SoK: Content Moderation for End-to-End Encryption

Sarah Scheffler
Princeton University
USA

Jonathan Mayer
Princeton University
USA

ABSTRACT

Popular messaging applications now enable end-to-end-encryption (E2EE) by default, and E2EE data storage is becoming common. These important advances for security and privacy create new content moderation challenges for online services, because services can no longer directly access plaintext content.

While ongoing public policy debates about E2EE and content moderation in the United States and European Union emphasize child sexual abuse material and misinformation in messaging and storage, we identify and synthesize a wealth of scholarship that goes far beyond those topics. We bridge literature that is diverse in both content moderation subject matter, such as malware, spam, hate speech, terrorist content, and enterprise policy compliance, as well as intended deployments, including not only privacy-preserving content moderation for messaging, email, and cloud storage, but also private introspection of encrypted web traffic by middleboxes.

In this work, we systematize the study of content moderation in E2EE settings. We set out a process pipeline for content moderation, drawing on a broad interdisciplinary literature that is not specific to E2EE. We examine cryptography and policy design choices at all stages of this pipeline, and we suggest areas of future research to fill gaps in literature and better understand possible paths forward.

KEYWORDS

end-to-end encryption, content moderation, privacy

1 INTRODUCTION

How can an online service implement content moderation when it cannot access content? This challenge, at the intersection of promoting information security and mitigating societal harms, is now a global public policy flashpoint and a focal point for scholarship.

Encryption has seen widespread adoption over the past decade and provides a core component of user security and privacy online. End-to-end encryption (E2EE) is available and often the default in popular messaging applications [14, 142, 195, 230, 253, 289], and it is the norm for web traffic between clients and servers [190, 314]. It is also available in email [53], file storage and sharing [16, 197], and audio/video chat [19, 71, 210]. Encryption of data at rest, including full-disk and file-based encryption, is common for devices [12, 15, 17, 119, 255, 344, 365]. In E2EE, a service platform cannot read or tamper with users' plaintext content. This provides privacy and security not only against external attackers but also against threats that compromise the platform, including data breaches, malicious insiders, and more problematic spying by the platform itself [289, 361]. The security and privacy provided by E2EE advances safeguard users worldwide, protecting journalists, activists, government officials, business leaders, and ordinary users alike.

However, E2EE provides this privacy to both use and abuse. Without access to plaintext or an ability to decrypt, an online service is limited in how it can respond to harmful content and facilitate accountability for criminal acts.

In a 2014 address, then-FBI Director James Comey characterized the challenge for law enforcement as "going dark" [256]. Investigators were struggling to conduct electronic surveillance and obtain electronic evidence, because they were increasingly encountering encryption that online services and device vendors could not bypass. Comey's remarks tapped into an existing debate [2, 30, 217] and set the stage for another decade of encryption policy tussles, initially centered on law enforcement access to data stored on devices and more recently emphasizing child exploitation that uses E2EE messaging and storage [92, 113, 188, 256, 275].

As child exploitation activities moved online, efforts to identify and investigate those activities also moved online. One of the primary techniques for proactive detection of online child abuse was—and remains—hash matching. In these detection systems, an online service hashes user content and compares the value against a database of known hashes of child sexual abuse material (CSAM). These hash databases are often coordinated by national child safety organizations, such as the National Center for Missing and Exploited Children (NCMEC) in the United States. NCMEC began assembling hashes of CSAM in the early 2000s, and it accelerated the practice in 2008 after adopting the PhotoDNA perceptual hash function—which is still in widespread use. Hash matching remains a best practice in CSAM detection [139, 170, 188, 266, 324, 362], and it has also been used or proposed to detect terrorist content [134], misinformation [308], and suspicious web links [141].

Hash-based detection methods, and more generally content-based detection methods, are not directly implementable in an E2EE setting because the online service cannot analyze content. In response, law enforcement agencies and child safety groups called for a halt to E2EE adoption until similar methods of detecting CSAM and other forms of online child abuse were available [92, 113, 275].

The encryption policy debate shifted again in 2021, when two independent groups—one in academia and one at Apple—proposed cryptographic protocols that would selectively report user media that matched a perceptual hash set [33, 212]. If deployed in an E2EE setting, these protocols would essentially create an exception to E2EE for matching content. While some stakeholders applauded these protocols as a breakthrough [69, 146], others (including the academic authors and eventually Apple [49, 240, 268]) were more reluctant. The proposals raised more questions than they answered, posing risks to privacy, security, and free expression [1, 117, 132, 244, 283]. Civil society groups were especially alarmed about the possibility that these systems would quickly expand to categories

Proceedings on Privacy Enhancing Technologies 2023(2), 403–429

https://doi.org/10.56553/popets-2023-0060

of content beyond CSAM, especially under pressure from foreign governments [1, 11, 117, 132].

Against this backdrop, security researchers who both appreciate the benefits of E2EE and are concerned about the societal harms that it could facilitate may be left wondering: now what?

This paper seeks to place debates about content moderation in E2EE settings on much-needed shared scientific footing. In the spirit of earnest intellectual investigation into this divisive topic, we aim to provide an evenhanded systematization of prior work on E2EE content moderation and offer guidance about possible constructive directions for future research. We go beyond the (important) problem of detecting CSAM and unite diverse areas of literature that address content moderation for E2EE systems, broadly conceived. Our synthesis of relevant prior work spans topics such as preventing spam, ensuring compliance for corporate networks, and defeating malware. Each problem area poses a distinct set of technical and policy challenges, and while many of these challenges cannot be addressed by technology alone, research *can* improve the security, privacy, efficacy, efficiency, and especially transparency of content moderation.

Our aim is that this paper will offer helpful guidance to three interrelated audiences: (1) researchers studying content moderation who wish to learn about possible system designs and open challenges under E2EE; (2) cryptographers and other researchers studying privacy-preserving systems who seek to understand content moderation objectives which might be met through novel designs; and (3) a broad range of stakeholders who are invested in encryption policy and wish to understand the capabilities and limitations of proposed systems in the research literature.

While this paper describes a number of content moderation systems for E2EE, we do not endorse the adoption of any particular design. We are especially concerned about systems that would disclose user content under E2EE, which we do not believe presently offer sufficient assurances of trustworthiness for deployment.

The core of our systematization is organized around a four-part model of content moderation, which we derive from prior work outside the E2EE setting. This model begins with a *problem context*: the societal harm to be addressed, the role of E2EE in facilitating the harm, the type of content to be moderated, and the parties of concern in sending or receiving the content. The next step is *detection* of the content to moderate, followed by a *response* which may or may not reveal the detection to a third party, such as the online service. Finally, *transparency* enables users to verify and contest the moderation system. The paper is organized as follows:

- **Section 2** offers background on E2EE, content moderation, and the challenges of content moderation under E2EE.
- **Section 3** describes the methods and results of our literature search, with the exception of work on middleboxes (see Appendix B). The search involved over 5,000 papers, and we ultimately identified 119 for detailed analysis and synthesis.
- **Section 4** discusses *problem context.* We show how content moderation objectives influence detection and response methods, and we propose future interdisciplinary research to better understand how E2EE interacts with societal harms and how possible content moderation systems could help.
- **Section 5** explains *detection* paradigms that appear in the literature. We encourage future work evaluating their comparative efficacy and improving perceptual hash functions.
- **Section 6** describes *responses*, some of which implicate user security and privacy. We identify and explain several response mechanisms that are tailored for E2EE settings.
- **Section 7** discusses proposed methods for *transparency* in E2EE content moderation systems. We suggest that this is a particularly promising direction for future research.

We hope to encourage future research that explores the design space for content moderation under E2EE, while respecting security, privacy, and accountability.

2 BACKGROUND AND DEFINITIONS

The topic of content moderation under E2EE implicates a vast array of technical and policy literature. Before turning to our survey of prior work, we begin with background on E2EE, content moderation, and the challenges of content moderation under E2EE.

2.1 End-to-end encryption

End-to-end encryption (E2EE) refers to any authenticated encryption scheme where the "ends" of the communication (a "sender" and one or more "receivers") can send messages to each other via an abstract central channel and where the channel does not have the cryptographic material necessary to read or invisibly alter the message (see Figure 1(a)). This model captures E2EE in one-to-one communication (one receiver), group communication (many receivers), and online storage (the receiver is also the sender). We formally define an E2EE scheme as follows.[1]

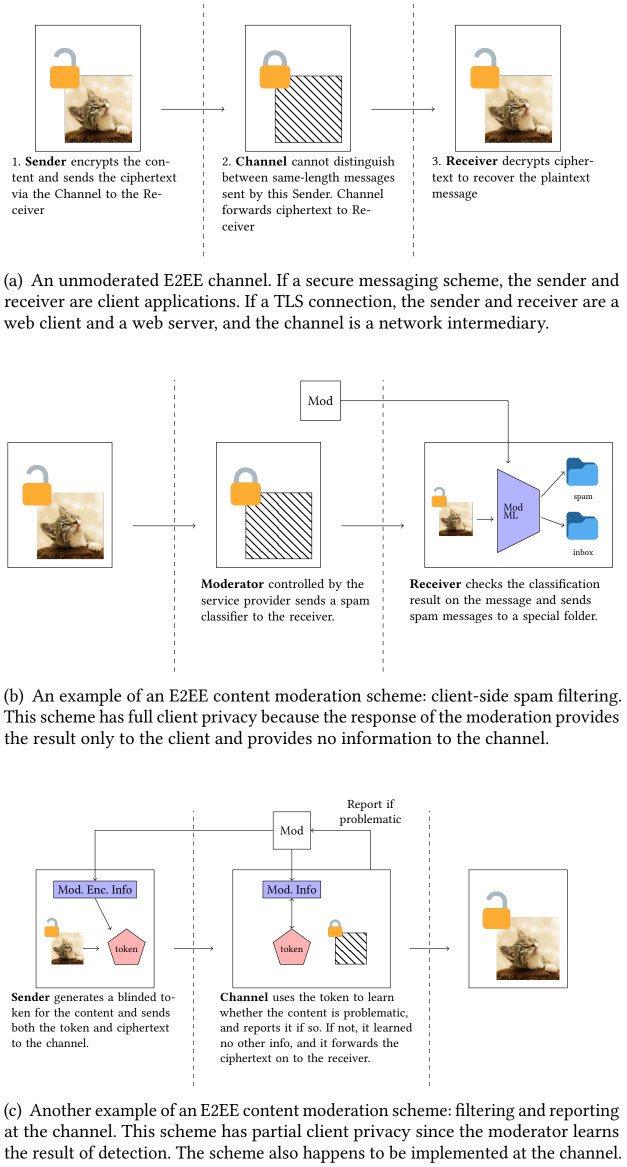
Definition 2.1 (End-to-end encryption)**.** Communication between at least two client "ends" (a sender and one or more receivers) over a channel is *end-to-end encrypted* if it has the following properties:

1. *Confidentiality*: The plaintext content of the message is indistinguishable from random under chosen plaintext attack by both network attackers and the operator of the channel facilitating message transmission.
2. *Integrity*: The receiver of a message can tell if a received message, along with associated header information, was modified from the sender's original message.
3. *Authentication*: The ends of the communication can confirm each other's identities with long-term cryptographic secrets.

These properties protect communication under Authenticated Encryption with Associated Data (AEAD) [319], providing IND$-CPA confidentiality [319] for the message, integrity for the message and a public header, and authentication for each end's identity.

Recent debates about content moderation for E2EE have predominantly focused on secure messaging applications, because of their growing popularity. There has also been a convergence in the technical implementation of E2EE for messaging: the Signal double-ratchet protocol [289], a successor to the Off-the-Record protocol [43], has been implemented with slight variations by messaging and audio/video chat services [51, 71, 142, 195, 253, 289].

[1] For a more detailed description of security properties for E2EE messaging, see Unger et al. [361]. Table II in that work shows near-universal agreement on confidentiality, integrity, and authentication, but some disagreement on other properties.


1. **Sender** encrypts the content and sends the ciphertext via the Channel to the Receiver
2. **Channel** cannot distinguish between same-length messages sent by this Sender. Channel forwards ciphertext to Receiver
3. **Receiver** decrypts ciphertext to recover the plaintext message
Mod
Mod ML
spam
inbox
Moderator controlled by the service provider sends a spam classifier to the receiver.
Receiver checks the classification result on the message and sends spam messages to a special folder.
Report if problematic
Mod
Mod. Enc. Info
token
Mod. Info
token
Sender generates a blinded token for the content and sends both the token and ciphertext to the channel.
Channel uses the token to learn whether the content is problematic, and reports it if so. If not, it learned no other info, and it forwards the ciphertext on to the receiver.

(a) An unmoderated E2EE channel. If a secure messaging scheme, the sender and receiver are client applications. If a TLS connection, the sender and receiver are a web client and a web server, and the channel is a network intermediary.

(b) An example of an E2EE content moderation scheme: client-side spam filtering. This scheme has full client privacy because the response of the moderation provides the result only to the client and provides no information to the channel.

(c) Another example of an E2EE content moderation scheme: filtering and reporting at the channel. This scheme has partial client privacy since the moderator learns the result of detection. The scheme also happens to be implemented at the channel.

Figure 1: The model of our underlying E2EE communication channel, and two examples of moderation schemes.

There are other applications and implementations of E2EE, however, that feature much less prominently in encryption policy tussles and that offer lessons for content moderation. For example, an HTTPS network connection encrypted with Transport Layer Security (TLS [314]) is end-to-end encrypted from the perspective of a network intermediary, such as a network operator or a security service provider monitoring traffic. In this scenario, the web client and web server are the ends of E2EE communication and the network intermediary is part of the channel. While some methods for TLS content moderation sacrifice security and privacy by terminating the connection at a middlebox with plaintext access [89, 179], there has been substantial research on alternative approaches. Some designs use machine learning to analyze encrypted traffic flows [9, 196, 221, 340]), and others use cryptographic methods to perform deep packet inspection (see Appendix B). These methods are applicable to content moderation goals in school or home networks (e.g., [304]) in addition to enterprise policy compliance. We include all of these settings and more in our analysis of content moderation under E2EE.

Signal and the latest version of TLS (1.3) also both provide two additional properties [56, 314, 361]:

(4) *Forward secrecy*: Even if the ends' current keys are compromised, messages sent in the past remain confidential.
(5) *Deniability*: No coalition of parties, including the receiver of a message, can prove a sender sent a specific message.

These properties can both be considered anti-surveillance measures: forward secrecy prevents, for example, routers from archiving encrypted traffic and then decrypting later after a key breach. Deniability ensures that, cryptographically speaking, there is no way to prove that a particular message was sent by a particular sender.[2]

Signal additionally provides the property of *post-compromise security*: after a key compromise, the clients can go through a "refresh" protocol that ensures future messages they send cannot be read using the compromised keys [56].

In this paper we are primarily concerned with non-anonymous E2EE, where the central server knows the identity of the sender and receiver(s), because that is the typical design of widely deployed E2EE services at present. There are "metadata-private" E2EE messaging schemes that provide anonymity for senders [87, 97, 233], receivers [28], or both [73, 74, 306, 316].

2.2 Content moderation

Online services and network operators have devised a vast and diverse toolkit for addressing problematic content. Prior work on content moderation is extensive and includes taxonomies of content moderation remedies and approaches [135, 232], analysis of user behavior [206, 332], examination of human and technical challenges with current content moderation systems [66, 100, 129–131, 143, 156, 181, 191, 238, 318, 331], and analysis of regulation and governance models [50, 147, 203, 347], among other topics. There is also expansive literature on the problem of automatically detecting particular kinds of content that could merit moderation, such as nudity [93, 184, 194, 317] and hate speech [10, 115, 235, 354, 371].

There are many motivations for content moderation, from friendly relocation of off-topic material in a group conversation [312, 353] to removal and reporting of illegal child abuse material [18, 223]. We use the term *problematic content* as an umbrella for user content that an online service may wish to take action on.

The service may seek to detect problematic content as part of a content moderation scheme, such as by placing the content in a matching dataset or training a machine learning classifier. The content moderation scheme could lead to false positive matches and reveal information about non-problematic content. Later on, we will refer to this class of problematic content as class C (e.g., the

[2] Deniability occasionally uses different meanings in different areas of literature; here we refer to it to mean "message repudiation" as defined by Unger et al. [361].

positive results of a spam message classifier) which is attempting to implement class $\tilde{C}$ (e.g., true spam messages).

Several formalized models for content moderation exist in the literature [68, 187, 342]. We adapt the three-part model of Singhal et al. [342] into a four-part model, with the following primary differences: (1) we expand the *terms of use* component into a broader *problem context* component; (2) we expand *enforcement* into *response*, because strict enforcement is just one of many possible responses; and (3) we add a *transparency* component that encompasses how users can verify that content moderation systems are functioning as described. Our content moderation framework is as follows:

1. *Problem context*: What is the goal of the content moderation? In particular, what is the societal harm to address, how does E2EE relate, who are the parties of concern, and what is the definition of relevant content? The problem context scopes the other three phases, placing limits on the detection, response, and transparency methods that would be viable.
2. *Detection*: What method is used to identify the content for moderation, and how accurate is it? What privacy protections are used to process the message content within E2EE? Where will the detection be performed—on the sender device, by the channel, on a receiver device, or some combination?
3. *Response*: When a detection occurs, what happens? Which parties are automatically informed, and what additional information is sent to those parties? What actions are taken automatically? What manual actions are made available?
4. *Transparency*: What information is disclosed to users about the system's purpose, methods, and effectiveness? How can users verify that the system is functioning as described? How can users contest content moderation actions?

These four phases represent four fundamental choices that must be made in the design space of a content moderation system under E2EE. The relationship between these choices can be nuanced. Selecting a machine learning approach to detection instead of matching against a dataset, for example, may reduce confidence in content identification, therefore a weaker response or heightened transparency may be appropriate.

After describing our literature search in Section 3, the paper is organized by these content moderation phases. For each component of a content moderation system, we characterize current research and recommend future directions.

2.3 Content moderation under E2EE

The challenges of content moderation for E2EE predominantly relate to maintaining data confidentiality for servers and clients. We discuss these considerations in turn, followed by particular risks of content moderation under E2EE and limitations of this work.

