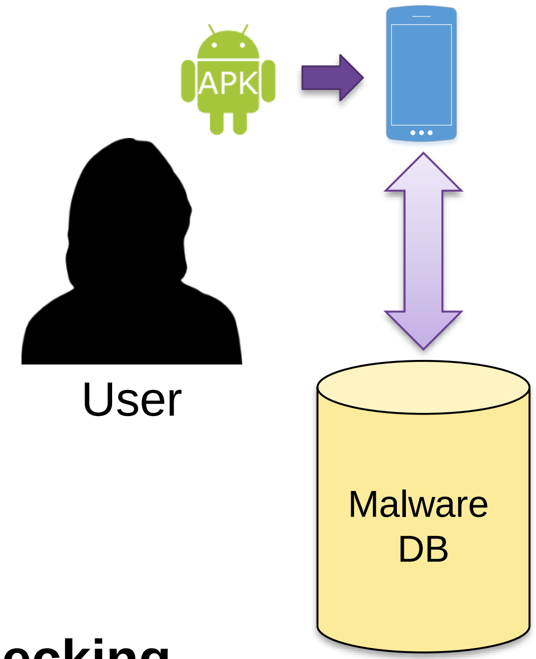
The Circle Game: Scalable Private Membership Test Using Trusted Hardware

Sandeep Tamrakar [1] *Jian Liu [1]* *Andrew Paverd [1]*
Jan-Erik Ekberg [2] *Benny Pinkas [3]* *N. Asokan [1]*

1. Aalto University, Finland *2. Darkmatter (work done at Trustonic)* *3. Bar-Ilan University, Israel*

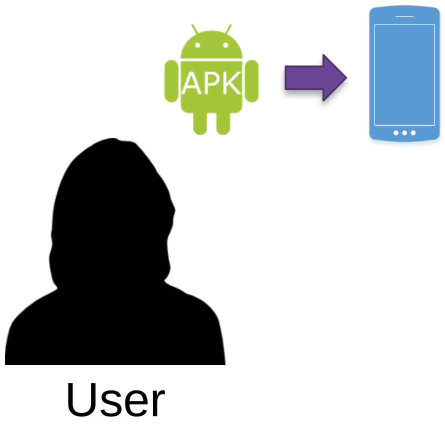
Malware Checking

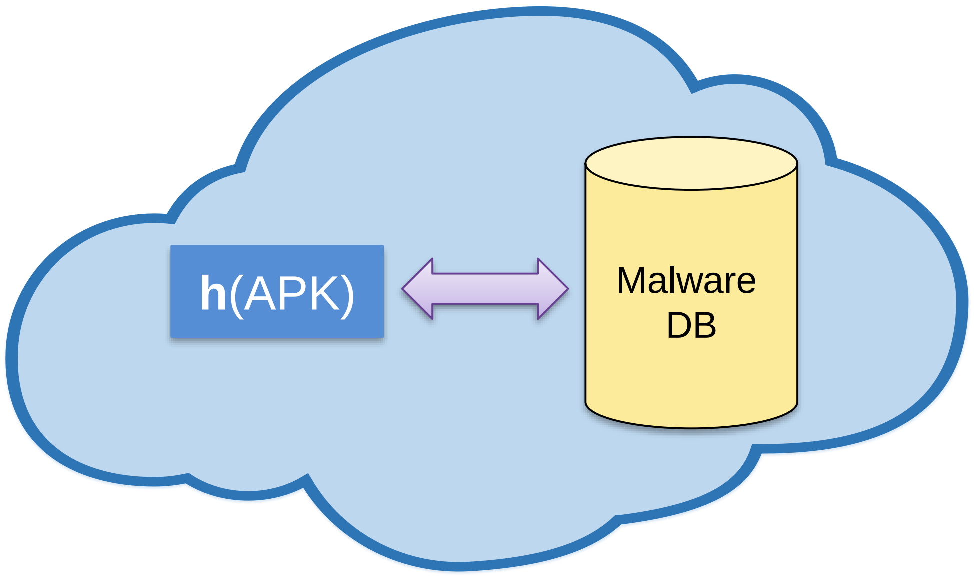


On-device checking

- High communication and computation costs
- Database changes frequently
- Database is revealed to everyone

Malware Checking





On-device checking

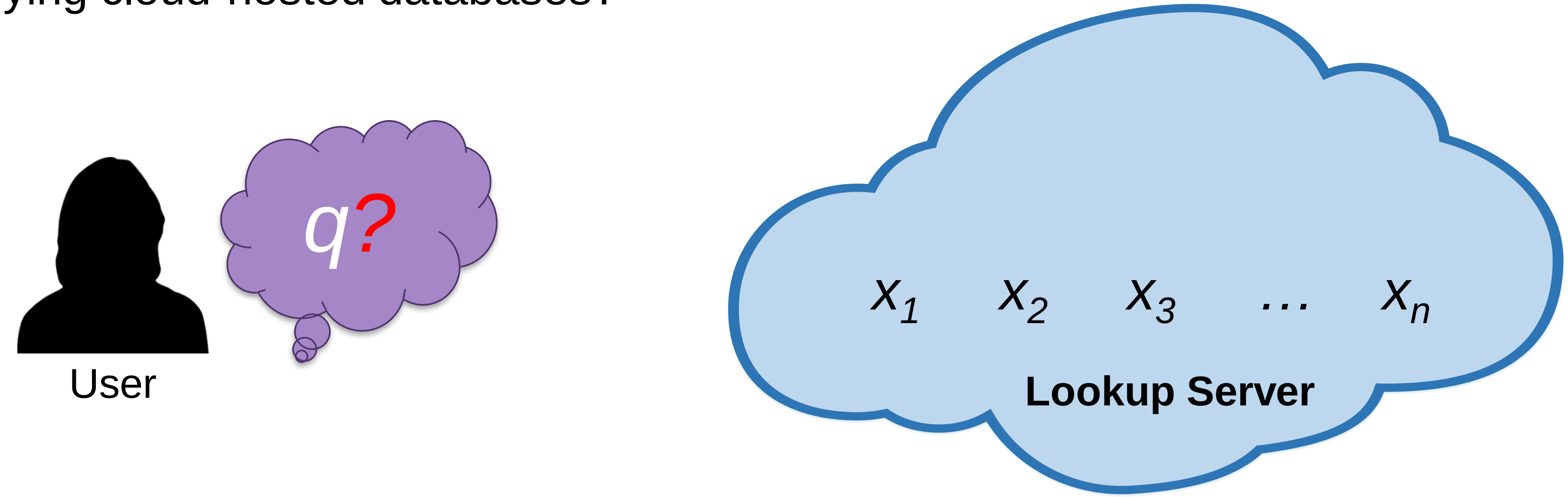
- High communication and computation costs
- Database changes frequently
- Database is revealed to everyone

Cloud-based checking

- Minimal communication and computation costs
- Database can change frequently
- Database is not revealed to everyone
- User privacy at risk!

Private Membership Test

The problem: How to preserve end user privacy when querying cloud-hosted databases?



Server must not learn contents of client query (q).

Current solutions (e.g. private set intersection, private information retrieval):

- Single server: expensive in both computation and/or communication
- Multiple independent servers: unrealistic in commercial setting

Private Membership Test with Trusted Hardware

Trusted Execution Environments (TEEs) are ubiquitous

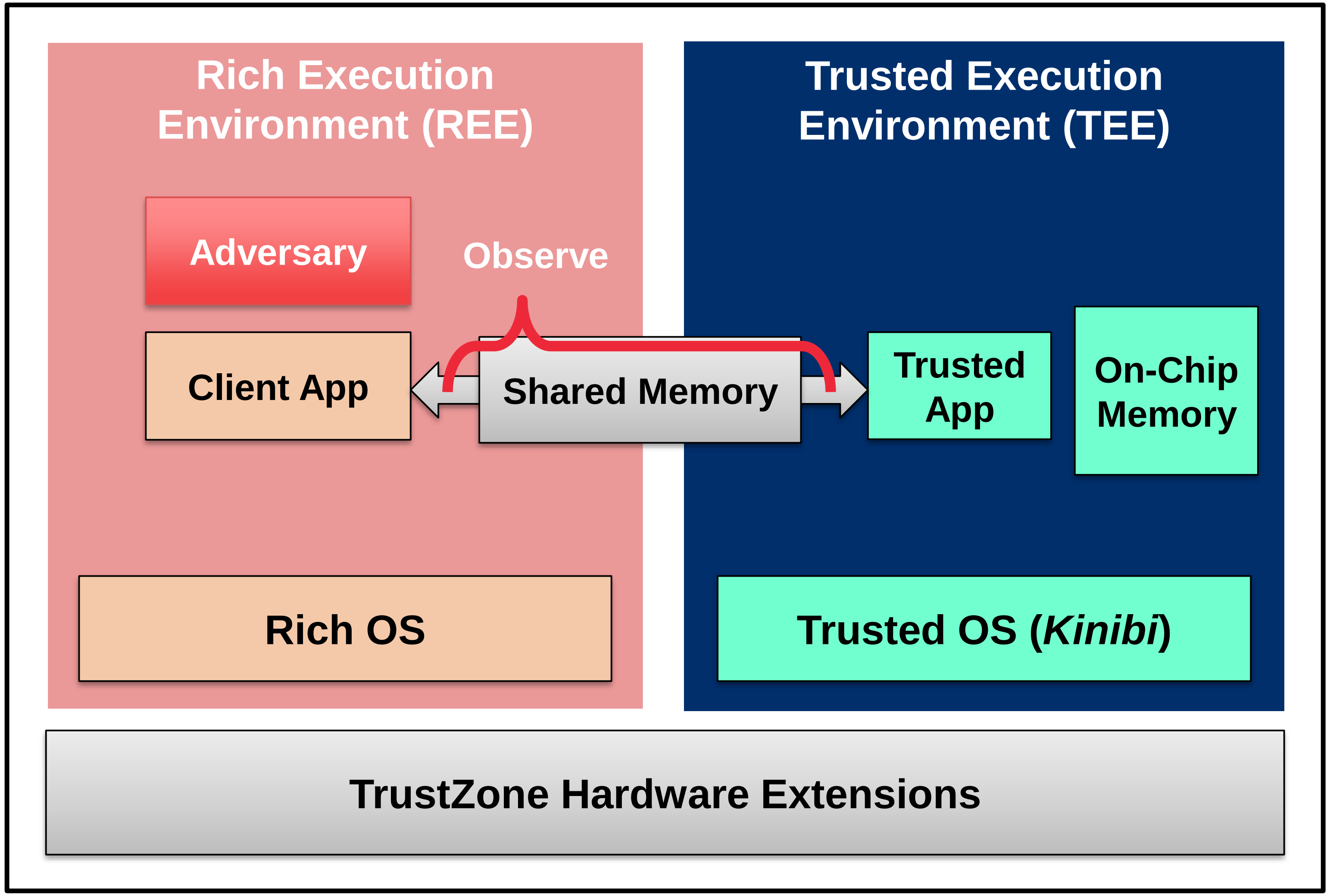
- ARM TrustZone, Intel SGX, …

Can TEEs provide a practical solution for Private Membership Test?

