Private SCT Auditing, Revisited

Lena Heimberger[1], Bas Westerbaan[2]

Graz University of Technology[1], Cloudflare[2]

November 20, 2025

Introduction

Introduction

- Certificates bind a domain name to a public key

Introduction

- Certificates bind a domain name to a public key
- add accountability to issuance process: public logging after Maximum Merge Delay (MMD)

Introduction

- Certificates bind a domain name to a public key
- add accountability to issuance process: public logging after Maximum Merge Delay (MMD)
- currently six logs that are widely trusted by browsers

Introduction

- Certificates bind a domain name to a public key
- add accountability to issuance process: public logging after Maximum Merge Delay (MMD)
- currently six logs that are widely trusted by browsers
- a certificate needs two valid SCTs(Signed Certificate Timestamps) for SCT-enforcing browsers

Introduction

- Certificates bind a domain name to a public key
- add accountability to issuance process: public logging after Maximum Merge Delay (MMD)
- currently six logs that are widely trusted by browsers
- a certificate needs two valid SCTs(Signed Certificate Timestamps) for SCT-enforcing browsers
- instead of trusting one CA: Trust two logs

Introduction

- Certificates bind a domain name to a public key
- add accountability to issuance process: public logging after Maximum Merge Delay (MMD)
- currently six logs that are widely trusted by browsers
- a certificate needs two valid SCTs(Signed Certificate Timestamps) for SCT-enforcing browsers
- instead of trusting one CA: Trust two logs
- more logs in the future: back to the start?

SCT Auditing

- SCTs need to be checked for inclusion in the log after the MMD

SCT Auditing

- SCTs need to be checked for inclusion in the log after the MMD
- checking directly is problematic for user privacy:

SCT Auditing

- SCTs need to be checked for inclusion in the log after the MMD
- checking directly is problematic for user privacy: leaks browser history

SCT Auditing

- SCTs need to be checked for inclusion in the log after the MMD
- checking directly is problematic for user privacy: leaks browser history
- very limited auditing by differentially private lookups

SCT Auditing

- SCTs need to be checked for inclusion in the log after the MMD
- checking directly is problematic for user privacy: leaks browser history
- very limited auditing by differentially private lookups
- Can we do better (e.g., PIR?)

SCT Auditing

- SCTs need to be checked for inclusion in the log after the MMD
- checking directly is problematic for user privacy: leaks browser history
- very limited auditing by differentially private lookups
- Can we do better (e.g., PIR?)
  Earlier analysis suggests this is impractical

This ~~Paper~~ Presentation

1. SCT auditing and slow issuance certificates: post-quantum certificates, alternatives to SCTs, and rough estimates

This ~~Paper~~ Presentation

1. SCT auditing and slow issuance certificates: post-quantum certificates, alternatives to SCTs, and rough estimates
2. revisiting SCT auditing with concrete requirements for PIR, and how the state of the art fulfills our requirements

This ~~Paper~~ Presentation

1. SCT auditing and slow issuance certificates: post-quantum certificates, alternatives to SCTs, and rough estimates
2. revisiting SCT auditing with concrete requirements for PIR, and how the state of the art fulfills our requirements
3. Benchmarks with ideas on database management to get an efficient PIR lookup

This ~~Paper~~ Presentation

1. SCT auditing and slow issuance certificates: post-quantum certificates, alternatives to SCTs, and rough estimates
2. revisiting SCT auditing with concrete requirements for PIR, and how the state of the art fulfills our requirements
3. Benchmarks with ideas on database management to get an efficient PIR lookup
4. Batching audits at little to no cost to scale to audits under 100 kB per day

Preliminaries

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates led to CA being distrusted!

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates led to CA being distrusted!
- CT requires all certificates to be publicly logged

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates led to CA being distrusted!
- CT requires all certificates to be publicly logged
- after logging, a log returns a Signed Certificate Timestamp (SCT)

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates led to CA being distrusted!
- CT requires all certificates to be publicly logged
- after logging, a log returns a Signed Certificate Timestamp (SCT)
- Firefox, Safari, Chrome, and Brave require at least two SCTs from distinct, trusted logs to validate a certificate.

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates led to CA being distrusted!
- CT requires all certificates to be publicly logged
- after logging, a log returns a Signed Certificate Timestamp (SCT)
- Firefox, Safari, Chrome, and Brave require at least two SCTs from distinct, trusted logs to validate a certificate.
- the log datastructure enables the computation of two proofs:

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates led to CA being distrusted!
- CT requires all certificates to be publicly logged
- after logging, a log returns a Signed Certificate Timestamp (SCT)
- Firefox, Safari, Chrome, and Brave require at least two SCTs from distinct, trusted logs to validate a certificate.
- the log datastructure enables the computation of two proofs: An inclusion proof that proves a specific node is included in the log,

Certificate Transparency

- 2011: DigiNotar wrongfully issued over 500 certificates led to CA being distrusted!
- CT requires all certificates to be publicly logged
- after logging, a log returns a Signed Certificate Timestamp (SCT)
- Firefox, Safari, Chrome, and Brave require at least two SCTs from distinct, trusted logs to validate a certificate.
- the log datastructure enables the computation of two proofs: An inclusion proof that proves a specific node is included in the log, and a consistency proof proving there are no split-view attacks

Private SCT auditing

- Controlling two CT logs is sufficient to serve a malicious certificate without immediate detection by the browser

Private SCT auditing

- Controlling two CT logs is sufficient to serve a malicious certificate without immediate detection by the browser
- Two phases in SCT auditing:

Private SCT auditing

- Controlling two CT logs is sufficient to serve a malicious certificate without immediate detection by the browser
- Two phases in SCT auditing:
  1. reporting phase where an SCT is collected

Private SCT auditing

- Controlling two CT logs is sufficient to serve a malicious certificate without immediate detection by the browser
- Two phases in SCT auditing:
  1. reporting phase where an SCT is collected
  2. auditing phase where an SCT is checked for inclusion

Private SCT auditing

- Controlling two CT logs is sufficient to serve a malicious certificate without immediate detection by the browser
- Two phases in SCT auditing:
  1. reporting phase where an SCT is collected
  2. auditing phase where an SCT is checked for inclusion
- Public logging of collected SCTs is sufficient!