Server privacy. Content moderation may make use of secret information held by a service provider or a third party. Systems for detecting CSAM, for example, typically rely on matching against a sensitive dataset of CSAM hashes that is kept secret to protect investigative methods and prevent evasion that could reveal investigative methods [33, 139, 212, 324]. When the service provider has secret information that must be kept from the user during content moderation, we refer to this property as *server privacy* [212].

Definition 2.2 (Server privacy)**.** A content moderation scheme has *server privacy* if it maintains the confidentiality of service provider or third-party secrets that are used in the scheme. These secrets could involve, for example, hashes of known harmful content or a fragile machine learning classifier. Formally, a content moderation scheme has server privacy if a computationally bounded client has at most negligible advantage in a security parameter at determining whether they are interacting with the real content moderation scheme as opposed to an "ideal" scheme where a corrupt client learns only the responses to chosen content.

While the server privacy property is not unique to E2EE settings, it poses a significant challenge under E2EE because a service provider cannot trivially implement all content moderation server-side. Server privacy also poses significant difficulty for transparency, because a client cannot readily verify that the server is only detecting the content it claims it is detecting. In Section 7, we discuss recent proposals for verifying important properties of server secrets and suggest directions for future work on the topic.

Client privacy. The primary concern in content moderation under E2EE, in comparison to ordinary content moderation, is respecting *client privacy.* E2EE establishes full confidentiality for content against the service provider and other third parties to a communication. Any content moderation scheme that automatically discloses information about content to a third party represents a reduction in the fundamental security and privacy guarantee.

If a content moderation system achieves the same confidentiality, integrity, and authenticity guarantees as the underlying E2EE channel, we call it *fully client private.*

Definition 2.3 (Full client privacy)**.** A content moderation scheme over an E2EE channel has *full client privacy* if it maintains the same confidentiality, integrity, and authenticity properties as the underlying E2EE channel. In particular, both detection and response are conducted in an end-to-end way, without giving any new information to the service provider or another third party.

In some problem contexts, stakeholders may believe that full client privacy does not achieve content moderation goals. Proposals to counter CSAM, for example, often center on alerting child safety groups and law enforcement. System designs like these, which would automatically notify a third party about problematic content, do not offer full client privacy. They could, however, offer a reduced guarantee which we term *partial client privacy.*

Definition 2.4 (Partial client privacy)**.** A content moderation scheme over an E2EE channel has *partial client privacy with respect to class C* if it maintains the same confidentiality, integrity, and authenticity properties as the underlying E2EE channel for all messages *except* those in class C. For messages in class C the scheme maintains the integrity and authenticity guarantees but may not provide confidentiality against designated third parties.

The class C of messages for which the confidentiality guarantee does not hold could be positives of a ML classification scheme, items that share a perceptual hash with a list, or other categories of content. The ideal version of this class, $\tilde{C}$, is a theoretical class (like "content perceptually similar to items on a particular list") that is measured imperfectly by the real class C (e.g., "content sharing a

PDQ hash with an item on that list"). This measurement will have both false positives and false negatives, and the false positive rate is the frequency of non-problematic messages whose confidentiality was nevertheless breached by the content moderation system.

In partially client-private systems, the moderator is in essence granted a special key that can be used to read elements of C as an exception to the E2EE system. This makes the choice of "ideal" class $\tilde{C}$ and trustworthiness of implementation C of utmost importance: an arbitrary or corrupted C could effectively reduce the entire channel to an unencrypted one. Although a full policy treatment is out of scope of this paper, we discuss some approaches to proving information about C in Section 7.

The terminology of full and partial client privacy. The term "client privacy" was originally used by Kulshrestha and Mayer [212] to mean what we here call "partial client privacy." We use the terms *full client privacy* and *partial client privacy* for two purposes. First, we seek to emphasize that a scheme that does not achieve even partial client privacy should not be called E2EE at all. Second, there is a meaningful difference in the privacy guarantee offered by full and partial client privacy: the channel or moderator effectively has keys for content in C, which makes the choice and trustworthiness of C of utmost importance. One could argue that partially client private systems, too, should not be called E2EE. Some civil society groups and researchers take this position [187, 244, 297]. While these observers are very uneasy with partial client privacy, and we respect their perspective, we consider partial client privacy within the scope of this systematization. The concept, as we formalize it in Definition 2.4, is self-consistent and central to the current discourse on content moderation and E2EE (e.g., [1, 33, 187, 212, 223]). We hope the term "partial client privacy" makes clear that the concept has a coherent definition, that it still provides meaningful security and privacy guarantees, and that it represents a significant departure from the typical E2EE setting of full client privacy.

Maintaining indistinguishability for non-problematic content. Note that our conception of E2EE-compatible content moderation maintains an *indistinguishability* notion of confidentiality for content outside class C. Formally, we require at least indistinguishability against chosen plaintext attack [189] for this content. We therefore exclude moderation that functions by, for example, sending hashes of all messages to the server to perform a match. Such an approach would allow a service to check whether a message contained any particular piece of media (say, a divisive political meme) by hashing the media and comparing the hash of the message. The service provider could essentially monitor user content, well beyond class C. We believe these constructions so completely defeat E2EE guarantees that they cannot defensibly be considered compatible, and we emphatically reject these proposed directions from both researchers and governments (e.g., [5, 200, 341]).

The risks of content moderation under encryption. The implementation of any content moderation system under E2EE reduces barriers to future surveillance. Service providers have mixed records responding to external pressure to monitor or censor content [13, 270, 358]. Due to the added power of the server to read some messages in partially client private systems, we see partial client privacy as an especially vulnerable setup: there will be more pressure to monitor more kinds of content for a variety of purposes, and the system will present a more attractive target for external attackers wishing to exploit the system, complicating one of the core benefits of E2EE. The risks are lower, but not zero, for fully client private systems: these can be adapted into partially client private systems by changing a small amount of client code to report detections to the server rather than keeping them on the client device [1].

Additionally, it is a well-documented phenomenon that even if a particular deployer of a content moderation scheme keeps the system tailored for a narrow purpose, other organizations may reuse the same system in more censorious settings [278, 302].

These topics emerge in many areas of tension between law enforcement, safety, and privacy; prior work offers thorough descriptions of the risks and mechanisms of bypassing encryption to expand surveillance [1, 2, 112, 264] or censorship [37, 50, 100]. Scholarship appropriately takes these risks very seriously. We do, however, hold out hope that further research in this field will improve the frontier of possible tradeoffs and could lead to systems that improve content moderation while maintaining strong security and privacy.

Limitations. Our work has three main limitations. First, we focus on content moderation performed by a centralized service. While we touch on user-driven content moderation methods, we do not explore the broad space of designs that could integrate community decision making into E2EE. We believe many forms of collective and delegated user-driven content moderation are feasible under E2EE, by implementing threshold and permission properties within cryptographic protocols. These constructions would maintain full client privacy and are a fruitful direction for future work.

Second, the focus of this work is on content moderation "of content" as opposed to other forms of moderation like blocking particular users, building user reputation, or verifying the identities of senders. A wide literature on these topics exists even in anonymous settings [23, 24, 88, 161, 208, 320, 375, 378]. We focus on moderation of content rather than users because it has been the most contested territory for E2EE moderation.

Finally, since we included fully client-side moderation approaches in our literature search, we restricted the search to papers which mentioned encryption or privacy explicitly. A full literature search of text and image classification methods for content moderation is beyond the scope of this work.

3 LITERATURE SEARCH: METHODS FOR CONTENT MODERATION UNDER E2EE

In this section we describe we describe our literature search and its initial findings.

Literature search and query terms. We provide a short summary of our literature search methods here, and we include full details in Appendix A. Our literature search initially surfaced papers by running the following queries in August 2022 in computer science and cryptography-related academic venues: content moderation, CSAM, end to end, malware, misinformation, porn, pornography, and spam. The academic venues were ACM CCS, CRYPTO, NDSS, PETS, IEEE S&P, Usenix Security, arXiv CS, and IACR ePrint. We additionally examined the top 200 results from Google Scholar for

Moderation Goal	Summary of Archetype	Sub-Archetypes	Works
Corporate network monitoring	A "middlebox" may act as a firewall, aim to detect intrusions, malicious data exfiltration, or act as some other policy-based content blocker. Generally aims for partial client privacy. Often has server privacy to increase difficulty of evasion and protect intellectual property. See Appendix B for more details about this setting.	MPC or Searchable Encryption	[8, 36, 54, 55, 59, 94, 107, 108, 150, 153, 182, 199, 215, 216, 225, 226, 228, 246, 273, 274, 295, 309, 311, 336, 382] (total: 25)
		Trusted Execution Environment	[78, 104, 157, 158, 213, 271, 294, 295, 328, 339, 356, 357, 369, 380] (total: 14)
		Other	[137, 295, 310, 337, 387] (total: 5)
User reporting (UR) of harassment, abuse, etc.	In secure messaging, enable users to report abusive or misleading messages to a moderator. Message franking (see Section 7.1) introduces additional integrity guarantees.	Message franking	[62, 98, 151, 164, 167, 173, 183, 222, 359, 376] (total: 10)
		Reveal source, traceback, or popular messages	[173, 231, 285, 360] (total: 4)
		Other user reporting	[26, 86, 128, 192, 207, 214, 237, 245, 248, 377, 384] (total: 11)
Spam filtering	In secure messaging or E2EE email, prevent high-volume spam, especially those containing scams. Full client privacy achieved, may or may not have server privacy.	AI/ML via general crypto or MPC	[34, 76, 155, 198, 284, 315, 323, 370, 383] (total: 9)
		AI/ML or matching fully client-side	[4, 86, 128, 138, 207, 214, 352, 366, 377] (total: 9)
		Metadata-based	[58, 176, 262, 368, 384] (total: 5)
		Other	[269, 329, 351] (total: 3)
Malware/phishing, "safe browsing"	In web browsing, messaging, or E2EE file transfer/storage, detect if a particular file or URL is suspicious or malware. Typically has full client privacy, may or may not have server privacy. Omitting 155 papers for detecting malware in encrypted TLS traffic by performing ML classification on the encrypted traffic flow; see surveys [9, 196, 221, 335, 340].	Matching via general crypto or MPC	[80, 81, 169, 205, 296, 303, 333, 349] (total: 9)
		Client-side or metadata-based	[177, 211, 366, 374] (total: 4)
		AI via MPC or federated learning	[64, 122, 333, 334] (total: 4)
		Matching in Trusted Execution Environment	[95, 350, 372] (total: 3)
Parental or educational control	A typical setting is to detect or block specific keywords, websites, or content in TLS traffic, usually with no special hardware.	MPC or Searchable Encryption	[54, 55, 108, 150, 182, 215, 216, 226, 274, 295, 304, 305, 336] (total: 13)
		Trusted Execution Environment	[328] (total: 1)
Child safety	In secure messaging or video chat, detect child sexual abuse. To detect imagery or video, either match against a list using a PHF, or use ML. Server privacy generally considered required.	CSAM detection via client-side AI	[18, 121, 162, 174] (total: 4)
		Matching with a server-held list of CSAM	[33, 83, 212] (total: 3)
		CSAM detection via filename metadata	[7, 280, 288] (total: 3)
		Other child safety	[110, 305] (total: 2)
Other	Papers for moderation of content with few results. Note that this field contains most of the "standard" content moderation topics.	Mis/disinformation	[26, 118, 192, 245, 248, 308] (total: 6)
		Hate/harassment	[237, 305, 307, 384] (total: 4)
		Nudity/NSFW	[281, 323] (total: 2)
		Terrorism & violent extremism	[76, 212] (total: 2)

Table 1: Literature search results for content moderation under E2EE sorted by goal. Some works appear in multiple categories.

"encrypted content moderation" and "end to end encrypted content moderation," the five entries to the recent UK Safety Tech Challenge (UK STC [163]), and the documentation for Apple iMessage, Google Messages, Signal, and WhatsApp. See Appendix A for all exact queries for each venue. These queries formed the initial set of works. We manually inspected those papers to identify relevant works using the criteria listed below, and we then identified additional relevant papers using snowball sampling.

We examined the papers manually to identify both their relevance for inclusion, and to extract the information shown in Tables 2 and 4. To be included for analysis, a work must have had at least one subsection in which it describes, constructs, or implements a content moderation system (thus excluding generic cryptography papers that could be applied to content moderation), and it must achieve at least partial client privacy according to Definition 2.4. Client-side systems must have mentioned that they intended to be used in an encrypted or private setting in order to be included, thus excluding a large number of papers on generic content moderation that could be run on client devices.

Results. Our search resulted in 119 relevant papers including those containing novel cryptographic proposals, metadata-based approaches, and client-side approaches, plus an additional 155 papers on malware detection in TLS traffic by performing machine learning on the encrypted network flow. We also found 19 papers that would have qualified but did not meet our confidentiality guarantee of indistinguishability on non-matches, 22 relevant surveys (of which 12 were about malicious traffic detection), 18 papers about perceptual hash functions, and numerous papers that concerned the topic of content moderation under encryption but contained no method for performing content moderation in E2EE. The 119 relevant papers are shown by moderation category in Table 1. Figure 2 shows the results by category and year. Table 2 shows the details of all non-middlebox works; the detailed middlebox results are deferred to Table 4 in Appendix B.

For the remainder of this work, we walk through our findings and suggestions in the four parts of our content moderation pipeline.

Problem context			Detection																Response		Transparency	
Work	Goal	Medium	User reporting	Metadata	Exact matching	Perceptual matching	Rules/patterns	Machine learning	General Crypto/MPC	Hom./Func. Encryption	Trusted hardware	Client-side	Size of blocklist/dataset	Latency	Communication	Predictive Performance	Security Against Server	Security Against Client	Server privacy	Client privacy	Transparency	Details
Nguyen et al. [269]	Spam	Email					●							×	×		Mal.	Mal.	○		○	
CADUE [262]	Spam	Email		●				●					587k	×	×	FPR: 0.003			○	●		
Wang et al. [370]	Spam	Email						●	●	●			40k	22s	×	Acc: 0.96	S.H.		●	●	○	
Bian et al. [34]	Spam	Email			●		●		●	●			33k	0.5s	×	Prec: 0.89	S.H.	S.H.		●	○	
Pretzel [155]	Spam	Email						●	●	●			33k	1s	×	Acc: 0.99	S.H.	S.H.	○	●	●	Concrete action
Ghatte and Rajmane [128]	Spam	Email	●									●	×	×	×	×	×	×				
SHIELD [198]	Spam	Email						●	●				×	10s	×		S.H.	S.H.		●	○	
Pathak et al. [284]	Spam	Email						●	●	●			206k	41s	Asymp.	×	S.H.	S.H.	●	●	○	
Wang and Chen [368]	Spam	Email		●								●	11k	×	×	Acc: 0.925		S.H.	○	●	○	
Yan and Cho [377]	Spam	Email	●									●	200k	Asymp.	×	FPR: 10^{-8}						
Kong et al. [207]	Spam	Email	●		●										×	×	S.H.	Mal.				
Damiani et al. [86]	Spam	Email	●		●								×	×	×		×	S.H.		★[3]		
Jakobsson et al. [176]	Spam	Email		●									×	×	×			Mal.			○	
Agarwal et al. [4]	Spam	Messaging		●				●				●	2.6M	×	×	Prec: 0.94		Mal.	○	●		
Resende et al. [315]	Spam	Messaging						●	●				5.6k	0.35s	×	FPR: 0.179	S.H.	S.H.	●	●	○	
Tarafdar et al. [351]	Spam	Messaging				●						●	50	×	×	Acc:[4] 1			○	●		
Google Spam Protection [138]	Spam	Messaging						●				●	×	×	×	×			○	★[5]		
Nuruzzaman et al. [352]	Spam	Messaging						●				●	875	0.5s	×	Acc: 0.983			○	●		
Hinky [214]	Spam	Messaging	●		●								50M	0.1s		FPR: Param.			○	●		
CashWeb [58]	Spam	Other		●									×	×	×	×	S.H.	S.H.			○	
Zhang et al. [383]	Spam	Other		●			●						128k	×	×	×	S.H.	S.H.	●	◐	○	
eNNclave [329]	Spam	Unspecified						●			●		150k	100s	×	Acc: 0.744		S.H.	●	●	●	Concrete action
Ryffel et al. [323]	Spam, Nudity	Messaging						●	●	●			×	×	×	Acc: 0.98	S.H.	S.H.	●	●	○	
Constantino et al. [76]	Spam, TVEC	Messaging						●	●	●			308	1147s	×	×[6]	S.H.	Mal.	●	◐	○	
Kogan and Corrigan-Gibbs [205]	Security	Browsing/TLS			●				●				3M	0.01s	1 KB	FPR: Negl.	Mal.(NC)		○	●	○	
Shah et al. [333]	Security	Browsing/TLS						●	●				10k	3.6s	1301MB	Prec: 0.95	S.H.	S.H.	●	●	○	
Chou et al. [64]	Security	Browsing/TLS						●	●	●			19k	0.7s	250 KB	×	S.H.	S.H.	●	●	○	
Ramezanian et al. [303]	Security	Cloud storage			●				●				2M	1.8s	24 KB	FPR: 10^{-4}	Mal.	Mal.	●	●	○	
Hwang and Yoon [169]	Security	Cloud storage			●				●	●			×	246s	×		Mal.		●	●	○	
PriMal [349]	Security	Cloud storage			●				●				131k	3s	2.5 MB	FPR: 10^{-6}	S.H.	S.H.	●	●	○	
Poon and Miri [296]	Security	Cloud storage			●				●	●			×	×	×	×	S.H.	S.H.	●	●	○	
EPMDroid [372]	Security	Other			●						●		600	0.4ms	Asymp.	FPR: 0.010	TEE		●	●	◐	Attestation
Galvez et al. [122]	Security	Other						●					40k	13s	×	F1: 0.959	S.H.	Mal.	○	●	○	
Cui et al. [80]	Security	Other			●				●				1260	0.14ms	1.33 KB	FPR: Param.	S.H.	S.H.	●	●	○	
Pvault [177]	Security	Other					●					●	×	×	×		Mal.			●		
TrustAV [95]	Security	Unspecified									●		30k	×	×	×	S.H.	S.H.	○	●	◐	Attestation
BigBing [211]	Security	Unspecified	●		●			●	●			●	15k	0.519s	×	F1: 0.976	S.H.	Mal.				
Tamrakar et al. [350]	Security	Unspecified			●						●		67M	0.25ms	×	FPR: 0.0009	Mal.	S.H.	●	●	○	
Shaik et al. [334]	Security	Browsing/TLS						●	●	●			100k	9.2s	131.1 KB	Acc: 0.956	×	×	●	●	○	
Verma et al. [366]	Security, Spam	Email						●				●	3k	×	×	FPR: 0.007			○	●		
WhatsApp Suspicious Messages [374]	Security, Spam	Messaging			●		●					●	×	×		×			○	●	○	
Jiang et al. [183]	UR	Messaging	●		●									13ms	×	FPR: Negl.	S.H.	Mal.			○	
Yamamuro et al. [376]	UR	Messaging	●											×	Asymp.							
Hirose [164]	UR	Messaging	●														Mal.	Mal.				
Tyagi et al. (B) [359]	UR	Messaging	●											7.3ms	489 B		Mal.	Mal.				
Huguenin-Dumittan and Leontiadis [167]	UR	Messaging	●														Mal.	Mal.				
Chen and Tang [62]	UR	Messaging	●														Mal.	Mal.				
Dodis et al. [98]	UR	Messaging	●											×	×		Mal.	Mal.				
Leontiadis and Vaudenay [222]	UR	Messaging	●														Mal.	Mal.				
Grubbs et al. [151]	UR	Messaging	●														Mal.	Mal.				
Hecate [173]	UR, Misinfo	Messaging	●											37ms	380 B		Mal.	Mal.		★[7]		
Peale et al. [285]	UR, Misinfo	Messaging	●											0.057ms	160 B			Mal.		★[8]		
FACTS [231]	UR, Misinfo	Messaging	●	●					●				1M	98ms	×	FPR: Param.	S.H.	Mal.		★[7]		
Tyagi et al. (A) [360]	UR, Misinfo	Messaging	●											8us	96 B			Mal.		★[9]		
SAFE [384]	UR, Spam	Other	●				●					●	128k	25s	×	×					●	Concrete action
Apple PSI [33]	CSAM	Cloud storage				●			●				×	×	×	PHF	Mal.	Mal.	●	◐	●	Concrete action
Pereira et al. [288]	CSAM	Cloud storage		●									73k	×		Prec: 0.938						
Al Nabki et al. [7]	CSAM	Cloud storage		●									65k	0.06ms		Prec: 0.84						
iCOP [280]	CSAM	Cloud storage		●									106k	×		Acc: 0.970						
Cyacomb (UK STC) [83]	CSAM	Messaging			●				●				×	×	×	×	×	Mal.	●	●	◔	Mention
Galaxkey (UK STC) [121]	CSAM	Messaging			●			●				●	×	×	×	×	×		○		◐	Consent
Apple Communication Safety [18]	CSAM	Messaging						●				●	×	×		×			○	●		
SafeToNet (UK STC) [110]	CSAM	Other						●				●	×	×	×	×	×		○	●	◔	Mention
DragonflAI (UK STC) [162]	CSAM	Unspecified						●				●	×	60ms		F1: 0.979[10]					◐	Consent
T3K Forensics (UK STC) [174]	CSAM	Unspecified						●				●	×	×	×	Prec: 0.88			○		○	
Kulshrestha and Mayer [212]	CSAM, TVEC	Messaging				●			●				16.7M	10.5s	395 KB	PHF	Mal.	Mal.	●	◐	◔	Mention
Filho and Shuen [237]	Misinfo	Messaging	●									●										
Kazemi and Garimella [192]	Misinfo	Messaging	●										977k	×	×							
Meedan [245]	Misinfo	Messaging	●										15k	×	×							
Reis et al. [308]	Misinfo	Messaging				●						●	810k	×		×			○	●		
Kauwa-Katte Fake News [26]	Misinfo	Messaging	●										×	×	×							
Melo et al. [118]	Misinfo	Messaging		●									400k								○	
Whatsapp Monitor [248]	Misinfo	Messaging	●										×	×	×							
Reich et al. [307]	Hate/harass	Messaging						●	●				10k	2.7s	×	Acc: 0.744			●	●	○	
Ramezanian and Niemi [305]	Hate/harass	Messaging					●	●	●				×	×	×	×	S.H.	S.H.	●	◐	○	
Pandey et al. [281]	Nudity	Other						●				●	85k	85ms		Prec: 0.98			○			

