

SCIENCE PASSION TECHNOLOGY

Enhancing Security Through Transparency SEAD 2025

Edona Fasllija May 8, 2025

edona.fasllija@tugraz.at

How is all of this related to you?

Enhancing Security Through Transparency May 8, 2025

Browsing the Web

ces Network Performance Memory Application Security X Lighthouse Recorder >>	3 : X Elements Console Sources	s Network Performance Memory Application Security × Lighthouse Recorder >>		
Security overview	Overview	Origin		
	Main origin	https://www.iaik.tugraz.at		
This page is secure (valid HTTPS).	Secure origins	View requests in Network Panel		
 https://www.iaik.tugraz.at Certificate - valid and trusted The connection to this site is using a valid, trusted server certificate issued by GEANT OV RSA CA 4. View certificate Connection - secure connection settings The connection to this site is encrypted and authenticated using TLS 1.2, ECDHE_RSA with X25519, and AES_256_GCM. Resources - all served securely 	https://www.iaik.tugraz.at	Connection Protocol TLS 1.2 Key exchange ECDHE_RSA with X25519 Server signature RSA-PSS with SHA-256 Cipher AES_256_GCM Certificate Subject *.iaik.tugraz.at SAN *.iaik.tugraz.at iaik.tugraz.at Valid from Mon. 13 May 2024 00:00:00 GMT		
All resources on this page are served securely.		Valid until Tue, 13 May 2025 23:59:59 GMT Issuer GEANT OV RSA CA 4 Open full certificate details Certificate Transparency SCT Google 'Xenon2025h1' log (Embedded in certificate, Verified) SCT Let's Encrypt 'Oak2025h1' (Embedded in certificate, Verified) SCT Google 'Argon2025h1' log (Embedded in certificate, Verified)		
	Network Performance Memory Application Security × Lighthouse Recorder >> Image: Security overview	ces Network Performance Memory Application Security × Lighthouse Recorder >> (2) : × Security overview 		

Web PKI



cons: Flaticon.com

Malicious Certificates



Browsing the Web

Certificate Transparency (CT)



6

Certificate Transparency (CT)



LOG SERVER

CERTIFICATE AUTHORITY

WEBSITE (EXAMPLE.COM) CLIENT (BROWSER)

"Transparency Technology provides accountability for data."

F. Valsorda. Modern transparency logs (RWC 2024)



Transparency-Enabled (Verifiable) Systems

- *Retain* the *trusted* authority
- Anyone can check that it's not misbehaving

- Add on a transparency layer
- Maintain the seamless user experience

Primitives

The 'root' of truth for your data

Primitives

Transparency Log : just a tamper-evident append-only list

- List [foo, bar, baz]
- Append-Only •
- Tamper-evident \bullet

[foo, bar, baz, qux]

[cat, bar, baz,qux] [bar, baz, qux]

Requirements:

- Efficient *proof* that a record is *in* the log
- Efficient *proof* that an earlier log is a *prefix* of the current log
- Efficient *iteration* over the records in the log

- ✓ Browsers verify that a certificate is recorded in a log
- ✓ A certificate can not disappear from the log undetected
- ✓ Anyone can scan the log to detect *misissued* certificates

Merkle Tree



Inclusion Proof / Proof of Membership



Inclusion Proof / Proof of Membership





Consistency Proof / Proof of Append-Only



Transparency Ecosystems

How to design a Transparent Ecosystem



https://timoelliott.com/blog/2023/05/why-ai-is-like-drawing-an-owl.html



Missing Pieces: End-to-end Transparent Systems

- Auditing
 - Point-in-time and global
 - Can be done by anyone
- Monitoring
 - Synchronous
 - Per-user
- Split-view Protection

Transparency Ecosystems



Other Transparency Systems

Why Making Key Management Transparent is So Challenging

End-to-end Encrypted Messaging



End-to-end Encrypted Messaging



Preliminaries

Undetectable Man-in-the-middle



Contact verification

QR Code Scanning:

- Physical proximity required
- Rerun verification with all contacts when a user's key changes

Your key changes whenever you:

- switch to a new phone
- factory-reset a phone
- uninstall and reinstall the app

▼⊿ 809

<

End-to-End Encrypted Messaging

17:48 0000. ₹⊿ 80% 48 0 0 0 VA 80 7:48 0 0 0 **Contact Verification** Verify encryption Verify encryption To verify end-to-end encryption, scan the QR code on Ayesha Pawar +1 (408) 555-1234 their device or ask them to scan your QR code. ast seen today at 14:07 Verifying ... End-to-end encryption was automatically verified Today at 1:23 PM Hey there! I'm using WhatsApp. Your QR Code 31 May, 2016 Other ways to verify encryption Other ways to verify encryption Mute notifications . . Scan a QR code Scan a QR code Custom notifications Compare a 60-digit number Compare a 60-digit number Media visibility --Off Kept messages Learn how this works Learn how this works 8 Encryption Messages and calls are end-to-end encrypted. Verifying. 0 0 111 < 111 Disappearing messages C 24 hours 0 Block contact Report contact 睅 Scan QR code on their device 111 0 <

WhatsApp Key Transparency Overview. White Paper.

