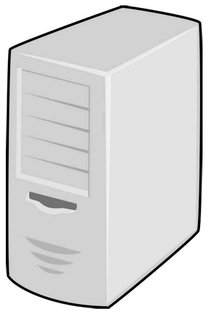
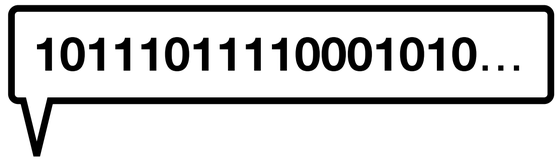
A way to achieve more private abuse-resistant location tracking

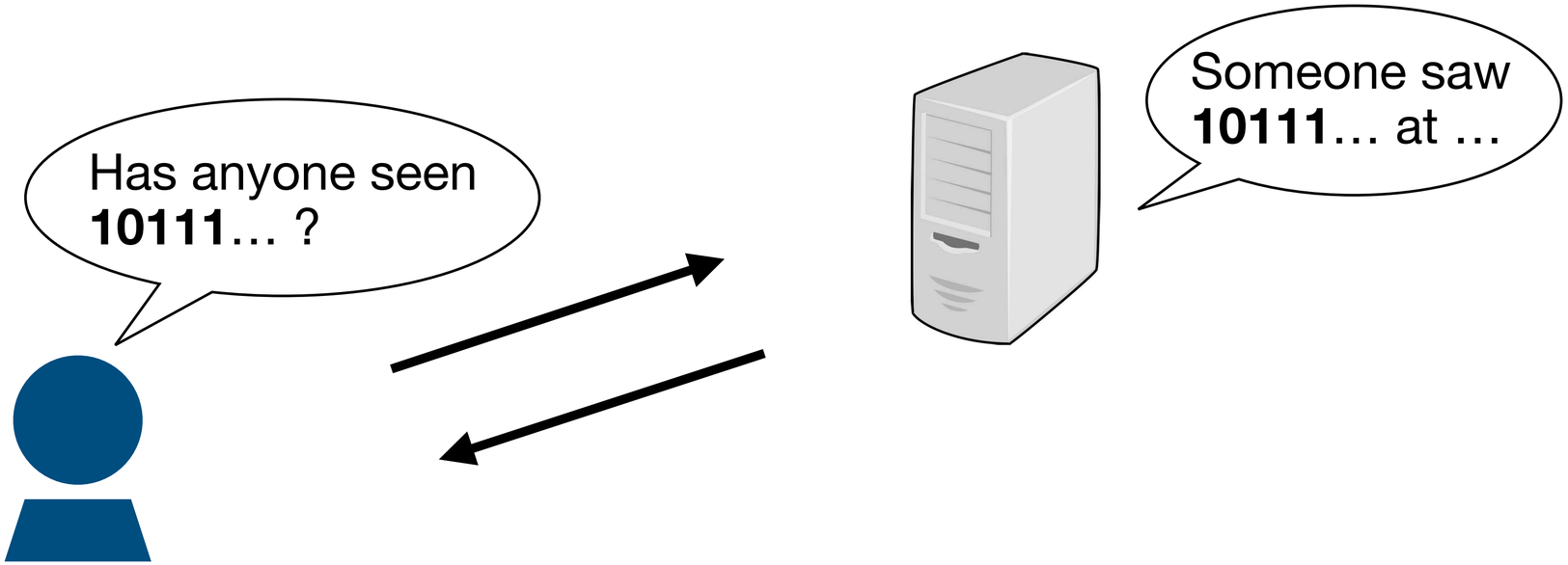
Based on paper by *Harry Eldridge,* ***Gabrielle Beck****, Matthew Green, Nadia Heninger, and Abhishek Jain.* **https://eprint.iacr.org/2023/1332.pdf**

Quick Refresher: Crowd-Sourced LT





1011101111… - Pseudorandom Identifier


Has anyone seen
10111… ?
Someone saw
10111… at …

Current Existing DULT Proposal [1]

- Based mainly off a design introduced by Apple
  - To combat *tracking* adversaries, identifier is periodically rotated
  - To combat *stalkers*, the identifiers rotate slower when disconnected from the owner

Current Existing DULT Proposal [1]

- Tags have two modes, **near-owner** and **separated**

Near-Owner	Separated
Identifiers rotate every **15** minutes	Identifiers rotate every **24** hours

