# How to Squeeze a Crowd: Reducing Bandwidth in Mixing Cryptocurrencies

**Alishah Chator** and Matthew Green

Johns Hopkins University

# Do Androids Dream of Electronic Cash?

- Having a digital equivalent to cash has been an open Privacy topic for decades

# Do Androids Dream of Electronic Cash?

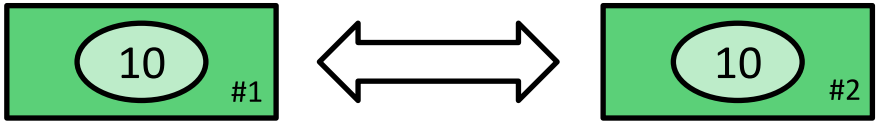
- Having a digital equivalent to cash has been an open Privacy topic for decades

- Any such system must be **fungible**: Any unit of currency is interchangeable with any other unit of equivalent value

# Do Androids Dream of Electronic Cash?

- Having a digital equivalent to cash has been an open Privacy topic for decades

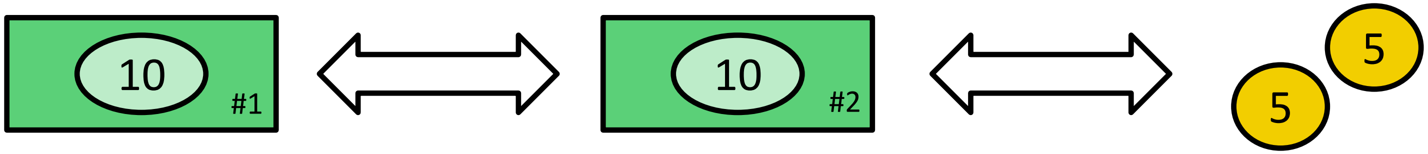
- Any such system must be **fungible**: Any unit of currency is interchangeable with any other unit of equivalent value



# Do Androids Dream of Electronic Cash?

- Having a digital equivalent to cash has been an open Privacy topic for decades

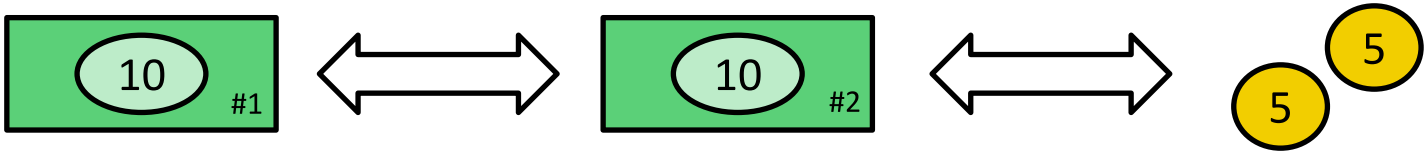
- Any such system must be **fungible**: Any unit of currency is interchangeable with any other unit of equivalent value



# Do Androids Dream of Electronic Cash?

- Having a digital equivalent to cash has been an open Privacy topic for decades

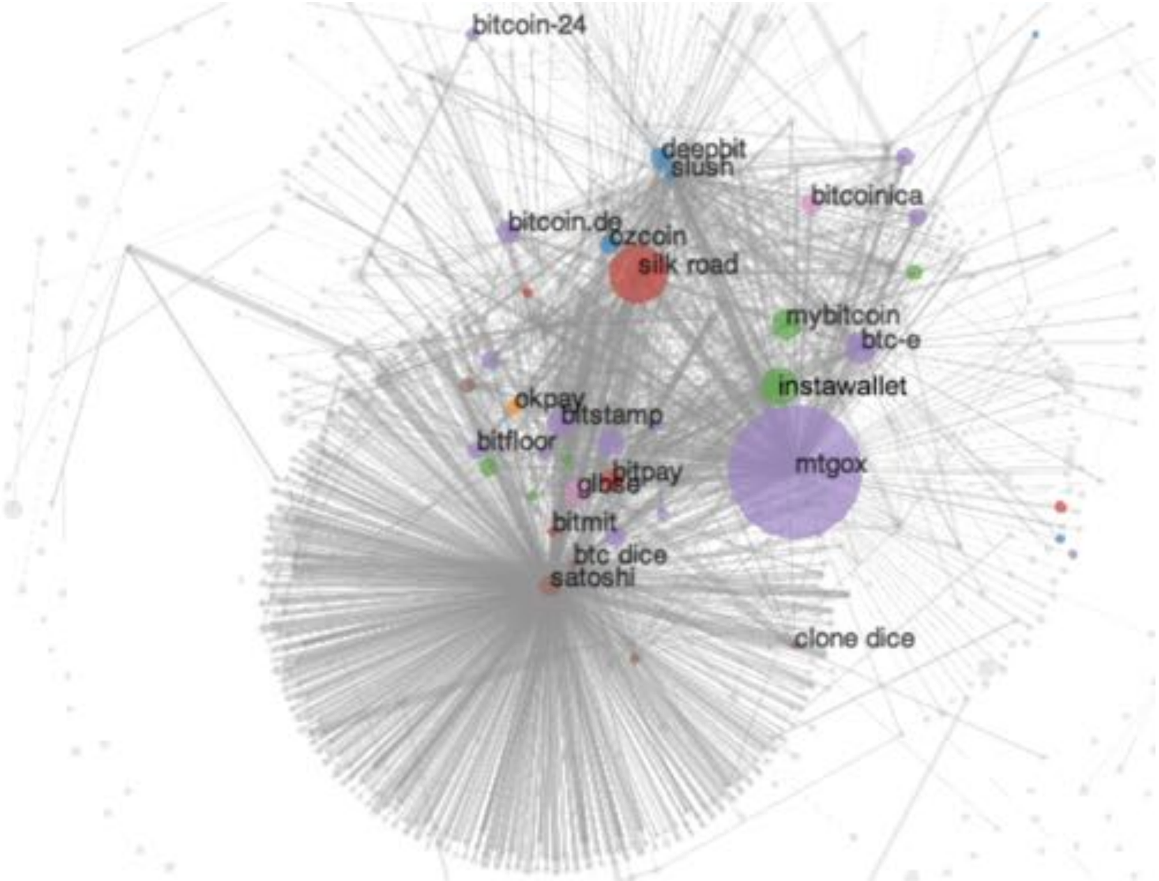
- Any such system must be **<u>fungible</u>**: Any unit of currency is interchangeable with any other unit of equivalent value



**Challenge: Decoupling currency from identity**

# Enter Cryptocurrencies

The problem of Linkability



# Key Problem

TX

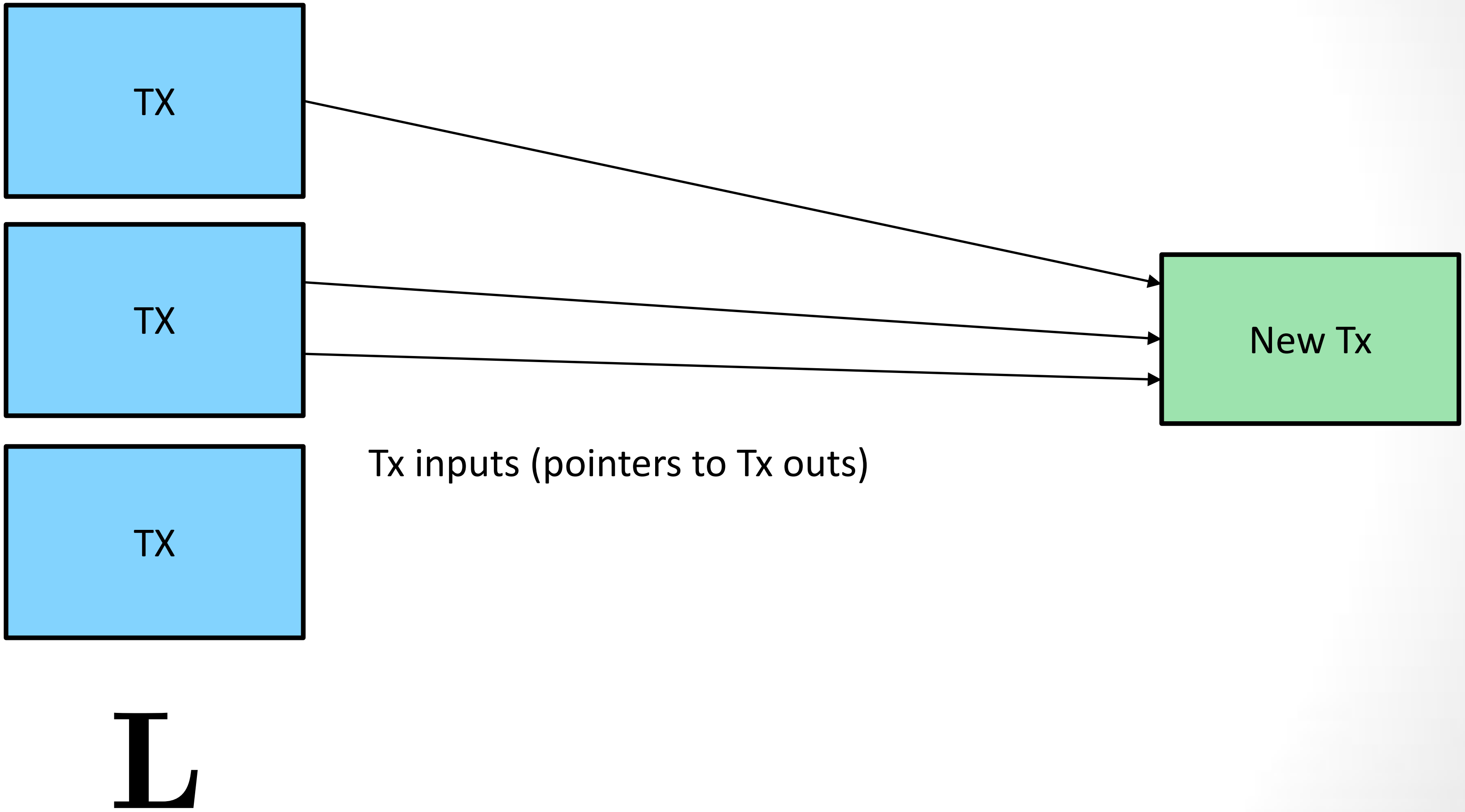
TX

TX

L

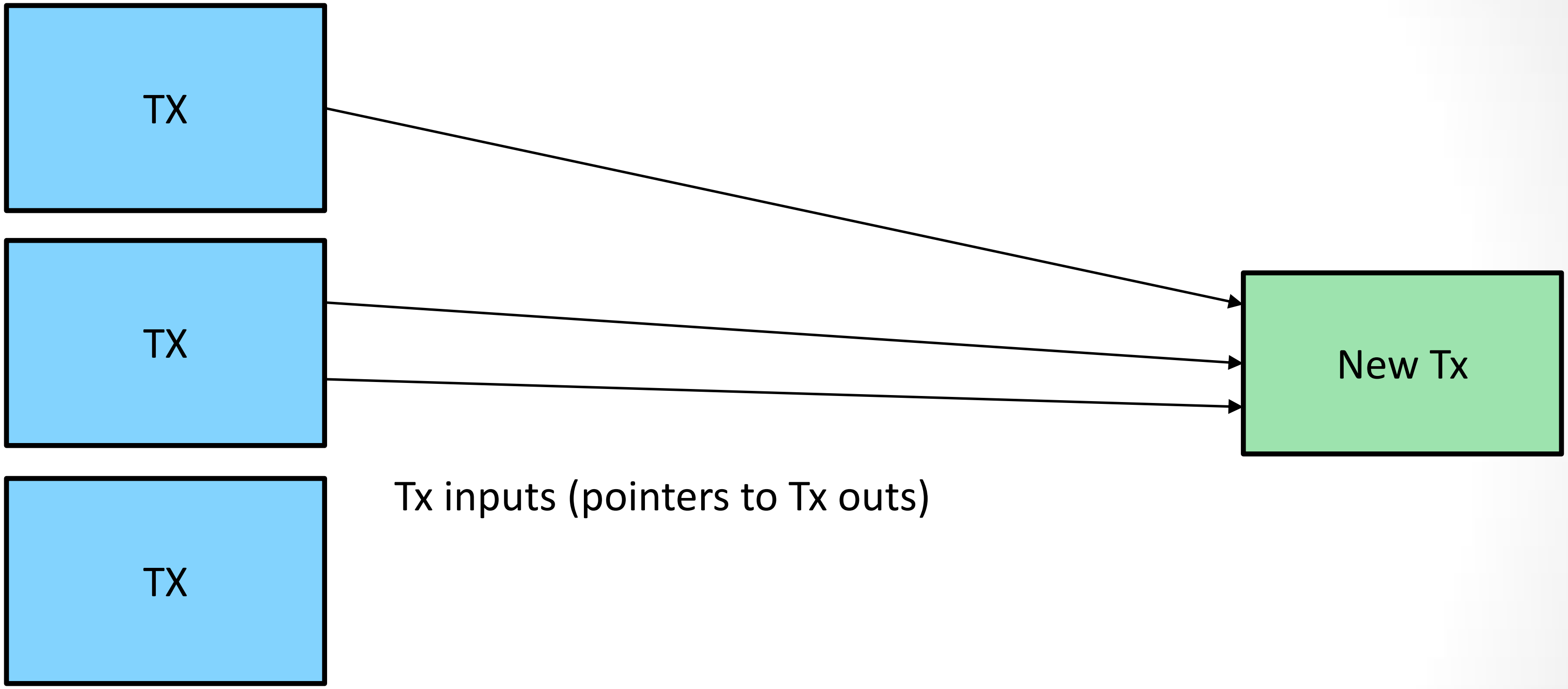
New Tx

# Key Problem



Tx inputs (pointers to Tx outs)

# Key Problem

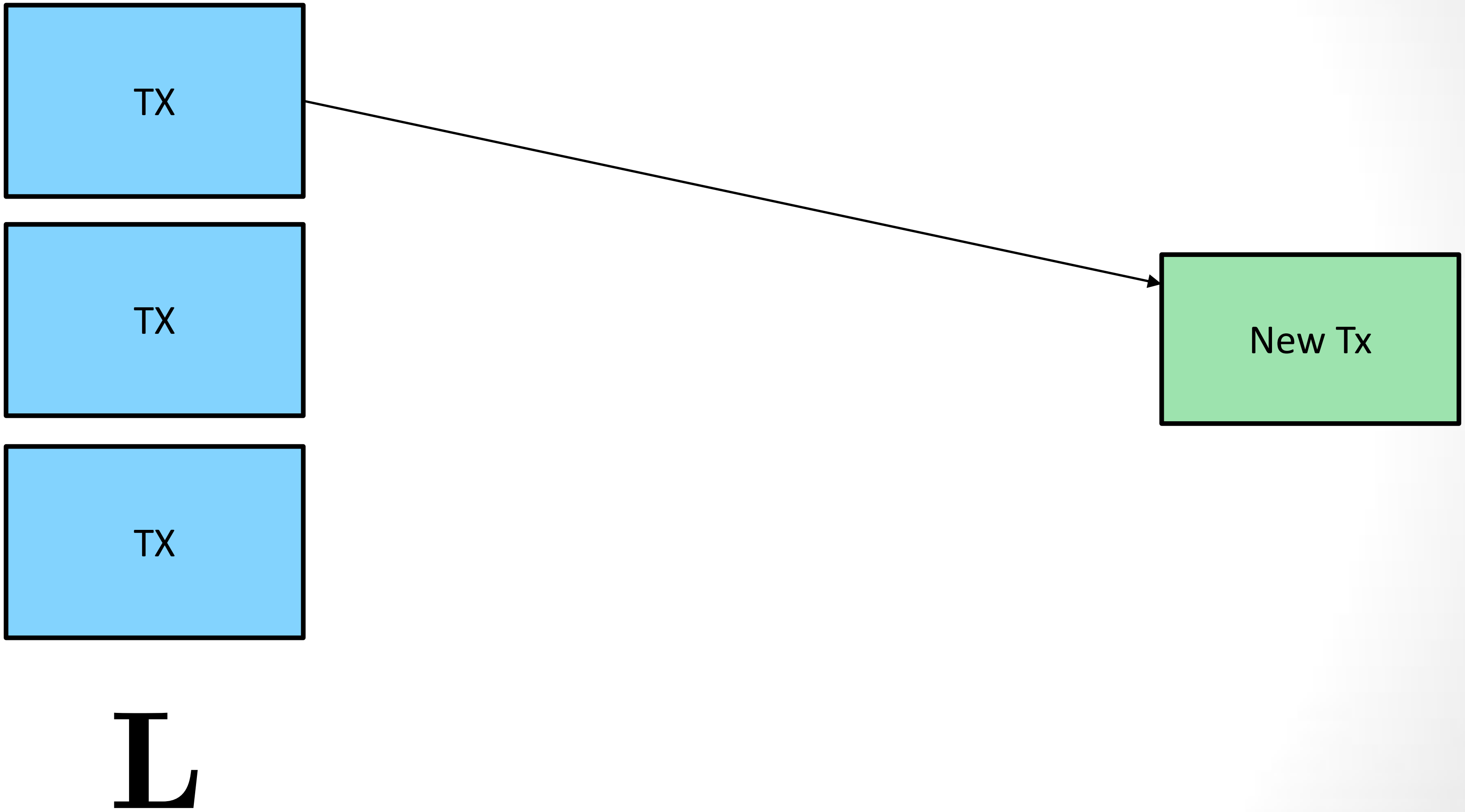


Tx inputs (pointers to Tx outs)

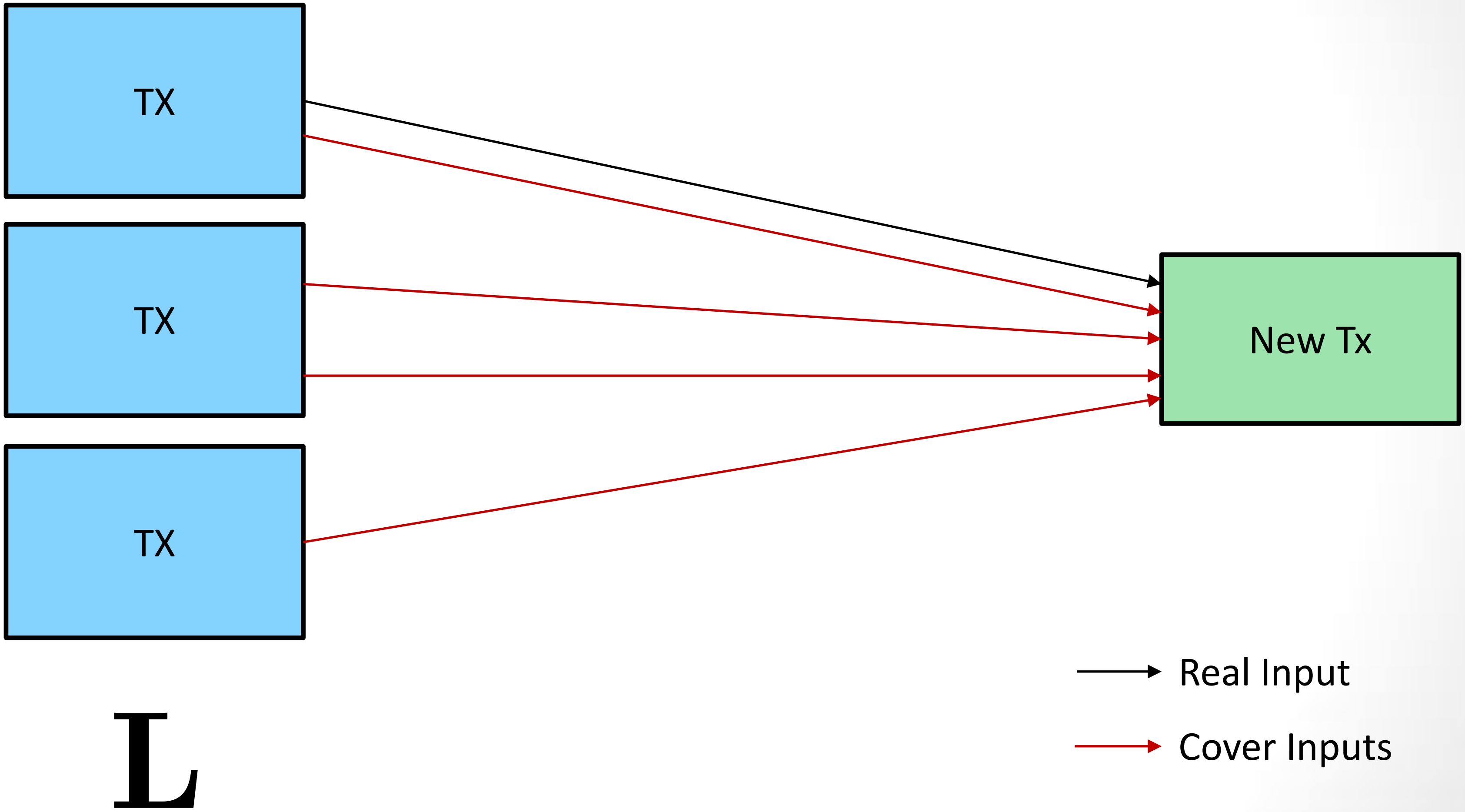
L

**Each Tx input has a publicly viewable history**

# Cryptographic Mixing



# Cryptographic Mixing



# What is the Crypto Magic

- Zerocoin and Zerocash:
    - Uses cryptographic accumulators and succinct proofs
    - Allows for Cover Set $\mathcal{T}$ to be all previous outputs
    - Relies on very strong cryptographic assumptions

# What is the Crypto Magic

- Zerocoin and Zerocash:
  - Uses cryptographic accumulators and succinct proofs
  - Allows for Cover Set $\mathcal{T}$ to be all previous outputs
  - Relies on very strong cryptographic assumptions