Private SCT auditing

- Controlling two CT logs is sufficient to serve a malicious certificate without immediate detection by the browser
- Two phases in SCT auditing:
  1. reporting phase where an SCT is collected
  2. auditing phase where an SCT is checked for inclusion
- Public logging of collected SCTs is sufficient!
- domain owners can monitor logs to check for misbehaviour

SCT Auditing in Practice: Google Chrome

- Chrome audits SCTs

SCT Auditing in Practice: Google Chrome

- Chrome audits SCTs
- SCTs from popular websites are stored in the browser

SCT Auditing in Practice: Google Chrome

- Chrome audits SCTs
- SCTs from popular websites are stored in the browser
- PwnedPasswords-like approach: query hash prefix, get hash postfixes.

SCT Auditing in Practice: Google Chrome

- Chrome audits SCTs
- SCTs from popular websites are stored in the browser
- PwnedPasswords-like approach: query hash prefix, get hash postfixes.
- possible network level attack-selectively target communication and drop it

SCT Auditing in Practice: Google Chrome

- Chrome audits SCTs
- SCTs from popular websites are stored in the browser
- PwnedPasswords-like approach: query hash prefix, get hash postfixes.
- possible network level attack-selectively target communication and drop it
  $\rightarrow$ persist-and-retry mechanism

Private Information Retrieval

- Privately retrieve an item from database

Private Information Retrieval

- Privately retrieve an item from database
- for SCT auditing: Trillian did not distinguish between sequencing and including an entry until last year

Private Information Retrieval

- Privately retrieve an item from database
- for SCT auditing: Trillian did not distinguish between sequencing and including an entry until last year lookup inclusion by hash!

Private Information Retrieval

- Privately retrieve an item from database
- for SCT auditing: Trillian did not distinguish between sequencing and including an entry until last year
  lookup inclusion by hash!
- hash certificates into Bloom filter, get single-bit response indicating if record exists

Private Information Retrieval

- Privately retrieve an item from database
- for SCT auditing: Trillian did not distinguish between sequencing and including an entry until last year
  lookup inclusion by hash!
- hash certificates into Bloom filter, get single-bit response indicating if record exists
  recompute Bloom filter on the server-side when a new certificate is added.

Private Information Retrieval

- Privately retrieve an item from database
- for SCT auditing: Trillian did not distinguish between sequencing and including an entry until last year
  lookup inclusion by hash!
- hash certificates into Bloom filter, get single-bit response indicating if record exists
  recompute Bloom filter on the server-side when a new certificate is added.
- PIR protocols improved: more efficient, single-server protocols

Private Information Retrieval

- Privately retrieve an item from database
- for SCT auditing: Trillian did not distinguish between sequencing and including an entry until last year
  lookup inclusion by hash!
- hash certificates into Bloom filter, get single-bit response indicating if record exists
  recompute Bloom filter on the server-side when a new certificate is added.
- PIR protocols improved: more efficient, single-server protocols
- Can we do a lookup by sequence number and get the hash as a response?

Slow Issuance Certificates

Slow-issuance certificates

- elliptic curve signatures are 65 bytes

Slow-issuance certificates

- elliptic curve signatures are 65 bytes
- ML-DSA-44 signatures are 2.4 kB

Slow-issuance certificates

- elliptic curve signatures are 65 bytes
- ML-DSA-44 signatures are 2.4 kB
- Adding ML-DSA-44 to a TLS handshake will more than double the size of the handshake

Slow-issuance certificates

- elliptic curve signatures are 65 bytes
- ML-DSA-44 signatures are 2.4 kB
- Adding ML-DSA-44 to a TLS handshake will more than double the size of the handshake
  $\rightarrow$ switch to non-X.509 PKI

Slow-issuance certificates

- elliptic curve signatures are 65 bytes
- ML-DSA-44 signatures are 2.4 kB
- Adding ML-DSA-44 to a TLS handshake will more than double the size of the handshake
  $\rightarrow$ switch to non-X.509 PKI
- Embedding a proof-of-inclusion in the certificate (instead of an SCT) would remove the need for auditing

Slow-issuance certificates

- elliptic curve signatures are 65 bytes
- ML-DSA-44 signatures are 2.4 kB
- Adding ML-DSA-44 to a TLS handshake will more than double the size of the handshake
  $\rightarrow$ switch to non-X.509 PKI
- Embedding a proof-of-inclusion in the certificate (instead of an SCT) would remove the need for auditing
- Chrome says ML-DSA-44 is undeployable

The problem of SCTs and post-quantum certificates

- Right now, certificates can be used immediately after they are issued: this is why SCT audits are necessary

The problem of SCTs and post-quantum certificates

- Right now, certificates can be used immediately after they are issued: this is why SCT audits are necessary
- giving up immediate issuance eliminates the need for SCTs and SCT auditing

The problem of SCTs and post-quantum certificates

- Right now, certificates can be used immediately after they are issued: this is why SCT audits are necessary
- giving up immediate issuance eliminates the need for SCTs and SCT auditing
- use proof of inclusion, certificate is usable after new tree heads propagate

The problem of SCTs and post-quantum certificates

- Right now, certificates can be used immediately after they are issued: this is why SCT audits are necessary
- giving up immediate issuance eliminates the need for SCTs and SCT auditing
- use proof of inclusion, certificate is usable after new tree heads propagate
- When is a new certificate needed immediately?

Scenarios for immediate issuance

1. **Brand New Domains** that were just created. Low traffic, but once for each domain.

Scenarios for immediate issuance

1. **Brand New Domains** that were just created. Low traffic, but once for each domain.
2. **Validity Gaps** from configuration errors. Hopefully disappears over time.

Scenarios for immediate issuance

1. **Brand New Domains** that were just created. Low traffic, but once for each domain.
2. **Validity Gaps** from configuration errors. Hopefully disappears over time.
3. **Urgent Re-Issuance** when a domain is moved on short notice, e.g., because of a DDoS attack.

