OSPEAD:
Optimal Ring Signatures

Research by Rucknium at the Monero Research Lab

Voiced by xenu
MoneroKon 2025

Goals

- An overview of how OSPEAD works.
  - Deep technical details: github.com/Rucknium/OSPEAD
- Implications for Monero user privacy
- Prospects for deployment of an improved decoy selection algorithm on Monero’s mainnet

Ring signatures

- Ring signatures hides which set of coins is spent in a Monero transaction by grouping the actually spent coin set with 15 other coin sets.
  - Ideally, an observer cannot know which of the 16 ring members was truly spent.

- The coin set that was actually spent is called the “real spend”.

Preview of results

- An anti-privacy adversary can use timing information in the Monero blockchain data to correctly guess the real spend in a ring signature with 23.5% probability.

- An improved decoy selection algorithm could reduce this probability to 7.6%.

Privacy implications

- The reduction in user privacy only affects one of Monero’s privacy techniques: Ring signatures, which help provide privacy for the sender of a transaction.
- Not affected:
  - **Confidential transactions**, which hide the amount of coins being sent
  - **Stealth addresses**, which help provide transaction receiver privacy
  - **Dandelion++**, which helps hide the IP address origin of transactions on the network.

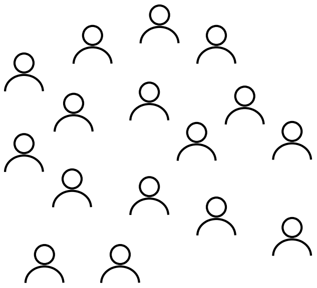
Privacy implications

- The attack on ring signature privacy might only be relevant for users with extreme threat models.

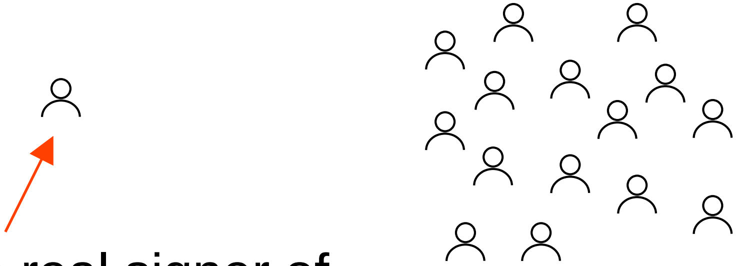
- The attack assumes that the adversary tries to guess the real spend on every single transaction. There is no “confirmation” of whether the guess is actually correct.

Ring signatures: Hiding in a crowd

Good:



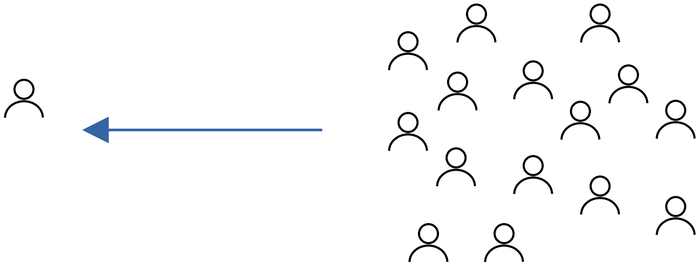
Bad:



Probably the real signer of a transaction

To serve its purpose, a decoy must look like the real thing, i.e. the real transaction signer.

Move the decoy crowd to the real signer



The timing, or age, of a transaction is its “location” on the blockchain

Therefore, move the age of the decoys to where the real signers are located.

Need for this was recognized early

“One solution to this problem is to determine a non-uniform method of choosing transaction outputs for ring signatures; choose transaction outputs based on their age such that the probability that these outputs are chosen for a ring signature is inversely related to the probability that they have been spent already. This would suggest that we, the developers of Monero, must estimate the probability distribution governing the age of transaction outputs.”

– Monero Research Lab bulletin #4, 2015

OSPEAD: Improved decoys

- Funded by Monero’s Community Crowdfunding System (CCS)

- Uses proven techniques from traditional statistics, not machine learning or AI.