Tag is expected to send out a broadcast every **2-4** seconds

Disadvantage of existing approach

- One problem with the current approach is it lacks **privacy** for honest tag users
  - In **separated** mode, any two broadcasts made from the same tag can be linked
    - This means there is little to no privacy for honest tag users against a tracking adversary

Disadvantage of existing approach

- **Separated** mode is not necessarily a less sensitive mode than near-owner
  - Separated only means separated from the owner device, not separated from all honest people
    - e.g. a friend takes your car, it has an AirTag in it, they can be tracked by anyone
  - If separated bxs are linkable to near-owner bxs or do not change across transitions between states this may also lead to deanonymization attacks

Reasons for existing tensions

- Difficult to achieve honest user privacy and stalker detection *simultaneously*
  - Seems like anything that prevents honest tag users from being tracked will help stalkers
  - Everyone here already understands this better than most

2. It is not possible to correlate the public keys broadcast across multiple epochs without knowing the shared key SK, which is only know to the owner. However, an observer who sees multiple beacons within the same epoch can correlate them, as they will have the same Y_i. However, fast key rotation also makes it more difficult to detect unwanted tracking, which relies on multiple observations of the same identifier over time.

From [2]

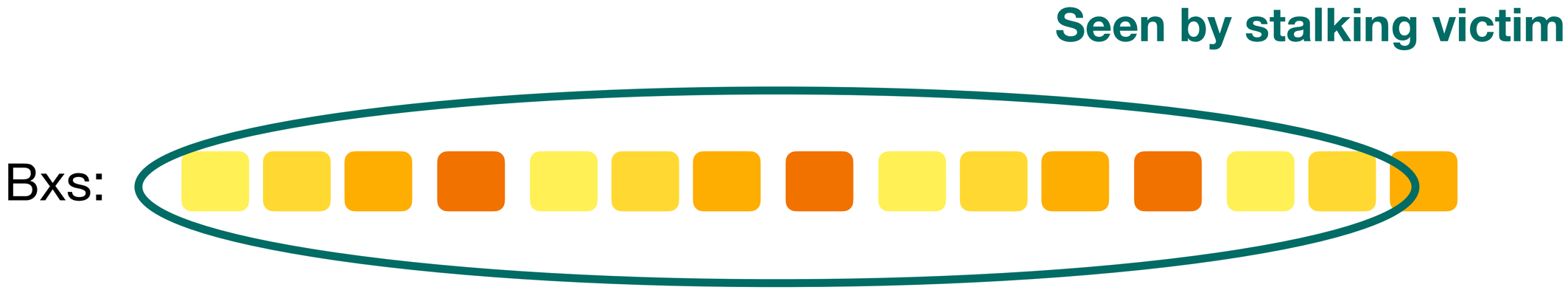
Is there an alternative approach?

- Suppose a tag produces y broadcasts in some window of time

Bxs:

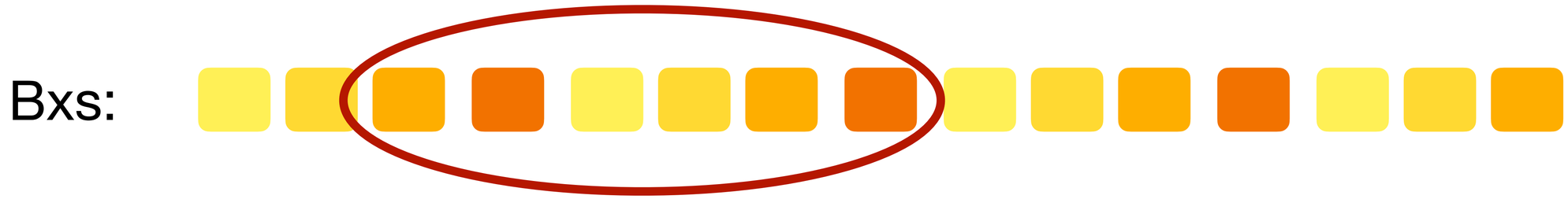
Is there an alternative approach?

- If tag is placed on a person, they will see the majority of broadcasts made by the device



Is there an alternative approach?