Client Support

1. Supporting with a reliable update mechanism, rarely requires fallbacks

Client Support

1. Supporting with a reliable update mechanism, rarely requires fallbacks
2. Supporting without getting reliable updates, falls back more often when becoming stale

Client Support

1. Supporting with a reliable update mechanism, rarely requires fallbacks
2. Supporting without getting reliable updates, falls back more often when becoming stale
3. No support, requires the fallback mechanism every time

Estimating a Slow-Issuance Ecosystem

- ecosystem does not exist, but we can try to estimate server behaviour

Estimating a Slow-Issuance Ecosystem

- ecosystem does not exist, but we can try to estimate server behaviour
- assuming X.509 certificates are replaced with slow-issuance certificates, we have two datasets:

Estimating a Slow-Issuance Ecosystem

- ecosystem does not exist, but we can try to estimate server behaviour
- assuming X.509 certificates are replaced with slow-issuance certificates, we have two datasets:
  - random domains with 616 142 certificates

Estimating a Slow-Issuance Ecosystem

- ecosystem does not exist, but we can try to estimate server behaviour
- assuming X.509 certificates are replaced with slow-issuance certificates, we have two datasets:
  - random domains with 616 142 certificatescorresponding to 123 733 unique domains

Estimating a Slow-Issuance Ecosystem

- ecosystem does not exist, but we can try to estimate server behaviour
- assuming X.509 certificates are replaced with slow-issuance certificates, we have two datasets:
  - random domains with 616 142 certificatescorresponding to 123 733 unique domains
  - top domains with 36 375 certificates from 75 out of the top 100 domains

Estimating a Slow-Issuance Ecosystem

- ecosystem does not exist, but we can try to estimate server behaviour
- assuming X.509 certificates are replaced with slow-issuance certificates, we have two datasets:
  - random domains with 616 142 certificatescorresponding to 123 733 unique domains
  - top domains with 36 375 certificates from 75 out of the top 100 domains
- Brand new domains:

Estimating a Slow-Issuance Ecosystem

- ecosystem does not exist, but we can try to estimate server behaviour
- assuming X.509 certificates are replaced with slow-issuance certificates, we have two datasets:
  - random domains with 616 142 certificatescorresponding to 123 733 unique domains
  - top domains with 36 375 certificates from 75 out of the top 100 domains
- Brand new domains: $\approx$ 95 000 new domains

Estimating Validity Gaps

- slow embedding crucially relies on correct updating mechanisms

Estimating Validity Gaps

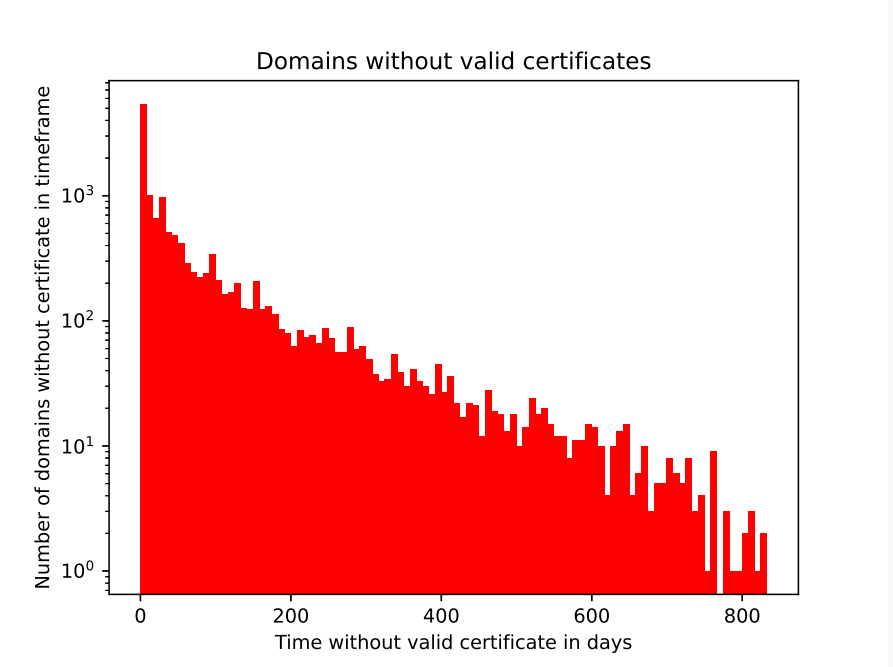
- slow embedding crucially relies on correct updating mechanisms
- Our statistics indicate that most domains without a valid certificate are renewed within a day

Estimating Validity Gaps

- slow embedding crucially relies on correct updating mechanisms
- Our statistics indicate that most domains without a valid certificate are renewed within a day
- we hope wrong configurations are eliminated over time

Estimating Validity Gaps

- slow embedding crucially relies on correct updating mechanisms
- Our statistics indicate that most domains without a valid certificate are renewed within a day
- we hope wrong configurations are eliminated over time
- 100 000 domains per day