Ekberg, Kostiainen, Asokan "Untapped potential of trusted execution environments", IEEE S&P 2014

Background: Kinibi on ARM TrustZone

Trusted
Untrusted



Kinibi

- Trusted OS from Trustonic

Remote attestation

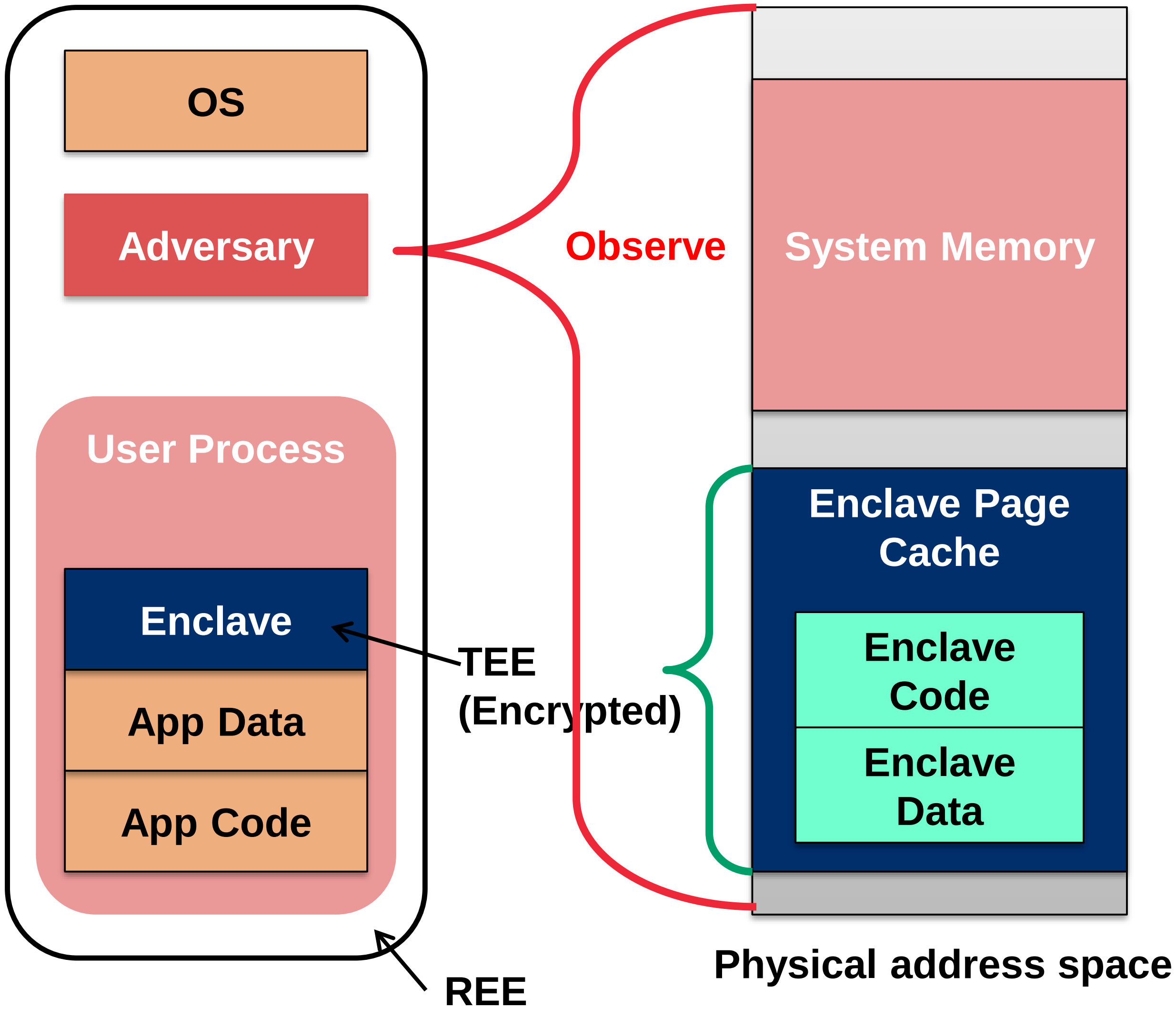
- Establish a trusted channel

Private memory

- Confidentiality
- Integrity
- *Obliviousness*

https://www.trustonic.com/solutions/trustonic-secured-platforms-tsp/

Background: Intel SGX



CPU enforced TEE (*enclave*)

Remote attestation

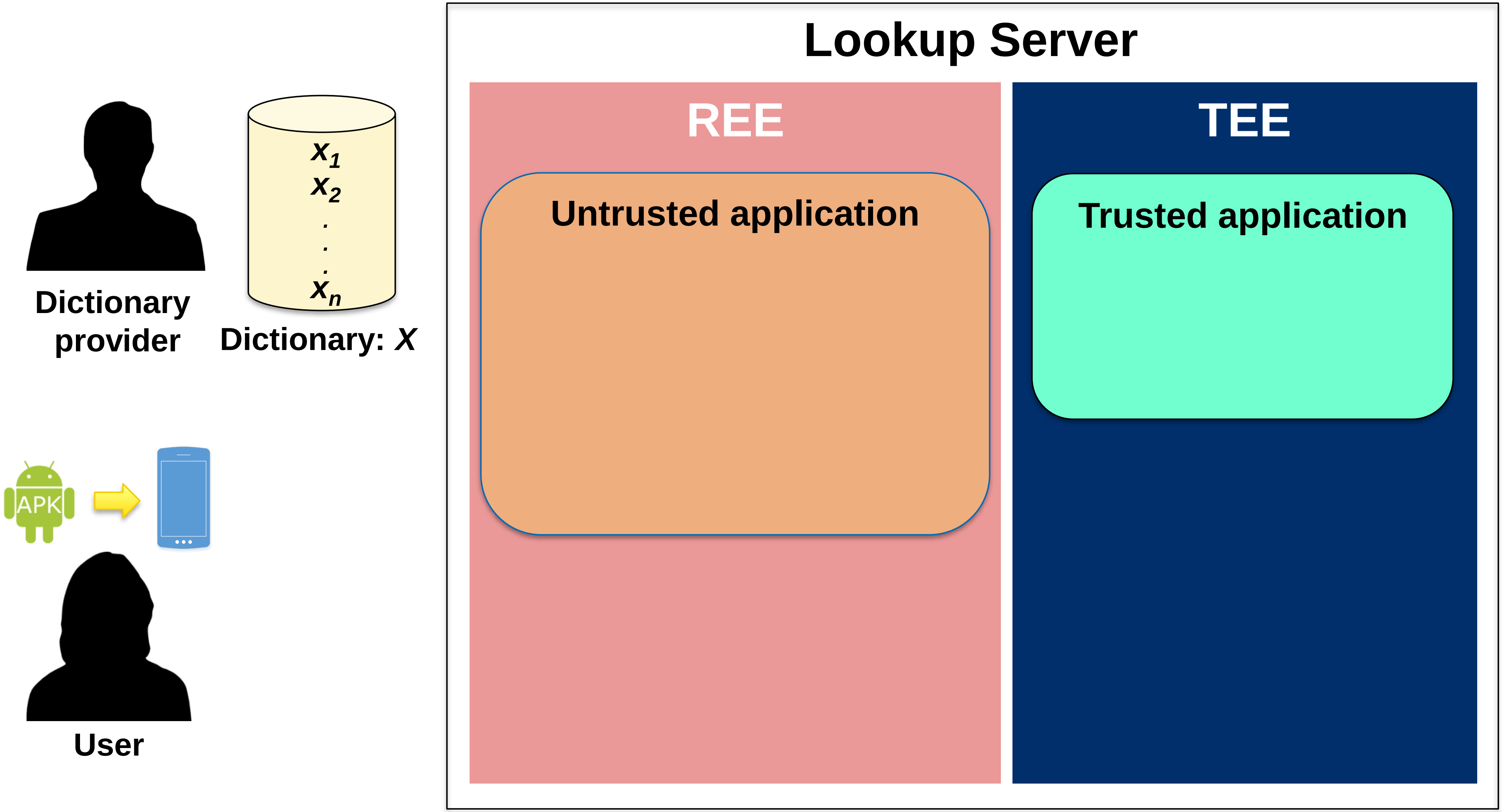
Secure memory

- Confidentiality
- Integrity

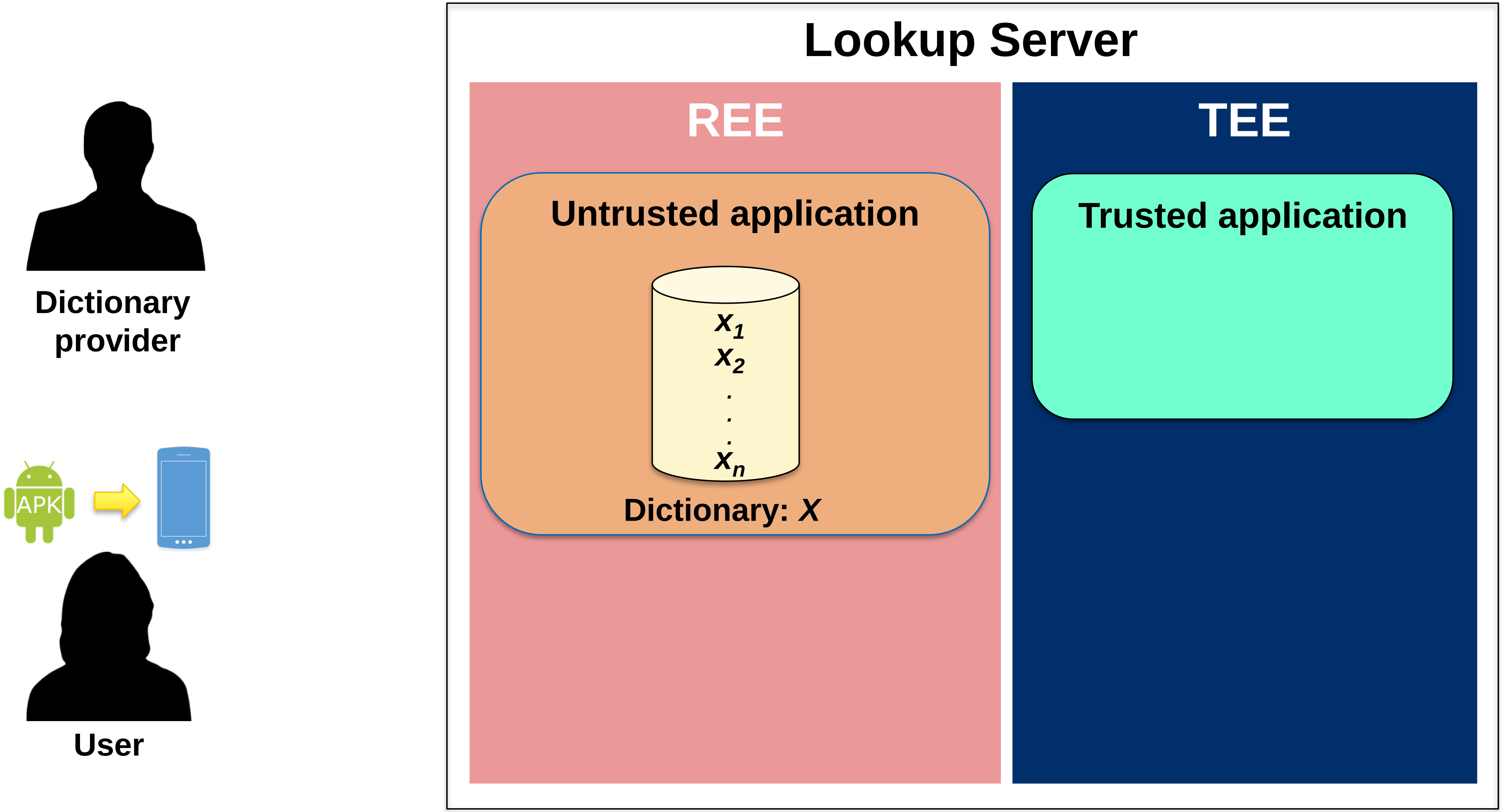
Obliviousness only within 4 KB page granularity

