Identifying Harmful Media in End-to-End Encrypted Communication

Efficient Private Membership Computation

Anunay Kulshrestha Jonathan Mayer

USENIX Security Symposium 2021

Center for Information Technology Policy
Princeton University

CENTER FOR INFORMATION TECHNOLOGY POLICY
CITP.PRINCETON.EDU

Motivation

Harmful Media

- Child sexual abuse material (CSAM), terrorist recruiting imagery, disinformation
- Facebook made 16.8 million reports of CSAM in 2018[1]
- In non-E2EE services, perceptual hash matching (PHM) is used
- Datasets of perceptual hashes collated by National Center for Missing and Exploited Children and Global Internet Forum to Counter Terrorism

End-to-end encryption (E2EE) $\Longrightarrow$ service providers cannot access user media

- Law enforcement agencies argue against E2EE deployment until providers can detect harmful media[2]
- Civil society and academics are skeptical about privacy-preserving detection[3]

[1] Patel et al. (2019)
[2] Patel et al. (2019, 2020)
[3] Portnoy (2019); Green (2019)

Perceptual Hash Matching

Perceptual Hash Function (PHF)

- Reduce media to hashes with Hamming distance locality for perceptual similarity
- However, this is not always true (detailed analysis in the paper)

0x044414050505454

0x044414050505054

0x8eeebe878fa5054

Perceptual Hash Matching (PHM)

- Client holds media I such that $x = \mathrm{PHF}_k(I)$
- Server holds set $\mathcal{B}$ of harmful perceptual hashes (hidden from public)
- $d_H(x, y)$ is the Hamming distance between $x, y \in \{0, 1\}^k$ and $\delta_H < k$ is a similarity threshold

Is $x \in \mathcal{B}$? (exact) or is $y \in \mathcal{B}$ s.t. $d_H(x, y) \leq \delta_H$? (approximate)

Problem Formulation

Private Exact Membership Computation (PEMC)

Delegated party learns whether $x \in B$

Private Approximate Membership Computation (PAMC)

Delegated party learns whether $y \stackrel{?}{\in} \mathcal{B}$ such that $d_H(x, y) \leq \delta_H$

Delegation	Either party can be delegated via Server- or Client-revealing variants
Server Privacy	Client learns no information about $\mathcal{B}$
Client Privacy	Server learns no information about x (in a Client-revealing protocol)
Security Model	Semi-honest (same as PHM systems) but one-sided security against a malicious Client who cheats in the protocol

Limitations

Potential for Abuse	Censorship or illegal surveillance (due to lack of trust in $\mathcal{B}$)
False Positives	Inherent in perceptual hash matching that break E2EE privacy guarantee for honest users
Attack Surface	Increase in attack surface for E2EE deployments
Adverse Externalities	International relations and market competition

We do not take a position on deployment. Our goal is to spark discussion and future work by formalizing the problem area and demonstrating technically feasible protocols.

Related Work

Content Moderation (E2EE)	Message franking and message traceback[4]
Private Membership Test	Client holding x wants to know $x \stackrel{?}{\in} B$ where B is held by a Server, without revealing anything about x[5]
Biometric Authentication	Client holding x wants to prove to a Server holding $\mathcal{B}$ that x is similar enough to $y \in \mathcal{B}$, without revealing anything else about x[6]

[4] Tyagi et al. (2019b,a); Grubbs et al. (2017)
[5] Ramezanian et al. (2019); Ali et al. (2019); Tamrakar et al. (2017)
[6] Yasuda (2017); Yasuda et al. (2015); Osadchy et al. (2010)

Protocol Overview

	Locality-Sensitive Hash Bucketization	**Private Information Retrieval**	**Private Equality Test**	**Private Threshold Comparison**
Private Exact Membership Computation	Bit Sampling →	Computationally Private Information Retrieval →	Private Exact Equality Test (ElGamal PHE)	N/A
Private Approximate Membership Computation	Miniature Perceptual Hashes →	Computationally Private Information Retrieval →	Private Approximate Equality Test (BFV FHE) →	Privacy-Preserving Comparison

Locality Sensitive Hash Bucketization

Goal

Reduce hash space without revealing any information about x or $\mathcal{B}$

Locality Sensitive Hashing $Pr[\mathcal{L}(x) = \mathcal{L}(y)] \propto \text{similarity}(x, y)$ for all $x, y \in \{0,1\}^k$

Bit Sampling Sample indices $i_1, \ldots, i_l$ from $[0, k-1]$, let $\mathcal{L}_b : \{0,1\}^k \to \{0,1\}^l$ s.t. $\mathcal{L}_b(x) = x_{i_1} \mid\mid \cdots \mid\mid x_{i_l}$. We use $\mathcal{L}_b(x) = x_0 \mid\mid \cdots \mid\mid x_{l-1}$. Works for PEMC, not for PAMC.