- …but a tracking adversary is likely to see a much smaller fraction of broadcasts (assuming they do not have full visibility, everywhere, at once)

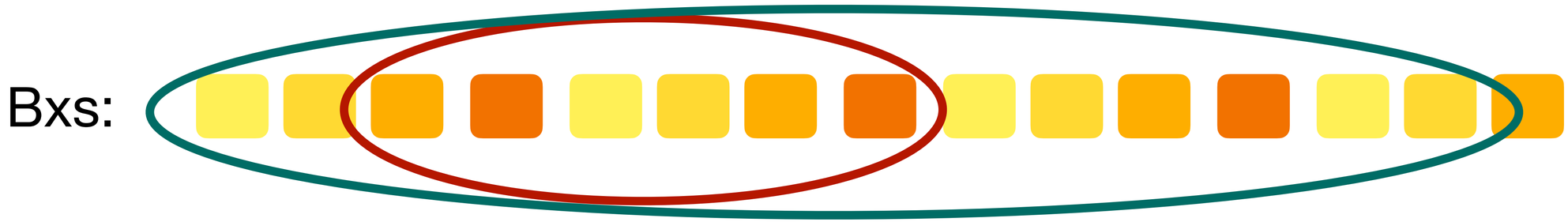


Is there an alternative approach?

Observation: This difference in *resource capabilities* can be leveraged to come up with a way to provide stalker detection while having privacy against a tracking adversary

Seen by stalking victim

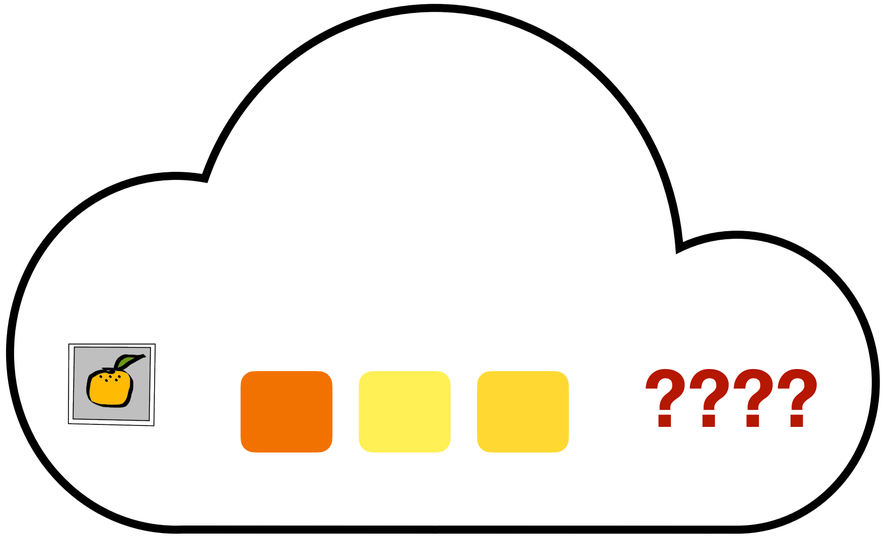
Seen by tracking adversary



What we want to achieve

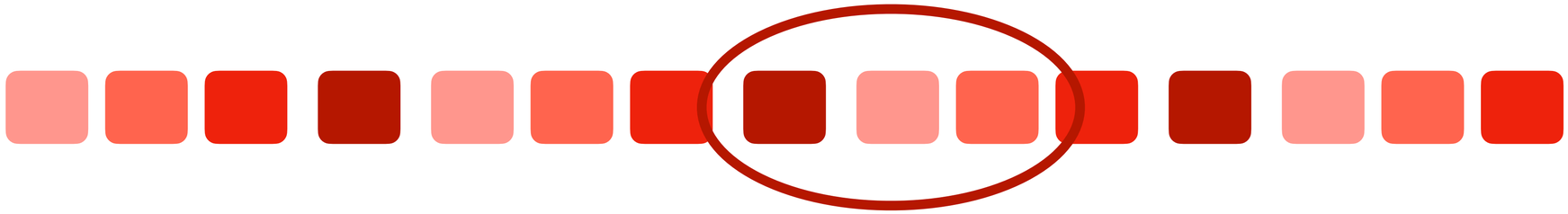
Come up with bxs such that an adversary who only sees a few cannot tell anything about the distribution of *who* sent the bxs