https://software.intel.com/sgx

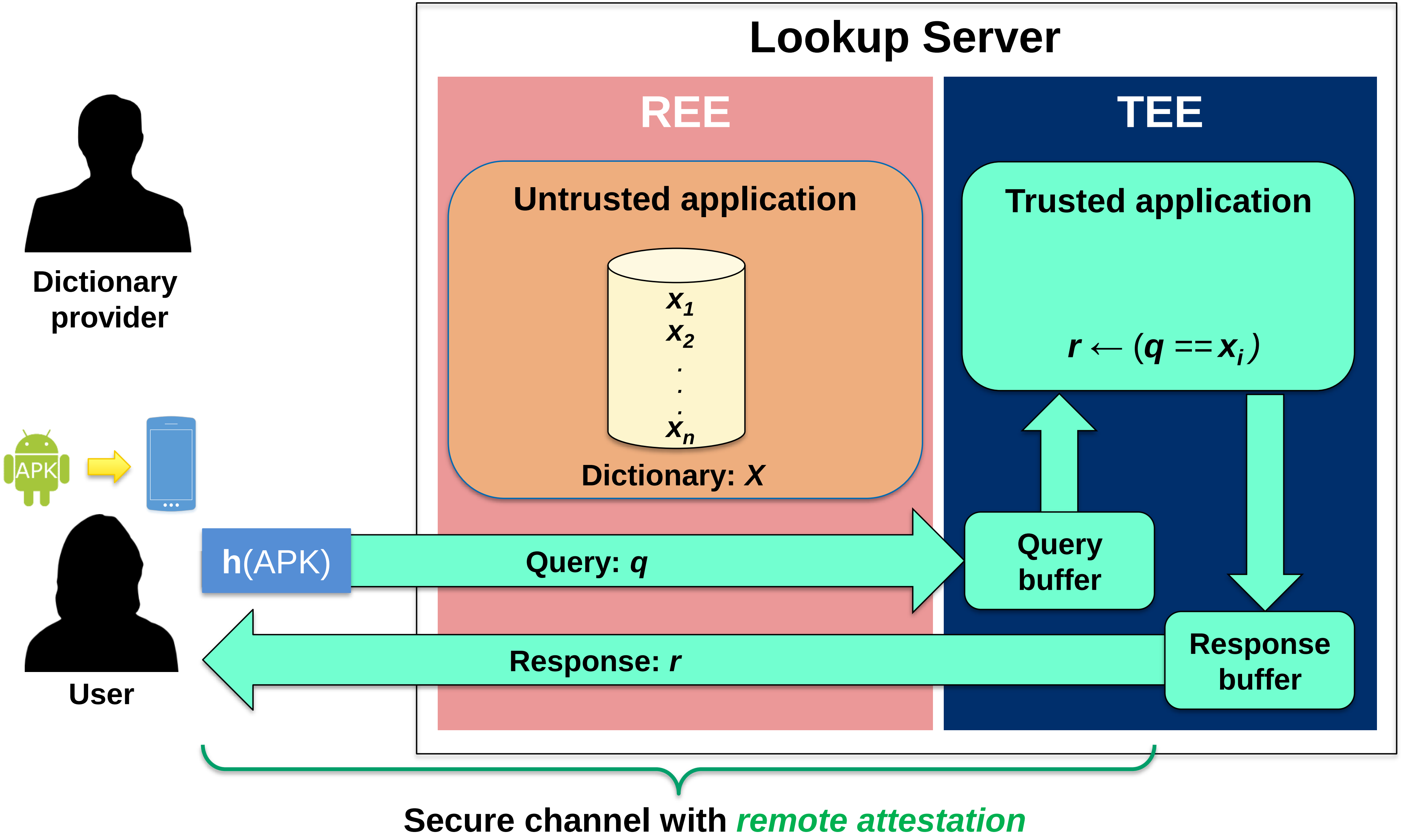
System Model



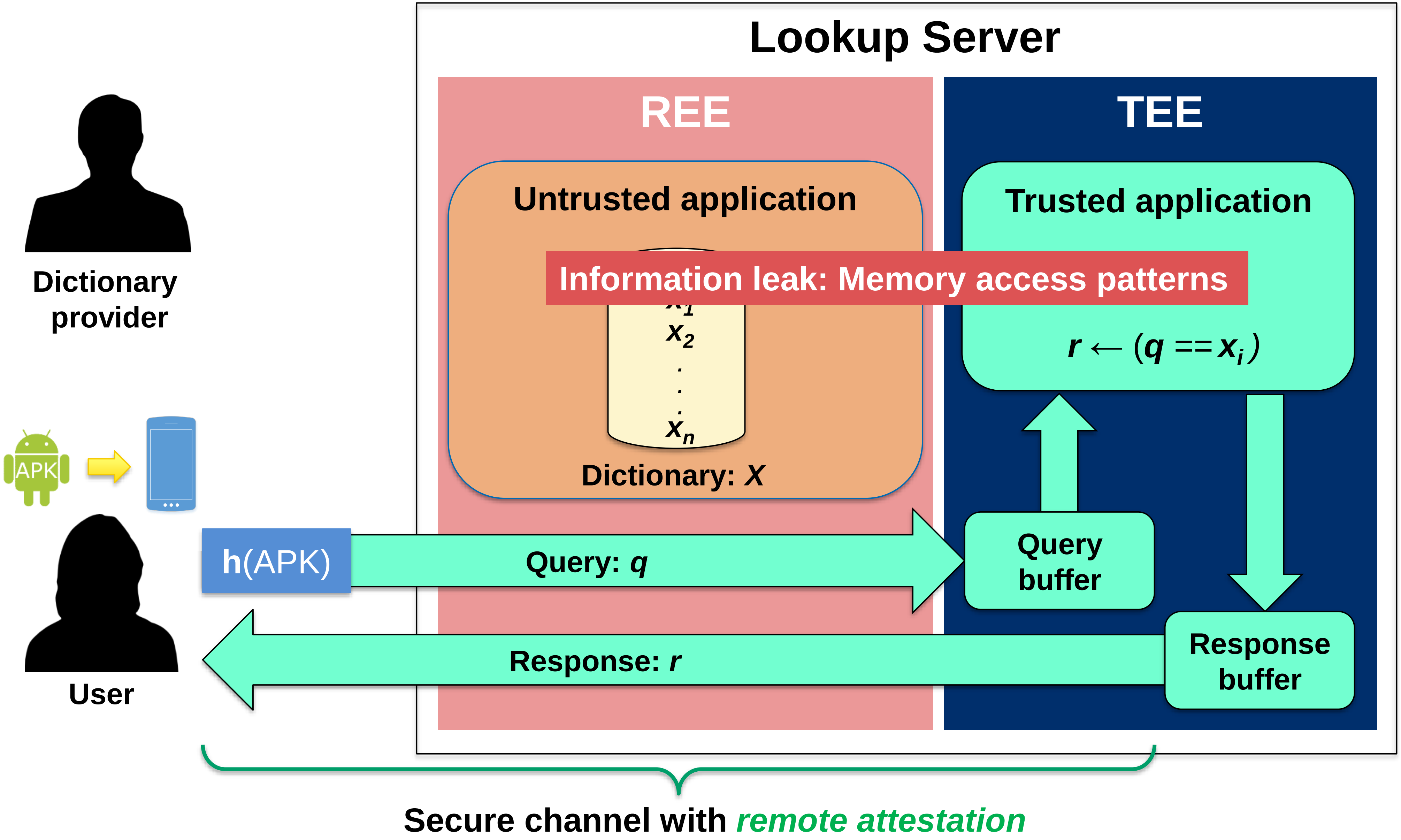
System Model



System Model



System Model



Requirements

Query Privacy: Adversary cannot learn/infer query or response content

- User can always choose to reveal query content

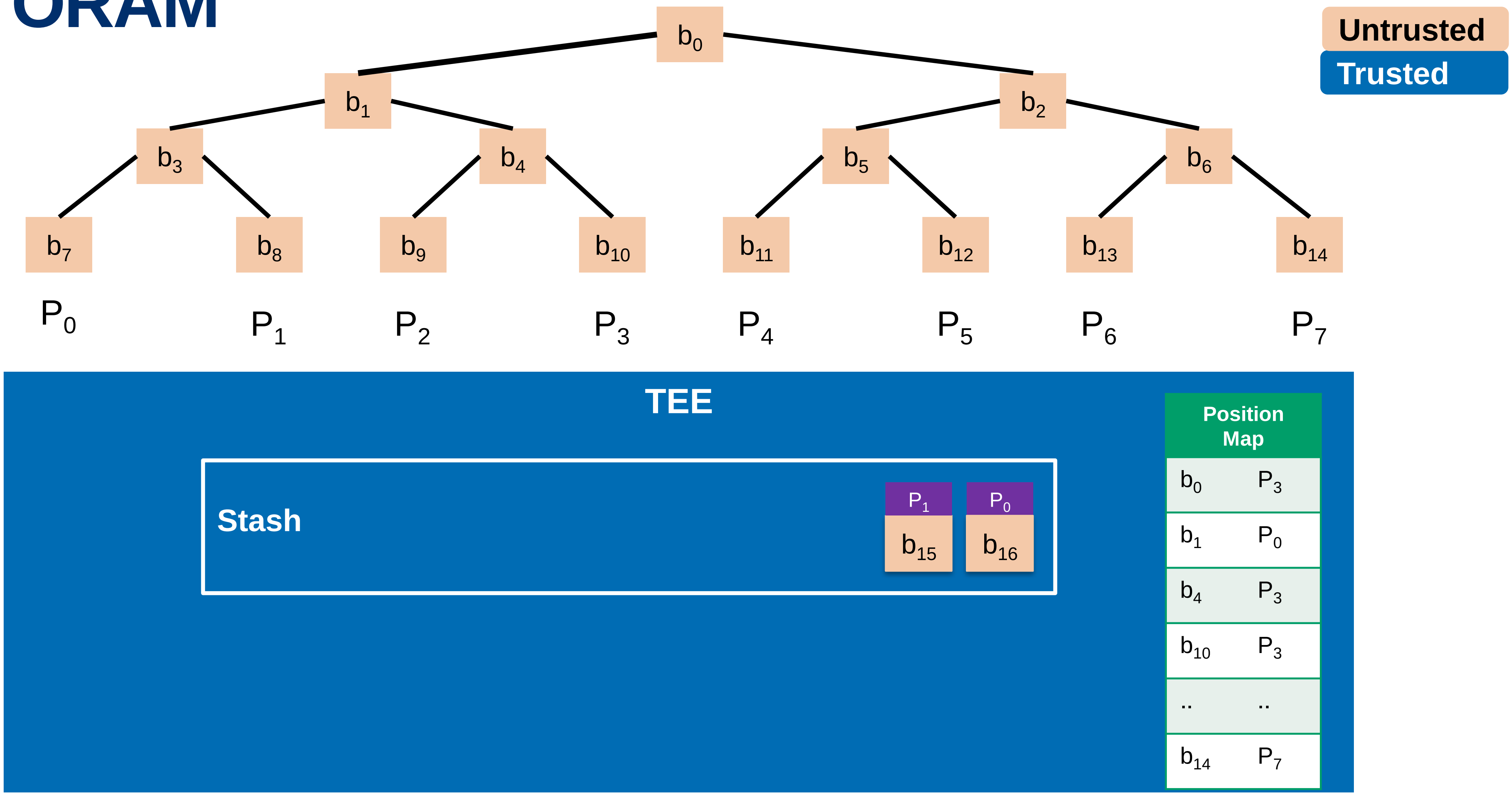
Accuracy: No false negatives

- However, some false positives are tolerable (i.e. non-zero false positive rate)

Response Latency: Respond quickly to each query

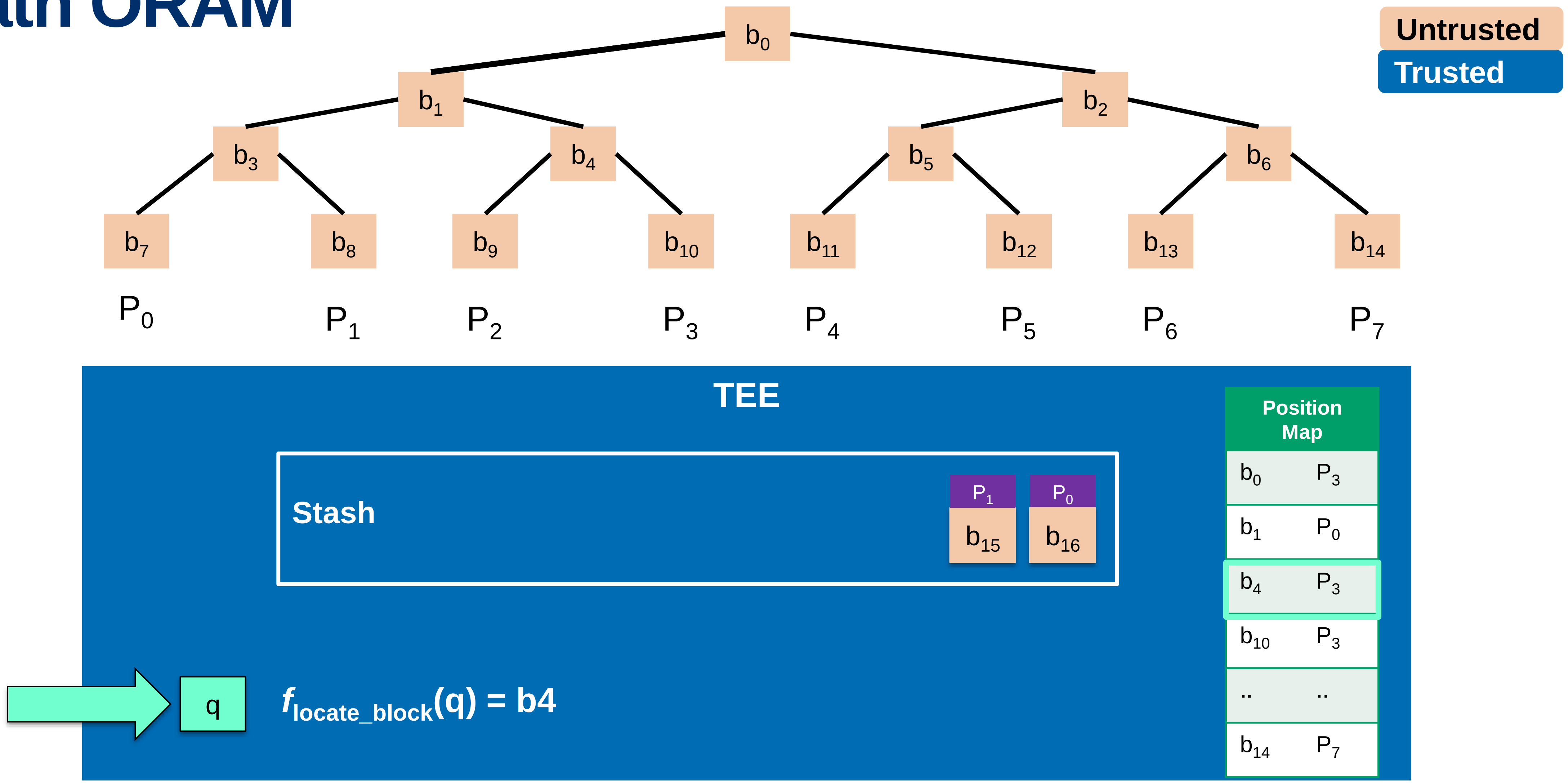
Server Scalability: Maximize overall throughput (queries per second)

