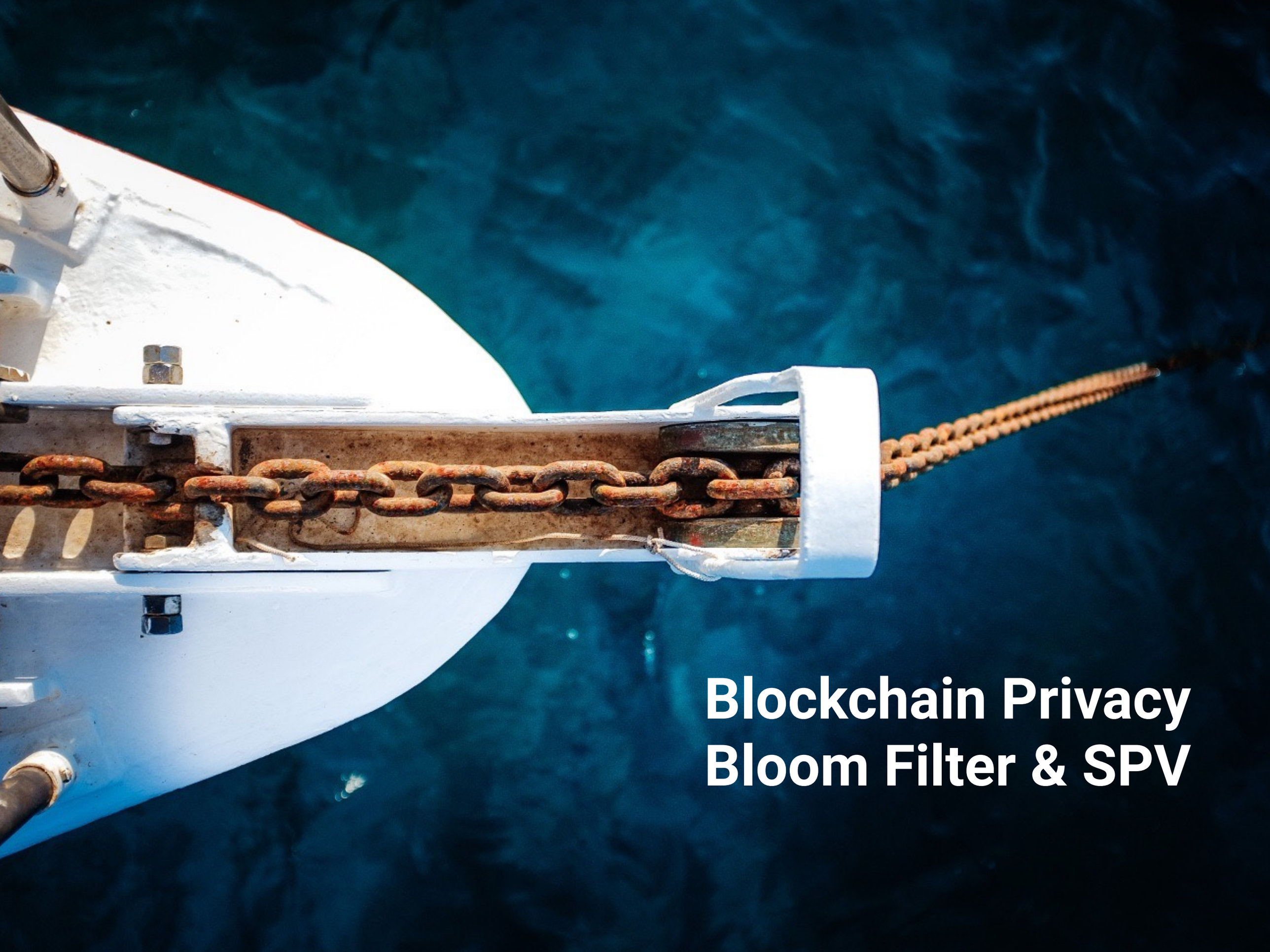
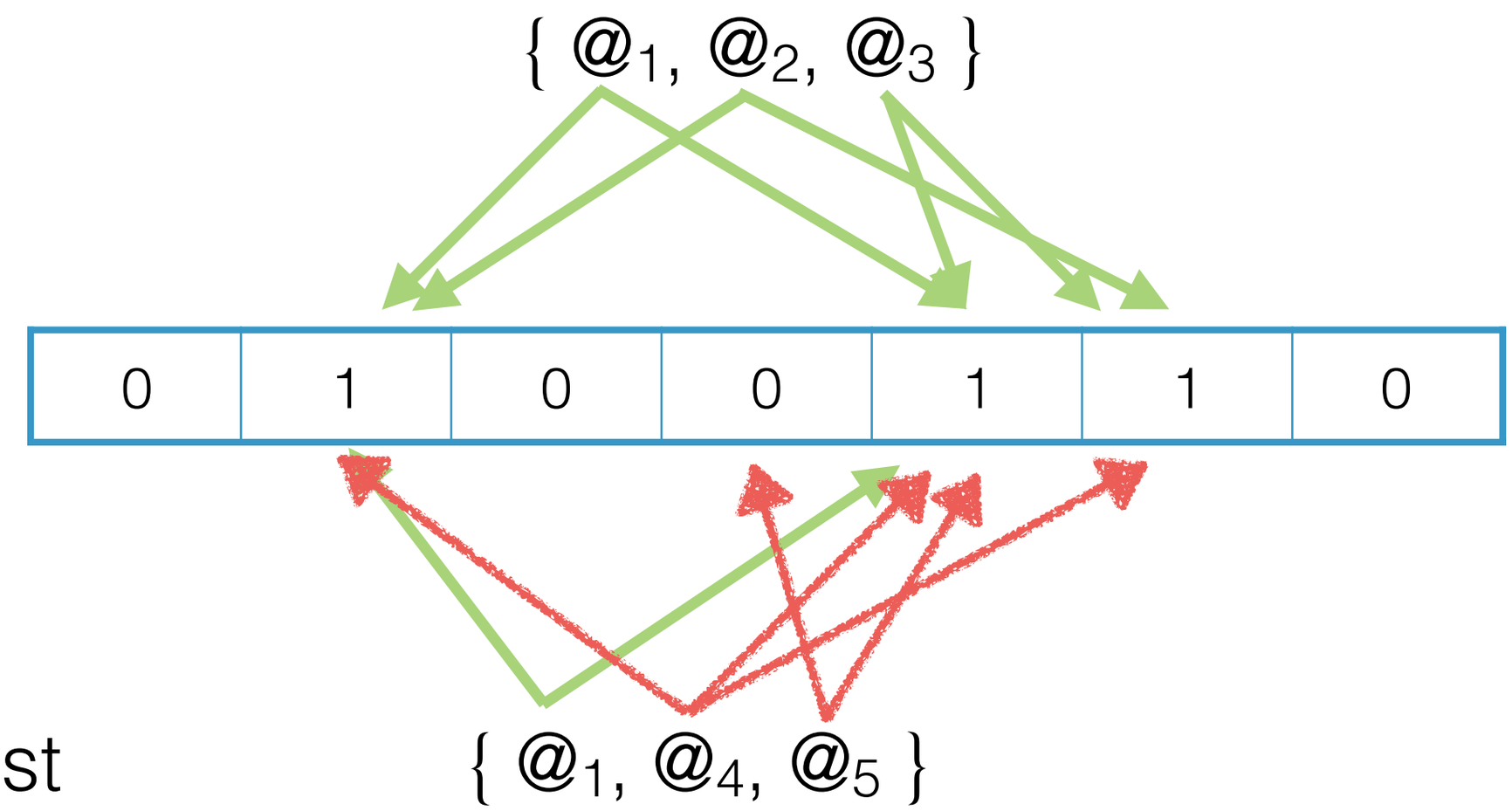


