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
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) \(\Rightarrow\) 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)

Perceptual Hash Matching (PHM)

- Client holds media $I$ such that $x = \text{PHF}_k(I)$
- Server holds set $B$ 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

\[\text{Is } x \in B? \text{ (exact) or is } y \in B \text{ s.t. } d_H(x, y) \leq \delta_H? \text{ (approximate)}\]
### Problem Formulation

<table>
<thead>
<tr>
<th><strong>Private Exact Membership Computation (PEMC)</strong></th>
<th>Delegated party learns whether $x \in B$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Private Approximate Membership Computation (PAMC)</strong></td>
<td>Delegated party learns whether $y \in B$ such that $d_H(x, y) \leq \delta_H$</td>
</tr>
</tbody>
</table>

**Delegation**
- Either party can be delegated via Server- or Client-revealing variants

**Server Privacy**
- Client learns no information about $B$

**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 $B$)

**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 \(x \in B\) 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 \(B\) that \(x\) is similar enough to \(y \in B\), 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

<table>
<thead>
<tr>
<th>Private Exact Membership Computation</th>
<th>Locality-Sensitive Hash Bucketization</th>
<th>Private Information Retrieval</th>
<th>Private Equality Test</th>
<th>Private Threshold Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit Sampling</td>
<td>Computationally Private Information Retrieval</td>
<td>Private Exact Equality Test (ElGamal PHE)</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Private Approximate Membership Computation</th>
<th>Locality-Sensitive Hash Bucketization</th>
<th>Private Information Retrieval</th>
<th>Private Equality Test</th>
<th>Private Threshold Comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miniature Perceptual Hashes</td>
<td>Computationally Private Information Retrieval</td>
<td>Private Approximate Equality Test (BFV FHE)</td>
<td>Privacy-Preserving Comparison</td>
<td></td>
</tr>
</tbody>
</table>
Locality Sensitive Hash Bucketization

**Goal**
Reduce hash space without revealing any information about $x$ or $B$

**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. $\mathcal{L}_b(x) = x_{i_1} || \cdots || x_{i_l}$. We use $\mathcal{L}_b(x) = x_0 || \cdots || x_{l-1}$. 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. $\mathcal{L}_p(x) = y_0 || \cdots || y_{l-1}$ where $x = \text{PHF}_k(M)$ and $y = \text{PHF}_u(M)$.

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

$\text{Ind}[i] = C_i = \{\text{Enc}(\cdot, y) : \mathcal{L}(y) = i\}$
Goal
Retrieve homomorphically encrypted hashes from reduced search space without revealing any information about $x$ or $B$

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 $\text{Ind}$

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

Client

- $x \in \{0, 1\}^k$
- Choose randomizer $r \in \{0, 1\}^k$
- $c = r \times_E (c_y +_E Enc(pk, -x))$

Server

- $y \in \{0, 1\}^k$
- $(pk, sk) = Gen(1^\lambda)$
- $c_y = Enc(pk, y)$
- DecCheck$(sk, c, 0) \overset{?}{=} \top$

$c = Enc(pk, r \cdot (y - x))$ and DecCheck$(sk, c, 0) \overset{?}{=} \top \iff x = y$
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 $\text{Pack}_1, \text{Pack}_2 : \{0, 1\}^k \rightarrow \mathbb{Z}[X]$

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

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

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

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

$^7$Yasuda et al. (2015)
Private Approximate Equality Test

Using fully homomorphic BFV Cryptosystem

Public: hash size $k$, security level $\lambda$, BFV parameter $n$

Client

- $x \in \{0, 1\}^k$
- $c_x = \text{Enc}(pk, \text{Pack}_1(x))$
- $c_r = \zeta(c_x, c_y)$

Server

- $y \in \{0, 1\}^k$
- $(pk, sk) = \text{Gen}(1^\lambda)$
- $c_y = \text{Enc}(pk, \text{Pack}_2(y))$
- $p_{x,y} = \text{Dec}(sk, c_r)$
- $d_H(x, y) = p_{x,y}(0)$

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 $\delta_H$

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

Osadchy et al. (2010) use Oblivious Transfer

Choose random poly. $r$

$$c_r = \zeta(c_x, c_y) + r$$

$$\nu_c = r(0)$$

$$\nu_s = p_{x,y}(0) - \delta_H = d_H(x, y) + r(0) - \delta_H$$

$$\text{PPC}(\cdot, \nu_c, \nu_s) = \perp \iff \nu_s \leq \nu_c \iff d_H(x, y) \leq \delta_H$$
Private Exact Membership Computation (PEMC)

Client

\[ x, i = \mathcal{L}(x) \]

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

\[ c_{-x} = \text{Enc}(pk, -x) \]

random \( r_1, \ldots, r_b \in \{0, 1\}^k \)

random permutation \( \pi \)

\[ c_{r,j} = r_j \times E(c_{y,j} + E c_{-x}) \]

Server

\[ \text{Ind} \]

\[ \text{pk} \]

\[ \forall 1 \leq j \leq b \text{DecCheck}(sk, c_{r,j}, 0) = \top \]

\[ \text{LSH Bucketization} \quad \text{PIR} \quad \text{Equality Test} \]
Private Approximate Membership Computation (PAMC)

**Client**

\[ \text{pk}_{\text{agg}}, \text{sk}_c, x, i = \mathcal{L}(x) \]

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

\[ c_x = \text{Enc}(\text{pk}_{\text{agg}}, \text{Pack}_1(x)) \]

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**

\[ \text{pk}_{\text{agg}}, \text{sk}_s, \text{Ind} \]

\[ \text{Ind} \]

\[ p_j = \text{Dec}(\text{sk}_s, c_{r,j}) \]

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

\[ o_j = \text{PPC}(\cdot, \nu_{c,j}, \nu_{s,j}) \]

\[ \vee_{1 \leq j \leq b} (\neg o_j) = \top \]

**Collective Key Generation**

**Collective Key Switching**

**PIR**

**LSH Bucketization**

**Equality Test**

**Threshold Comparison**
Implementation and Benchmarks

- Implementation: C++ using SEAL, SealPIR, NTL, and Botan
- 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

| \(|B|\) | 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 | – | – | – |

---

8Chen et al. (2017); Angel et al. (2018); Shoup (2020); Lloyd (2020)
Thank you!

Please send questions to anunay@cs.princeton.edu


References


References


