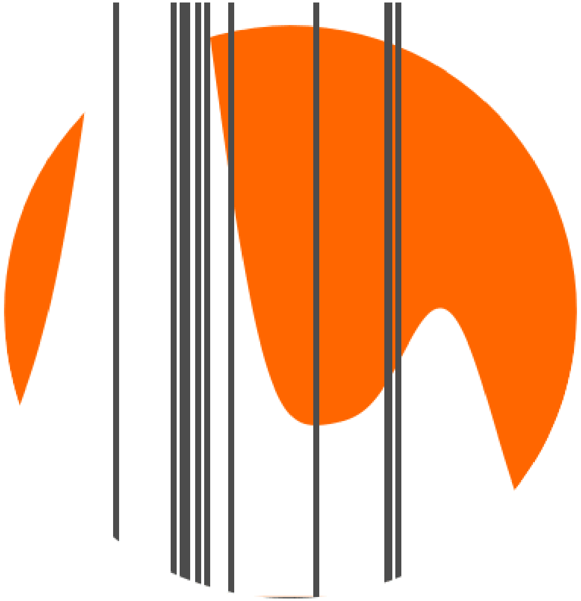
A Statistical Research Agenda for Monero



Secure. Private. Untraceable.
Resistant to statistical attack.
N = 11

Rucknium

Monero Research Lab Researcher
and
Empirical Microeconomist

Slides are available at:
https://github.com/Rucknium/presentations

Some of my contributions to Monero

- Discovered a way to speed up transaction confirmations by 60 seconds.[1]
- Computed the privacy impact of P2Pool decentralized mining payouts. sech1 and duggavo doubled payout efficiency.[2]
- Analyzed the privacy effects of Mordinals (Monero NFTs).[3]
- Contributed to “Fingerprinting a Flood” Analysis of July-August 2021 transaction volume anomaly.[4]
- Ongoing development of Optimal Static Parametric Estimation of Arbitrary Distributions (OSPEAD) for an improved decoy selection algorithm.[5]
- Serve on the MAGIC Monero Fund committee.
- Maintenance of Monero research community resources.[6]

[1] https://www.reddit.com/r/Monero/comments/11nu4aj/monero_transaction_confirmations_are_now_60
[2] https://www.reddit.com/r/MoneroMining/comments/1095730/psa_p2pool_network_upgrade_aka_hardfork_on_march/
and https://github.com/monero-project/research-lab/issues/109
[3] https://www.reddit.com/r/Monero/comments/12kv5m0/empirical_privacy_impact_of_mordinals_monero_nfts/
[4] https://www.reddit.com/r/Monero/comments/pvm634/fingerprinting_a_flood_forensic_statistical/
[5] https://github.com/monero-project/research-lab/issues/93
[6] https://moneroresearch.info , Open Research Questions: https://github.com/monero-project/research-lab/issues/94

Quiz

Equations written by Satoshi Nakamoto in the original bitcoin white paper involved:

A) A concept from cryptography

B) A concept from probability

C) Both

D) Neither

Quiz

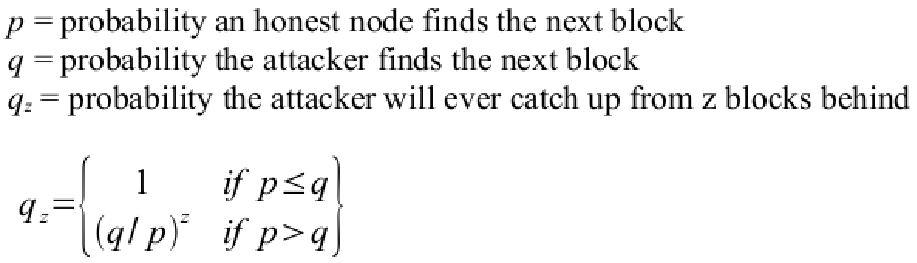
Equations written by Satoshi Nakamoto in the original bitcoin white paper involved:

A) A concept from cryptography

[✓ B) A concept from probability]

C) Both

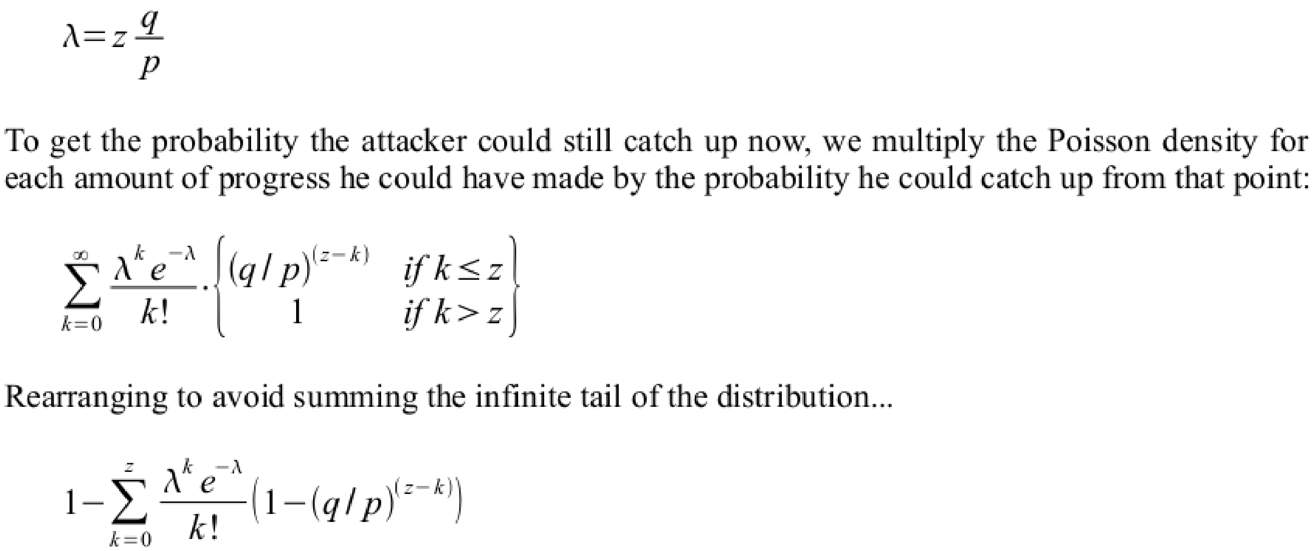
D) Neither



p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{Bmatrix} 1 & \textit{if } p \le q \\ (q/p)^z & \textit{if } p > q \end{Bmatrix}$$

[...]



$$\lambda = z\frac{q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{Bmatrix} (q/p)^{(z-k)} & \textit{if } k \le z \\ 1 & \textit{if } k > z \end{Bmatrix}$$

Rearranging to avoid summing the infinite tail of the distribution...

