ETH zürich

On the Privacy Provisions of Bloom Filters in Lightweight Bitcoin Clients

Arthur Gervais, Ghassan O. Karame, Damian Gruber, Srdjan Čapkun

ETH Zurich, NEC Research

ACSAC 2014

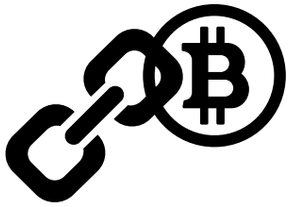
Bitcoin

- Peer-to-peer decentralized currency

- Users keep Bitcoins in a wallet containing multiple **addresses** (@)

- Unlinkability between @

- Log of **all** transactions



Bitcoin's scalability problems

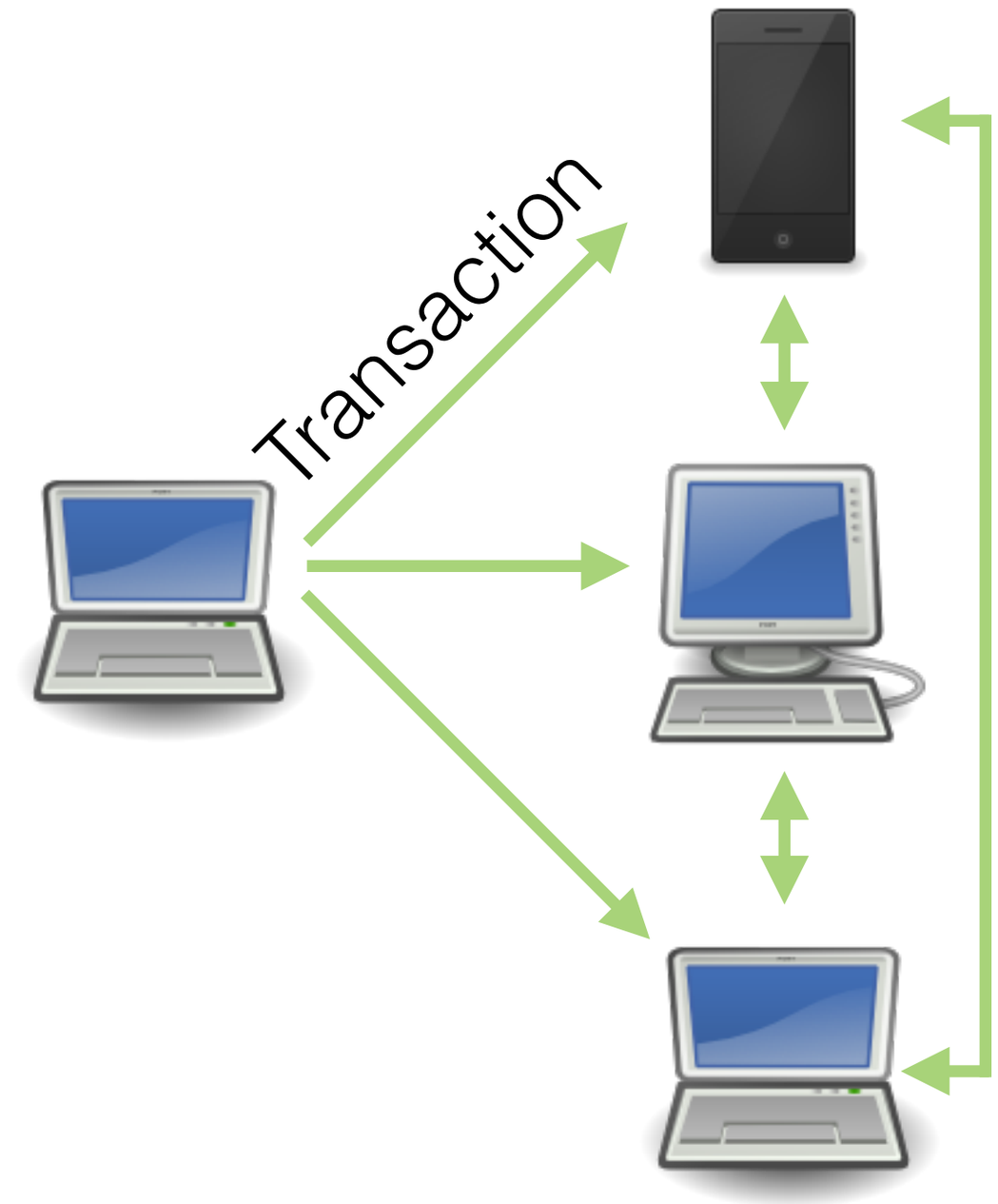
1. Log of transactions (>25 GB)