# Blockchain Privacy
# Bloom Filter & SPV

# Enable mobile Bitcoin clients

Insertion



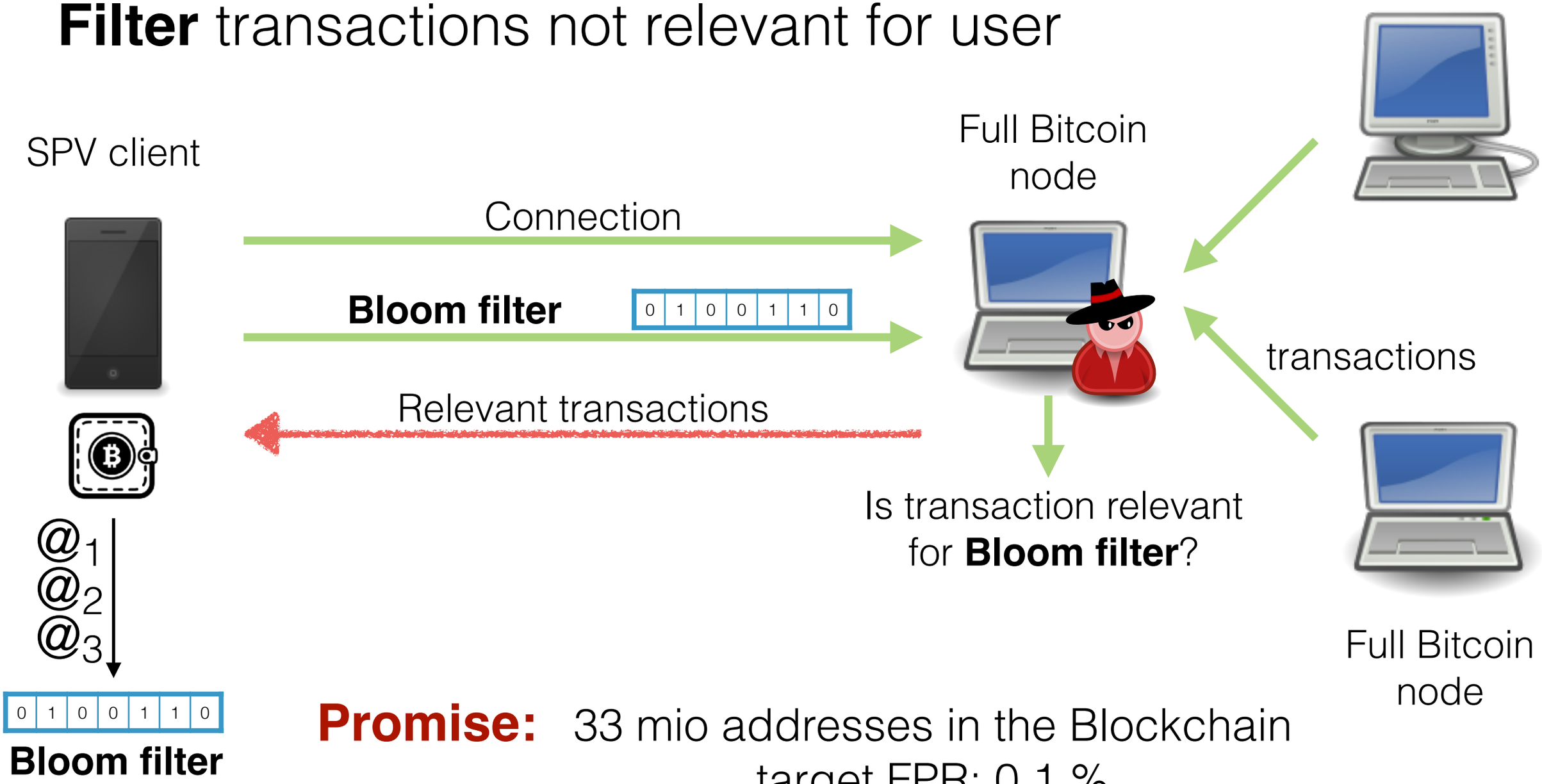

Membership test

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

@5 True negative