- CryptoNote and RingCT
  - Uses Ring Signatures
  - Each Transaction has a randomly sampled Cover Set $\mathcal{T}$
  - Amount of Anonymity depends on $|\mathcal{T}|$
  - Focus of this work

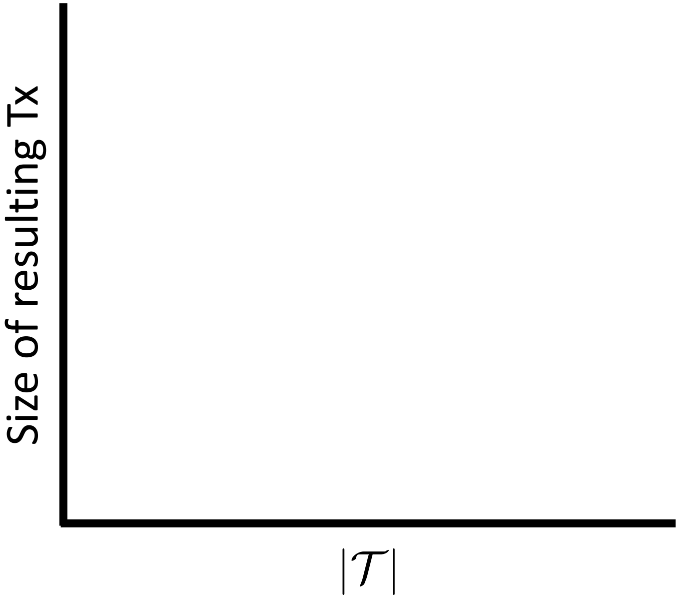
# What is the Crypto Magic

- Zerocoin and Zerocash:
  - Uses cryptographic accumulators and succinct proofs
  - Allows for Cover Set $\mathcal{T}$ to be all previous outputs
  - Relies on very strong cryptographic assumptions

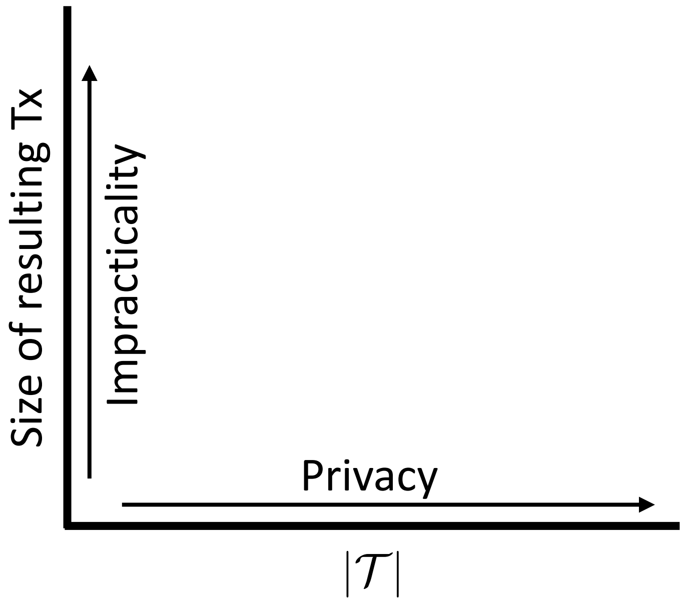
- CryptoNote and RingCT
  - Uses Ring Signatures
  - Each Transaction has a randomly sampled Cover Set $\mathcal{T}$
  - Amount of Anonymity depends on $|\mathcal{T}|$
  - Focus of this work

Note: Non-cryptographic mixing techniques exist but out of the scope of this work

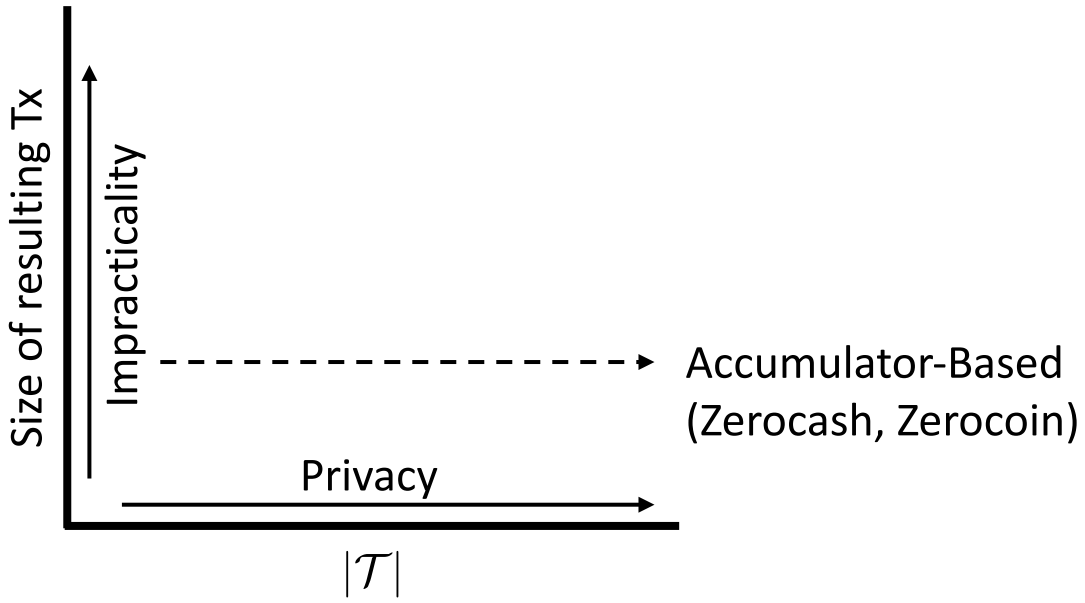
# Anonymity Tradeoff



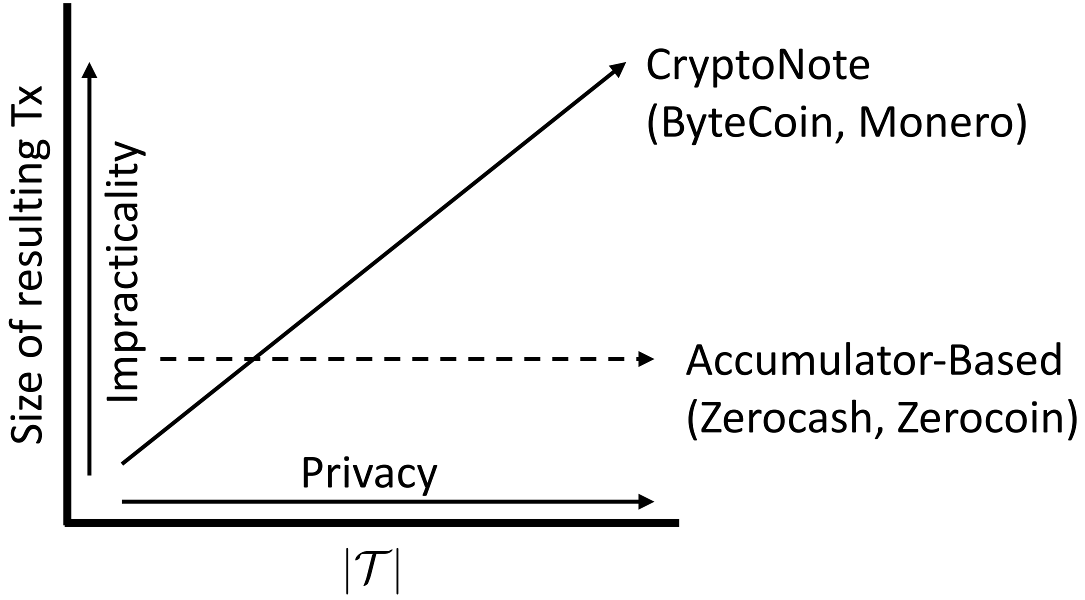
# Anonymity Tradeoff



# Anonymity Tradeoff



# Anonymity Tradeoff



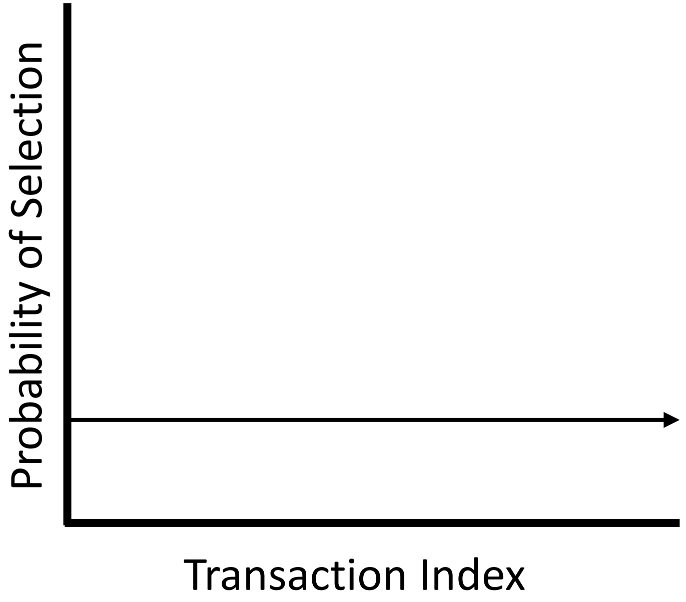
# Anonymity in ByteCoin/Monero

- Samples a Cover Set

- But how is this sampling performed?

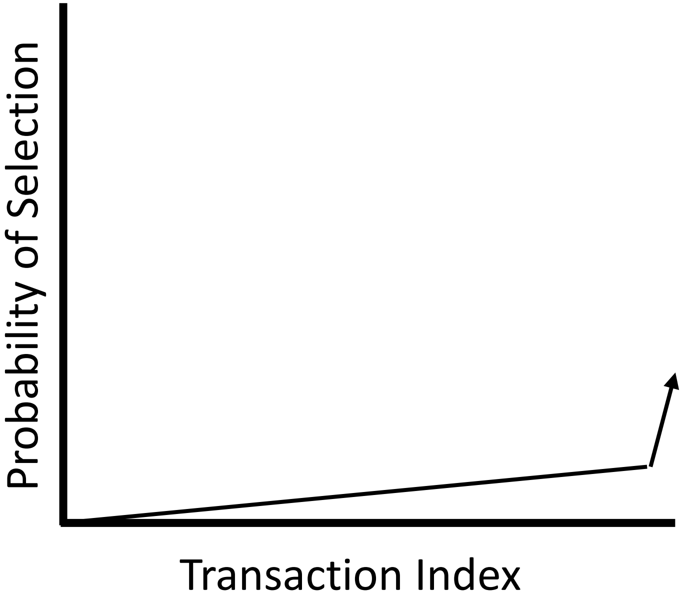
# Anonymity in ByteCoin/Monero

- Bytecoin:



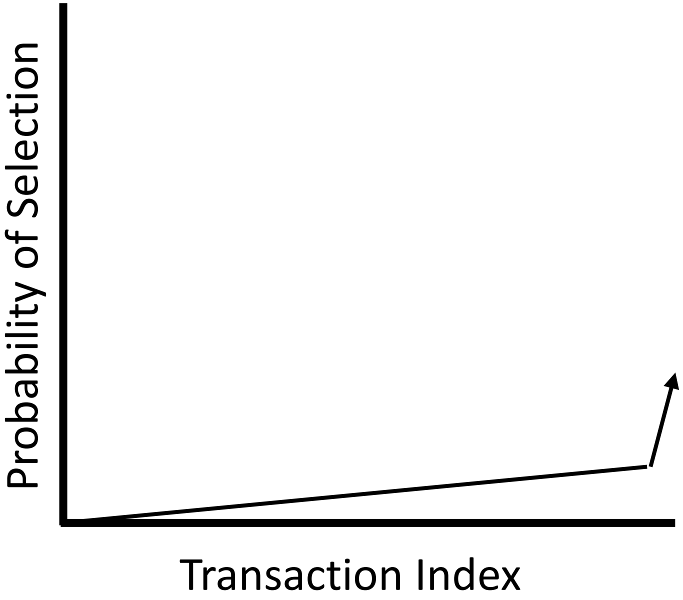
# Anonymity in ByteCoin/Monero

- Monero:



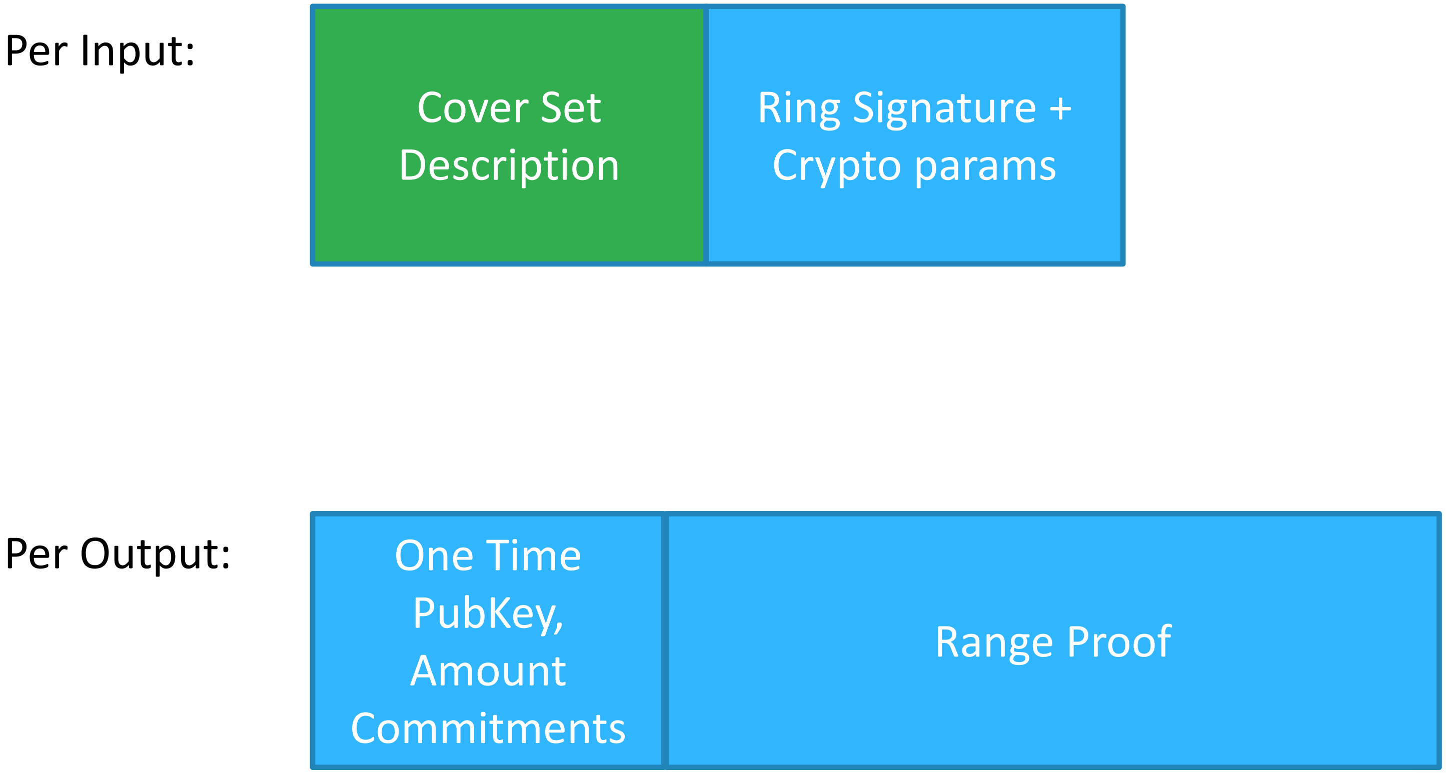
# Anonymity in ByteCoin/Monero

- Monero:



Bias toward more recent outputs

# Simplified Monero Transaction



Per Input:

Cover Set Description

Ring Signature + Crypto params

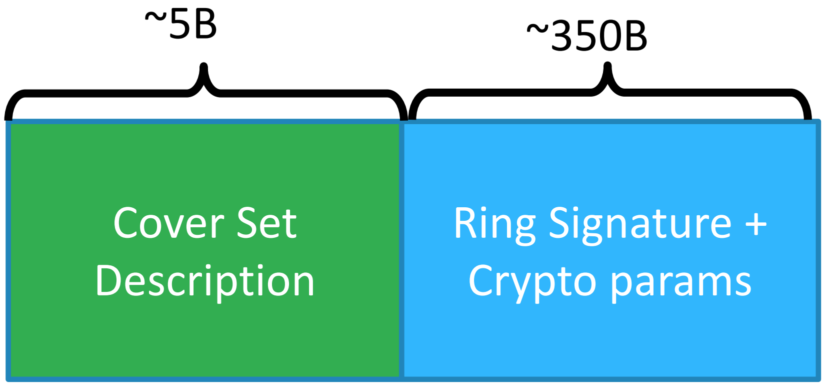
Per Output:

One Time PubKey, Amount Commitments

Range Proof

# Simplified Monero Transaction

Per Input:



~5B

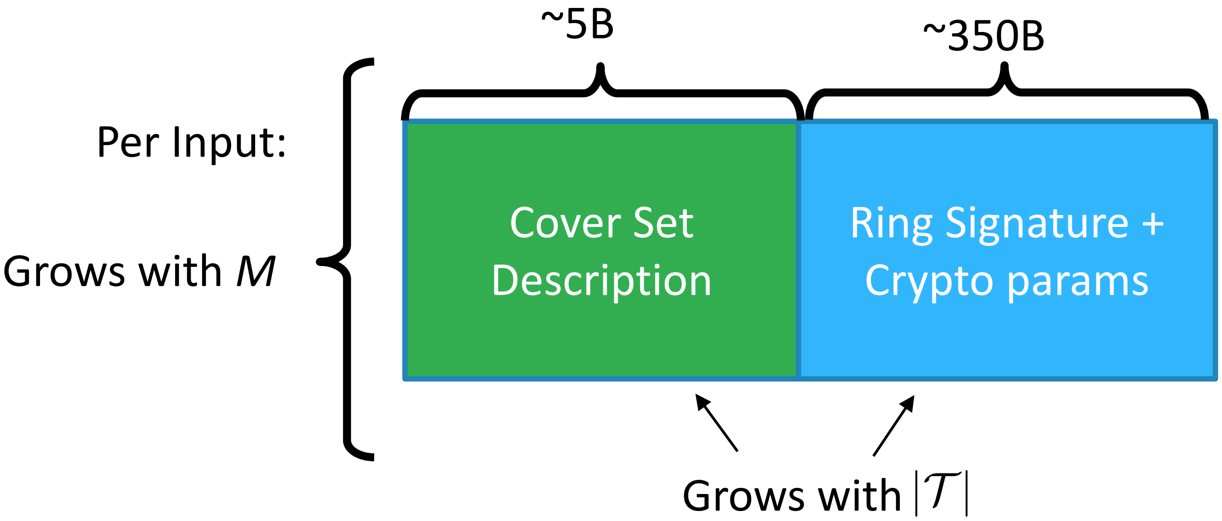
~350B

Cover Set Description

Ring Signature + Crypto params

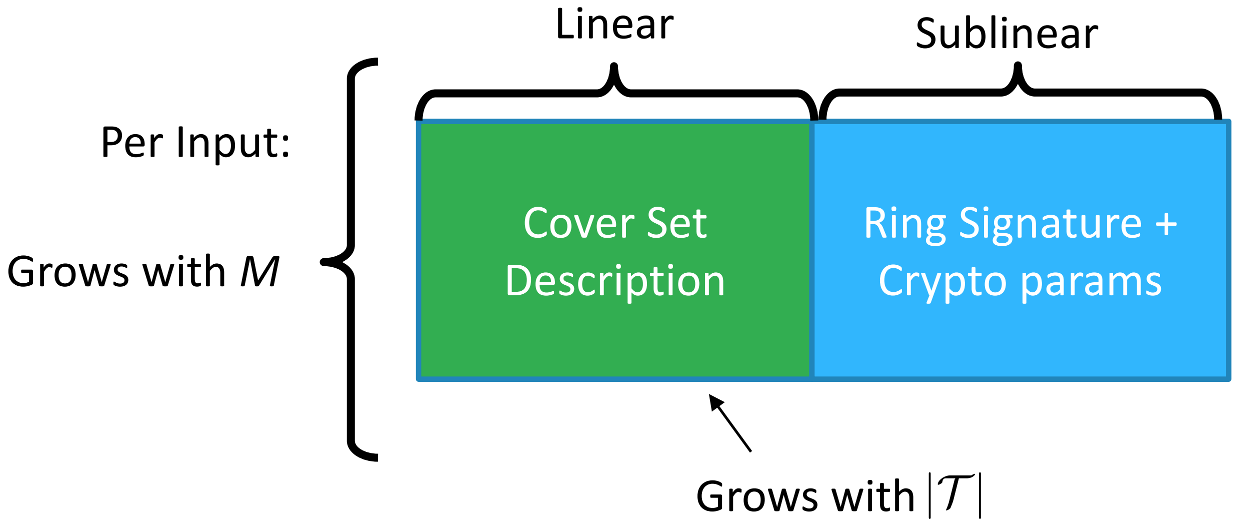
For a Cover Set with a size of 5

# Simplified Monero Transaction

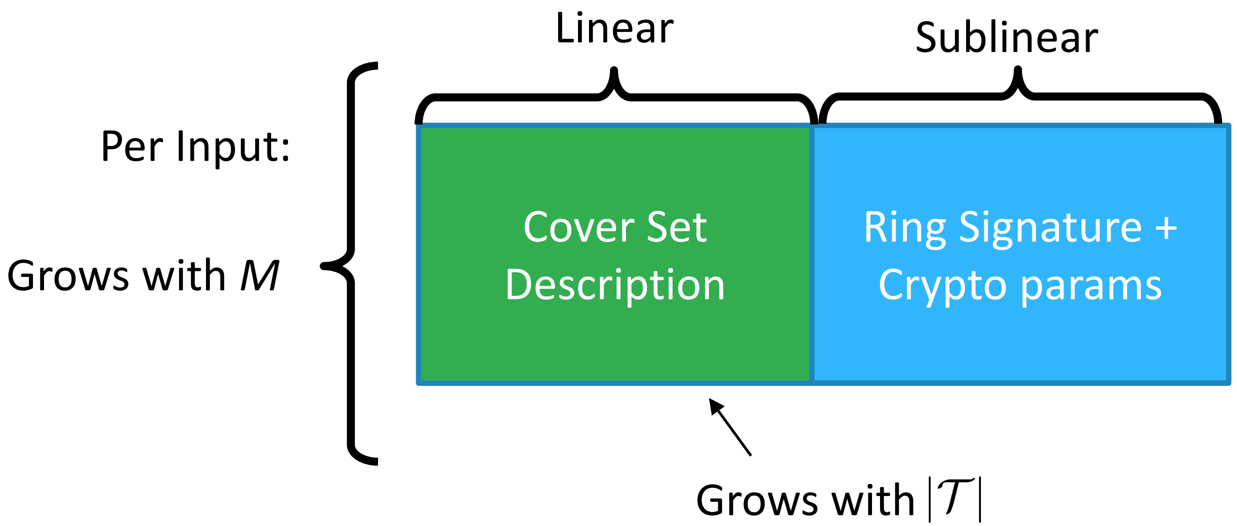


For a Cover Set with a size of 5

# Future Monero Transaction



# Future Monero Transaction



Linear

Sublinear

Per Input:

Grows with *$M$*

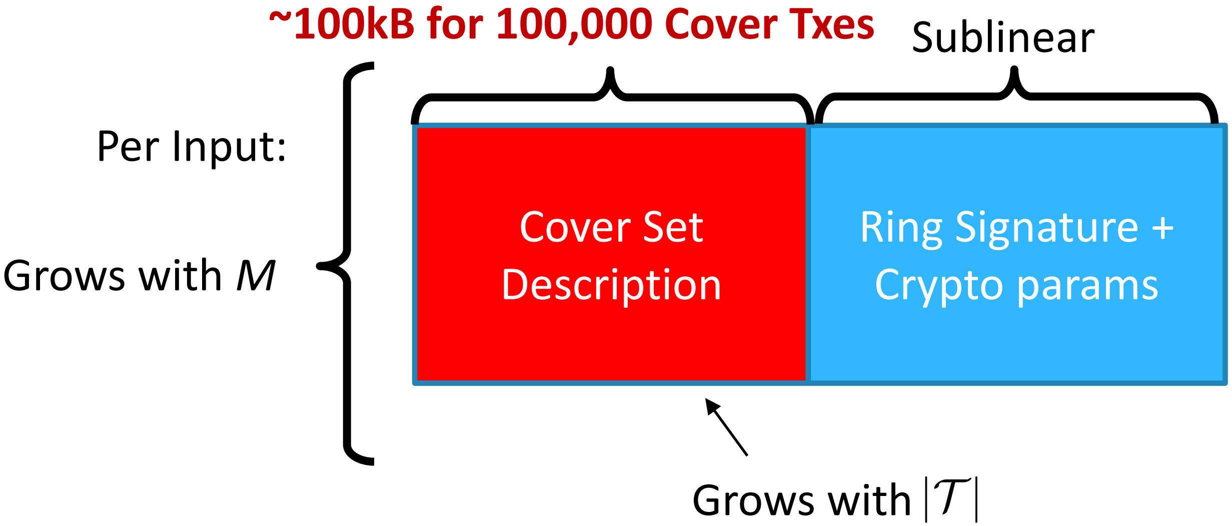
Cover Set Description

Ring Signature + Crypto params

Grows with $|\mathcal{T}|$

**In theory, supports much higher levels of privacy**

# Future Monero Transaction


~100kB for 100,000 Cover Txes
Sublinear
Per Input:
Grows with $M$
Cover Set Description
Ring Signature + Crypto params
Grows with $|\mathcal{T}|$

**In theory, supports much higher levels of privacy**

# Basic Sampling Strategy

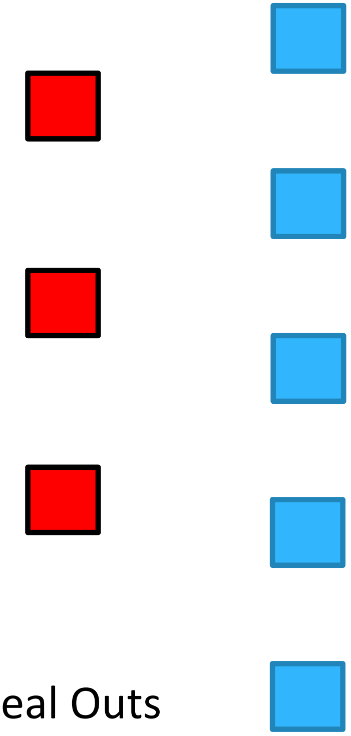
🟥

🟥

🟥

Real Outs

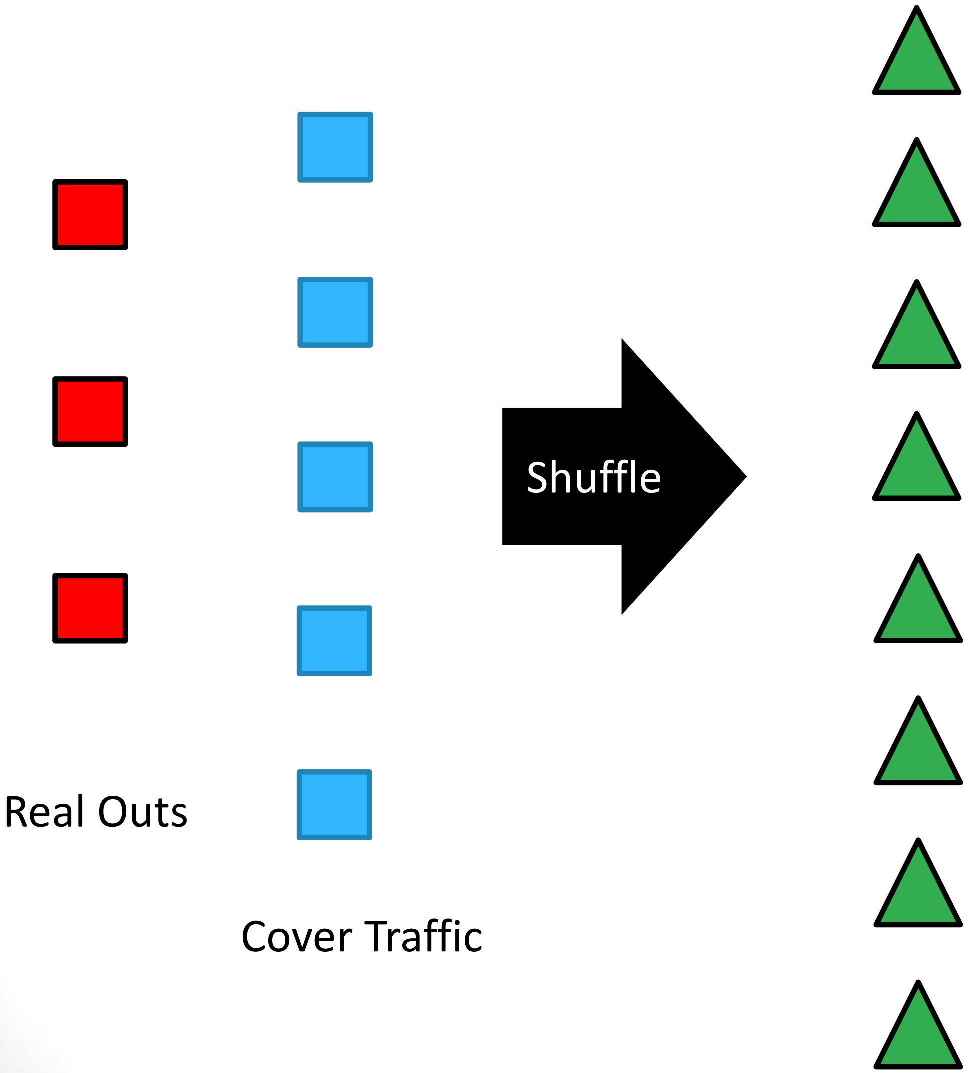
# Basic Sampling Strategy



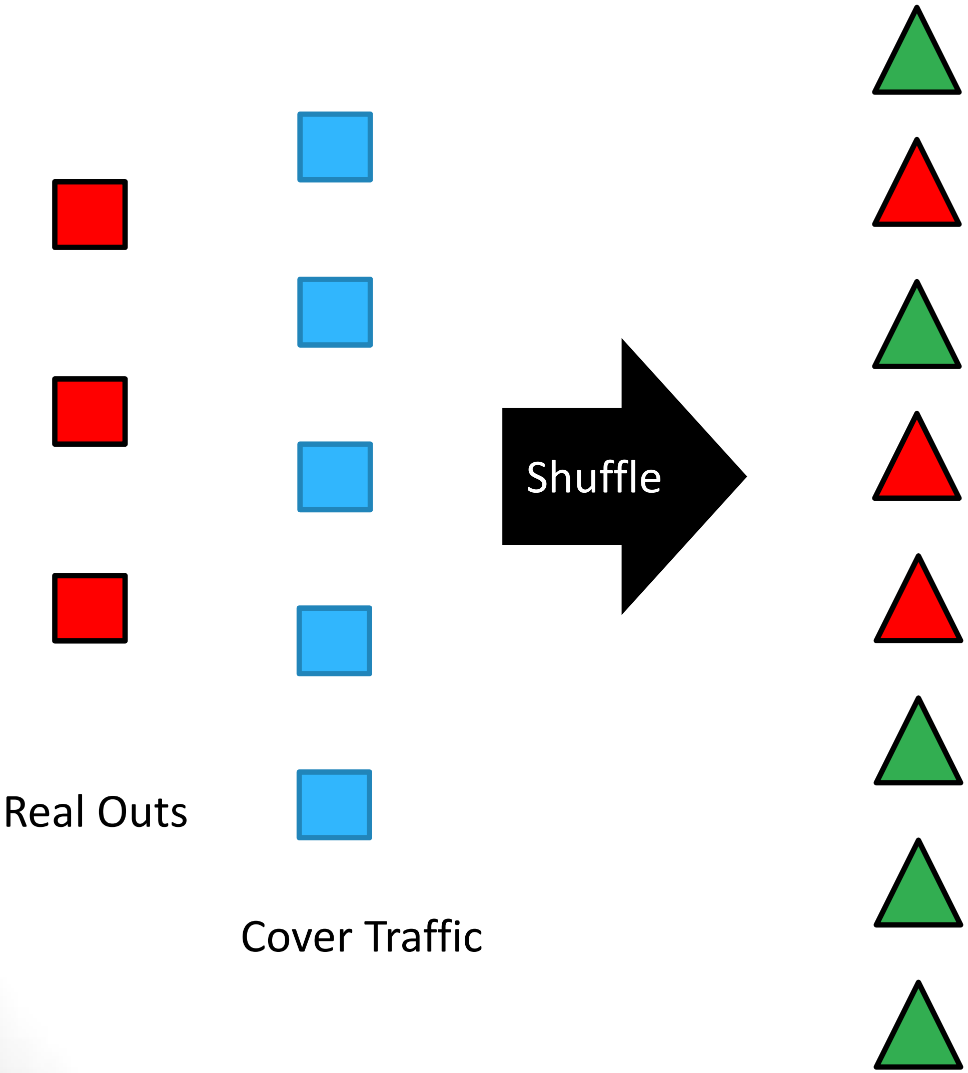
Real Outs

Cover Traffic

# Basic Sampling Strategy

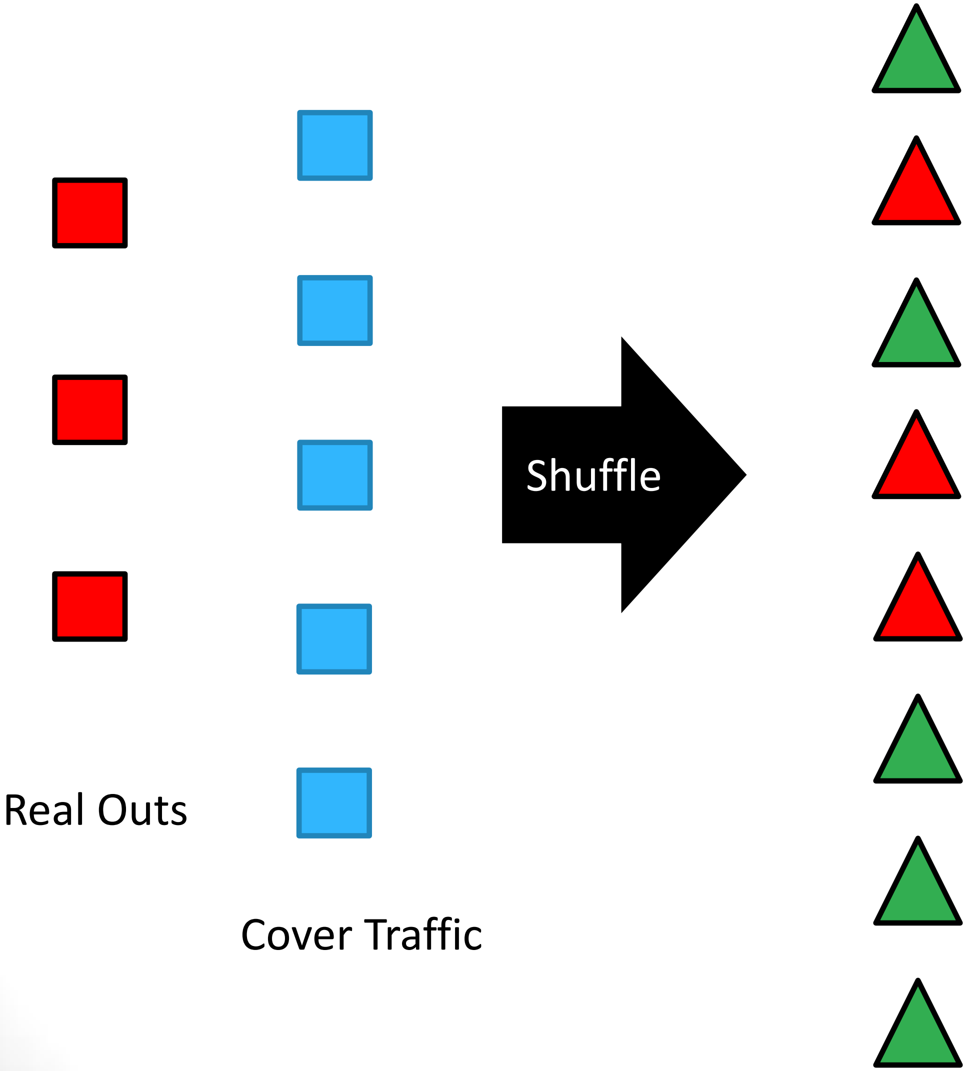


# Basic Sampling Strategy



Real outputs may be obviously different from cover traffic

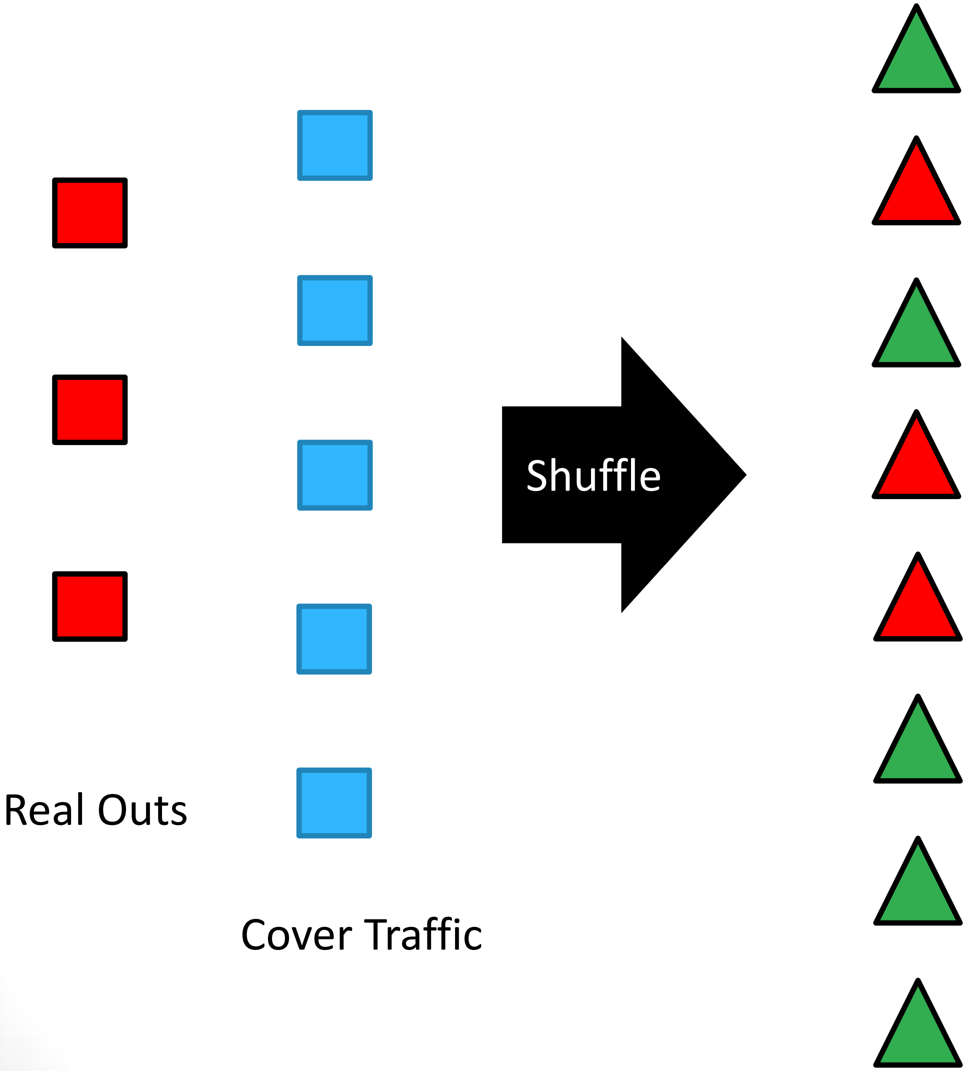
# Basic Sampling Strategy



Real outputs may be obviously different from cover traffic

**Outside scope of this work**

# Basic Sampling Strategy



Cover traffic is still randomly distributed in this scenario

# The Recoverable Sampling Scheme

# The Recoverable Sampling Scheme

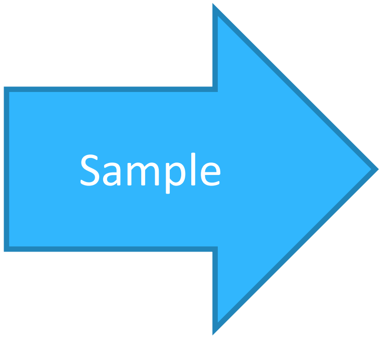
Sample:

$I_1$
$I_2$
$\dots$
$I_{M-1}$
$I_M$

# The Recoverable Sampling Scheme

Sample:

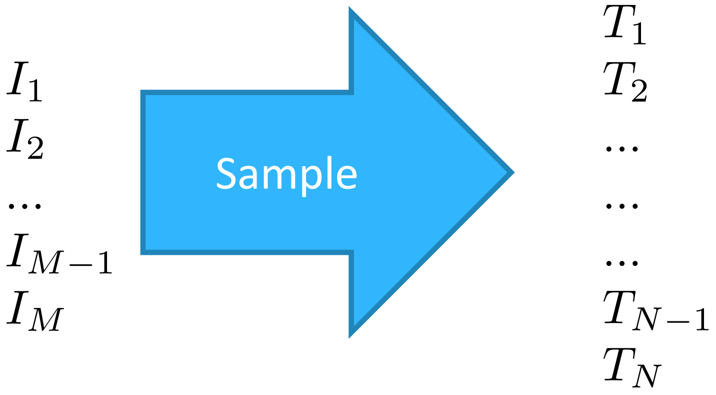
$I_1$
$I_2$
$\ldots$
$I_{M-1}$
$I_M$



Sample

# The Recoverable Sampling Scheme

Sample:



$I_1$

$I_2$

$\dots$

$I_{M-1}$

$I_M$

$T_1$

$T_2$

$\dots$

$\dots$

$\dots$

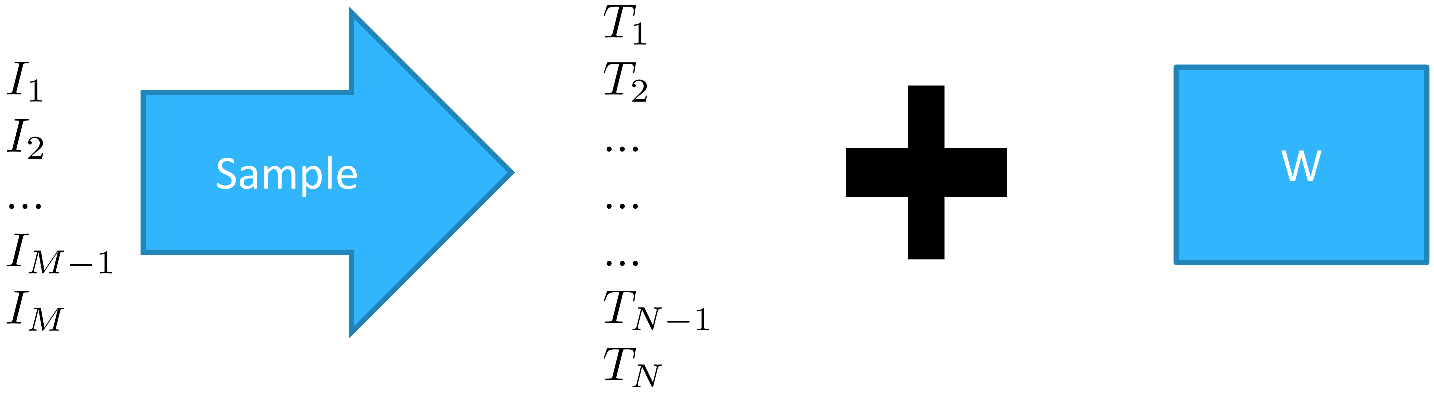
$T_{N-1}$

$T_N$

Grows Linearly with *N*

# The Recoverable Sampling Scheme

Sample:



$I_1$
$I_2$
$\ldots$
$I_{M-1}$
$I_M$

$T_1$
$T_2$
$\ldots$
$\ldots$
$\ldots$
$T_{N-1}$
$T_N$

Grows Linearly with *N*

Grows Sublinearly with *N*

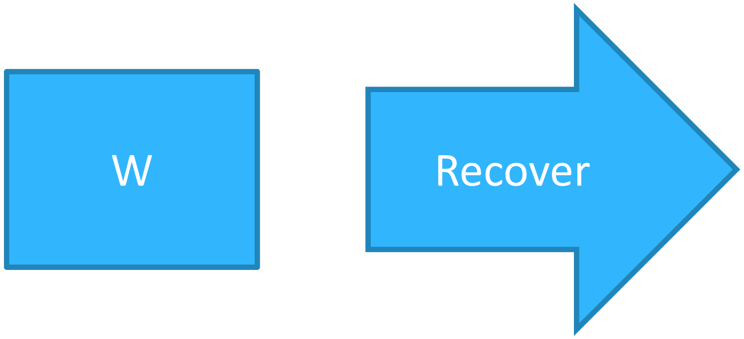
# The Recoverable Sampling Scheme

Recover:

w

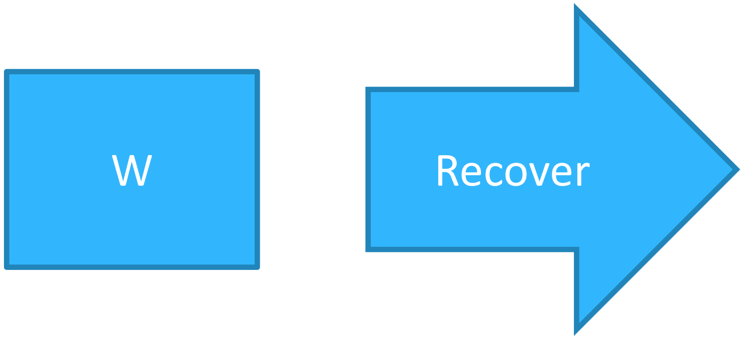
# The Recoverable Sampling Scheme

Recover:



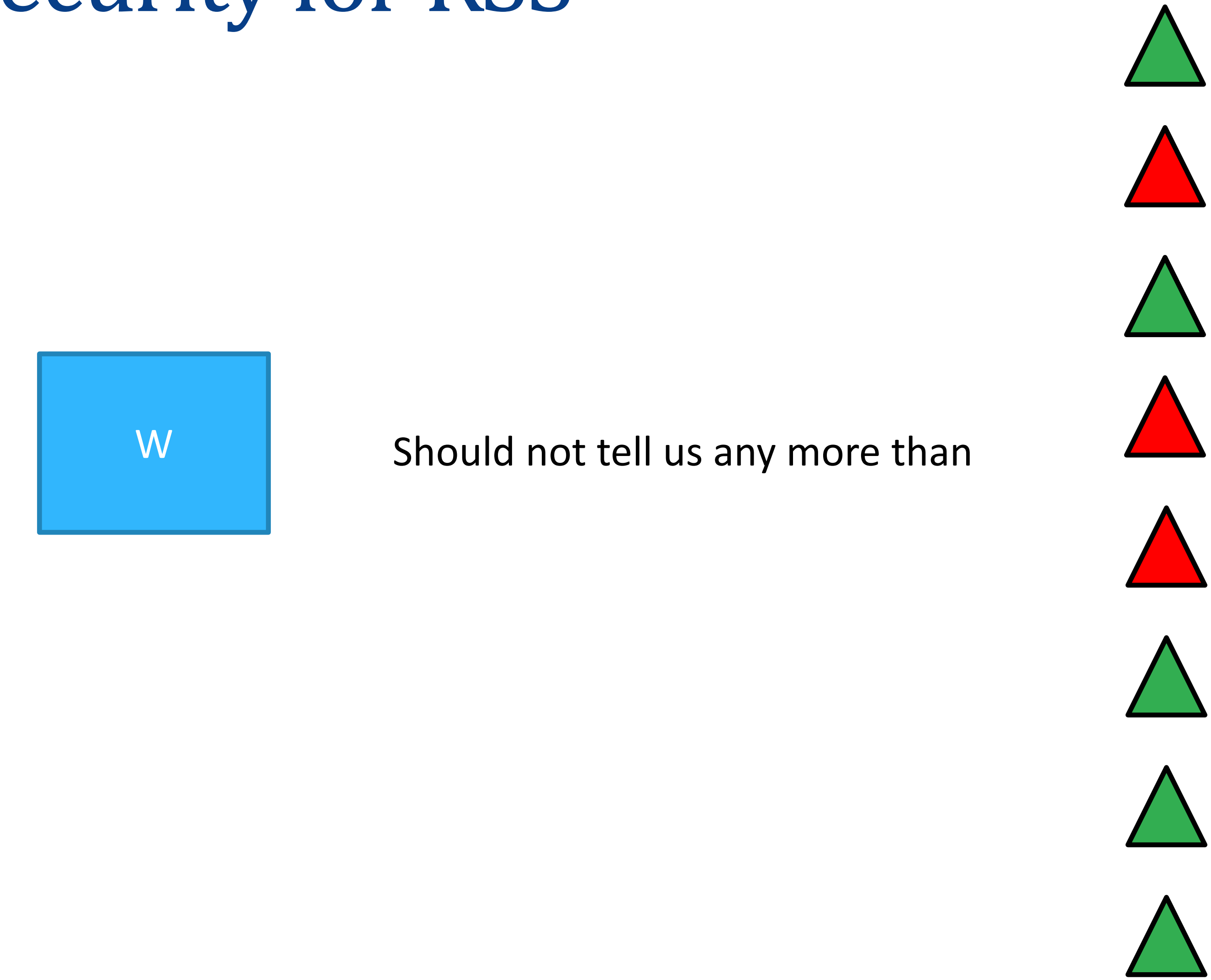
# The Recoverable Sampling Scheme

Recover:



$T_1$
$T_2$
$\ldots$
$\ldots$
$\ldots$
$T_{N-1}$
$T_N$

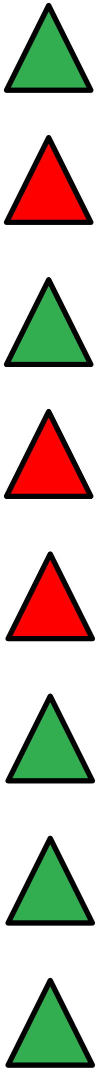
# Security for RSS



Should not tell us any more than

# Security for RSS

Should not tell us any more than



**What does W look like in practice?**

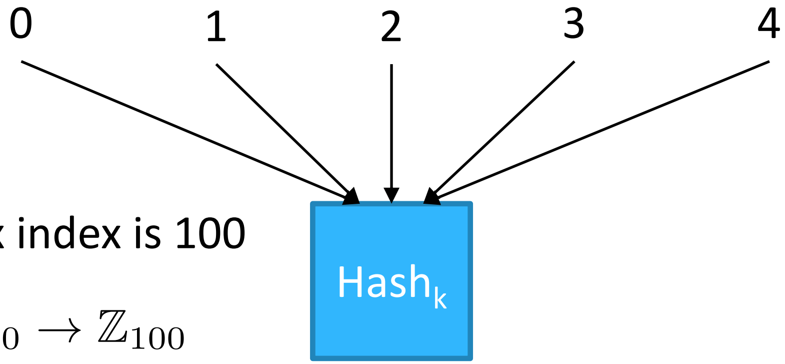
# Naïve RSS

Uniform Sampling and *M*=1

0 1 2 3 4

# Naïve RSS

Uniform Sampling and *M*=1



Assume max index is 100

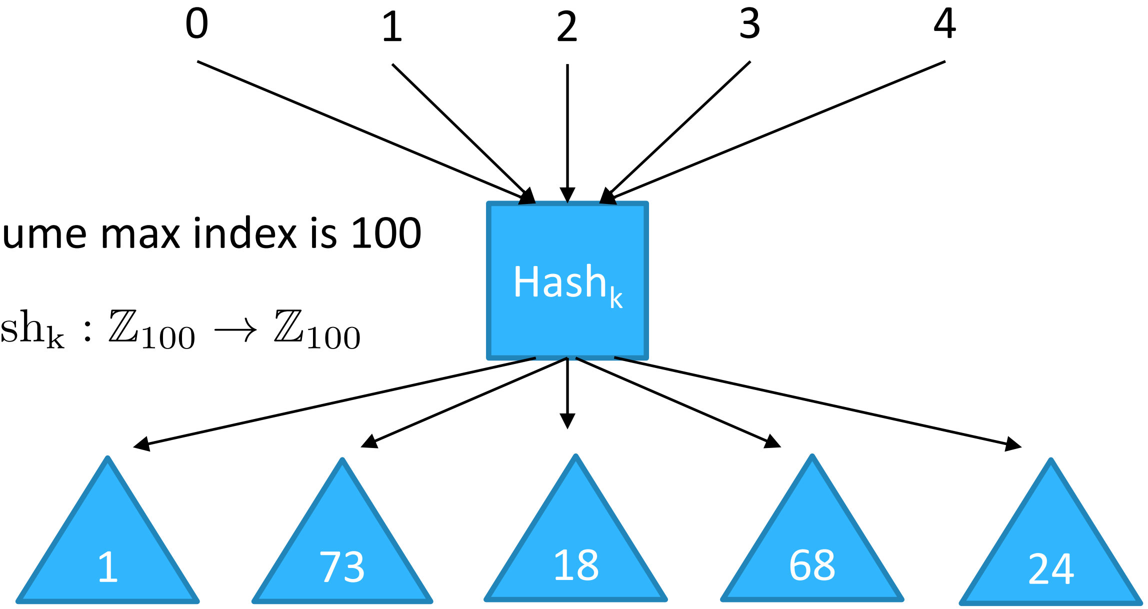
$\mathrm{Hash}_\mathrm{k} : \mathbb{Z}_{100} \rightarrow \mathbb{Z}_{100}$

# Naïve RSS

Uniform Sampling and *M*=1

Assume max index is 100

$\mathrm{Hash}_\mathrm{k} : \mathbb{Z}_{100} \to \mathbb{Z}_{100}$

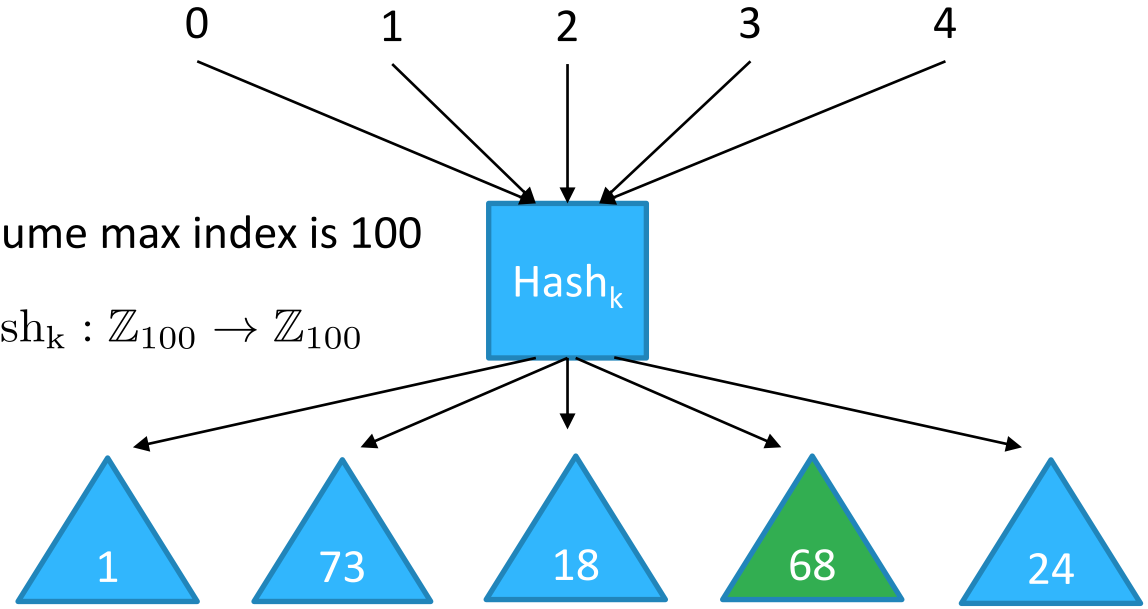


# Naïve RSS

Uniform Sampling and *M*=1

Assume max index is 100

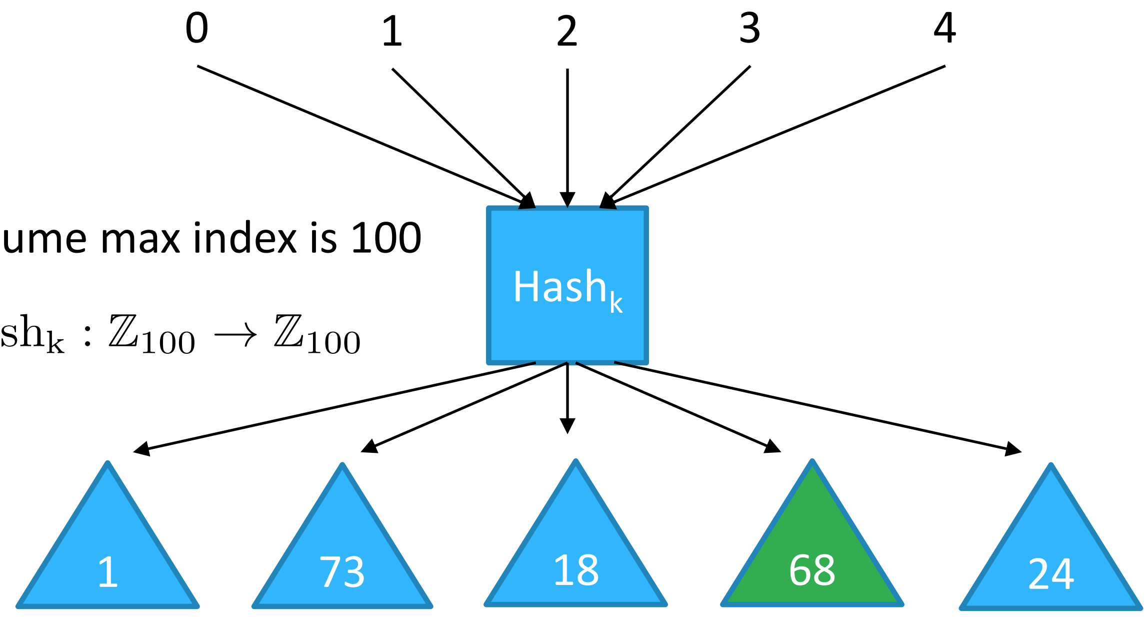
$\mathrm{Hash}_\mathrm{k} : \mathbb{Z}_{100} \rightarrow \mathbb{Z}_{100}$



Want this to be 80

# Naïve RSS

Uniform Sampling and *M*=1



Assume max index is 100

$\mathrm{Hash}_\mathrm{k} : \mathbb{Z}_{100} \rightarrow \mathbb{Z}_{100}$

Want this to be 80

**Idea: use modular addition**

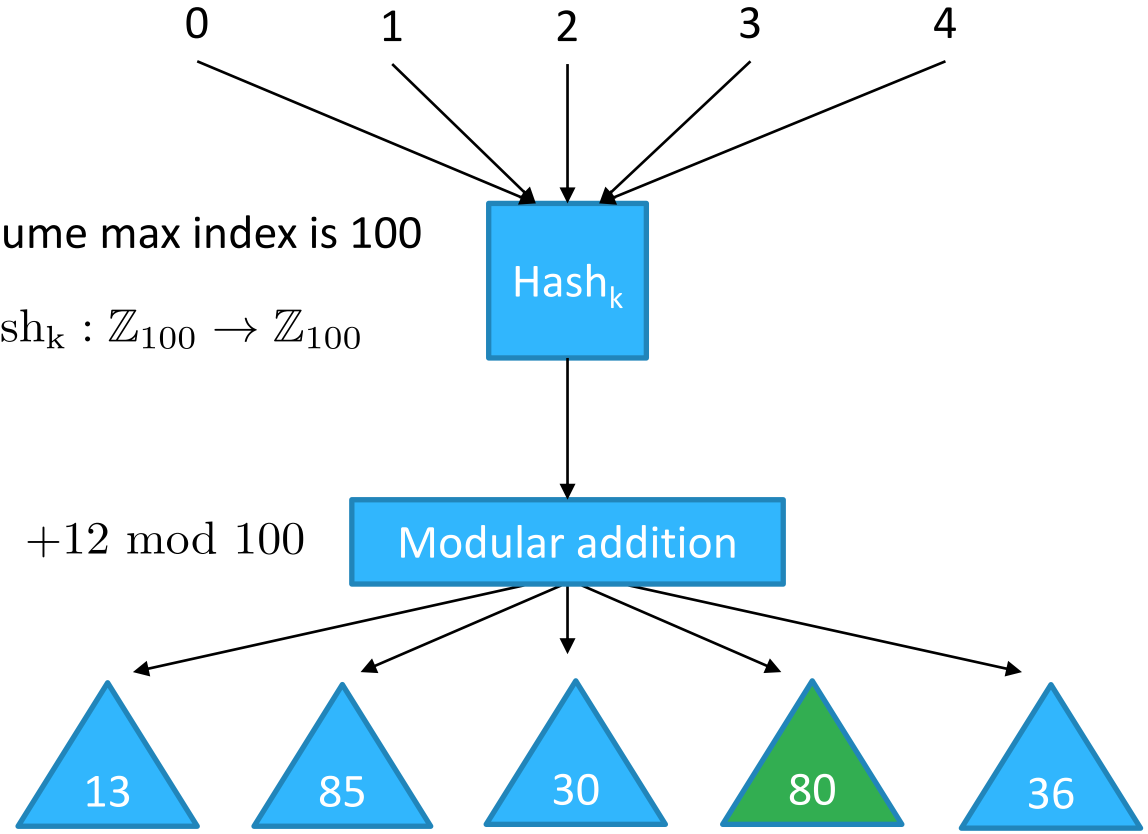
# Naïve RSS

Uniform Sampling and *M*=1

Assume max index is 100

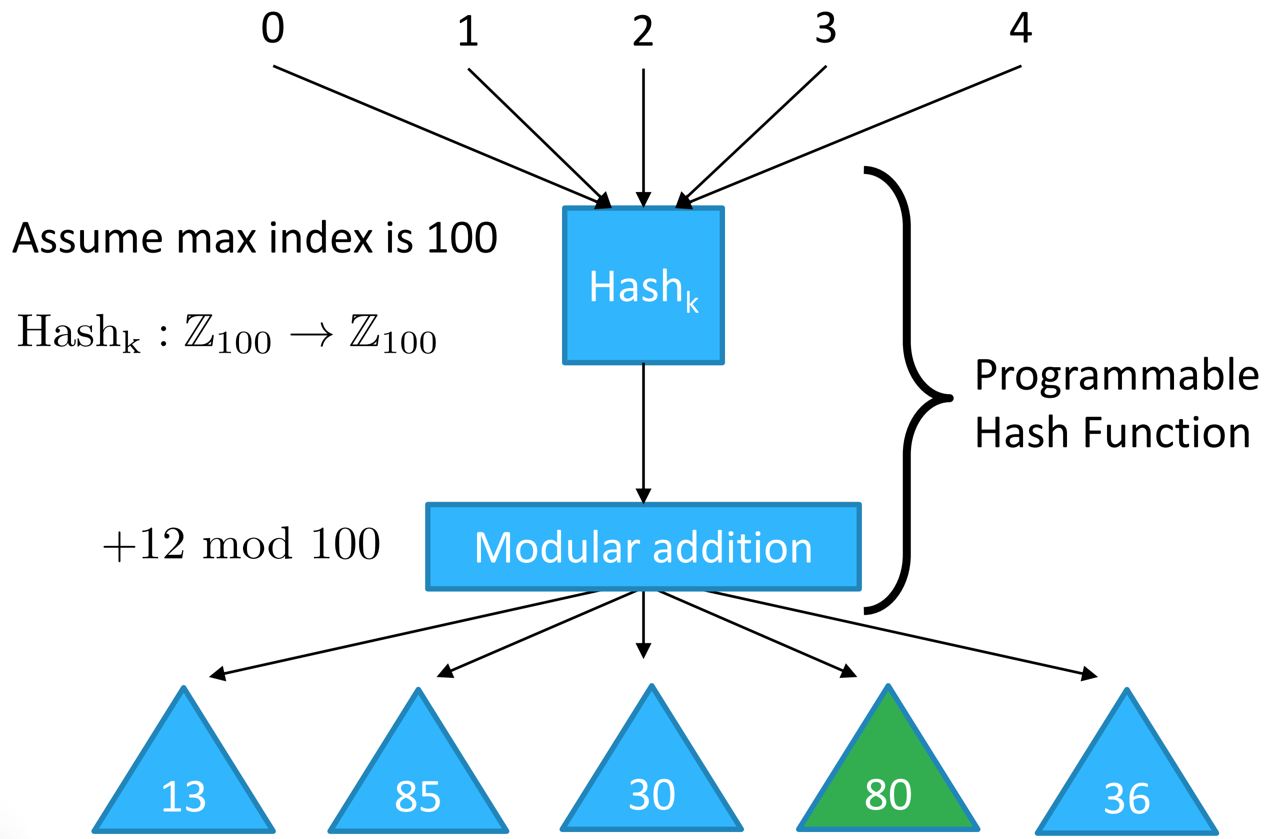
$\mathrm{Hash_k} : \mathbb{Z}_{100} \to \mathbb{Z}_{100}$