Could be from a different tag…



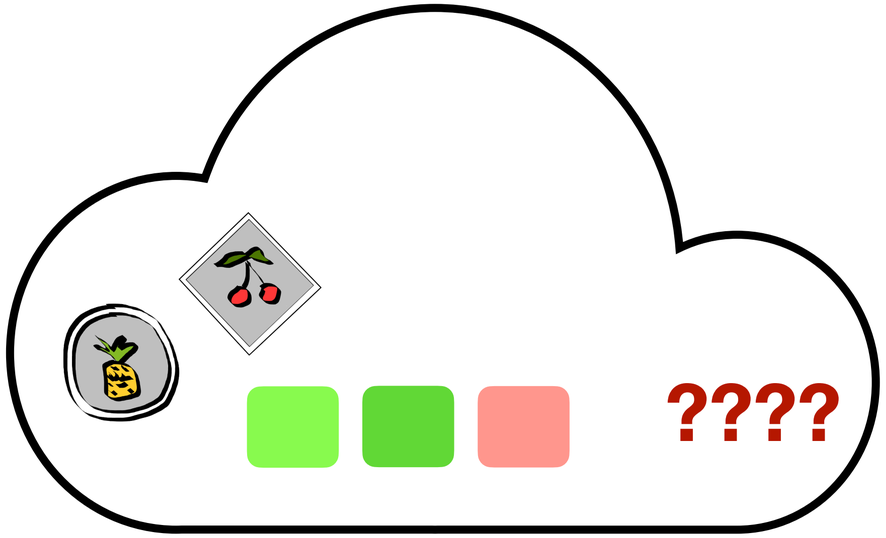
γy bxs, $0 \leq \gamma < \delta$, for some parameter δ

Bxs:



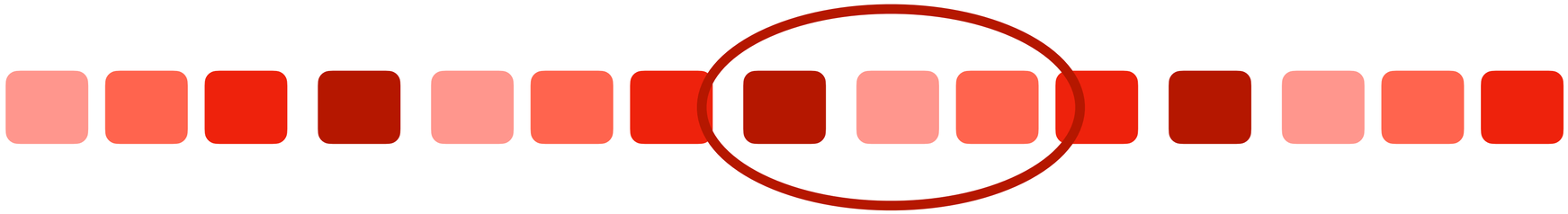
What we want to achieve

… or many different tags, or any combination of the user's tag and other users



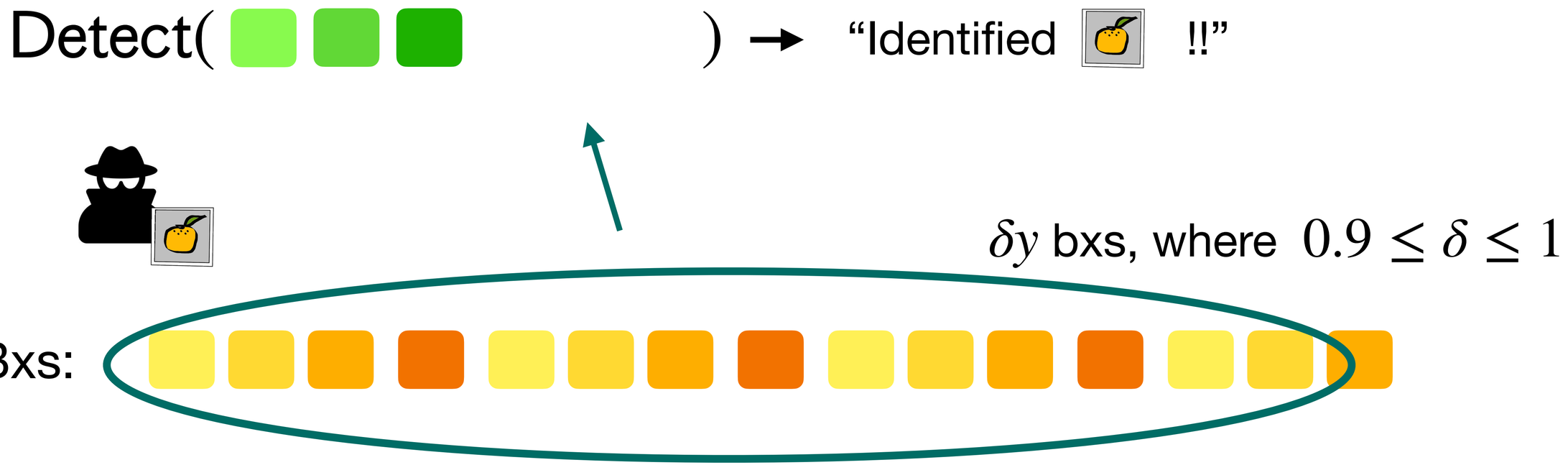
γy bxs, $0 \leq \gamma < \delta$, for some parameter δ