2. Clients receive irrelevant transactions

3. Limited data traffic over 3G/4G

Bitcoin's scalability problems

1. Log of transactions (>25 GB)
2. Clients receive irrelevant transactions
3. Limited data traffic over 3G/4G



ETH zürich

Enable mobile Bitcoin clients

Bloom filter

0	0	0	0	0	0	0

Enable mobile Bitcoin clients

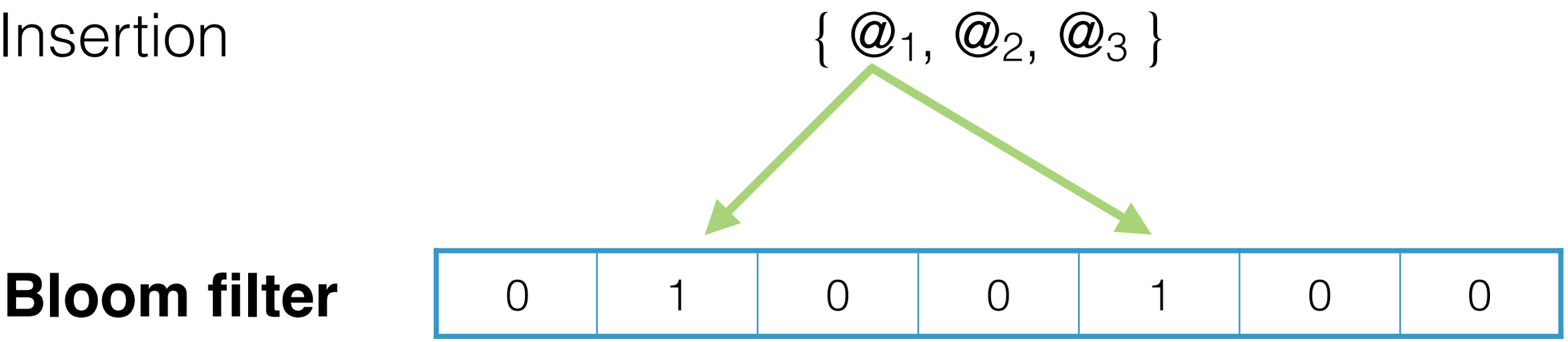
Insertion $\{ @_1, @_2, @_3 \}$

Bloom filter

0	0	0	0	0	0	0

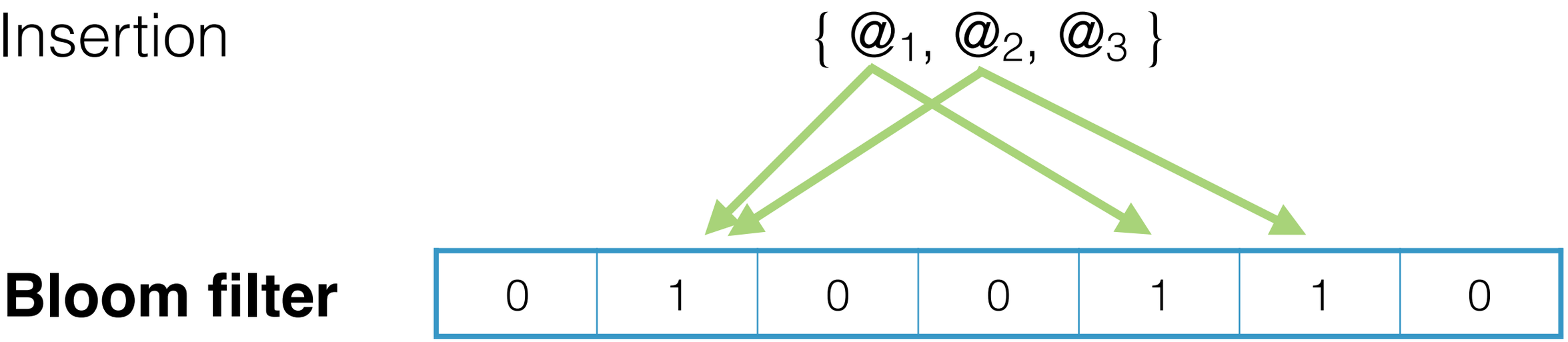
Enable mobile Bitcoin clients

Insertion



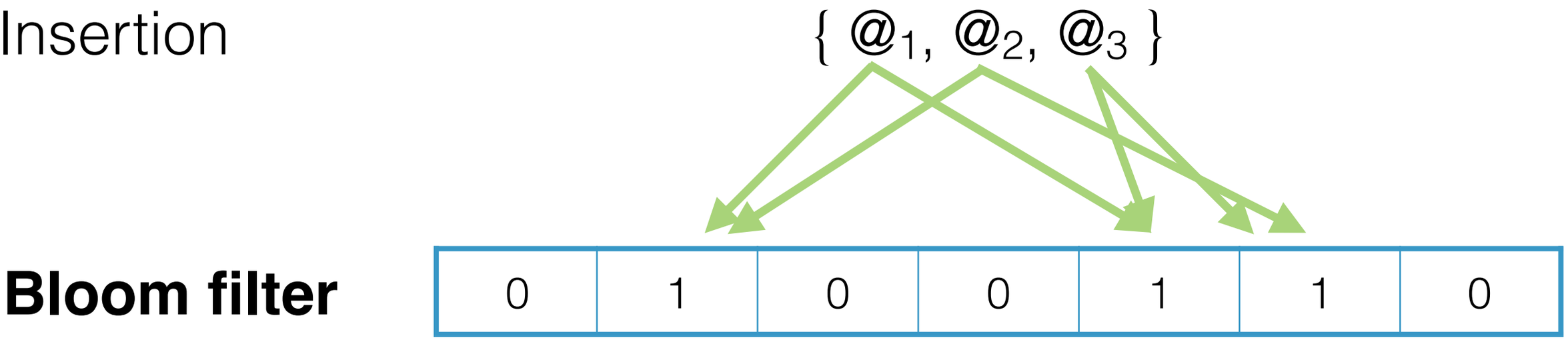
Enable mobile Bitcoin clients

Insertion



Enable mobile Bitcoin clients

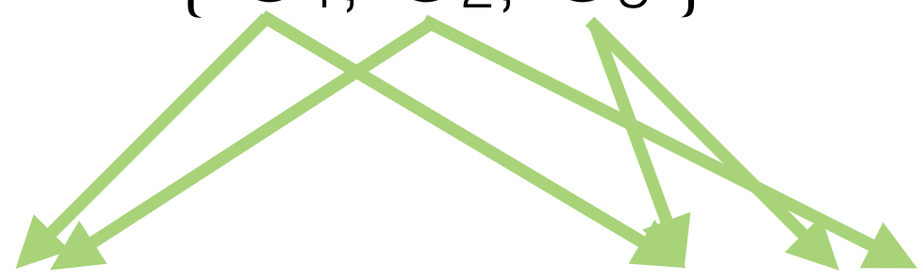
Insertion



Enable mobile Bitcoin clients

Insertion

$\{ @_1, @_2, @_3 \}$



Bloom filter

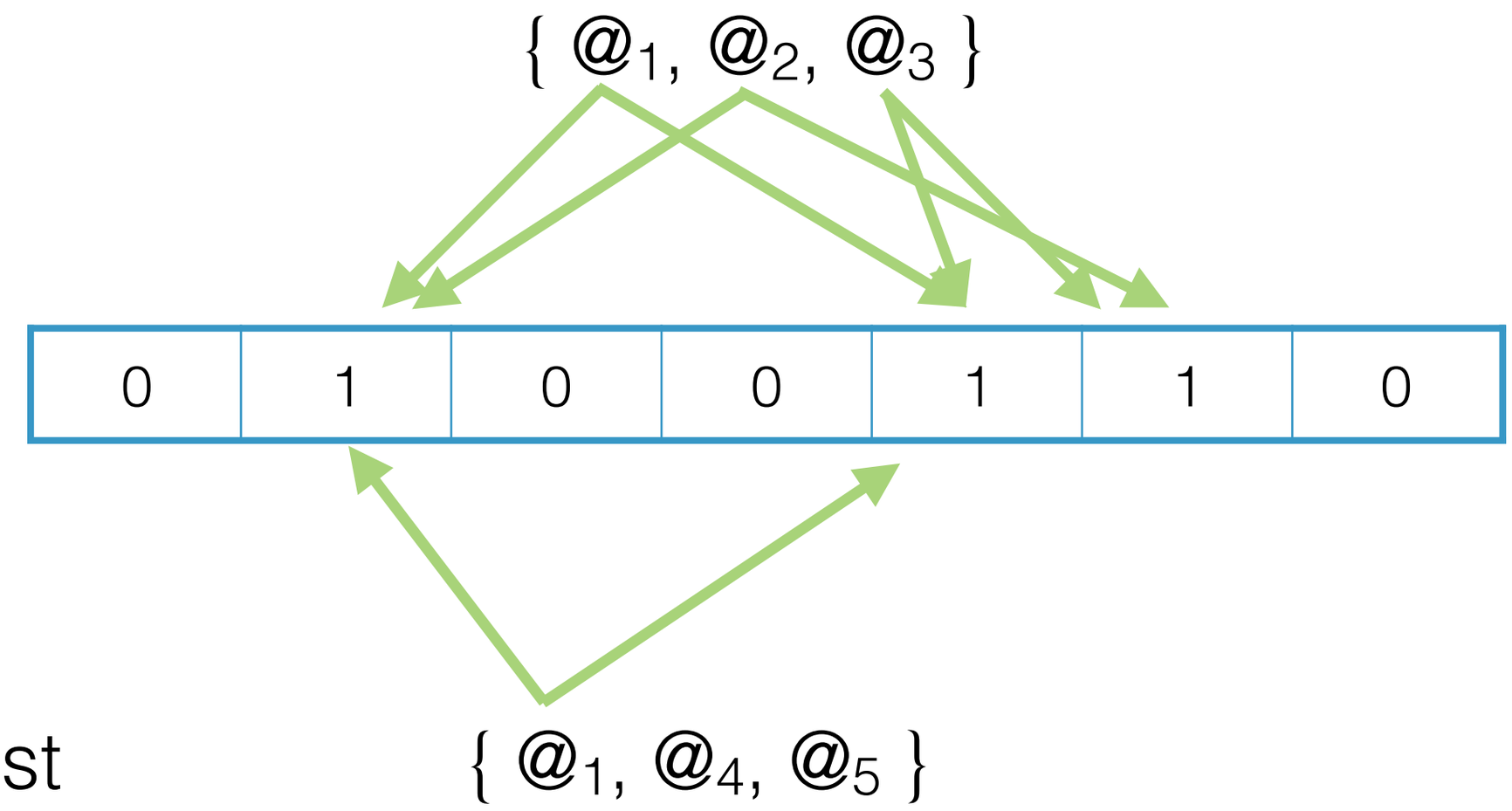
0	1	0	0	1	1	0

Membership test

$\{ @_1, @_4, @_5 \}$

Enable mobile Bitcoin clients

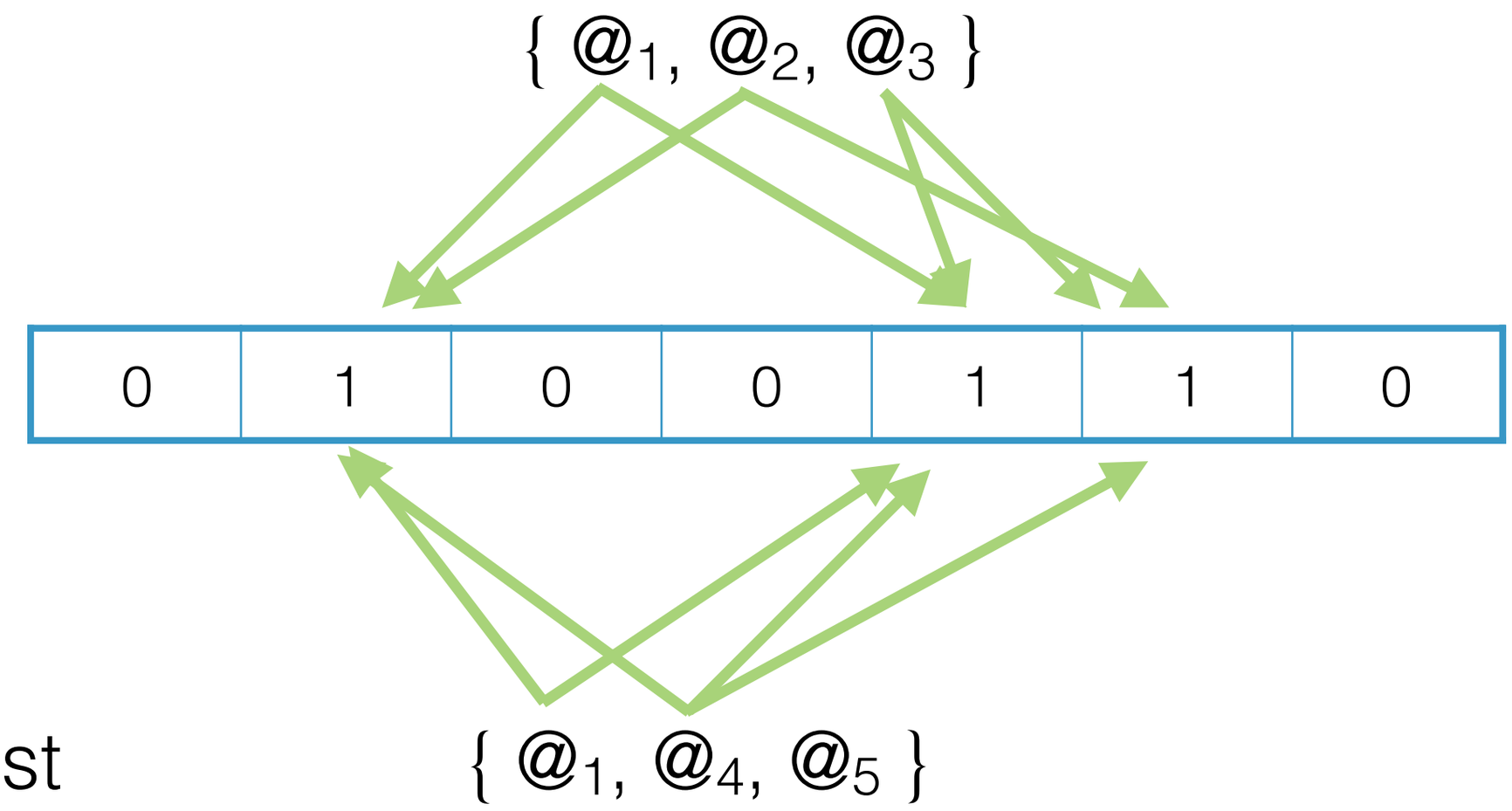
Insertion



Membership test

Enable mobile Bitcoin clients

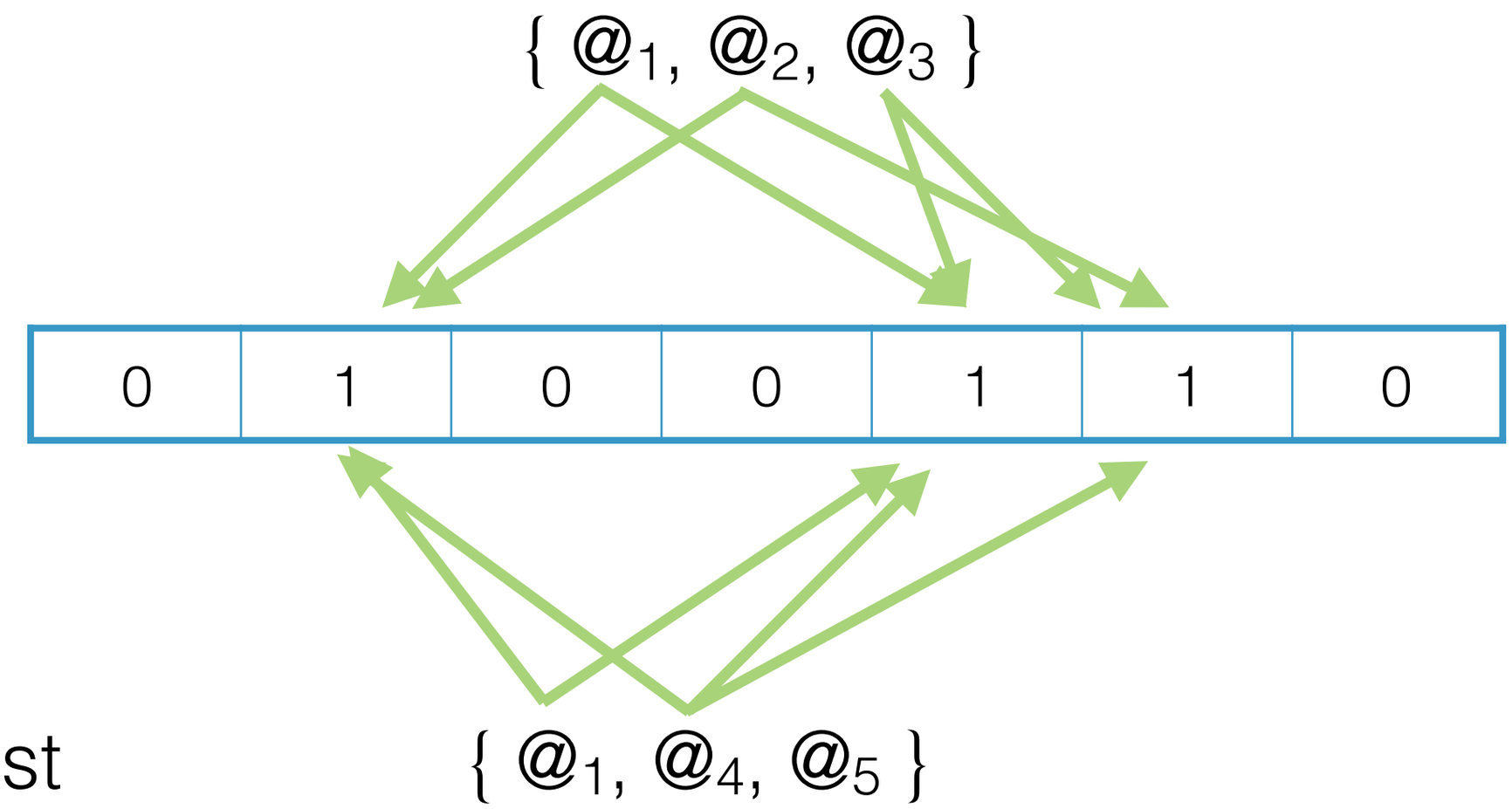
Insertion



Membership test

Enable mobile Bitcoin clients

Insertion

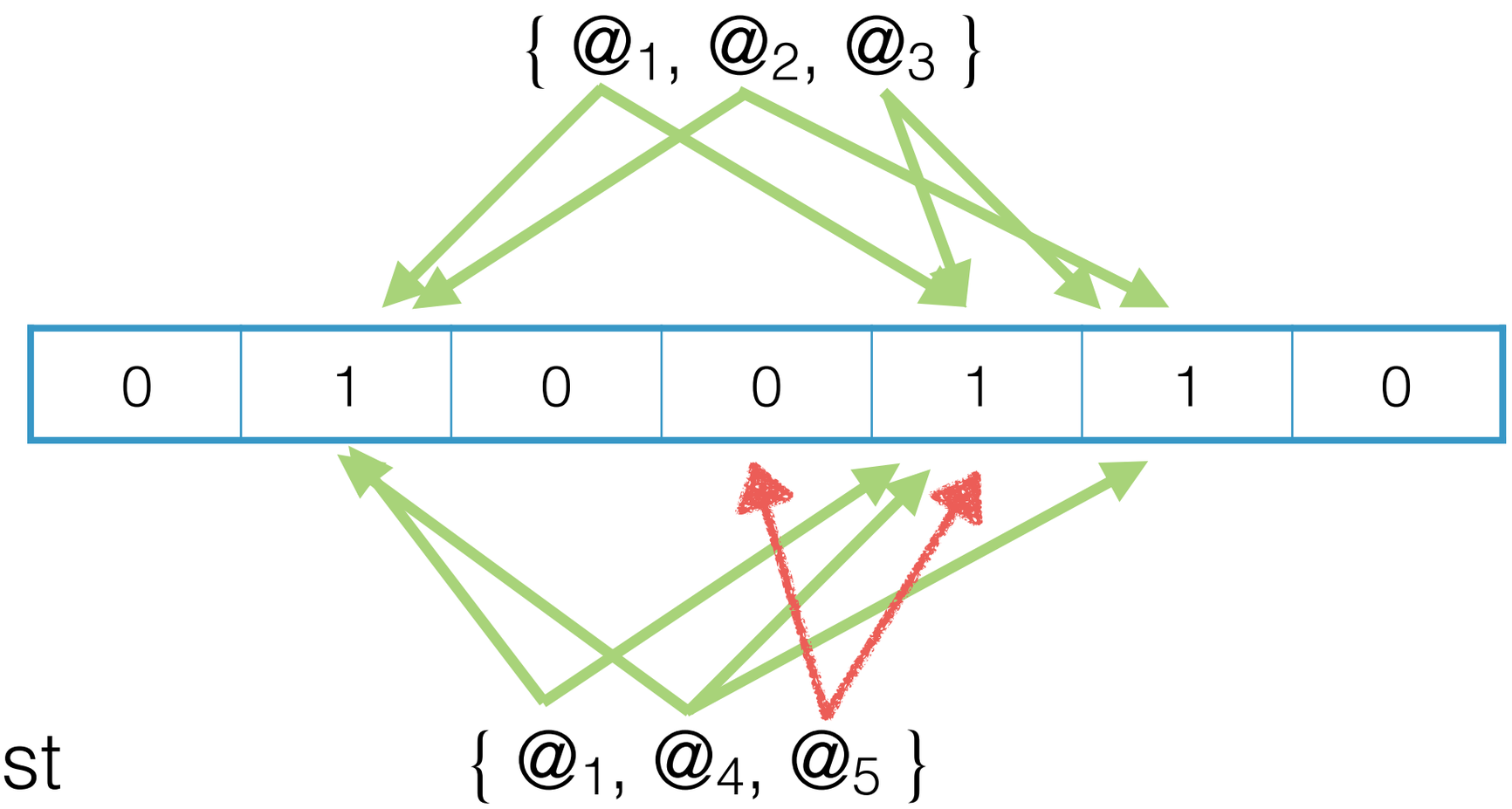


Membership test

! $@_4$ False positive → **target False Positive Rate** (FPR)

Enable mobile Bitcoin clients

Insertion

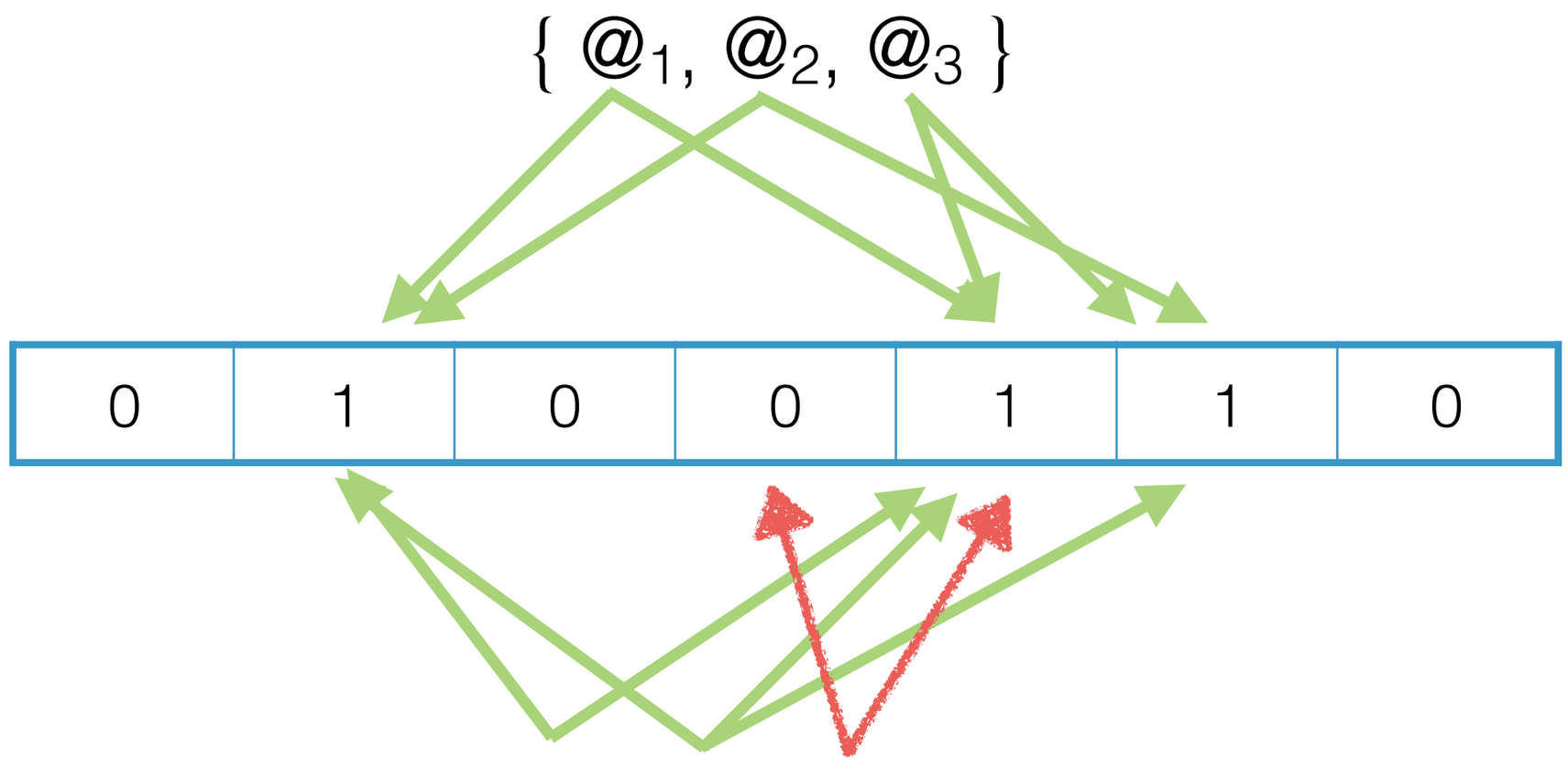


Membership test

! $@_4$ False positive → **target False Positive Rate** (FPR)

Enable mobile Bitcoin clients

Insertion



Membership test

{ @1, @4, @5 }

! @4 False positive → **target False Positive Rate** (FPR)

@5 True negative

Simple Payment Verification (SPV)

Filter transactions not relevant for user

SPV client

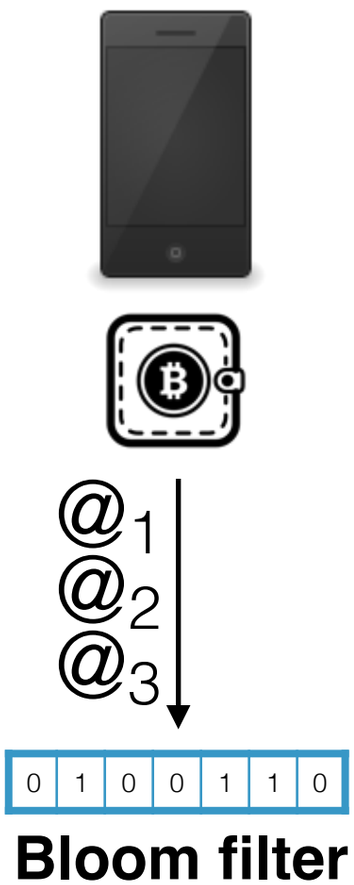
Full Bitcoin node

Full Bitcoin node

Simple Payment Verification (SPV)

Filter transactions not relevant for user

SPV client

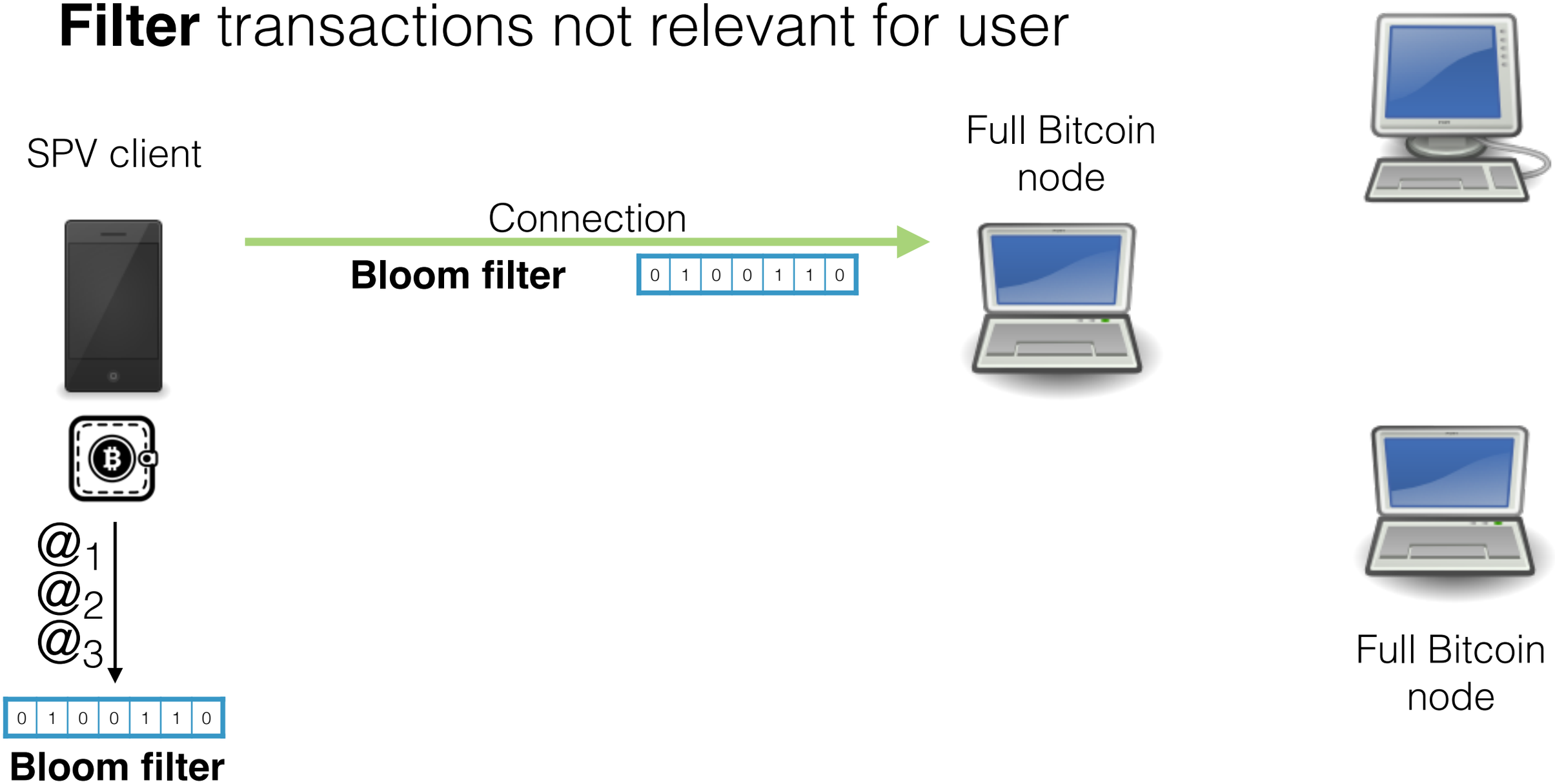


Full Bitcoin node

Full Bitcoin node

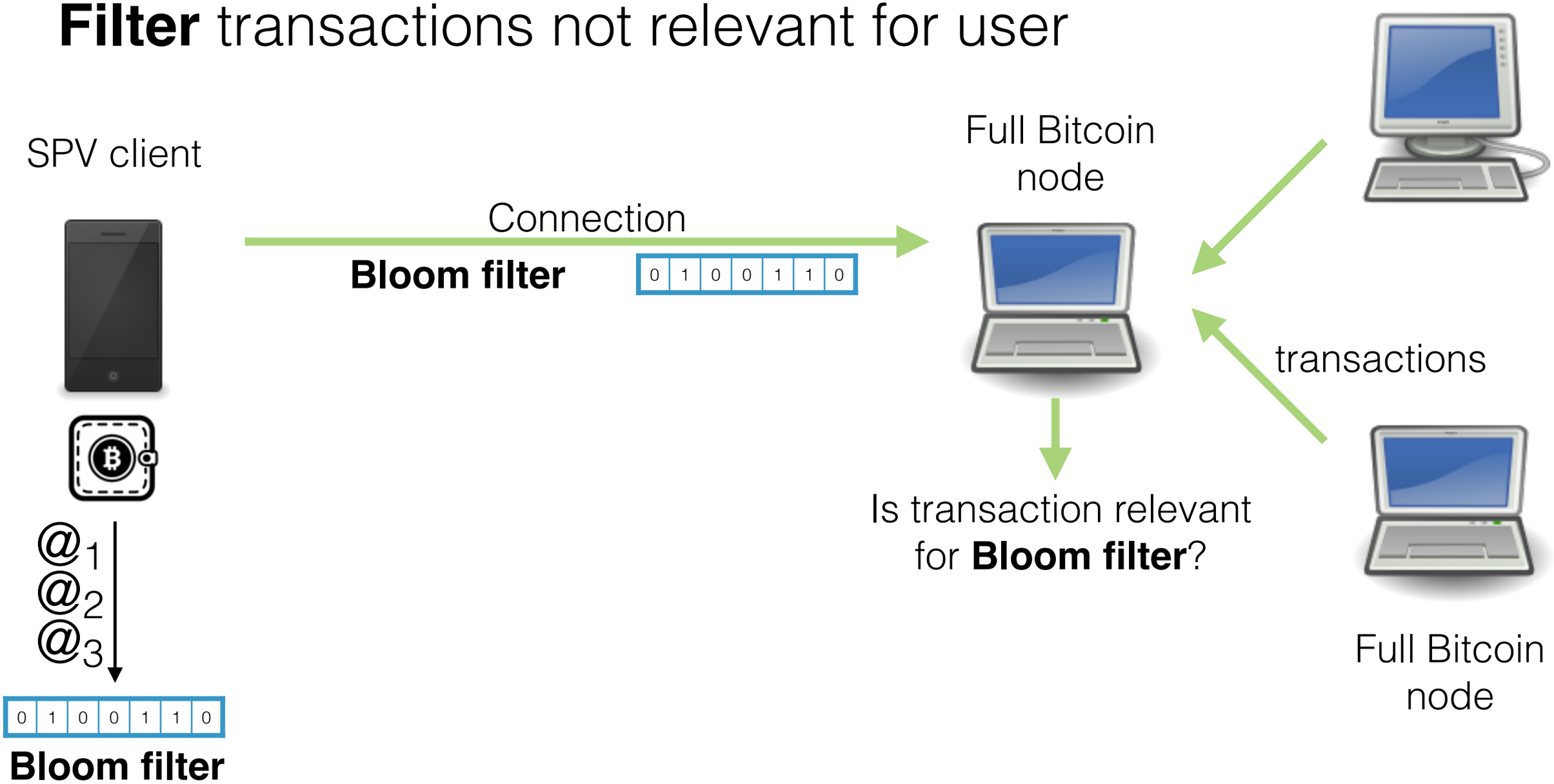
Simple Payment Verification (SPV)

Filter transactions not relevant for user



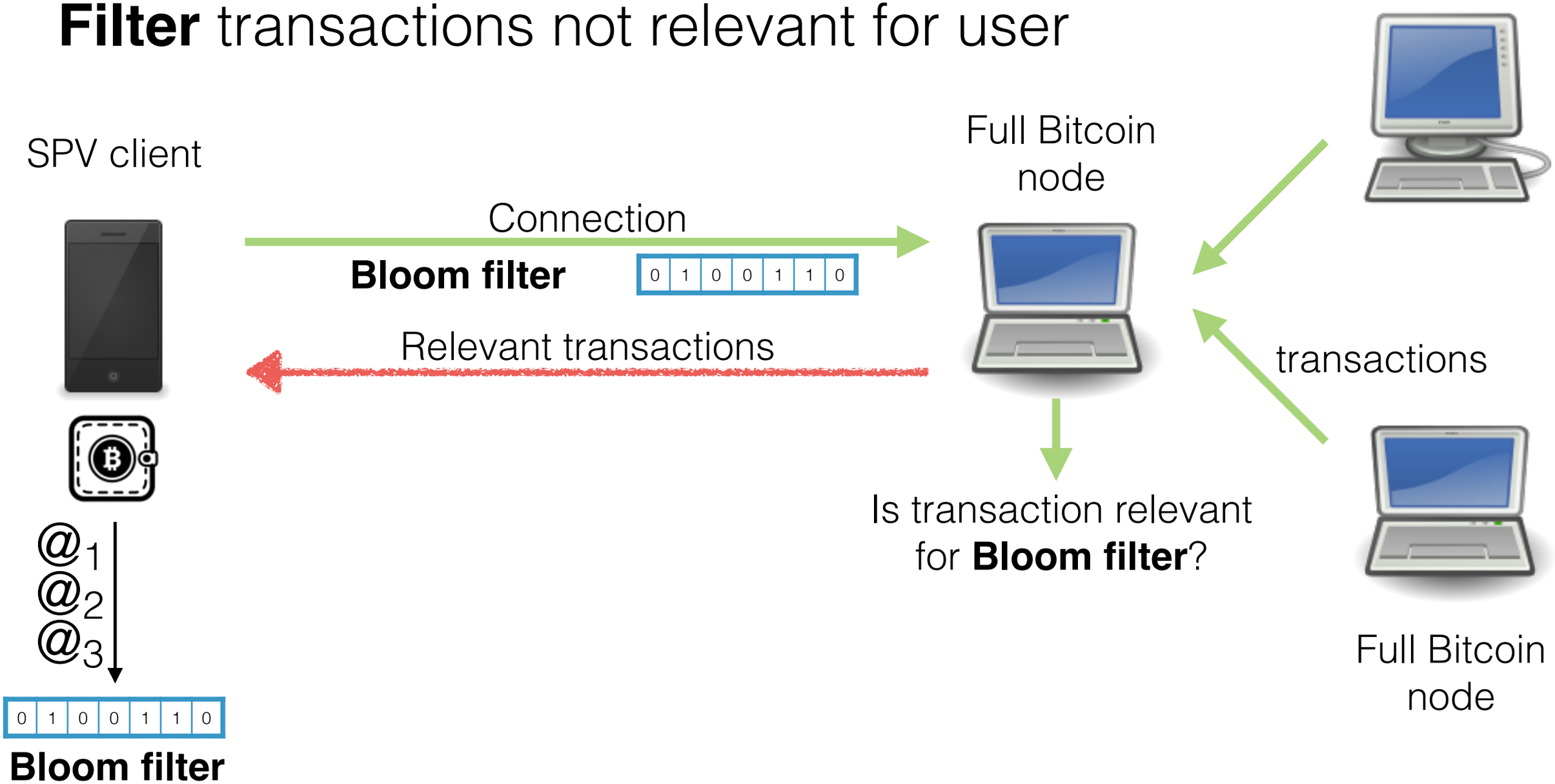
Simple Payment Verification (SPV)