Key Transparency (KT)



- The SP regularly posts commitments to Key Directory.
- Bob queries Alice's PK and gets *proof* that it is correct.
- Alice's device in the background regularly *monitors* her key w.r.t. the commitments.

Key Transparency (KT) - Requirements



- Directory:
 - Map usernames → public keys
- Publish *commitments* to a directory
- Proof that the returned pk is **correct**.

Logs vs Maps

Logs

- History Tree
 - Grows from left to right
 - Efficient append-only proofs
 - Not so efficient to lookup specific entries

Мар

- Prefix Tree
 - (Key, Value) Store
 - Lexicographical order of leaf nodes
 - Efficient to lookup values

Key Transparency

Verifiable Maps – Sparse Merkle Trees



- Dictionary:
 - {(Alice, pk_A), (Bob, pk_B), Charlie, pk_C)}
 - H(Alice) = 0011
 - H(Bob) = 0100
 - H(Charlie)=1100

Sparse Merkle Tree Construction



Approach limitations:

Privacy:

- Brute force search reveals usernames
 - Phone numbers, email addresses
- Reveals when keys are updated
- Even when not directly queried



Key Transparency

Approach limitations:

- Dictionary:
 - {(Alice, pk_A), (Bob, pk_B), Charlie, pk_C)}
 - H(Alice) = 0011
 - H(Bob) = 0100
 - H(Charlie)=1100
 - H(Mallory)=0011
- Querying Alice and Bob → Mallory is not in the database
- Querying Bob reveals Alice's key updates



Key Transparency

Masking usernames with VRFs

- Dictionary:
 - {(Alice, pk_A), (Bob, pk_B), Charlie, pk_C)}
 - VRF_k(Alice) = 0011
 - VRF_k(Bob) = 0100
 - VRF_k(Charlie)=1100
 - VRF_k(Mallory)=?
- SP needs to prove the VRF is computed correctly.



_

www.tugraz.at

Periodic Publish

hroot1 hroot2			hroot3 t 3	
t 1	t2			
PK	Username	РК	Username	PK
<i>pk</i> _{A,1} , <i>t</i> ₁	Alice	$pk_{A,1}, t_1,$	Alice	$pk_{A,1}, t_1$
pk_{B_1}, t_1		pk _{A,2} ', t ₂		$pk_{A,2}', t_{2}$
	Bob	$pk_{B,1}, t_1$	Bob	pk _{B,1} , t ₁ pk _{B,2} ', t
	t 1 PK $pk_{A,1}, t_1$ $pk_{B,1}, t_1$	hrootz t1 t2 PK Username pk _{A,1} , t ₁ Alice pk _{B,1} , t ₁ Bob	hroot: hroot: t1 t2 PK Username PK $pk_{A,1}, t_1$ Alice $pk_{A,1}, t_1, pk_{A,2}, t_2$ $pk_{B,1}, t_1$ Bob $pk_{B,1}, t_1$	hrootihrootzht1t21PKUsernamePKUsername $pk_{A,1}, t_1$ Alice $pk_{A,1}, t_1, t_1, pk_{A,2}, t_2$ Alice $pk_{B,1}, t_1$ Bob $pk_{B,1}, t_1$ Bob

Periodic Publish – Auditing Append-only Consistency



Auditor checks:

Leaves of old tree are subset of new one New leaves have correct epoch t

Large Append-Only Proofs



- Consistency Proofs
 - Contain only leaf values (hashes), not the raw public keys themselves
 - However, proofs are O(MlogN) in size,
 - M = number of updates/epoch
 - N = total number of leaves in the tree

Client Queries



Log Equivocation

Consistency is key

Enhancing Security Through Transparency May 8, 2025

Split View Attack



How do you post commitments?

- Bulletin Board consistently viewed by all users
- Remove the Bulletin Board, and instead
 - Gossiping
 - Witnessing

- Alternatives:
 - 3rd Party
 - Blockchain
 - Trusted Hardware

Gossiping



- Out-of-band
 - Extra-infrastructure
- In-band
 - Piggybacking
 - Modification of protocol

Witnessing

- *Proactive* vs reactive (Gossip)
- Multiple Independent Parties
- Part of a *M-of-N* Trust Policy

- Check consistency proofs
- Co-sign Commitments (checkpoints)

Other Applications

More Transparency Applications

- Binary Transparency
- Signature Transparency
- Firmware Transparency
- Al model Transparency

Open Problems

Open Problems



Real-time Detection of Misbehavior:

How can we develop faster, automated systems that continuously monitor transparency logs and promptly flag suspicious activity?



Privacy-Preserving Logging Schemes:

How can transparency logging schemes be designed to preserve privacy?



Verifiable Data Structures:

What is the optimal verifiable data structure for Transparency Systems?



Reducing costs of monitoring and auditing:

How can the proofs offered by Transparency systems be made more compact, to allow for quick audits and verifications?

Evolution of bar of trust

- {THING}
- {THING} + {THING}.sig
- {THING} is *logged*
- {THING} is *transparent*

Transparency.dev Summit 2024 - Keynote

- Someone is accountable for {THING}
- {THING} is *discoverable*
- Claims about {THING} are falsifiable
- There exist **entities** able to *verify* claims about {THING}.

Enjoy your weekend!