# Simple Payment Verification (SPV)

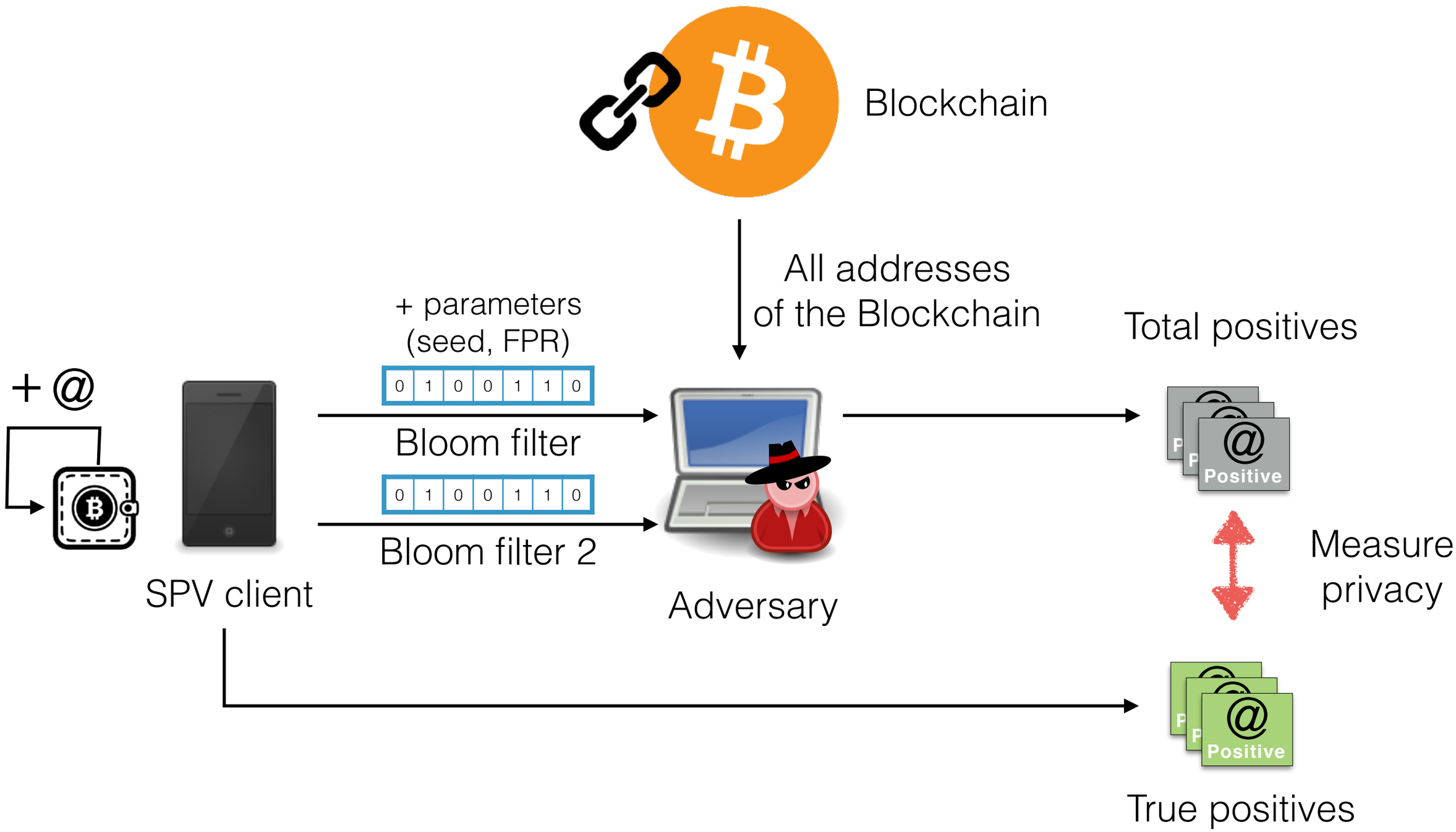
**Filter** transactions not relevant for user




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

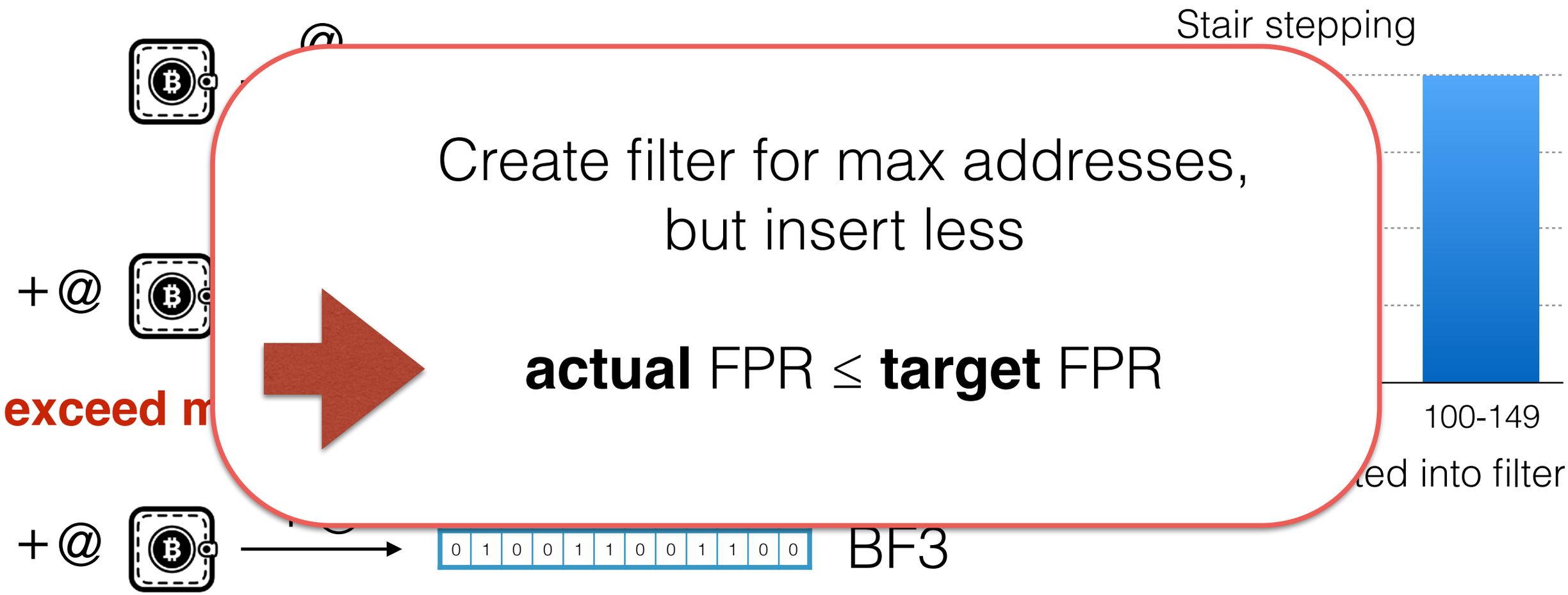
"User addresses hidden amongst 33 000" false positives