Filter transactions not relevant for user



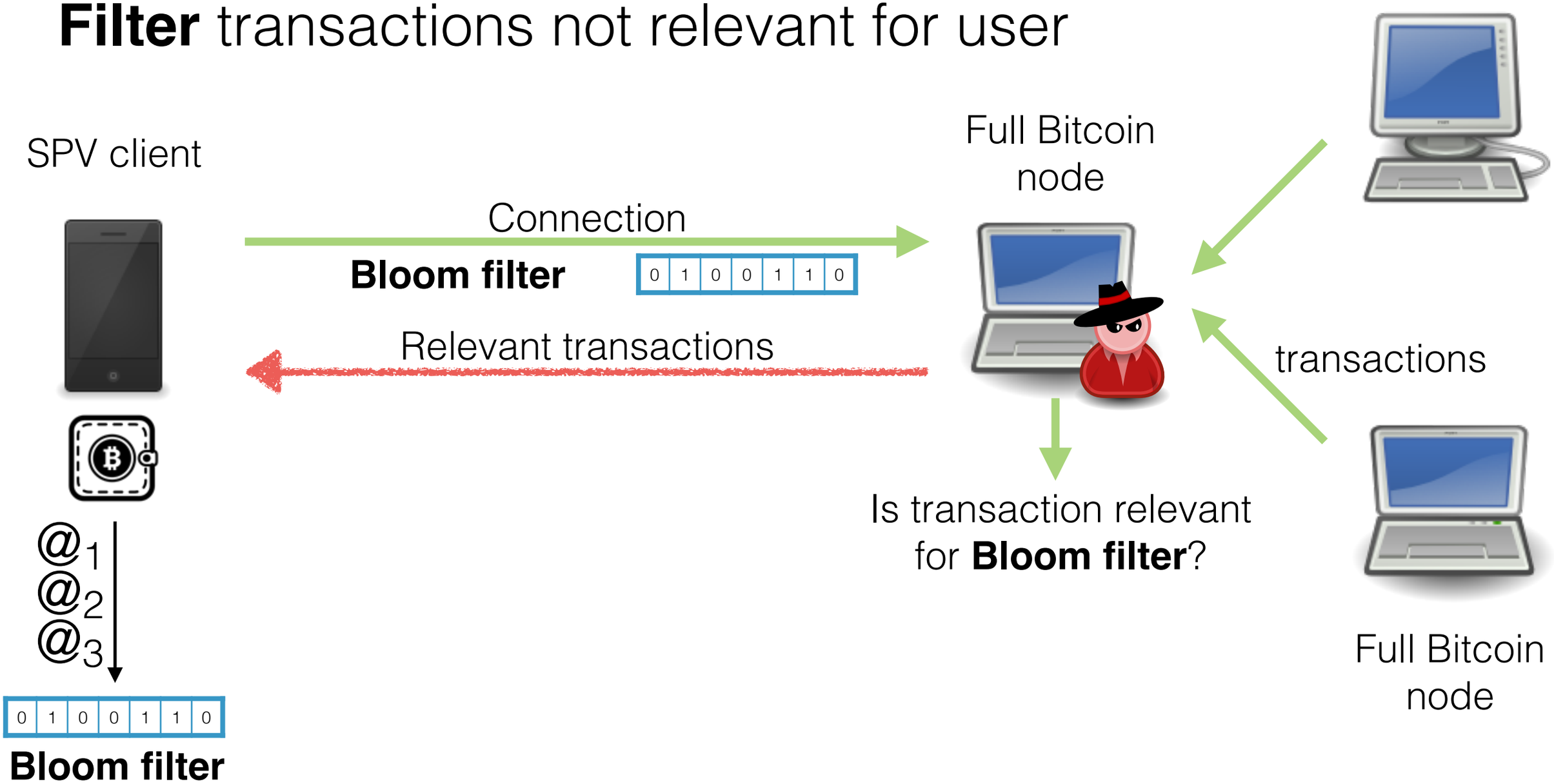
Simple Payment Verification (SPV)

Filter transactions not relevant for user



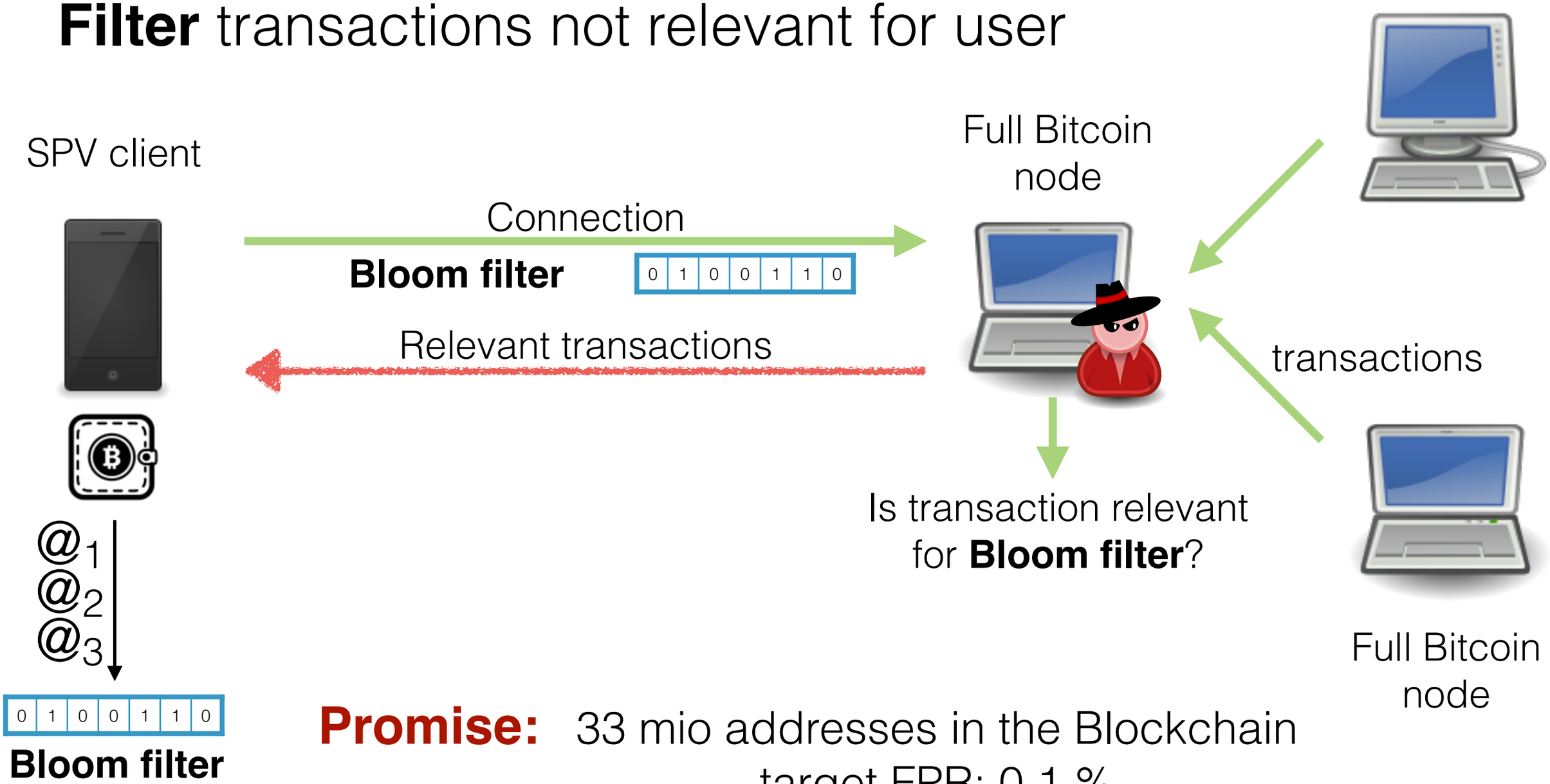
Simple Payment Verification (SPV)

Filter transactions not relevant for user



Simple Payment Verification (SPV)

Filter transactions not relevant for user



Promise: 33 mio addresses in the Blockchain
target FPR: 0.1 %

"User addresses hidden amongst 33 000" false positives

Main contributions

1. Given **one** Bloom filter, Bitcoin addresses partially linkable
   - Addresses linkable if < 20 addresses in wallet

2. Given **multiple** Bloom filter, addresses nearly always linkable

3. Propose a lightweight and efficient **countermeasure**
   - **Significantly enhances** the privacy offered by SPV clients
   - Requires **minimum modifications** to Bitcoin