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

Satoshi used probability theory to calculate the chances of an attacker re-writing the bitcoin blockchain for a double-spend attack.

*Slight correction to this calculation by Grunspan and Pérez-Marco (2018):

https://bitcoinmagazine.com/technical/how-satoshi-messed-his-math-and-how-these-academics-just-fixed-it

Two categories of Monero statistical research questions:

1) Involves ring signatures

2) Does not involve ring signatures

Opinion: Ring signatures will be with Monero for many more years

Global/full blockchain membership proofs

- With ring signatures: The truly spent transaction output is one of a small subset of the total set of outputs. (Only 16 now. 128+ with Seraphis.)
- Global membership proofs: Whenever an output is spent, it could be *any* of the millions of transaction outputs on the blockchain.
- Zcash has had this model in their shielded transaction pools since 2016.

Problems with Zcash's zk-SNARK model

1) High CPU and RAM requirements for transaction construction

2) Trusted setup

3) Cryptography based on new foundations: not yet battle-tested

Note: Bulletproofs, which Monero uses, is a zero-knowledge proof. It is *not* a zk-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) since it is not "succinct" (not "small" size or "fast" proving/verification time).

Problems with Zcash's zk-SNARK model

1) High CPU and RAM requirements for transaction construction

Reduced with 2018 Sapling upgrade

2) Trusted setup

Orchard/Halo 2 upgrade in 2022 provided zk-SNARKS with no trusted setup

3) Cryptography based on new foundations: not yet battle-tested

Not solved

Many “Close Calls” Have Threatened Private Digital Cash Protocols

Often problems with the cryptography mathematics itself, not the computer code that implements it. Garbage in, garbage out.

Case Study 1:
Zcash Counterfeiting Flaw

“Zcash Counterfeiting Vulnerability Successfully Remediated” excerpt
Josh Swihart, Benjamin Winston and Sean Bowe | February 5, 2019

> “On March 1, 2018, Ariel Gabizon, a cryptographer employed by the Zcash Company at the time, discovered a subtle cryptographic flaw in the [BCTV14] paper that describes the zk-SNARK construction used in the original launch of Zcash. The flaw allows an attacker to create **counterfeit shielded value** in any system that depends on parameters which are generated as described by the paper.”

(emphasis added)

https://electriccoin.co/blog/zcash-counterfeiting-vulnerability-successfully-remediated

“Zcash Counterfeiting Vulnerability Successfully Remediated” (cont.)

> **“This vulnerability is so subtle that it evaded years of analysis by expert cryptographers focused on zero-knowledge proving systems and zk-SNARKs.** In an analysis [Parno15] in 2015, Bryan Parno from Microsoft Research discovered a different mistake in the paper. However, the vulnerability we discovered appears to have evaded his analysis. The vulnerability also appears in the subversion zero-knowledge SNARK scheme of [Fuchsbauer17], where an adaptation of [BCTV14] inherits the flaw. The vulnerability also appears in the ADSNARK construction described in [BBFR14]. Finally, the vulnerability evaded the Zcash Company’s own cryptography team, which includes experts in the field that had identified several flaws in other parts of the system.”

(emphasis added)

“Zcash Counterfeiting Vulnerability Successfully Remediated” (cont.)

> “Importantly, the [BCTV14] construction did not have a dedicated security proof, as noted in [Parno15], and relied mainly on the [PGHR13] security proof and the similarity between the two schemes. The Zcash Company team did attempt to write a security proof in [BGG17], but it did not uncover this vulnerability.”

[PGHR13] Parno, Howell, Gentry, and Raykova (2013) “Pinocchio: Nearly Practical Verifiable Computation.”

[BCTV14] Ben-Sasson, Chiesa, Tromer, and Virza (2014) “Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture.”

[BBFR14] Backes, Barbosa, Fiore, and Reischuk (2014) "ADSNARK: Nearly Practical and Privacy-Preserving Proofs on Authenticated Data."

[Parno15] Parno (2015) "A Note on the Unsoundness of vnTinyRAM's SNARK."

[Fuchsbauer17] Fuchsbauer (2017) "Subversion-zero-knowledge SNARKs"

[BGG17] Bowe, Gabizon, and Green (2017) “A multi-party protocol for constructing the public parameters of the Pinocchio zk-SNARK.”

Case Study 2:
Cryptonote (Monero) Counterfeiting Flaw

"Disclosure of a Major Bug in CryptoNote Based Currencies" excerpt luigi1111 and Riccardo "fluffypony" Spagni | May 17, 2017

> “In Monero we've discovered and patched a critical bug that affects all CryptoNote-based cryptocurrencies, and **allows for the creation of an unlimited number of coins** in a way that is undetectable to an observer unless they know about the fatal flaw and can search for it.”
>
> “We patched it quite some time ago, and confirmed that the Monero blockchain had NEVER been exploited using this, but until the hard fork that we had a few weeks ago we were unsure as to whether or not the entire network had updated.”

(emphasis added)

https://web.getmonero.org/2017/05/17/disclosure-of-a-major-bug-in-cryptonote-based-currencies.html

Case Study 3:
Secret Network (SCRT) privacy exposure

Secret Network (SCRT) privacy exposure

- Secret Network is a smart contract blockchain that promises privacy for users.
- It uses Intel's Software Guard Extension (SGX), a type of hardware Trusted Execution Environment (TEE), to protect user privacy.

van Schaik et al. (2022)
“SoK: SGX.Fail: How Stuff Gets eXposed”

“The Secret Network has been vulnerable to the xAPIC and MMIO vulnerabilities that were publicly disclosed on August 9, 2022. These vulnerabilities could be used to extract the *consensus seed*, a master decryption key for the private transactions on the Secret Network. **Exposure of the consensus seed would enable the complete retroactive disclosure of all Secret-4 private transactions since the chain began.** We have helped Secret Network to deploy mitigations, especially the Registration Freeze on October 5, 2022.”