Path ORAM



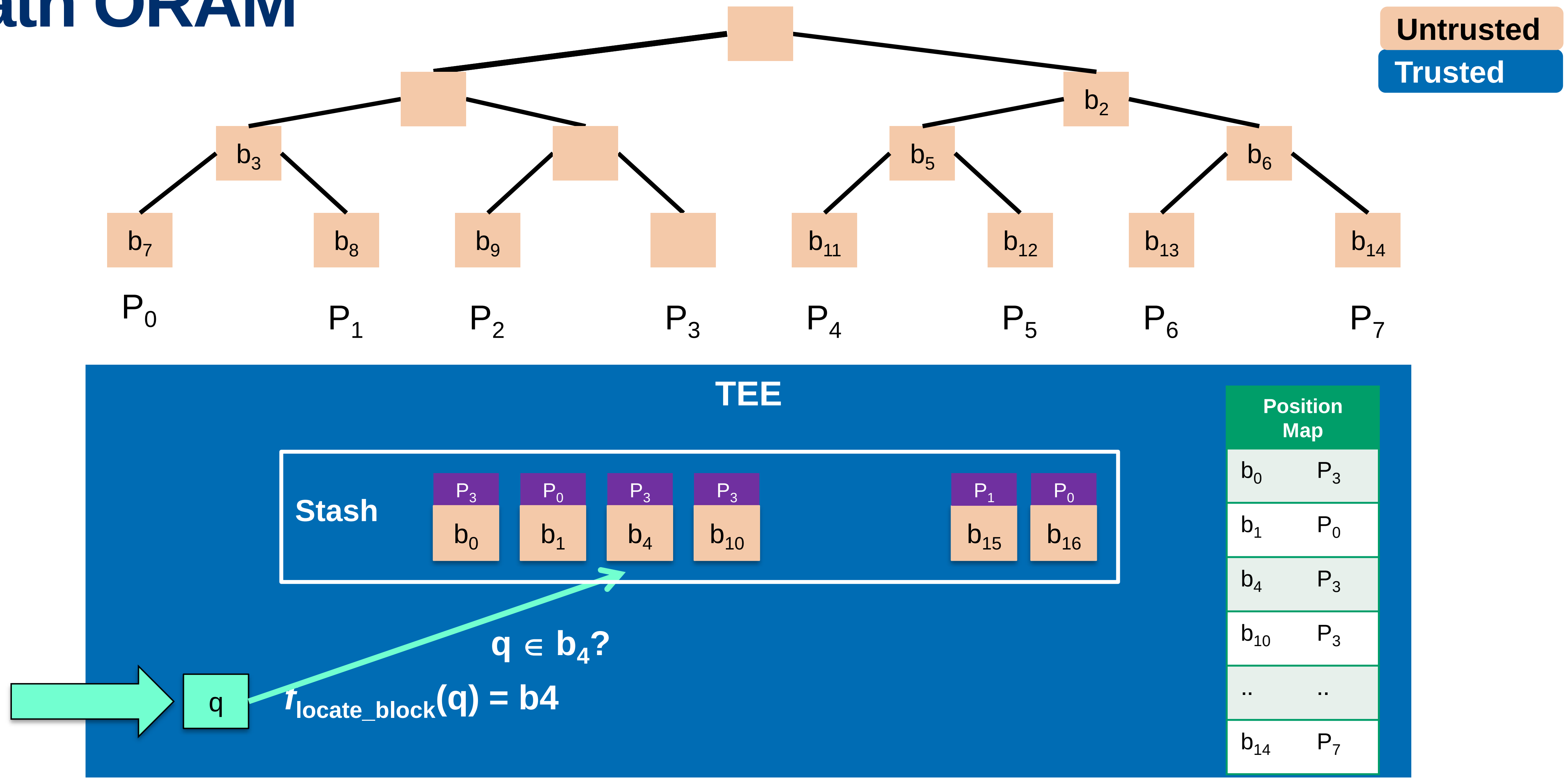
Stefanov et al. ACM CCS 2013, https://dl.acm.org/citation.cfm?id=2516660

Path ORAM



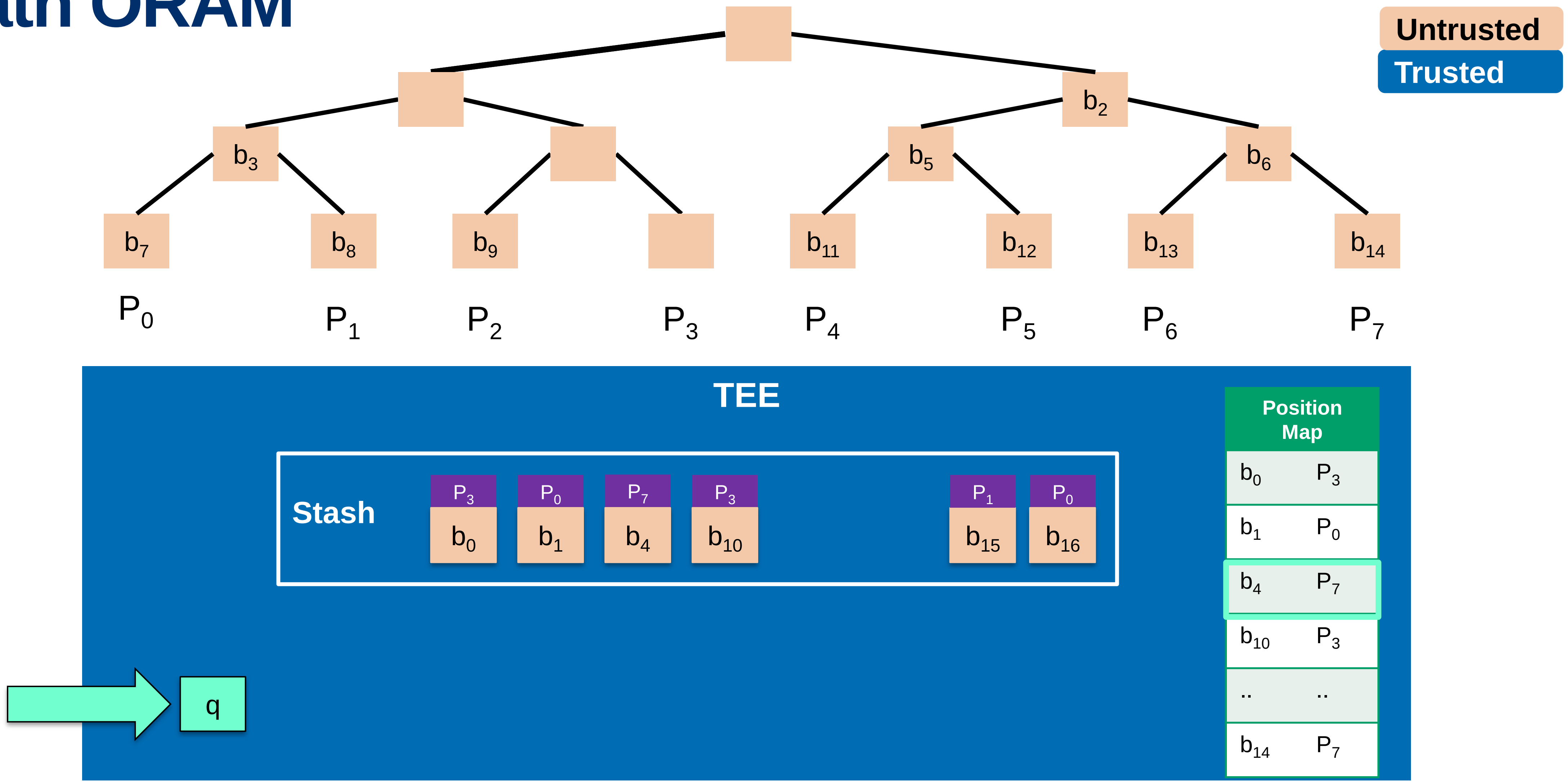
Stefanov et al. ACM CCS 2013, https://dl.acm.org/citation.cfm?id=2516660

Path ORAM



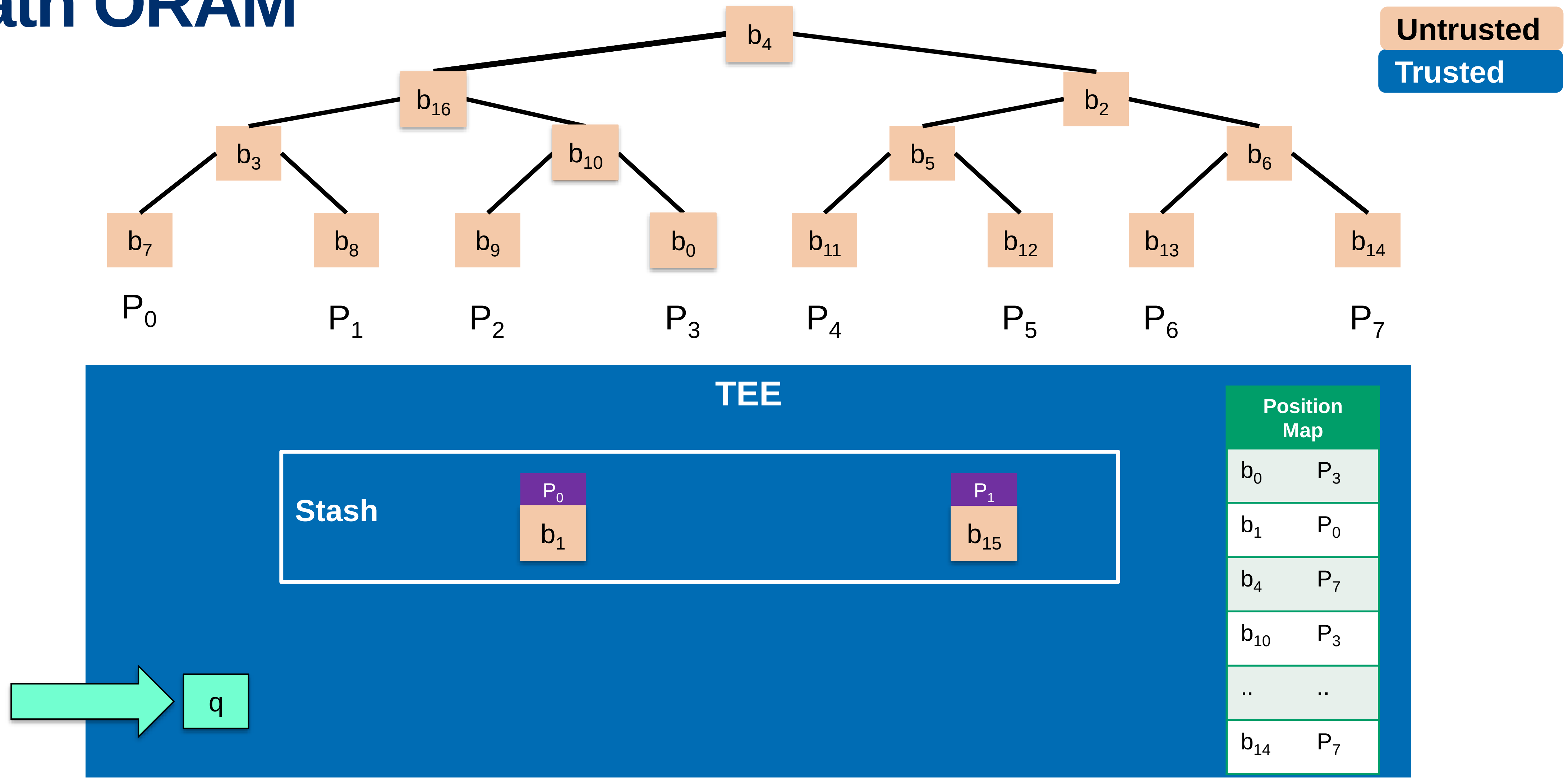
Stefanov et al. ACM CCS 2013, https://dl.acm.org/citation.cfm?id=2516660