Estimating Domain Moves

- Certificates have clear renewal patterns

Estimating Domain Moves

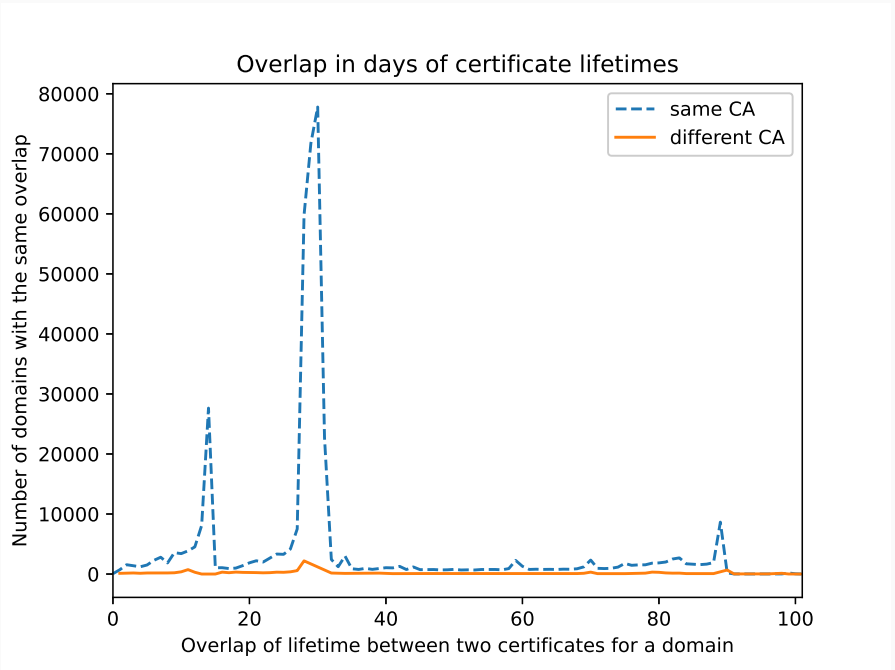
- Certificates have clear renewal patterns
- new CA + renewal outside of pattern + no other valid certificate present: counted as triggering a fallback

Estimating Domain Moves

- Certificates have clear renewal patterns
- new CA + renewal outside of pattern + no other valid certificate present: counted as triggering a fallback
- 100 000 certificates per day

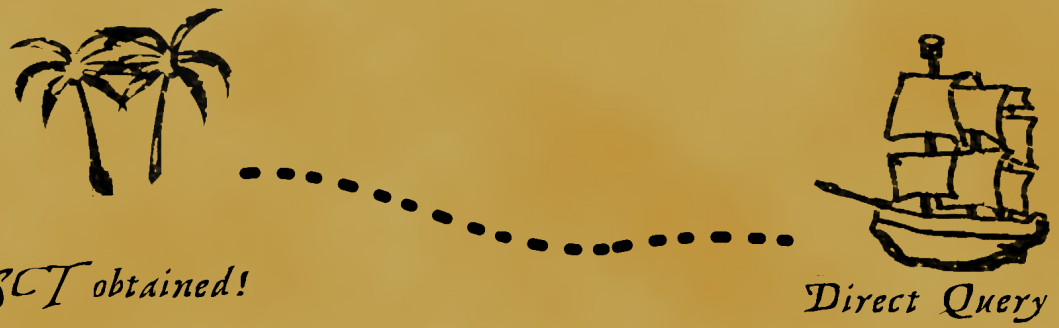
Estimating Domain Moves

- Certificates have clear renewal patterns
- new CA + renewal outside of pattern + no other valid certificate present: counted as triggering a fallback
- 100 000 certificates per day