“However, there is no way to know for certain whether this attack has been attempted previously. It is also possible that ordinary node operators may have unintentionally prepared the attack, if they were active nodes prior to the mitigations, and may opportunistically decide to complete it in the future. We urge privacy-conscious users to re-evaluate their risk considering that their past transactions may be exposed.”

(emphasis added)

More info: https://sgx.fail/

Actually exploited flaws in private digital cash protocols

- 2017: Bytecoin, a Cryptonote-based coin, was exploited by the counterfeiting flaw that Monero discovered and patched
- 2017: A malicious party exploited a counterfeiting bug in the code of Firo (named Zerocoin then)
- 2021: Counterfeiting exploit against Haven due to several flaws

How confident do we need to be?

- In the last few years, cryptocurrency protocols have been exploited for over 5 billion USD.
  - Flawed code, flawed safeguards, flawed cryptography, flawed economics.
  - Source: https://rekt.news/leaderboard/
- How confident should Monero be in new cryptographic foundations before they are irrevocably activated on mainnet?
  - 95% confident? 99% confident? 99.9% confident?
- How confident were the researchers and developers of these exploited protocols?

Peer review, audits, and battle testing

- Academic peer review: Fellow cryptographers check the mathematics of the security proofs in new papers
  - Incentive: Professional responsibility to science. Build and maintain reputation.
- Code audits: An independent party checks that the code correctly implements the paper’s math and does not contain exploitable holes.
  - Incentive: Build and maintain reputation.
- Battle testing: Every basement hacker and government agent declares open season on the protocol.
  - Incentive: Wealth. Pursuit of government objectives.

Trustless zk-SNARKs are not yet battle-tested

- Just a few years old
- Many protocols are not properly peer reviewed
- As far as I know, there is no explicit financial incentive to find flaws
  - Zcash has no “bug bounty” program
  - Monero has a 1,061 XMR bug bounty program.[1]
- Granted, the “bounty” for counterfeiting flaws is self-executing. But not for privacy flaws!

[1] https://github.com/monero-project/meta/blob/master/VULNERABILITY_RESPONSE_PROCESS.md

Verify, don’t trust?

- Cryptocurrency users are encouraged to verify, not trust, the blockchain data and its cryptography.
  - It’s an ideal that is rarely achieved in practice.
- The Monero Research Lab has few researchers now who can write and/or evaluate mathematics security proofs of new cryptography.
  - Mathematics at this level is extremely difficult.
  - Understanding how something works is not the same level of knowledge to be able to discover flaws.
  - How many engineers and scientists use calculus? How many can write the calculus mathematics proofs and recognize flawed proofs?

“The Monero Core Team and the Monero Research Lab would like to follow the development philosophy that it is wise to start with smaller changes at first and then ramp those changes up over time, rather than start with drastic changes and try to scale them back.”

Adam Mackenzie, Surae Noether, and Monero Core Team (2015), "Improving Obfuscation in the CryptoNote Protocol." Monero Research Lab. Research Bulletin #3.

Monero cannot afford to be "experimental"

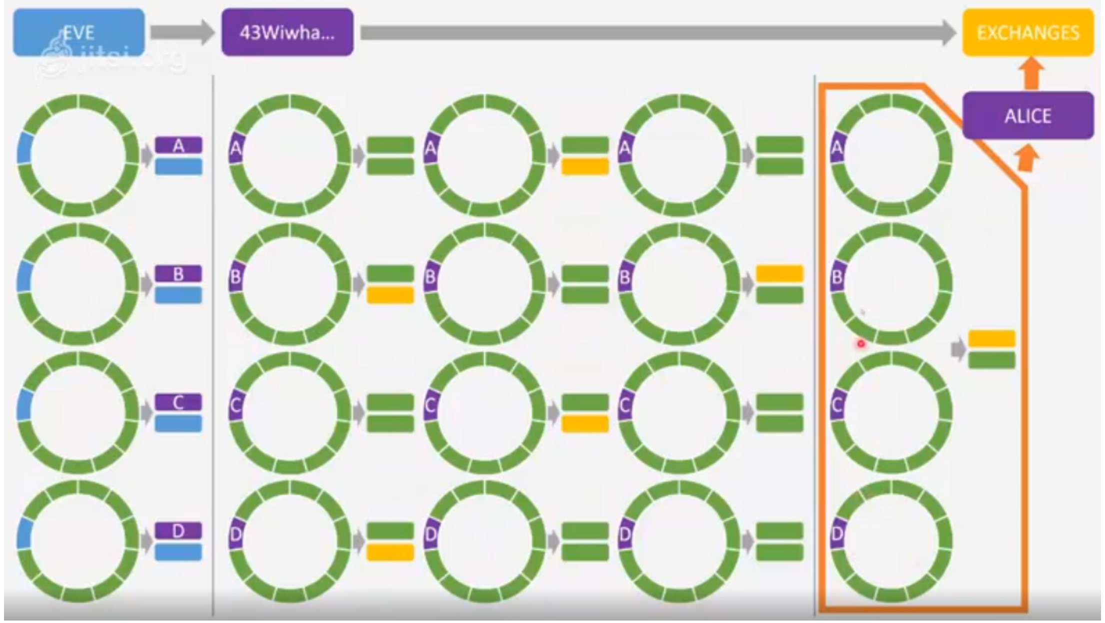
- More people may rely on Monero to protect their privacy than any other private digital cash protocol. There is a moral responsibility!
- Monero has no non-private "transparent" pool like many other protocols.
  - Coins like Zcash and Firo can quarantine possible counterfeiting exploits by their "turnstile" rules that prohibit a more coins exiting the private pool than have entered.

