegram

Introducing

Cryptocurrency Research Review

discourse.mitcryptoresearch.org/

digital currency initiative



- Interdisciplinary (CS+economics+law)
- Experiment: speed, overlay, reviews, and submissions
- One place to look for high-quality, reviewed research

digital currency initiative

mit media lab

dci.mit.edu

@neha narula@mit.edu

digital currency initiative

mit media lab

@neha

dci.mit.edu

narula@mit.edu