Miniature PHFs If u is the least hash size $\geq l$ supported by PHF and M is arbitrary media, let $\mathcal{L}_p : \{0,1\}^k \to \{0,1\}^l$ s.t. $\mathcal{L}_p(x) = y_0 \mid\mid \cdots \mid\mid y_{l-1}$ where $x = \text{PHF}_k(M)$ and $y = \text{PHF}_u(M)$.

Server builds LSH index **Ind** using an l-bit LSH family $\mathcal{L}(\cdot)$. Bucket $i \in [0, 2^l - 1]$ is mapped to a set of ciphertexts C_i

$$Ind[i] = C_i = \{\text{Enc}(\cdot, y) : \mathcal{L}(y) = i\}$$

Private Information Retrieval

Goal

Retrieve homomorphically encrypted hashes from reduced search space without revealing any information about x or $\mathcal{B}$

Private Information Retrieval Client can retrieve e_j from a Server holding n elements $e_1, \ldots, e_n$ without revealing j

Recall that Server builds LSH index **Ind**

Client with input x can compute $j = \mathcal{L}(x)$ and retrieve $\mathbf{Ind}[j] = C_j$ via PIR

Private Exact Equality Test

Goal

Check if two ciphertexts decrypt to the same value, without revealing anything else

Using partially homomorphic ElGamal Cryptosystem

Public: hash size k, security level λ

Client		Server
$x \in \{0,1\}^k$		$y \in \{0,1\}^k$
		$(\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^\lambda)$
	$\xleftarrow{\mathsf{pk},\, c_y}$	$c_y = \mathsf{Enc}(\mathsf{pk}, y)$
Choose randomizer $r \in \{0,1\}^k$		
$c = r \times_E (c_y +_E \mathsf{Enc}(\mathsf{pk}, -x))$	$\xrightarrow{c}$	
		$\mathsf{DecCheck}(\mathsf{sk}, c, 0) \stackrel{?}{=} \top$

$$c = \mathsf{Enc}(\mathsf{pk}, r \cdot (y - x)) \text{ and } \mathsf{DecCheck}(\mathsf{sk}, c, 0) \stackrel{?}{=} \top \iff x = y$$

Private Approximate Equality Test

Goal

Compute an encryption of the Hamming distance between two encrypted strings

Using fully homomorphic BFV Cryptosystem

BFV ciphertexts are polynomials, so define packings $\mathrm{Pack}_1, \mathrm{Pack}_2 : \{0,1\}^k \to \mathbb{Z}[X]$

J_x, J_y are constant polynomials (given bit size k and BFV parameter n)[7]

$$\mathrm{Pack}_1(m) = \sum_{i=0}^{k-1} m_i X^i \quad \mathrm{Pack}_2(m) = m_0 - \sum_{i=1}^{k-1} m_i X^{n-i} \qquad J_x = \sum_{i=0}^{k-1} X^i \quad J_y = -\sum_{i=0}^{k-1} X^{n-i}$$

$$\zeta(c_x, c_y) = -2^{-1}(2c_x - J_x)(2c_y - J_y) + 2^{-1} J_x J_y$$

$$c_x = \mathrm{Enc}(\cdot, \mathrm{Pack}_1(x)) \qquad c_y = \mathrm{Enc}(\cdot, \mathrm{Pack}_2(y))$$

[7] Yasuda et al. (2015)