Statistical research on ring signatures

- In my opinion, Monero’s privacy will be in a very good position once Seraphis-level 128+ ring size and an improved mimicking decoy selection algorithm like OSPEAD are implemented.
- The main attack that will remain is the EAE attack.
- Egger, Lai, Ronge, Woo, & Yin (2022) “On Defeating Graph Analysis of Anonymous Transactions” argued that a ring size as low as 24 would protect Monero users from a specific type of de-anonymizing attack called chain reactions (assuming a partitioning decoy selection algorithm.)

Ring signature research:
Defeat poisoned outputs/EAE/overseer attacks

“We model examples where two colluding parties ‘E’ attempt to learn information about an individual ‘A.’ By sending outputs to individuals and tracing their transaction graphs, these colluding parties may perform powerful statistical tests to learn significant information, especially for repeated transactions where they would not normally occur by chance.”



https://www.moneroutreach.org/breaking-monero/poisoned-outputs.html

Ring signature research:
Defeat poisoned outputs/EAE/overseer attacks (cont.)

Research questions:

1) How to formally describe the EAE attack? What is the average false positive and false negative rate of the attack?

2) Is churning a defense against the EAE attack?

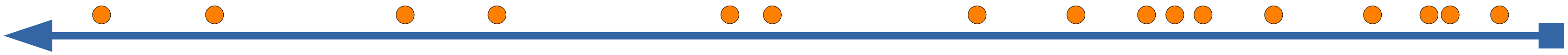
   - For best privacy, what should the wait time between churns be? Should it match the decoy selection algorithm?

   - When, if ever, should outputs be combined while churning?

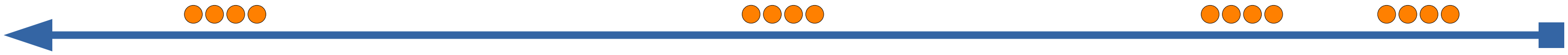
- Knacc, Surae, and Sarang made some attempts at EAE research.

Ring signature research:
Ring member binning

Randomly choose each decoy independently (current method):



Randomly choose locations for a few bins. Then select decoys within those bins:



The goal of binning is to provide a second layer of defense if the adversary can guess the age of the real spend.

The current version of the Seraphis code implements binning.

Ring signature research:
Ring member binning (cont.)

Research questions:

- **What are the costs and benefits of binning?**
- **Does binning improve privacy for some threat models but worsen privacy for others?**

Monero Research Lab issue: https://github.com/monero-project/research-lab/issues/84

Papers that analyze “one big bin” (partitioning):

Yu, Au, & Esteves-Verissimo (2019) “Re-thinking Untraceability in the Cryptonote-style Blockchain.”

Ronge, Egger, Lai, Schröder, & Yin (2021) “Foundations of Ring Sampling.”

Egger, Lai, Ronge, Woo, & Yin (2022) “On Defeating Graph Analysis of Anonymous Transactions.”

Computer scientists, but no statisticians, analyze multiple bins:

Möser et al. (2018) “An Empirical Analysis of Traceability in the Monero Blockchain.”

Ring signature research:
Improved mimicking decoy selection

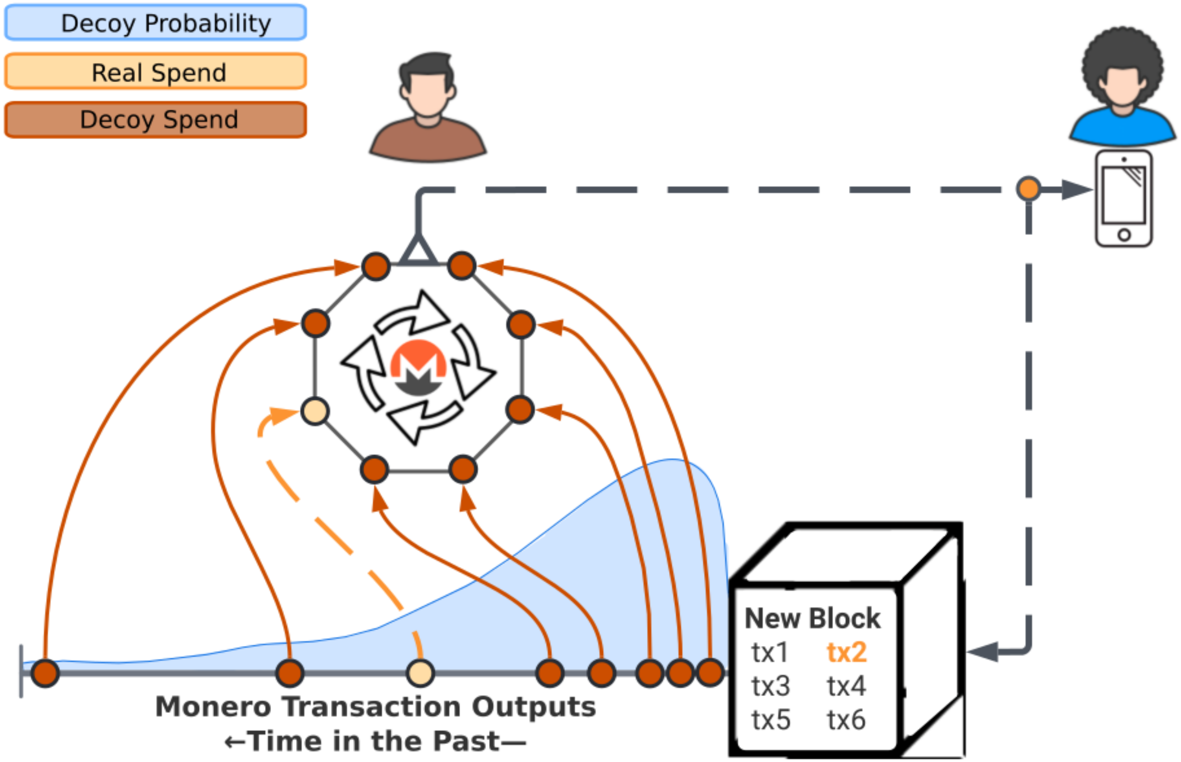


Diagram image derived from ACK-J’s image

Ring signature research:
Improved mimicking decoy selection

Many papers argue that of matching the real spend age distribution as closely as possible is very important to protect user privacy:

Mackenzie, Noether, & Monero Core Team (2015) "Improving Obfuscation in the CryptoNote Protocol."