General
●: Property present
○: Property absent
×: Property could not be determined
(Blank): Property irrelevant
★: Special (see footnote)

Efficiency
Blocklist/dataset size is for the largest described.
Latency and communications are for the size of the largest blocklist/dataset, where provided.
Asymp.: Asymptotic but no concrete efficiency provided

Predictive Performance
Highest value given among FPR, Acc, Prec, F1, AUC.
FPR: False positive rate
Acc.: Accuracy (total correct over total classifications)
Prec.: Precision
F1: F1-score
Param.: Tunable parameter with FPR/FNR tradeoff
Negl.: Negligible in a security parameter

Security & Client Privacy
S.H.: Semi-honest
Mal.: Malicious
NC: Non-collusion assumption
TEE: Assumption that TEE use is honest and secure
●: Full client privacy
◐: Partial client privacy

Table 2: Details of non-middlebox methods for E2EE content moderation found in our survey. See Table 4 for middleboxes.

4 PROBLEM CONTEXT

In our terminology, the *problem context* is the externally-provided goal of what kind(s) of content the service provider wishes to moderate. In Table 1 we sorted the works in E2EE content moderation by their moderation context. The goal of moderation highly influences the design choices for detection, response, and transparency. For an example set of considerations on how this is true in or out of the encrypted setting, see three short case studies in Appendix C. For the rest of this section, we describe our findings on how the problem context influences the E2EE detection and response, and suggest that future security research be more specialized.

4.1 Problem context affects choice of detection and response mechanisms

Naively, we would expect different detection methods to be used in different problem contexts. Our literature search bears this out, as shown in Tables 2 and 4. Overall, TLS traffic inspection mainly used rule and pattern matching (93%), identifying misinformation relied mainly on user reporting (71%), detecting malware URLs or binaries mainly used exact matching (58%), and other categories were more mixed. Interestingly, perceptual hash functions were rare, which we discuss further in Section 5.

We also saw differences in the client privacy of the scheme based on problem context. Works detecting threats or annoyances to the user, such as malware or spam, were nearly universally fully client private. Works focusing on child safety concerns were more mixed: of the eight proposals whose main focus was child safety, two offered partial client privacy [33, 212], two offered full client privacy [18, 110], and the remaining four were prototypes agnostic as to the final setting of client privacy [83, 121, 162, 174].

Although these choices naively make sense, we are not aware of quantitative research that evaluates the difference in effectiveness of these systems across different problem contexts in encrypted settings, and we suggest this as an area for future research.

We also observed detection and response methods that were specific to certain problem contexts. In the literature on countering spam, for example, we saw specific interventions that disincentivized its creation by forcing the sender to pay in money or computational work each time a message is sent [58, 176], taking advantage of the fact that spam is by its nature sent to many recipients simultaneously. We did not see any other examples of such methods being applied, and we would expect them to be inappropriate for most problem contexts. For misinformation, two works specifically attempted to detect or limit "viral" disinformation [118, 231] rather than general misinformation. We hope to see more research and innovation on this topic both in and out of the encrypted setting.

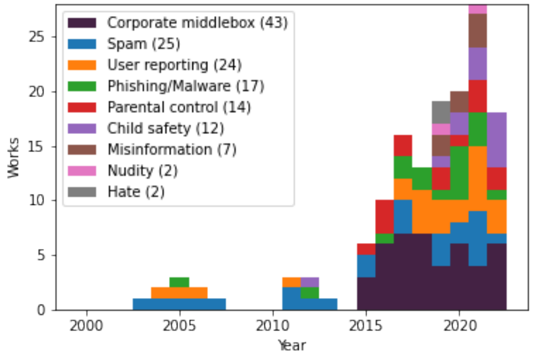


Figure 2: E2EE content moderation papers by year and topic. Papers about multiple topics appear in multiple categories.

4.2 A need for interdisciplinary work and improved domain expertise

Figure 2 shows the moderation goals of all content moderation papers found in our literature search, excluding 155 papers on identifying malware and network anomalies by performing machine learning on encrypted TLS traffic flows. It shows that most cryptography and security work for exposing rule-violating content was performed in the corporate setting of monitoring TLS traffic for security purposes, with spam mitigation a distant runner-up. Additionally, aside from a few works with creative response mechanisms we discuss in Section 6.2, very little work unites end-to-end encryption with other potential content moderation responses [129, 135], such as lowering content visibility or reputation, or (outside the spam setting) disincentivizing its creation in the first place.

We thus call for more interdisciplinary research and greater attention by the security community to the literature on specific content moderation topics. We see two barriers to such work. The first is simply researcher inertia: for years, cryptography had no reason to interact with content moderation topics other than security-adjacent concerns like malware. Cryptographers and security researchers lack domain expertise in topics like misinformation and child safety that were not traditionally considered security issues, and on the flip side the bulk of social science research on these topics generally either avoids the difficulties associated with encryption, describes them only briefly, or has few answers to share (e.g., [50, 123, 131, 156, 325]). Another obstacle is the difficulty of obtaining data, especially for highly sensitive issues of child safety.

5 DETECTION

The main body of this section describes the results of our literature search on the detection methods used in different content moderation contexts. In Section 5.1 we discuss automated detection of content, in Section 5.2 we discuss methods based on user reporting and on metadata (of *content*, not of *users*, as mentioned in Section 2.3). In Section 5.3 we discuss the fundamental assumptions all these works place on client behavior.

[3] Peers and super-peers learn reported messages.
[4] Extremely small sample size.
[5] Reveals telephone number.
[6] Crypto induces negligible additional errors. Accuracy of the classifier not given.
[7] Reveals the source.
[8] Reveal the source or forwarding tree.
[9] Reveals the forwarding path.
[10] Obtained from https://www.dragonflai.co/ on 11/23/2022.

Paradigm	Efficiency	False positive rate	Client privacy?	Server privacy?	Crypto methods[11]	Threat model	Evasion
Exact matching	Fast-medium (1ms-2s)	Negligible ($\sim 10^{-38}$)	Full	Possible but rare	HE, FE, PSI, or client-side	Usually malicious	Easiest
Rules/patterns	Fast ($40\mu s$-400ms)[12]	Not determined	Typically partial	Usually	SE, TEEs, or MPC	Usually semi-honest	Moderate-easy
Perceptual matching	Similar to Exact	Medium-high (10^{-8}-10^{-3})[13]	Typically partial	Usually	PSI or client-side	Usually malicious	Moderate-hard
Machine learning	Slow (500ms-10s)	High (10^{-2}-10^{-1})	Usually	Sometimes	HE, FE, or client-side	Usually semi-honest	Not determined

Table 3: Paradigms of automated moderation methods. All numbers aside from the PHF FPR estimates[13] are one significant figure of the first and third quartiles from our literature search results.

5.1 Automated detection of content in the literature

Most works we examined (all middlebox works and 72% of non-middlebox works) performed some kind of automated detection of content. In Section 4 we discussed how different problem contexts are amenable to different detection methods. In this section, we examine the technical properties of those different detection methods, including cryptographic methods, accuracy, efficiency, and threat model. The four main paradigms in content-based detection are:

(1) **Exact matching**: typically accomplished with Private Set Intersection, generic Multi-Party Computation, Searchable Encryption, or client-side lists.
(2) **Rule or pattern matches**: typically accomplished by Searchable Encryption or Trusted Execution Environments (TEEs).
(3) **Perceptual matching**: typically accomplished by Private Set Intersection or client-side lists.
(4) **Machine learning classification**: typically accomplished by using Homomorphic or Functional Encryption, or by running the classifier client-side.

We provide a summary of the findings in Table 3, more details on individual schemes can be found in Tables 2 and 4. More details and references for cryptographic methods are in Appendix E.

Exact matching. Exact matching refers to systems which attempt to detect content that matches exactly[14] with a particular list of problematic content, typically by comparing the cryptographic hash of the content against the list. Exact matching was primarily used for detecting malware-hosting web URLs, malware itself, or performing exact queries within TLS network monitoring. The TLS middlebox functions generally provided partial client privacy [182, 228], but all other systems provided full client privacy.

The server-private works mainly achieved their goals via homomorphic or functional encryption, or general cryptography protocols (see Appendix E for more information on cryptographic tools). Those that did not provide server privacy mainly faced a technical challenge of compressing the information to be stored on the client device as much as possible. The works that used exact matching typically had extremely low false positive rates, typically 0 or negligible in the cryptographic sense. Those works that did have higher false positive rates typically made use of a bloom filter [38] with a tunable parameter for false positives; these also had full client privacy, avoiding privacy issues from a false positive.

We did not include certificate transparency in our SoK since it is a moderation of identities rather than content. However, certificate transparency often performs fully client-private exact matching with negligible or zero false positives, often by pushing as much as possible to the client [219, 263, 286, 330, 338, 343, 345].

Rules and patterns. A second approach to content moderation in E2EE is to set a rule, predicate, pattern, or other simple search query, and to selectively reveal or block messages or traffic whose plaintext matched the search.

This detection approach was overwhelmingly used in TLS middleboxes implementing deep packet inspection via searchable encryption or trusted hardware; see Appendix B for details. Less frequently, this approach was also used for other use cases, including text filtering [269], and detection of spam [34, 383], malware and phishing [177, 374], and hate speech [305, 384].

The very core of policy-based moderation, of course, presupposes that the service provider has some relatively concise description of the policy the clients should follow. In some cases, this could become a short list for matching. A service provider could, for example, list keywords or specific content that the matching protocol would detect and disallow. However, a rule or pattern can often achieve more efficient results than a large list. Lists and keywords, when searched directly, are also easily avoidable without more sophisticated learning [388] and many of the papers we will discuss shortly about ML-based detection describe the inadequacy of policy-based detection for many tasks.

Unfortunately, the technologies most frequently used to implement this paradigm—searchable encryption and trusted execution environments—have some drawbacks that would require care to integrate with full client privacy. Searchable encryption frequently exhibits leakage that is unacceptable for E2EE (e.g., [171, 186, 279]), and only some searchable encryption schemes are compatible with forward secrecy [46, 385]. Trusted hardware also has many known side channel attacks, many of which are known to fully exfiltrate encryption keys [60, 61, 144, 229, 272, 363]. In the enterprise setting,

[11] See Appendix E for descriptions of cryptographic objects: Homomorphic encryption (HE), Functional encryption (FE), Private set intersection (PSI), Searchable encryption (SE), Trusted Execution Environments (TEEs).

[12] The difference between exact and pattern-based matching is likely reflective of the fact that the middlebox setting uses trusted hardware more than other approaches.

[13] The *worst* estimate of 10^{-3} presented here for PHFs' false positive rate is the optimistic *best* performance of all but one PHF provided in Figure 3 of Jain et al. [175]. The best estimate of 10^{-8} is approximately the claimed FPR of Apple's NeuralHash [21], which is the best FPR of any PHF we know of.

[14] Strictly speaking, cryptographic hashing is not "exact" matching; the chance of collision is negligible in a designer-chosen security parameter. However, that chance is typically set around at most $2^{-128} \approx 10^{-38}$. In contrast, the lowest false positive rate these authors know of among perceptual hash functions is 3-in-100-million $\approx 10^{-8}$ [21]. Assuming 4.5 billion non-problematic images shared daily on WhatsApp (a conservative 2017 estimate [175]) the best perceptual hash function would yield about 135 false positives per day; a cryptographic hash would not yield a single false positive for more than the age of the universe in expectation. Thus, although matching with a cryptographic hash is not perfectly exact, it is exact for all practical purposes and is clearly in a different regime than perceptual hashes.

where it is common for middleboxes to fully decrypt TLS traffic and read plaintext packets [89], this privacy leakage presents only moderate concern. But these issues are a major problem in E2EE settings where privacy is the norm.

Perceptual matching. A challenging aspect of content moderation is that senders of problematic content will often try to evade the detection mechanism. Users bypass word filters by misspelling words [149, 282], and for images and video users use common methods to evade matching: altering a small number of pixels, changing the size, rotating the image, or changing the aspect ratio [84, 109]. These approaches will completely alter the cryptographic hash of an image or message, but they may not interfere much with human perception of the content.

The response to this evasion is the *perceptual hash function* (PHF) [364]. PHFs are locality-sensitive hashes [133] that return the same or similar hash values even if the input has been put through a class of perturbations. A variety of PHFs are used in industry for content moderation, primarily for images and video. These include Microsoft's PhotoDNA [254], Facebook's PDQ for images and TMK+PDQF [250] for video, and Apple's proposed PHF for CSAM called NeuralHash [20]. The academic literature contains more PHFs [35, 103, 109, 202, 267, 276].

PHFs do *not* aim to achieve the same level of collision resistance as cryptographic hash functions (CHFs), instead they aim to provide collision resistance for images that are not perceptually similar [373]. Unsurprisingly given their goals, PHFs have higher false positive rates than CHFs even for unrelated images [159, 175, 346], between 1-in-1000 [175] to 1-in-10-million [341] to 3-in-100-million [21]. See Appendix D for a summary of recent attacks on PHFs, and for benchmarks of common PHFs, see [355].

We saw very few examples of PHF-based content moderation in E2EE in our literature search. PHFs appear only in Reis et al.'s 2020 work for misinformation in WhatsApp that provides full client privacy [308] and the two 2021 partially client private proposals for matching CSAM [33, 212]. A few more papers we examined use locality-sensitive hashes for identifying spam similar to previously-seen spam [86, 351, 380], however aside from these PHFs are rare in the literature we examined. We hope to see both improved PHFs and improved scrutiny of PHFs in the future.