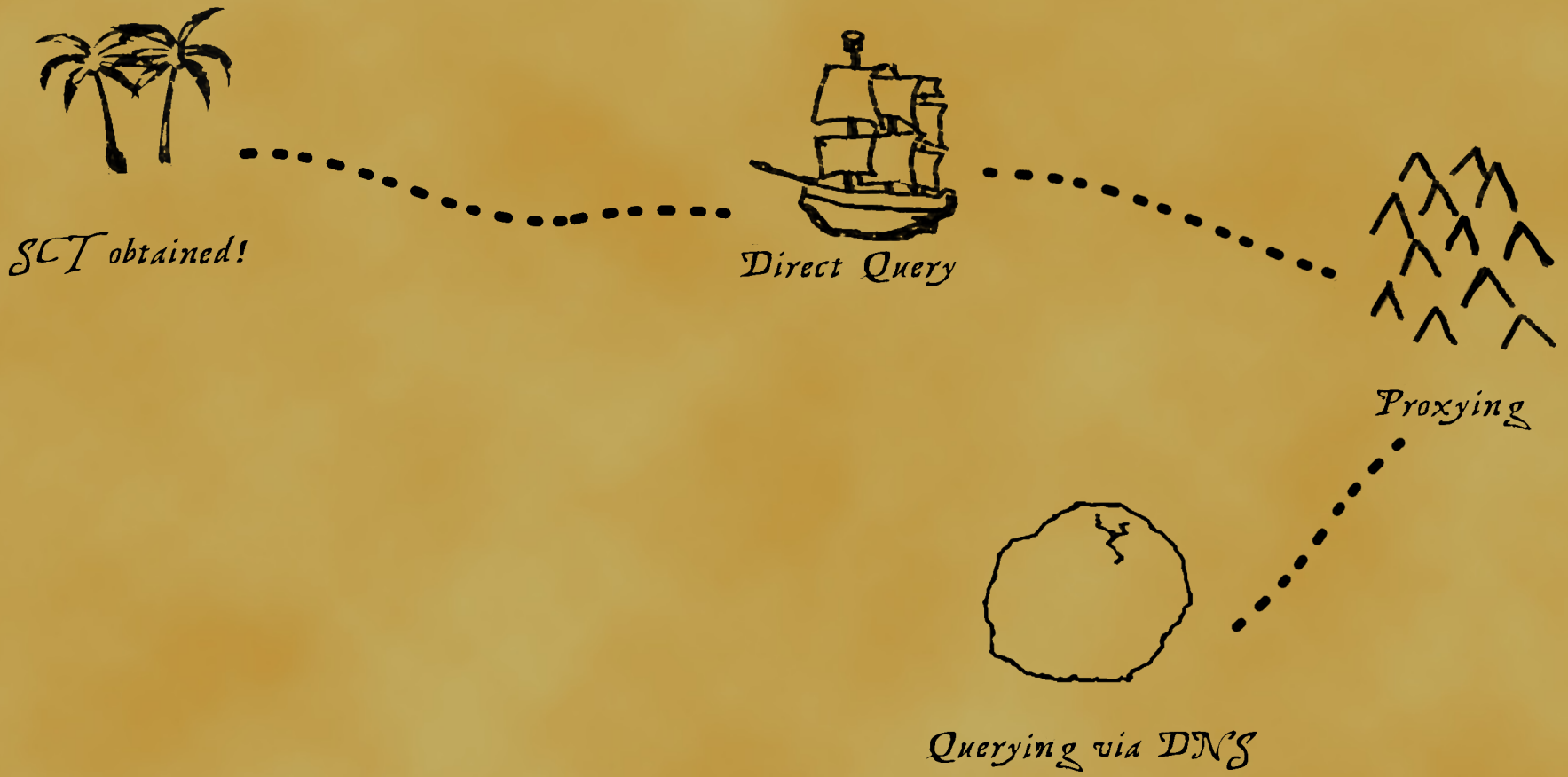


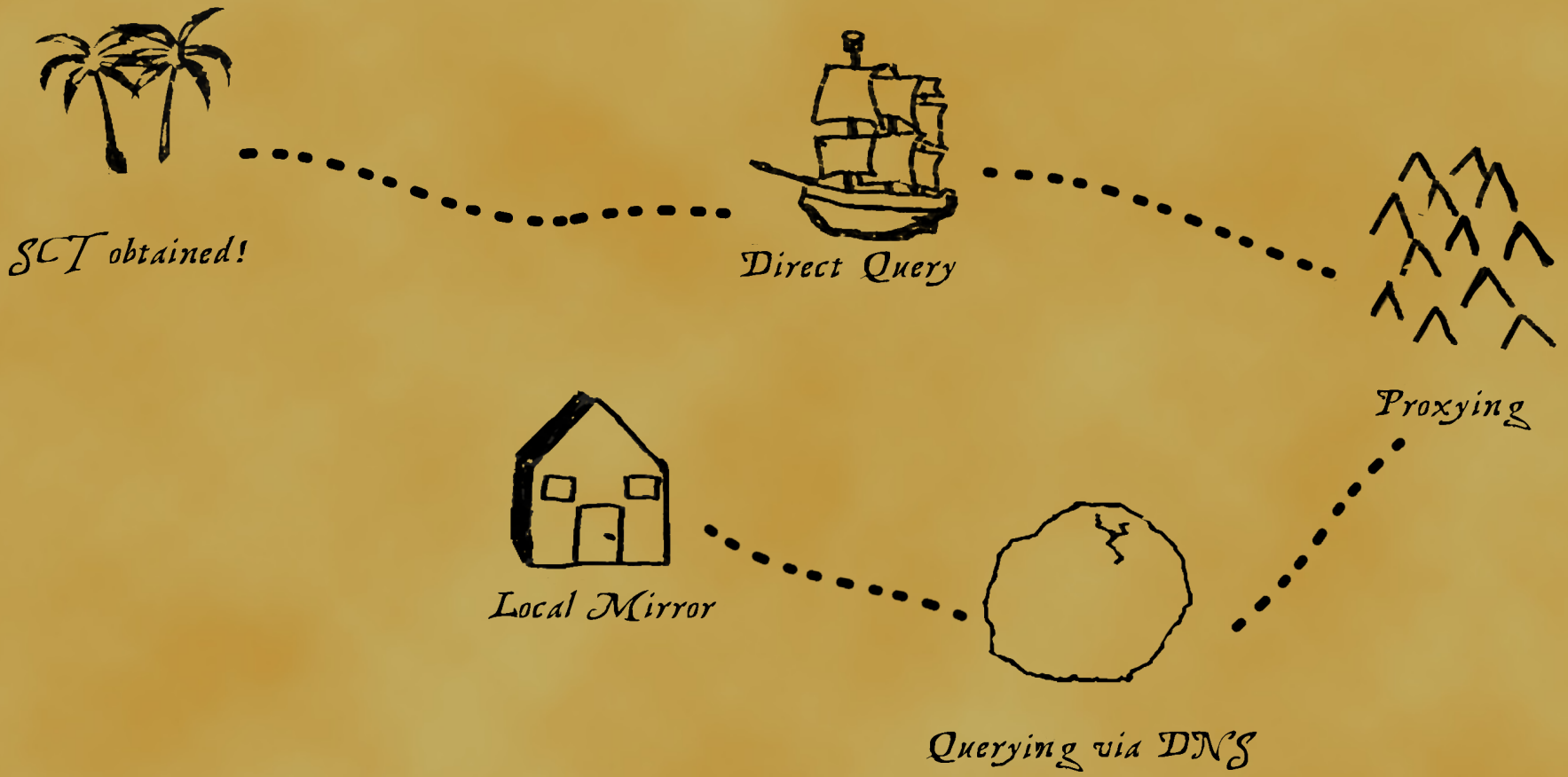
Towards Fully Private Auditing

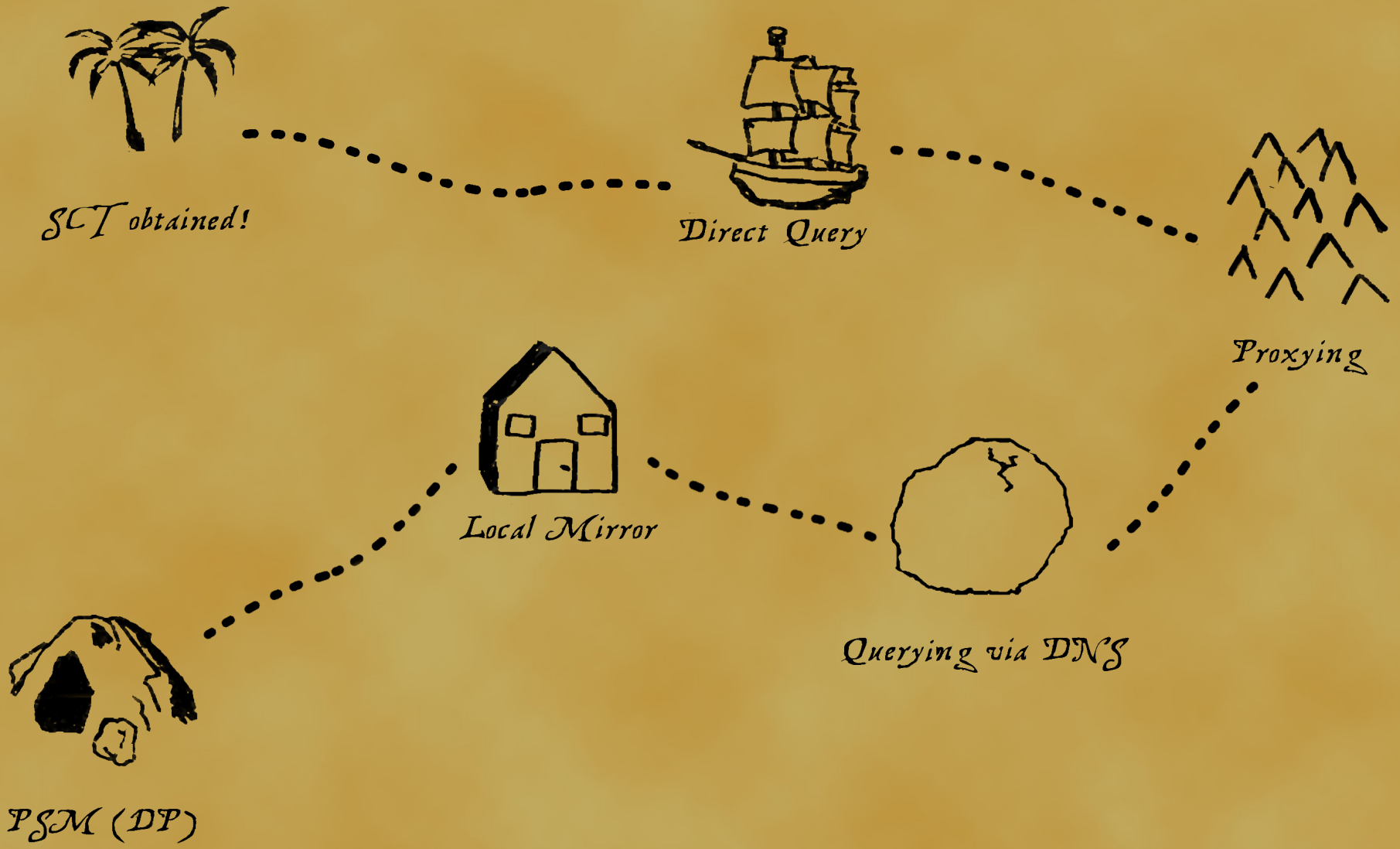
SCT obtained!


SCT obtained!
Direct Query


SCT obtained!
Direct Query
Proxying


SCT obtained!
Direct Query
Proxying
Querying via DNS


SCT obtained!
Direct Query
Proxying
Local Mirror
Querying via DNS


SCT obtained!
Direct Query
Proxying
Local Mirror
Querying via DNS
PSM (DP)


SCT obtained!
Direct Query
Proxying
Local Mirror
Querying via DNS
PSM (DP)
PIR

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake.

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake. A fractional daily increase of 100 kB is also tolerable

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake. A fractional daily increase of 100 kB is also tolerable
- **Compute** One core-second per user per day

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake. A fractional daily increase of 100 kB is also tolerable
- **Compute** One core-second per user per day $\rightarrow$ a thousand 64-core servers

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake. A fractional daily increase of 100 kB is also tolerable
- **Compute** One core-second per user per day $\rightarrow$ a thousand 64-core servers
  minimize server compute overhead (e.g. updating Bloom filters)

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake. A fractional daily increase of 100 kB is also tolerable
- **Compute** One core-second per user per day $\rightarrow$ a thousand 64-core servers
  minimize server compute overhead (e.g. updating Bloom filters)
- **Reusing key material** Reusing public keys is acceptable for private audits

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake. A fractional daily increase of 100 kB is also tolerable
- **Compute** One core-second per user per day $\rightarrow$ a thousand 64-core servers
  minimize server compute overhead (e.g. updating Bloom filters)
- **Reusing key material** Reusing public keys is acceptable for private audits - allows linking queries -

Constraints for SCT auditing with PIR

- **Lookup Requirements** lookup certificate hash by sequence number with trusted third party, e.g. browser auditing infrastructure.
- **Database Preprocessing** minimize per-user, user-independent is feasible