Kumar, Fischer, Tople, & Saxena (2017). "A Traceability Analysis of Monero's Blockchain."

Möser et al. (2018) "An Empirical Analysis of Traceability in the Monero Blockchain."

Ye, Ojukwu, Hsu, & Hu (2020) "Alt-coin Traceability."

Ronge, Egger, Lai, Schröder, & Yin (2021) "Foundations of Ring Sampling."

Ring signature research:
Improved mimicking decoy selection

The decoy selection algorithm should match the real spend age probability distribution as closely as possible.

Research Questions:

1) How to estimate the real spend age distribution of Monero users?

2) How to “fit the curve” to translate the estimate into a decoy selection algorithm?

- **Dynamic (adjust over time) or static (remain the same over time)?**
- **Parametric (simple mathematical formula) or nonparametric (free-form)?**

Optimal Static Parametric Estimation of Arbitrary Distributions (OSPEAD) is my own project that proposes answers to these questions.[1]

[1] https://github.com/monero-project/research-lab/issues/93

Ring signature research:
Transaction flooding detection

Research questions:

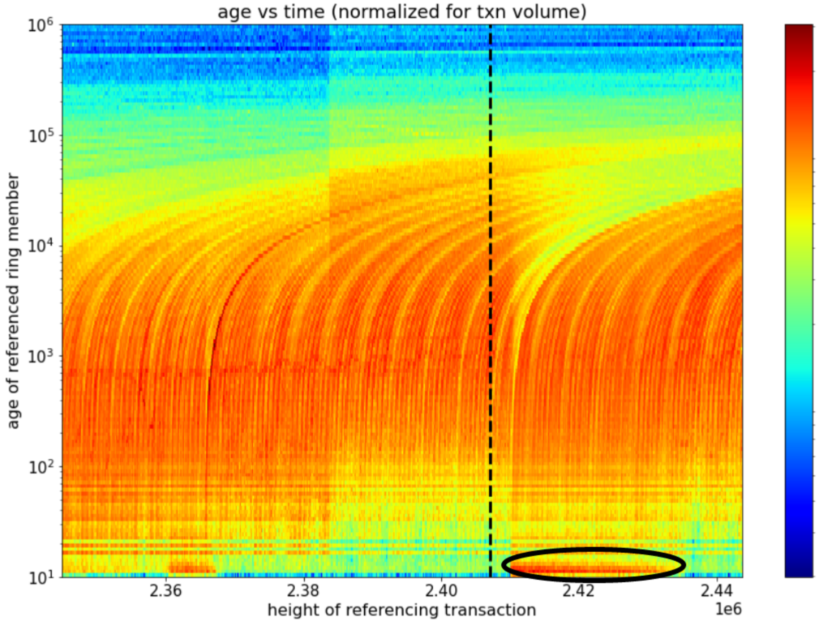
1) Is there a general method for detecting attempted de-anonymizing flooding episodes?

2) Are there any viable flooding countermeasures, like a decentralized counter-flood?

Analysis of flooding impact on user privacy (“black marbles”):

Surae Noether, Sarang Noether, & Adam Mackenzie (2014) “A Note on Chain Reactions in Traceability in CryptoNote 2.0.”

Chervinski, Kreutz, & Yu (2021) “Analysis of Transaction Flooding Attacks Against Monero.”



Graph from Isthmus (Mitchell P. Krawiec-Thayer), Neptune, Rucknium, Jberman, & Carrington (2021) "Fingerprinting a flood: forensic statistical analysis of the mid-2021 Monero transaction volume anomaly"

More ring signature questions

- Can changing the decoy selection algorithm between hard forks allow adversaries to fingerprint old/new wallet versions?
- How should decoys be chosen when a new transaction type is implemented, e.g. Seraphis transaction type?[1]
- Are there any big downsides to excluding coinbase outputs from the standard decoy selection algorithm? (Coinbase exclusion is a planned change.)[2]
- Can the 10 block lock on spending new outputs be reduced or eliminated? Can wallet-level changes like Monerujo's proposed PocketChange negatively affect user privacy?[3]

[1] https://github.com/monero-project/research-lab/issues/94#issuecomment-1098385712
[2] https://github.com/monero-project/research-lab/issues/109
[3] https://github.com/monero-project/research-lab/issues/102 and
Borggren & Yao (2020) "Correlations of Multi-input Monero Transactions."