ML Classification. Our final category of automated content-based detection is machine learning (ML) classification. Over our entire search, 24% of papers performed privacy-preserving ML to do a content moderation task, spread across the moderation goals of improving security, spam, and other topics (see Tables 2 and 4). The year 2021 saw newfound activity for using ML to detect CSAM in various forms: Four of the five contest entries to the recent U.K. Safety Tech Challenge [163] utilized a client-side ML model to detect CSAM [110, 121, 162, 174], including self-generated CSAM [110], and Apple's Communication Safety uses ML detecting nude pictures in children's chat messages [18] with full client privacy.

Our accuracy findings (see Table 3) provide evidence toward the notion that ML typically has a higher false positive rate than matching via perceptual hash functions [223], however further work is needed to see if these results remain true for the best classifiers.

Of the machine learning based detectors in our search, 54% used some form of cryptographic protocol to aid private computation of the machine learning, and 39% were implemented client-side.

All but three ML-based designs maintained full client privacy; those three were in especially controlled settings of enterprises or parental control[8, 304, 305]. (Three UK Safety Tech Challenge entries were agnostic as to the client privacy setting [121, 162, 174].)

5.2 Approaches that do not rely on automated content detection

Just under half of the non-middlebox works (46%) incorporated user reporting or analysis of content-agnostic metadata.

User reporting. User reporting was a primary detection mechanism of 33% of non-middlebox works. It was used most frequently in the context of mis/disinformation and general reporting of abusive messages. For mis/disinformation, one key area was tiplines or monitors for fake news on WhatsApp [26, 192, 237, 245, 248]. Some spam and malware works also relied on user detection to identify malicious messages, then blocked future copies automatically.

In Section 7.1 we will discuss *message franking*, an important component of E2EE content moderation which adds cryptographic verification to the process of user reporting, ensuring that malicious receivers cannot frame senders for content they did not send, and honest receivers can prove a sender really did send a particular message. These works are displayed with the goal of User Reporting (UR) in Table 2, but we defer in-depth discussion of these works to Section 7.1 since they implement an accountability property on the existing detection mechanism of user reporting.

Metadata-based measurement. 14% of non-middlebox works incorporated some analysis of metadata about content, as opposed to (or in addition to) analysis of the content itself. These fall into two main categories: One group identifies identifying spam or phishing alerts; these works attempted to discourage or block spam based on volume or the existence of particular links. The other group of metadata-based works were those that attempted to detect encrypted files containing CSAM by performing machine learning on filename metadata [7, 280, 288]. Some works also incorporated metadata-based analysis alongside content-based analysis.

We encourage research in new methods using non-content sources, and research measuring the efficacy, accuracy, and other properties of user reporting and metadata-based measurement compared to each of the automated content-based detection paradigms.

5.3 Assumptions on client behavior

Any detection mechanism for catching clients who are trying to evade detection requires at least some assumptions beyond the cryptographic threat model. At one extreme, it is impossible to thwart a sufficiently motivated and sophisticated colluding sender and receiver from using the channel to send problematic content: the sender and receiver could run their own key exchange on top of the existing channel and build their own layer of encryption on top of the existing one. Any detection scheme capable of detecting the encrypted content would be able to break encryption generally. However, adding this additional layer of encryption requires a good amount of technical sophistication and cooperation on the part of both the sending *and receiving* client; in situations where both the sender and receiver are colluding, we conjecture that malicious senders would prefer to send perceptually-recognizable problematic content (although we know of no research into this question).

Much of the middlebox literature explicitly assumes the existence of one honest client as a core requirement, to avoid this exact problem and assume away any malicious out-of-band pre- or post-processing [336]. The appropriate choice of client threat model is highly dependent on the problem context.

The remaining automated detection options become a "cat-and-mouse" adversarial game: the platform tries to cast a wide enough net to catch users exchanging problematic content without catching unacceptably many false positives, and malicious users try to modify the content they send just enough so that it evades detection but is still recognizable. The existence of perceptual hash functions is a concession to this cat-and-mouse game: PHFs have high false positive rates compared to exact matching, but are harder to evade.

If PHFs become a key feature of content moderation under E2EE, then their improvement and analysis will lead directly to improved accuracy, simultaneously reducing the privacy loss, increasing the difficulty of evasion, and increasing the difficulty of maliciously induced false positives (i.e., where a user sends an image that appears innocuous but that has the same hash as harmful media). Recent work analyzes the security properties of perceptual hash functions [84, 102, 109, 175, 300] (see Appendix D for more) and we encourage more research on this front.

6 RESPONSE

In this section we briefly describe the results of our search on client privacy (see Appendix F for more details) and then we go into the response mechanisms we saw in the literature that are unique to E2EE and irrelevant for standard content moderation. We see fruitful areas of future research there which go beyond the proposals to improve detection in a more obvious way.

6.1 Client privacy in moderation response

Of the non-middlebox works for which client privacy was relevant and we could identify the a client privacy setting, 88% offered *full* client privacy. At the same time, within the TLS middlebox literature, 98% of middlebox designs offered *partial* client privacy.

This difference is stark. We see at least two factors that explain this gap. First, as we observed in our discussion of rule-based detection in Section 5, it is common for non-privacy-preserving middleboxes to break the TLS connection entirely. This makes partial client privacy a step up in privacy rather than a step down, as it would be for most content moderation under E2EE. Second, in the corporate settings where middleboxes are frequently used, there is often an expectation that all activities are monitored that is absent in typical E2EE deployments. A 2017 survey by O'Neill et al. [277] shows much stronger public support for general TLS proxies in the corporate setting — and to a lesser extent, schools — than any other context. Understanding these factors, as well as any other considerations that help choose full or partial client privacy for different situations, is a useful area for future research. See Appendix F for more details on the client privacy difficulties for different detection paradigms.

6.2 Responses unique to end-to-end encryption

In the non-encrypted setting, already a wide variety of content moderation responses exist, including banning, suspending, lowering visibility, fining or withholding money, and so on [135].

Most papers in our literature search handled content moderation responses in the same way the issue would be handled in a standard content moderation setting: either informing the server or a moderator of the detection (allowing whatever actions the server deems appropriate), warning the client of the detection (as in malware), or invisibly sending the content to another folder until the client re-identifies it (as with spam).

However, some literature utilized specific information about the E2EE setting that enabled new responses that are not applicable in the standard setting. These mainly revealed new information about a previously-encrypted message, once it has been detected by user reporting or automated methods.

Peale et al. [285], and later Issa et al. [173], implemented *source tracking*. Source tracking effectively encodes the original source of a sender into a message: if a message is originally sent from A to B, B forwards it to C, and C reports it, the service provider will learn that A was the original sender. Peale et al. also create an extra confidentiality property of the forwarding path: if a client receives the same message from two different sources, the "tree-unlinkability" property ensures that the client will not know whether the message was received via the same forwarding path both times.

Tyagi et al. [360] implemented *traceback* for E2EE messaging: after a detection, the service provider gains the ability to "trace" the forwarding path the message took to get to the receiver in one of two ways. Suppose A sends a message to B and C. B forwards the message to D, and separately, C forwards the message to E. E later reports the message. Under *path forwarding*, the service provider learns the message path $A \longrightarrow C \longrightarrow E$, and could take action on that path (e.g., in the setting of misinformation, could warn the users after the fact that the information was suspicious). Under *tree forwarding*, the service provider learns the paths $A \longrightarrow C \longrightarrow E$, and also the path $A \longrightarrow B$, though not the fact that B forwarded the message to D.

In a different take on user reporting, Liu et al. [231] described an approach for revealing messages once they reached a specific *threshold* of reports globally across all users, by constructing a "collaborative counting bloom filter." Their goal was to reveal misinformation that was especially "viral" and thus by definition reached a large number of users; after this the server would be free to take action on those specific images (e.g. by sending it to client devices to perform matching to append a warning if it is seen again by future clients). This is reminiscent of earlier schemes that perform similar goals for spam [86, 207].

We encourage more researchers to think outside the box of "binary detection," which captures the majority of literature, and to continue exploring content moderation responses that could respond to societal harms while respecting full client privacy.

7 TRANSPARENCY AND ACCOUNTABILITY IN CONTENT MODERATION

One of the strongest criticisms of the 2021 Apple CSAM detection tool was the risk that the system would inevitably bow to pressure to expand the use cases of the surveillance system to other purposes [1, 70, 117, 132, 145, 240, 244, 252, 283, 313, 326]. As others have pointed out [187], once the plaintext can be processed for one purpose (detecting CSAM), additional exceptions could be carved out for other purposes (e.g., detecting threats to national security, terrorist content, hate speech, misinformation, or more), or the system could be exploited by external malicious parties to undermine privacy in other ways. There is also likely to be significant international pressure to censor or track specific political memes [117, 270, 358].

The tension between free speech and content moderation was already an issue in non-encrypted content moderation [96, 101, 129, 143, 147, 204, 218]. However, the stakes are higher in content moderation that bypasses encryption, because encryption is one of the few methods by which over-broad surveillance can be avoided. Thus, under encryption, maintaining transparency, oversight, and verification of content moderation is paramount.

Numerous transparency mechanisms have been proposed and enacted in the non-encrypted setting, including community guidelines and terms of service, aggregate transparency reports [140, 251], legal or contractual boundaries [101, 218], oversight boards [106], third-party audits (e.g., [33, 261]), and a variety of other governance approaches [90, 101]. These approaches should be applied in E2EE content moderation as well, though they also have recognizable limits even without encryption [143].

We propose that cryptographic means of enforcing transparency be utilized not only in the E2EE setting but also the unencrypted setting. Section 7.1 describes transparency mechanisms we saw in our literature search, and in Section 7.2 we propose future research.

7.1 Transparency methods in the literature

Verifying the server. In the Transparency column of Tables 2 and 4, we identify any transparency properties by which the client can verify the system's correct behavior.

We identified 82 works where the client must rely on some promise or information held by the server (e.g., a secret dataset or model, or an honesty assumption). Of these, 49% made at least some mention of the need for transparency. Of the works that mentioned some form of transparency goal, 30% provide no concrete guidance on how to achieve it. An additional 20% make an explicit assumption of honesty on some party, typically a "rule generator" party with no other input. A small number of works mention third-party audits (5%), or getting consent from clients (5%).

Works that use trusted execution environments often mention attestation as a means of establishing transparency (77%), however in all but one case [329], the attestation would only be verifiable to the server, not the client.

The remaining four works had some concrete actionable transparency proposal: Section 4.4 of the Pretzel spam detector for E2EE by Gupta et al. [155] is dedicated to transparency issues. In addition to preventing all but one bit of leakage against a malicious server, Gupta et al. also discuss a particular client action that would allow the client to "opt out with plausible deniability" by garbling the incorrect function without the server's knowledge. Second, Apple's PSI proposal for detecting CSAM [33] suggested cryptographic methods for verifying the server's set that could be implemented by a third-party auditor in a secure environment to ensure that the content moderation system only relied on CSAM hashes from child safety groups [33, p. 13]. The eNNclave work [329] made use of the trusted hardware attestation functionality as well, but the client checked the code rather than the service provider. Fourth, SAFE [384] describes a protocol in which clients share hashes of items they wish to filter (e.g. hate or spam). The protocol used a Merkle tree [249] to authenticate the filters.

We also know of two works that build novel cryptographic transparency mechanisms which became available in preprint after the conclusion of our literature search. First, Bartusek et al. [29] offer the ability to enforce a predicate on the class C that forms the partial client privacy exception. They describe constructions of "set-pre-constrained" group signatures and encryption in which all parties can verify, for example, that a list C used for matching is capped at size at most n, or other predicates about C. Second, Scheffler et al. [327] perform policy analysis of partially client-private systems using exact or perceptual matching for CSAM, and suggest three protocols to improve their transparency: (1) use threshold signatures among child safety groups providing hash sets, (2) allow the server to prove that particular elements are *not* in the hash set, and (3) ensure that users with matching content (true or false positive) eventually learn that their content was revealed, after a delay.

In general, we see cryptographic transparency and auditability methods as a useful area of future research: many of the works mentioned the importance of transparency, but few had concrete methods for allowing the client to verify the server's behavior. Since this topic is also at the heart of the debate over content moderation in E2EE generally, we believe it is a worthy research agenda within both technical and non-technical approaches.

Fully client-side content moderation. Some works avoid the problem of verifying the server's behavior by avoiding the server's direct involvement at all parts of the content moderation pipeline, and thus control of the scheme is essentially always held by the client. In these settings, false positives or negatives may cause other problems such as being unable to view important messages, but no privacy was lost. Many of these works also state or imply that they are meant to be "soft" moderation methods: the client is able to bypass the categorization if they wish (e.g. they can view the "spam folder" and remove items from it). In these designs, code inspection and continued use should demonstrate the correct functionality of the system; there is no remote server that needs periodic inspection or auditing. A privacy improvement can also be achieved even for partial client privacy, by reducing the amount of information that is sent to the server [185] or by performing detection only after a threshold of problematic content was detected [33, 231].

A future research line fusing the literature on E2EE content moderation, verifiable programming, privacy, and systems security would help develop the transparency and security properties needed for content moderation running fully on client devices.

Authenticating client reports. In addition to verifying the server's behavior, one strong area of cryptography research in the literature is in the realm of verifying client behavior during user reporting: users are cryptographically prevented from forging a user report that would frame an innocent sender. Technically speaking, *message franking* is a three-party protocol between a sender, receiver and moderator that adds two additional accountability properties to user reporting, at the cost of some deniability. In a message franking scheme, senders always attach a "signature" to every message sent, in such a way that if a sender sends problematic content to a receiver, that receiver can report the message to a moderator, who will check the signature to ensure the sender truly sent it. This cryptographically ensures that receivers cannot report to the moderator messages that are "forged" to appear as if they were from the sender; the moderator cannot be convinced any party sent a message they did not send. These schemes have two key accountability properties [151, 359]:

(1) *Sender binding*: If a sender sends a message that can evade the moderator's verification, the receiver will refuse to validate the message at all, treating it as malformed.
(2) *Receiver binding*: The receiver is unable to forge the sender's signature to the moderator.

These two properties together also ensure that no one can impersonate a sender to the receiver [359].

Message franking is a key component of user reporting in E2EE secure messaging. User reporting is an attractive option in E2EE because one of the "ends" of the communication must take positive action before any new information is revealed to the server.

These accountability properties come at a subtle cost to the typical deniability property of E2EE (see Section 2.1). Under the proposed designs of message franking the E2EE deniability property will no longer hold against the moderator, although it will still hold against third parties who do not know the moderator key. See the full version of [359] for variants of message franking with different deniability properties. All the message franking schemes we reviewed in Section 3 achieve some variant of these transparency properties, including some schemes that are compatible with sender-private networks and some that achieve the forward secrecy property of E2EE in addition to accountability.

7.2 Suggested future research in transparency

We saw two key areas for future research in transparency that were not well-examined in the current literature.

Privacy-preserving protocols for aggregate statistics. Prior work has stressed the difficulties that E2EE will pose on measuring the accuracy and effectiveness of a content moderation system (e.g. [156, 188]). Service providers are understandably nervous about losing the ability to measure the aggregate performance of their systems under E2EE; a reason to avoid E2EE in the first place.

We propose that *privacy-preserving aggregate telemetry and measurement* of the content moderation system is warranted and helpful for both clients and service providers alike. These systems could be based on methods for secure E2EE telemetry [72, 166, 241], federated learning [160], accountability in other settings [116], or methods for privately measuring aggregate information in Tor [63, 77, 111, 178, 247, 367].

Giving service providers access to telemetry and aggregate statistics will also enable measurements and improvements of other aspects of the system like algorithmic fairness [65, 201], allowing continuation of techniques used in the non-encrypted setting [27, 32]. The capability of gaining aggregate statistics about a detector is also likely to make the implementation of E2EE and other privacy-preserving systems more palatable for services currently on the fence about providing E2EE.

We believe the development of these systems would serve several purposes. Not only would they allow service providers to monitor and improve their content moderation systems in an aggregate way, they would also help clients verify certain claims about the content moderation system, such as its false positive rate on a global scale. The ability for the public to audit or verify these claims is a key principle in this and other areas of cryptography policy [1, 112].

Enabling and enforcing notice, appeal, and redress in E2EE systems. The Carnegie principles and Abelson et al. [1, 112] detail the importance of the principle of "Accountability: When a phone is accessed, the action is auditable to enable proper oversight, and is eventually made transparent to the user (even if in a delayed fashion due to the need for law enforcement secrecy)" [112]. In contrast to the aggregate transparency mechanisms suggested above, we also note that individual notice to users who have had their content moderated is an important aspect of moderation. Providing explanations for content removal above and beyond the fact of removal itself has also been found to improve user behavior in the future [180]. Appeal is also critical for any content moderation system, and reporting moderation decisions to the user is an important prerequisite for any appeal and redress mechanisms.

These ideas receive very little attention in the technical literature on E2EE content moderation. Partially client-private systems which report detections to the server rather than client sometimes have no mechanism in place—cryptographic or otherwise— for ensuring the client receives notice for her moderated content, let alone the ability to appeal it.

For automated systems in E2EE, appeal also poses a technical challenge: if a sender sends a benign message that is falsely flagged as problematic content, and the content is reported to a human moderator who determines the content is not problematic, then what technical means should be taken to ensure the client is able to send their message as soon as possible? Naive solutions like allowlists leave many privacy and efficiency issues unanswered, and so we encourage future work in this area as well.

8 CONCLUSION

Grimmelmann summarized the difficulty of content moderation by saying that "responsible content moderation is necessary and . . . responsible content moderation is impossibly hard" [148]. The same is doubly true on both counts under end-to-end encryption: encryption allows people to hide bad behavior from reasonable moderation, but also remains one of the only bastions against unreasonable government and corporate surveillance.

Although there will likely never be a perfect content moderation system, let alone one operating under E2EE, the current systems leave much to be desired and have tractable problems that can be addressed with future research over the coming decade. We hope our work provides a foundation on which to do further research that will enable forward progress towards this demanding goal.

ACKNOWLEDGMENTS

Many thanks to Anunay Kulshrestha for assisting in manual extraction and confirmation of information from the works chosen for analysis in the literature search. Additional thanks to Andrew Fishberg for assistance in the process of downloading and running automated extraction of forward and backward citations in the initial phase of the literature search. We also thank the anonymous reviewers for their high-quality feedback and suggestions for improvements to this work. This work was supported by the Center for Information Technology Policy at Princeton University, and also by a cryptography research grant from Ripple.

