Seraphis/Jamtis

Overview of proposed upgrades to core features in Monero

BACKGROUND

Triptych (‘20/’21): larger ring sizes, not-so-great multisig.

https://eprint.iacr.org/2020/018.pdf

Lelantus Spark (~‘21): larger ring sizes, nice multsig, developed for Firo around the same time as Seraphis.

https://eprint.iacr.org/2021/1173.pdf

Seraphis: larger ring sizes, nice multisig, modular transaction protocol purpose-built for Monero with a suite of new features.

https://github.com/UkoeHB/Seraphis

Jamtis: a set of well thought out upgrades to Monero’s core features.

https://gist.github.com/tevador/50160d160d24cfc6c52ae02eb3d17024

FEATURE OVERVIEW

1. “View Balance” key
2. Light wallets with stronger privacy properties
3. New addresses
4. Simplified address scheme
5. Certified addresses
6. Recipient Identifiers (RIDs)
7. “Generate Address” key
8. Transaction chaining

View Balance key

VIEW BALANCE KEY

Private key that can view all incoming ***and outgoing*** funds.

PROS

- Enables a safer, fully featured watch-only wallet.
- Vastly improves the offline cold signing, hardware wallet, and multisig user experience.
- Has materially similar privacy properties to **today’s** view key.
- Enables light wallets with stronger privacy properties. Will be discussed further.

VIEW BALANCE KEY

Private key that can view all incoming ***and outgoing*** funds.

CONS

- **After ring sizes increase a significant amount**, view balance keys can be more powerful surveillance tools than view keys at that point.

  Outcome 1: powers that be mandate sharing view balance keys.

  > *Users can already be compelled to give up view keys and signed key images today.*

  Outcome 2: a centralized wallet service may collect view balance keys to offer instant wallet loading.

  > *Mitigated by an improved light wallet tier. Next section.*

- *https://github.com/monero-project/research-lab/issues/92#issuecomment-1146810255*

Light wallets with stronger privacy

LIGHT WALLETS W/STRONGER PRIVACY

- In Monero, a user must scan all transactions on the blockchain since wallet creation to identify theirs. This can be *slow*.

- Light wallets offload scanning to a separate device (to a “light wallet server”).

- Light wallet servers:

  monero-lws: *https://github.com/vtnerd/monero-lws*
  openmonero: *https://github.com/moneroexamples/openmonero*

- Today, a light wallet server can identify a user’s outputs and see amounts.

THE PROPOSED UPGRADE...

LIGHT WALLETS W/STRONGER PRIVACY

A light wallet server that **cannot see amounts**, and **cannot definitively identify a user's outputs** ("enotes") *

* so long as a user **does not reuse an address** to receive Monero

LIGHT WALLETS W/STRONGER PRIVACY

A light wallet server that **cannot see amounts**, and **cannot definitively identify a user's outputs** ("enotes") *

WIP

* so long as a user **does not reuse an address** to receive Monero

LIGHT WALLETS W/STRONGER PRIVACY

PROS

- A user experience ideally on par with today’s light wallet servers, with a gain in privacy.
- Can reduce the threat a centralized light wallet server poses to the anonymity set.
- Run a light wallet server at home for yourself, family, and friends. Scale “Uncle Jim.”

LIGHT WALLETS W/STRONGER PRIVACY

Credit: @Diverter_NoKYC

LIGHT WALLETS W/STRONGER PRIVACY

CONS

- If a user reuses an address, they reveal to the light wallet server which enotes were received to that address.
- Timing analysis may enable the server to make decent guesses at enotes spent and received.
- Receiving 2+ enotes in a transaction can more easily enable a server to identify that those enotes belong to the user.

New addresses

NEW ADDRESSES

- Practically large rings require a migration to new addresses.
- What this means for users:
  - Creating a new seed is NOT required, old seeds would work fine.
  - Funds *received* in the past to an old address would ALWAYS be spendable.
  - Funds *sent* after the upgrade must ALWAYS be sent to a new address.

NEW ADDRESSES

OLD

46nWts8E8QDgeAjiH3XnWu3sHaVvMTGkRY
LapJhiJmGvdL5k6jQh3UZc8z5Qq3mcYB9nS
mPC4YH26L5EAeyETeUkVgEqJGZ

NEW

xmr1majob1977bw3ympyh2yxd7hjymrw8c
rc9kinodkm8d3wdu8jdhf3fkdpmgxfkbywbb
9mdwkhkya4jtfnod5h7s49bfyji1936w19tyf3
96ypjo9n64runqjrxwp6k2s3phxwm6wrb5co
b6c1ntrg2mugeocwdgnnr7u7bgknya9arksrj

- base-58 (old) vs base-32 (easier to read/copy)
- 1 extra public key in new

Seraphis presents a unique opportunity to use years of user feedback to vastly improve the Monero address scheme.

Jamtis is seizing this opportunity.

The result is a robust, feature complete, simplified address scheme.