- **Batching** Audits are ideally performed immediately, but batching or simplified batching are within scope
- **Communication** $\approx 1\,000$ TLS connections/day, each 2.5-5 kB handshake. A fractional daily increase of 100 kB is also tolerable
- **Compute** One core-second per user per day $\rightarrow$ a thousand 64-core servers
  minimize server compute overhead (e.g. updating Bloom filters)
- **Reusing key material** Reusing public keys is acceptable for private audits - allows linking queries - add randomized delays

PIR: State of the Art

PIR from Fully Homomorphic Encryption

- SealPIR encrypts the record index as a polynomial x^i, privately expands it into a base vector and performs oblivious multiplication with the database

PIR: State of the Art

PIR from Fully Homomorphic Encryption

- SealPIR encrypts the record index as a polynomial x^i, privately expands it into a base vector and performs oblivious multiplication with the database
- Queries can get smaller with recursion at the cost of larger public parameters and per-client storage

PIR: State of the Art

PIR from Fully Homomorphic Encryption

- SealPIR encrypts the record index as a polynomial x^i, privately expands it into a base vector and performs oblivious multiplication with the database
- Queries can get smaller with recursion at the cost of larger public parameters and per-client storage

PIR from Learning-With-Errors

- LWE-based schemes send a base vector indicating the index

PIR: State of the Art

PIR from Fully Homomorphic Encryption

- SealPIR encrypts the record index as a polynomial x^i, privately expands it into a base vector and performs oblivious multiplication with the database
- Queries can get smaller with recursion at the cost of larger public parameters and per-client storage

PIR from Learning-With-Errors

- LWE-based schemes send a base vector indicating the index
- LWE is faster for computation, RLWE has smaller elements

PIR: State of the Art

PIR from Fully Homomorphic Encryption

- SealPIR encrypts the record index as a polynomial x^i, privately expands it into a base vector and performs oblivious multiplication with the database
- Queries can get smaller with recursion at the cost of larger public parameters and per-client storage