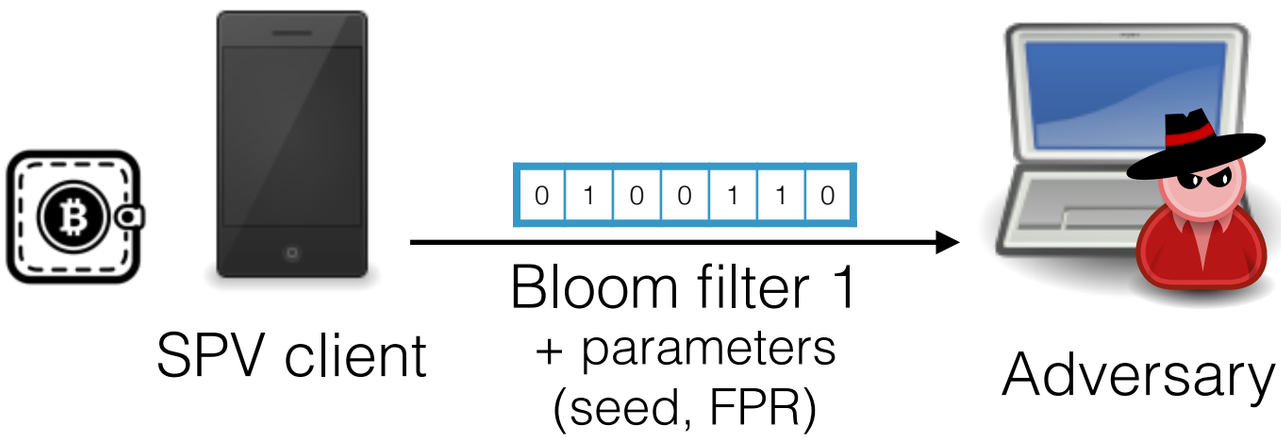
Model and Privacy measure

Blockchain

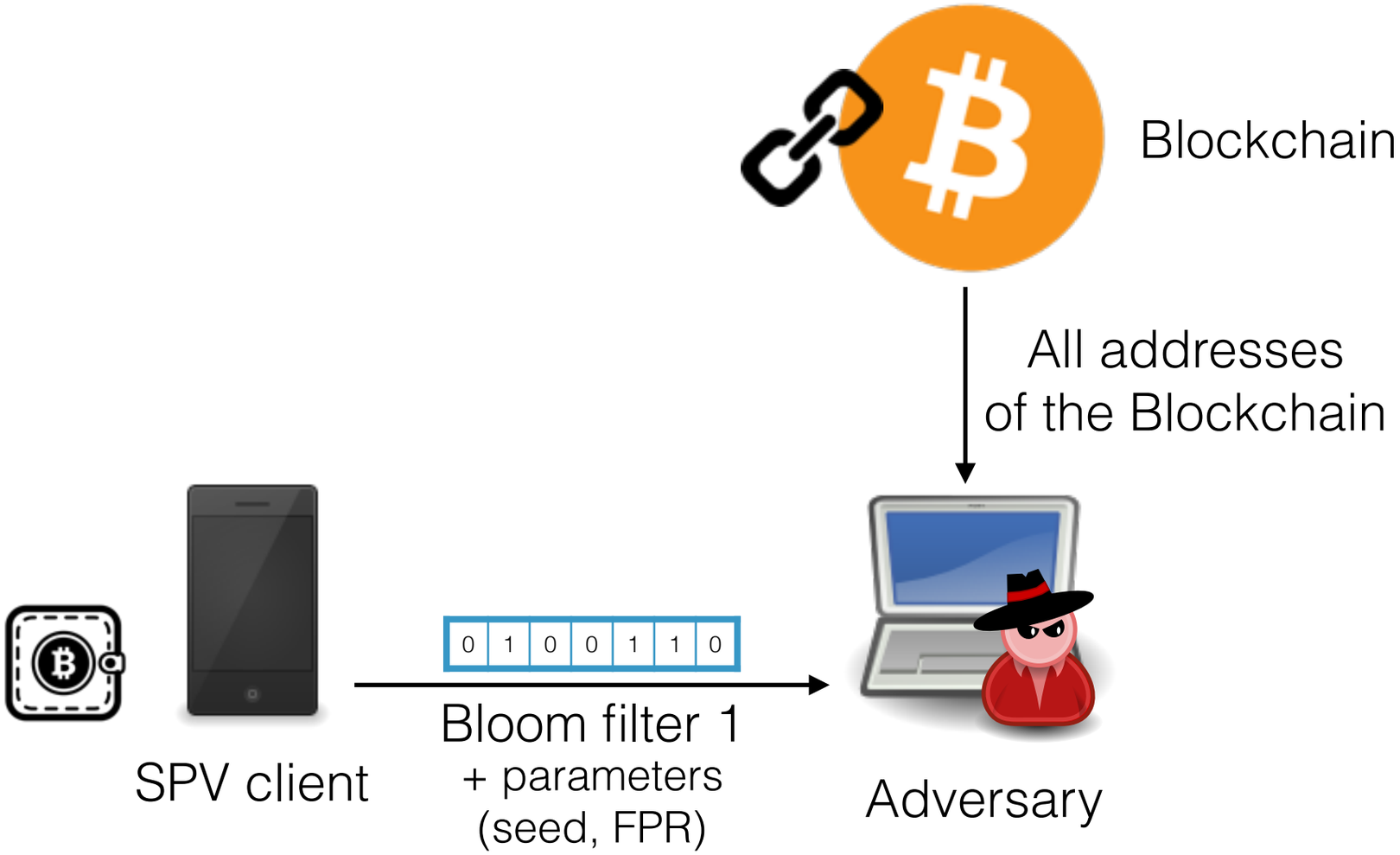
SPV client

Adversary

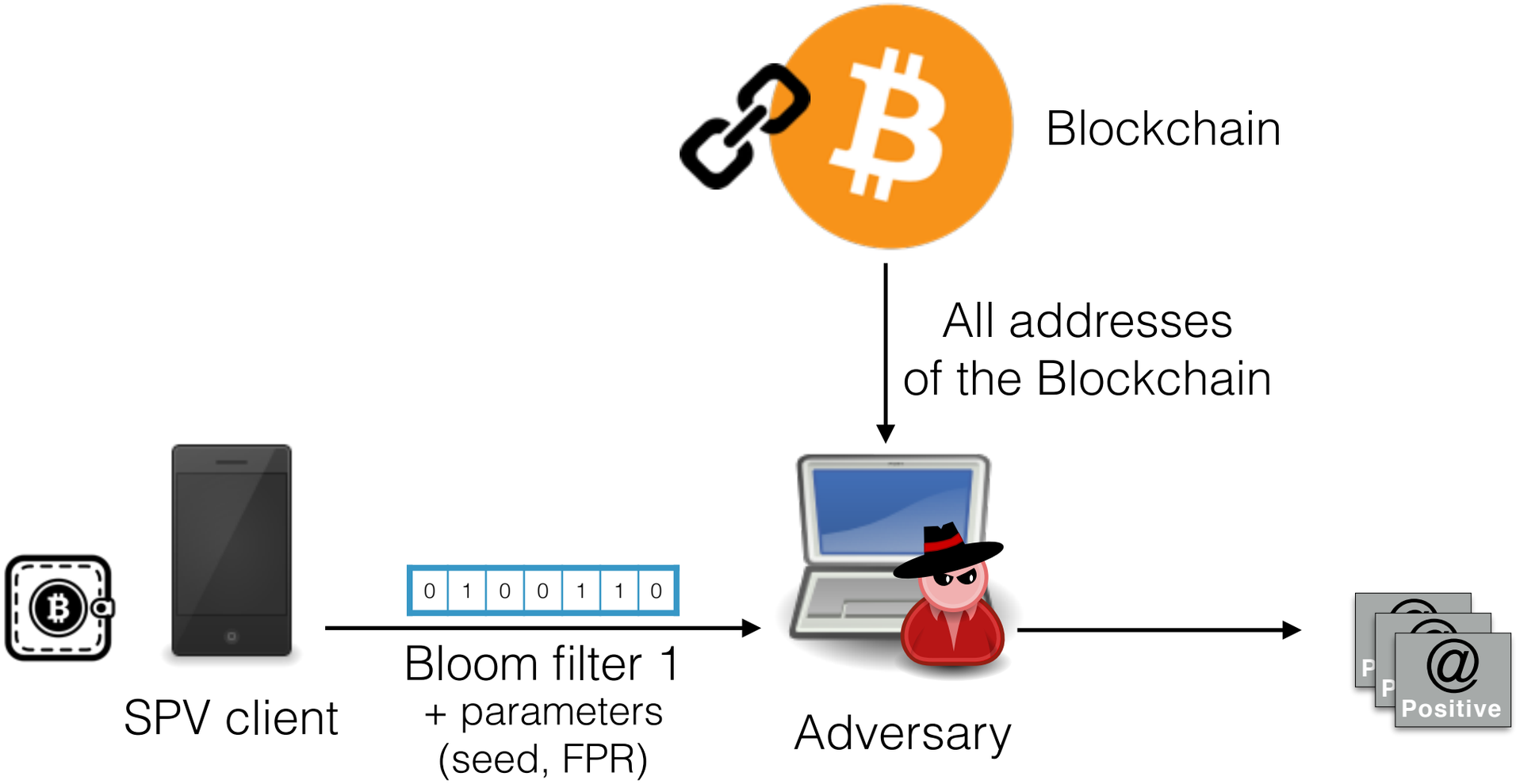
Model and Privacy measure



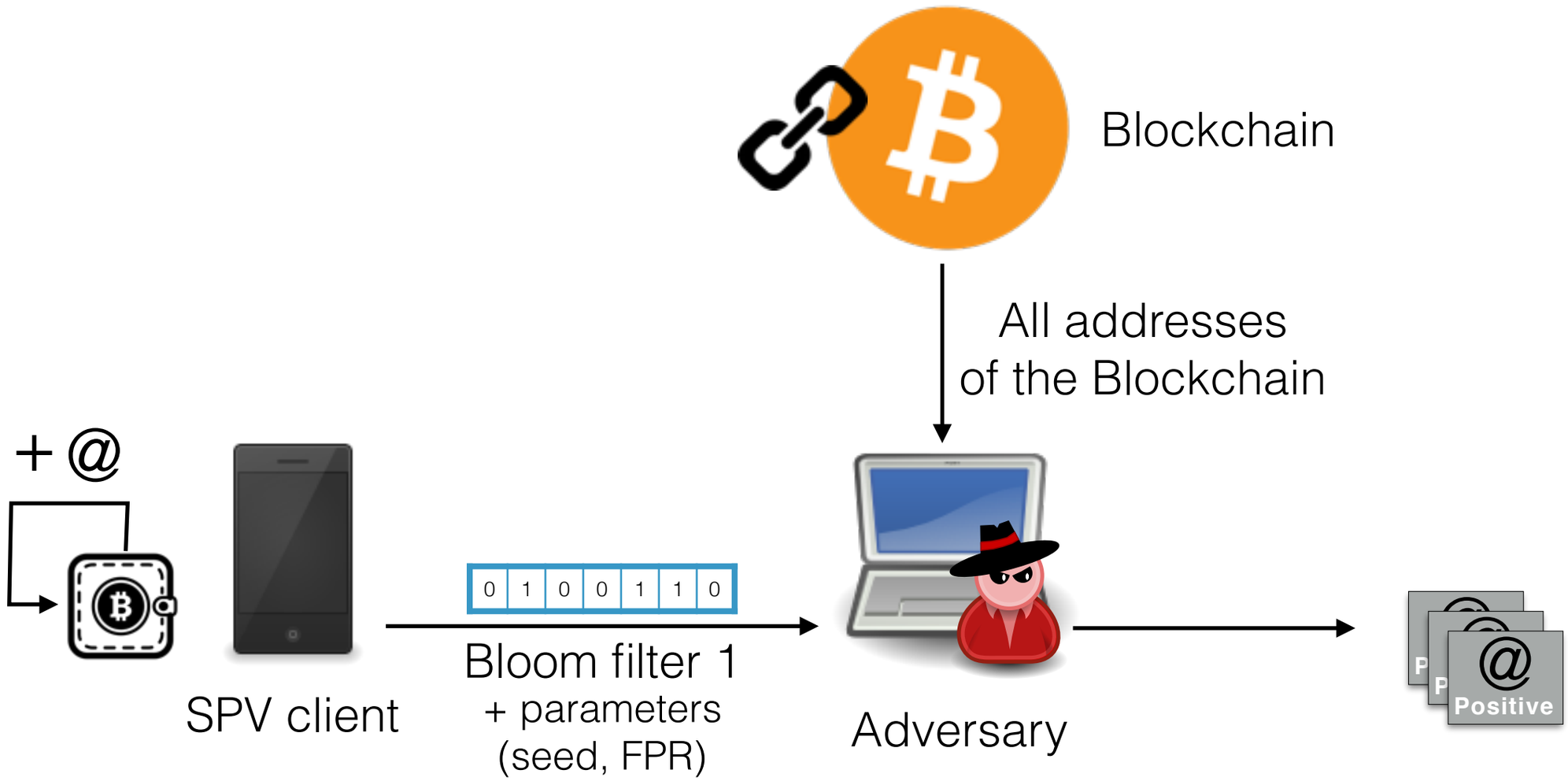
Model and Privacy measure



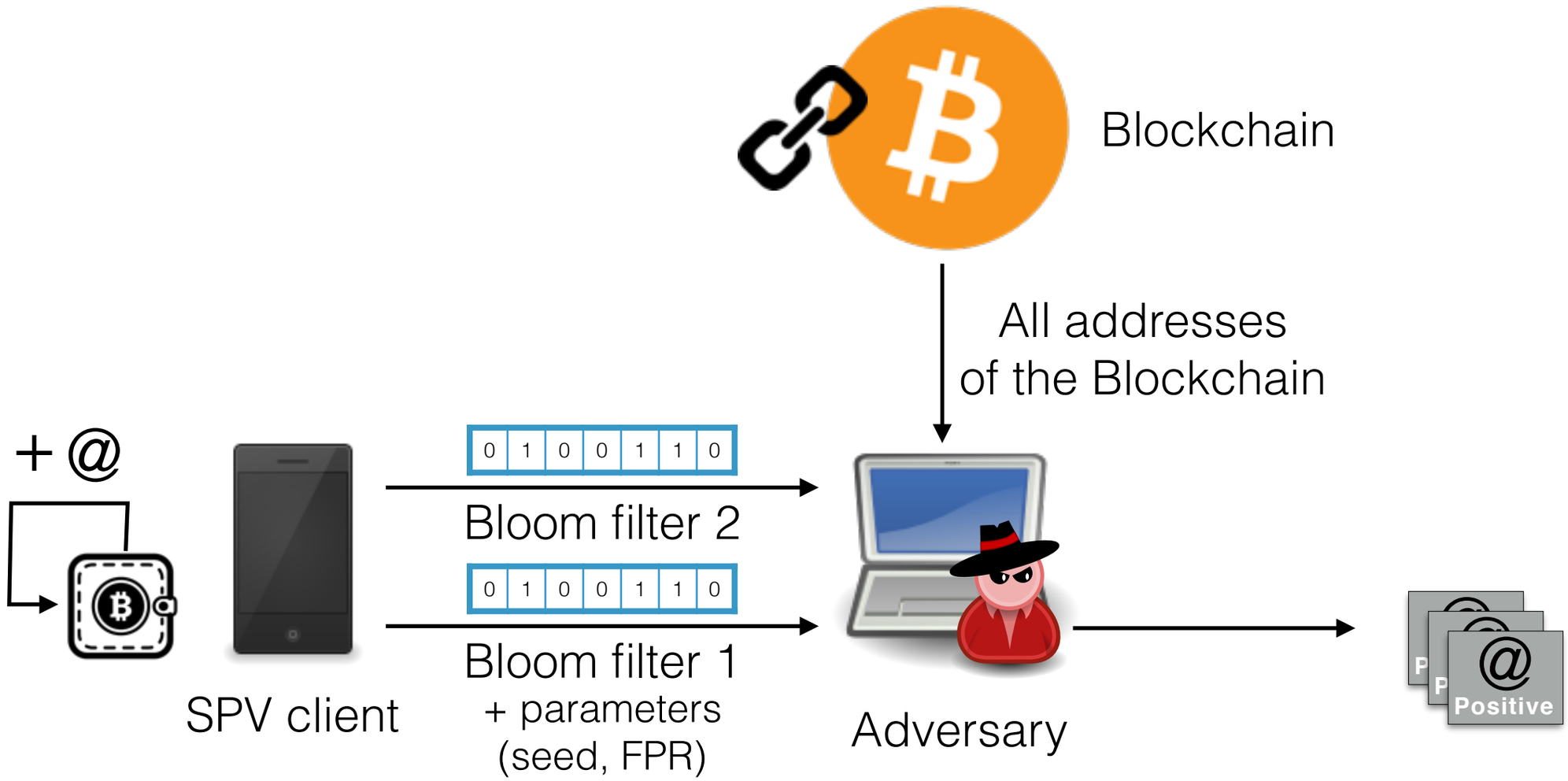
Model and Privacy measure



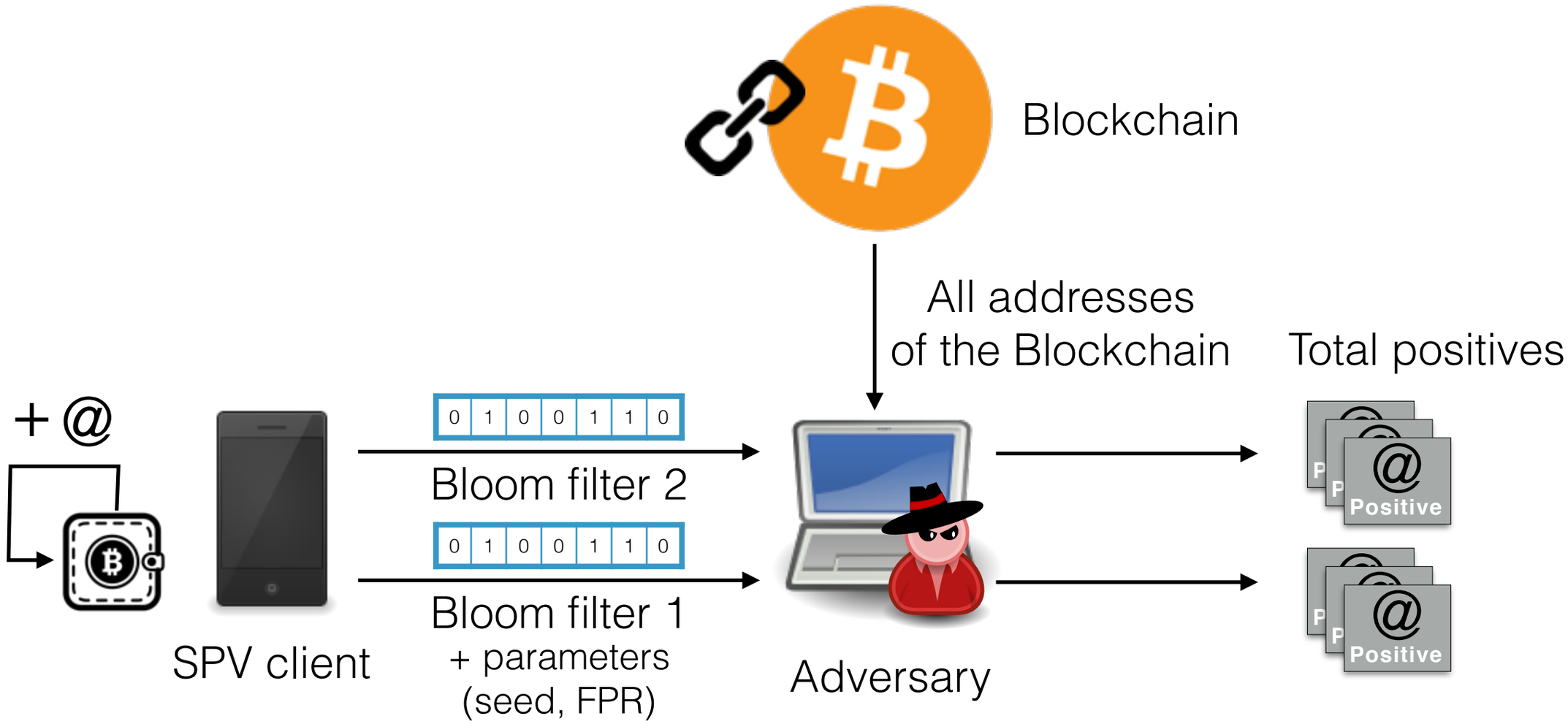
Model and Privacy measure



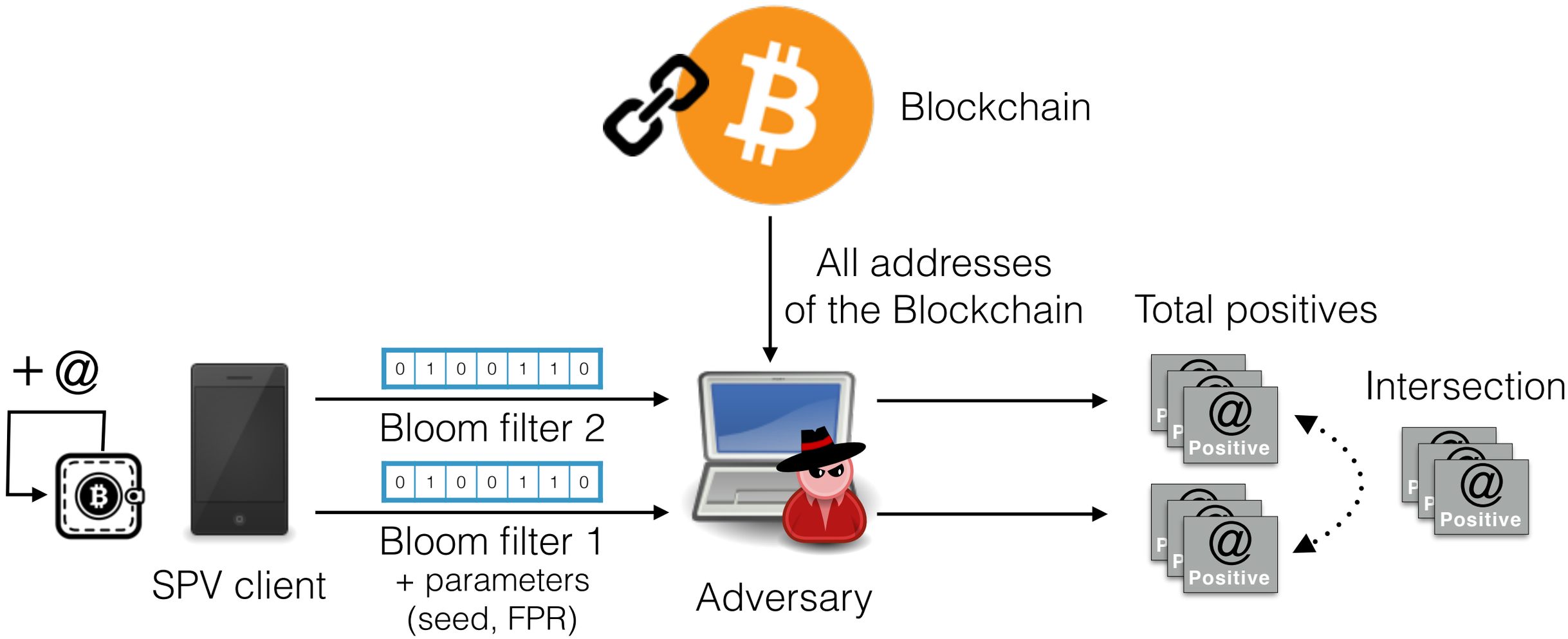
Model and Privacy measure



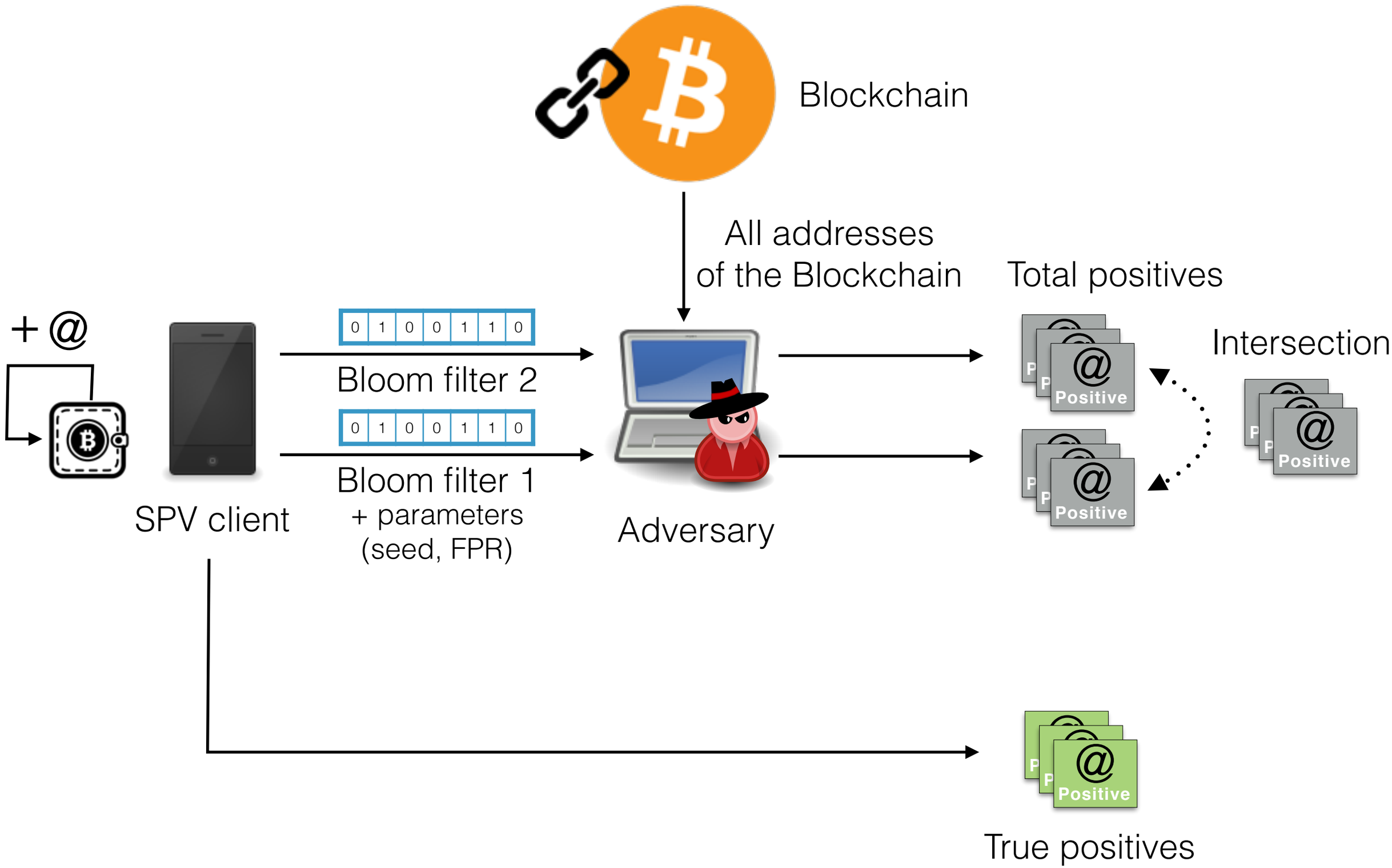
Model and Privacy measure



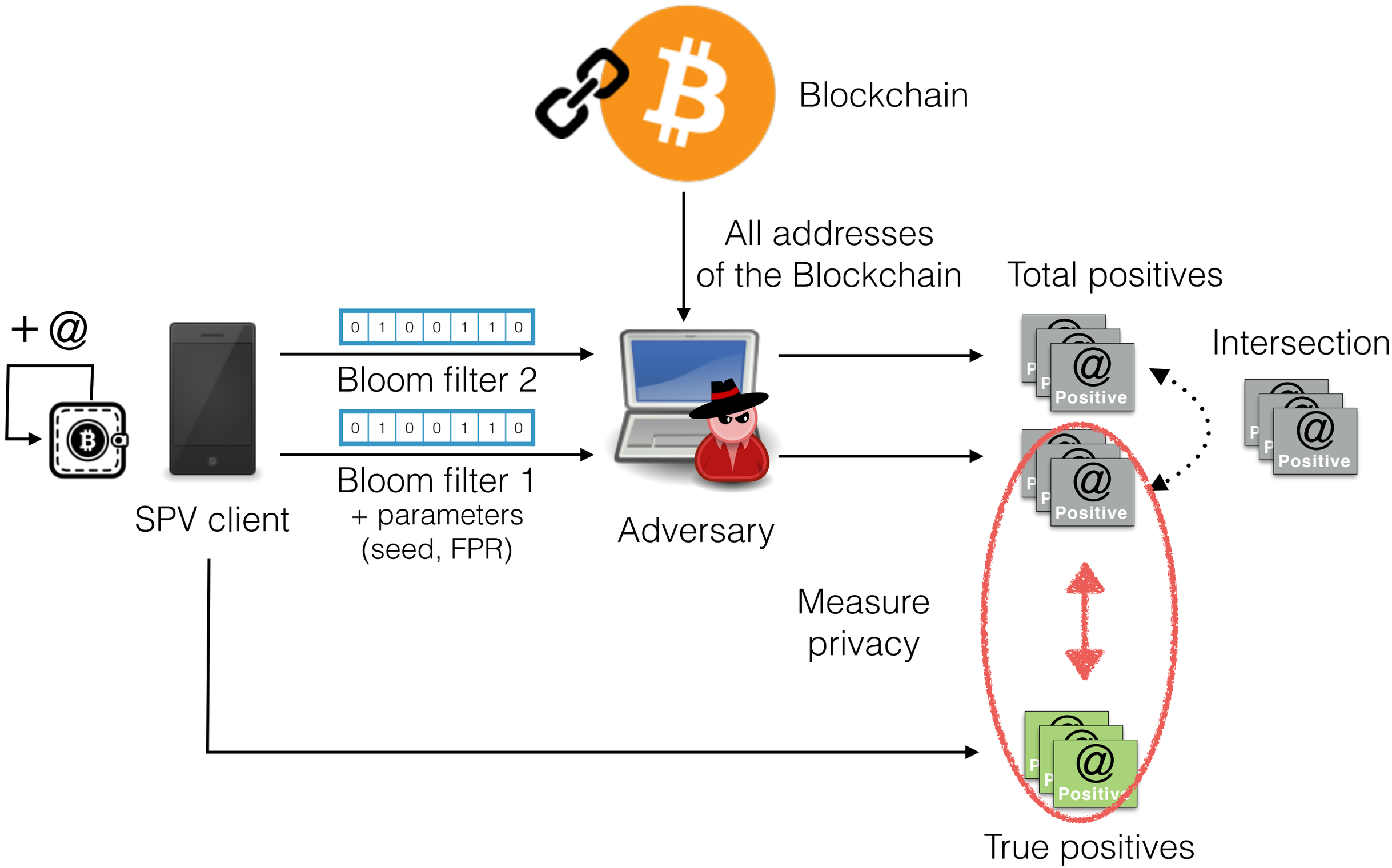
Model and Privacy measure



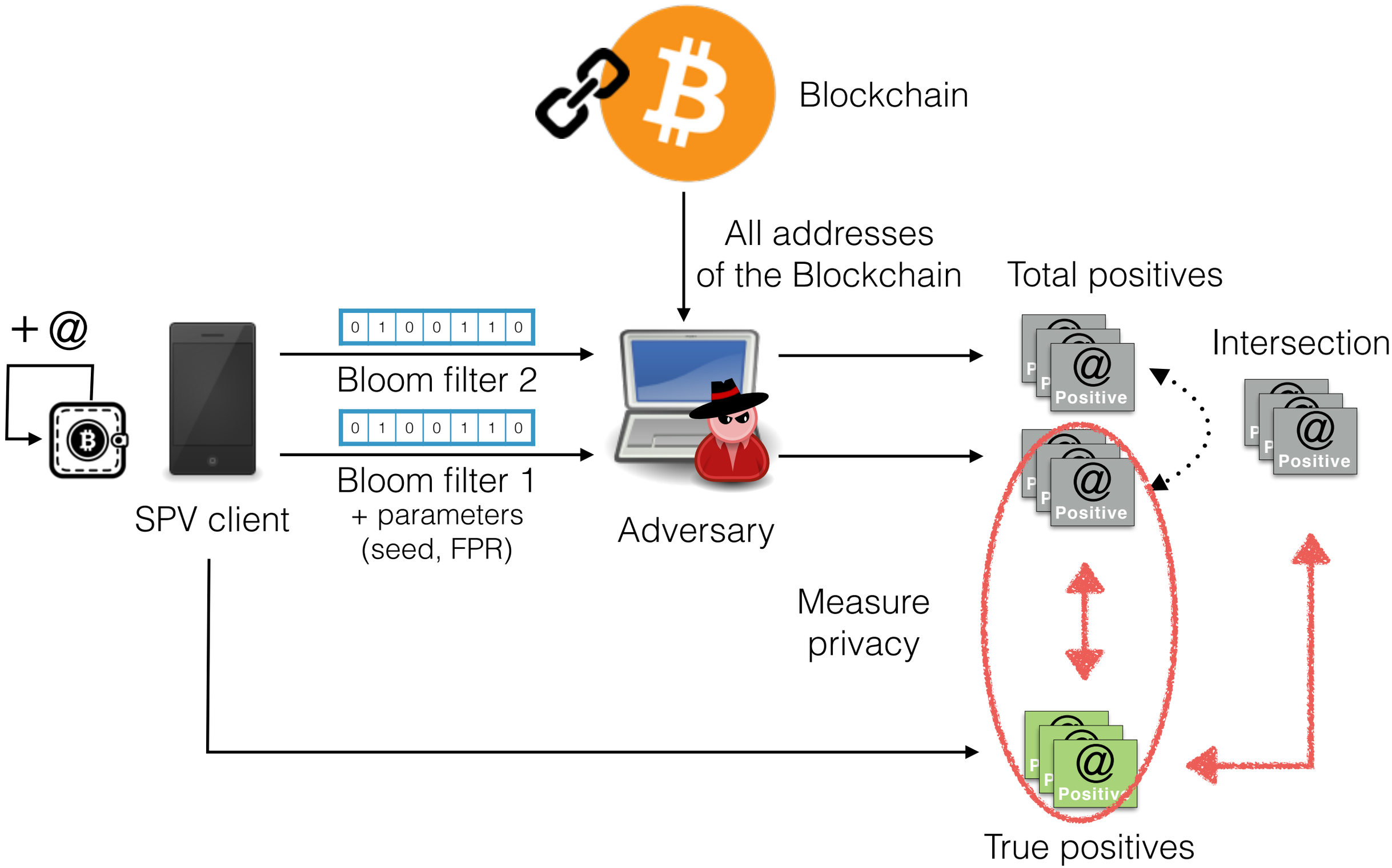
Model and Privacy measure



Model and Privacy measure

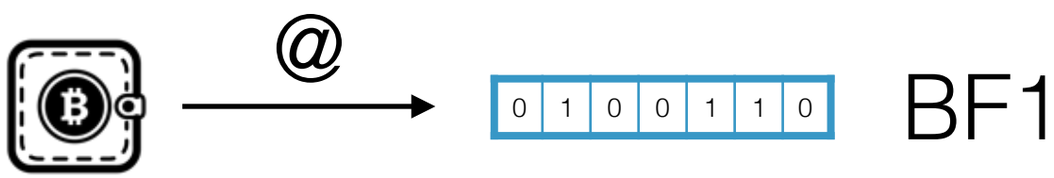


Model and Privacy measure



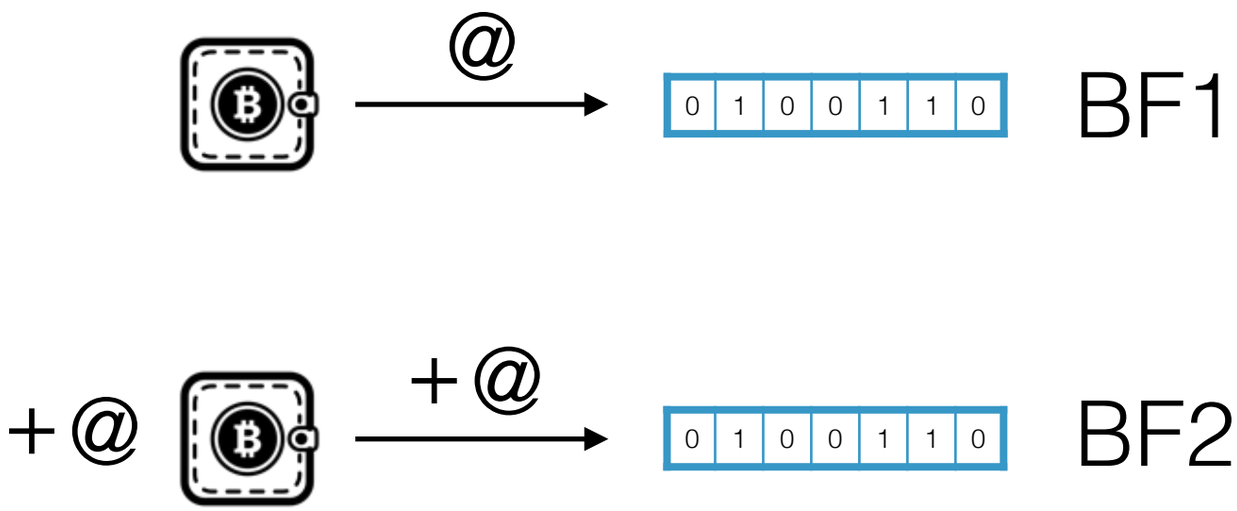
Stair stepping

Bloom filter designed for
- N **addresses**
- **target FPR** when N addresses inserted



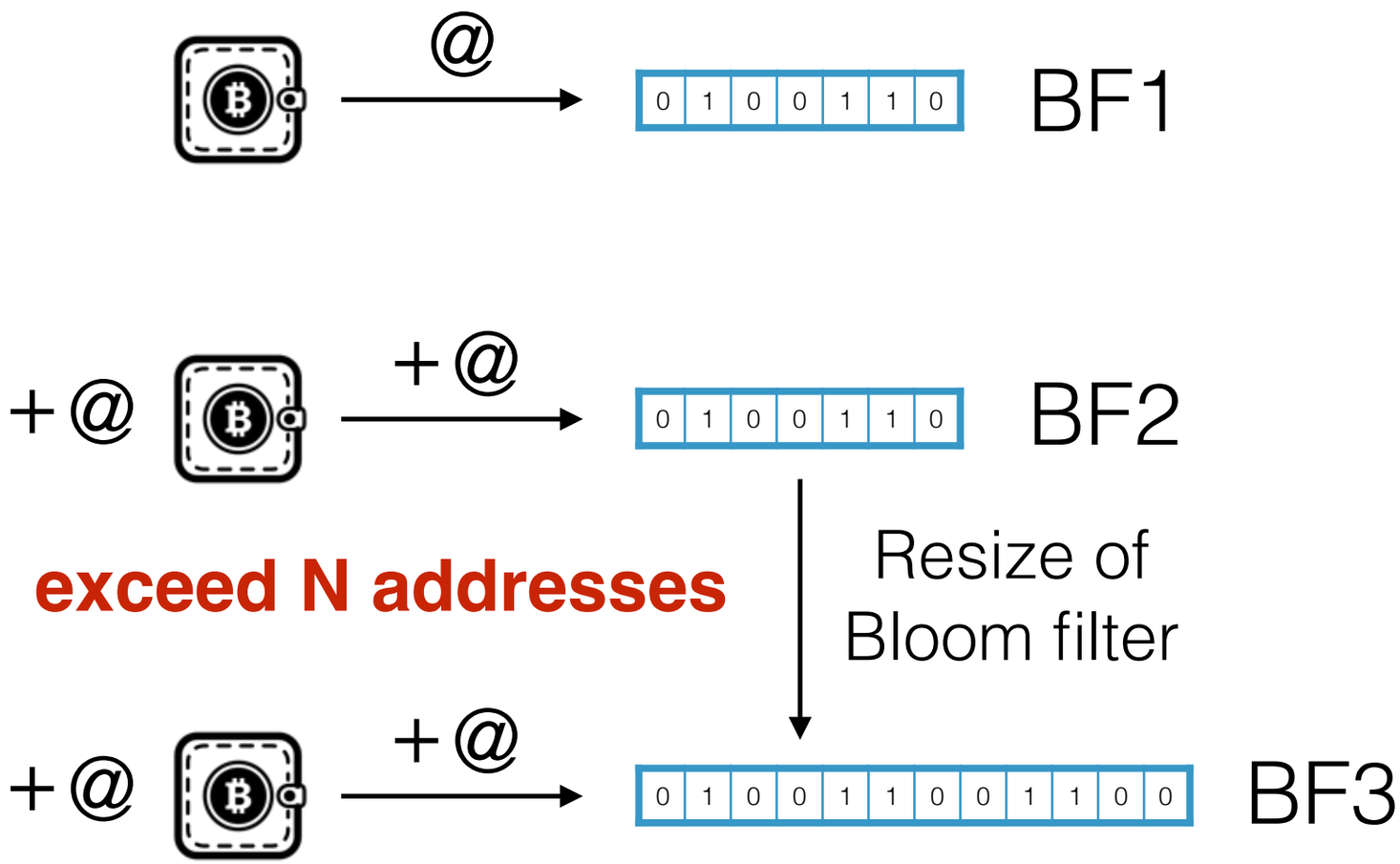
Stair stepping

Bloom filter designed for
- N **addresses**
- **target FPR** when N addresses inserted



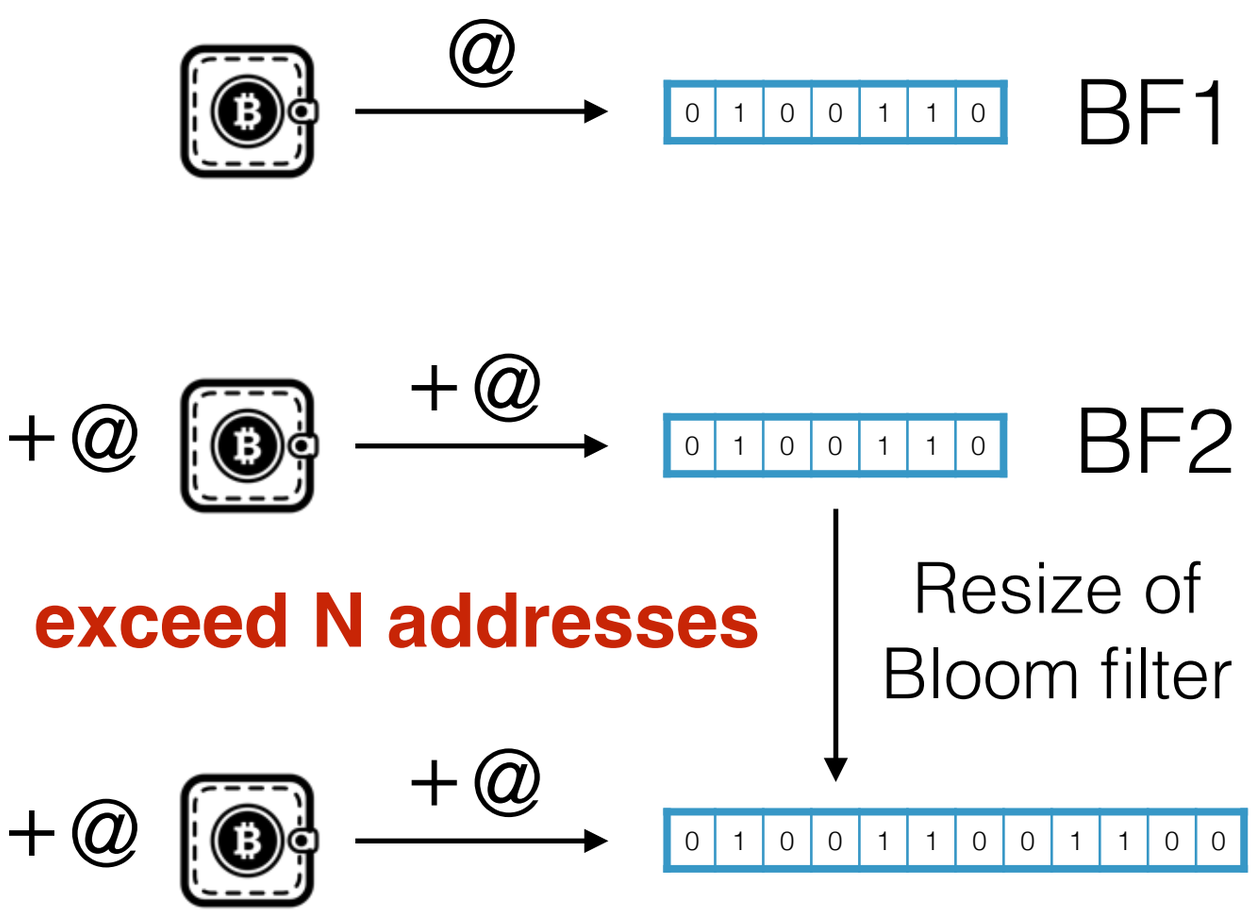
Stair stepping