# Model and Privacy measure



# Stair stepping

Bloom filter designed for
- max number of **addresses**
- **target FPR** when max addresses inserted

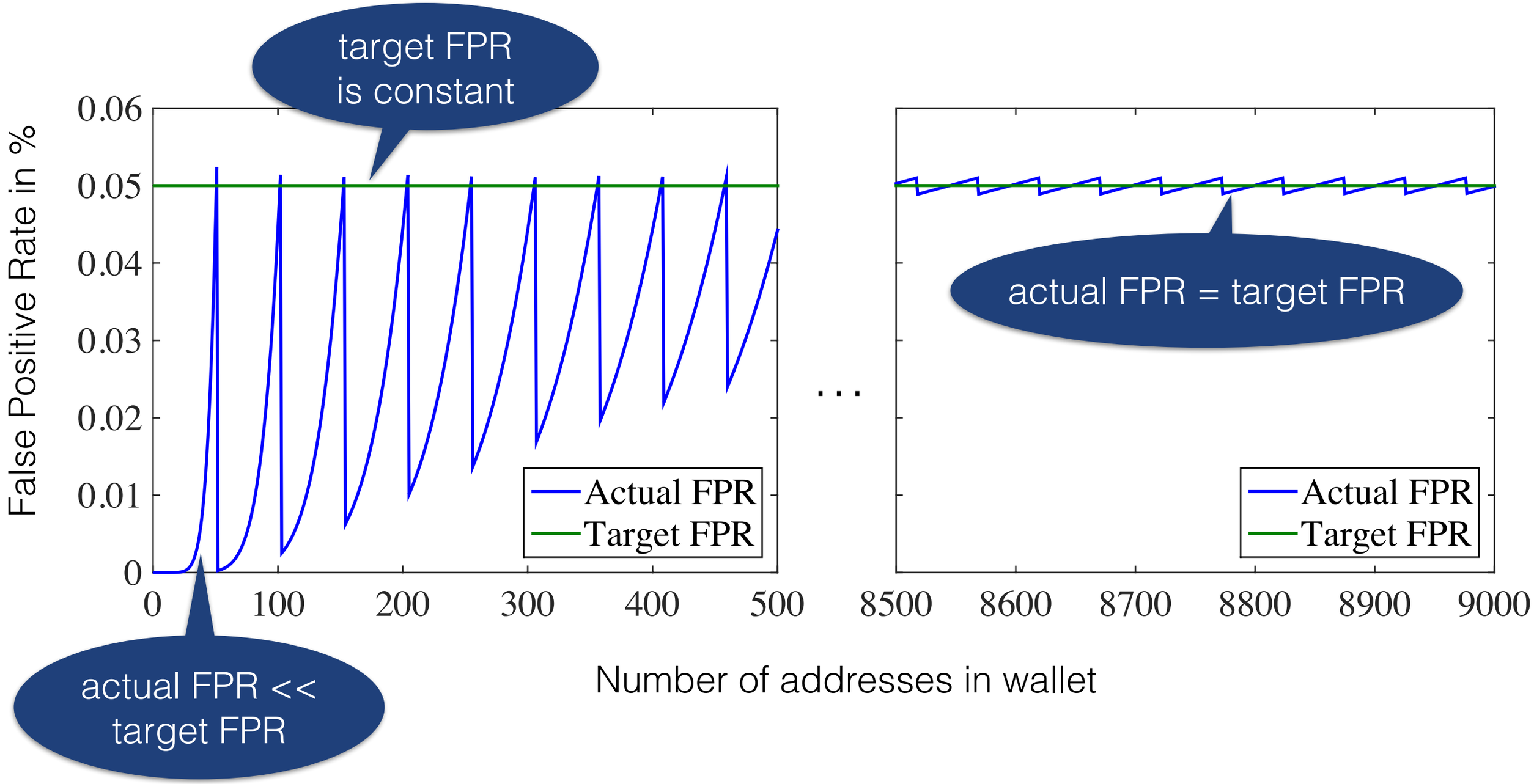



Create filter for max addresses,
but insert less

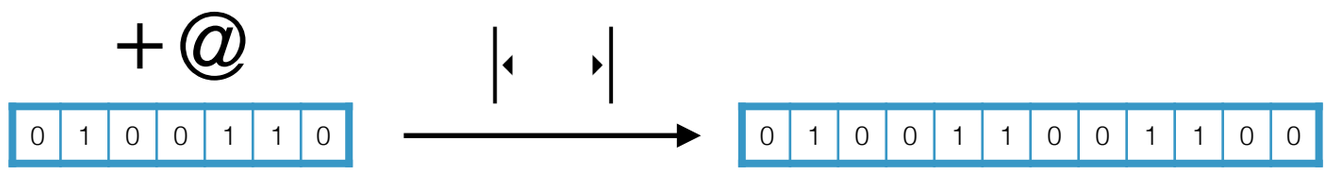
**actual** FPR ≤ **target** FPR

Rationale: **avoid** filters with different sizes