PIR from Learning-With-Errors

- LWE-based schemes send a base vector indicating the index
- LWE is faster for computation, RLWE has smaller elements
- HintlessPIR and TiptoePIR eliminate the hint by bootstrapping: outer RLWE element with inner LWE element

PIR: State of the Art

PIR from Fully Homomorphic Encryption

- SealPIR encrypts the record index as a polynomial x^i, privately expands it into a base vector and performs oblivious multiplication with the database
- Queries can get smaller with recursion at the cost of larger public parameters and per-client storage

PIR from Learning-With-Errors

- LWE-based schemes send a base vector indicating the index
- LWE is faster for computation, RLWE has smaller elements
- HintlessPIR and TiptoePIR eliminate the hint by bootstrapping: outer RLWE element with inner LWE element
- YPIR uses an algebraic transformation between RLWE and LWE, tiny responses

Downsizing the Database

Downsizing the Database

- PIR schemes perform a lot better with smaller databases

Downsizing the Database

- PIR schemes perform a lot better with smaller databases
- SCT Merkle trees grow steadily in size

Downsizing the Database

- PIR schemes perform a lot better with smaller databases
- SCT Merkle trees grow steadily in size
- shard database for better performance

Sharding by Structure: Tiling

- SCT logs group entries into tiles

Sharding by Structure: Tiling

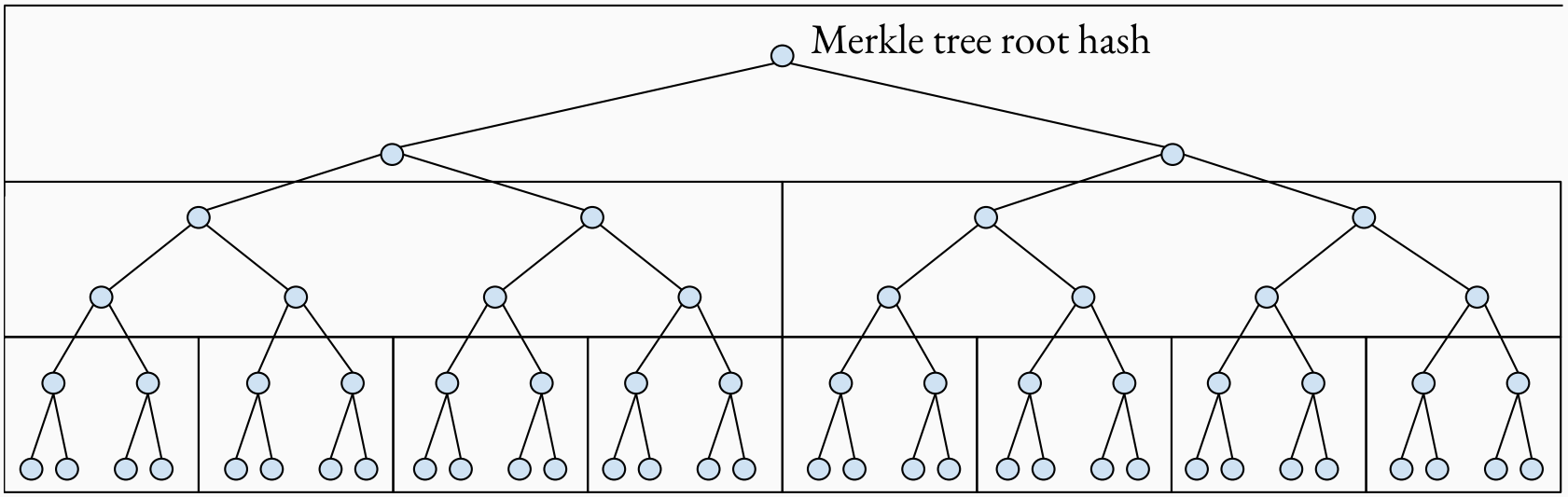
- SCT logs group entries into tiles
- check whether tile is in the tree

Sharding by Structure: Tiling

- SCT logs group entries into tiles
- check whether tile is in the tree
- remaining tile has to be included in the certificate or database record

Sharding by Structure: Tiling

- SCT logs group entries into tiles
- check whether tile is in the tree
- remaining tile has to be included in the certificate or database record
- $2^{15.5}$ to $2^{23.5}$ database records (2 levels up, one level up)



Each rectangle represents a tile with a height of 2, containing up to 6 hashes

Sharding by Epoch:

- CT logs grow constantly, but we only want to audit valid certificates!

Sharding by Epoch:

- CT logs grow constantly, but we only want to audit valid certificates!
- previously discussed as STH discipline

Sharding by Epoch:

- CT logs grow constantly, but we only want to audit valid certificates!
- previously discussed as STH discipline
- group certificates and publish new temporal treeheads periodically, e.g. every second

Sharding by Epoch:

- CT logs grow constantly, but we only want to audit valid certificates!
- previously discussed as STH discipline
- group certificates and publish new temporal treeheads periodically, e.g. every second
- audit temporal tree head

Sharding by Epoch:

- CT logs grow constantly, but we only want to audit valid certificates!
- previously discussed as STH discipline
- group certificates and publish new temporal treeheads periodically, e.g. every second
- audit temporal tree head
- further reduction: group certificates together again after a longer period, e.g., a day.