Path ORAM



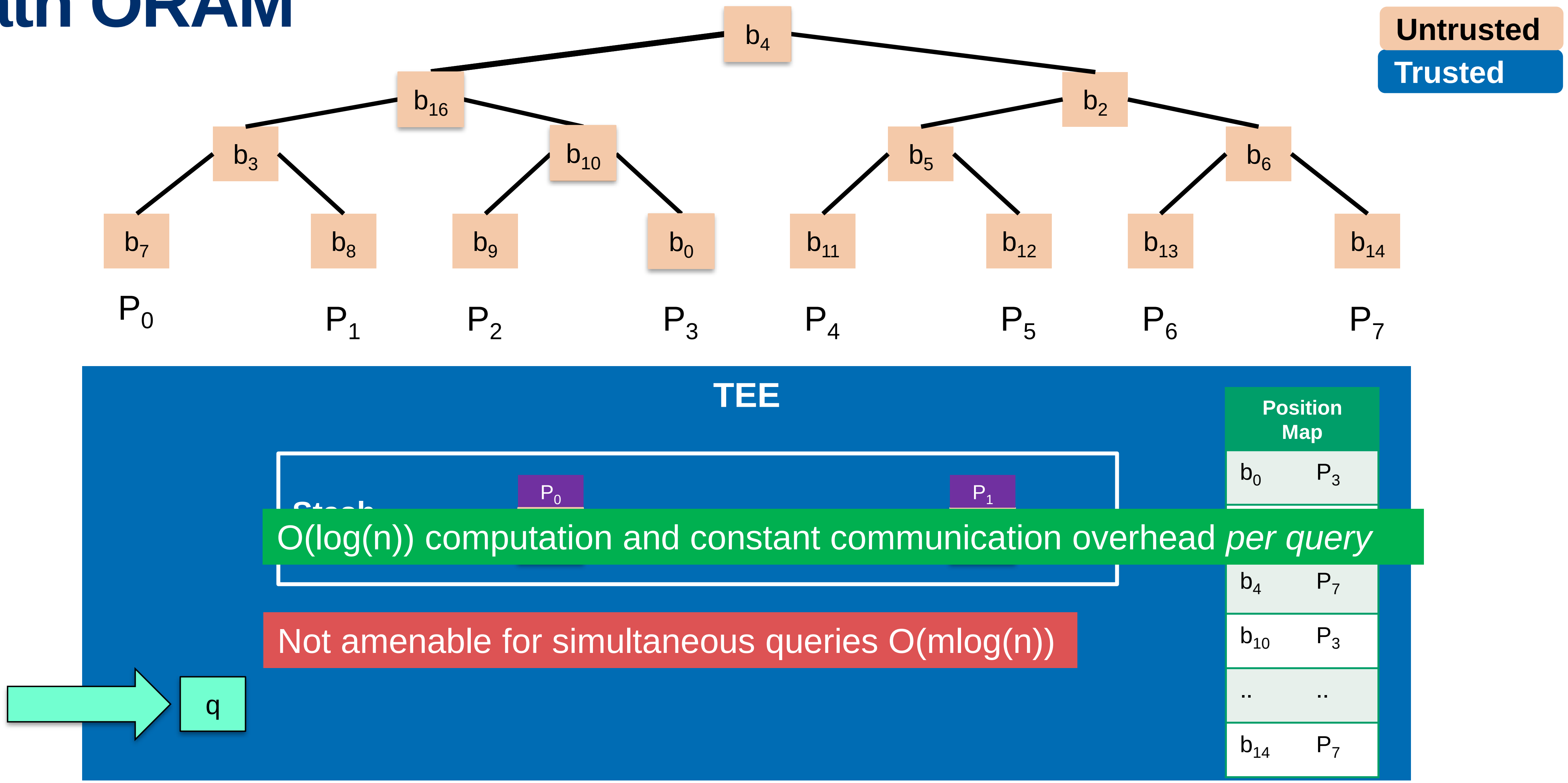
Stefanov et al. ACM CCS 2013, https://dl.acm.org/citation.cfm?id=2516660

Path ORAM



Stefanov et al. ACM CCS 2013, https://dl.acm.org/citation.cfm?id=2516660

Path ORAM

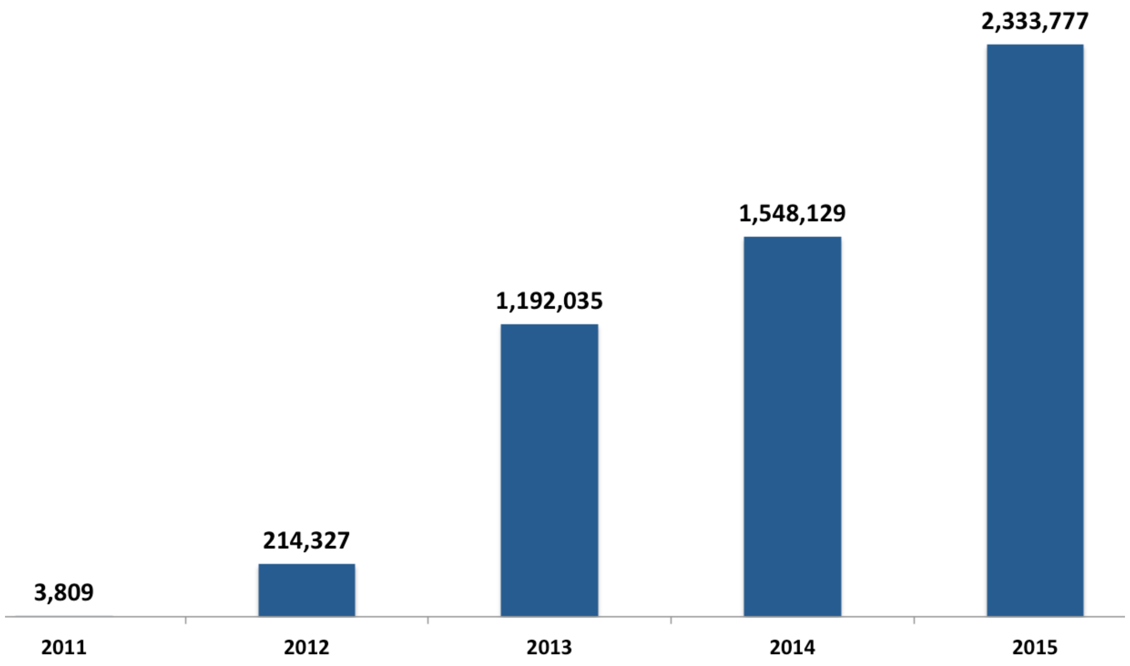


O(log(n)) computation and constant communication overhead *per query*

Not amenable for simultaneous queries O(mlog(n))

Stefanov et al. ACM CCS 2013, https://dl.acm.org/citation.cfm?id=2516660

Android App Landscape



Unique Android malware samples
Source: G Data https://secure.gd/dl-en-mmwr201504

On average a user installs 95 apps (Yahoo Aviate)
Yahoo Aviate study
Source: https://yahooaviate.tumblr.com/image/95795838933

Current dictionary size $< 2^{22}$ entries

Even comparatively "high" FPR (e.g., $\sim 2^{-10}$) may have negligible impact on privacy

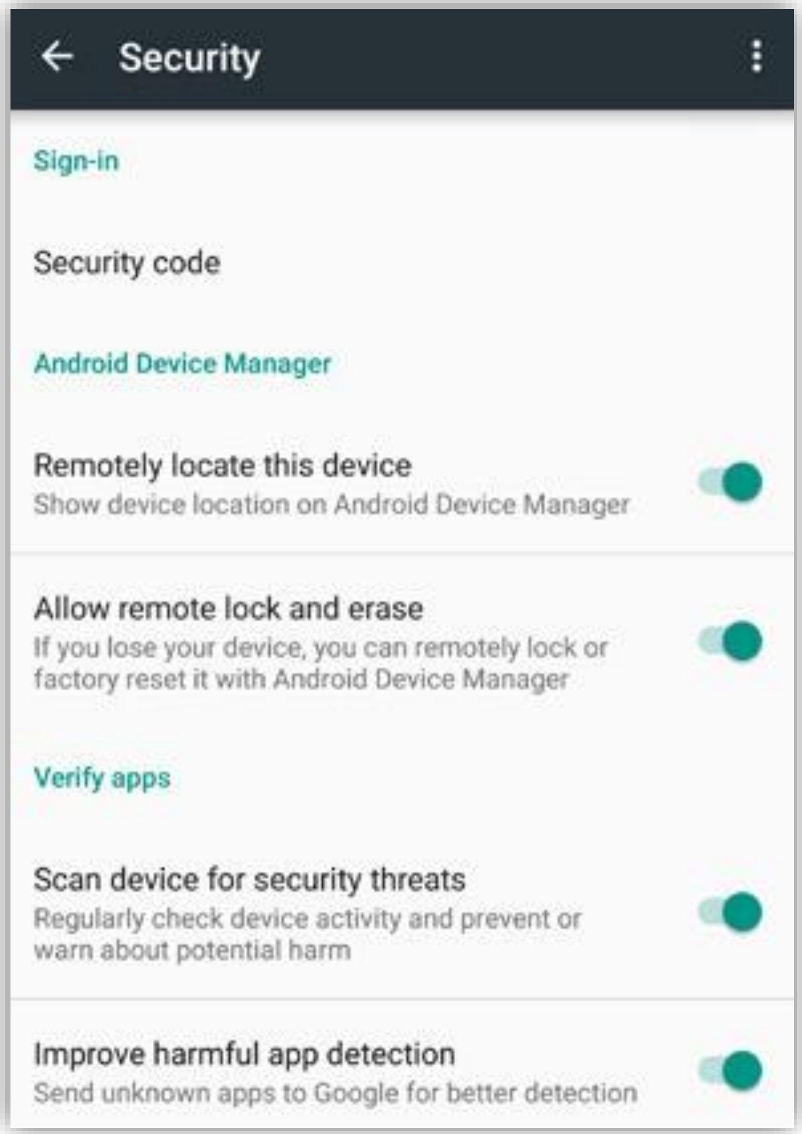
Cloud Scale PMT

***Verify Apps*: cloud-based service to check for harmful Android apps prior to installation**

"… over 1 billion devices protected by Google's security services, and over 400 million device security scans were conducted per day"

[Android Security 2015 Year in Review]

(c.f. ~2 million malware samples)



Requirements Revisited

Query Privacy: Adversary cannot learn/infer query or response content

- User can always choose to reveal query content

Accuracy: No false negatives

- However, some false positives are tolerable (i.e. non-zero false positive rate)