REFERENCES

[1] Hal Abelson, Ross Anderson, Steven M Bellovin, Josh Benaloh, Matt Blaze, Jon Callas, Whitfield Diffie, Susan Landau, Peter G Neumann, Ronald L Rivest, et al. 2021. Bugs in our Pockets: The Risks of Client-Side Scanning. *arXiv preprint arXiv:2110.07450* (2021).

[2] Harold Abelson, Ross Anderson, Steven M Bellovin, Josh Benaloh, Matt Blaze, Whitfield Diffie, John Gilmore, Matthew Green, Susan Landau, Peter G Neumann, et al. 2015. Keys under doormats: mandating insecurity by requiring government access to all data and communications. *Journal of Cybersecurity* 1, 1 (2015), 69–79.

[3] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. 2018. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)* 51, 4 (2018), 1–35.

[4] Pushkal Agarwal, Aravindh Raman, Damiola Ibosiola, Nishanth Sastry, Gareth Tyson, and Kiran Garimella. 2022. Jettisoning Junk Messaging in the Era of End-to-End Encryption: A Case Study of WhatsApp. In *Proceedings of the ACM Web Conference 2022*. 2582–2591.

[5] Surabhi Agarwal. 2021. India proposes alpha-numeric hash to track WhatsApp chat. *The Economic Times* (23 3 2021). https://economictimes.indiatimes.com/tech/technology/govt-proposes-alpha-numeric-hash-to-track-whatsapp-chat/articleshow/81638939.cms

[6] Sohail Akhtar, Valerio Basile, and Viviana Patti. 2020. Modeling annotator perspective and polarized opinions to improve hate speech detection. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, Vol. 8. 151–154.

[7] Mhd Wesam Al Nabki, Eduardo Fidalgo, Enrique Alegre, and Rocío Alaíz-Rodríguez. 2020. File Name Classification Approach to Identify Child Sexual Abuse.. In *ICPRAM*. 228–234.

[8] Abdulatif Alabdulatif, Heshan Kumarage, Ibrahim Khalil, and Xun Yi. 2017. Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption. *J. Comput. System Sci.* 90 (2017), 28–45.

[9] Arwa Aldweesh, Abdelouahid Derhab, and Ahmed Z Emam. 2020. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems* 189 (2020), 105124.

[10] Sai Saketh Aluru, Binny Mathew, Punyajoy Saha, and Animesh Mukherjee. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465* (2020).

[11] An Open Letter 2021. An Open Letter Against Apple's Privacy-Invasive Content Scanning Technology. https://appleprivacyletter.com/

[12] Android Documentation. 2022. *File-based Encryption*. https://source.android.com/docs/security/encryption/file-based

[13] Apple. 2016. Answers to your questions about Apple and security. https://www.apple.com/customer-letter/answers/

[14] Apple. 2021. *iMessage Security Overview*. https://support.apple.com/guide/security/imessage-security-overview-secd9764312f/1/web/1

[15] Apple. 2021. Intro to FileVault. https://support.apple.com/guide/deployment/intro-to-filevault-dep82064ec40/web

[16] Apple. 2022. Advanced Data Protection for iCloud. https://support.apple.com/guide/security/advanced-data-protection-for-icloud-sec973254c5f/web

[17] Apple. 2022. *Apple Platform Security*. https://help.apple.com/pdf/security/en_US/apple-platform-security-guide.pdf

[18] Apple. 2022. Expanded Protections for Children. https://www.apple.com/child-safety/

[19] Apple. 2022. *FaceTime security*. https://support.apple.com/guide/security/facetime-security-seca331c55cd/web

[20] Apple, Inc. 2021. *CSAM Detection Technical Summary*. https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf

[21] Apple, Inc. 2021. *Security Threat Model Review of Apple's Child Safety Features*. https://www.apple.com/child-safety/pdf/Security_Threat_Model_Review_of_Apple_Child_Safety_Features.pdf

[22] Anish Athalye. 2021. *Inverting PhotoDNA*. https://www.anishathalye.com/2021/12/20/inverting-photodna/

[23] Man Ho Au and Apu Kapadia. 2012. PERM: practical reputation-based blacklisting without TTPS. In *ACM CCS 2012: 19th Conference on Computer and Communications Security*, Ting Yu, George Danezis, and Virgil D. Gligor (Eds.). ACM Press, Raleigh, NC, USA, 929–940. https://doi.org/10.1145/2382196.2382294

[24] Michael Backes, Jeremy Clark, Aniket Kate, Milivoj Simeonovski, and Peter Druschel. 2014. BackRef: Accountability in Anonymous Communication Networks. In *ACNS 14: 12th International Conference on Applied Cryptography and Network Security (Lecture Notes in Computer Science, Vol. 8479)*, Ioana Boureanu, Philippe Owesarski, and Serge Vaudenay (Eds.). Springer, Heidelberg, Germany, Lausanne, Switzerland, 380–400. https://doi.org/10.1007/978-3-319-07536-5_23

[25] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. 2008. Public key encryption with keyword search revisited. In *International conference on Computational Science and Its Applications*. Springer, 1249–1259.

[26] Abhishek Bagade, Ashwini Pale, Shreyans Sheth, Megha Agarwal, Soumen Chakrabarti, Kameswari Chebrolu, and S Sudarshan. 2020. The Kauwa-Kaate Fake News Detection System. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. 302–306.

[27] Chloé Bakalar, Renata Barreto, Miranda Bogen, Sam Corbett-Davies, Melissa Hall, Isabel Kloumann, Michelle Lam, Joaquin Qui nonero Candela, Manish Raghavan, Joshua Simons, Jonathan Tannen, Edmund Tong, Kate Vredenburgh, and Jiejing Zhao. 2021. *Fairness On The Ground: Applying Algorithmic Fairness Approaches To Production Systems*. Facebook. https://ai.facebook.com/research/publications/applying-algorithmic-fairness-approaches-to-production-systems

[28] Adam Barth, Dan Boneh, and Brent Waters. 2006. Privacy in encrypted content distribution using private broadcast encryption. In *International Conference on Financial Cryptography and Data Security*. Springer, 52–64.

[29] James Bartusek, Sanjam Garg, Abishek Jain, and Guru-Vamsi Policharla. 2022. End-to-End Secure Messaging with Traceability Only for Illegal Content. https://eprint.iacr.org/2022/1643.pdf

[30] Steven M Bellovin, Matt Blaze, Sandy Clark, and Susan Landau. 2012. Going bright: Wiretapping without weakening communications infrastructure. *IEEE Security & Privacy* 11, 1 (2012), 62–72.

[31] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. 2019. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. 351–371.

[32] Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Allison Woodruff, Christine Luu, Pierre Kreitmann, Jonathan Bischof, and Ed H. Chi. 2019. *Putting Fairness Principles into Practice: Challenges, Metrics, and Improvements*. Google. https://research.google/pubs/pub47763/

[33] Abhishek Bhowmick, Dan Boneh, Steve Myers, Kunal Talwar, and Karl Tarbe. 2021. *The Apple PSI System*. Technical Report. Apple, Inc. https://www.apple.com/child-safety/pdf/Apple_PSI_System_Security_Protocol_and_Analysis.pdf

[34] Song Bian, Masayuki Hiromoto, and Takashi Sato. 2019. Towards practical homomorphic email filtering: A hardware-accelerated secure Naïve Bayesian filter. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*. 621–626.

[35] Rubel Biswas and Pablo Blanco-Medina. 2021. State of the Art: Image Hashing. *arXiv preprint arXiv:2108.11794* (2021).

[36] Anis Bkakria, Nora Cuppens, and Frédéric Cuppens. 2020. Privacy-preserving pattern matching on encrypted data. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 191–220.

[37] Hannah Bloch-Wehba. 2021. Content Moderation as Surveillance. *Berkeley Technology Law Journal* 36 (2021), 21–37.

[38] Burton H. Bloom. 1970. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Commun. ACM* 13, 7 (1970), 422–426. https://doi.org/10.1145/362686.362692

[39] Dasha Bogdanova, Paolo Rosso, and Thamar Solorio. 2014. Exploring high-level features for detecting cyberpedophilia. *Computer speech & language* 28, 1 (2014), 108–120.

[40] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. 2004. Public key encryption with keyword search. In *International conference on the theory and applications of cryptographic techniques*. Springer, 506–522.

[41] Dan Boneh, Amit Sahai, and Brent Waters. 2011. Functional encryption: Definitions and challenges. In *Theory of Cryptography Conference*. Springer, 253–273.

[42] Dan Boneh and Victor Shoup. 2020. A graduate course in applied cryptography. *Draft 0.5* (2020).

[43] Nikita Borisov, Ian Goldberg, and Eric Brewer. 2004. Off-the-record communication, or, why not to use PGP. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. 77–84.

[44] Danielle Borrelli and Sherrie Caltagirone. 2020. Non-traditional cyber adversaries: Combatting human trafficking through data science. *Cyber Security: A Peer-Reviewed Journal* 4, 1 (2020), 77–90.

[45] Christoph Bösch, Pieter Hartel, Willem Jonker, and Andreas Peter. 2014. A survey of provably secure searchable encryption. *ACM Computing Surveys (CSUR)* 47, 2 (2014), 1–51.

[46] Raphael Bost. 2016. $\Sigma o\varphi o\varsigma$: Forward secure searchable encryption. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1143–1154.

[47] Patrick Bours and Halvor Kulsrud. 2019. Detection of cyber grooming in online conversation. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 1–6.

[48] Samantha Bradshaw and Laura DeNardis. 2018. The politicization of the Internet's Domain Name System: Implications for Internet security, universality, and freedom. *new media & society* 20, 1 (2018), 332–350.

[49] Thomas Brewster. 2021. Apple Delays iPhone Child Sexual Abuse Scanning After Uproar. *Forbes* (3 9 2021). https://www.forbes.com/sites/thomasbrewster/2021/09/03/apple-delays-iphone-child-sexual-abuse-scanning-after-uproar/?sh=4cfb0e4212d2

[50] Nina I Brown. 2020. Regulatory Goldilocks: Finding the Just and Right Fit for Content Moderation on Social Platforms. *Tex. A&M L. Rev.* 8 (2020), 451.

[51] Timothy Buck. 2022. *Express Yourself in Messenger's End-to-End Encrypted Chats.* https://messengernews.fb.com/2022/01/27/express-yourself-in-messengers-end-to-end-encrypted-chats/

[52] Elie Bursztein, Einat Clarke, Michelle DeLaune, David M Elifff, Nick Hsu, Lindsey Olson, John Shehan, Madhukar Thakur, Kurt Thomas, and Travis Bright. 2019. Rethinking the detection of child sexual abuse imagery on the Internet. In *The world wide web conference*. 2601–2607.

[53] Jon Callas, Lutz Donnerhacke, Hal Finney, and Rodney Thayer. 1998. *RFC 2440: OpenPGP message format.* Technical Report. Internet Engineering Task Force.

[54] Sébastien Canard, Aïda Diop, Nizar Kheir, Marie Paindavoine, and Mohamed Sabt. 2017. BlindIDS: Market-compliant and privacy-friendly intrusion detection system over encrypted traffic. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 561–574.

[55] Sébastien Canard and Chaoyun Li. 2021. Towards practical intrusion detection system over encrypted traffic. *IET Information Security* 15, 3 (2021), 231–246.

[56] Ran Canetti, Palak Jain, Marika Swanberg, and Mayank Varia. 2022. Universally Composable End-to-End Secure Messaging. *Cryptology ePrint Archive* (2022).

[57] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. 2015. Leakage-abuse attacks against searchable encryption. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 668–679.

[58] Shammah Chancellor, Harry Barber, and David Schlesinger. 2021. CashWeb. https://getstamp.io/whitepaper.pdf

[59] Dajiang Chen, Hao Wang, Ning Zhang, Xuyun Nie, Hong-Ning Dai, Kuan Zhang, and Kwang Raymond Choo. 2022. Privacy-Preserving Encrypted Traffic Inspection with Symmetric Cryptographic Techniques in IoT. *IEEE Internet of Things Journal* (2022).

[60] Guoxing Chen. 2019. *Exploitable Hardware Features and Vulnerabilities Enhanced Side-Channel Attacks on Intel SGX and Their Countermeasures.* The Ohio State University.

[61] Guoxing Chen, Sanchuan Chen, Yuan Xiao, Yinqian Zhang, Zhiqiang Lin, and Ten H Lai. 2018. Sgxpectre attacks: Leaking enclave secrets via speculative execution. *arXiv preprint arXiv:1802.09085* (2018).

[62] Long Chen and Qiang Tang. 2018. People Who Live in Glass Houses Should not Throw Stones: Targeted Opening Message Franking Schemes. Cryptology ePrint Archive, Report 2018/994. https://eprint.iacr.org/2018/994.

[63] Seung Geol Choi, Dana Dachman-Soled, Mukul Kulkarni, and Arkady Yerukhimovich. 2020. Differentially-Private Multi-Party Sketching for Large-Scale Statistics. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (July 2020), 153–174. https://doi.org/10.2478/popets-2020-0047

[64] Edward J Chou, Arun Gururajan, Kim Laine, Nitin Kumar Goel, Anna Bertiger, and Jack W Stokes. 2020. Privacy-preserving phishing web page classification via fully homomorphic encryption. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2792–2796.

[65] Alexandra Chouldechova. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data* 5, 2 (2017), 153–163. https://doi.org/10.1089/big.2016.0047

[66] Danielle Keats Citron. 2014. *Hate crimes in cyberspace.* Harvard University Press.

[67] Thomas Claburn. 2021. Apple didn't engage with the infosec world on CSAM scanning – so get used to a slow drip feed of revelations. *The Register* (18 8 2021). https://www.theregister.com/2021/08/18/apples_csam_hashing/

[68] MacKenzie F Common. 2020. Fear the reaper: How content moderation rules are enforced on social media. *International Review of Law, Computers & Technology* 34, 2 (2020), 126–152.

[69] Julie Cordua. 2021. Apple's expanded child protections and the future of digital child protection. *Thorn* (5 8 2021).

[70] Gareth Corfield. 2021. Apple's bright idea for CSAM scanning could start 'persecution on a global basis' – 90+ civil rights groups. *The Register* (19 8 2021). https://www.theregister.com/2021/08/19/apple_csam_condemned/

[71] Saúl Ibarra Corretgé and Emil Ivov. 2021. *End-to-End Encryption in Jitsi Meet.* https://jitsi.org/wp-content/uploads/2021/08/jitsi-e2ee-1.0.pdf

[72] Henry Corrigan-Gibbs and Dan Boneh. 2017. Prio: Private, robust, and scalable computation of aggregate statistics. In *14th USENIX symposium on networked systems design and implementation (NSDI 17)*. 259–282.

[73] Henry Corrigan-Gibbs and Bryan Ford. 2010. Dissent: accountable anonymous group messaging. In *Proceedings of the 17th ACM conference on Computer and communications security*. 340–350.

[74] Henry Corrigan-Gibbs, David Isaac Wolinsky, and Bryan Ford. 2013. Proactively accountable anonymous messaging in verdict. In *22nd USENIX Security Symposium (USENIX Security 13)*. 147–162.

[75] Victor Costan and Srinivas Devadas. 2016. Intel SGX explained. *Cryptology ePrint Archive* (2016).

[76] Gianpiero Costantino, Antonio La Marra, Fabio Martinelli, Andrea Saracino, and Mina Sheikhalishahi. 2017. Privacy-preserving text mining as a service. In *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 890–897.

[77] Moshé Cotacallapa, Lilian Berton, Leonardo N Ferreira, Marcos G Quiles, Liang Zhao, Elbert EN Macau, and Didier A Vega-Oliveros. 2020. Measuring the engagement level in encrypted group conversations by using temporal networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[78] Michael Coughlin, Eric Keller, and Eric Wustrow. 2017. Trusted click: Overcoming security issues of NFV in the cloud. In *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. 31–36.

[79] Kate Crawford and Tarleton Gillespie. 2016. What is a flag for? Social media reporting tools and the vocabulary of complaint. *New Media & Society* 18, 3 (2016), 410–428. https://doi.org/10.1177/1461444814543163 arXiv:https://doi.org/10.1177/1461444814543163

[80] Helei Cui, Yajin Zhou, Cong Wang, Qi Li, and Kui Ren. 2018. Towards privacy-preserving malware detection systems for android. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 545–552.

[81] Helei Cui, Yajin Zhou, Cong Wang, Xinyu Wang, Yuefeng Du, and Qian Wang. 2019. PPSB: An open and flexible platform for privacy-preserving safe browsing. *IEEE Transactions on Dependable and Secure Computing* 18, 4 (2019), 1762–1778.

[82] Reza Curtmola, Juan Garay, Seny Kamara, and Rafail Ostrovsky. 2006. Searchable symmetric encryption: improved definitions and efficient constructions. In *Proceedings of the 13th ACM conference on Computer and communications security*. 79–88.

[83] Cyacomb. 2021. Cyan Protect. https://cyanforensics.com/wp-content/uploads/2021/03/CyanProtect-Leaflet.pdf

[84] Janis Dalins, Campbell Wilson, and Douglas Boudry. 2019. PDQ & TMK+ PDQF–A Test Drive of Facebook's Perceptual Hashing Algorithms. *arXiv preprint arXiv:1912.07745* (2019).

[85] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. 2012. Multiparty computation from somewhat homomorphic encryption. In *Annual Cryptology Conference*. Springer, 643–662.

[86] Ernesto Damiani, S De Capitani Di Vimercati, Stefano Paraboschi, and Pierangela Samarati. 2004. P2P-based collaborative spam detection and filtering. In *Proceedings. Fourth International Conference on Peer-to-Peer Computing, 2004. Proceedings.* IEEE, 176–183.

[87] George Danezis, Roger Dingledine, and Nick Mathewson. 2003. Mixminion: Design of a type III anonymous remailer protocol. In *2003 Symposium on Security and Privacy, 2003.* IEEE, 2–15.