Statistical research questions beyond ring signature issues

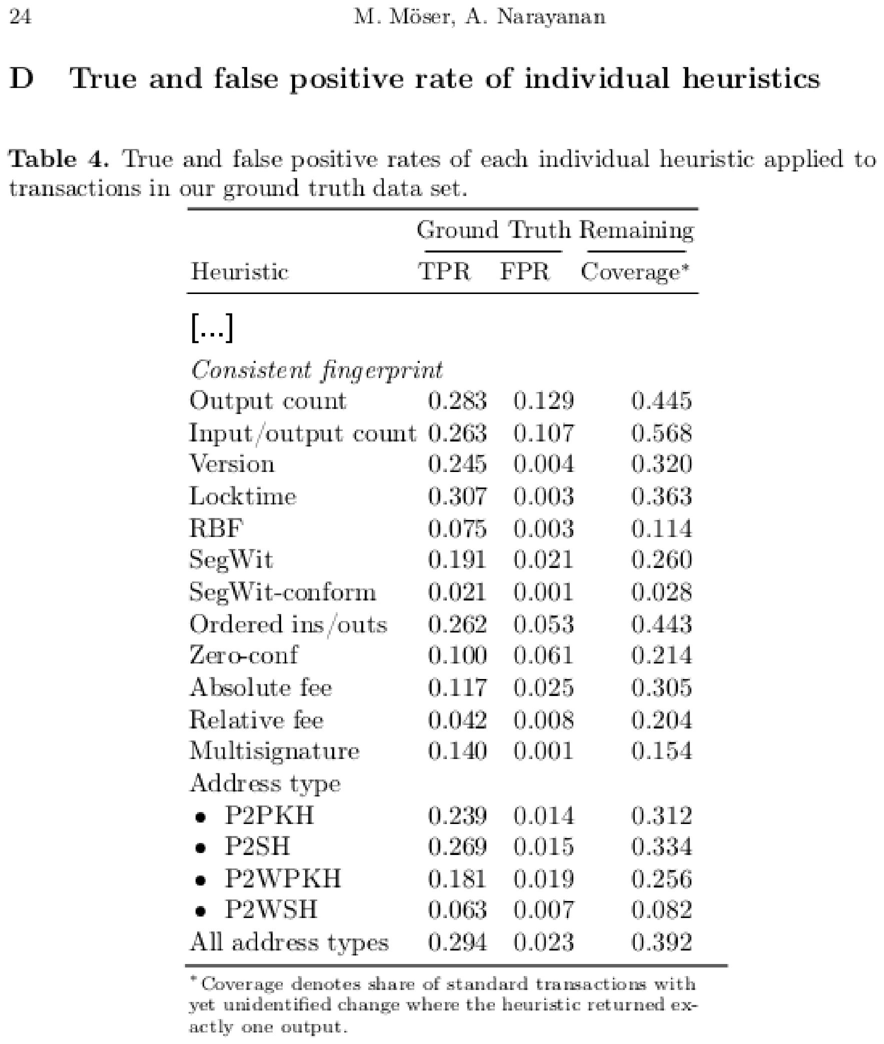
Transaction format strictness

- Variations in transaction format are a goldmine for cryptocurrency tracing.

- In bitcoin-like blockchains, wallet software "fingerprints" can indicate when coins have changed custody.

- "The design [of bitcoin] supports a tremendous variety of possible transaction types that I designed years ago."

  - Satoshi Nakamoto, 2010



24 M. Möser, A. Narayanan

D True and false positive rate of individual heuristics

Table 4. True and false positive rates of each individual heuristic applied to transactions in our ground truth data set.

	Ground Truth		Remaining
Heuristic	TPR	FPR	Coverage*
[...]			
Consistent fingerprint			
Output count	0.283	0.129	0.445
Input/output count	0.263	0.107	0.568
Version	0.245	0.004	0.320
Locktime	0.307	0.003	0.363
RBF	0.075	0.003	0.114
SegWit	0.191	0.021	0.260
SegWit-conform	0.021	0.001	0.028
Ordered ins/outs	0.262	0.053	0.443
Zero-conf	0.100	0.061	0.214
Absolute fee	0.117	0.025	0.305
Relative fee	0.042	0.008	0.204
Multisignature	0.140	0.001	0.154
Address type			
• P2PKH	0.239	0.014	0.312
• P2SH	0.269	0.015	0.334
• P2WPKH	0.181	0.019	0.256
• P2WSH	0.063	0.007	0.082
All address types	0.294	0.023	0.392

* Coverage denotes share of standard transactions with yet unidentified change where the heuristic returned exactly one output.

Table from:

Möser & Narayanan (2022) “Resurrecting Address Clustering in Bitcoin”

Monero transaction format: Loose aspects changed to strict

- Could use amount-hiding RingCT or old format with transparent amounts
  - 2017: All transactions must use RingCT (except under special circumstances)
- User-chosen ring size.
  - 2018: All transactions have same ring size
- Custom wallet software doesn't follow 10 block lock on spending new transaction outputs
  - 2019: 10 block lock enforced by blockchain consensus rules
- Transactions with a single output allowed, indicating a likely self-spend
  - 2019: All transactions must have at least two outputs (with possible zero-amount outputs)

Making Monero's transaction format more strict: **Custom lock time**

- Users can prevent their transaction outputs from being spent for a custom amount of time by setting **unlock_time**.
- Downsides: Code complexity, transaction fingerprinting, risk to merchants who don't check **unlock_time** when receiving payments.
- Upsides: Not many. **Research question: Can we discover any useful applications with custom lock time?**
- The current version of Seraphis code removes the option to set a custom **unlock_time**.

Monero Research Lab issue:
https://github.com/monero-project/research-lab/issues/78

Making Monero's transaction format more strict: **Fee discretization**

- The “standard” Monero wallet software has 4 fee levels.
- But other wallet software is free to set fees, which could accidentally fingerprint the wallet software. (This actually happened with MyMonero. Now fixed.)[1]
- Problem can be worse in Monero since it has 12 digits of precision after the decimal point. Bitcoin has 8.
- Current Seraphis code[2]: Blockchain consensus rules would require fees that are a power of 1.5, rounded to 1 significant digit: $round\left(1.5^x\right)$

[1] https://github.com/mymonero/mymonero-core-cpp/pull/36
[2] https://gist.github.com/UkoeHB/f508a6ad973fbf85195403057e87449e#transaction-uniformity

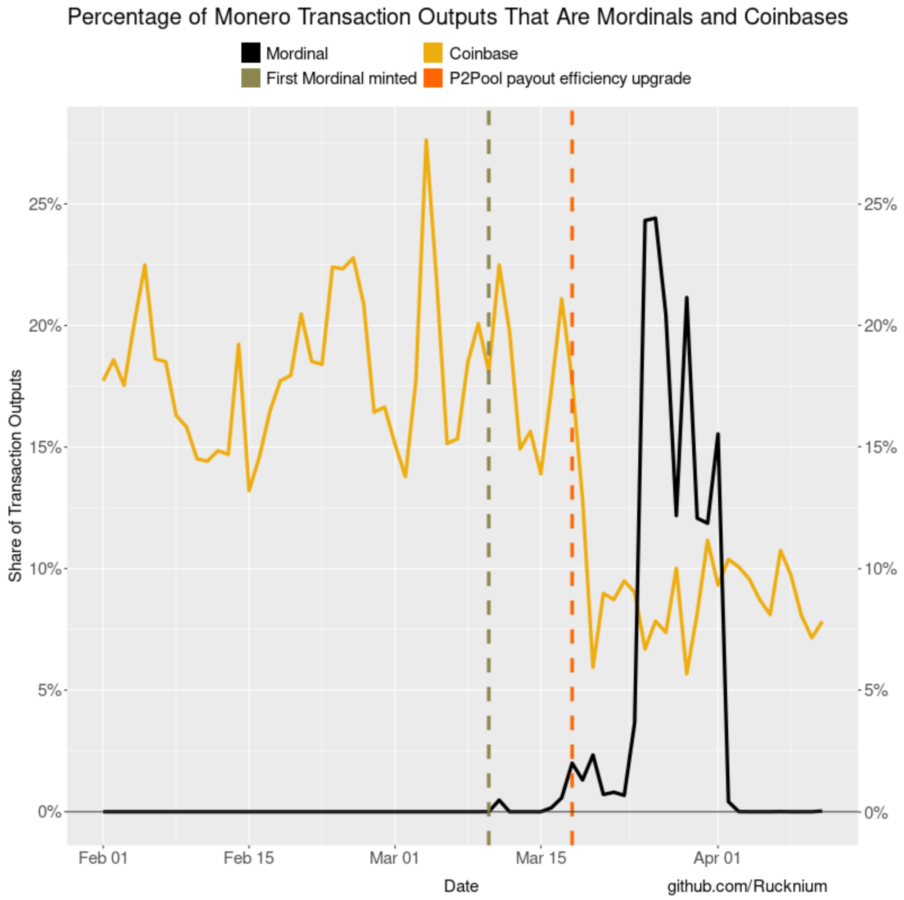
Making Monero's transaction format more strict: **Fee discretization (cont.)**