OSPEAD

Optimal

Static

Parametric

Estimation of

Arbitrary

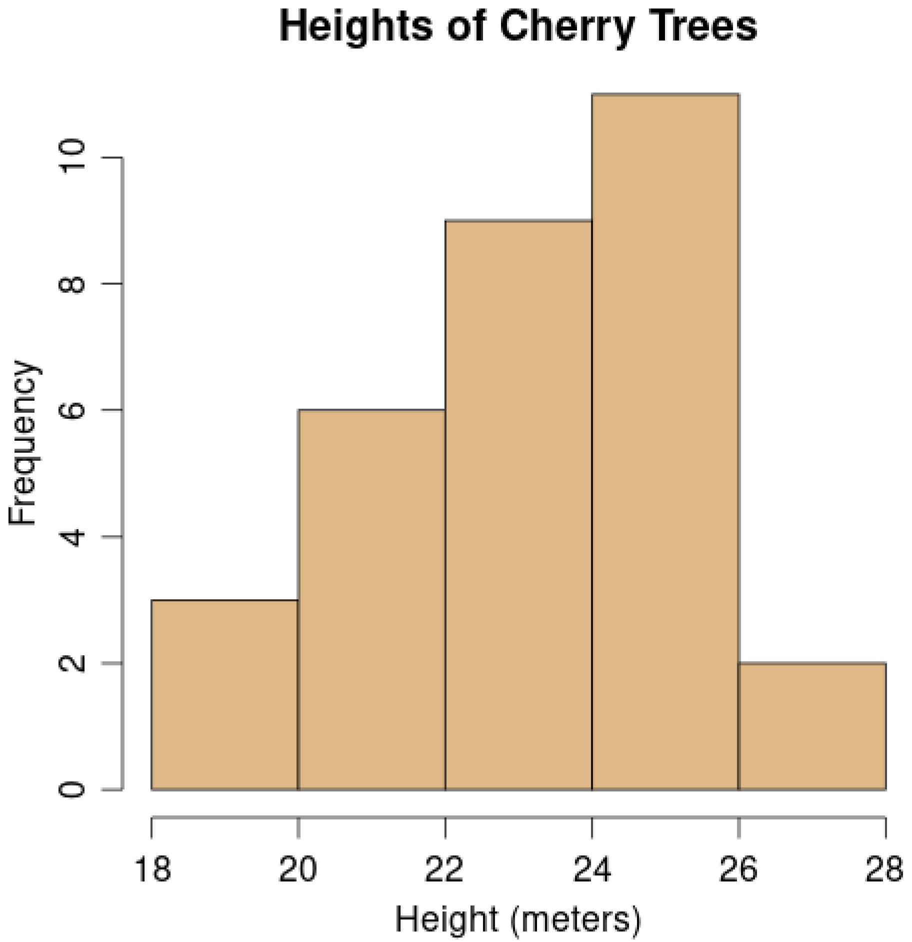
Distributions

Defining terms:
Probability Density Function (PDF)