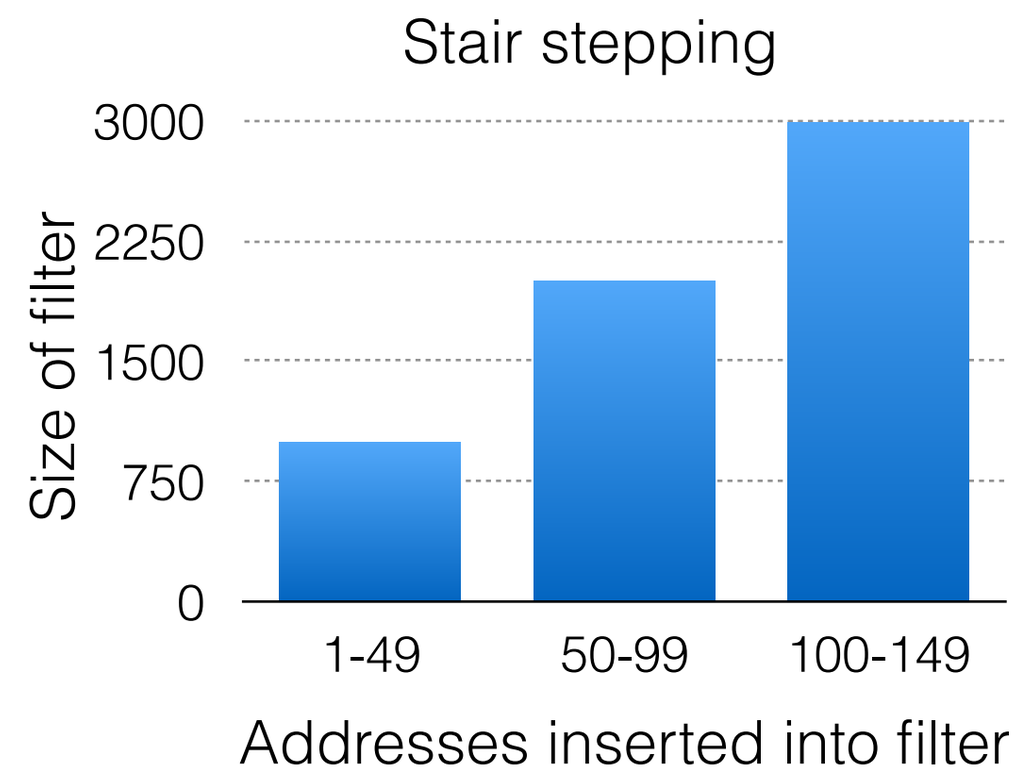
Bloom filter designed for
- N **addresses**
- **target FPR** when N addresses inserted



Stair stepping

Bloom filter designed for
- N **addresses**
- **target FPR** when N addresses inserted

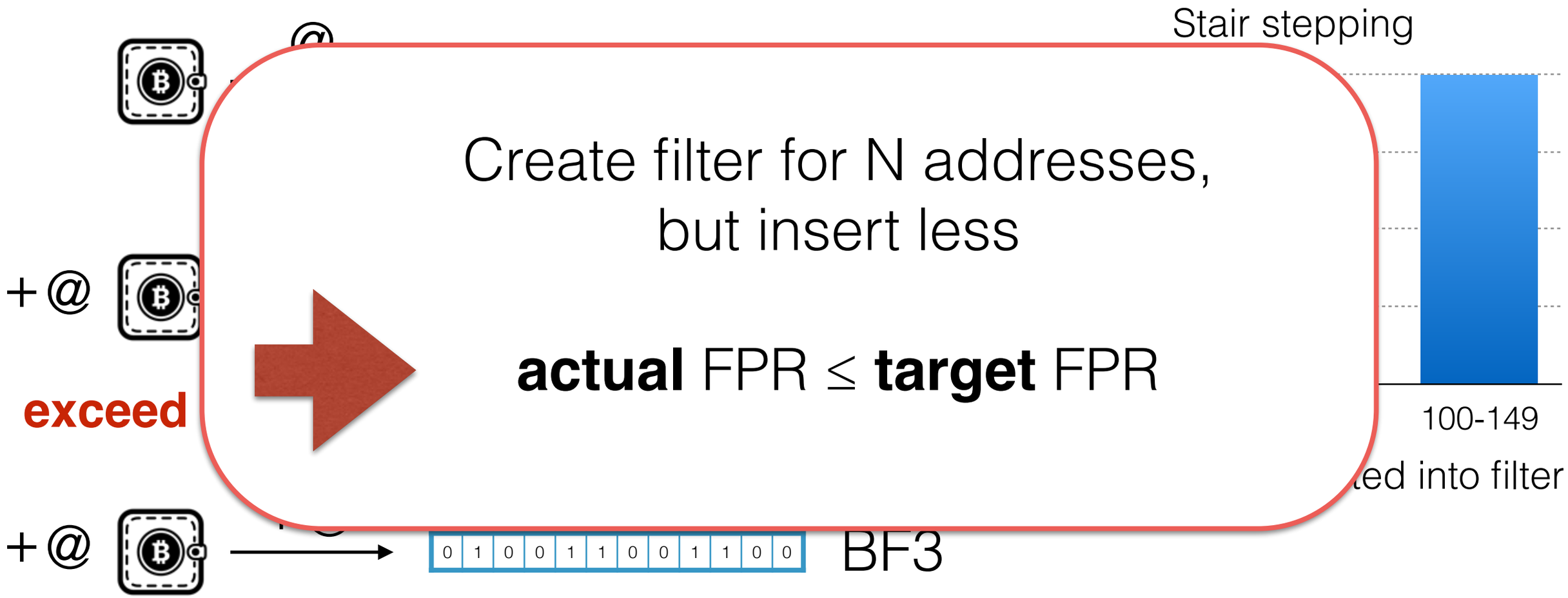




Rationale: **avoid** filters with different sizes

Stair stepping

Bloom filter designed for
- N **addresses**
- **target FPR** when N addresses inserted

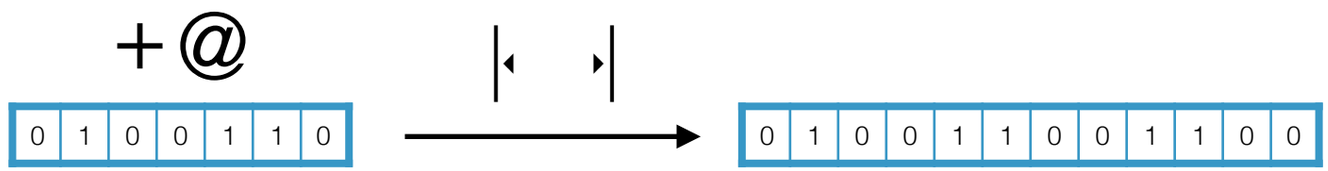


Create filter for N addresses, but insert less

actual FPR ≤ **target** FPR

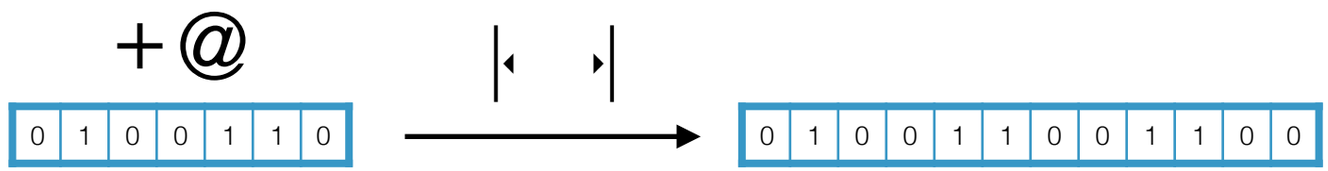
Rationale: **avoid** filters with different sizes

Resizing



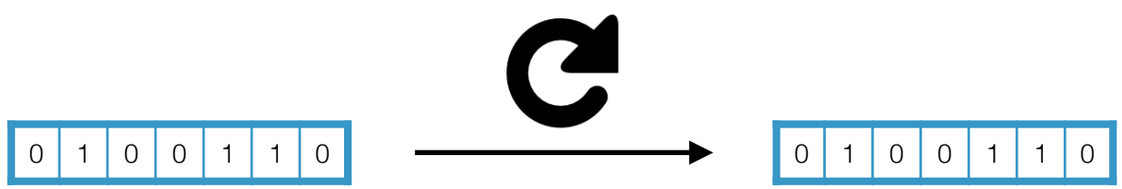
- Hash functions adapted to fill space of new Bloom filter
- **Consequence**: New filter yields **different** false positives

Resizing



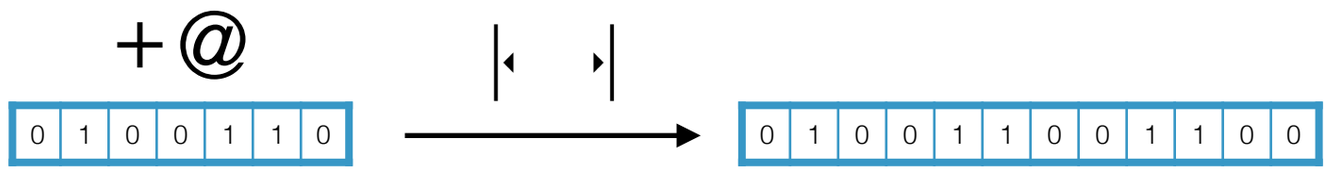
- Hash functions adapted to fill space of new Bloom filter
- **Consequence**: New filter yields **different** false positives

Restarting



- Fresh seed value for hash functions of new Bloom filter
- **Consequence**: New filter yields **different** false positives