FPR* = 2^{-10}

Response Latency: Respond quickly to each query

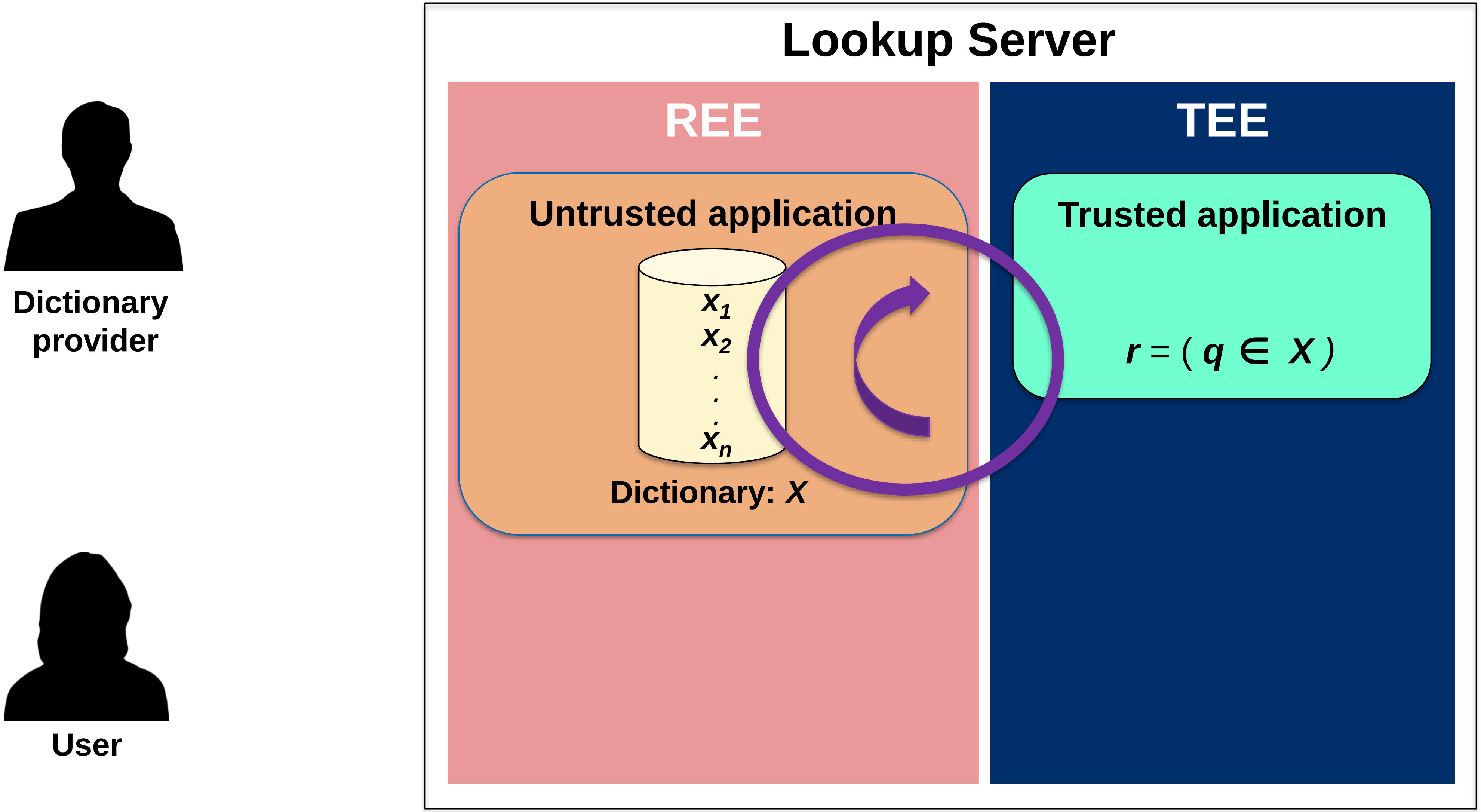
Latency* ~ 1s

Server Scalability: Maximize overall throughput (queries per second)

Dictionary size* = 2^{26} entries (~ 67 million entries)

**** parameters suggested by a major anti-malware vendor***

Carousel Approach



Carousel Caveats

1. **Adversary can measure dictionary processing time**
   - Spend equal time processing each dictionary entry

2. **Adversary can measure query-response time**
   - Only respond after one full carousel cycle

Both impact response latency (recall Requirements)

Therefore, aim to *minimize carousel cycle time*

How to Minimize Cycle Time?

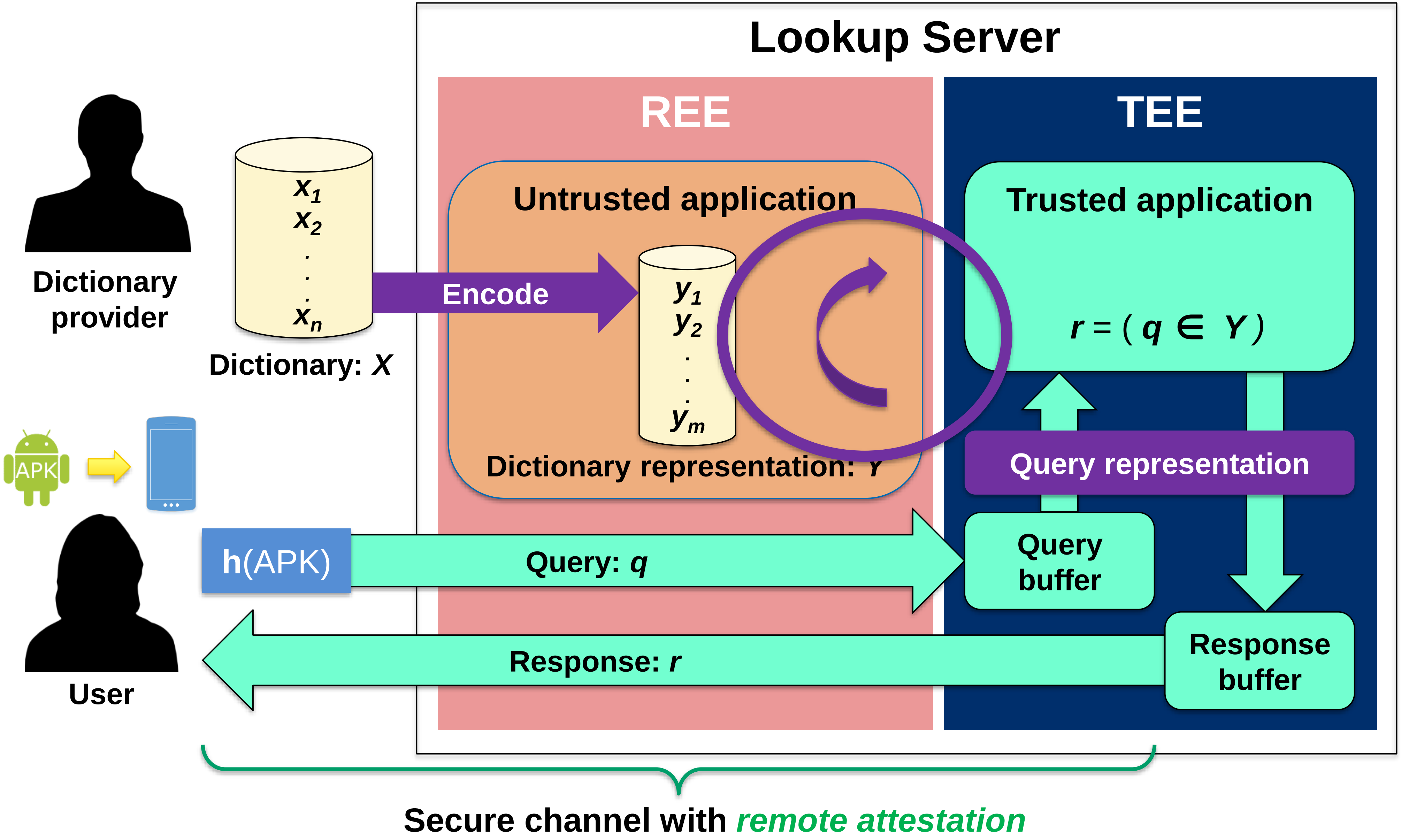
Represent dictionary using efficient data structure

Various existing data structures support membership test:

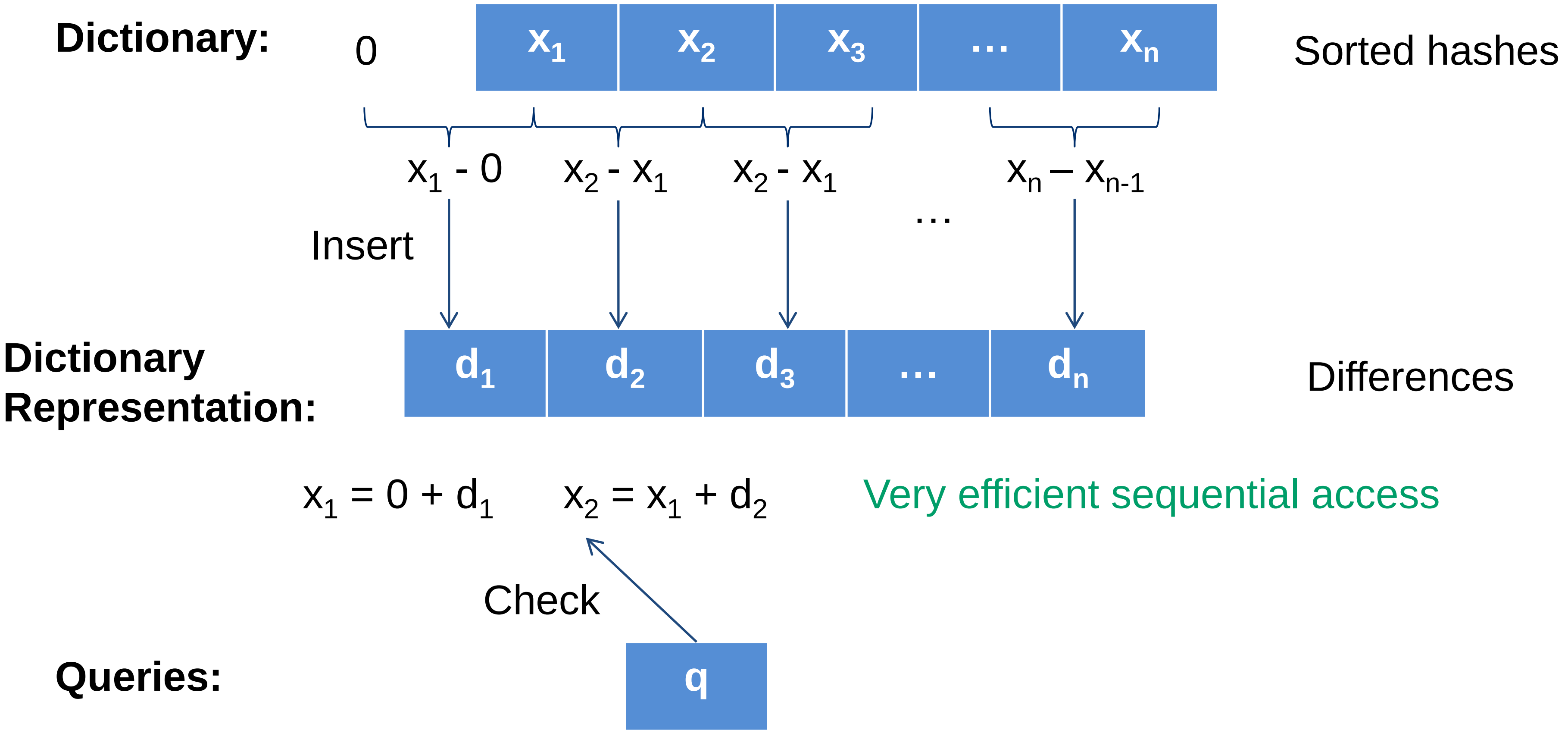
- Bloom Filter
- Cuckoo hash

Experimental evaluation required for carousel approach

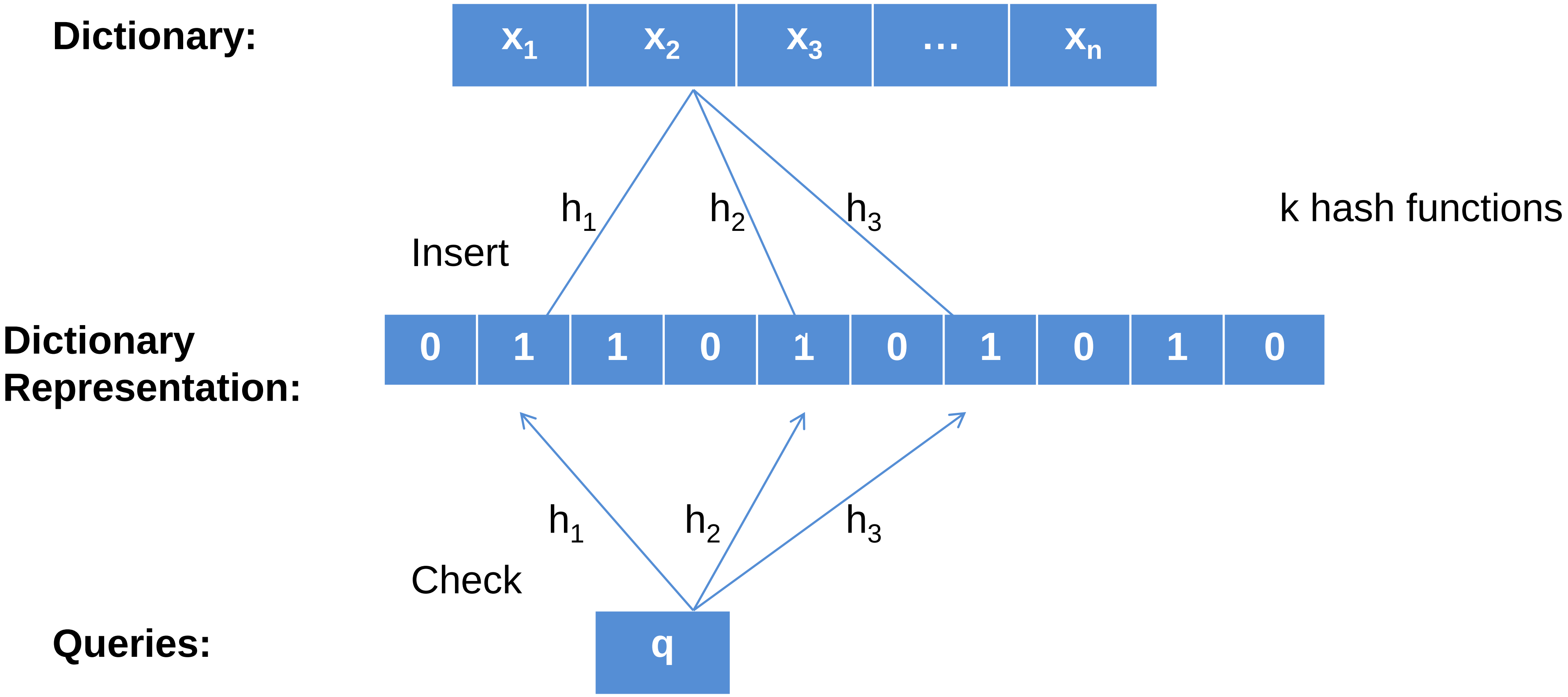
Carousel Approach



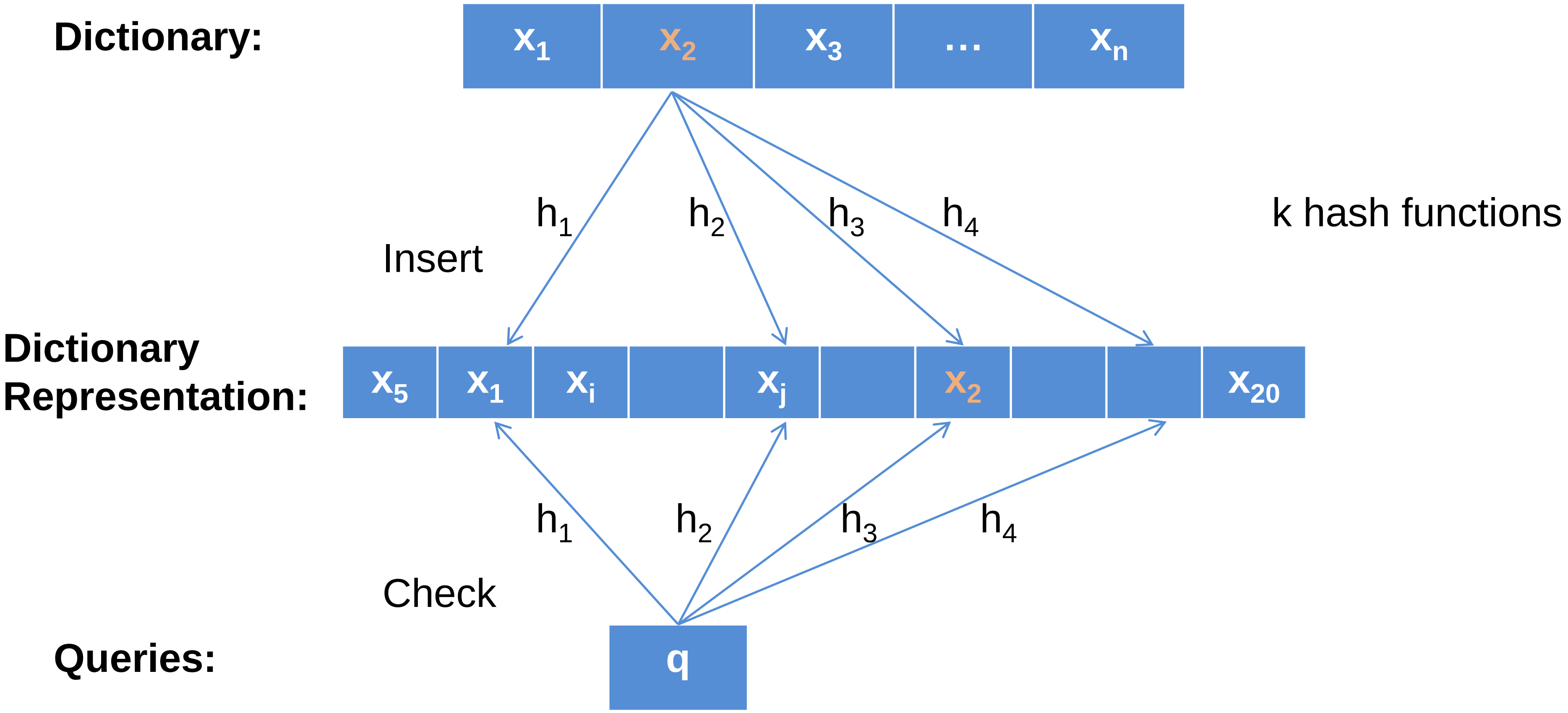
Sequence of differences



Bloom Filter



Cuckoo Hash


Dictionary:
x_1 x_2 x_3 … x_n
h_1 h_2 h_3 h_4
Insert
k hash functions
Dictionary Representation:
x_5 x_1 x_i x_j x_2 x_{20}
h_1 h_2 h_3 h_4
Check
Queries:
q

Experimental Evaluation

Kinibi on ARM TrustZone

- Samsung Exynos 5250 (Arndale)
- 1.7 GHz dual-core ARM Cortex-A17
- Android 4.2.1
- ARM GCC compiler and Kinibi libraries
- Maximum TA private memory: 1 MB
- Maximum shared memory: 1 MB

Intel SGX

- HP EliteDesk 800 G2 desktop
- 3.2 GHz Intel Core i5 6500 CPU
- 8 GB RAM
- Windows 7 (64 bit), 4 KB page size
- Microsoft C/C++ compiler
- Intel SGX SDK for Windows

Experimental Evaluation

Kinibi on ARM TrustZone

- Samsung Exynos 5250 (Arndale)
- 1.7 GHz dual-core ARM Cortex-A17
- Android 4.2.1
- ARM GCC compiler and Kinibi libraries
- Maximum TA private memory: 1 MB
- Maximum shared memory: 1 MB

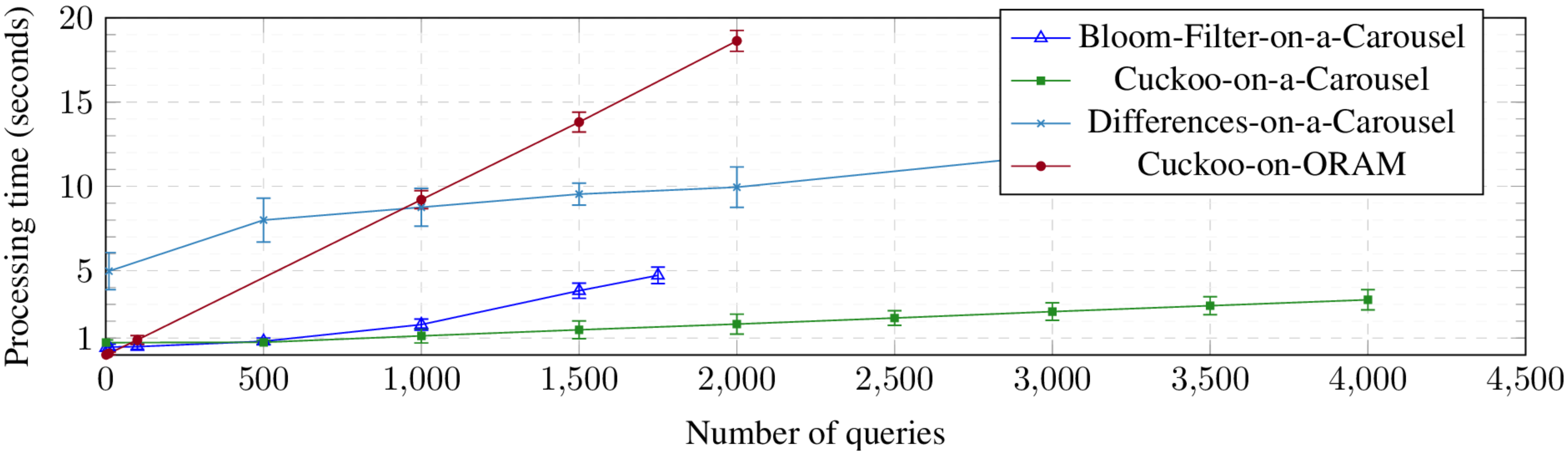
Intel SGX

- HP EliteDesk 800 G2 desktop
- 3.2 GHz Intel Core i5 6500 CPU
- 8 GB RAM
- Windows 7 (64 bit), 4 KB page size
- Microsoft C/C++ compiler
- Intel SGX SDK for Windows

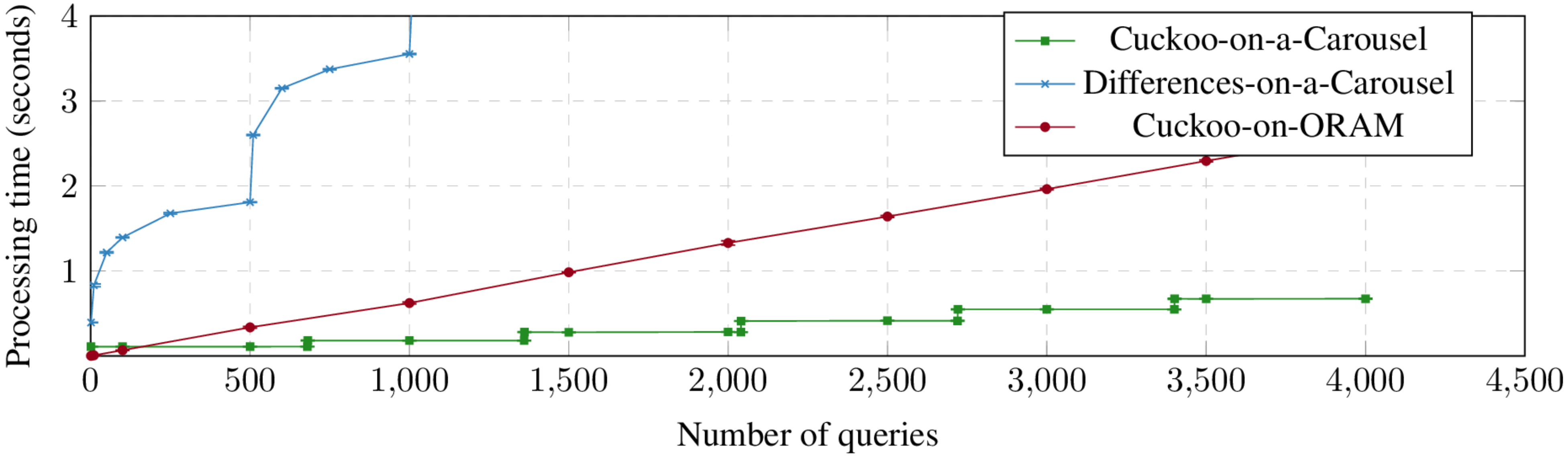
Note: Different CPU speeds and architectures