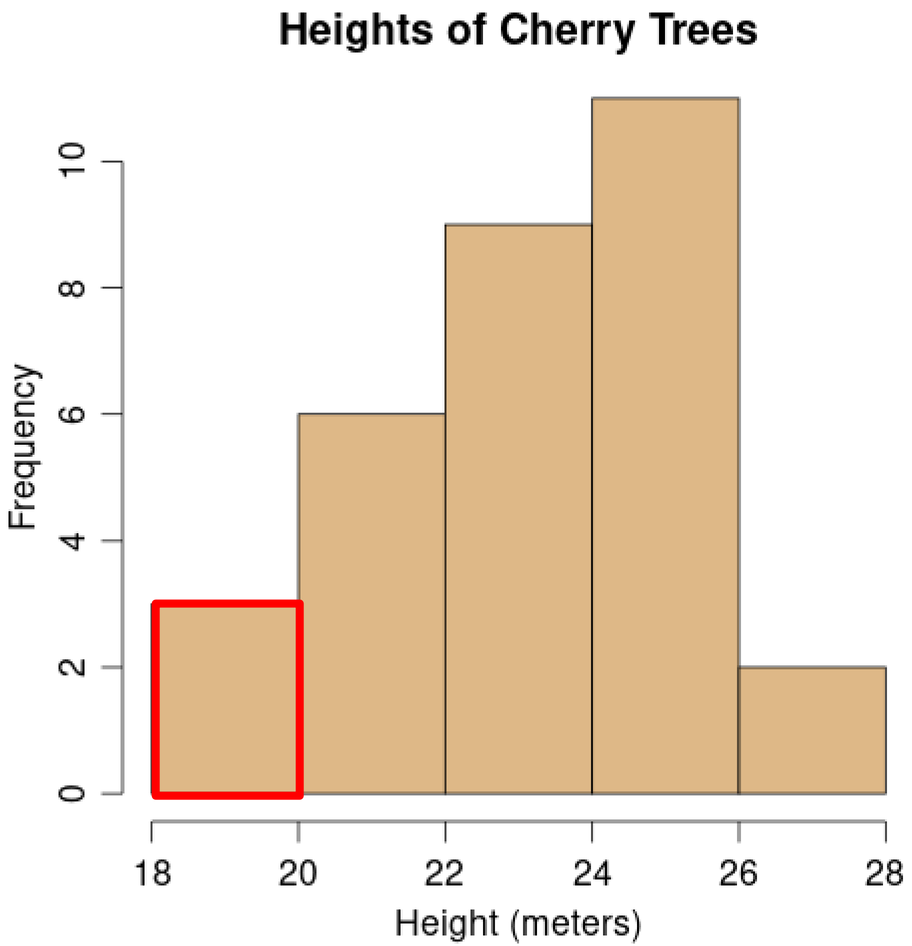
- A Probability Density Function (PDF) mathematically describes how likely a random variable’s value will occur.

- Usually, not all outcomes are equally likely.