$+12 \bmod 100$



# Naïve RSS

Uniform Sampling and *M*=1



Assume max index is 100

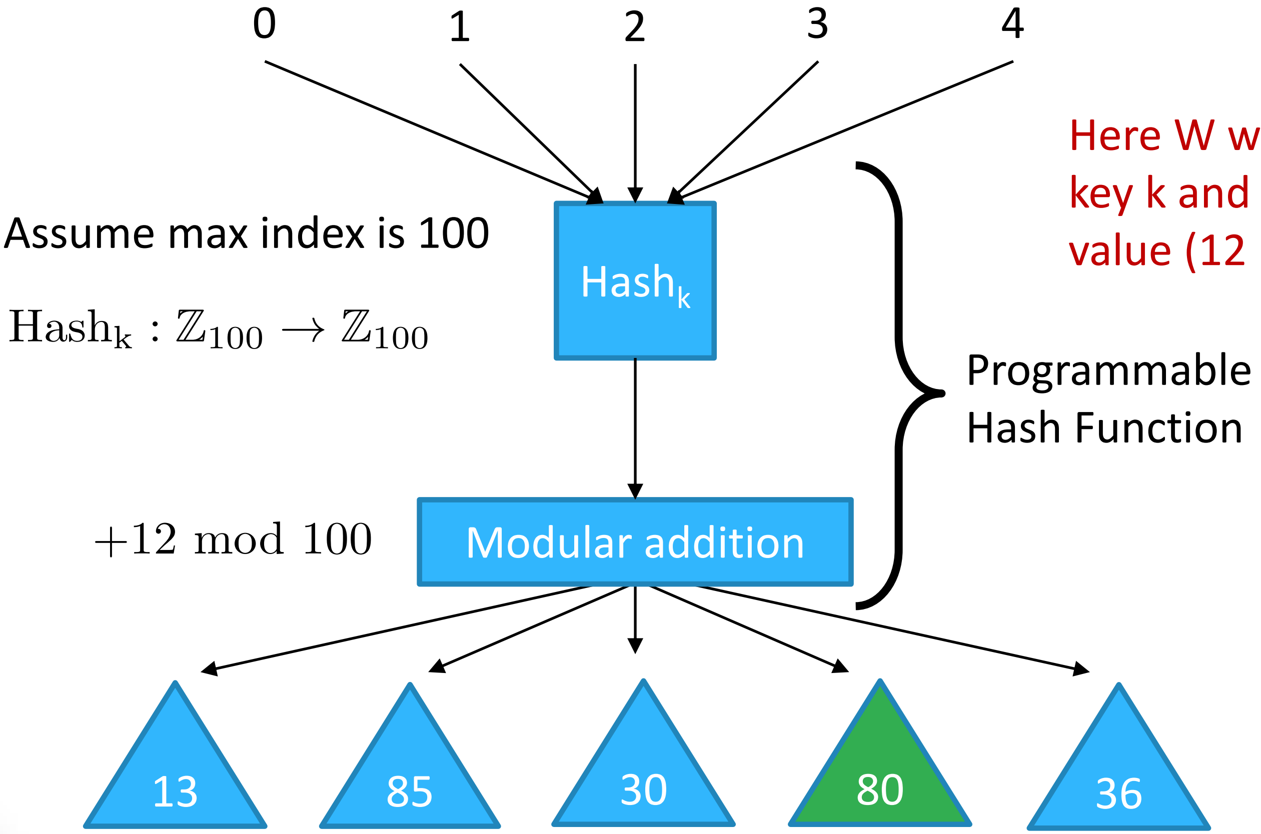
$\mathrm{Hash}_\mathrm{k} : \mathbb{Z}_{100} \to \mathbb{Z}_{100}$

$+12 \bmod 100$

Programmable Hash Function

# Naïve RSS

Uniform Sampling and *M*=1



Assume max index is 100

$$\text{Hash}_k : \mathbb{Z}_{100} \to \mathbb{Z}_{100}$$

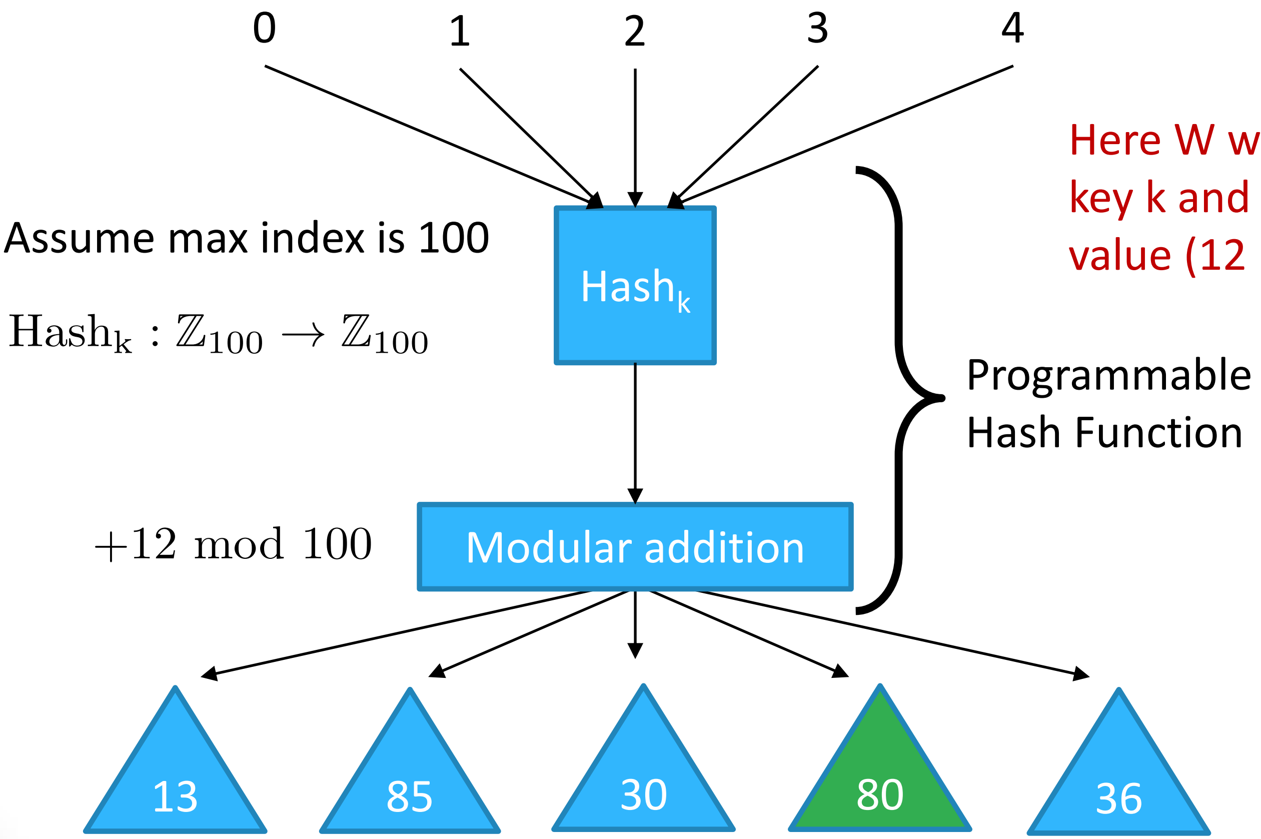
$$+12 \bmod 100$$

Programmable Hash Function

Here W would simply be the Hash key k and the modular addition value (12 in this example)

# Naïve RSS

Uniform Sampling and *M*=1



Assume max index is 100

$\mathrm{Hash_k} : \mathbb{Z}_{100} \to \mathbb{Z}_{100}$

$+12 \bmod 100$

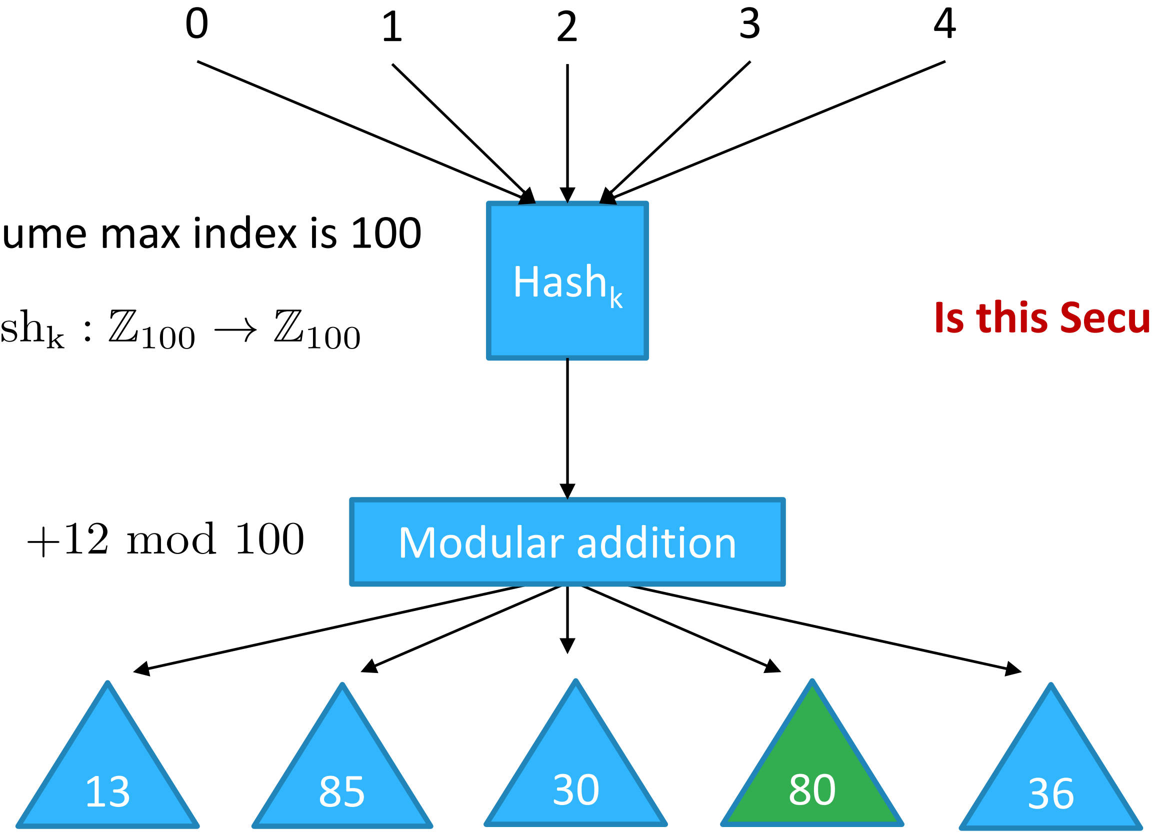
Programmable Hash Function

Here W would simply be the Hash key k and the modular addition value (12 in this example)

Constant Size!

# Naïve RSS

Uniform Sampling and *M*=1



Assume max index is 100

$\mathrm{Hash}_{\mathrm{k}} : \mathbb{Z}_{100} \to \mathbb{Z}_{100}$

$+12 \bmod 100$

**Is this Secure?**

# Duplicate Handling

- This process may introduce duplicate outputs in the Cover Set

- Unlikely to occur for reasonably large Cover Sets

- Can be further handled by resampling or oversampling

# Towards Generalized RSS

How do we support M > 1

# Towards Generalized RSS

How do we support M > 1

- Currently Monero and Bytecoin support 1 real in N Cover transactions

# Towards Generalized RSS

How do we support M > 1

- Currently Monero and Bytecoin support 1 real in N Cover transactions
  - resample for each input

# Towards Generalized RSS

How do we support M > 1

- Currently Monero and Bytecoin support 1 real in N Cover transactions
  - resample for each input
    - Inefficient

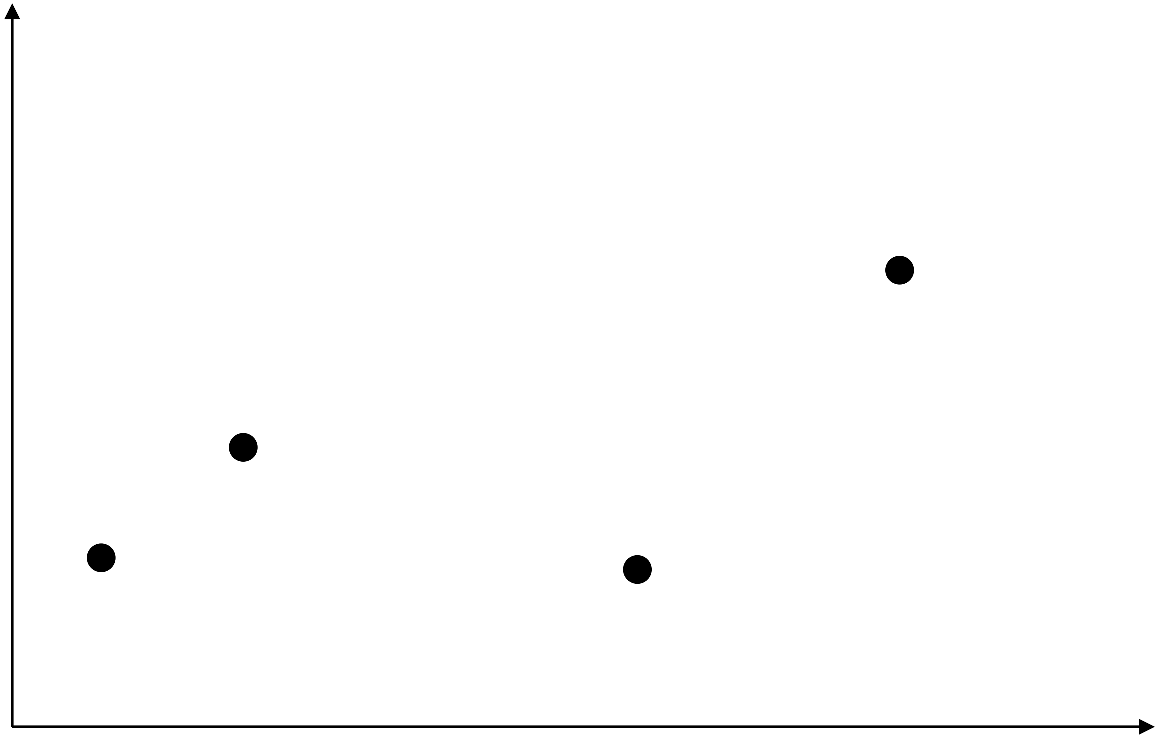
# Towards Generalized RSS

How do we support M > 1

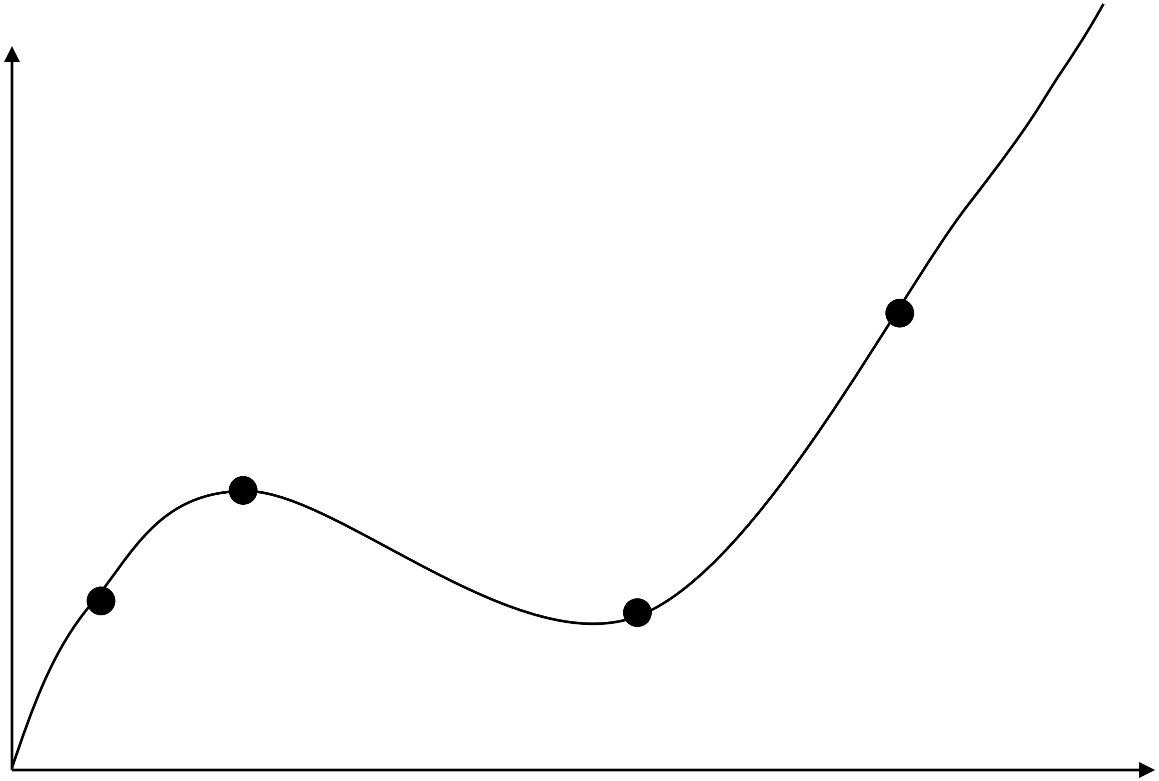
- Currently Monero and Bytecoin support 1 real in *N* Cover transactions
  - resample for each input
    - Inefficient