# Analytical results - Actual FPR vs. Target FPR



## Resizing



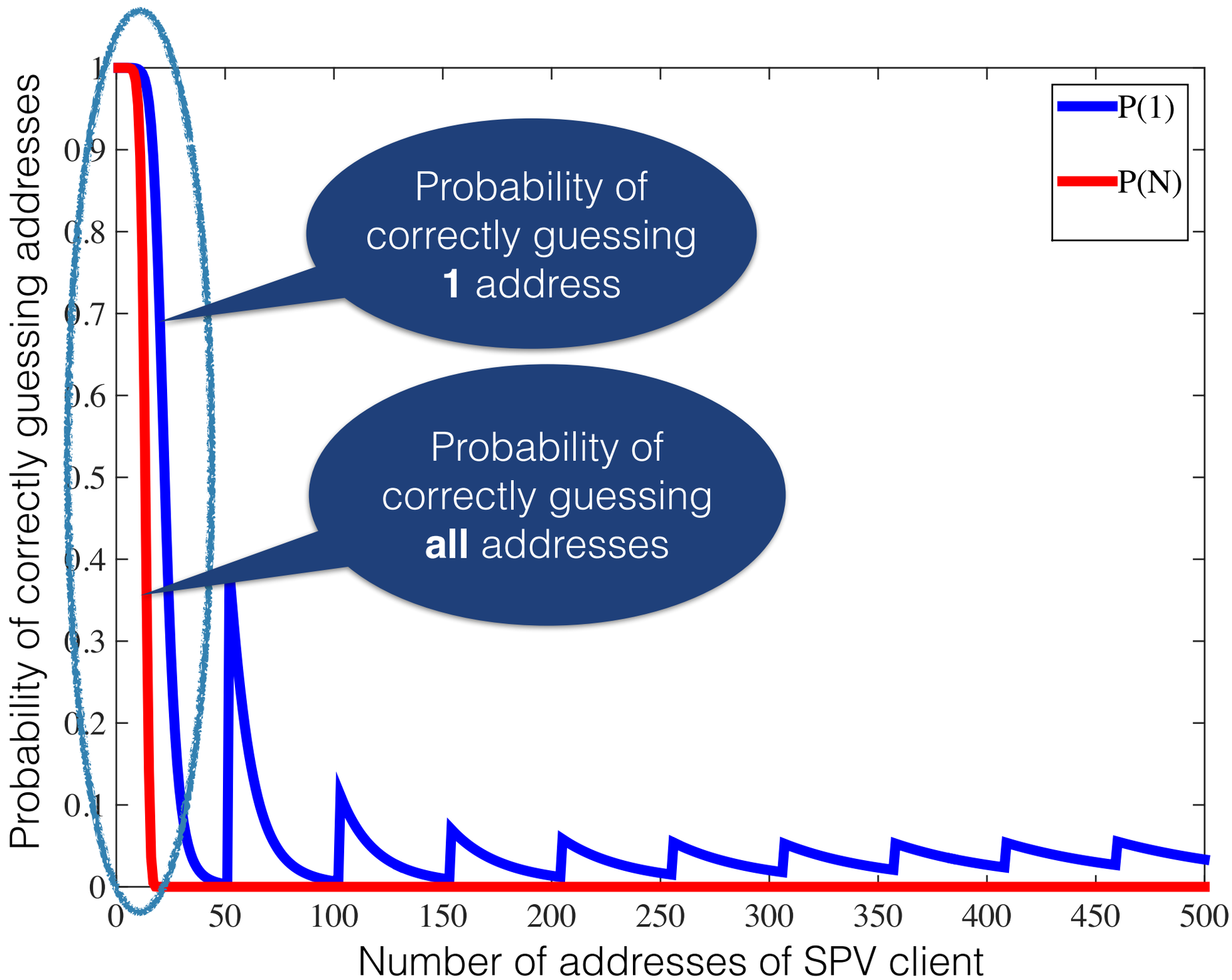

Once max addresses inserted —> bigger filter

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

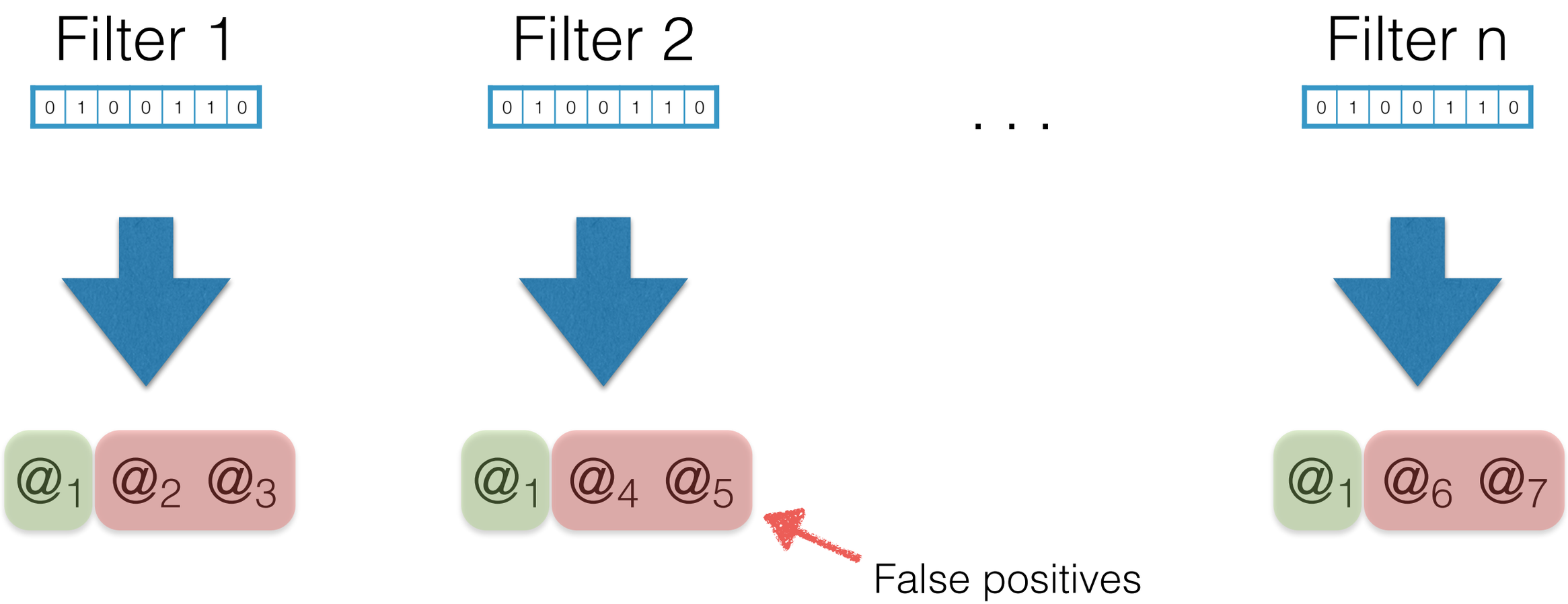
Summary of current SPV design choices

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