[88] Vanesa Daza, Abida Haque, Alessandra Scafuro, Alexandros Zacharakis, and Arantxa Zapico. 2021. Mutual Accountability Layer: Accountable Anonymity within Accountable Trust. Cryptology ePrint Archive, Report 2021/596. https://eprint.iacr.org/2021/596.

[89] Xavier de Carné de Carnavalet and Paul C van Oorschot. 2020. A survey and analysis of TLS interception mechanisms and motivations. *arXiv preprint arXiv:2010.16388* (2020).

[90] Giovanni De Gregorio. 2020. Democratising online content moderation: A constitutional framework. *Computer Law & Security Review* 36 (2020), 105374.

[91] Daniel Demmler, Thomas Schneider, and Michael Zohner. 2015. ABY-A framework for efficient mixed-protocol secure two-party computation.. In *NDSS*.

[92] Department of Justice, United States. 2020. International Statement: End-To-End Encryption and Public Safety. https://www.justice.gov/opa/pr/international-statement-end-end-encryption-and-public-safety

[93] Thomas Deselaers, Lexi Pimenidis, and Hermann Ney. 2008. Bag-of-visual-words models for adult image classification and filtering. In *2008 19th International Conference on Pattern Recognition*. IEEE, 1–4.

[94] Nicolas Desmoulins, Pierre-Alain Fouque, Cristina Onete, and Olivier Sanders. 2018. Pattern matching on encrypted streams. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 121–148.

[95] Dimitris Deyannis, Eva Papadogiannaki, Giorgos Kalivianakis, Giorgos Vasiliadis, and Sotiris Ioannidis. 2020. Trustav: Practical and privacy preserving malware analysis in the cloud. In *Proceedings of the tenth ACM conference on data and application security and privacy*. 39–48.

[96] Thiago Dias Oliva. 2020. Content moderation technologies: Applying human rights standards to protect freedom of expression. *Human Rights Law Review* 20, 4 (2020), 607–640.

[97] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. *Tor: The second-generation onion router.* Technical Report. Naval Research Lab Washington

DC.

[98] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. 2018. Fast Message Franking: From Invisible Salamanders to Encryptment. In *Advances in Cryptology – CRYPTO 2018, Part I (Lecture Notes in Computer Science, Vol. 10991)*, Hovav Shacham and Alexandra Boldyreva (Eds.). Springer, Heidelberg, Germany, Santa Barbara, CA, USA, 155–186. https://doi.org/10.1007/978-3-319-96884-1_6

[99] Brian Dolhansky and Cristian Canton Ferrer. 2020. Adversarial collision attacks on image hashing functions. *arXiv preprint arXiv:2011.09473* (2020).

[100] Evelyn Douek. 2021. More Content Moderation Is Not Always Better. *Wired* (21 6 2021). https://www.wired.com/story/more-content-moderation-not-always-better/

[101] Evelyn Douek. 2022. The Siren Call of Content Moderation Formalism. *NEW TECHNOLOGIES OF COMMUNICATION AND THE FIRST AMENDMENT: THE INTERNET, SOCIAL MEDIA AND CENSORSHIP (Lee Bollinger & Geoffrey Stone eds., Forthcoming 2022)* (2022).

[102] Andrea Drmic, Marin Silic, Goran Delac, Klemo Vladimir, and Adrian S Kurdija. 2017. Evaluating robustness of perceptual image hashing algorithms. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 995–1000.

[103] Ling Du, Anthony TS Ho, and Runmin Cong. 2020. Perceptual hashing for image authentication: A survey. *Signal Processing: Image Communication* 81 (2020), 115713.

[104] Huayi Duan, Cong Wang, Xingliang Yuan, Yajin Zhou, Qian Wang, and Kui Ren. 2019. LightBox: Full-stack Protected Stateful Middlebox at Lightning Speed. In *ACM CCS 2019: 26th Conference on Computer and Communications Security*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, 2351–2367. https://doi.org/10.1145/3319535.3339814

[105] David Evans, Vladimir Kolesnikov, Mike Rosulek, et al. 2018. A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security* 2, 2-3 (2018), 70–246.

[106] Facebook. 2022. Oversight Board. https://oversightboard.com

[107] Jingyuan Fan, Chaowen Guan, Kui Ren, Yong Cui, and Chunming Qiao. 2017. Spabox: Safeguarding privacy during deep packet inspection at a middlebox. *IEEE/ACM Transactions on Networking* 25, 6 (2017), 3753–3766.

[108] Zhongqi Fan, Yong Zeng, Xiaoyan zhu, and Jianfeng Ma. 2020. A group key agreement based encrypted traffic detection scheme for Internet of Things. In *Proceedings of the 1st ACM International Workshop on Security and Safety for Intelligent Cyber-Physical Systems*. 19–26.

[109] Hany Farid. 2021. An Overview of Perceptual Hashing. *Journal of Online Trust and Safety* 1, 1 (2021).

[110] Tom Farrell and Matt Burns. 2022. Understanding real-time threat detection and the SafeToWatch mission. https://www.cameraforensics.com/blog/2022/01/12/understanding-real-time-threat-detection-and-the-safetowatch-mission/

[111] Ellis Fenske, Akshaya Mani, Aaron Johnson, and Micah Sherr. 2017. Distributed Measurement with Private Set-Union Cardinality. In *ACM CCS 2017: 24th Conference on Computer and Communications Security*, Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu (Eds.). ACM Press, Dallas, TX, USA, 2295–2312. https://doi.org/10.1145/3133956.3134034

[112] Carnegie Endowment for International Peace Encryption Working Group. 2019. Moving the Encryption Policy Conversation Forward. https://carnegieendowment.org/2019/09/10/moving-encryption-policy-conversation-forward-pub-79573

[113] National Center for Missing and Exploited Children. 2019. *End-to-end encryption: ignoring abuse won't stop it.* https://www.missingkids.org/blog/2019/post-update/end-to-end-encryption

[114] National Center for Missing and Exploited Children. 2022. The Issues. https://www.missingkids.org/theissues

[115] Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 1–30.

[116] Jonathan Frankle, Sunoo Park, Daniel Shaar, Shafi Goldwasser, and Daniel Weitzner. 2018. Practical accountability of secret processes. In *27th USENIX Security Symposium (USENIX Security 18)*. 657–674.

[117] Sharon Bradford Franklin and Greg Nojeim. 2021. International Coalition Calls on Apple to Abandon Plan to Build Surveillance Capabilities into iPhones, iPads, and Other Products. https://cdt.org/insights/international-coalition-calls-on-apple-to-abandon-plan-to-build-surveillance-capabilities-into-iphones-ipads-and-other-products/

[118] Philipe de Freitas Melo, Carolina Coimbra Vieira, Kiran Garimella, Pedro OS Melo, and Fabrício Benevenuto. 2019. Can WhatsApp counter misinformation by limiting message forwarding?. In *International conference on complex networks and their applications*. Springer, 372–384.

[119] Clemens Fruhwirth. 2018. LUKS1 On-Disk Format Specification Version 1.2.3. https://gitlab.com/cryptsetup/cryptsetup/-/wikis/LUKS-standard/on-disk-format.pdf

[120] UK Safety Tech Challenge Fund. 2022. End of Programme Supplier Showcase. https://www.eventbrite.co.uk/e/safety-tech-challenge-fund-end-of-programme-supplier-showcase-tickets-302675218727 Personal attendance.

[121] Galaxkey. 2021. Galaxkey wins UK Government funding to enable children to communicate safely online. https://www.galaxkey.com/blog/galaxkey-wins-uk-government-funding-to-enable-children-to-communicate-safely-online/

[122] Rafa Gálvez, Veelasha Moonsamy, and Claudia Diaz. 2020. Less is More: A privacy-respecting Android malware classifier using federated learning. *arXiv preprint arXiv:2007.08319* (2020).

[123] Bharath Ganesh and Jonathan Bright. 2020. Countering extremists on social media: Challenges for strategic communication and content moderation. , 6–19 pages.

[124] Manuel B Garcia, Teodoro F Revano, Beau Gray M Habal, Jennifer O Contreras, and John Benedic R Enriquez. 2018. A pornographic image and video filtering application using optimized nudity recognition and detection algorithm. In *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE, 1–5.

[125] Gayathri Garimella, Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. 2021. Oblivious key-value stores and amplification for private set intersection. In *Annual International Cryptology Conference*. Springer, 395–425.

[126] R. Stuart Geiger. 2016. Bot-based collective blocklists in Twitter: the counterpublic moderation of harassment in a networked public space. *Information, Communication & Society* 19, 6 (2016), 787–803. https://doi.org/10.1080/1369118X.2016.1153700 arXiv:https://doi.org/10.1080/1369118X.2016.1153700

[127] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 169–178.

[128] SB GHATTE and AB RAJMANE. 2017. Study Of Preserving Privacy In Mobile Social Network By Personalization Of Fine Grained Spam Filtering Scheme. *JournalNX* 3, 09 (2017), 61–63.

[129] Tarleton Gillespie. 2018. *Custodians of the Internet*. Yale University Press.

[130] Tarleton Gillespie. 2020. Content moderation, AI, and the question of scale. *Big Data & Society* 7, 2 (2020), 2053951720943234.

[131] Tarleton Gillespie, Patricia Aufderheide, Elinor Carmi, Ysabel Gerrard, Robert Gorwa, Ariadna Matamoros-Fernández, Sarah T Roberts, Aram Sinnreich, and Sarah Myers West. 2020. Expanding the debate about content moderation: Scholarly research agendas for the coming policy debates. *Internet Policy Review* 9, 4 (2020), Article–number.

[132] Daniel Kahn Gillmor. 2021. Apple's New 'Child Safety' Plan for iPhones Isn't So Safe. *American Civil Liberties Union* (24 8 2021). https://www.aclu.org/news/privacy-technology/apples-new-child-safety-plan-for-iphones-isnt-so-safe/

[133] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity Search in High Dimensions via Hashing. In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases, September 7-10, 1999, Edinburgh, Scotland, UK*, Malcolm P. Atkinson, Maria E. Orlowska, Patrick Valduriez, Stanley B. Zdonik, and Michael L. Brodie (Eds.). Morgan Kaufmann, 518–529.

[134] Global Internet Forum 2022. Global Internet Forum to Counter Terrorism. https://gifct.org/

[135] Eric Goldman. 2021. Content moderation remedies. *Michigan Technology Law Review (Forthcoming)* (2021).

[136] Oded Goldreich, Silvio Micali, and Avi Wigderson. 2019. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. 307–328.

[137] David Goltzsche, Signe Rüsch, Manuel Nieke, Sébastien Vaucher, Nico Weichbrodt, Valerio Schiavoni, Pierre-Louis Aublin, Paolo Cosa, Christof Fetzer, Pascal Felber, et al. 2018. Endbox: Scalable middlebox functions using client-side trusted execution. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 386–397.

[138] Google. 2019. Your chats stay private with spam detection: How we protect your data. https://support.google.com/messages/answer/9327903#zippy=%2Chow-we-protect-your-data

[139] Google. 2021. NCMEC, Google and Image Hashing Technology. https://safety.google/intl/en_US/stories/hash-matching-to-help-ncmec/

[140] Google. 2022. Google Transparency Report. https://transparencyreport.google.com/

[141] Google. 2022. How Hash-Based Safe Browsing Works in Google Chrome. *Google Security Blog* (8 8 2022). https://security.googleblog.com/2022/08/how-hash-based-safe-browsing-works-in.html

[142] Google. 2022. *Messages End-to-End Encryption Overview*. https://www.gstatic.com/messages/papers/messages_e2ee.pdf

[143] Robert Gorwa, Reuben Binns, and Christian Katzenbach. 2020. Algorithmic content moderation: Technical and political challenges in the automation of platform governance. *Big Data & Society* 7, 1 (2020), 2053951719897945.

[144] Johannes Götzfried, Moritz Eckert, Sebastian Schinzel, and Tilo Müller. 2017. Cache attacks on Intel SGX. In *Proceedings of the 10th European Workshop on Systems Security*. 1–6.

[145] Matthew D. Green and Alex Stamos. 2021. Apple Wants to Protect Children. But It's Creating Serious Privacy Risks. *New York Times* (11 8 2021). https://www.nytimes.com/2021/08/11/opinion/apple-iphones-privacy.html

[146] Andy Greenberg. 2021. Apple Walks a Privacy Tightrope to Spot Child Abuse in iCloud. *Wired* (5 8 2021).

[147] James Grimmelmann. 2015. The virtues of moderation. *Yale JL & Tech.* 17 (2015), 42.

[148] James Grimmelmann. 2017. The platform is the message. *Geo. L. Tech. Rev.* 2 (2017), 217.

[149] Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N Asokan. 2018. All you need is" love" evading hate speech detection. In *Proceedings of the 11th ACM workshop on artificial intelligence and security*. 2–12.

[150] Paul Grubbs, Arasu Arun, Ye Zhang, Joseph Bonneau, and Michael Walfish. 2022. Zero-Knowledge Middleboxes. In *31st USENIX Security Symposium (USENIX Security 22)*. 4255–4272.

[151] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. 2017. Message Franking via Committing Authenticated Encryption. In *Advances in Cryptology – CRYPTO 2017, Part III (Lecture Notes in Computer Science, Vol. 10403)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, Heidelberg, Germany, Santa Barbara, CA, USA, 66–97. https://doi.org/10.1007/978-3-319-63697-9_3

[152] Enrique Guerra and Bryce G Westlake. 2021. Detecting child sexual abuse images: traits of child sexual exploitation hosting and displaying websites. *Child Abuse & Neglect* 122 (2021), 105336.

[153] Yu Guo, Cong Wang, Xingliang Yuan, and Xiaohua Jia. 2018. Enabling privacy-preserving header matching for outsourced middleboxes. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[154] Yu Guo, Mingyue Wang, Cong Wang, Xingliang Yuan, and Xiaohua Jia. 2020. Privacy-preserving packet header checking over in-the-cloud middleboxes. *IEEE Internet of Things Journal* 7, 6 (2020), 5359–5370.

[155] Trinabh Gupta, Henrique Fingler, Lorenzo Alvisi, and Michael Walfish. 2017. Pretzel: Email encryption and provider-supplied functions are compatible. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 169–182.

[156] Alon Halevy, Cristian Canton-Ferrer, Hao Ma, Umut Ozertem, Patrick Pantel, Marzieh Saeidi, Fabrizio Silvestri, and Ves Stoyanov. 2022. Preserving integrity in online social networks. *Commun. ACM* 65, 2 (2022), 92–98.

[157] Juhyeng Han, Seongmin Kim, Daeyang Cho, Byungkwon Choi, Jaehyeong Ha, and Dongsu Han. 2020. A secure middlebox framework for enabling visibility over multiple encryption protocols. *IEEE/ACM Transactions on Networking* 28, 6 (2020), 2727–2740.

[158] Juhyeng Han, Seongmin Kim, Jaehyeong Ha, and Dongsu Han. 2017. SGX-box: Enabling visibility on encrypted traffic using a secure middlebox module. In *Proceedings of the First Asia-Pacific Workshop on Networking*. 99–105.

[159] Qingying Hao, Licheng Luo, Steve T. K. Jan, and Gang Wang. 2021. It's Not What It Looks Like: Manipulating Perceptual Hashing based Applications. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi (Eds.). ACM, 69–85. https://doi.org/10.1145/3460120.3484559

[160] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604* (2018).

[161] Ryan Henry and Ian Goldberg. 2011. Formalizing Anonymous Blacklisting Systems. In *2011 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, Berkeley, CA, USA, 81–95. https://doi.org/10.1109/SP.2011.13

[162] Sophia Hicks. 2021. DragonflAI. https://www.safetytechnetwork.org.uk/provider/dargonflai/ [sic].

[163] Sofia Hicks. 2021. Government announces the winners of the Safety Tech Challenge Fund. *Safety Tech Innovation Network* (17 11 2021). https://www.safetytechnetwork.org.uk/government-announces-the-winners-of-the-safety-tech-challenge-fund/

[164] Shoichi Hirose. 2020. Compactly Committing Authenticated Encryption Using Tweakable Block Cipher. In *International Conference on Network and System Security*. Springer, 187–206.

[165] Sui Lyn Hor, Hezerul Abdul Karim, Mohd Haris Lye Abdullah, Nouar AlDahoul, Sarina Mansor, Mohammad Faizal Ahmad Fauzi, John See, and Abdulaziz Saleh Ba Wazir. 2021. An Evaluation of State-of-the-Art Object Detectors for Pornography Detection. In *2021 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*. IEEE, 191–196.

[166] Sharon Huang, Subodh Iyengar, Sundar Jeyaraman, Shiv Kushwah, Chen-Kuei Lee, Zutian Luo, Payman Mohassel, Ananth Raghunathan, Shaahid Shaikh, Yen-Chieh Sung, and Albert Zhang. 2021. *DIT: De-Identified Authenticated Telemetry at Scale.* https://scontent-lga3-2.xx.fbcdn.net/v/t39.8562-6/246534149_588854725718321_8923613326138589821_n.pdf

[167] Loïs Huguenin-Dumittan and Iraklis Leontiadis. 2018. A Message Franking Channel. Cryptology ePrint Archive, Report 2018/920. https://eprint.iacr.org/2018/920.

[168] Kyle Hunt, Puneet Agarwal, and Jun Zhuang. 2022. Monitoring misinformation on Twitter during crisis events: a machine learning approach. *Risk analysis* 42, 8 (2022), 1728–1748.

[169] Taeyoon Hwang and Ji Won Yoon. 2019. Static Malware Analysis in Encrypted Domain. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 47–52.

[170] Internet Watch Foundation. 2022. Image Hash List. https://www.iwf.org.uk/our-technology/our-services/image-hash-list/

[171] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. 2012. Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation. In *ISOC Network and Distributed System Security Symposium – NDSS 2012*. The Internet Society, San Diego, CA, USA.

[172] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. 2012. Access pattern disclosure on searchable encryption: ramification, attack and mitigation.. In *Ndss*, Vol. 20. Citeseer, 12.

[173] Rawane Issa, Nicolas Alhaddad, and Mayank Varia. 2022. Hecate: Abuse reporting in secure messengers with sealed sender. , 2335–2352 pages.