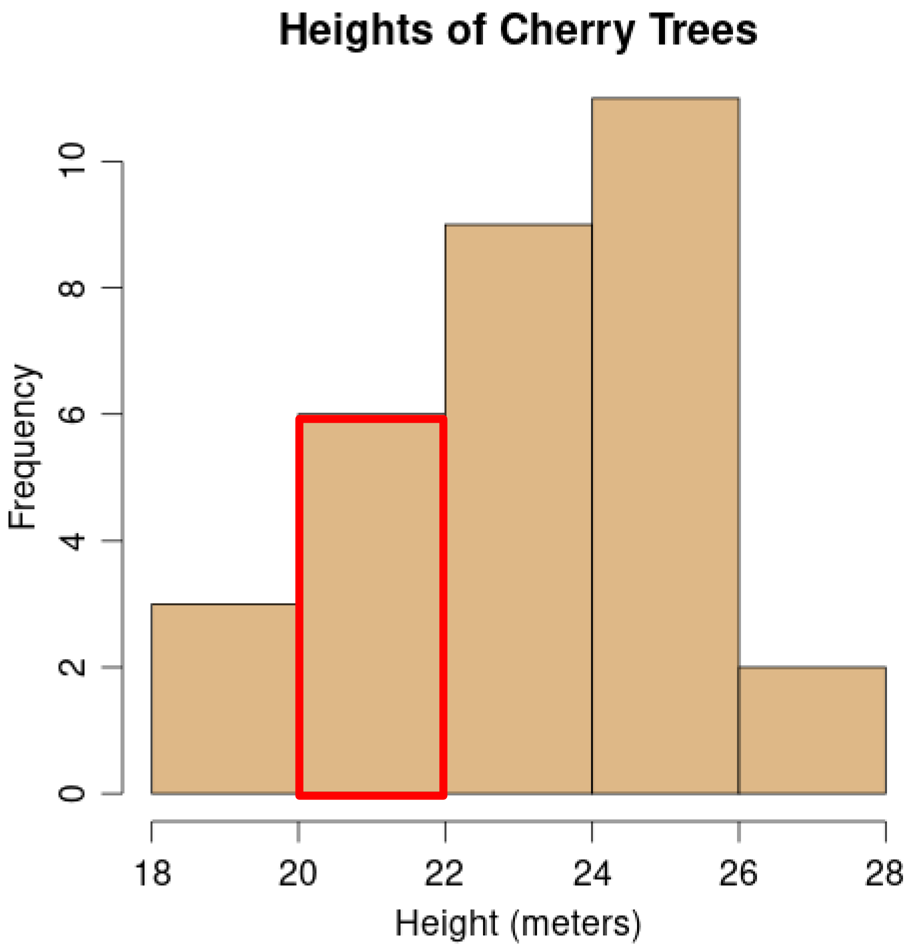
Histogram example



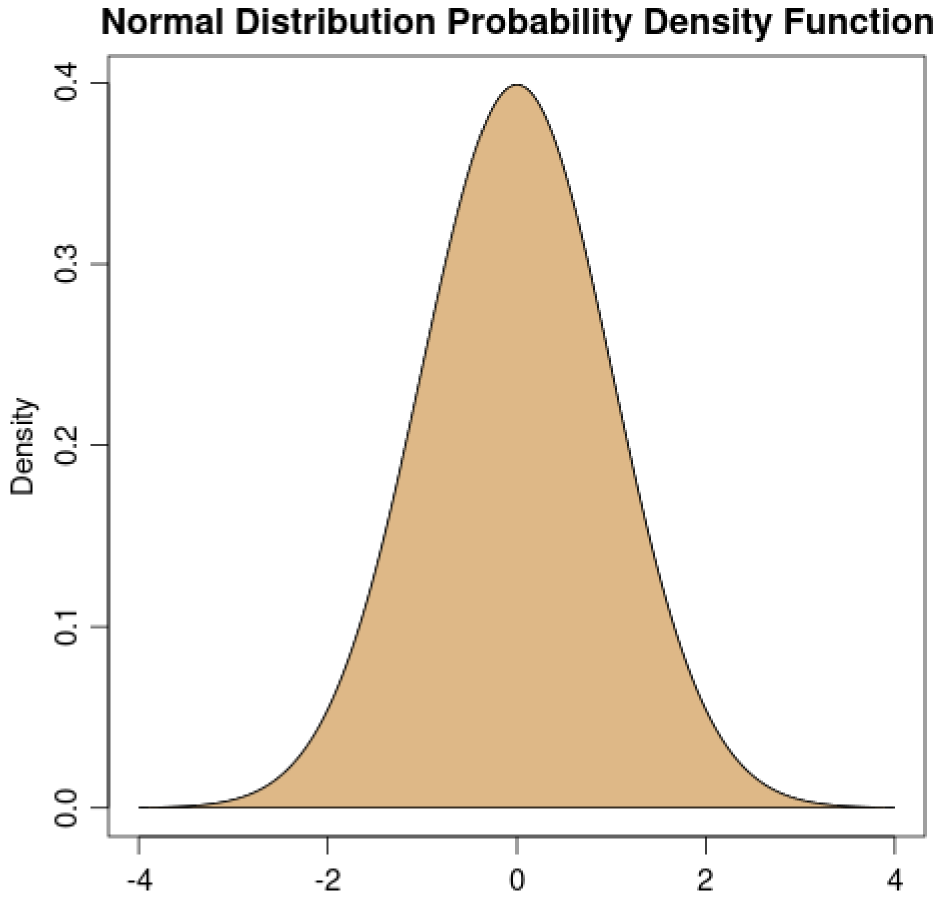
Histogram example



Histogram example

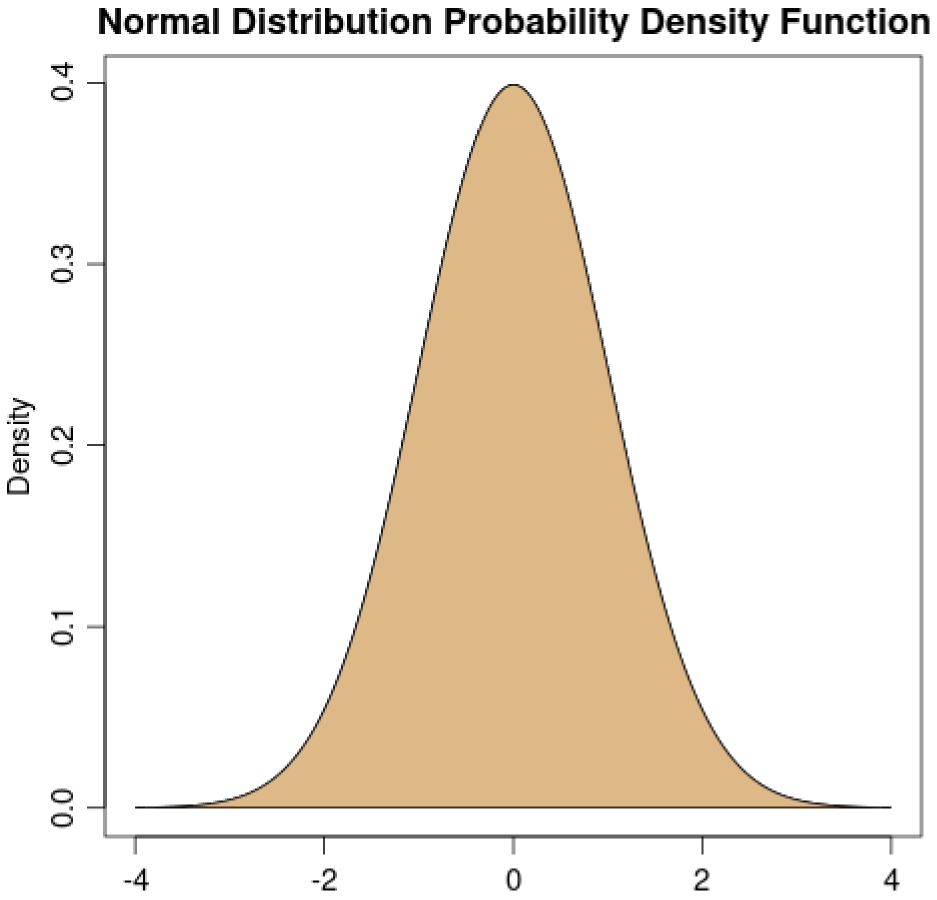


Normal distribution – “Bell curve”



PDF as a formula

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Monero wallets use a probability distribution to select 15 decoys from transactions in the past.