Resizing



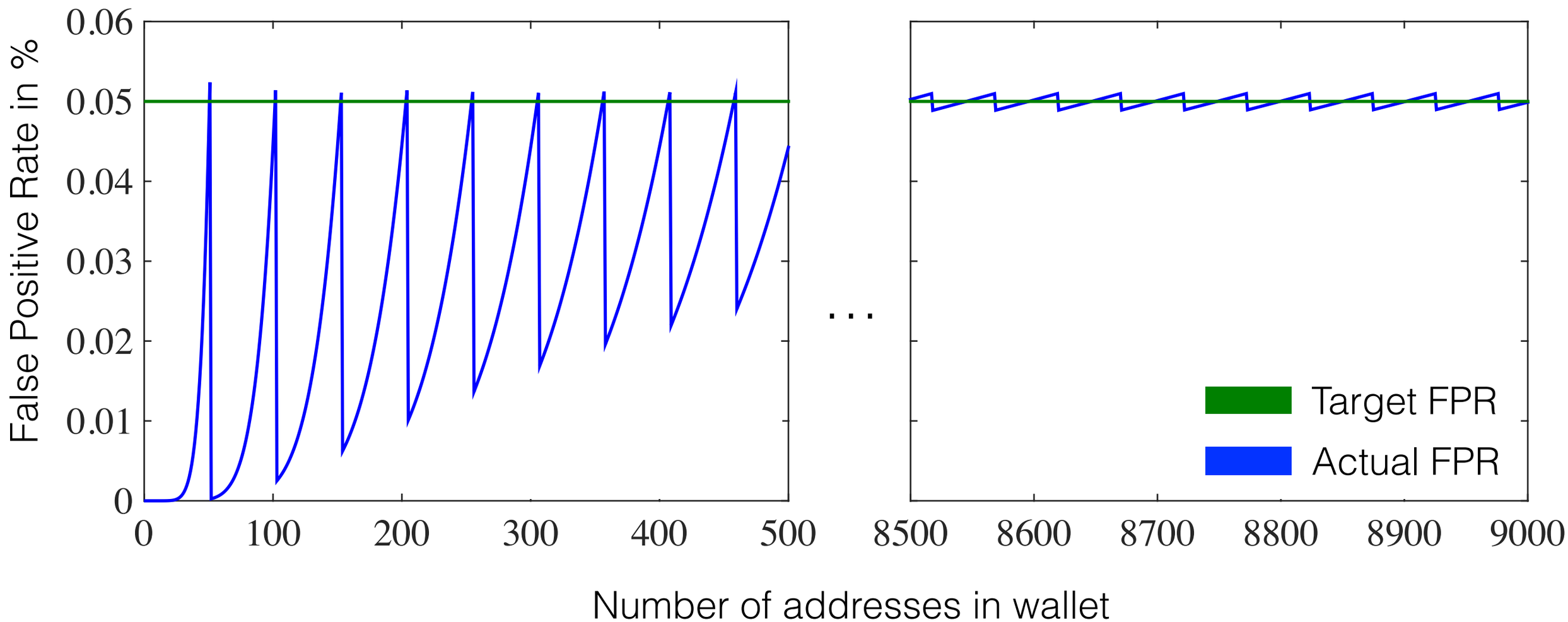
- I

Summary of current SPV design choices

1. Stair stepping → actual FPR ≤ target FPR
2. Resizing → different False Positives
3. Restarting → different False Positives

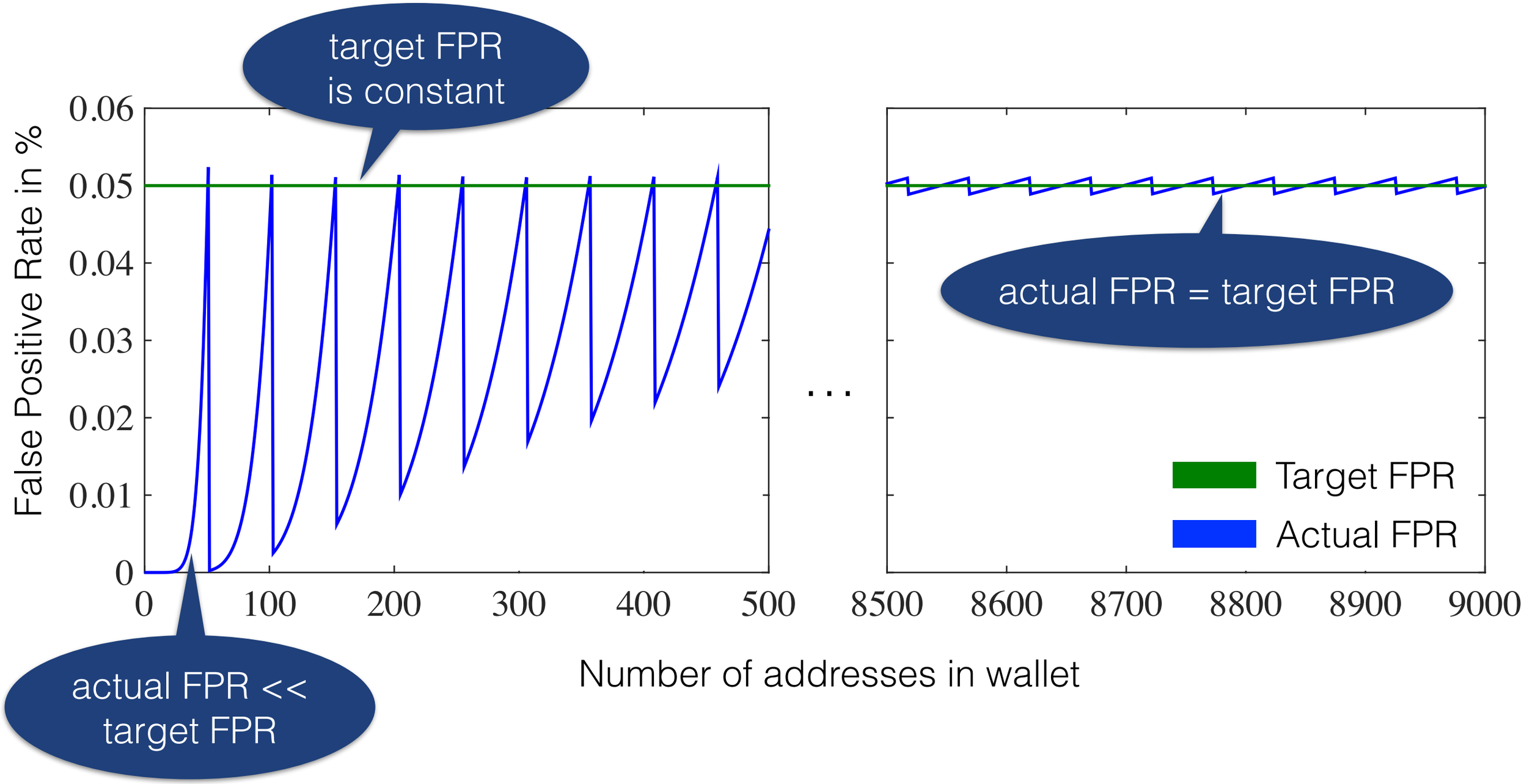
- **Consequence**: New filter yields **different** false positives

Stair stepping - Actual FPR vs. Target FPR



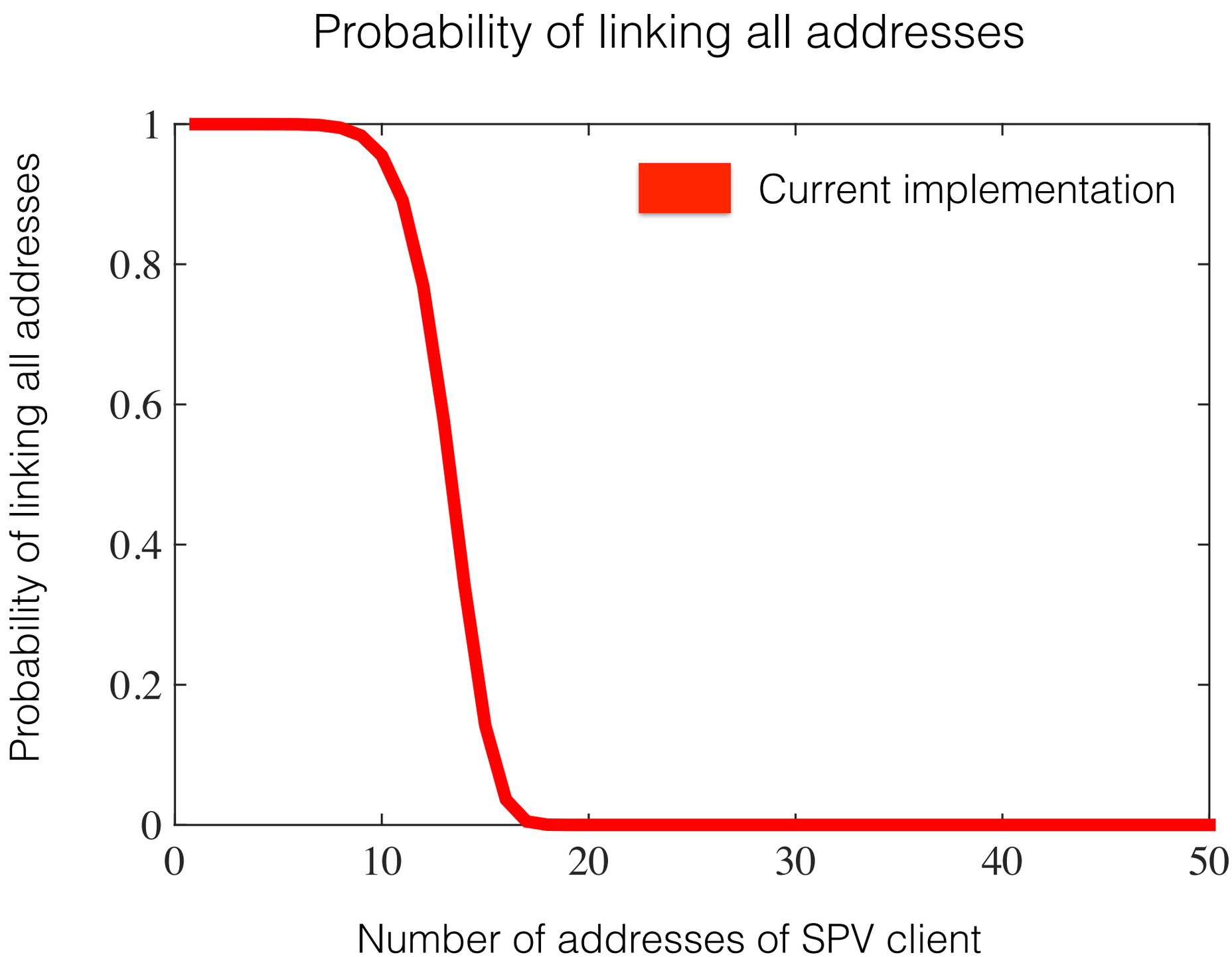
ETH zürich

Stair stepping - Actual FPR vs. Target FPR



ETH zürich

One Bloom filter



ETH zürich

Multiple Bloom filters

Filter 1

0	1	0	0	1	1	0

$@_1$ $@_2$ $@_3$

ETH zürich

Multiple Bloom filters

Filter 1

0	1	0	0	1	1	0

Filter 2

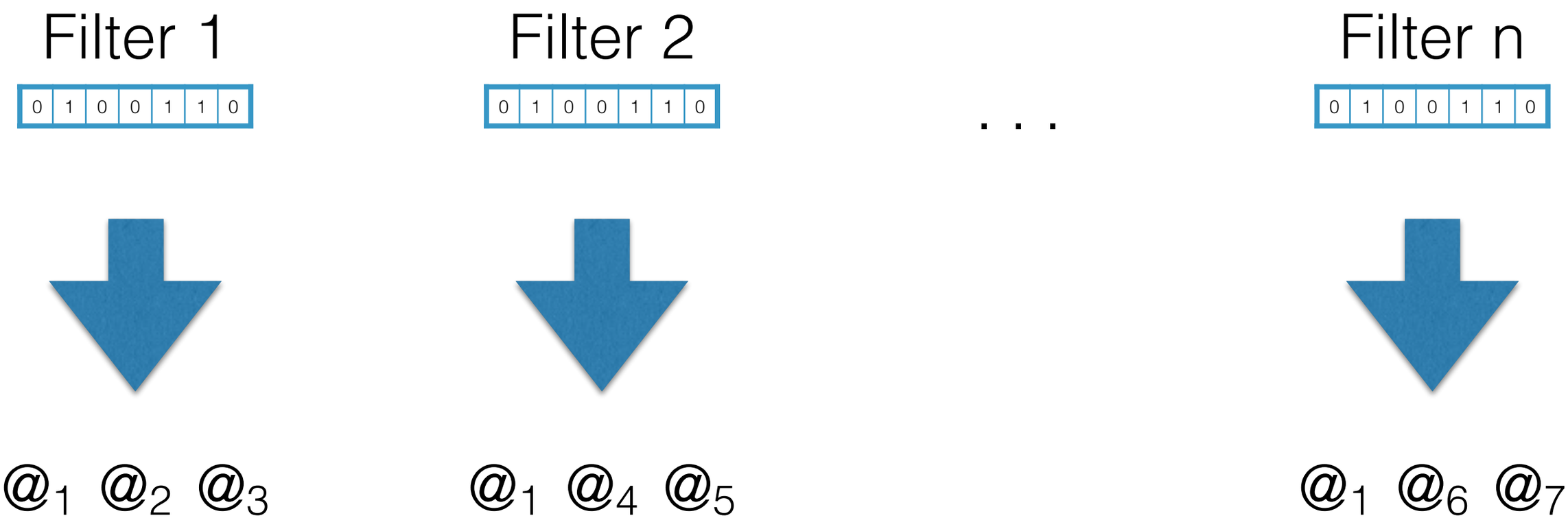
0	1	0	0	1	1	0

$@_1$ $@_2$ $@_3$

$@_1$ $@_4$ $@_5$

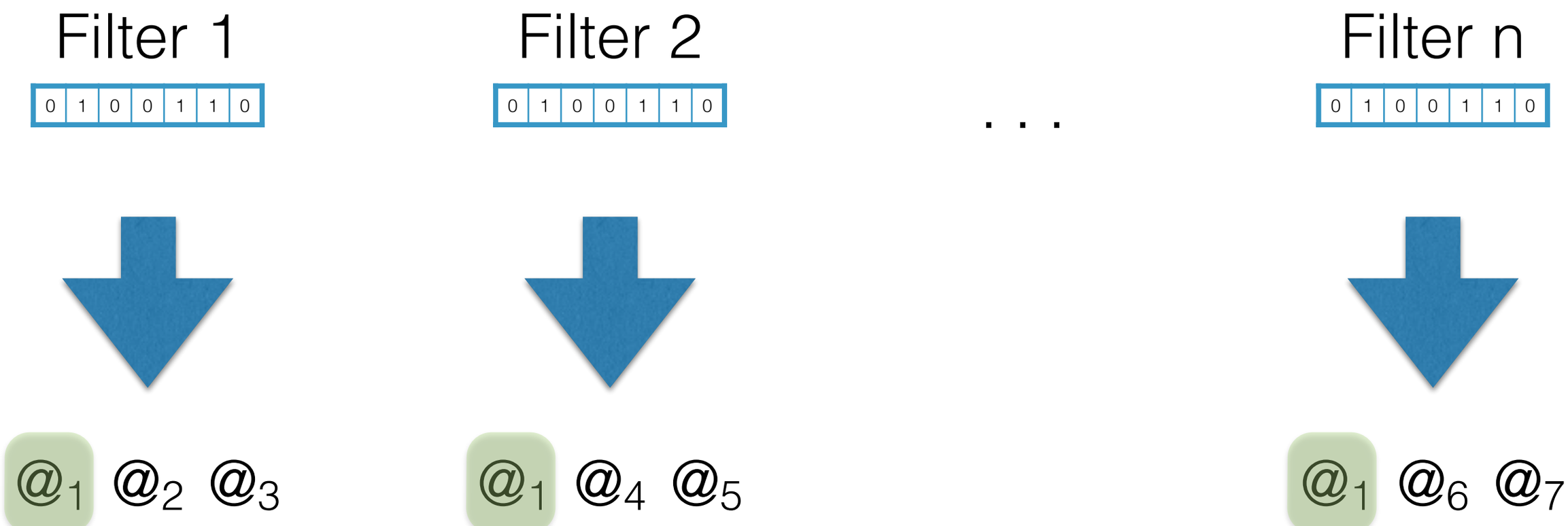
ETH zürich

Multiple Bloom filters



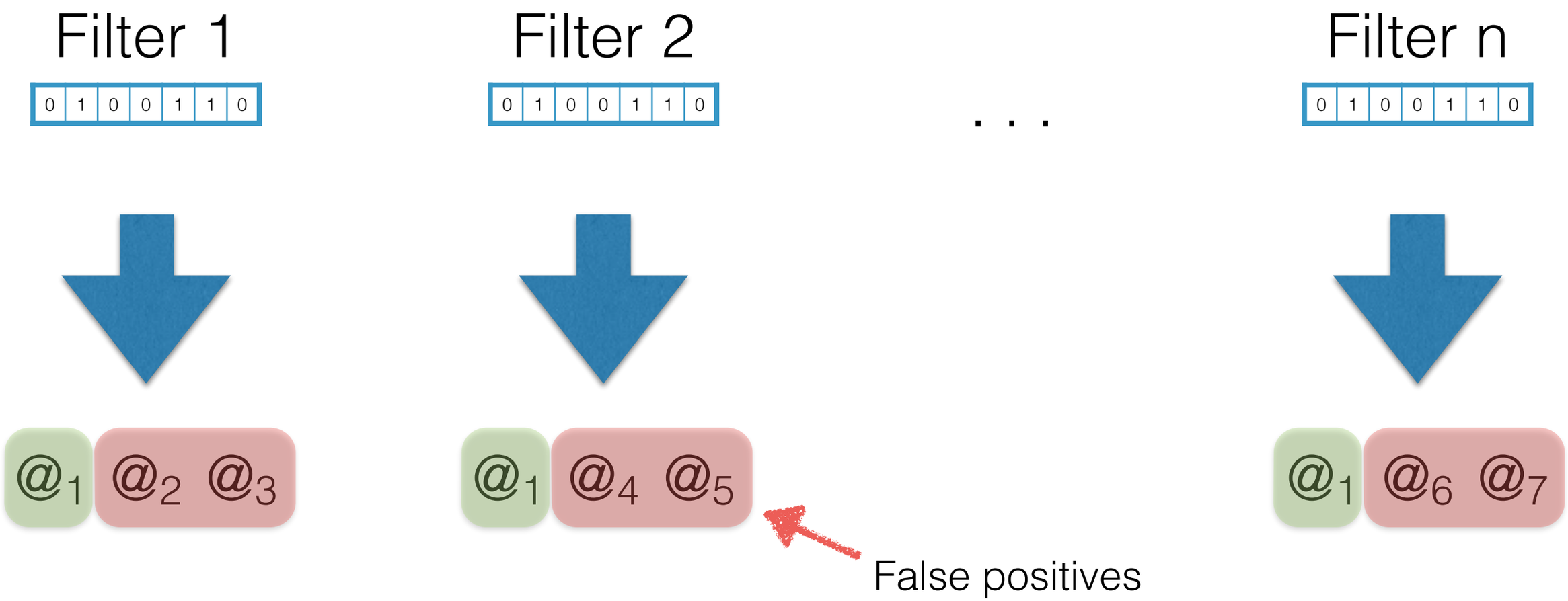
ETH zürich

Multiple Bloom filters



ETH zürich

Multiple Bloom filters



ETH zürich

Experiment 1 - No resize

Exp.	Client	Seed	Size
No resize	**Same**	**Same**	**Same**
Resize	Same	Same	Different
Restart	Same	Different	Same
> 2 filter	Same	Different	Different

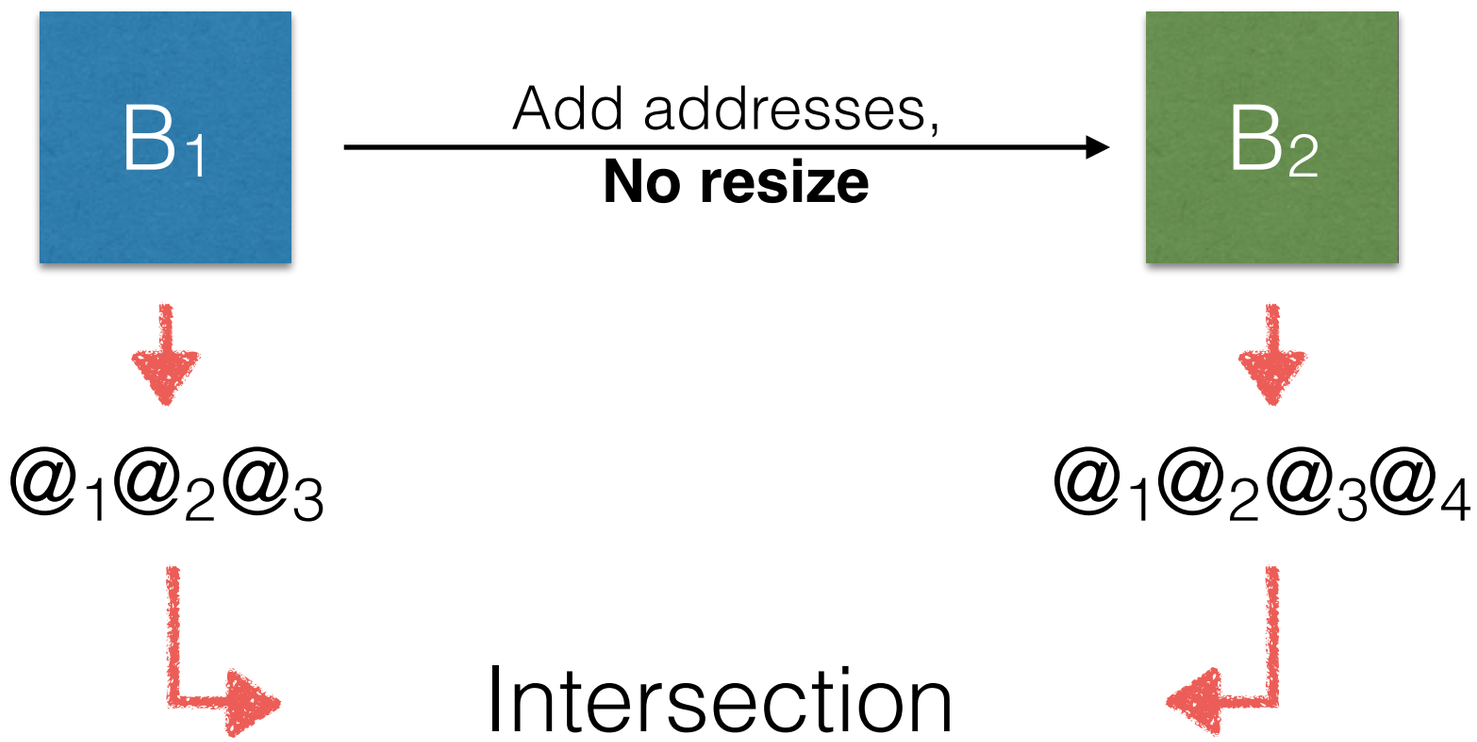
ETH zürich

Experiment 1 - No resize

Exp.	Client	Seed	Size
No resize	**Same**	**Same**	**Same**
Resize	Same	Same	Different
Restart	Same	Different	Same
> 2 filter	Same	Different	Different