Private Approximate Equality Test

Using fully homomorphic BFV Cryptosystem

Public: hash size k, security level λ, BFV parameter n

Client		Server
$x \in \{0,1\}^k$		$y \in \{0,1\}^k$
		$(\mathrm{pk}, \mathrm{sk}) = \mathrm{Gen}(1^\lambda)$
	$\xleftarrow{\mathrm{pk},\, c_y}$	$c_y = \mathrm{Enc}(\mathrm{pk}, \mathrm{Pack}_2(y))$
$c_x = \mathrm{Enc}(\mathrm{pk}, \mathrm{Pack}_1(x))$		
$c_r = \zeta(c_x, c_y)$	$\xrightarrow{c_r}$	
		$p_{x,y} = \mathrm{Dec}(\mathrm{sk}, c_r)$
		$d_H(x,y) = p_{x,y}(0)$

Server learns $d_H(x,y) = p_{x,y}(0)$

Private Threshold Comparison

Goal

Server learns the Hamming distance but want to reveal only whether it is at most δ_H

Idea: Use additive randomization, undo it using Privacy Preserving Comparison (PPC)

Osadchy et al. (2010) use Oblivious Transfer

Client		Server
Choose random poly. r		
$c_r = \zeta(c_x, c_y) + r$	$\xrightarrow{c_r}$	
		$p_{x,y} = \mathrm{Dec}(\mathrm{sk}, c_r)$
$\nu_c = r(0)$	$\xleftrightarrow{\text{PPC}}$	$\nu_s = p_{x,y}(0) - \delta_H = d_H(x,y) + r(0) - \delta_H$

$$\mathrm{PPC}(\cdot, \nu_c, \nu_s) = \perp \iff \nu_s \leq \nu_c \iff d_H(x,y) \leq \delta_H$$

Private Exact Membership Computation (PEMC)

Client		Server
$x, i = \mathcal{L}(x)$		
$C_i = \mathrm{Ind}[i] = \{c_{y,1}, \ldots, c_{y,b}\}$	$\xleftrightarrow{\text{PIR}}$	Ind
	$\xleftarrow{\text{pk}}$	$(\mathrm{pk}, \mathrm{sk})$
$c_{-x} = \mathrm{Enc}(\mathrm{pk}, -x)$		
random $r_1, \ldots, r_b \in \{0,1\}^k$		
random permutation π		
$c_{r,j} = r_j \times_E (c_{y,j} +_E c_{-x})$	$\xrightarrow{\pi\{c_{r,j}\} \text{ for } 1 \leq j \leq b}$	$\bigvee_{1 \leq j \leq b} \mathrm{DecCheck}(\mathrm{sk}, c_{r,j}, 0) \stackrel{?}{=} \top$

■ LSH Bucketization ■ PIR ■ Equality Test

Private Approximate Membership Computation (PAMC)

Client — **Server**

Collective Key Generation (Client ↔ Server)

Client:

$\mathrm{pk}_{\mathrm{agg}}, \mathrm{sk}_c, x, i = \mathcal{L}(x)$

$C_i = \mathrm{Ind}[i] = \{c_{y,1}, \ldots, c_{y,b}\}$

$c_x = \mathrm{Enc}(\mathrm{pk}_{\mathrm{agg}}, \mathrm{Pack}_1(x))$

random poly. $r_1, \ldots, r_b$

$c_{r,j} = \zeta(c_x, c_{y,j}) + r_j$

$\nu_{c,j} = r_j(0)$

Server:

$\mathrm{pk}_{\mathrm{agg}}, \mathrm{sk}_s, \mathrm{Ind}$

Ind

$p_j = \mathrm{Dec}(\mathrm{sk}_s, c_{r,j})$

$\nu_{s,j} = p_j(0) - \delta_H$

$o_j = \mathrm{PPC}(\cdot, \nu_{c,j}, \nu_{s,j})$

$\vee_{1 \le j \le b}(\neg o_j) \stackrel{?}{=} \top$

Messages:

PIR (Client ↔ Server)

$\{c_{r,j}\}$ for $1 \le j \le b$ (Client → Server)

Collective Key Switching (Client ↔ Server)

b rounds of PPC (Client ↔ Server)

■ LSH Bucketization ■ PIR ■ Equality Test ■ Threshold Comparison

Implementation and Benchmarks

- Implementation: C++ using SEAL, SealPIR, NTL, and Botan[8]
- Source: https://github.com/citp/pmc
- Benchmarks: 6-core Intel i7-10710U@1.10GHz, 12MB cache, 32GB RAM using 256-bit hashes, a 20-bit LSH, SealPIR parameters $(n, d) = (2048, 2)$ and parties running locally