[174] Ryan Jackson. 2022. T3K.AI. https://www.safetytechnetwork.org.uk/provider/t3k-ai/

[175] Shubham Jain, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. 2022. Adversarial Detection Avoidance Attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In *31st USENIX Security Symposium (USENIX Security 22)*.

[176] Markus Jakobsson, John Linn, and Joy Algesheimer. 2003. How to Protect Against a Militant Spammer. Cryptology ePrint Archive, Report 2003/071. https://eprint.iacr.org/2003/071.

[177] Ravi Chandra Jammalamadaka, Sharad Mehrotra, and Nalini Venkatasubramanian. 2005. Pvault: a client server system providing mobile access to personal data. In *Proceedings of the 2005 ACM workshop on Storage security and survivability*. 123–129.

[178] Rob Jansen and Aaron Johnson. 2016. Safely Measuring Tor. In *ACM CCS 2016: 23rd Conference on Computer and Communications Security*, Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi (Eds.). ACM Press, Vienna, Austria, 1553–1567. https://doi.org/10.1145/2976749.2978310

[179] Jeff Jarmoc and DSCT Unit. 2012. SSL/TLS interception proxies and transitive trust. *Black Hat Europe* (2012).

[180] Shagun Jhaver, Amy Bruckman, and Eric Gilbert. 2019. Does transparency in moderation really matter? User behavior after content removal explanations on reddit. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–27.

[181] Shagun Jhaver, Sucheta Ghoshal, Amy S. Bruckman, and Eric Gilbert. 2018. Online Harassment and Content Moderation: The Case of Blocklists. *ACM Trans. Comput. Hum. Interact.* 25, 2 (2018), 12:1–12:33. https://doi.org/10.1145/3185593

[182] Xi Jia and Meng Zhang. 2022. Encrypted Packet Inspection Based on Oblivious Transfer. *Security and Communication Networks* 2022 (2022).

[183] Peng Jiang, Baoqi Qiu, and Liehuang Zhu. 2022. Report When Malicious: Deniable and Accountable Searchable Message-Moderation System. *IEEE Transactions on Information Forensics and Security* 17 (2022), 1597–1609.

[184] Feng Jiao, Wen Gao, Lijuan Duan, and Guoqin Cui. 2001. Detecting adult image using multiple features. In *2001 International Conferences on Info-Tech and Info-Net. Proceedings (Cat. No. 01EX479)*, Vol. 3. IEEE, 378–383.

[185] Haojian Jin, Gram Liu, David Hwang, Swarun Kumar, Yuvraj Agarwal, and Jason I Hong. 2022. Peekaboo: A Hub-Based Approach to Enable Transparency in Data Processing within Smart Homes. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 303–320.

[186] Seny Kamara, Abdelkarim Kati, Tarik Moataz, Thomas Schneider, Amos Treiber, and Michael Yonli. 2022. SoK: Cryptanalysis of Encrypted Search with LEAKER–A framework for LEakage AttacK Evaluation on Real-world data. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 90–108.

[187] Seny Kamara, Mallory Knodel, Emma Llansó, Greg Nojeim, Lucy Qin, Dhanaraj Thakur, and Caitlin Vogus. 2021. Outside looking in: Approaches to content moderation in end-to-end encrypted systems. *Center for Democracy and Technology* (2021). https://cdt.org/wp-content/uploads/2021/08/CDT-Outside-Looking-In-Approaches-to-Content-Moderation-in-End-to-End-Encrypted-Systems.pdf

[188] Daniel Kardefelt-Winther, Emma Day, Gabrielle Berman, Sabine K Witting, Anjan Bose, et al. 2020. *Encryption, Privacy and Children's Right to Protection from Harm.* UNICEF Office of Research-Innocenti.

[189] Jonathan Katz and Yehuda Lindell. 2020. *Introduction to Modern Cryptography: Third Edition.* CRC press.

[190] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen, and T. Kivinen. 2014. RFC 7296: Internet Key Exchange Protocol Version 2 (IKEv2).

[191] David A Kaye. 2019. *Speech police: The global struggle to govern the Internet.* Columbia Global Reports.

[192] Ashkan Kazemi, Kiran Garimella, Gautam Kishore Shahi, Devin Gaffney, and Scott A Hale. 2021. Tiplines to combat misinformation on encrypted platforms: a case study of the 2019 Indian election on WhatsApp. *arXiv preprint arXiv:2106.04726* (2021).

[193] Georgios Kellaris, George Kollios, Kobbi Nissim, and Adam O'neill. 2016. Generic attacks on secure outsourced databases. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1329–1340.

[194] Wayne Kelly, Andrew Donnellan, and Derek Molloy. 2008. Screening for objectionable images: A review of skin detection techniques. In *2008 International Machine Vision and Image Processing Conference*. IEEE, 151–158.

[195] Gail Kent. 2018. *Why does WhatsApp Enable End-to-End Encryption?* https://about.fb.com/news/2018/05/end-to-end-encryption/

[196] Kinan Keshkeh, Aman Jantan, Kamal Alieyan, and Usman Mohammed Gana. 2021. A Review on TLS Encryption Malware Detection: TLS Features, Machine Learning Usage, and Future Directions. In *International Conference on Advances in Cyber Security*. Springer, 213–229.

[197] Keybase. 2022. Security on Keybase. https://book.keybase.io/security

[198] Alhassan Khedr, Glenn Gulak, and Vinod Vaikuntanathan. 2015. SHIELD: scalable homomorphic implementation of encrypted data-classifiers. *IEEE Trans. Comput.* 65, 9 (2015), 2848–2858.

[199] Jongkil Kim, Seyit Camtepe, Joonsang Baek, Willy Susilo, Josef Pieprzyk, and Surya Nepal. 2021. P2DPI: Practical and Privacy-Preserving Deep Packet Inspection. In *ASIACCS 21: 16th ACM Symposium on Information, Computer and Communications Security*, Jiannong Cao, Man Ho Au, Zhiqiang Lin, and Moti Yung (Eds.). ACM Press, Virtual Event, Hong Kong, 135–146. https://doi.org/10.1145/3433210.3437525

[200] So-Yeon Kim, Sun-Woo Yun, Eun-Young Lee, So-Hyeon Bae, and Il-Gu Lee. 2020. Fast packet inspection for end-to-end encryption. *Electronics* 9, 11 (2020), 1937.

[201] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. 2017. Inherent Trade-Offs in the Fair Determination of Risk Scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA (LIPIcs, Vol. 67)*, Christos H. Papadimitriou (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 43:1–43:23. https://doi.org/10.4230/LIPIcs.ITCS.2017.43

[202] Evan Klinger and David Starkweather. 2009. pHash: The open source perceptual hash library. https://phash.org/

[203] Kate Klonick. 2017. The new governors: The people, rules, and processes governing online speech. *Harv. L. Rev.* 131 (2017), 1598.

[204] Kate Klonick. 2020. Content moderation modulation. *Commun. ACM* 64, 1 (2020), 29–31.

[205] Dmitry Kogan and Henry Corrigan-Gibbs. 2021. Private blocklist lookups with checklist. In *30th USENIX Security Symposium (USENIX Security 21)*. 875–892.

[206] Peter Kollock and Marc Smith. 1996. Managing the virtual commons. *Computer-mediated communication: Linguistic, social, and cross-cultural perspectives* (1996), 109–128.

[207] Joseph S Kong, P Oscar Boykin, Behnam Attaran Rezaei, Nima Sarshar, and Vwani P Roychowdhury. 2005. Scalable and Reliable Collaborative Spam Filters: Harnessing the Global Social Email Networks.. In *CEAS*. Citeseer.

[208] Stefan Köpsell, Rolf Wendolsky, and Hannes Federrath. 2006. Revocable Anonymity. In *Emerging Trends in Information and Communication Security, International Conference, ETRICS 2006, Freiburg, Germany, June 6-9, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 3995)*, Günter Müller (Ed.). Springer, 206–220. https://doi.org/10.1007/11766155_15

[209] Neil Krawetz. 2021. PhotoDNA and Limitations. https://www.hackerfactor.com/blog/index.php?/archives/931-PhotoDNA-and-Limitations.html

[210] Max Krohn. 2020. *Zoom rolling out end-to-end encryption offering.* https://blog.zoom.us/zoom-rolling-out-end-to-end-encryption-offering/

[211] Yunus Kucuk, Nikhil Patil, Zhan Shu, and Guanhua Yan. 2018. BigBing: privacy-preserving cloud-based malware classification service. In *2018 IEEE Symposium on Privacy-Aware Computing (PAC)*. IEEE, 43–54.

[212] Anunay Kulshrestha and Jonathan Mayer. 2021. Identifying Harmful Media in End-to-End Encrypted Communication: Efficient Private Membership Computation. In *30th USENIX Security Symposium (USENIX Security 21)*. 893–910.

[213] Dmitrii Kuvaiskii, Somnath Chakrabarti, and Mona Vij. 2018. Snort intrusion detection system with intel software guard extension (intel sgx). *arXiv preprint arXiv:1802.00508* (2018).

[214] Abdelkader Lahmadi, Laurent Delosieres, and Olivier Festor. 2011. Hinky: Defending against text-based message spam on smartphones. In *2011 IEEE International Conference on Communications (ICC)*. IEEE, 1–5.

[215] Shangqi Lai, Xingliang Yuan, Shi-Feng Sun, Joseph K Liu, Ron Steinfeld, Amin Sakzad, and Dongxi Liu. 2021. Practical encrypted network traffic pattern matching for secure middleboxes. *IEEE Transactions on Dependable and Secure Computing* 19, 4 (2021), 2609–2621.

[216] Chang Lan, Justine Sherry, Raluca Ada Popa, Sylvia Ratnasamy, and Zhi Liu. 2016. Embark: Securely outsourcing middleboxes to the cloud. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. 255–273.

[217] Susan Landau. 2011. Going Dark: Lawful Electronic Surveillance in the Face of New Technologies. *House Judiciary Committee* (17 2 2011).

[218] Kyle Langvardt. 2017. Regulating online content moderation. *Geo. LJ* 106 (2017), 1353.

[219] James Larisch, David R. Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. 2017. CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers. In *2017 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Jose, CA, USA, 539–556. https://doi.org/10.1109/SP.2017.17

[220] Hee-Eun Lee, Tatiana Ermakova, Vasilis Ververis, and Benjamin Fabian. 2020. Detecting child sexual abuse material: A comprehensive survey. *Forensic Science International: Digital Investigation* 34 (2020), 301022.

[221] Sang-Woong Lee, Mokhtar Mohammadi, Shima Rashidi, Amir Masoud Rahmani, Mohammad Masdari, Mehdi Hosseinzadeh, et al. 2021. Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. *Journal of Network and Computer Applications* 187 (2021), 103111.

[222] Iraklis Leontiadis and Serge Vaudenay. 2018. Private Message Franking with After Opening Privacy. Cryptology ePrint Archive, Report 2018/938. https://eprint.iacr.org/2018/938.

[223] Ian Levy and Crispin Robinson. 2022. Thoughts on child safety on commodity platforms. *arXiv preprint arXiv:2207.09506* (2022).

[224] Ian Levy and Crispin Robinson. 2022. Thoughts on child safety on commodity platforms. *arXiv preprint arXiv:2207.09506* (2022).

[225] Chunxiao Li, Yu Guo, and Xia Wang. 2022. Towards privacy-preserving dynamic deep packet inspection over outsourced middleboxes. *High-Confidence Computing* 2, 1 (2022), 100033.

[226] Jie Li, Jinshu Su, Xiaofeng Wang, Hao Sun, and Shuhui Chen. 2017. CloudDPI: cloud-based privacy-preserving deep packet inspection via reversible sketch. In *International Symposium on Cyberspace Safety and Security*. Springer, 119–134.

[227] Konstantinos Limniotis. 2021. Cryptography as the Means to Protect Fundamental Human Rights. *Cryptography* 5, 4 (2021), 34.

[228] Yi-Hui Lin, Shan-Hsiang Shen, Ming-Hong Yang, De-Nian Yang, and Wen-Tsuen Chen. 2016. Privacy-preserving deep packet filtering over encrypted traffic in software-defined networks. In *2016 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.

[229] Yehuda Lindell. 2018. The security of intel SGX for key protection and data privacy applications. (2018).

[230] LINE. 2022. About Letter Sealing. https://help.line.me/line/?contentId=50000087

[231] Linsheng Liu, Daniel S Roche, Austin Theriault, and Arkady Yerukhimovich. 2021. Fighting Fake News in Encrypted Messaging with the Fuzzy Anonymous Complaint Tally System (FACTS). *arXiv preprint arXiv:2109.04559* (2021).

[232] Thomas Llanos. 2020. *Current Approaches to Terrorist and Violent Extremist Content among the Global Top 50 Online Content-Sharing Services*. Technical Report. Organization for Economic Co-Operation and Development. https://doi.org/10.1787/68058b95-en

[233] Joshua Lund. 2018. *Technology preview: Sealed sender for Signal*. Technical Report. Signal. https://signal.org/blog/sealed-sender/

[234] Nikolaos Lykousas and Constantinos Patsakis. 2021. Large-scale analysis of grooming in modern social networks. *Expert Systems with Applications* 176 (2021), 114808.

[235] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS one* 14, 8 (2019), e0221152.

[236] Shervin Malmasi and Marcos Zampieri. 2018. Challenges in discriminating profanity from hate speech. *Journal of Experimental & Theoretical Artificial Intelligence* 30, 2 (2018), 187–202. https://doi.org/10.1080/0952813X.2017.1409284 arXiv:https://doi.org/10.1080/0952813X.2017.1409284

[237] Jorge Araújo Martins Filho and Li-Chang Shuen. 2021. Fact-checking in a crumbling democracy: Implementation of "Sem Migué" platform during local elections in São Luís. *Revista Observatório* 7, 3 (2021), a5en–a5en.

[238] J Nathan Matias. 2019. The civic labor of volunteer moderators online. *Social Media+ Society* 5, 2 (2019), 2056305119836778.

[239] Jonathan Mayer. 2019. Content moderation for end-to-end encrypted messaging.

[240] Jonathan Mayer and Anunay Kulshrestha. 2021. We built a system like Apple's to flag child sexual abuse material — and concluded the tech was dangerous. *Washington Post* (19 8 2021). https://www.washingtonpost.com/opinions/2021/08/19/apple-csam-abuse-encryption-security-privacy-dangerous/

[241] Damon McCoy, Kevin S. Bauer, Dirk Grunwald, Tadayoshi Kohno, and Douglas C. Sicker. 2008. Shining Light in Dark Places: Understanding the Tor Network. In *PETS 2008: 8th International Symposium on Privacy Enhancing Technologies (Lecture Notes in Computer Science, Vol. 5134)*, Nikita Borisov and Ian Goldberg (Eds.). Springer, Heidelberg, Germany, Leuven, Belgium, 63–76. https://doi.org/10.1007/978-3-540-70630-4_5

[242] India McGhee, Jennifer Bayzick, April Kontostathis, Lynne Edwards, Alexandra McBride, and Emma Jakubowski. 2011. Learning to identify internet sexual predation. *International Journal of Electronic Commerce* 15, 3 (2011), 103–122.

[243] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R Savagaonkar. 2013. Innovative instructions and software model for isolated execution. *Hasp@ isca* 10, 1 (2013).

[244] India McKinney and Erica Portnoy. 2021. Apple's Plan to "Think Different" About Encryption Opens a Backdoor to Your Private Life. *Electronic Frontier Foundation* (5 8 2021). https://www.eff.org/deeplinks/2021/08/apples-plan-think-different-about-encryption-opens-backdoor-your-private-life

[245] Meedan. 2020. One year of running the WhatsApp End-to-End Fact-Checking project. https://meedan.com/post/one-of-year-of-running-the-end-end-to-fact-checking-project

[246] Luca Melis, Hassan Jameel Asghar, Emiliano De Cristofaro, and Mohamed Ali Kaafar. 2016. Private processing of outsourced network functions: Feasibility and constructions. In *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. 39–44.

[247] Luca Melis, George Danezis, and Emiliano De Cristofaro. 2016. Efficient Private Statistics with Succinct Sketches. In *ISOC Network and Distributed System Security Symposium – NDSS 2016*. The Internet Society, San Diego, CA, USA.

[248] Philipe Melo, Johnnatan Messias, Gustavo Resende, Kiran Garimella, Jussara Almeida, and Fabrício Benevenuto. 2019. Whatsapp monitor: A fact-checking system for whatsapp. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 13. 676–677.

[249] Ralph C Merkle. 1987. A digital signature based on a conventional encryption function. In *Conference on the theory and application of cryptographic techniques*. Springer, 369–378.

[250] Meta 2019. *Open-Sourcing Photo- and Video-Matching Technology to Make the Internet Safer.* Meta. https://about.fb.com/news/2019/08/open-source-photo-video-matching/

[251] Meta. 2022. Transparency Center. https://transparency.fb.com/

[252] Nat Meysenburg, Lauren Sarkesian, Ross Schulman, and Andi Wilson Thompson. 2021. A Technical Explainer on Apple's Concerning Privacy Changes. *New America* (30 8 2021). https://www.newamerica.org/oti/briefs/a-technical-explainer-on-apples-concerning-privacy-changes/

[253] Microsoft. 2018. *Skype Private Conversation Technical Whitepaper.* https://az705183.vo.msecnd.net/onlinesupportmedia/onlinesupport/media/skype/documents/skype-private-conversation-white-paper.pdf

[254] Microsoft 2021. *PhotoDNA.* Microsoft. https://www.microsoft.com/en-us/photodna

[255] Microsoft Docs. 2022. BitLocker. https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-overview

[256] Bonnie Mitchell, Krystle Kaul, G. S. McNamara, Michelle Tucker, Jacqueline Hicks, Colin Bliss, Rhonda Ober, Danell Castro, Amber Wells, Catalina Reguerin, Cindy Green-Ortiz, and Ken Stavinoha. 2017. *Going dark: impact to intelligence and law enforcement and threat mitigation.*

[257] Miljana Mladenović, Vera Ošmjanski, and Staša Vujičić Stanković. 2021. Cyber-aggression, cyberbullying, and cyber-grooming: a survey and research challenges. *ACM Computing Surveys (CSUR)* 54, 1 (2021), 1–42.