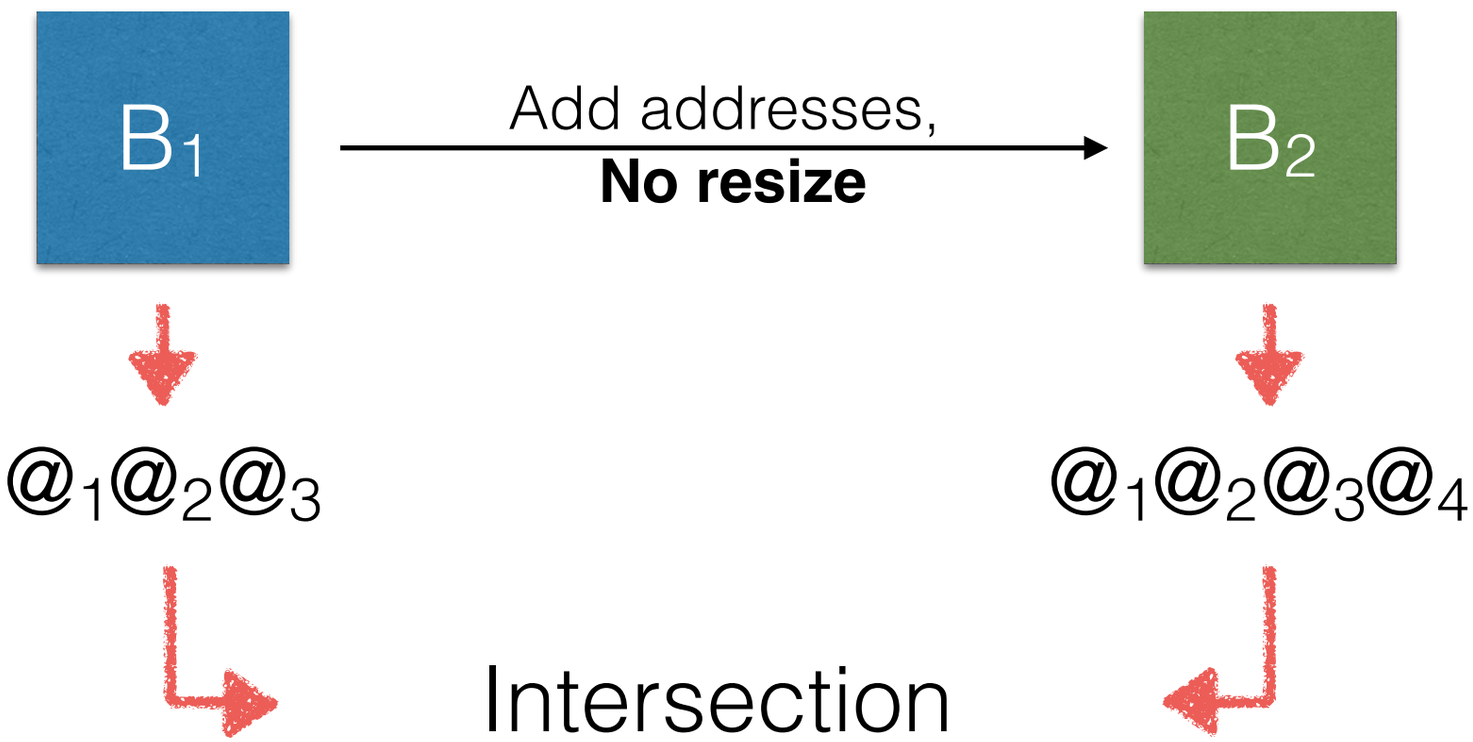
# One Bloom filter



# Multiple Bloom filters



# 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 |

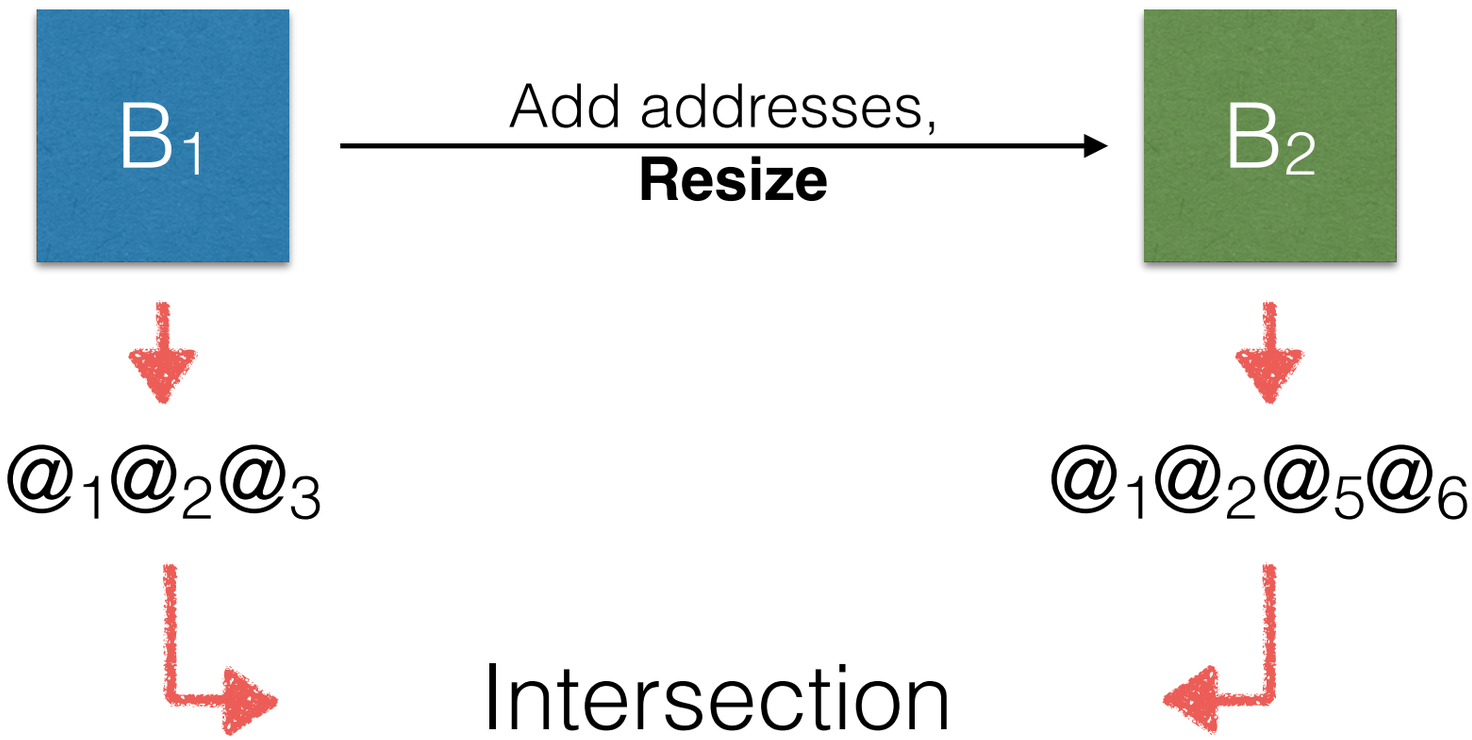
## Results

| Target FPR (%) | P(1) with 2 BF | P(1) with 1 BF |
|---|---|---|
| 0.05 | 0.2990 | 0.2910 |
| 0.1 | 0.1020 | 0.1070 |
| 0.5 | 0.0078 | 0.0075 |