Bxs:



What we want to achieve

If a stalking victim sees a large number of broadcasts, they should be able to do detection



How the approach works

- Turn the broadcasts into Shamir secret shares
- *Secret sharing*
  - A way to "split" a secret s into *n shares* $sh_1, \ldots, sh_n$ so that
    - Any subset of shares of size $\leq k$ does not reveal the secret
    - For any subset of shares of size $\geq t$, (where t is some parameter, $t > k$), there is an efficient procedure to recover s

How the approach works

- *Shamir Secret Sharing* uses polynomials for sharing
  - Choose at random a polynomial $f(x) = a_k x^k + \ldots + a_0$ with constraint $a_0 = s$
  - Choose some set of distinct points $e_1, \ldots, e_n$, the shares are simply $f(e_1), \ldots, f(e_n)$
  - To recover *s*, assuming no errors in input, do *lagrangian interpolation* with any $k + 1$ points

How the approach works

- To use in a crowd-sourced tracking solution, each bx must include a shamir share
- Shares and the identifier rotate at the same speed
    - Identifier not used for stalking detection
    - Assuming each tag chooses its x coordinates $e_1, \ldots, e_n$ at random, different broadcasts cannot be tracked

How the approach works

- If there is only one tag - and no other noise - you can just do lagrangian interpolation
- What if there are other tags?
  - Can try and do lagrangian interpolation with at all subsets of size $k+1$??
    - There are $\binom{n}{k+1}$ of those!!!

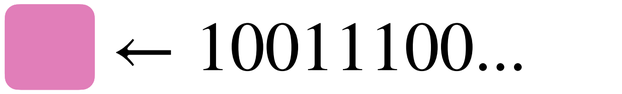
How the approach works

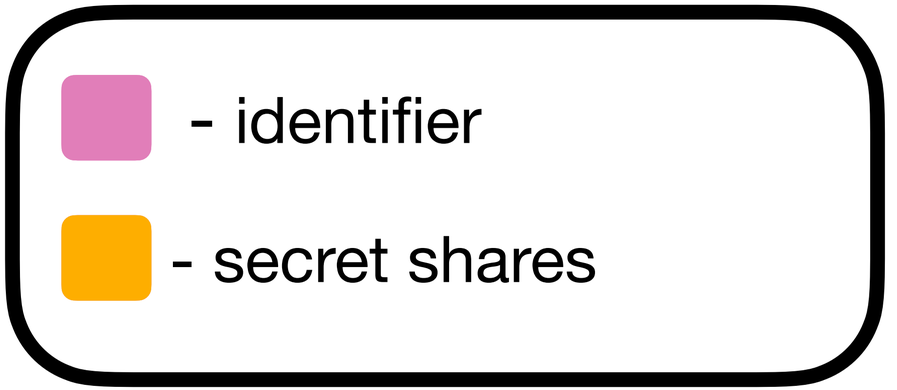
- One way around this is to take advantage of techniques from coding theory, since Shamir secret shares are the same as Reed-Solomon codes
  - Treat Shamir secret shares as symbols from a codeword
  - Broadcasts from other tags can be considered “errors” in the codeword
    - If number of bxs coming from other tag is small, do unique decoding
    - Otherwise, use list decoding algorithms

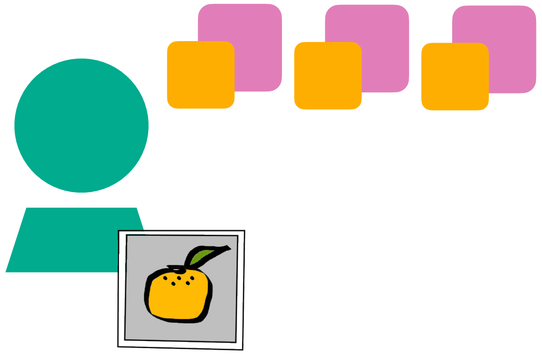
How the approach works

- Many list-decoding algorithms have large gaps between the number of points needed for recovery and the degree of the polynomial
  - This affects achievable privacy and detectability
  - With lagrangian interpolation the gap is one
- A lot of effort in the paper is spent on coming up with a new encoding and decoding scheme that get closer to this gap