Adequate decoy probability distribution

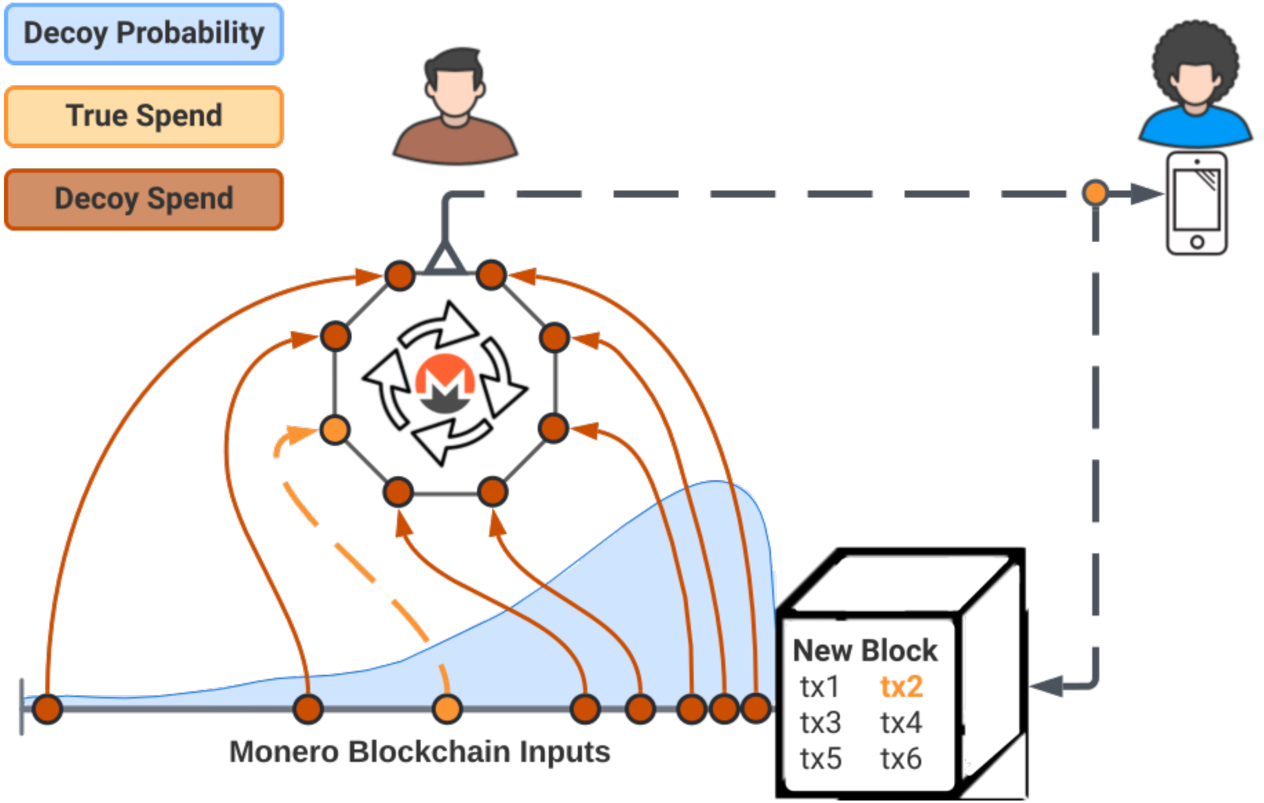


Image by ACK-J

Defective decoy probability distribution

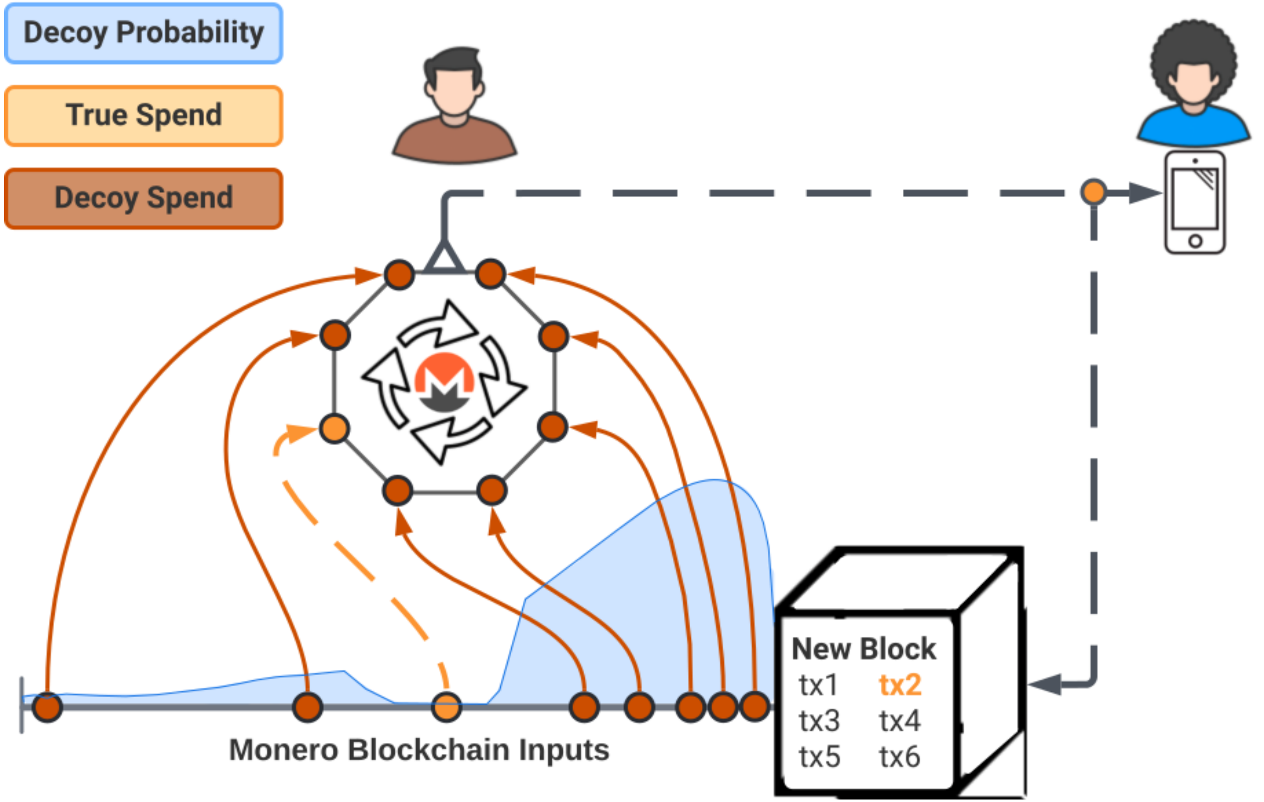


Image by ACK-J

How can you use these ideas to attack Monero user privacy?

Aeeneh et al., (2021) "New attacks on the untraceability of transactions in cryptonote-style blockchains."

Maximum a Posteriori (MAP) Decoder

- Based on time-tested ideas developed by Neyman, Pearson, and Fisher in the 1930s
  - MAP Decoder is the optimal timing-based attack
- Compute the ratio of two probability distributions
- The highest ratio is the most likely real signer