no change of privacy

- Yield the same positives
- The adversary does not learn a lot

# Experiment 2 - Resize




## Results

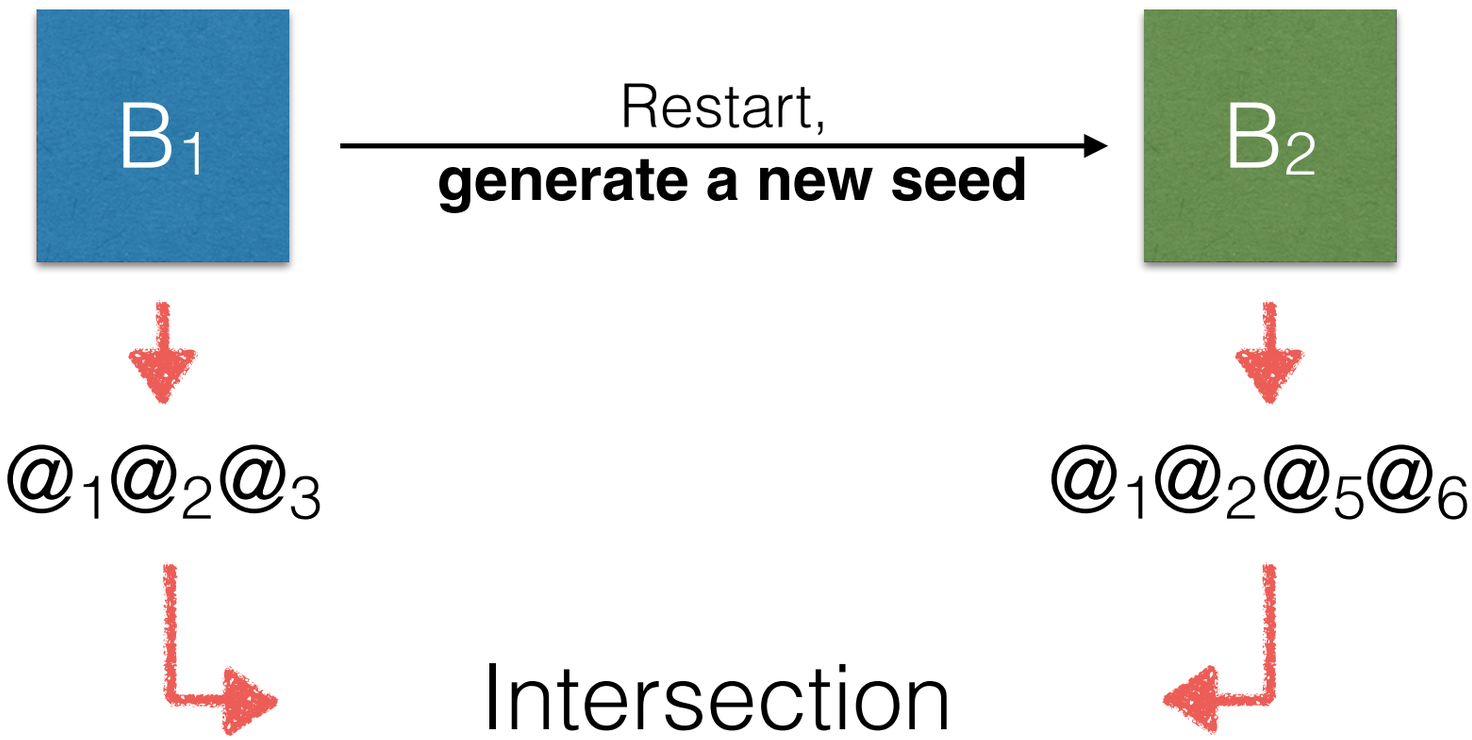
| Target FPR (%) | P(1) with 2 BF | P(1) with 1 BF |
|---|---|---|
| 0.1 | 0.98 | 0.03 |