$\lvert\mathcal{B}\rvert$	PEMC			PAMC		
	Setup (s)	*Query (s)*	*Comm. (KB)*	*Setup (s)*	*Query (s)*	*Comm. (KB)*
2^{20}	175.0	0.75	394.32	37.2	27.5	508.07
2^{21}	352.3	1.34	394.45	37.4	27.5	586.06
2^{22}	698.9	2.60	394.71	37.4	27.7	742.03
2^{23}	1421.0	5.37	395.25	37.7	28.3	1053.98
2^{24}	2841.0	13.00	396.30	–	–	—

[8] Chen et al. (2017); Angel et al. (2018); Shoup (2020); Lloyd (2020)

Thank you!

Please send questions to anunay@cs.princeton.edu

References

A. Ali, T. Lepoint, S. Patel, M. Raykova, P. Schoppmann, K. Seth, and K. Yeo. Communication–computation trade-offs in pir. Cryptology ePrint Archive, Report 2019/1483, 2019.

S. Angel, H. Chen, K. Laine, and S. Setty. Pir with compressed queries and amortized query processing. In *IEEE Symposium on Security and Privacy*, 2018.

H. Chen, K. Laine, and R. Player. Simple encrypted arithmetic library - seal v2.1. In *International Conference on Financial Cryptography and Data Security*, 2017.

Global Internet Forum to Counter Terrorism. Joint tech innovation. URL `https://www.gifct.org/joint-tech-innovation/`.

References

M. Green. Can end-to-end encrypted systems detect child sexual abuse imagery?, 12 2019. URL https://blog.cryptographyengineering.com/2019/12/08/on-client-side-media-scanning/.

P. Grubbs, J. Lu, and T. Ristenpart. Message franking via committing authenticated encryption. In J. Katz and H. Shacham, editors, *International Cryptology Conference*, pages 66–97, 2017. ISBN 978-3-319-63697-9.

J. Lloyd. *Botan: Crypto and TLS for Modern C++*, 2020. URL https://botan.randombit.net/.

National Center for Missing and Exploited Children. NCMEC's Statement Regarding End-to-End Encryption, 10 2019. URL https://missingkids.org/blog/2019/post-update/end-to-end-encryption.

References

M. Osadchy, B. Pinkas, A. Jarrous, and B. Moskovich. Scifi – a system for secure face identification. In *IEEE Symposium on Security and Privacy*, 2010.

P. Patel, W. Barr, K. McAleenan, and P. Dutton. Open Letter: Facebook's "Privacy First" Proposals, 10 2019. URL `https://www.justice.gov/opa/press-release/file/1207081/download`.

P. Patel, W. Barr, P. Dutton, A. Little, and B. Blair. International Statement: End-to-End Encryption and Public Safety, 10 2020. URL `https://www.justice.gov/opa/pr/international-statement-end-end-encryption-and-public-safety`.

References

E. Portnoy. Why adding client-side scanning breaks end-to-end encryption, 11 2019. URL `https://www.eff.org/deeplinks/2019/11/why-adding-client-side-scanning-breaks-end-end-encryption`.

S. Ramezanian, T. Meskanen, M. Naderpour, V. Junnila, and V. Niemi. Private membership test protocol with low communication complexity. *Digital Communications and Networks*, 2019.

V. Shoup. *NTL: A Library for Doing Number Theory*, 2020. URL `https://libntl.org/`.

S. Tamrakar, J. Liu, A. Paverd, J.-E. Ekberg, B. Pinkas, and N. Asokan. The circle game: Scalable private membership test using trusted hardware. In *ACM Asia Conference on Computer and Communications Security*, 2017.

References

N. Tyagi, P. Grubbs, J. Len, I. Miers, and T. Ristenpart. Asymmetric message franking: Content moderation for metadata-private end-to-end encryption. In *International Cryptology Conference*, 2019a.

N. Tyagi, I. Miers, and T. Ristenpart. Traceback for end-to-end encrypted messaging. In *ACM Conference on Computer and Communications Security*, 2019b.

M. Yasuda. Secure hamming distance computation for biometrics using ideal-lattice and ring-lwe homomorphic encryption. *Information Security Journal: A Global Perspective*, 26(2):85–103, 2017.

M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba. New packing method in somewhat homomorphic encryption and its applications. *Security and Communication Networks*, 8(13):2194–2213, 2015.