**Problem: how do we support *M* real out of *N* Cover Transactions**

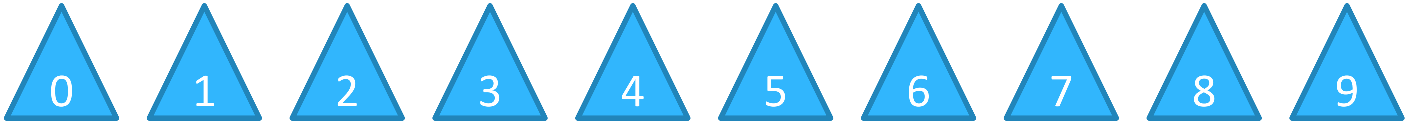
# Polynomial Interpolation



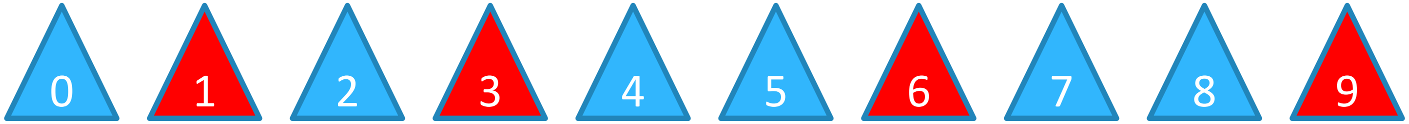
# Polynomial Interpolation



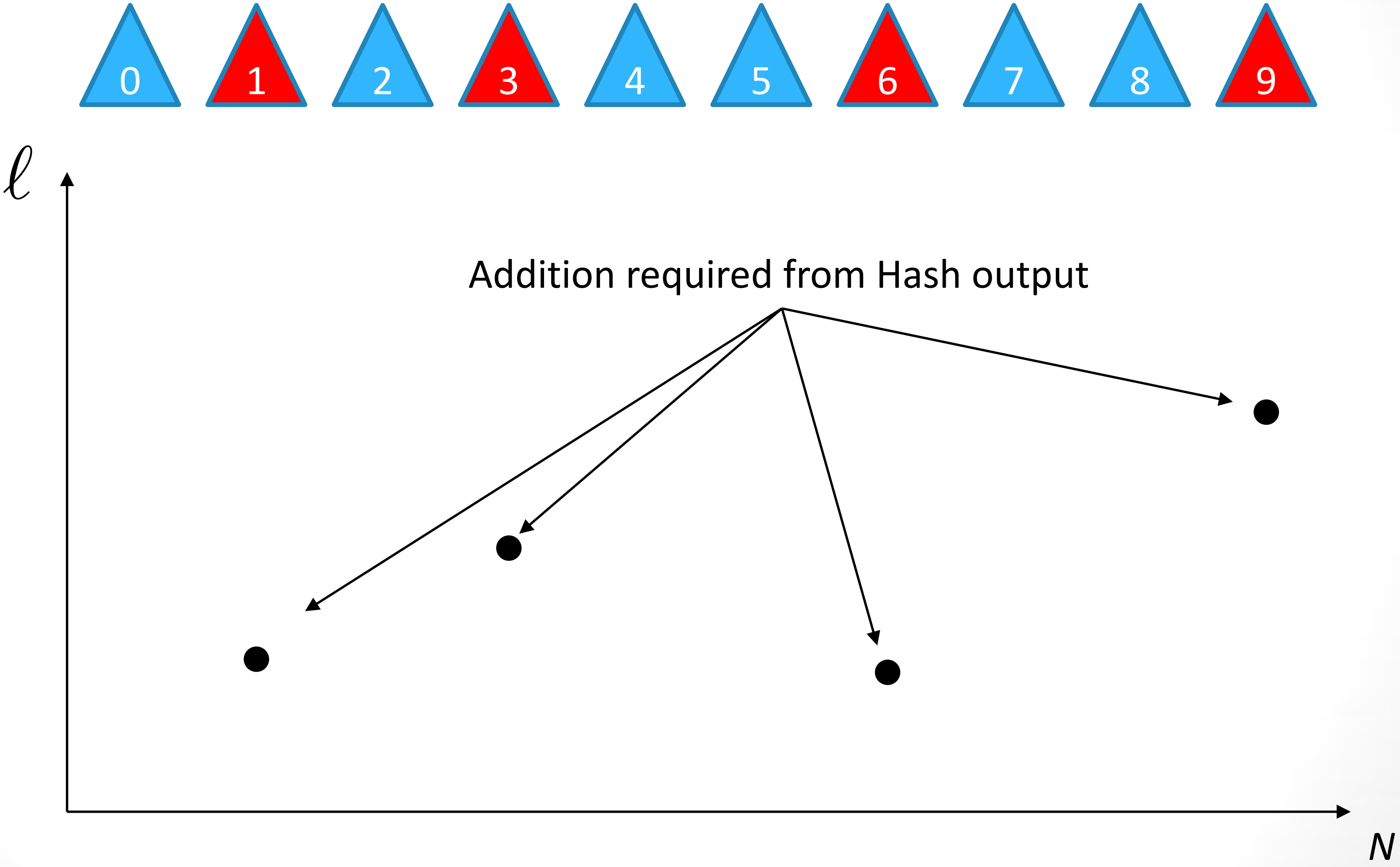
# Applying it to RSS



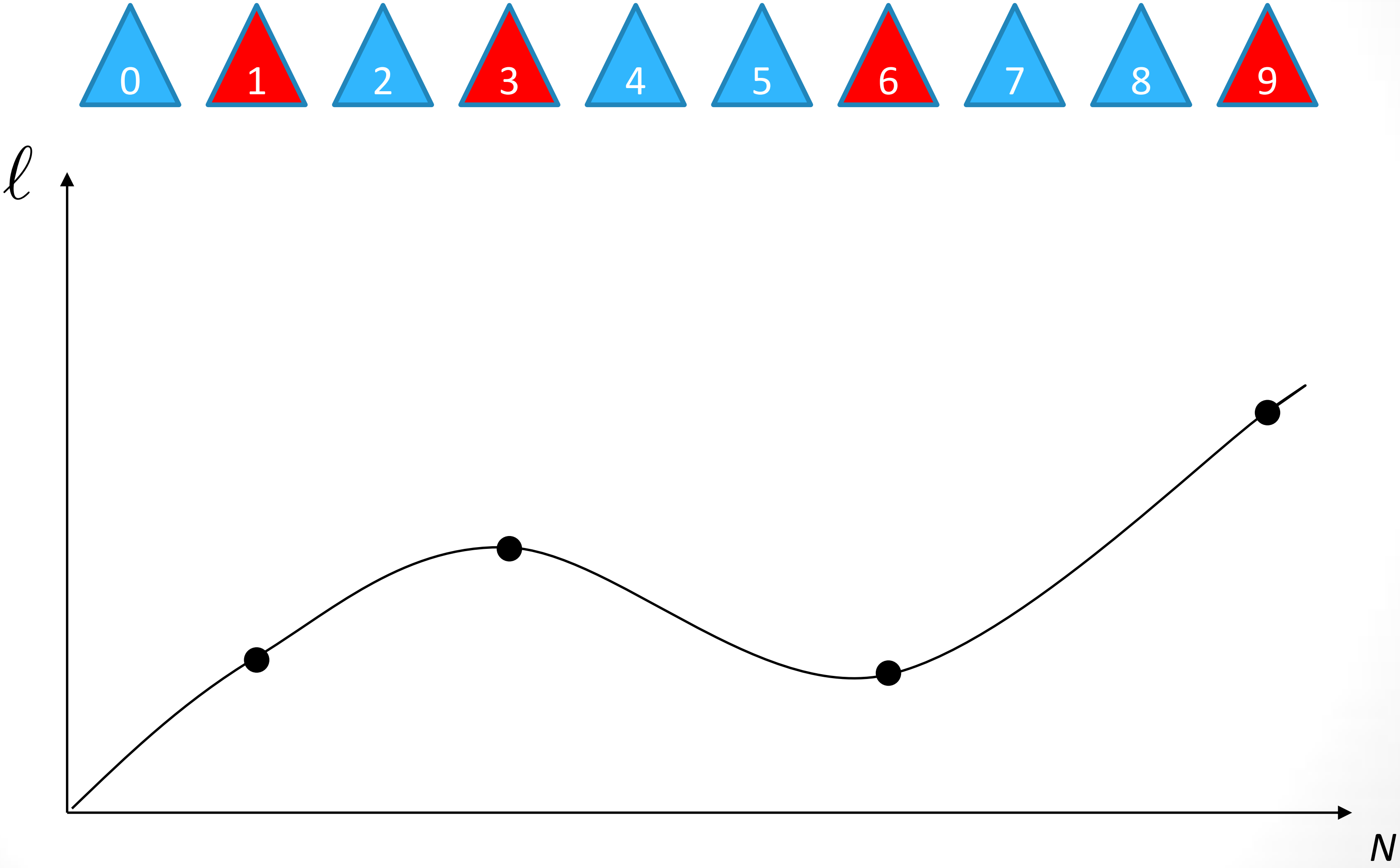
# Applying it to RSS



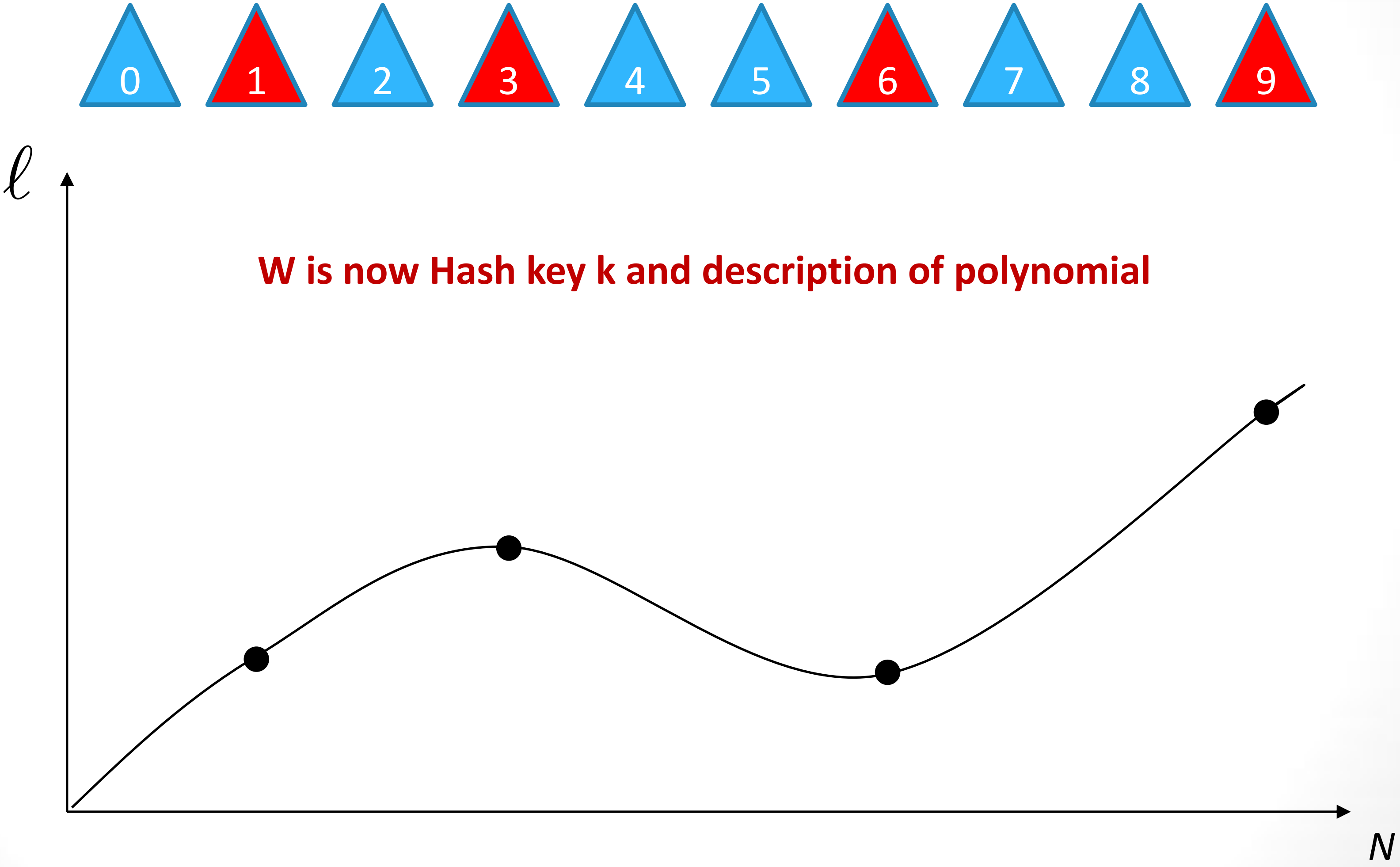
# Applying it to RSS



# Applying it to RSS

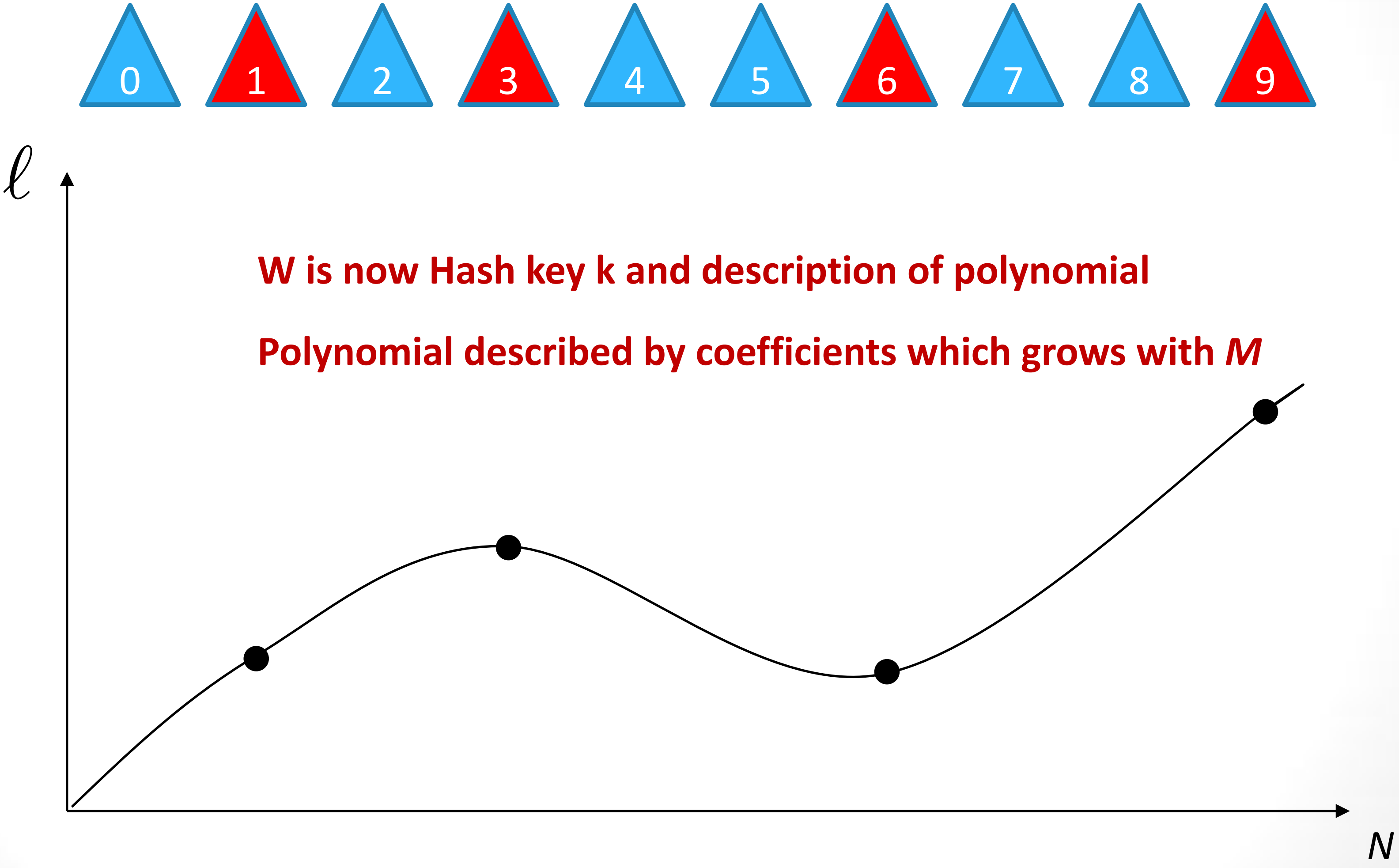


# Applying it to RSS



W is now Hash key k and description of polynomial

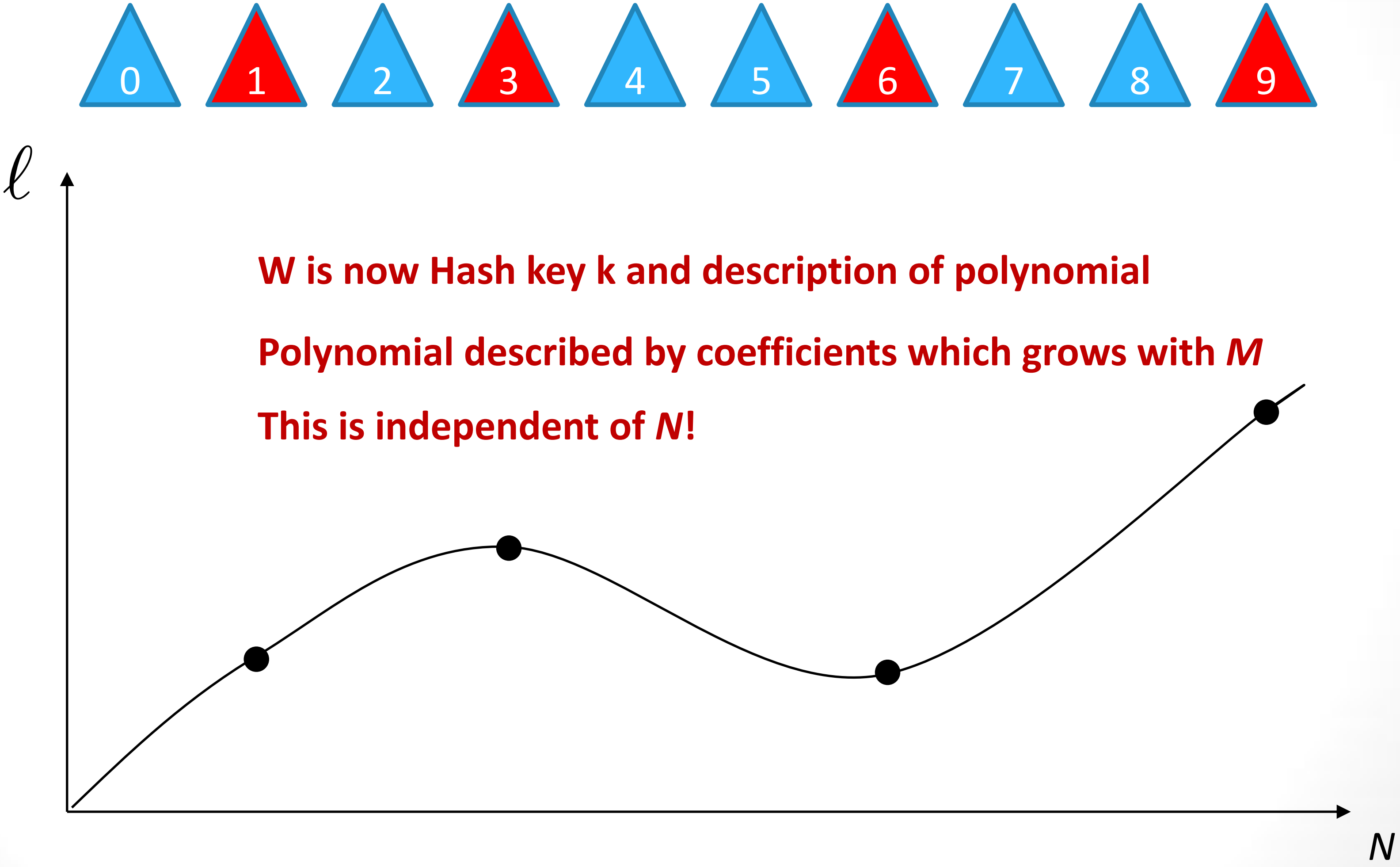
# Applying it to RSS



**W is now Hash key k and description of polynomial**

**Polynomial described by coefficients which grows with *M***

# Applying it to RSS

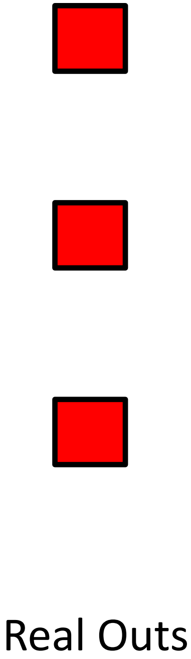


**W is now Hash key k and description of polynomial**

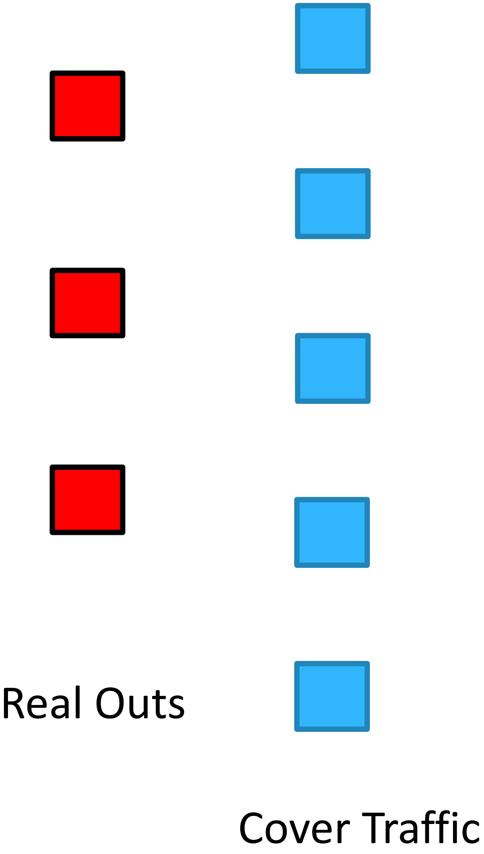
**Polynomial described by coefficients which grows with *M***