Research questions:

1) What form of discretization is best? Any potential problems with exponents?
2) Would fee discretization create “fee bubbles” because users cannot select intermediate fees?
3) How should wallet software estimate the fee needed to be included in the blockchain at high transaction volumes?

Making Monero's transaction format more strict: **Prohibit arbitrary data in transactions**

- Using **tx_extra**, the Mordinals protocol created “NFTs”.
- Eliminating **tx_extra** does not prevent arbitrary data in transactions. Data can be injected into the parts of transactions intended for cryptographic elements, e.g. public keys.



Making Monero's transaction format more strict: **Prohibit arbitrary data in transactions (cont.)**

Research questions:

1) Can a statistical test detect and exclude arbitrary data with acceptable tradeoffs?

   - Low power candidates: Shannon’s entropy, Chi-square test

   - High power candidates: Binary matrix rank, book stack, birthday spacings.

2) Can cryptography be used to enforce encryption when the keys are unknown? (Probably not)

Making Monero's transaction format more strict: **Require a standard decoy selection algorithm**

- Non-standard wallet software can choose decoys for ring signatures any way its developers want.
- Fingerprinting issue that will get worse as ring size increases. More data (more ring members) means greater ability to statistically distinguish them.
- We know that there are multiple non-standard decoy selection algorithms being used “in the wild” on the Monero blockchain.
  - Includes “cached” ring members that directly reveal the real spend (Mordinals and isthmus’s Nov. 2021 discovery)

Making Monero's transaction format more strict:
Require a standard decoy selection algorithm (cont.)

Research questions:

1) Should a standard decoy selection algorithm be required? What are at the downsides?
2) Should the requirement be a blockchain consensus rule or just a node transaction relay rule?
   - For example, minimum fee is relay, not consensus. New restriction on **tx_extra** size is relay rule.
3) If standardized, how often could the decoy selection algorithm be updated?
4) Could you have an automatically-updating algorithm based on changing aggregate user behavior? Record the algorithm in the block headers to “conduct the orchestra”?
5) Could auto-updating allow adversaries to manipulate it?

Monero Research Lab issue: https://github.com/monero-project/research-lab/issues/87

Making Monero's transaction format more strict: **Restrict number of inputs/outputs per transaction**

- Now: Between 2 and 16 outputs per transaction allowed. Number of inputs unlimited, but transaction must fit in a single Monero block.
- Seraphis proposal: 2-16 outputs. 1-112 inputs.
- The number of inputs/outputs may reveal some information about the user.
  - Many inputs: Could be a merchant consolidating payments
  - Many outputs: Could be an exchange or mining pool
- Many inputs can reveal information about one user owning several outputs, which gives more information than just single output ownership.[1]

[1] Borggren & Yao (2020). “Correlations of Multi-input Monero Transactions.”

Making Monero's transaction format more strict:

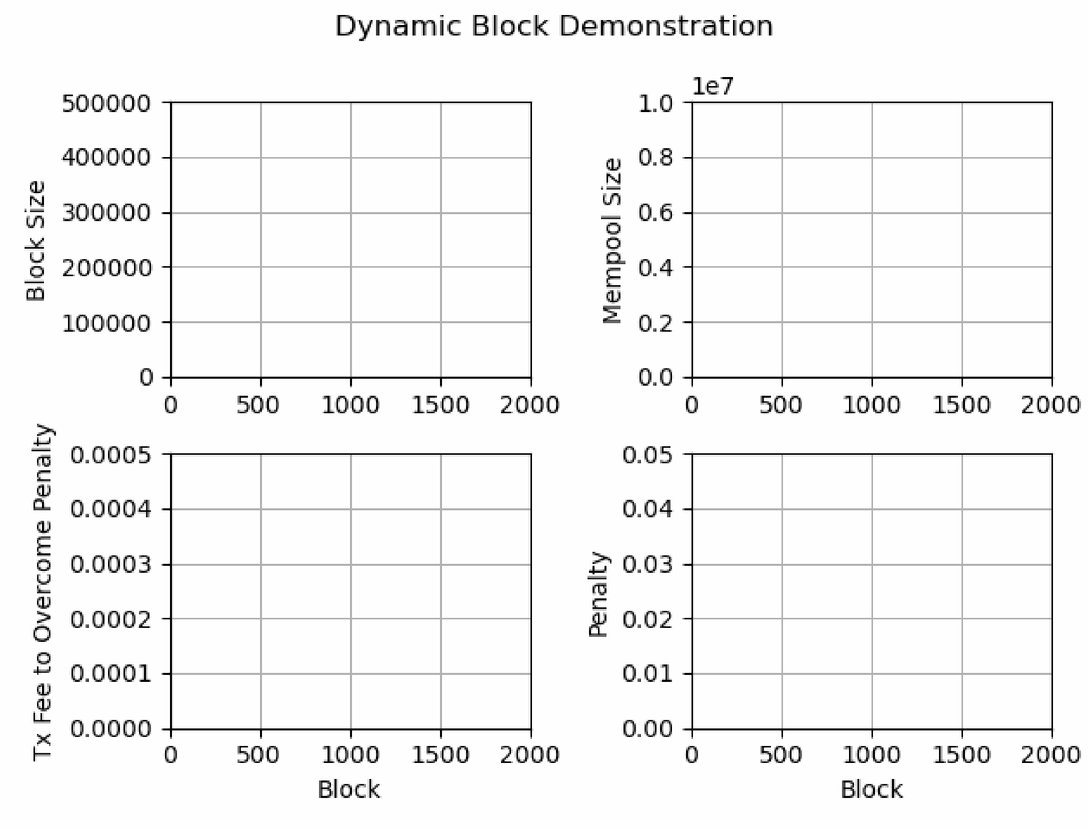
Restrict number of inputs/outputs per transaction (cont.)