Performance: Batch Queries

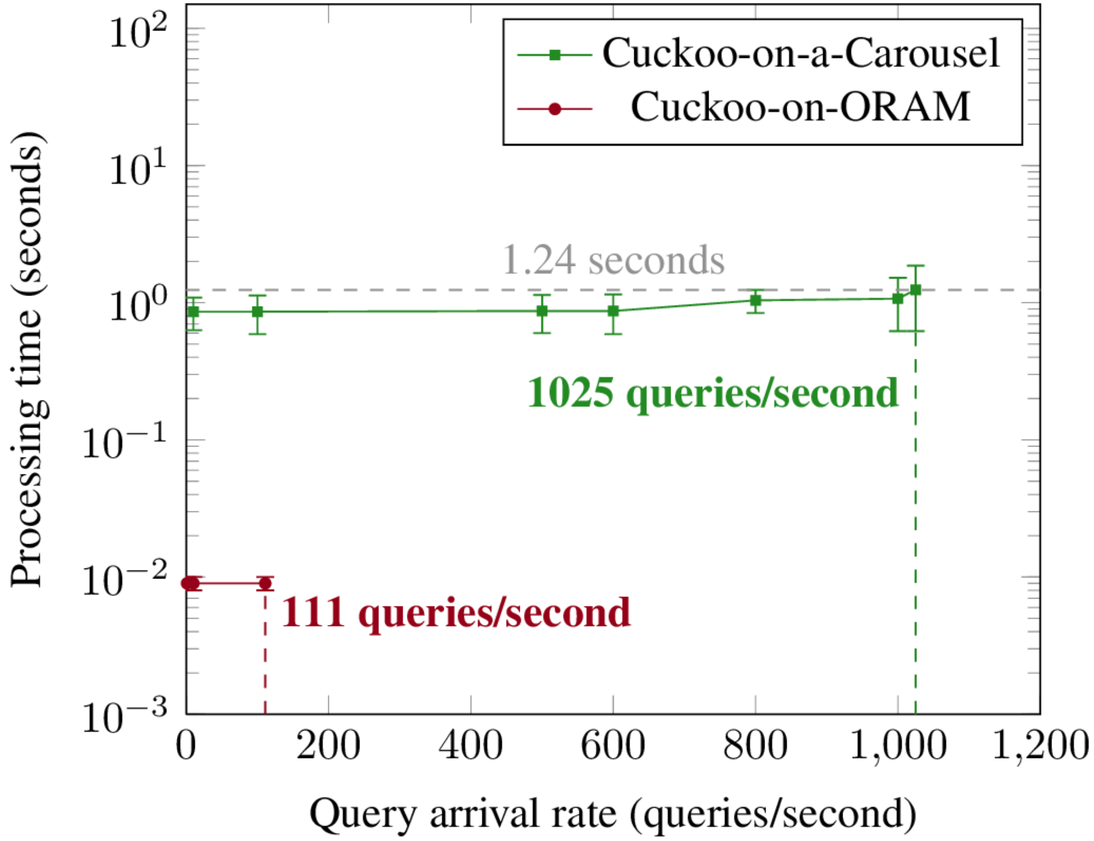
Kinibi on ARM TrustZone



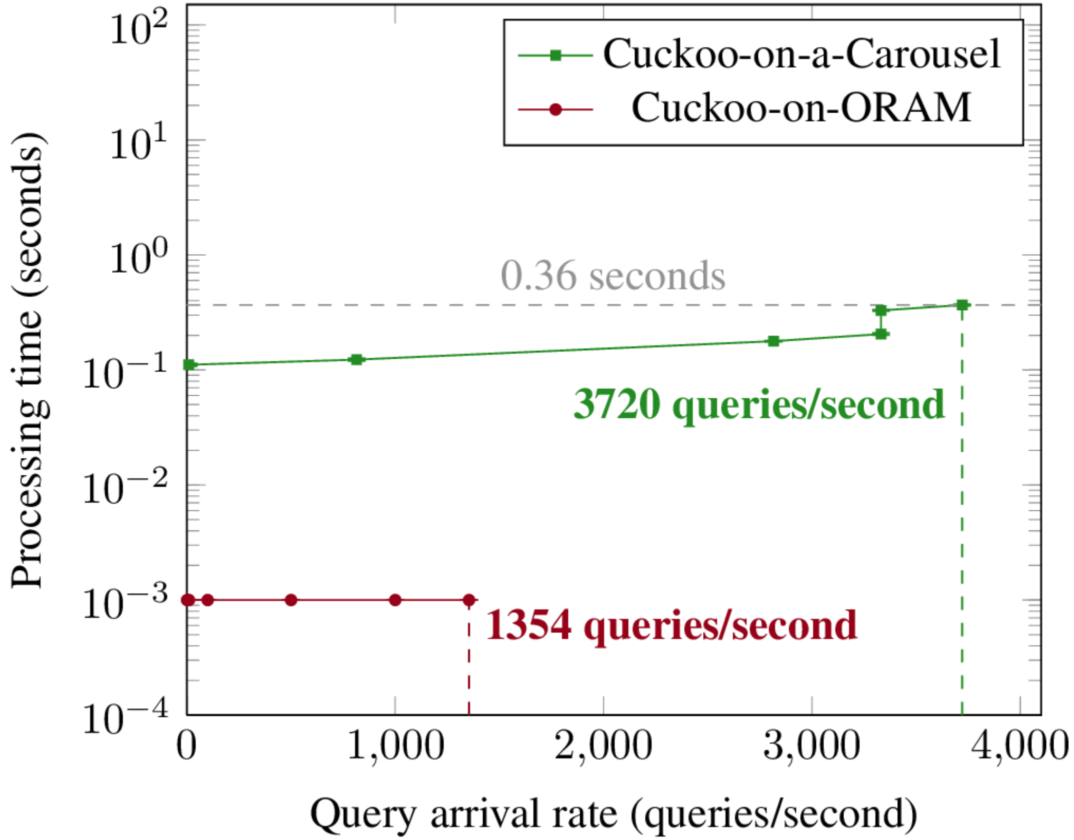
Intel SGX



Performance: Steady State Query Arrival

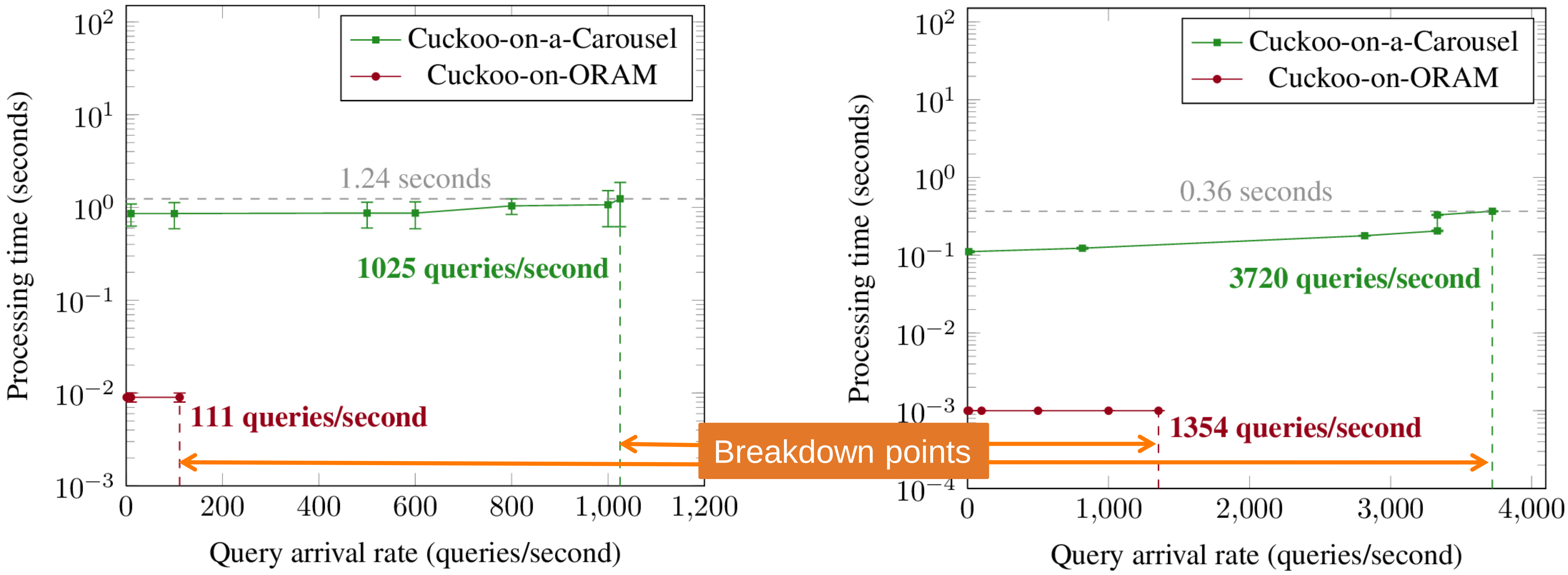


Kinibi on ARM TrustZone



Intel SGX

Performance: Steady State Query Arrival



Kinibi on ARM TrustZone **Intel SGX**

Beyond ***breakdown point*** query response latency increases over time

Evaluation Summary

Cuckoo hash provides best performance

Average response latency

	Kinibi on ARM TrustZone	Intel SGX
Cuckoo on (Path) ORAM	0.009 s	0.001 s
Cuckoo on a Carousel	1.240 s	0.360 s

Sustainable query throughput

	Kinibi on ARM TrustZone	Intel SGX
Cuckoo on (Path) ORAM	111 q/s	1354 q/s
Cuckoo on a Carousel	1025 q/s	3720 q/s

Private Contact Discovery in Signal

Signal SUPPORT BLOG DEVELOPERS JOBS

Technology preview: Private contact discovery for Signal

moxie0 on 26 Sep 2017

At Signal, we've been thinking about the difficulty of private contact discovery for a long time. We've been working on strategies to improve our current design, and today we've published a new private contact discovery service.

Using this service, Signal clients will be able to efficiently and scalably determine whether the contacts in their address book are Signal users **without revealing the contacts in their address book to the Signal service**.

Background

Signal is social software. We don't believe that privacy is about austerity, or that a culture of sharing and communication should mean that privacy is a thing of the past. We want to enable online social interactions that are rich and expressive in all the ways that people desire, while simultaneously ensuring that communication is only visible to its intended recipients.

https://signal.org/blog/private-contact-discovery/

Private Contact Discovery in Signal

“An SGX enclave on the server-side would enable a service to perform computations on encrypted client data *without learning the content of the data or the result of the computation*.”

“Private contact discovery using SGX is fairly simple at a high level:

1. Run a contact discovery service in a secure SGX enclave.
2. Clients that wish to perform contact discovery negotiate a secure connection over the network all the way through the remote OS to the enclave.
3. Clients perform remote attestation to ensure that the code which is running in the enclave is the same as the expected published open source code.
4. Clients transmit the encrypted identifiers from their address book to the enclave.
5. The enclave looks up a client’s contacts in the set of all registered users and encrypts the results back to the client.”

https://signal.org/blog/private-contact-discovery/

Private Contact Discovery in Signal

“Unfortunately, doing private computation in an SGX enclave is more difficult than it may initially seem.”

“However, the host OS can still see *memory access patterns*, even if the OS can’t see the contents of the memory being accessed.”

“This class of problems has been studied under the discipline of Oblivious RAM (ORAM).”

“There are some elegant generalized ORAM techniques, like Path ORAM, but unfortunately they don’t work well for this problem.”

https://signal.org/blog/private-contact-discovery/

Private Contact Discovery in Signal

“By keeping one big linear scan over the registered user data set, access to unencrypted RAM remains “oblivious,” since the OS will simply see the enclave touch every item once for each contact discovery request.”

“The full linear scan is fairly high latency, but by batching many pending client requests together, it can be high throughput.”

https://signal.org/blog/private-contact-discovery/

Conclusions

Cloud-assisted services raise new security/privacy concerns

- But naïve solutions may conflict with privacy, usability, deployability, …

http://arxiv.org/abs/1606.01655

Cloud-assisted malware scanning

- Carousel approach is promising
- Implementing ORAM on SGX is hard!

In future

- Efficient oblivious data structures for trusted hardware
- New use cases for Carousel (e.g. leaked passwords, …)

ASIACCS 2017
Sandeep Tamrakar, Jian Liu, Andrew Paverd, Jan-Erik Ekberg, Benny Pinkas, N. Asokan.
The Circle Game: Scalable Private Membership Test Using Trusted Hardware