Simplified address scheme

SIMPLIFIED ADDRESS SCHEME

- Today in Monero there are 3 address types:
- **Standard addresses**: the default wallet address. 1 per wallet.

  49gZEMFG...
- **Subaddresses**: use 1 wallet to receive to different addresses.

  84waNJ28...
- **Integrated addresses**: address + payment ID. Enables merchants to embed unique payment identifiers on an address for order fulfillment, while also maintaining a trusted address for repeat customers.
- Jamtis proposes:
  - Eliminating the difference between standard/subaddresses in favor of Jamtis addresses.
  - Replacing integrated addresses with “certified” addresses.

SIMPLIFIED ADDRESS SCHEME

Problems with subaddresses today

- Vulnerable to a Janus attack.
- Generating subaddresses requires keeping track of the ones already provisioned.
- Recovering funds received to subaddresses *automatically* is not robust.

  Example: *https://github.com/monero-project/monero/issues/8138*
- Transactions with 3+ outputs involving subaddresses are identifiable as such on chain.
- Subaddresses take extra development effort to support and increase the complexity of the transaction protocol.

SIMPLIFIED ADDRESS SCHEME

Problems with subaddresses today (cont.)

- Best practice is to use a new subaddress with every counter-party; the existence of the standard address detracts from this.
- The choice itself to use a standard address or subaddress is a tidbit of metadata.
- “Should I use a subaddress? Should I stick with the standard? I don’t know!” (relevant XKCD)

SIMPLIFIED ADDRESS SCHEME

THERE! IF WE STEAL ONE OF THOSE CARS, WE CAN GET TO THE BASE AND DEFUSE THE BOMB!
HMM, THE ONE ON THE LEFT ACCELERATES FASTER BUT HAS A LOWER TOP SPEED.
OOH, THE RIGHT ONE HAS GOOD TRACTION CONTROL. ARE THE ROADS WET?

PROTIP: IF YOU EVER NEED TO DEFEAT ME, JUST GIVE ME TWO VERY SIMILAR OPTIONS AND UNLIMITED INTERNET ACCESS.

https://xkcd.com/1801/

SIMPLIFIED ADDRESS SCHEME

Jamtis address solutions

- No more “primary” address, *just* Jamtis addresses!
- Jamtis addresses can be generated randomly and offline.
  - Compatible with UUID’s thanks to 16-byte address indexes.
- Jamtis addresses can be recovered without needing a lookahead. They are simply decrypted in real-time when scanning transactions.
- Mitigate the Janus attack.
- Single “Jamtis address” type means easier to develop for. One and done.

Certified addresses

CERTIFIED ADDRESSES

Integrated addresses can be useful for merchants who want to generate unique payment identifiers for each order, while maintaining a static address for repeat customers.

PROBLEMS

- Can’t send to >1 integrated address in a single transaction.
- All transactions have an 8-byte payment identifier on chain.
- Proposal to deprecate: *https://github.com/monero-project/monero/issues/7889*

SOLUTION

- **Certified addresses** are uniquely generated Jamtis addresses that are *signed by a single private key*.
- Users add a merchant’s RID to an address book, and all future payments made to the certified address are safe. No MITM risk.

Recipient Identifiers (RIDs)

RECIPIENT IDENTIFIERS (RIDs)

PROBLEM

- Copy pasting and visually comparing addresses sucks.

SOLUTION

xmr1majob1977bw3ympyh2yxd7hjymrw8crc9kinodkm8d3wdu8jdhf3fkdpmgx
fkbywbb9mdwkhkya4jtfnod5h7s49bfyji1936w19tyf396ypjo9n64runqjrxwp6k2
s3phxwm6wrb5cob6c1ntrg2mugeocwdgnnr7u7bgknya9arksrj

↓

regne-hwbna-u21gh-b54no-8x36q

“Generate Address” key

GENERATE ADDRESS KEY

PROBLEMS

- Generating new subaddresses today requires access to a private view key.
- Integrated addresses can be generated without a a private view key, but they have their own set of problems.

SOLUTION

- A new “Generate Address” key that can ONLY *generate* Jamtis addresses.

Transaction chaining

TRANSACTION CHAINING

PROBLEMS

- Today it’s impossible to “pre-sign” a transaction that spends outputs from a transaction that does not yet exist in the chain.
- Without pre-signing, atomic swaps only work where Bitcoin moves first.

  See: *https://comit.network/blog/2021/07/02/transaction-presigning*

SOLUTION

- Transaction chaining would enable someone to do exactly that: “chain” two transactions together before submitting to the blockchain, such that the 2nd spends from the 1st.

Thank you!

FEEDBACK WELCOME

Feedback from all welcome & encouraged, especially merchants.

Review the specs in much greater detail and chime in:

https://gist.github.com/tevador/50160d160d24cfc6c52ae02eb3d17024

https://github.com/monero-project/research-lab/issues/92

Or ask questions in IRC/matrix channels:

#monero-research-lounge (less formal)

#monero-research-lab

Questions