- Extreme proposal: Require all transactions to have only 2 inputs and 2 outputs.
- Could be major annoyance for entities like merchants and exchanges that usually use many inputs and outputs, especially with the 10 block lock.
- **Research questions:**

1) **How common are 3+in/3+out transactions, in Monero and transparent blockchains?**
2) **What is the privacy benefit of requiring 2in/2out?**
3) **How much of an inconvenience would requiring 2in/2out be?**

Dynamic block size and fee policy

Monero’s dynamic block size has not received the same amount of research scrutiny as its privacy features.



Graphs created by spackle-xmr
https://github.com/spackle-xmr/Dynamic_Block_Demo

Dynamic block size and fee policy (cont.)

- Research questions:

1) Can the dynamic block size parameters result in undesirable outcomes, e.g. too fast or too slow block size increase?
2) The interaction of block size and fee policy is supposed to adjust fee to the purchasing power of a unit of XMR in the future (a type of "oracle" problem). Can this go wrong?[1]
3) Is it possible to have a fee policy that discourages adversarial spam but provides low fees for people around the globe?

[1] "Bidding oracle": Huberman, Leshno, & Moallemi (2021). "Monopoly without a Monopolist: An Economic Analysis of the Bitcoin Payment System."

Dynamic block size and fee policy (cont.)

- The dynamic block size rules assume miners will choose to raise the block size when there is “fee pressure” to maximize their profits.
- **Research questions:**

1) **Is raising block size the economically rational choice for miners?**
2) **Are miners fully rational or do they have “bounded rationality”?**
   - I discovered that mining pools were “leaving fees on the table” until I informed them of their configuration problem.

Related papers:

Sapirshtein, Sompolinsky, & Zohar (2017) “Optimal Selfish Mining Strategies in Bitcoin.”

Hou et al. (2021) “SquirRL: Automating Attack Analysis on Blockchain Incentive Mechanisms with Deep Reinforcement Learning.”

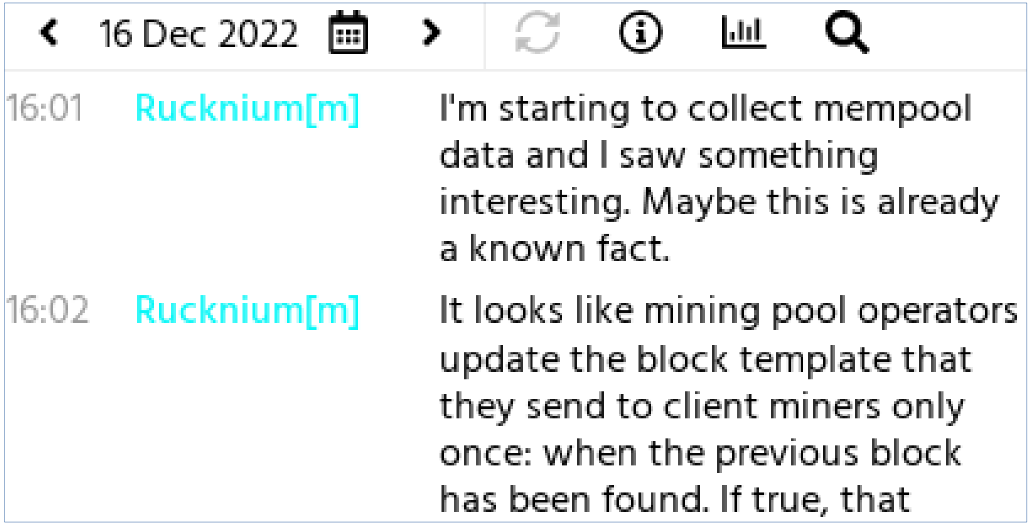
Lee & Kim (2022) “Rethinking Selfish Mining Under Pooled Mining.”

Reducing mining pool centralization

- Research question:
- **Could a dynamic mining pool fee be developed that would encourage miners to join smaller pools (or P2Pool) when a mining pool has a large percentage of total hash rate?**
- The MineXMR pool raised fees when it gained a large share of hard rate in 2022.
- The dynamic fee would require voluntary adoption by mining pool operators

Surprise unforeseen research questions

- Some of the best statistical research questions appear when you happen to see something unexpected in the data.
- When I discovered that transaction confirmations could be sped up by 60 second, I was working on researching fee discretization, a completely different topic.



16 Dec 2022

16:01 Rucknium[m] I'm starting to collect mempool data and I saw something interesting. Maybe this is already a known fact.

16:02 Rucknium[m] It looks like mining pool operators update the block template that they send to client miners only once: when the previous block has been found. If true, that

How to get to work on the statistical research agenda?

- Active Monero Research Lab statistical researchers: me (Rucknium), ACK-J, isthmus (Mitchell P. Krawiec-Thayer), and neptune.
- We need more:
  - "The Monero Project should actively recruit technical talent from universities"
  - https://www.reddit.com/r/Monero/comments/pkg3d6/the_monero_project_should_actively_recruit/
- The MAGIC Monero Fund put requests for research proposals in research grant databases.
- We may have a new research project on EAE attacks soon.
- Recruitment is hard.

Questions and Feedback

Contact:
https://rucknium.me
https://github.com/Rucknium
Rucknium@Protonmail.com