**This is independent of *N*!**

# Security of RSS



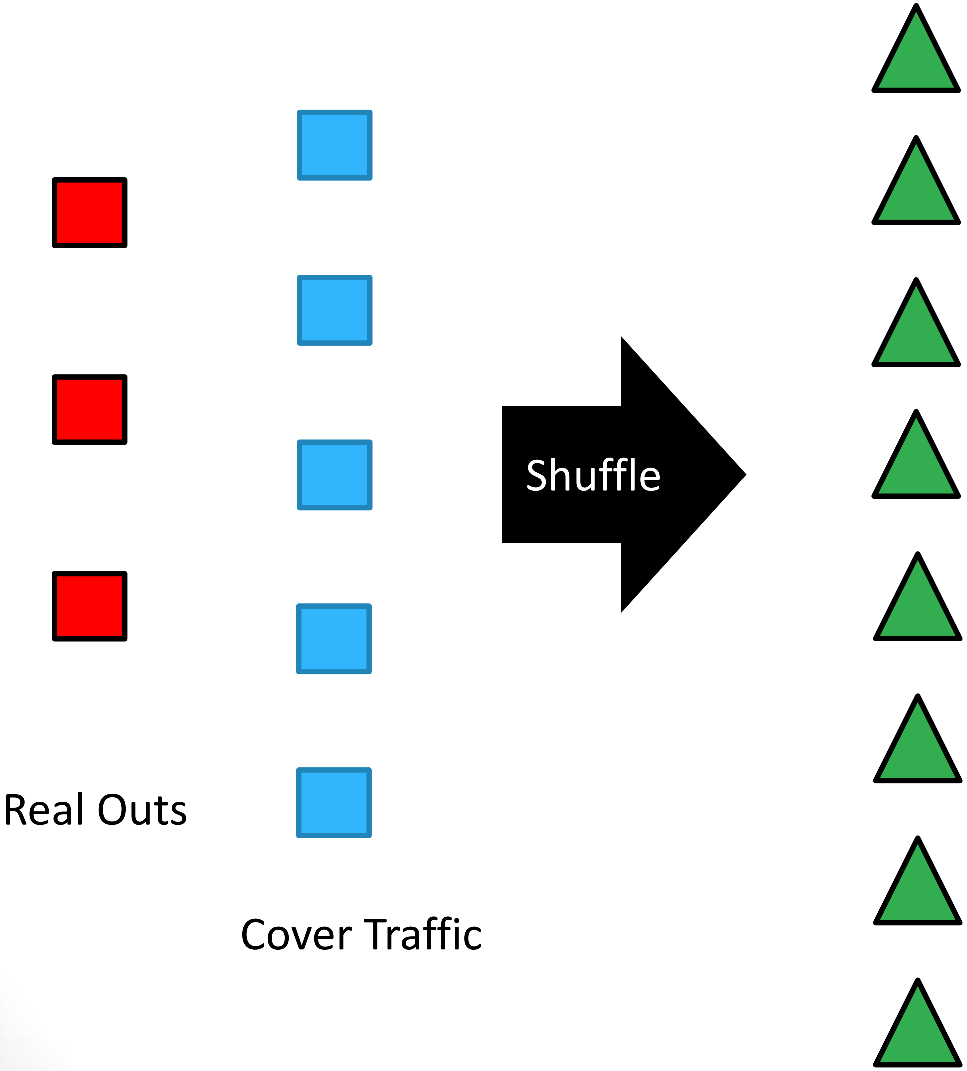
# Security of RSS: Ideal Model



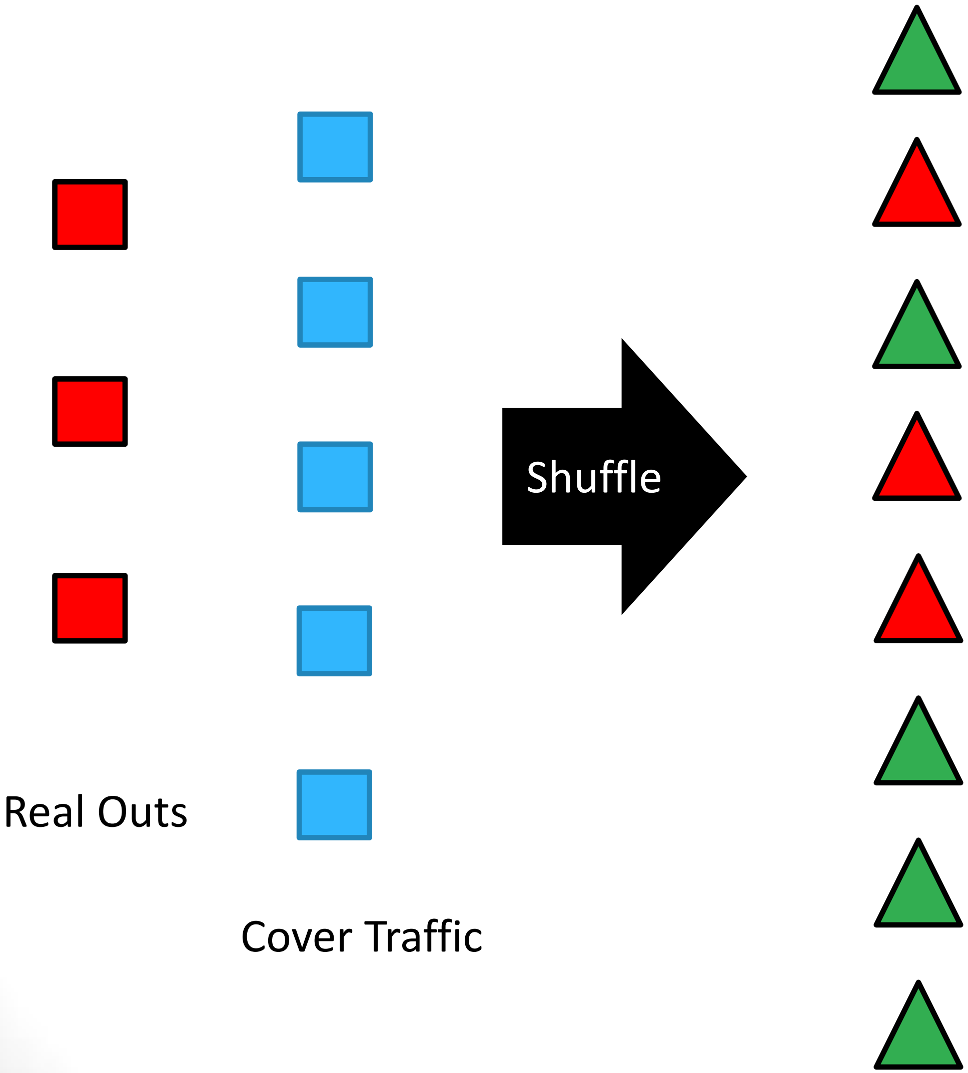
Real Outs

Cover Traffic

# Security of RSS: Ideal Model

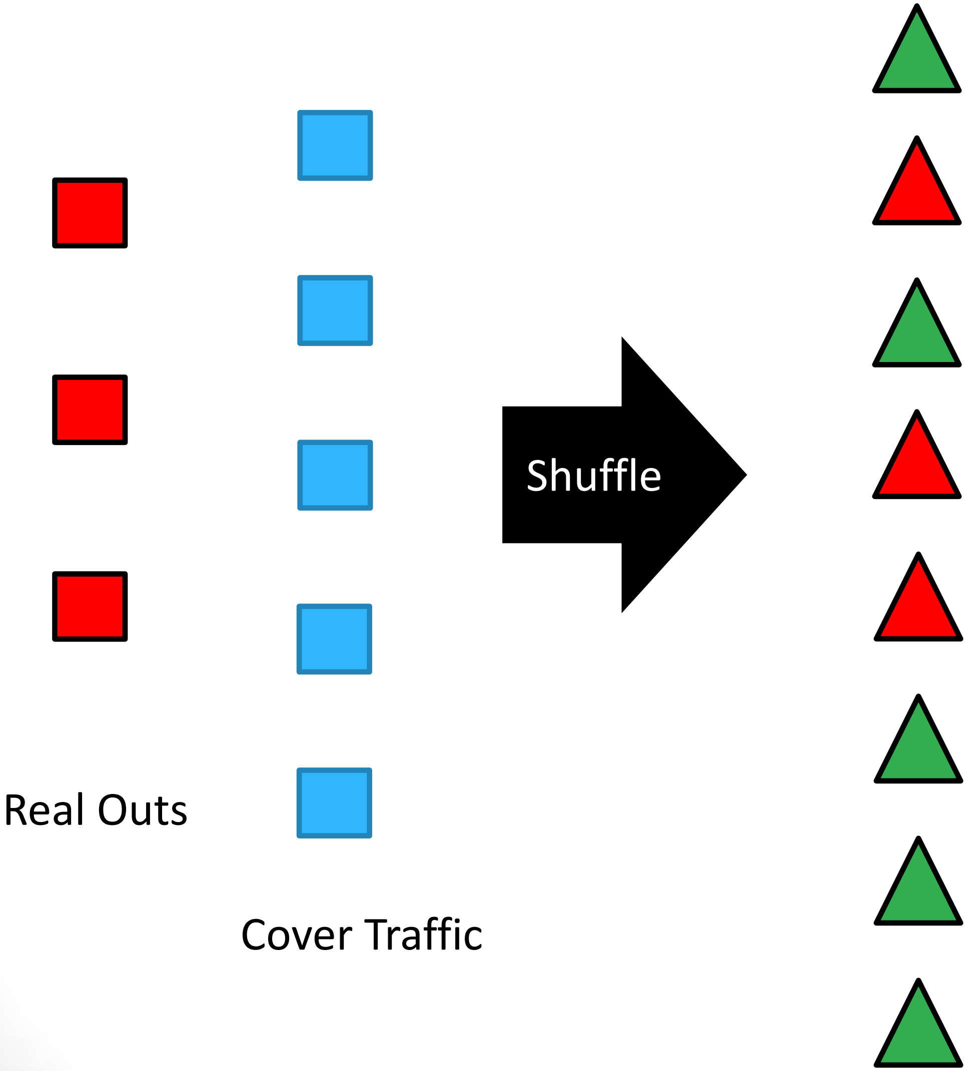


# Security of RSS: Ideal Model



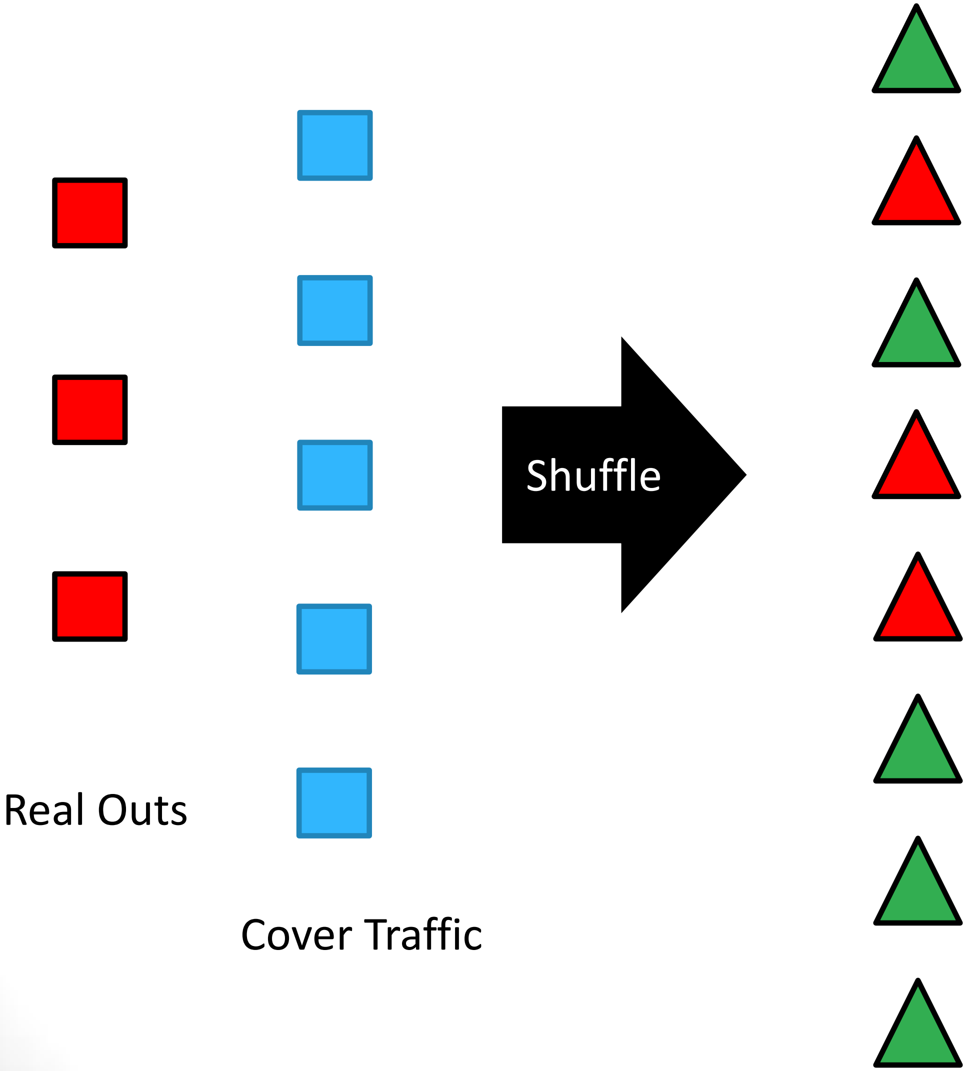
Real outputs may be obviously different from cover traffic

# Security of RSS: Ideal Model



Real outputs may be obviously different from cover traffic

# Security of RSS: Ideal Model



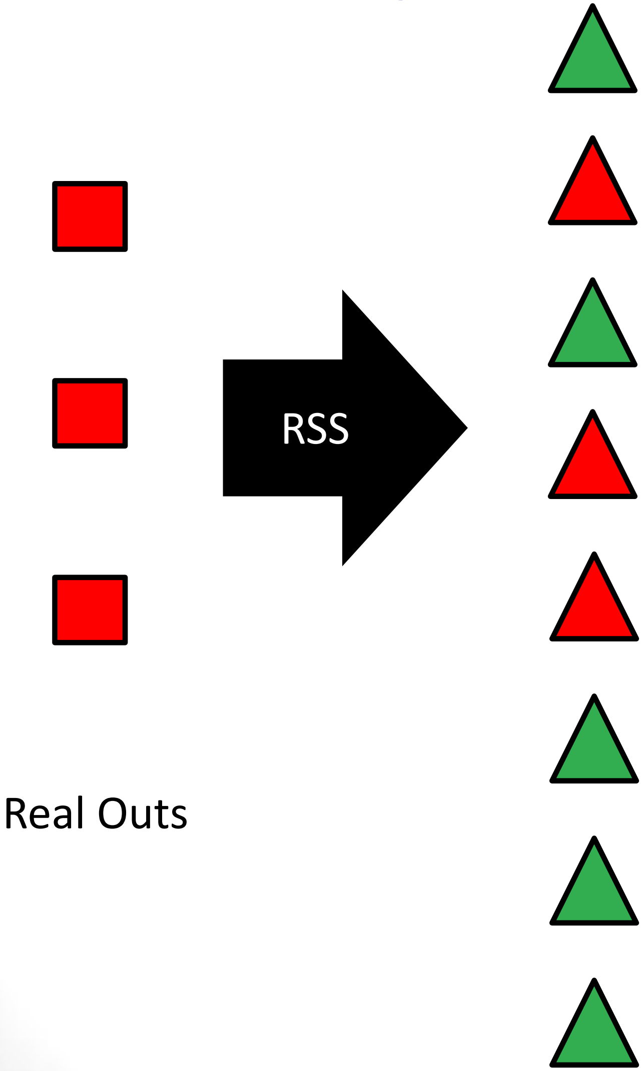
Real Outs

Cover Traffic

Shuffle

Cover traffic is still randomly distributed in this scenario

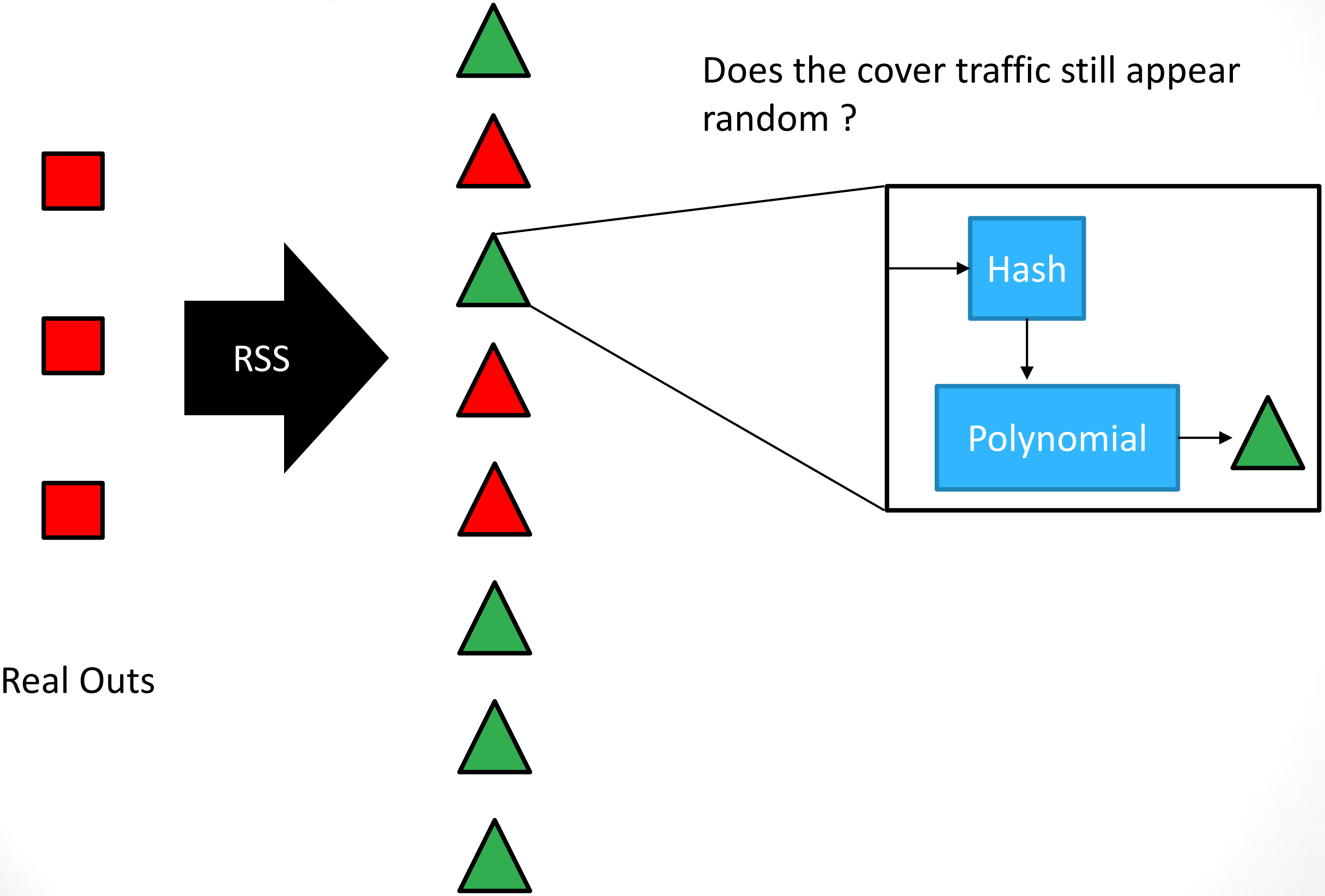
# Security of RSS: Real Model


RSS

Real Outs

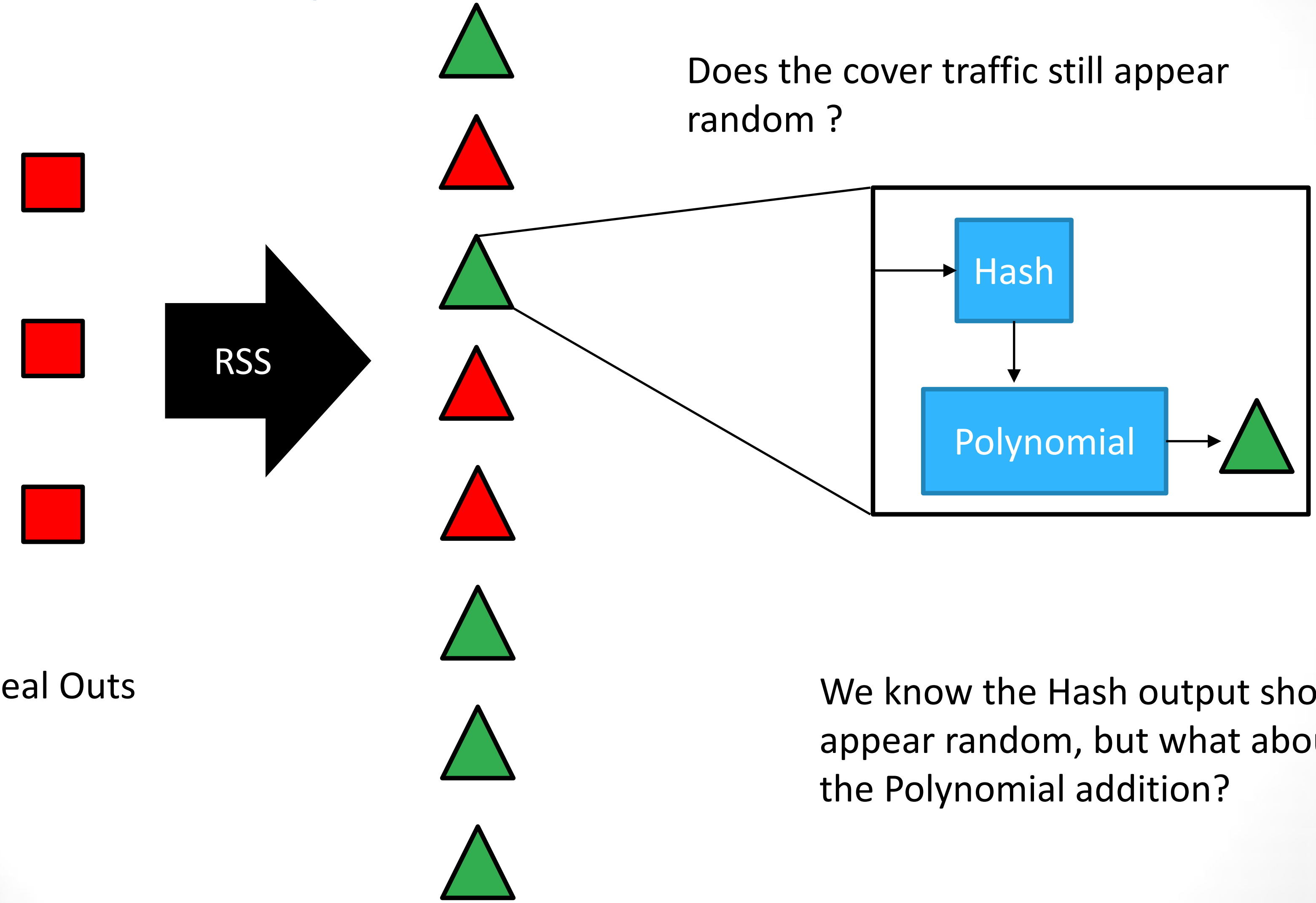
Does the cover traffic still appear random ?

# Security of RSS: Real Model



Does the cover traffic still appear random ?

# Security of RSS: Real Model



Does the cover traffic still appear random ?

Real Outs

We know the Hash output should appear random, but what about the Polynomial addition?

# Security of RSS: Real Model

- Polynomial is uniquely defined by coefficients
- Coefficients uniquely determined by interpolated points
- Interpolated points determined by Hash Output
- Hash output appears to be Random

Thus, Polynomial addition is also random!