Defective decoy probability distribution

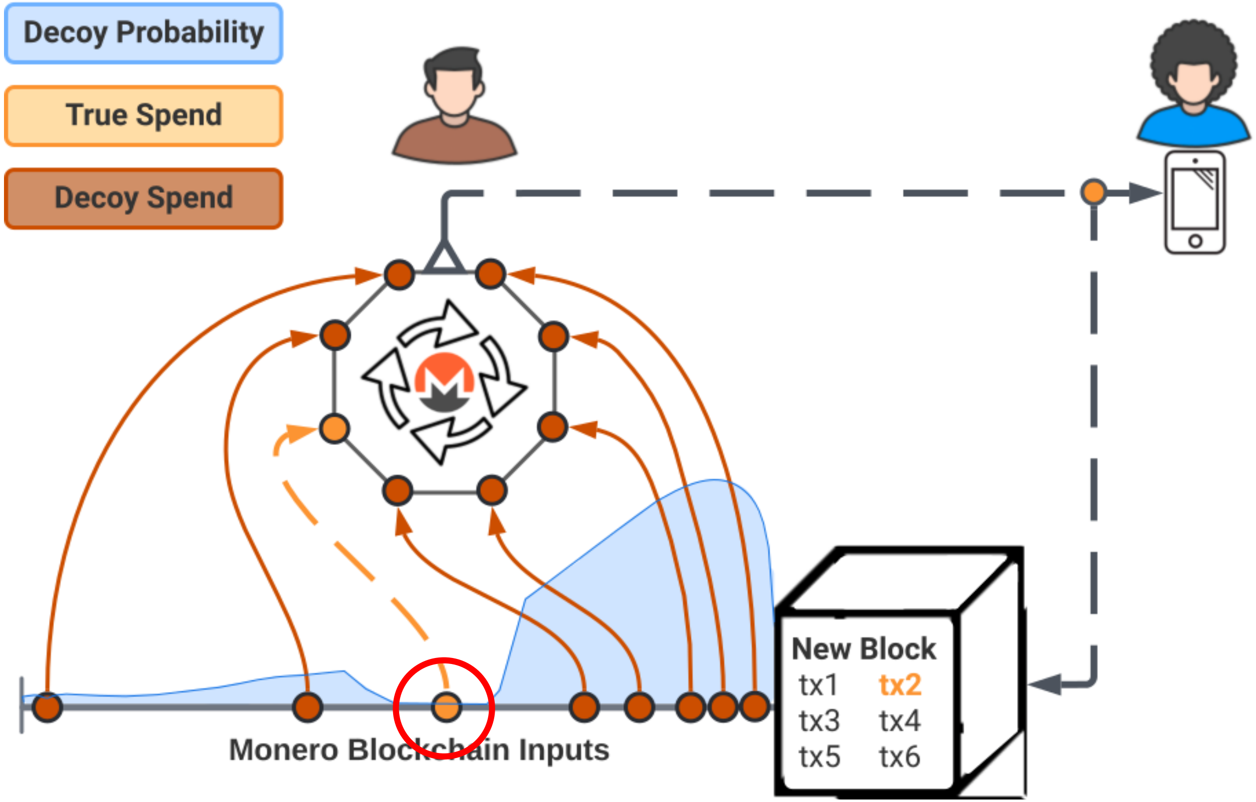


Image by ACK-J

PDF definitions

$f_D(x)$ Decoy distribution

$f_S(x)$ Real Spend distribution

MAP Decoder algorithm

1) Compute ratio for each ring member: $$\frac{f_S(x)}{f_D(x)}$$

2) Guess that the ring member with the highest ratio is the real spend

But, how do we determine $f_S(x)$, the real spend distribution?

Aren’t the real spends mixed with decoys in the blockchain data?

Definition: Mixture distribution

- A mixture distribution is a probability distribution formed by combining two or more probability distributions

Monero ring mixture distribution

$f(x)$ is the ring data on the blockchain (known)

$f_S(x)$ is the real spend distribution (unknown)

$f_D(x)$ is the decoy distribution (known)

1/16 is the proportion of real spends

15/16 is the proportion of decoys

$$f(x)=\frac{1}{16}f_S(x)+\frac{15}{16}f_D(x)$$

“Inversion estimator” from Patra & Sen (2016)

$$f(x) = \frac{1}{16} f_S(x) + \frac{15}{16} f_D(x)$$

↓

$$f_S(x) = 16 f(x) - 15 f_D(x)$$

Problem: Multiple wallet implementations

- Many transaction characteristics are prescribed by blockchain consensus rules.

- But wallet software is free to choose any decoy selection distribution.

Aeeneh at al. (2021)

“We showed that if the distribution used for selecting decoy [transaction outputs] are known, then a malicious party can significantly weaken the privacy features of CryptoNote by tracing back the transactions. However, our studies on Monero, a CryptoNote-based blockchain, show that users are using different source codes each having a different algorithms for mixin selection. Fortunately, since there is no distribution that a strong majority of users are using for mixin selection, an attacker cannot derive the distribution of truly spent transactions and arrange the [MAP Decoder] attack on Monero.”

Separating the non-standard decoy distributions

- The main contribution of the OSPEAD research is finding this missing ingredient.

- These non-standard decoy distributions form their own mixture distribution that can be broken apart, using the right technique.

BJR estimator

Bonhomme, Jochmans, & Robin (2016):

“Non-parametric estimation of finite mixtures from repeated measurements.”

BJR Estimator steps

1) Assume that there are up to 4 major decoy selection distributions being used “in the wild”

2) Run BJR on the blockchain ring data

3) Assume that the most common distribution corresponds to the “standard” wallet implementation.

4) Plug this distribution into the Patra-Sen inversion estimator.

Real spend distribution estimate is a double-edged sword

- It provides the necessary “ammunition” for the MAP Decoder “weapon” against user privacy
- However, deployment of a decoy selection distribution that matches the real spend distribution would make MAP Decoder ineffective

MAP Decoder effectiveness

- At current Monero ring size of 16, the theoretical minimum attack success probability through completely random guessing would be 1/16 = 6.25%.
- If an adversary used the MAP Decoder attack against the last few years of transactions, the real spend would be guessed correctly in 23.5% of cases.
- The OSPEAD techniques suggest a new decoy distribution, which would reduce the average attack success probability to 7.6%.

Privacy risk of deployment

- Without a hard fork, users are not required to update their wallet software.
- Users who update early could stand out from the crowd, until the majority updates.
- Research on this problem is ongoing
  - So far, it seems that the benefits of deployment would outweigh the risks

The future is bright

- Full-Chain Membership Proofs (FCMP) would eliminate ring signatures and their imperfect privacy.
  - The technology is expected to deploy on Monero’s mainnet “soon”.
- In the meantime, it is still worth the effort to improve ring signatures because real people are depending on ring signature privacy today.