ETH zürich

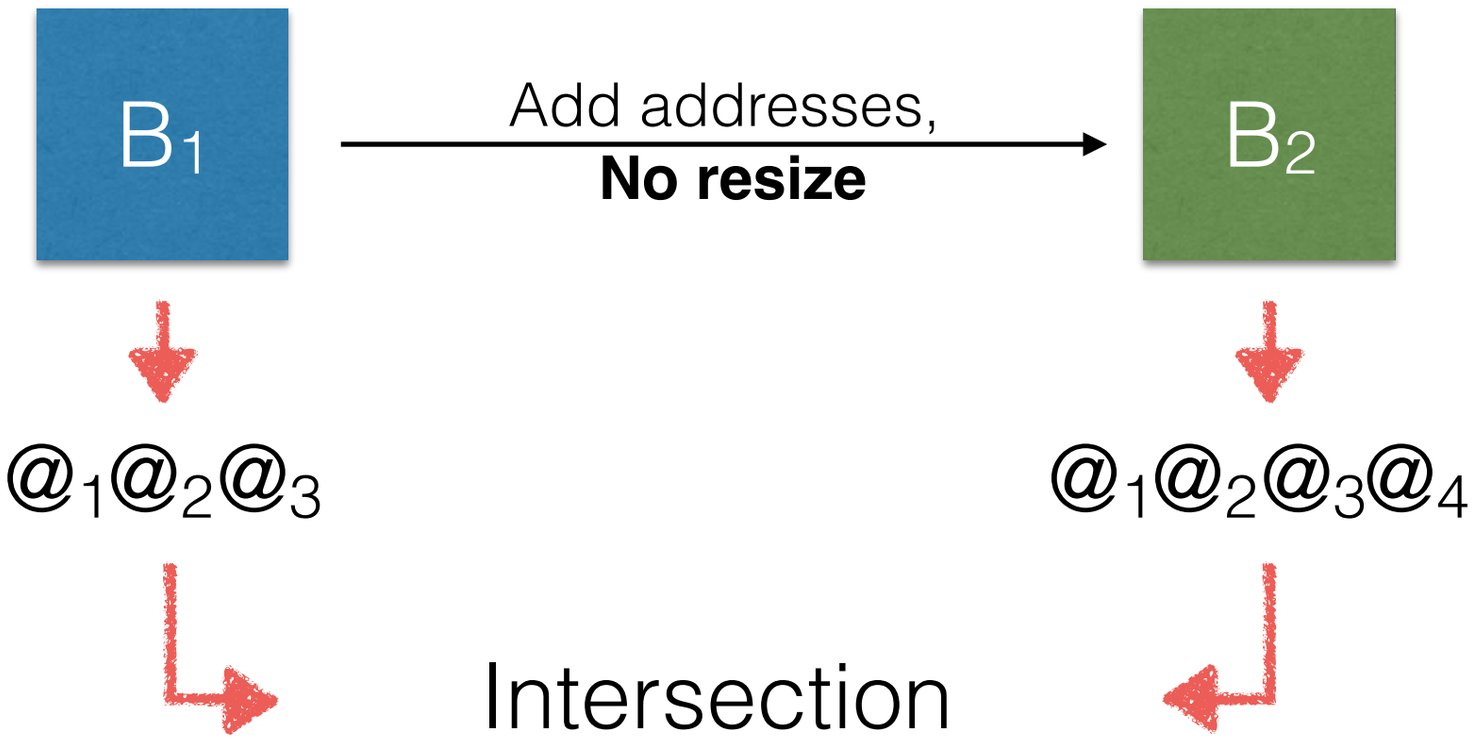
Experiment 1 - No resize



Exp.	Client	Seed	Size
<u>No resize</u>	**Same**	**Same**	**Same**
Resize	Same	Same	Different
Restart	Same	Different	Same
> 2 filter	Same	Different	Different

ETH zürich

Experiment 1 - No resize



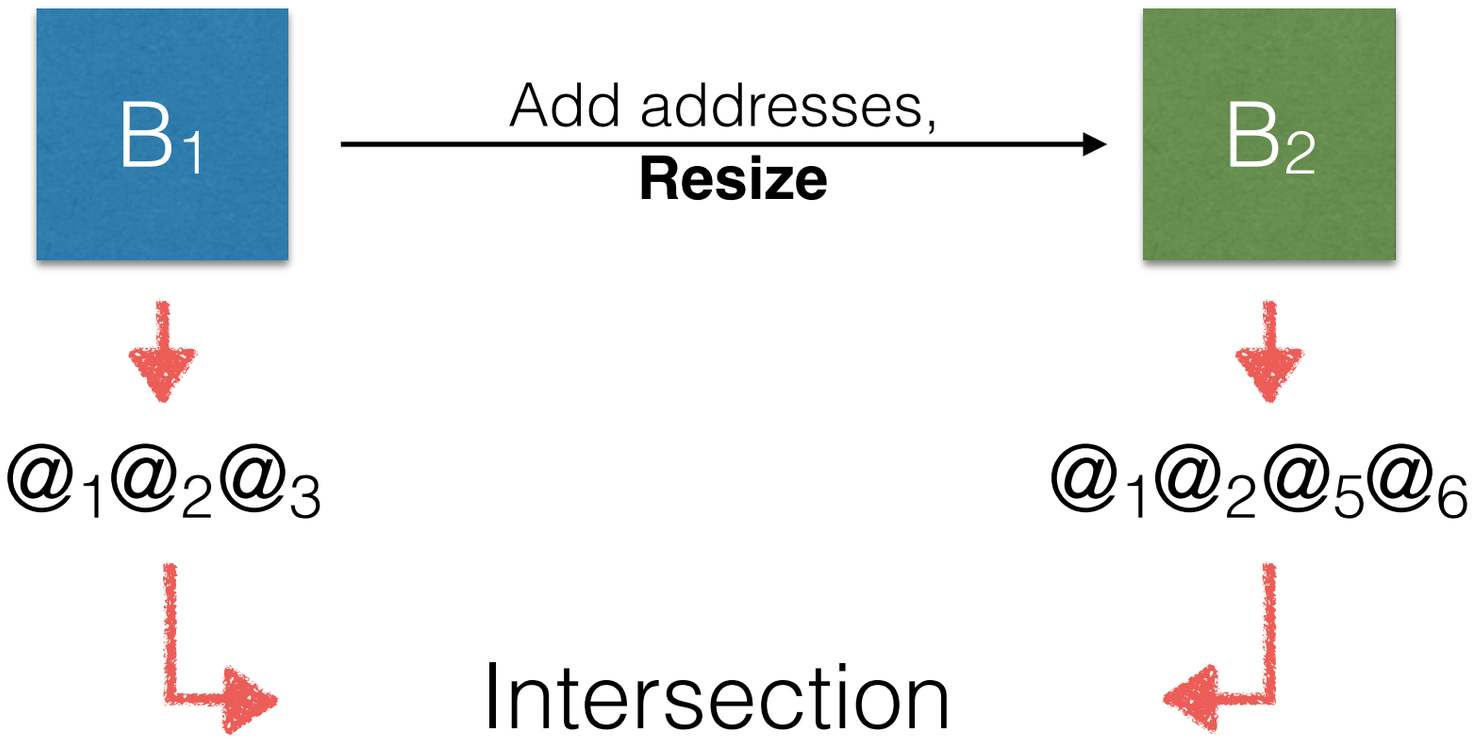
Exp.	Client	Seed	Size
No resize	**Same**	**Same**	**Same**
Resize	Same	Same	Different
Restart	Same	Different	Same
> 2 filter	Same	Different	Different

Results

- B_2 yields additional positives compared to B_1
- The adversary does not learn a lot

ETH zürich

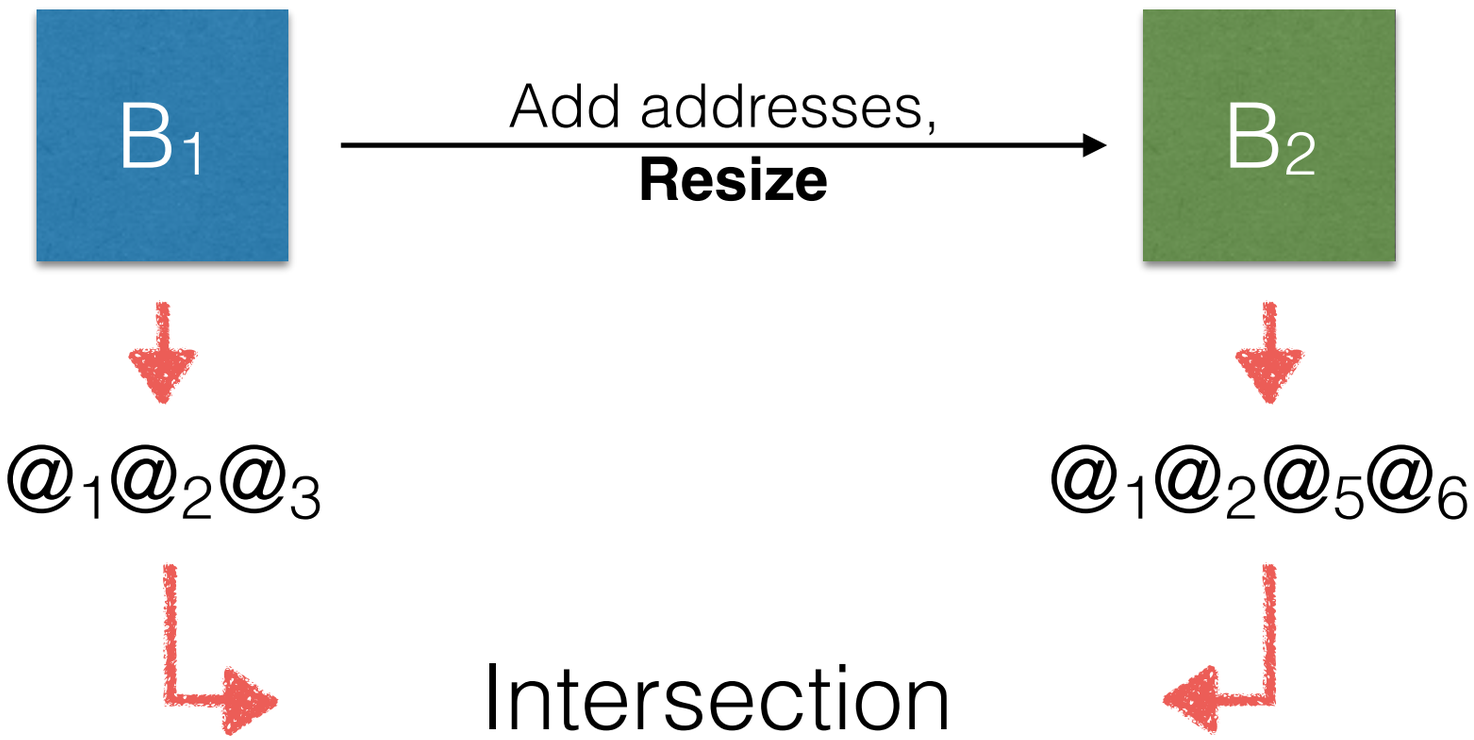
Experiment 2 - Resize



Exp.	Client	Seed	Size
No resize	Same	Same	Same
Resize	**Same**	**Same**	**Different**
Restart	Same	Different	Same
> 2 filter	Same	Different	Different

ETH zürich

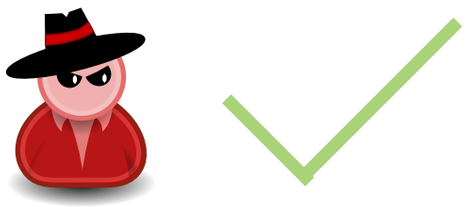
Experiment 2 - Resize



Exp.	Client	Seed	Size
No resize	Same	Same	Same
Resize	**Same**	**Same**	**Different**
Restart	Same	Different	Same
> 2 filter	Same	Different	Different

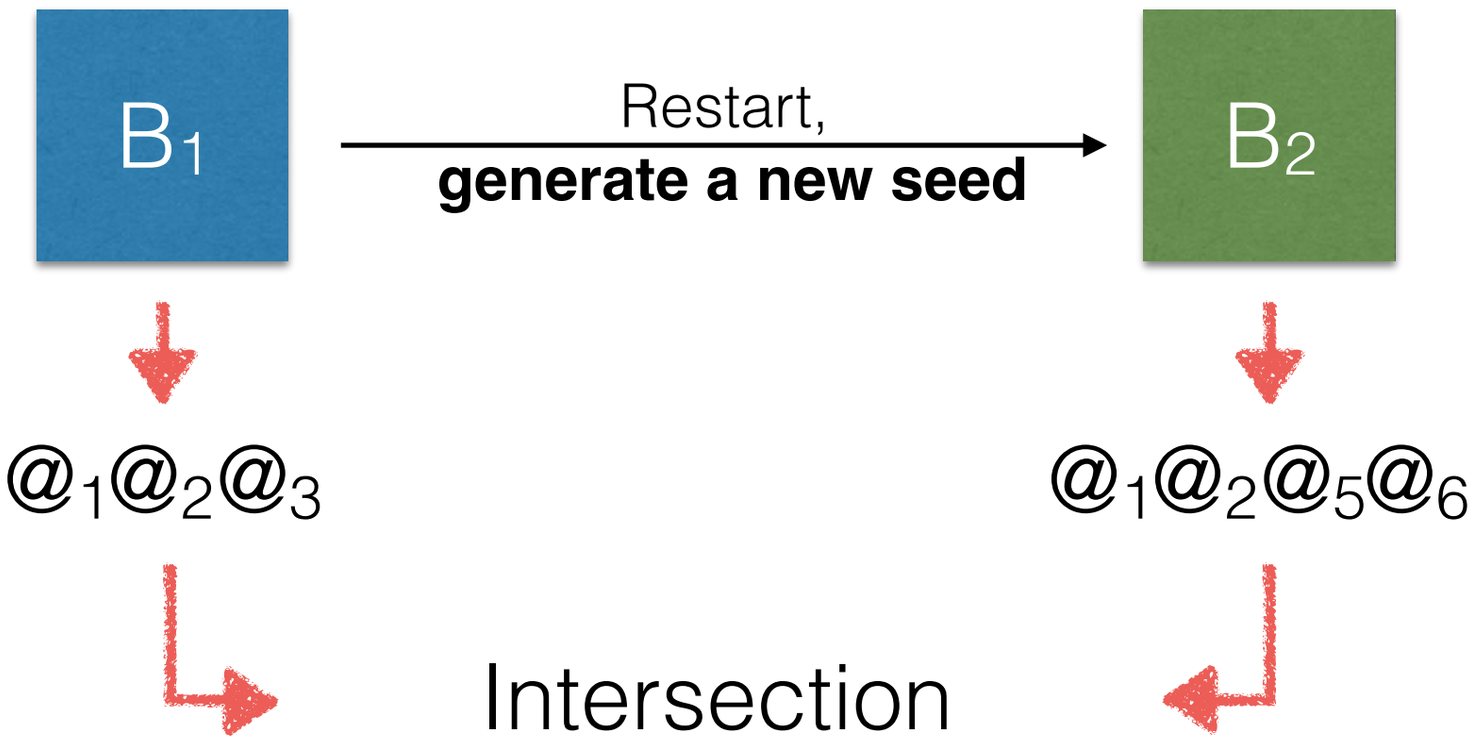
Results

- Yield mostly different positives
- Can be used for intersection attack



ETH zürich

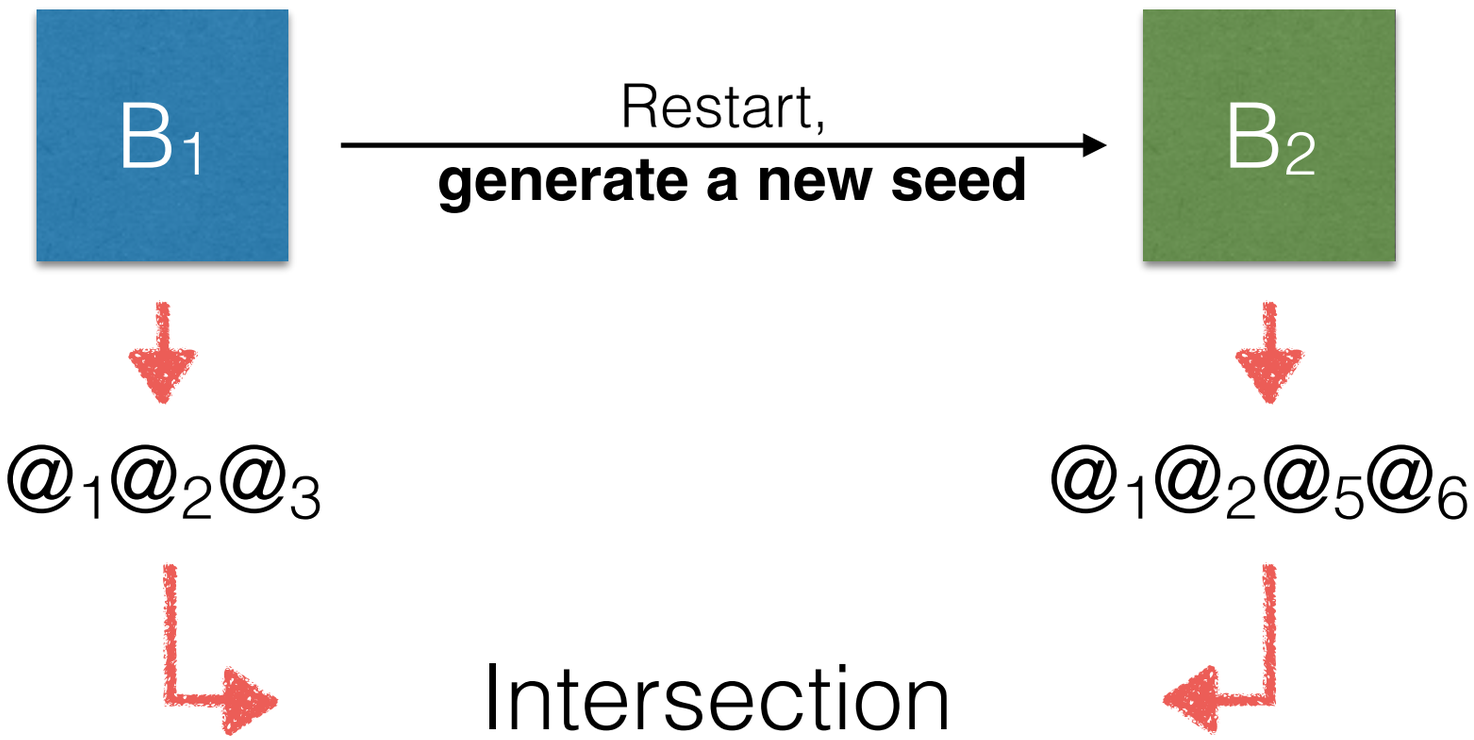
Experiment 3 - Restart



Exp.	Client	Seed	Size
No resize	Same	Same	Same
Resize	Same	Same	Different
Restart	**Same**	**Different**	**Same**
> 2 filter	Same	Different	Different

ETH zürich

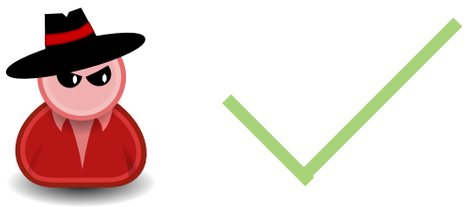
Experiment 3 - Restart



Exp.	Client	Seed	Size
No resize	Same	Same	Same
Resize	Same	Same	Different
Restart	**Same**	**Different**	**Same**
> 2 filter	Same	Different	Different