Sharding by Epoch:

- CT logs grow constantly, but we only want to audit valid certificates!
- previously discussed as STH discipline
- group certificates and publish new temporal treeheads periodically, e.g. every second
- audit temporal tree head
- further reduction: group certificates together again after a longer period, e.g., a day.
- especially interesting for the fast-issuance fallback case

Benchmarks

Database parameters	Scheme	Request Size	Response Size
$\lvert DB\rvert = 2^{31.5}$ $\lvert r\rvert = 1$	plain YPIR	735 kB	12 kB
$\lvert DB\rvert = 2^{15.5}$ $\lvert r\rvert = 256$	SealPIR $d = 1$	**46 kB**	46 kB
	SealPIR $d = 2$	93 kB	185 kB
	Hintless PIR	370 kB	5.9 MB
	YPIR-SimplePIR	932 kB	**12 kB**
$\lvert DB\rvert = 2^{15.5}$ $\lvert r\rvert = 131072$	SealPIR $d = 1$	-	-
	SealPIR $d = 2$	-	-
	Hintless PIR	-	-
	YPIR-SimplePIR	932 kB	61 kB
$\lvert DB\rvert = 2^{17}$ $\lvert r\rvert = 256$	SealPIR $d = 1$	**46 kB**	46 kB
	SealPIR $d = 2$	93 kB	186 kB
	Hintless PIR	371 kB	6.0 MB
	YPIR-SimplePIR	1.4 MB	**12 kB**

Batch Auditing

Batch Auditing

- auditing 1 000 certificates per day at 46 kB is prohibitive

Batch Auditing

- auditing 1 000 certificates per day at 46 kB is prohibitive
- combine STH discipline with slow-issuance certificates: a single audit per day

Batch Auditing

- auditing 1 000 certificates per day at 46 kB is prohibitive
- combine STH discipline with slow-issuance certificates: a single audit per day
- amortize PIR cost by packing multiple queries into one per day

Batch Auditing

- auditing 1 000 certificates per day at 46 kB is prohibitive
- combine STH discipline with slow-issuance certificates: a single audit per day
- amortize PIR cost by packing multiple queries into one per day
- sum of hashes is enough

Private Retrieval of Sums of Hashes

- LinPIR describes querying arbitrary linear combinations $\sum_i a_i \mathrm{DB}_i$

Private Retrieval of Sums of Hashes

- LinPIR describes querying arbitrary linear combinations $\sum_i a_i \mathrm{DB}_i$
- Batching with LWE: Encode multiple indices in the query vector

Private Retrieval of Sums of Hashes

- LinPIR describes querying arbitrary linear combinations $\sum_i a_i \mathrm{DB}_i$
- Batching with LWE: Encode multiple indices in the query vector
- Batching with FHE: Encode multiple indices in the polynomial

Private Retrieval of Sums of Hashes

- LinPIR describes querying arbitrary linear combinations $\sum_i a_i \mathrm{DB}_i$
- Batching with LWE: Encode multiple indices in the query vector
- Batching with FHE: Encode multiple indices in the polynomial $x^i + x^j$

Private Retrieval of Sums of Hashes

- LinPIR describes querying arbitrary linear combinations $\sum_i a_i \mathrm{DB}_i$
- Batching with LWE: Encode multiple indices in the query vector
- Batching with FHE: Encode multiple indices in the polynomial $x^i + x^j$
  expansion is linear

Security of retrieving Sums of Hashes

- Without batching: n-bit hash needs to be collision resistant

Security of retrieving Sums of Hashes

- Without batching: n-bit hash needs to be collision resistant
- retrieving sums: AdHASH map needs to be collision resistant

Security of retrieving Sums of Hashes

- Without batching: n-bit hash needs to be collision resistant
- retrieving sums: AdHASH map needs to be collision resistant(against published hashes)

Security of retrieving Sums of Hashes

- Without batching: n-bit hash needs to be collision resistant
- retrieving sums: AdHASH map needs to be collision resistant(against published hashes)
- Wagner's algorithm: Collisions in $O(k2^{\frac{n}{2+\lg k}+1})$ for k elements

Security of retrieving Sums of Hashes

- Without batching: n-bit hash needs to be collision resistant
- retrieving sums: AdHASH map needs to be collision resistant(against published hashes)
- Wagner's algorithm: Collisions in $O(k2^{\frac{n}{2+\lg k}+1})$ for k elements
- unclear if the attack is optimal: lower bound is $\Omega(2^{\frac{n}{2k}})$

Retrieving signed sums of hashes

- Solution: leverage algebraic structure to introduce noise

Retrieving signed sums of hashes

- Solution: leverage algebraic structure to introduce noise
- query signed sum or multiples of elements

Retrieving signed sums of hashes

- Solution: leverage algebraic structure to introduce noise
- query signed sum or multiples of elements
- e.g. $x^i + x^j + x^k \to -2x^i + 5x^j + x^k$

Conclusion

Conclusion

- SCT auditing with PIR is possible, thanks to sequence numbers and better PIR proposals

Conclusion

- SCT auditing with PIR is possible, thanks to sequence numbers and better PIR proposals
- Batching or slow-issuance certificates keep it under 100 kB/day

Conclusion

- SCT auditing with PIR is possible, thanks to sequence numbers and better PIR proposals
- Batching or slow-issuance certificates keep it under 100 kB/day
- Future Work
  - Estimate the slow-issuance ecosystem by slowly rolling it out

Conclusion

- SCT auditing with PIR is possible, thanks to sequence numbers and better PIR proposals
- Batching or slow-issuance certificates keep it under 100 kB/day
- Future Work
  - Estimate the slow-issuance ecosystem by slowly rolling it out
  - SCT-friendly PIR: probably some design space

Conclusion

- SCT auditing with PIR is possible, thanks to sequence numbers and better PIR proposals
- Batching or slow-issuance certificates keep it under 100 kB/day
- Future Work
  - Estimate the slow-issuance ecosystem by slowly rolling it out
  - SCT-friendly PIR: probably some design space
  - Practical challenges for deployment:

Conclusion

- SCT auditing with PIR is possible, thanks to sequence numbers and better PIR proposals
- Batching or slow-issuance certificates keep it under 100 kB/day
- Future Work
  - Estimate the slow-issuance ecosystem by slowly rolling it out
  - SCT-friendly PIR: probably some design space
  - Practical challenges for deployment:
    - blocked traffic/captive portals

Conclusion

- SCT auditing with PIR is possible, thanks to sequence numbers and better PIR proposals
- Batching or slow-issuance certificates keep it under 100 kB/day
- Future Work
  - Estimate the slow-issuance ecosystem by slowly rolling it out
  - SCT-friendly PIR: probably some design space
  - Practical challenges for deployment:
    - blocked traffic/captive portals
    - reporting open problem: actionable and concrete evidence of log's misbehavior