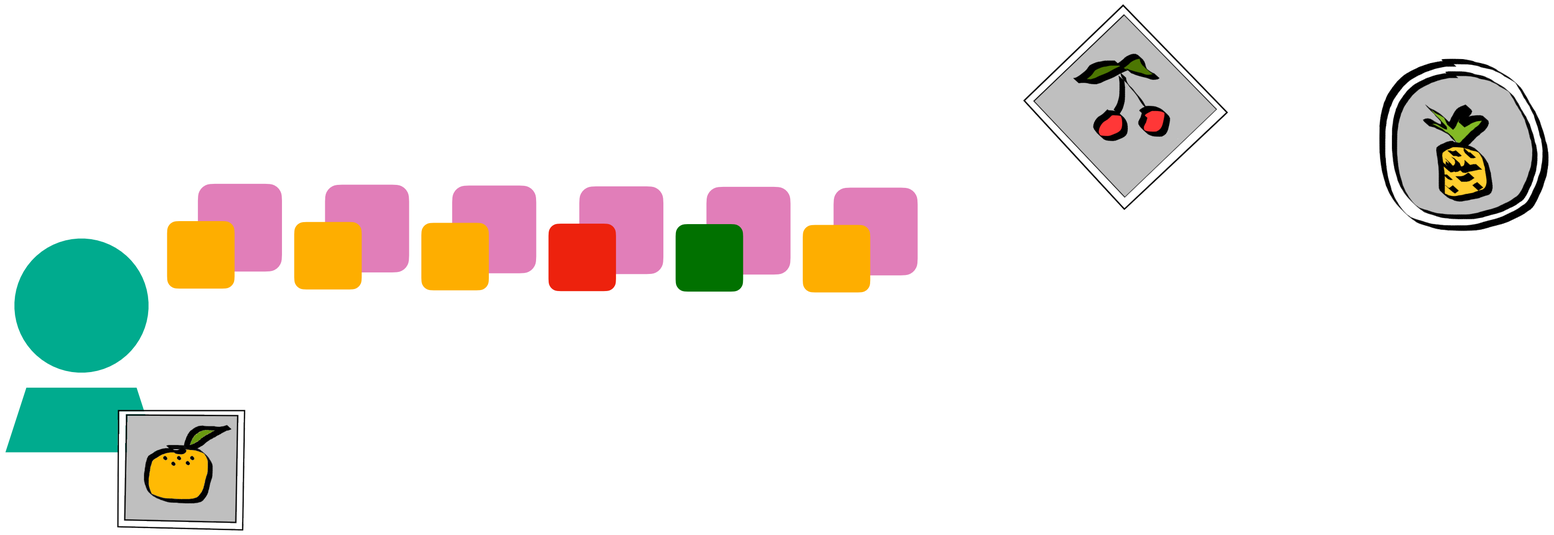
How the approach works



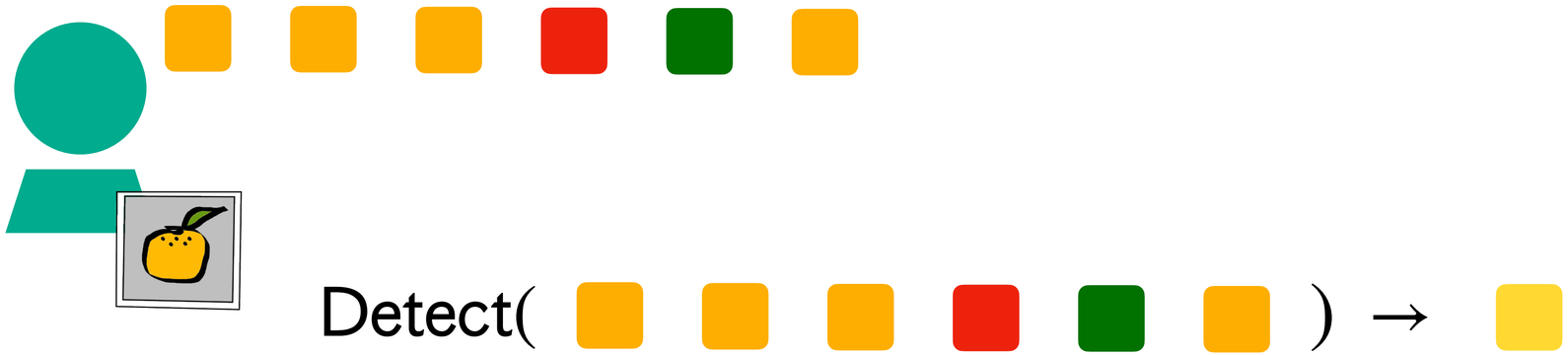




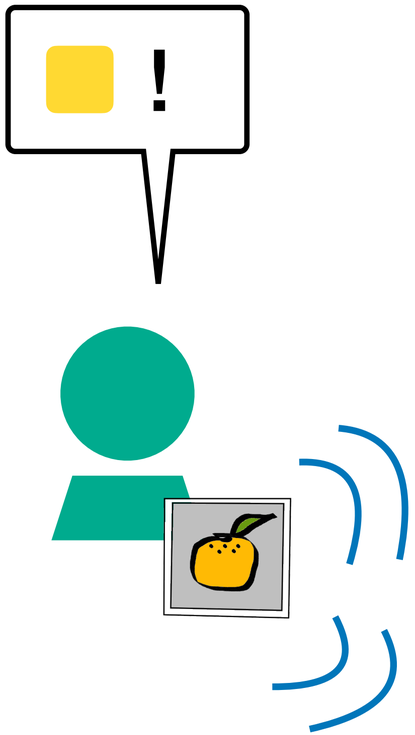
How the approach works



How the approach works



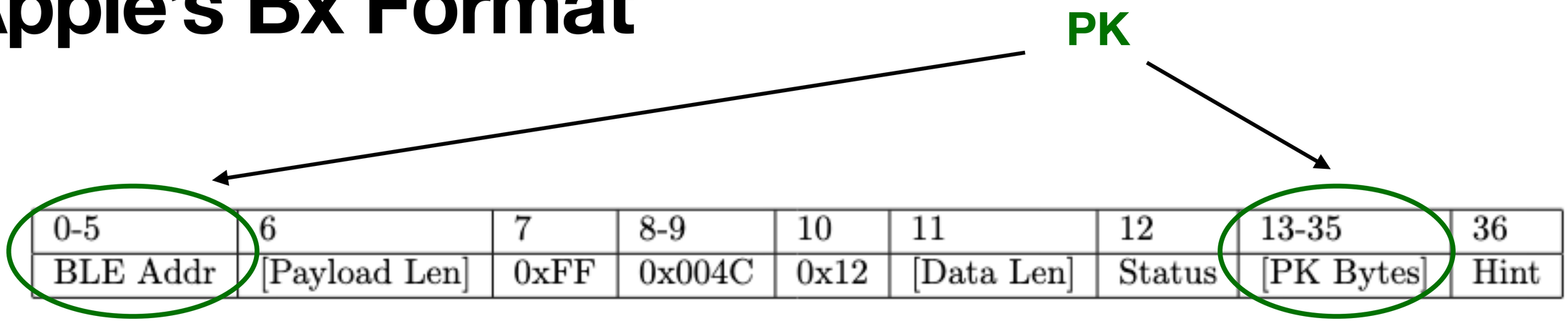
How the approach works



Design Considerations

- Pre-BLEv5 size of packet can only be 37 bytes (with the BLE address and other relevant fields)
  - There is no room for additional data
  - For compatibility, identifier may need to be sent separately from MDSS content
    - Broadcast every 2 seconds, alternating sending each

Apple's Bx Format



0-5	6	7	8-9	10	11	12	13-35	36
BLE Addr	[Payload Len]	0xFF	0x004C	0x12	[Data Len]	Status	[PK Bytes]	Hint

- Pseudorandom identifier in Apple is a public key for a PKE
- PK is 28 bytes
  - 22 bytes + 2 bits of key are immediately after the "Status" byte
  - 5 bytes + 6 bits of key are in BLE Addr

From [3]

Proposed Bx Format

Replace with MDSS content

0-5	6	7	8-9	10	11	12	13-35	36
BLE Addr	[Payload Len]	0xFF	0x004C	0x12	[Data Len]	Status	[PK Bytes]	Hint

- New broadcast has similar format to the old one
  - Keeping the address and other important fields in tact, we can afford **25 bytes = 200 bits** for the MDSS content