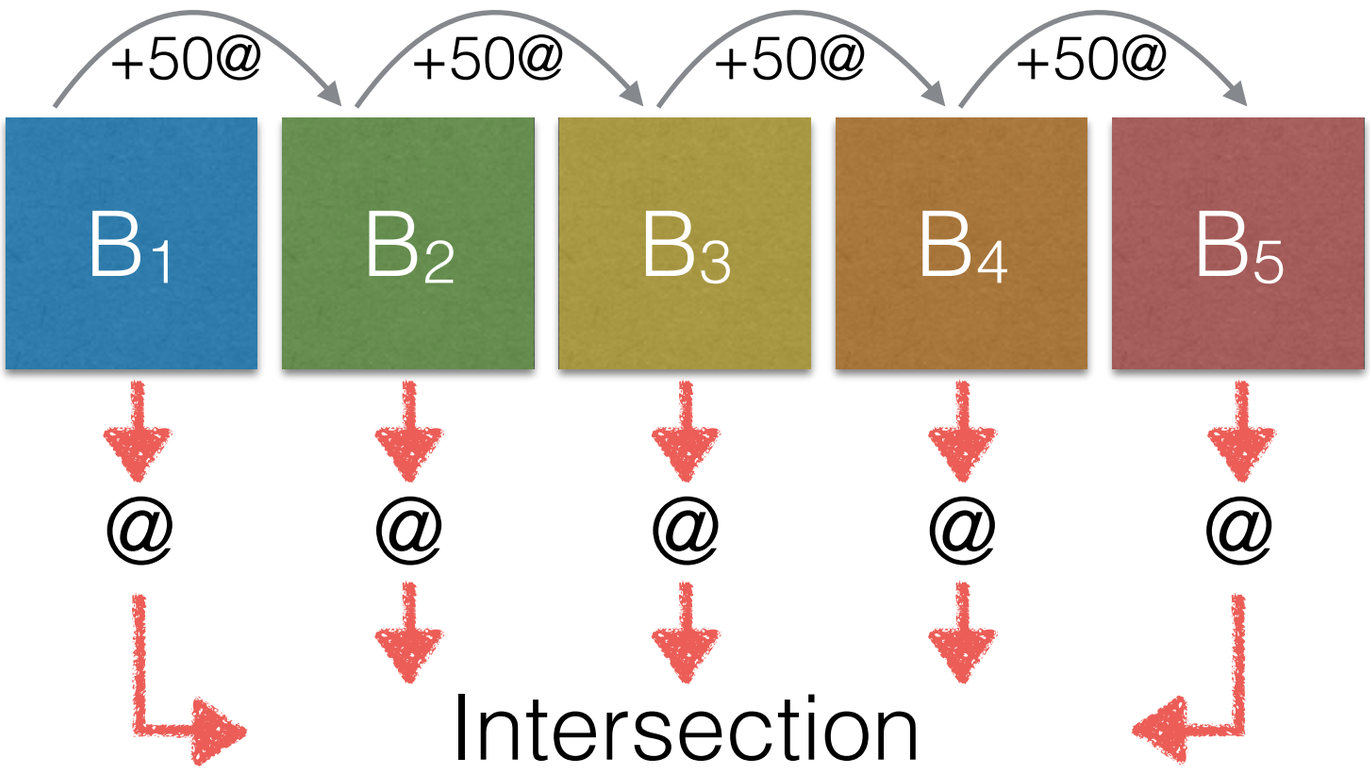
Results

- Yield mostly different positives
- Can be used for intersection attack



ETH zürich

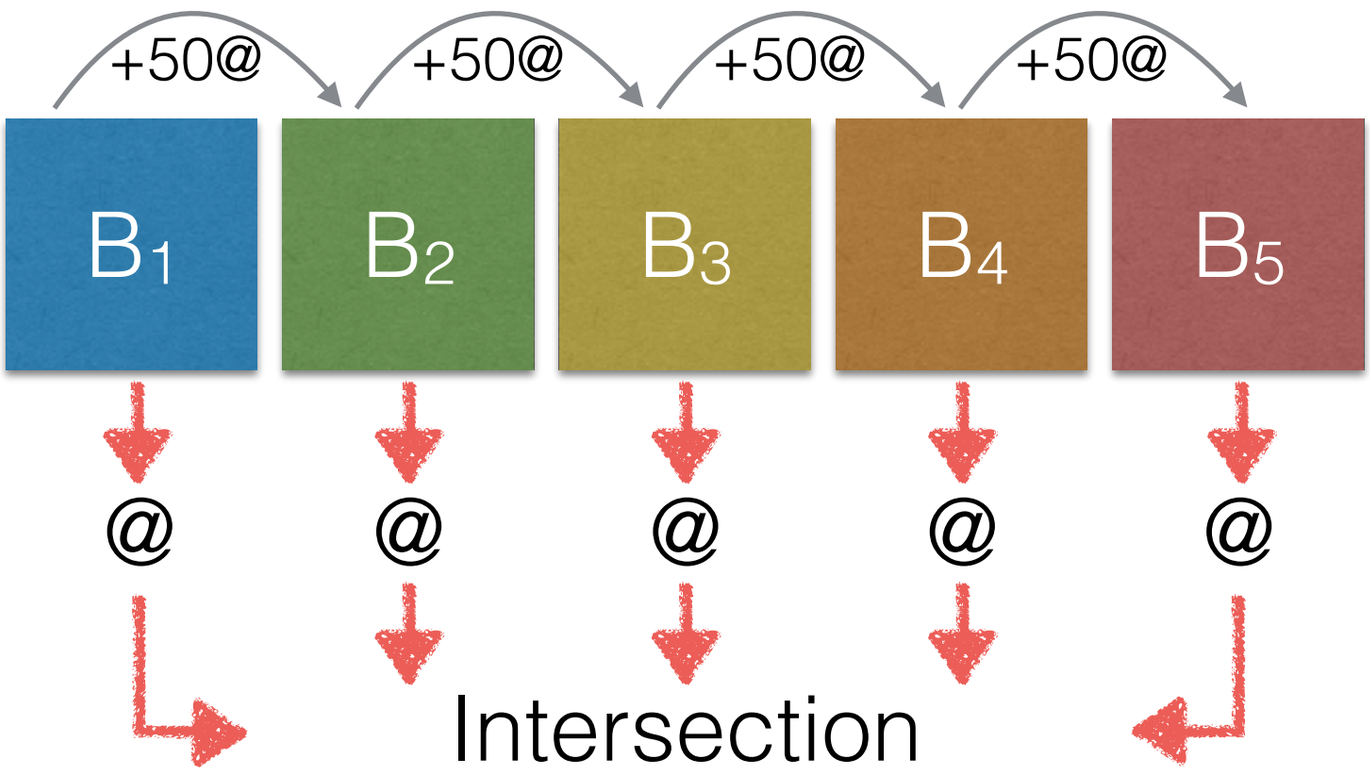
Experiment 4 - More than 2 filter



Exp.	Client	Seed	Size
No resize	Same	Same	Same
Resize	Same	Same	Different
Restart	Same	Different	Same
> 2 filter	**Same**	**Different**	**Different**

ETH zürich

Experiment 4 - More than 2 filter



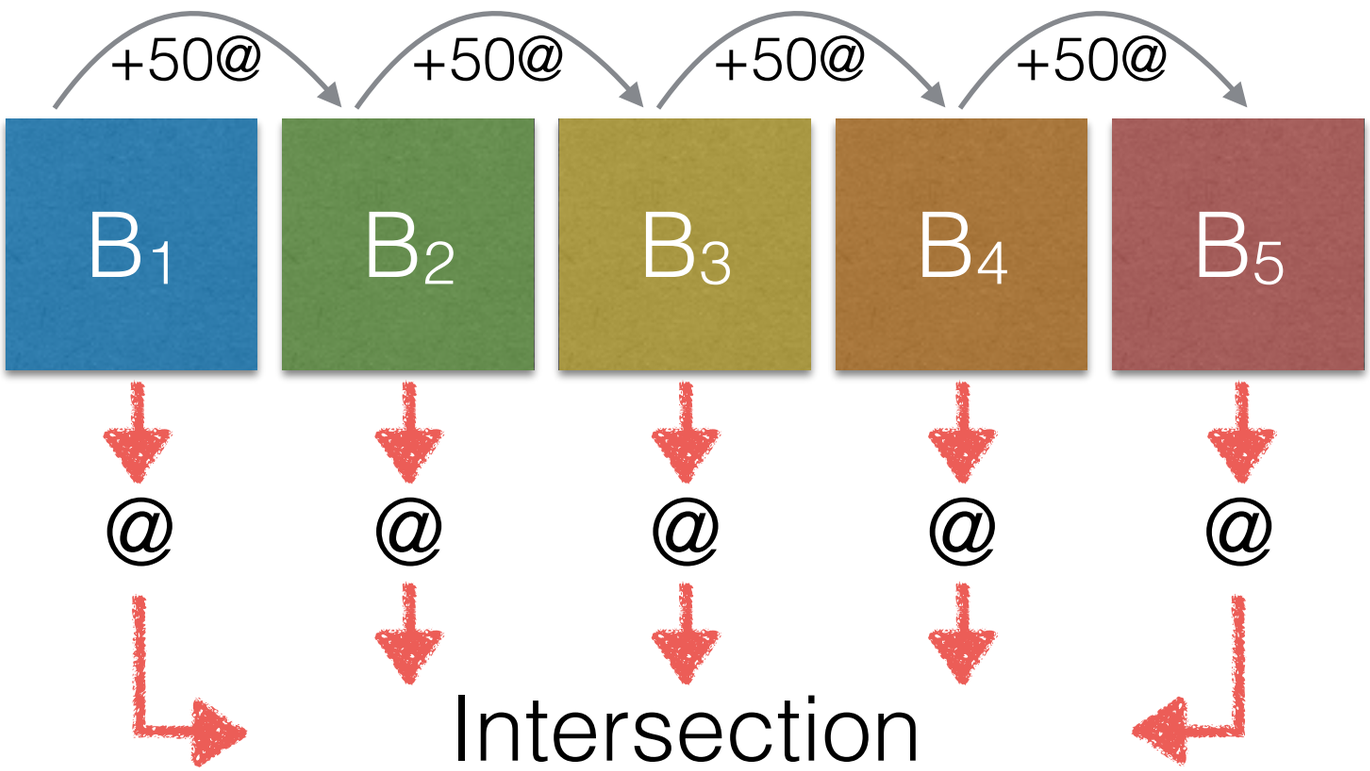
Exp.	Client	Seed	Size
No resize	Same	Same	Same
Resize	Same	Same	Different
Restart	Same	Different	Same
> 2 filter	**Same**	**Different**	**Different**

Results

Target FPR (%)	Probability linking all addresses with 3+ BF
0.05	~1
0.1	~1

ETH zürich

Experiment 4 - More than 2 filter



Exp.	Client	Seed	Size
No resize	Same	Same	Same
Resize	Same	Same	Different
Restart	Same	Different	Same
> 2 filter	**Same**	**Different**	**Different**

Results

Target FPR (%)	Probability linking all addresses with 3+ BF
0.05	~1
0.1	~1

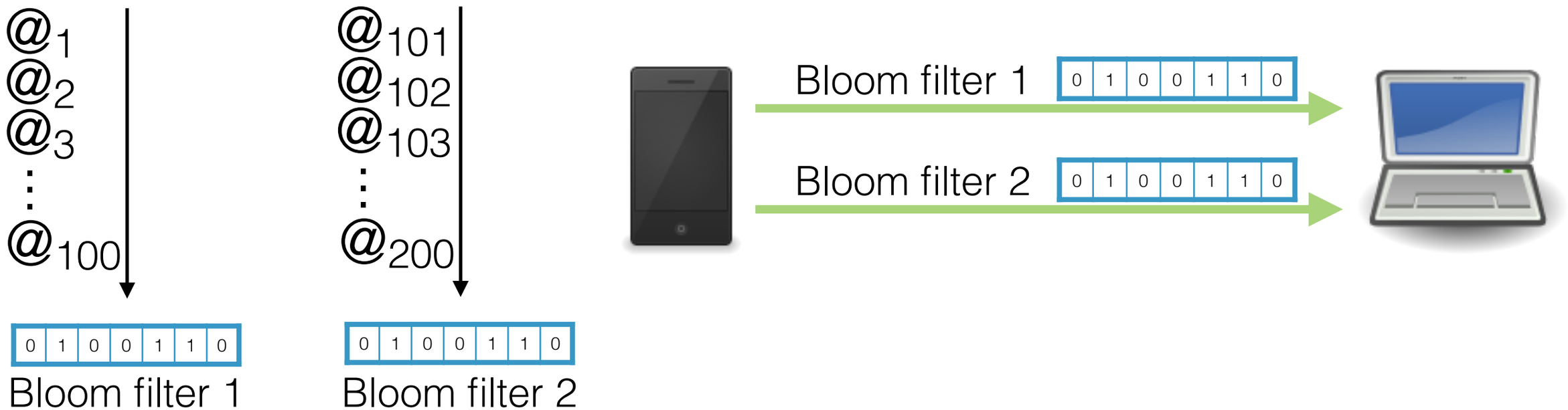
3 Bloom filter

All addresses inserted into B_1 can be linked

Observations

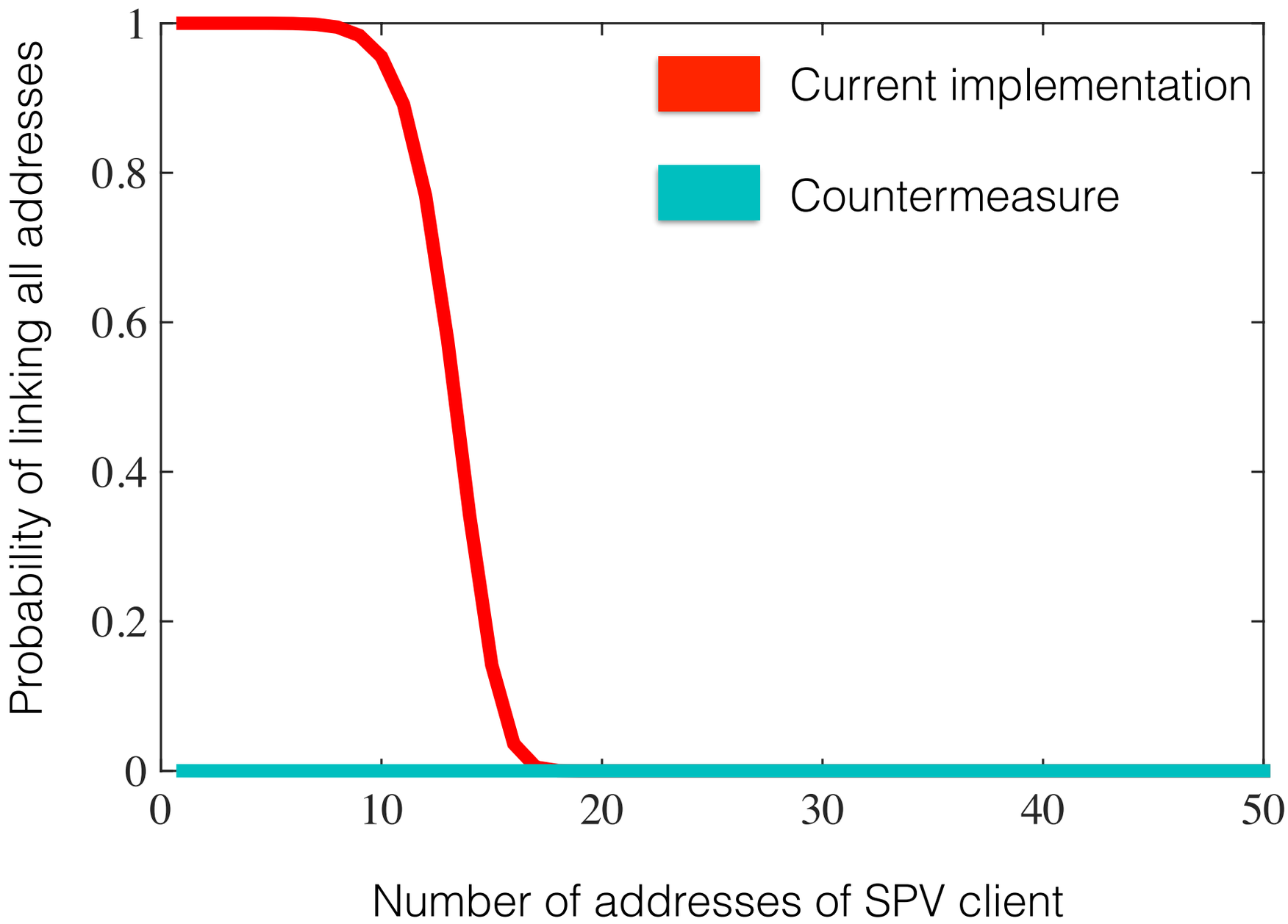
1. Need of constant false positive rate
2. Multiple Bloom filter with different parameters
3. SPV clients should keep state (e.g., about seed)

Proposed solution



- Pre-generate Bitcoin addresses and insert into filter
- Keep state about outsourced Bloom filter
- Overhead: For 100 addresses, < 1 kb

Proposed solution



ETH zürich

Information leakage through Bloom Filters in SPV clients

Analytical and Empirical evaluation

- 1 Bloom filter critical if < 20 Bitcoin addresses 
- 3+ Bloom filter intersection attack particularly strong 

Summary

ETH zürich

Information leakage through Bloom Filters in SPV clients

Analytical and Empirical evaluation

- 1 Bloom filter critical if < 20 Bitcoin addresses 
- 3+ Bloom filter intersection attack particularly strong 

Lightweight countermeasure

- **Significantly** reduces leakage
- Intersection attack **not effective**
- Requires **few** changes

Summary

ETH zürich

Information leakage through Bloom Filters in SPV clients

Analytical and Empirical evaluation

- 1 Bloom filter critical if < 20 Bitcoin addresses 
- 3+ Bloom filter intersection attack particularly strong 

Lightweight countermeasure

- **Significantly** reduces leakage
- Intersection attack **not effective**
- Requires **few** changes

Conclusion

- Bloom filter for privacy is delicate
- Designed carefully we can achieve proper privacy

Summary

ETH zürich

Information leakage through Bloom Filters in SPV clients

Analytical and Empirical evaluation

- 1 Bloom filter critical if < 20 Bitcoin addresses 
- 3+ Bloom filter intersection attack particularly strong 

Lightweight countermeasure

- **Significantly** reduces leakage
- Intersection attack **not effective**
- Requires **few** changes

Conclusion

- Bloom filter for privacy is delicate
- Designed carefully we can achieve proper privacy

Thank you!

Privacy metric

- Probability of correctly guessing j **real addresses** of a filter
- N: # of addresses inserted into filter
- S: # of false positives

Privacy metric

- Probability of correctly guessing j **real addresses** of a filter
- N: # of addresses inserted into filter
- S: # of false positives

$$P(1) = \frac{N}{N+S}$$

Privacy metric

- Probability of correctly guessing j **real addresses** of a filter
- N: # of addresses inserted into filter
- S: # of false positives

$$P(j) = \prod_{k=0}^{j-1} \frac{N-k}{N+S-k} = \frac{N}{N+S} \cdot \frac{N-1}{N+S-1} \cdots$$

Privacy metric

- Probability of correctly guessing j **real addresses** of a filter
- N: # of addresses inserted into filter
- S: # of false positives

$$P(j) = \prod_{k=0}^{j-1} \frac{N-k}{N+S-k} = \frac{N}{N+S} \cdot \frac{N-1}{N+S-1} \cdots$$

- Guessing all addresses correctly → link all addresses

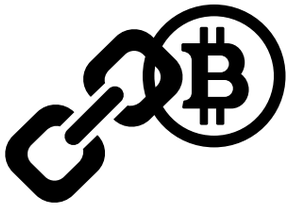
$$P(N) = \prod_{k=0}^{N-1} \frac{N-k}{N+S-k} = \frac{N!S!}{(N+S)!}$$

Adversary's model

Goal: Link Bitcoin addresses inserted within a Bloom filter

- Operates full Bitcoin nodes

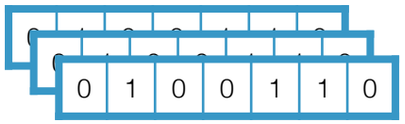
- Parses the Blockchain for addresses

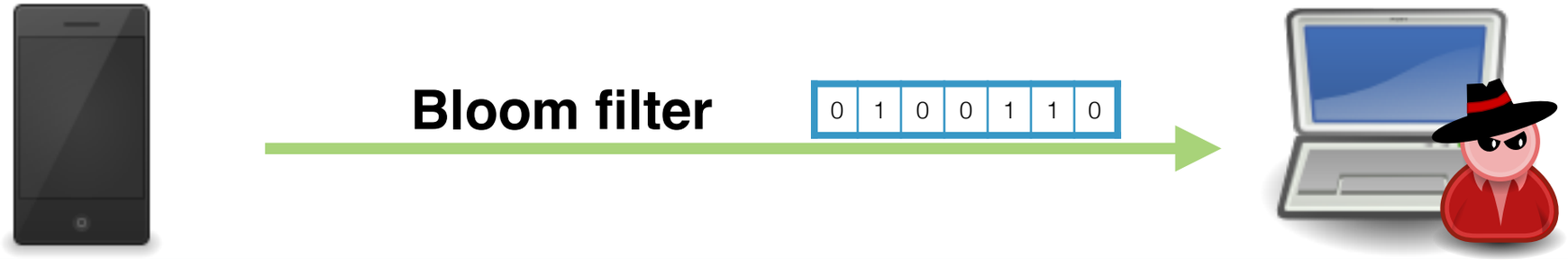


- Knows parameter for Bloom filter creation

  Target false positive rate

- Collects multiple Bloom filters per SPV client





Bloom filter false positive rate

$$\mathrm{FPR}(m) = \left(1 - \left(1 - \frac{1}{n}\right)^{km}\right)^{k}$$

Notation

n	Size of the Bloom filter in bits
m	Number of elements inserted into the Bloom filter
k	Number of hash functions of the Bloom filter