significant change

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

- Different BF sizes improve the attack

# Experiment 3 - restart




## Results

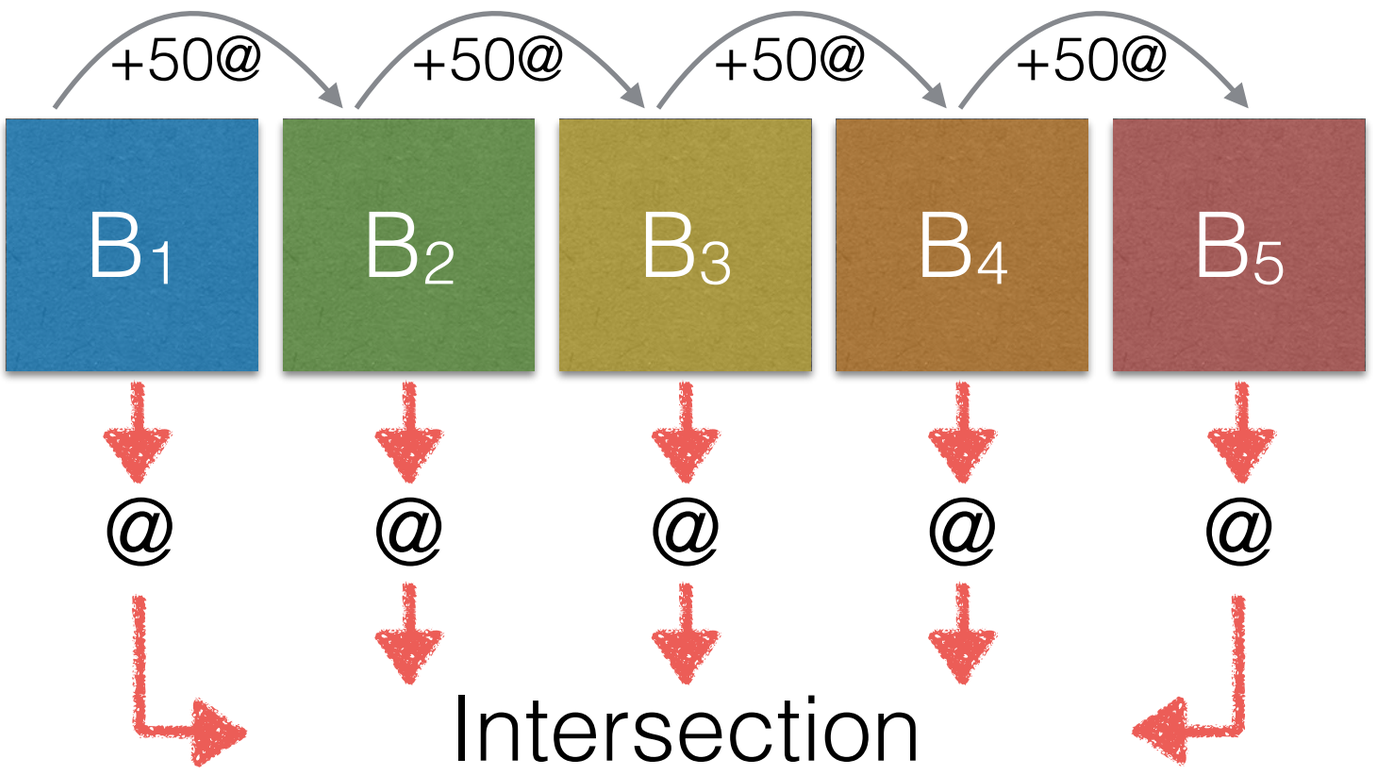
| Target FPR (%) | P(1) with 2 BF | P(1) with 1 BF |
|---|---|---|
| 0.1 | 0.99 | 0.04 |

significant change

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

- Different BF seeds improve the attack

# Experiment 4 - More than 2 filter




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

## Results

Guessing all addresses

| Target FPR (%) | P(N) given 3 or more BF |
|---|---|
| 0.05 | ~1 |
| 0.1 | ~1 |

3 Bloom filter

All addresses yielded by $B_1$ are leaked

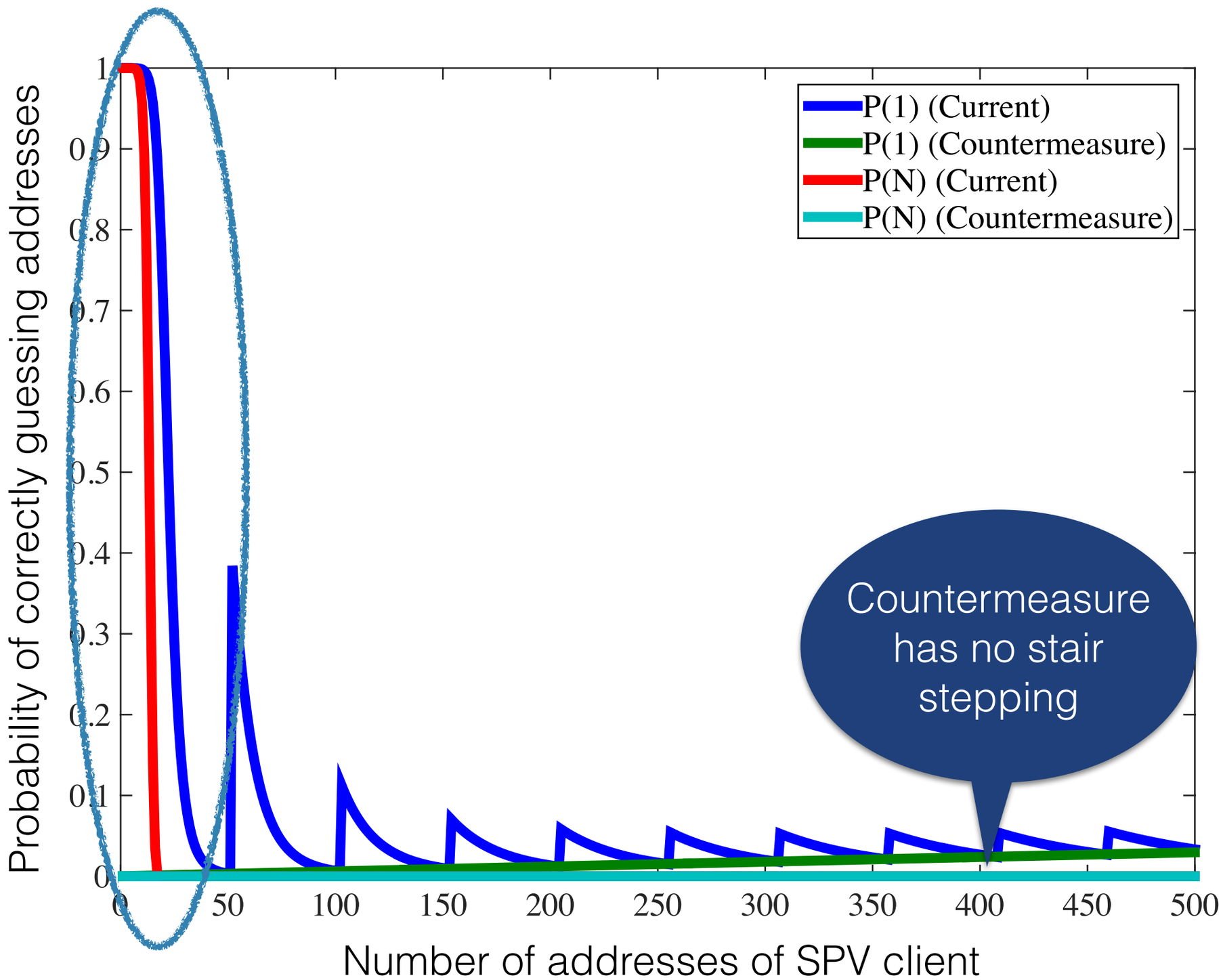
# Observations

1. Need constant FPR

2. Multiple Bloom filter with different parameters

3. SPV clients should keep state (e.g., about seed)

# Proposed solution



## 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!**