# Non Uniform Distributions

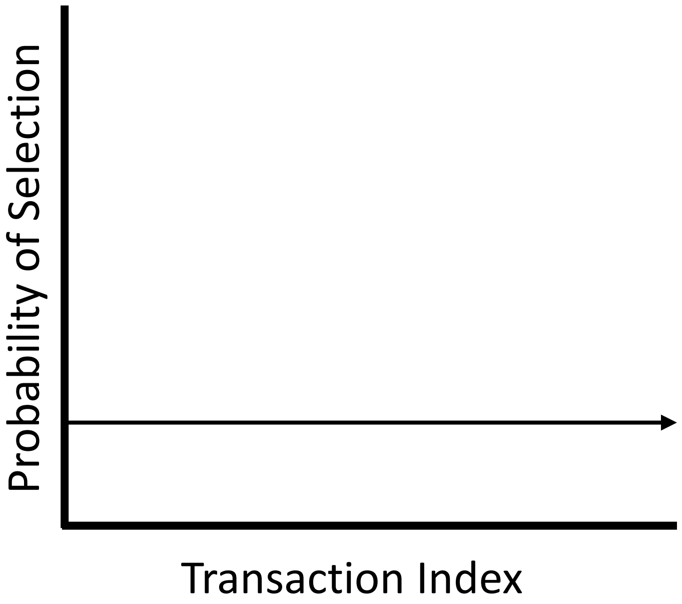
- This process works well enough for protocols that use uniform sampling like ByteCoin

- We can generalize RSS by using Inverse Transform Sampling

- For Monero, its even easier!

# Non Uniform Distributions

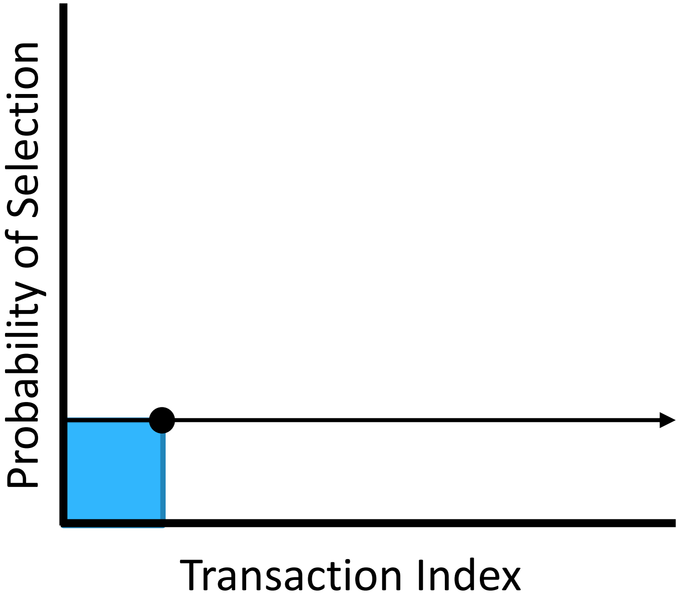
- This process works well enough for protocols that use uniform sampling like ByteCoin

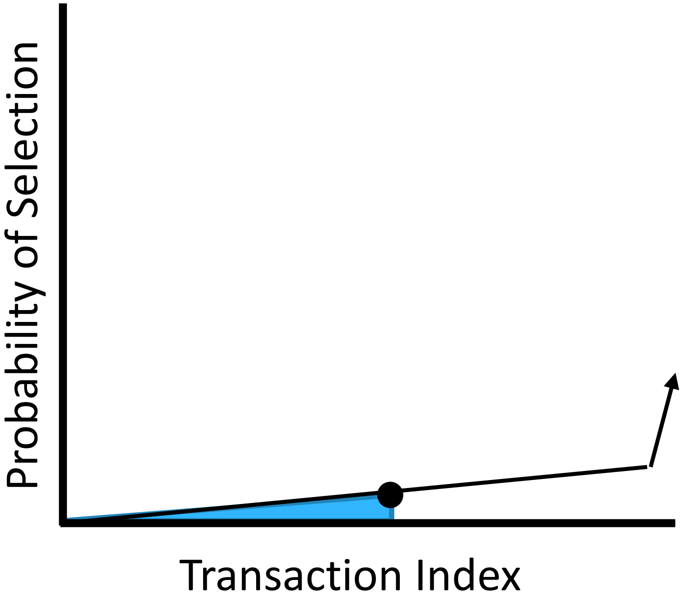


- Hash function essentially performs a uniform sample

# Non Uniform Distributions

- Fairly straight forward technique to adapt uniform samples to other distributions





- Essentially map to points that have the same cumulative probability

# RSS in Practice

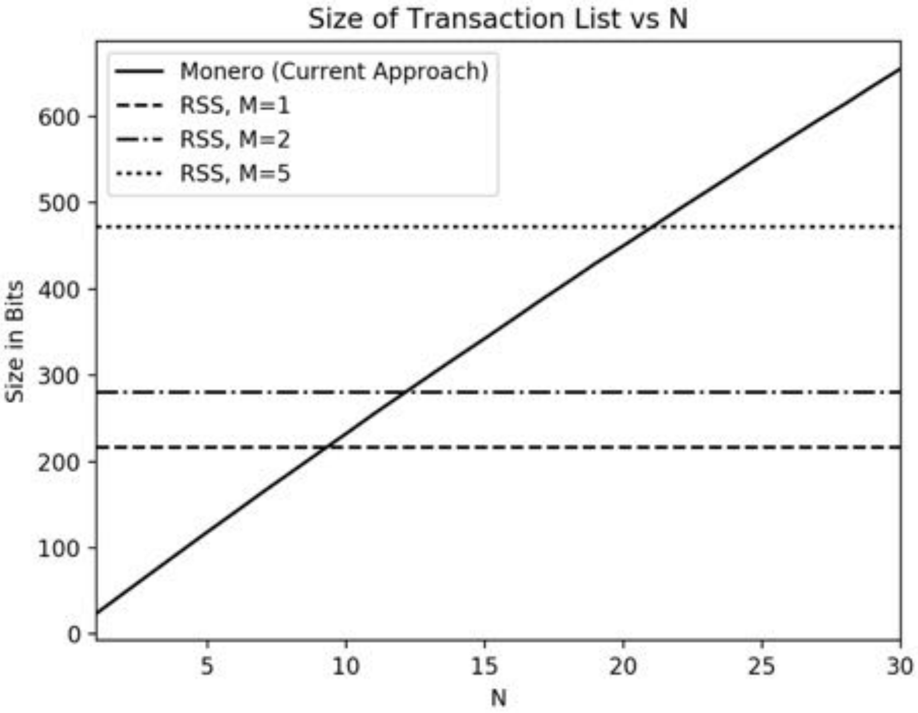
- Want to compare anonymity costs of RSS to existing implementations

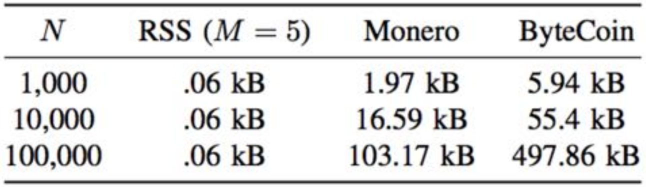
# RSS in Practice

- Want to compare anonymity costs of RSS to existing implementations

- Measure the bandwidth of W vs traditional Cover Set description
  - We not measure computation here, but found it to be negligible in our simulations

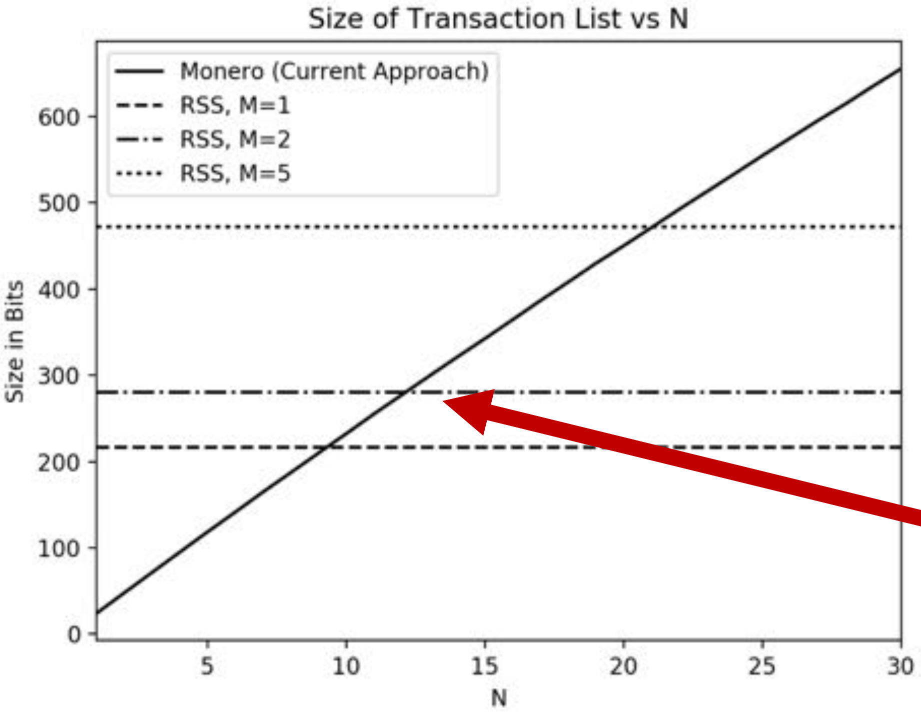
# RSS in Practice

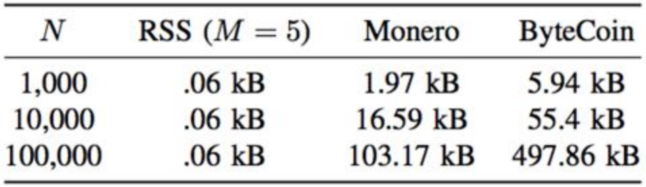




| $N$ | RSS $(M = 5)$ | Monero | ByteCoin |
|---|---|---|---|
| 1,000 | .06 kB | 1.97 kB | 5.94 kB |
| 10,000 | .06 kB | 16.59 kB | 55.4 kB |
| 100,000 | .06 kB | 103.17 kB | 497.86 kB |

# RSS in Practice





| $N$ | RSS ($M = 5$) | Monero | ByteCoin |
|---|---|---|---|
| 1,000 | .06 kB | 1.97 kB | 5.94 kB |
| 10,000 | .06 kB | 16.59 kB | 55.4 kB |
| 100,000 | .06 kB | 103.17 kB | 497.86 kB |

RSS out performs other schemes in practice after this point

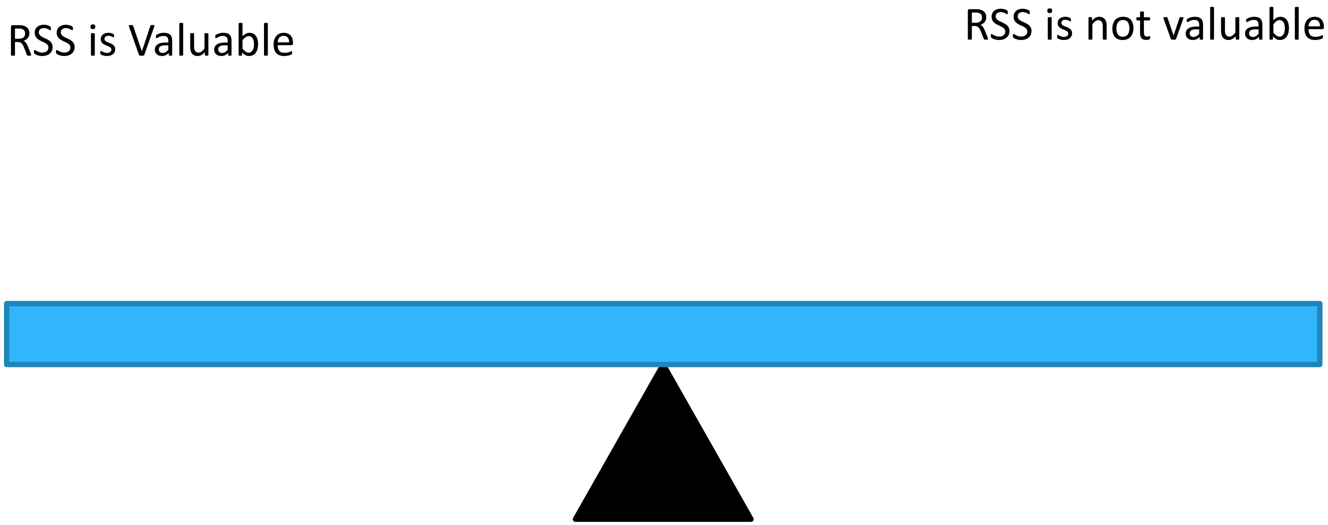
# Looking Ahead

- Having a programmable sampling method seems to be generally useful
  - Providing stronger Anonymity in other contexts
  - Client-Server Puzzles with modified difficulty

# Conclusion

- Mixing Cryptocurrencies such as Bytecoin and Monero are currently lacking in level of anonymity provided
- Recent work is drastically reducing the cost of proofs
- Cover Set description will soon dominant size costs
- RSS provides a way to drastically decrease this cost

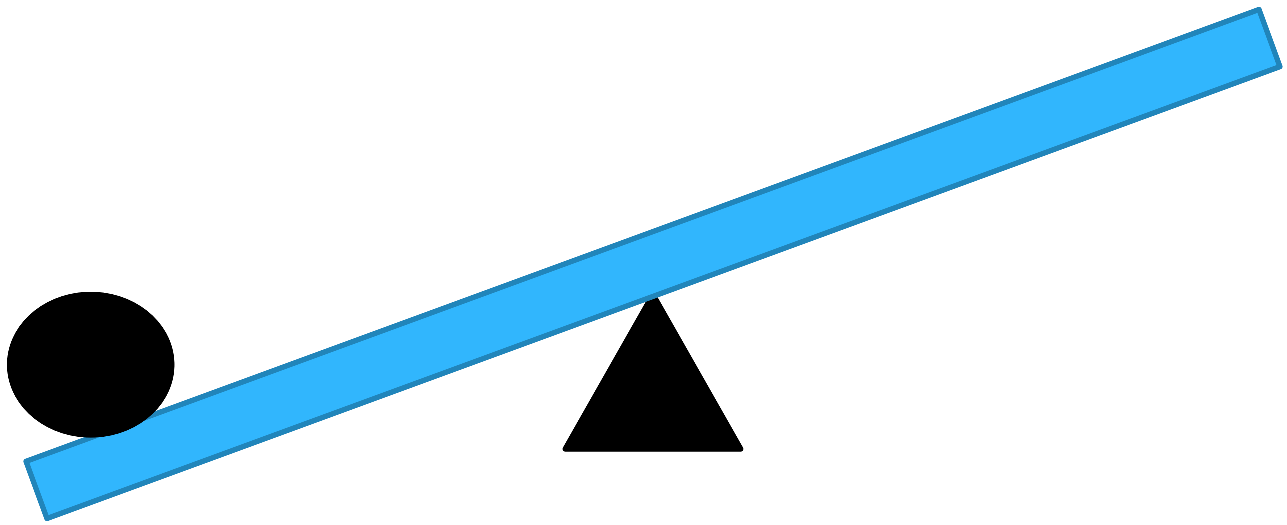
# Conclusion



RSS is Valuable

RSS is not valuable

# Conclusion

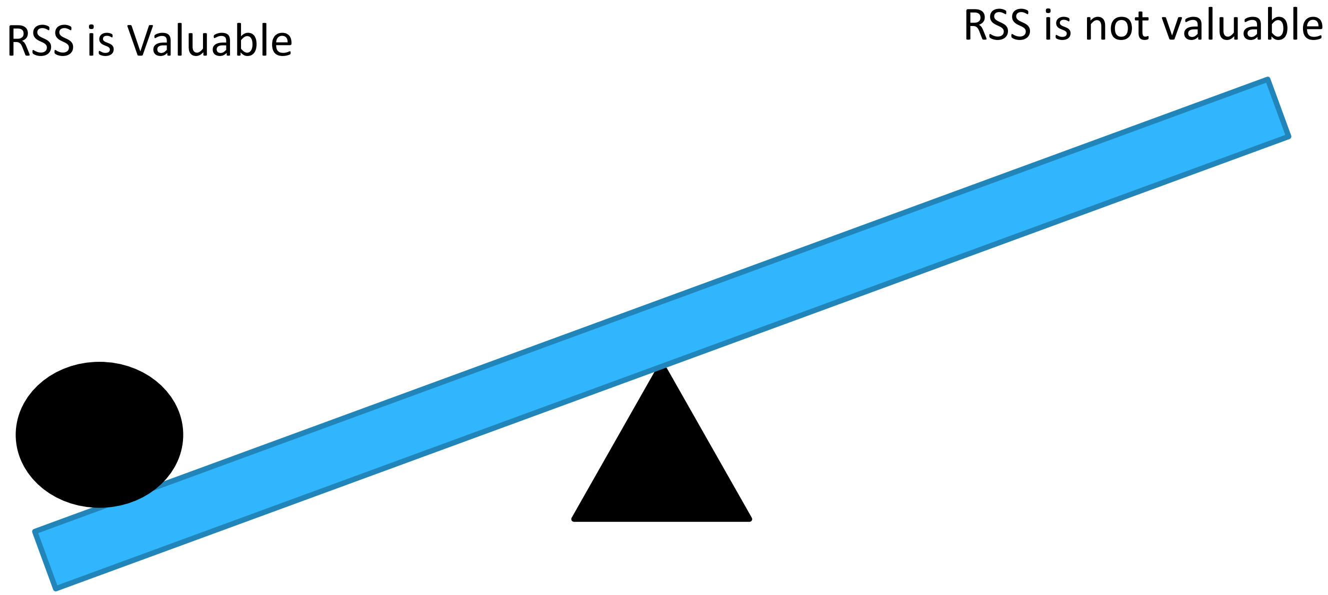


RSS is Valuable

RSS is not valuable

Monero moves to significantly larger Cover Sets

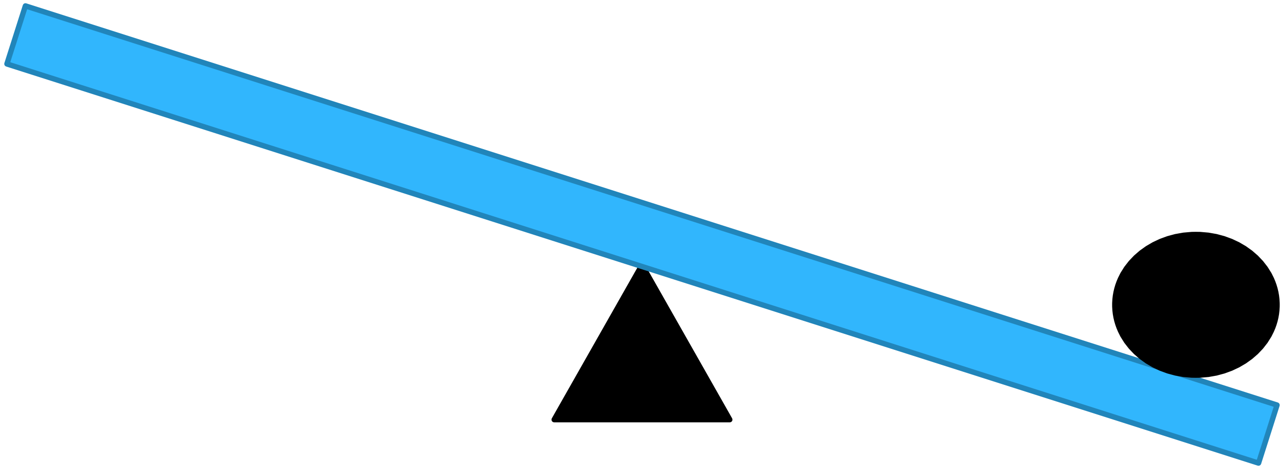
# Conclusion



Monero moves to significantly larger Cover Sets

**RSS offers a clear way to do this efficiently**

# Conclusion

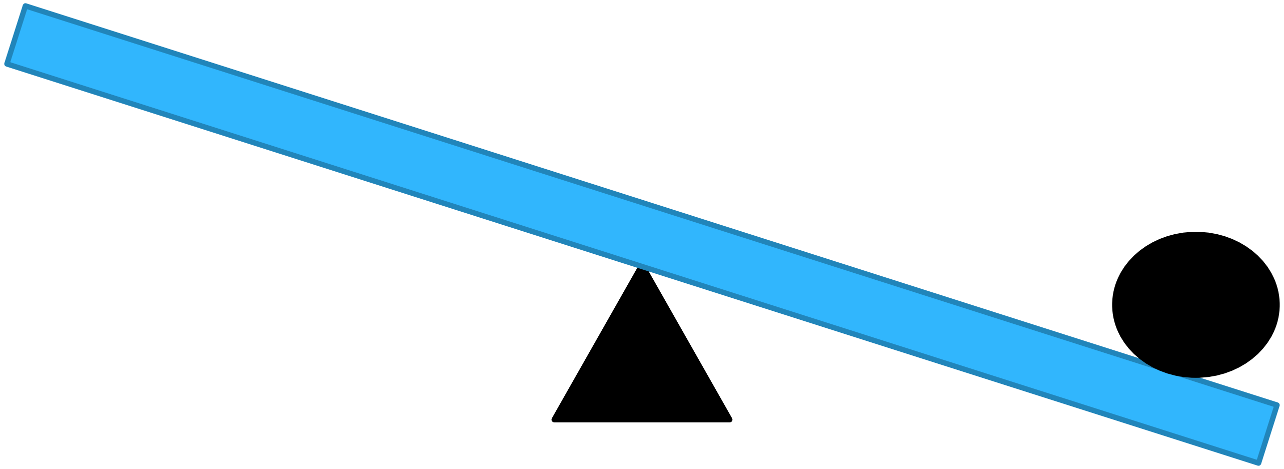


RSS is Valuable

RSS is not valuable

Monero continues to have small Cover Sets

# Conclusion



Monero continues to have small Cover Sets

**Limits anonymity, key feature of Monero**

# THANK YOU!

Email: alishahc@cs.jhu.edu