Parameters - High Level

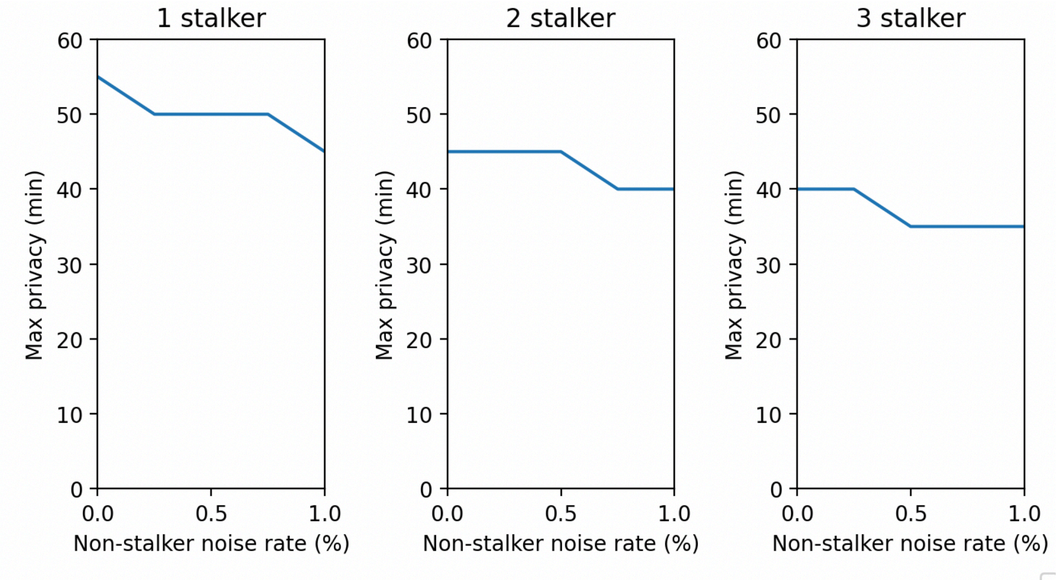
- When attempting to deploy MDSS in this setting, implementors must also consider…
  - Identifier rotation rate
  - The environment
    - max number of stalking tags expected to be placed on a victim
    - max amount of bxs expected from honest tags

Parameters - High Level

- Faster identifier rotation means…
  - More unique broadcasts given to detection algorithm
    - This will make detection algorithms slower
  - better privacy is achievable (typically)

Parameters - High Level

- If more stalkers and noise must be tolerated from other tags
  - Privacy for honest users will likely degrade
  - Detection algorithms may also be slower



Parameters - High Level

- Packet size also effects achievable privacy and can also effect algorithm running time
    - The larger the packets, the better the privacy parameter can be made (subject to identifier rotation rates)
    - Detection is slower in most if not all cases

ID rotation rate	Privacy (min)	MDSS Content (bits)
every 5 min	35	168
	40	260
	45	315

Choosing Parameters

- We assume…
  - pre BLEv5 limits on packet size
  - stalkers should be detected in an hour
  - there are at most…
    - 3 tags placed on individual from a stalker
    - “noise” from honest tags equal to 1/2 the broadcasts made by a single stalking tag

Choosing Parameters

ID rotation rate	Privacy for honest users (min)	Bx Size (bits)
every 5 min	35	168
every 10 min	30	200
every 15 min	30	170

Note: All sizes are below or at the 200 bit limit

Benchmarking

Epoch duration	# Unique bxs received	Detection runtime (no stalkers)	Detection runtime (stalkers present)	Polynomial recovery (all stalkers)
5 min	42	10 ms	20 ms	30 ms
10 min	21	10 ms	10 ms	20 ms
15 min	14	10 ms	10 ms	10 ms

Table 1: Benchmarks for detection on a MacbookPro, assuming broadcasts are being received at a rate that is equivalent to the number that would be produced by 3 tags.

Detection algorithms are faster than in paper because rotation period is much longer, meaning there are less unique points

Overall Takeaways

- **Advantages**
  - Solution has more privacy for honest tag users, while still retaining ability to detect stalking tags
- **Disadvantages**
  - Does require knowledge of environment conditions to achieve detection
    - Must have upper bound on number of tags to tolerate stalking one person and amount of “noise” from non-malicious tags

Further Considerations

- Benchmarking does not take into account malicious security against rogue tags
  - If you want to combine this solution with BlindMy would require further work
- Experiments would need to be re-done to figure out what reasonable amount of noise is in given environments for a particular rotation rate
  - May be able to do additional filtering because of longer rotation periods

References

- [1] https://datatracker.ietf.org/doc/draft-ledvina-dult-accessory-protocol/
- [2] https://datatracker.ietf.org/doc/draft-fossaceca-dult-finding/
- [3] Heinrich et al. AirGuard - Protecting Android Users From Stalking Attacks By Apple Find My Devices. https://arxiv.org/pdf/2202.11813