[258] Saeid Mofrad, Fengwei Zhang, Shiyong Lu, and Weidong Shi. 2018. A comparison study of Intel SGX and AMD memory encryption technology. In *Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy*. 1–8.

[259] Evgeny Morozov. 2013. *To save everything, click here: The folly of technological solutionism.* Public Affairs.

[260] Colin Morris and Graeme Hirst. 2012. Identifying Sexual Predators by SVM Classification with Lexical and Behavioral Features.. In *CLEF (Online Working Notes/Labs/Workshop)*, Vol. 12. 29.

[261] Laura Murphy and Megan Cacace. 2020. Facebook's civil rights audit–Final report. *Menlo Park, CA: Facebook* (2020).

[262] Mohamed Nabeel, Enes Altinisik, Haipei Sun, Issa Khalil, Hui Wang, and Ting Yu. 2021. CADUE: Content-Agnostic Detection of Unwanted Emails for Enterprise Security. In *24th International Symposium on Research in Attacks, Intrusions and Defenses*. 205–219.

[263] Maithili Narasimha and Gene Tsudik. 2007. Privacy-preserving revocation checking with modified crls. In *European Public Key Infrastructure Workshop*. Springer, 18–33.

[264] National Academies of Sciences, Engineering, and Medicine and others. 2018. *Decrypting the Encryption Debate: A Framework for Decision Makers.* National Academies Press.

[265] National Center for Missing and Exploited Children. 2022. 2021 CyberTipline Reports by Electronic Service Providers (ESP). https://www.missingkids.org/content/dam/missingkids/pdfs/2021-reports-by-esp.pdf

[266] National Center for Missing and Exploited Children. 2022. CyberTipline 2021 Report. https://www.missingkids.org/gethelpnow/cybertipline/cybertiplinedata

[267] Arambam Neelima and Kh Manglem Singh. 2016. Perceptual hash function based on scale-invariant feature transform and singular value decomposition. *Comput. J.* 59, 9 (2016), 1275–1281.

[268] Lily Hay Newman. 2022. Apple Kills Its Plan to Scan Your Photos for CSAM. Here's what's next. *Wired* (7 12 2022). https://www.wired.com/story/apple-photo-scanning-csam-communication-safety-messages/

[269] Khoa Nguyen, Reihaneh Safavi-Naini, Willy Susilo, Huaxiong Wang, Yanhong Xu, and Neng Zeng. 2021. Group Encryption: Full Dynamicity, Message Filtering and Code-Based Instantiation. In *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part II (Lecture Notes in Computer Science, Vol. 12711)*, Juan Garay (Ed.). Springer, Heidelberg, Germany, Virtual Event, 678–708. https://doi.org/10.1007/978-3-030-75248-4_24

[270] Jack Nicas, Raymond Zhong, and Daisuke Wakabayashi. 2021. Inside Apple's Compromises in China: A Times Investigation. *New York Times* (17 5 2021). https://www.nytimes.com/2021/05/17/technology/apple-china-censorship-data.html

[271] Pegah Nikbakht Bideh and Nicolae Paladi. 2022. Chuchotage: In-line Software Network Protocol Translation for (D) TLS. In *International Conference on Information and Communications Security*. Springer, 589–607.

[272] Alexander Nilsson, Pegah Nikbakht Bideh, and Joakim Brorsson. 2020. A survey of published attacks on Intel SGX. *arXiv preprint arXiv:2006.13598* (2020).

[273] Jianting Ning, Xinyi Huang, Geong Sen Poh, Shengmin Xu, Jia-Ch'ng Loh, Jian Weng, and Robert H. Deng. 2020. Pine: Enabling Privacy-Preserving Deep Packet Inspection on TLS with Rule-Hiding and Fast Connection Establishment. In *ESORICS 2020: 25th European Symposium on Research in Computer Security, Part I (Lecture Notes in Computer Science, Vol. 12308)*, Liqun Chen, Ninghui Li, Kaitai Liang, and Steve A. Schneider (Eds.). Springer, Heidelberg, Germany, Guildford, UK, 3–22. https://doi.org/10.1007/978-3-030-58951-6_1

[274] Jianting Ning, Geong Sen Poh, Jia-Ch'ng Loh, Jason Chia, and Ee-Chien Chang. 2019. PrivDPI: Privacy-Preserving Encrypted Traffic Inspection with Reusable Obfuscated Rules. In *ACM CCS 2019: 26th Conference on Computer and Communications Security*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, 1657–1670. https://doi.org/10.1145/3319535.3354204

[275] No Place To Hide 2022. No Place To Hide. https://www.noplacetohide.org.uk

[276] Mohammad Norouzi, Ali Punjani, and David J Fleet. 2012. Fast search in hamming space with multi-index hashing. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 3108–3115.

[277] Mark O'Neill, Scott Ruoti, Kent Seamons, and Daniel Zappala. 2017. TLS inspection: how often and who cares? *IEEE Internet Computing* 21, 3 (2017), 22–29.

[278] OpenNet Initiative. 2011. West Censoring East: The Use of Western Technologies by Middle East Censors. (2011). https://opennet.net/sites/opennet.net/files/ONI_WestCensoringEast.pdf

[279] Simon Oya and Florian Kerschbaum. 2021. Hiding the Access Pattern is Not Enough: Exploiting Search Pattern Leakage in Searchable Encryption. In *USENIX Security 2021: 30th USENIX Security Symposium*, Michael Bailey and Rachel Greenstadt (Eds.). USENIX Association, 127–142.

[280] Alexander Panchenko, Richard Beaufort, and Cedrick Fairon. 2012. Detection of child sexual abuse media on p2p networks: Normalization and classification of associated filenames. In *Proceedings of the LREC Workshop on Language Resources for Public Security Applications*. 27–31.

[281] Anchal Pandey, Sukumar Moharana, Debi Prasanna Mohanty, Archit Panwar, Dewang Agarwal, and Siva Prasad Thota. 2021. On-Device Content Moderation. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.

[282] Etienne Papegnies, Vincent Labatut, Richard Dufour, and Georges Linarès. 2017. Detection of abusive messages in an on-line community.. In *CORIA*. 153–168.

[283] Nilay Patel, Riana Pfefferkorn, and Jennifer King. 2021. Here's why Apple's new child safety features are so controversial. https://www.theverge.com/22617554/apple-csam-child-safety-features-jen-king-riana-pfefferkorn-interview-decoder

[284] Manas A Pathak, Mehrbod Sharifi, and Bhiksha Raj. 2011. Privacy preserving spam filtering. *arXiv preprint arXiv:1102.4021* (2011).

[285] Charlotte Peale, Saba Eskandarian, and Dan Boneh. 2021. Secure Complaint-Enabled Source-Tracking for Encrypted Messaging. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi (Eds.). ACM, 1484–1506. https://doi.org/10.1145/3460120.3484539

[286] Roel Peeters and Andreas Pashalidis. 2013. Privacy-friendly checking of remote token blacklists. In *IFIP Working Conference on Policies and Research in Identity Management*. Springer, 18–33.

[287] Nick Pendar. 2007. Toward spotting the pedophile telling victim from predator in text chats. In *International Conference on Semantic Computing (ICSC 2007)*. IEEE, 235–241.

[288] Mayana Pereira, Rahul Dodhia, Hyrum Anderson, and Richard Brown. 2020. Metadata-based detection of child sexual abuse material. *arXiv preprint arXiv:2010.02387* (2020).

[289] Tervor Perrin and Moxie Marlinspike. 2016. *The Double Ratchet Algorithm.* https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf

[290] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. 2019. Spot-light: Lightweight private set intersection from sparse OT extension. In *Annual International Cryptology Conference*. Springer, 401–431.

[291] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. 2020. PSI from PaXoS: fast, malicious private set intersection. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 739–767.

[292] Benny Pinkas, Thomas Schneider, and Michael Zohner. 2014. Faster private set intersection based on OT extension. In *23rd USENIX Security Symposium (USENIX Security 14)*. 797–812.

[293] Benny Pinkas, Thomas Schneider, and Michael Zohner. 2018. Scalable private set intersection based on OT extension. *ACM Transactions on Privacy and Security (TOPS)* 21, 2 (2018), 1–35.

[294] Rishabh Poddar, Chang Lan, Raluca Ada Popa, and Sylvia Ratnasamy. 2018. SafeBricks: Shielding Network Functions in the Cloud. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*. 201–216.

[295] Geong Sen Poh, Dinil Mon Divakaran, Hoon Wei Lim, Jianting Ning, and Achintya Desai. 2021. A survey of privacy-preserving techniques for encrypted traffic inspection over network middleboxes. *arXiv preprint arXiv:2101.04338* (2021).

[296] Hoi Ting Poon and Ali Miri. 2016. Scanning for viruses on encrypted cloud storage. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*. IEEE, 954–959.

[297] Erica Portnoy. 2019. Why Adding Client-Side Scanning Breaks End-To-End Encryption. *Electronic Frontier Foundation* (1 11 2019). https://www.eff.org/deeplinks/2019/11/why-adding-client-side-scanning-breaks-end-end-encryption

[298] Helena Pozniak. 2020. The child safety protocol: In dark corners of the internet, there have been horrific consequences to children living more online during the coronavirus lockdown. Are tech giants doing enough to protect them? And will greater privacy measures allow abuse to go unchecked? *Engineering & Technology* 15, 7/8 (2020), 74–77.

[299] Matthew Prince. 2022. Blocking Kiwifarms. https://blog.cloudflare.com/kiwifarms-blocked/

[300] Jonathan Prokos, Tushar M Jois, Neil Fendley, Roei Schuster, Matthew Green, Eran Tromer, and Yinzhi Cao. 2021. Squint Hard Enough: Evaluating Perceptual Hashing with Machine Learning. *Cryptology ePrint Archive* (2021).

[301] Chaitanya Rahalkar and Anushka Virgaonkar. 2022. SoK: Content Moderation Schemes in End-to-End Encrypted Systems. *arXiv preprint arXiv:2208.11147* (2022).

[302] Ram Sundara Raman, Adrian Stoll, Jakub Dalek, Reethika Ramesh, Will Scott, and Roya Ensafi. 2020. Measuring the Deployment of Network Censorship Filters at Global Scale.. In *NDSS*. https://www.ndss-symposium.org/wp-content/uploads/2020/02/23099-paper.pdf

[303] Sara Ramezanian, Tommi Meskanen, Masoud Naderpour, Ville Junnila, and Valtteri Niemi. 2020. Private membership test protocol with low communication complexity. *Digital Communications and Networks* 6, 3 (2020), 321–332.

[304] Sara Ramezanian, Tommi Meskanen, and Valtteri Niemi. 2021. Parental control with edge computing and 5g networks. In *2021 29th Conference of Open Innovations Association (FRUCT)*. IEEE, 290–300.

[305] Sara Ramezanian and Valtteri Niemi. 2019. Privacy preserving cyberbullying prevention with ai methods in 5g networks. In *2019 25th Conference of Open Innovations Association (FRUCT)*. IEEE, 265–271.

[306] Kyle Rankin. 2018. Tor Hidden Services. *Linux Journal* (23 5 2018). https://www.linuxjournal.com/content/tor-hidden-services

[307] Devin Reich, Ariel Todoki, Rafael Dowsley, Martine De Cock, et al. 2019. Privacy-preserving classification of personal text messages with secure multi-party computation. *Advances in Neural Information Processing Systems* 32 (2019).

[308] Julio CS Reis, Philipe Melo, Kiran Garimella, and Fabrício Benevenuto. 2020. Detecting Misinformation on WhatsApp without Breaking Encryption. *CoRR* (2020).

[309] Hao Ren, Hongwei Li, Dongxiao Liu, Guowen Xu, Nan Cheng, and Xuemin Sherman Shen. 2020. Privacy-preserving efficient verifiable deep packet inspection for cloud-assisted middlebox. *IEEE Transactions on Cloud Computing* (2020).

[310] Hao Ren, Hongwei Li, Dongxiao Liu, Guowen Xu, and Xuemin Sherman Shen. 2021. Enabling Secure and Versatile Packet Inspection with Probable Cause Privacy for Outsourced Middlebox. *IEEE Transactions on Cloud Computing* (2021).

[311] Hao Ren, Hongwei Litt, Dongxiao Liu, and Xuemin Sherman Shen. 2019. Toward efficient and secure deep packet inspection for outsourced middlebox. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[312] Yuqing Ren and Robert E Kraut. 2011. A simulation for designing online community: Member motivation, contribution, and discussion moderation. *Information Systems Research* (2011), 1–41.

[313] Eric Rescorla. 2021. More on Apple's client-side CSAM scanning. *Educated Guesswork* (16 8 2021). https://educatedguesswork.org/posts/apple-csam-more/

[314] Eric Rescorla and Tim Dierks. 2018. *RFC 8446: The transport layer security (TLS) protocol version 1.3.* Technical Report. Internet Engineering Task Force.

[315] Amanda Resende, Davis Railsback, Rafael Dowsley, Anderson C. A. Nascimento, and Diego F. Aranha. 2021. Fast Privacy-Preserving Text Classification based on Secure Multiparty Computation. Cryptology ePrint Archive, Report 2021/069. https://eprint.iacr.org/2021/069.

[316] ricochet-im. 2017. Anonymous metadata-resistant instant messaging that just works. https://github.com/ricochet-im/ricochet

[317] Christian X Ries and Rainer Lienhart. 2014. A survey on visual adult image recognition. *Multimedia tools and applications* 69, 3 (2014), 661–688.

[318] Sarah Roberts. 2019. *Behind the Screen: Content Moderation in the Shadows of Social Media*. Yale University Press.

[319] Phillip Rogaway. 2002. Authenticated-Encryption With Associated-Data. In *ACM CCS 2002: 9th Conference on Computer and Communications Security*, Vijayalakshmi Atluri (Ed.). ACM Press, Washington, DC, USA, 98–107. https://doi.org/10.1145/586110.586125

[320] Michael Rosenberg, Mary Maller, and Ian Miers. 2022. Snarkblock: Federated anonymous blocklisting from hidden common input aggregate proofs. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 948–965.

[321] Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. *CoRR* abs/1701.08118 (2017). arXiv:1701.08118 http://arxiv.org/abs/1701.08118

[322] Jenna Ruddock and Justin Sherman. 2021. Widening the Lens on Content Moderation. *Joint PIJIP/TLS Research Paper Series* 69 (7 2021). https://digitalcommons.wcl.american.edu/research/69

[323] Théo Ryffel, David Pointcheval, Francis Bach, Edouard Dufour-Sans, and Romain Gay. 2019. Partially encrypted deep learning using functional encryption. *Advances in Neural Information Processing Systems* 32 (2019).

[324] Safer. 2020. How to Employ Company Policies & Best Practices to Fight the Spread of CSAM. https://safer.io/resources/platform-protection-101/

[325] Punyajoy Saha, Binny Mathew, Kiran Garimella, and Animesh Mukherjee. 2021. "Short is the Road that Leads from Fear to Hate": Fear Speech in Indian WhatsApp Groups. In *Proceedings of the Web Conference 2021*. 1110–1121.

[326] Julian Sanchez. 2021. Apple's iPhone: Now With Built-In Surveillance. *Cato Institute* (6 8 2021). https://www.cato.org/blog/apples-iphone-now-built-surveillance

[327] Sarah Scheffler, Anunay Kulshrestha, and Jonathan Mayer. 2022. Public Verification for Private Hash Matching.

[328] Liron Schiff and Stefan Schmid. 2016. PRI: privacy preserving inspection of encrypted network traffic. In *2016 IEEE Security and Privacy Workshops (SPW)*. IEEE, 296–303.

[329] Alexander Schlögl and Rainer Böhme. 2020. eNNclave: offline inference with model confidentiality. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*. 93–104.

[330] Aaron Schulman, Dave Levin, and Neil Spring. 2014. RevCast: Fast, Private Certificate Revocation over FM Radio. In *ACM CCS 2014: 21st Conference on Computer and Communications Security*, Gail-Joon Ahn, Moti Yung, and Ninghui Li (Eds.). ACM Press, Scottsdale, AZ, USA, 799–810. https://doi.org/10.1145/2660267.2660376

[331] Joseph Seering. 2020. Reconsidering Community Self-Moderation: the Role of Research in Supporting Community-Based Models for Online Content Moderation. *Proc. ACM Hum.-Comput. Interact* 3 (2020).

[332] Haeseung Seo, Aiping Xiong, and Dongwon Lee. 2019. Trust it or not: Effects of machine-learning warnings in helping individuals mitigate misinformation. In *Proceedings of the 10th ACM Conference on Web Science*. 265–274.

[333] Akash Shah, Nishanth Chandran, Mesfin Dema, Divya Gupta, Arun Gururajan, and Huan Yu. 2021. Secure Featurization and Applications to Secure Phishing Detection. Cryptology ePrint Archive, Report 2021/1448. https://eprint.iacr.org/2021/1448.

[334] Imtiyazuddin Shaik, Nitesh Emmadi, Harshal Tupsamudre, Harika Narumanchi, and Rajan Mindigal Alasingara Bhattachar. 2021. Privacy Preserving Machine Learning for Malicious URL Detection. In *International Conference on Database and Expert Systems Applications*. Springer, 31–41.

[335] Wazen M Shbair, Thibault Cholez, Jérôme François, and Isabelle Chrisment. 2020. A survey of HTTPS traffic and services identification approaches. *arXiv preprint arXiv:2008.08339* (2020).

[336] Justine Sherry, Chang Lan, Raluca Ada Popa, and Sylvia Ratnasamy. 2015. Blindbox: Deep packet inspection over encrypted traffic. In *Proceedings of the 2015 ACM conference on special interest group on data communication*. 213–226.

[337] Junjie Shi, Yuan Zhang, and Sheng Zhong. 2015. Privacy-preserving network functionality outsourcing. *arXiv preprint arXiv:1502.00389* (2015).

[338] Xiaofeng Shi, Shouqian Shi, Minmei Wang, Jonne Kaunisto, and Chen Qian. 2021. On-device IoT Certificate Revocation Checking with Small Memory and Low Latency. In *ACM CCS 2021: 28th Conference on Computer and Communications Security*, Giovanni Vigna and Elaine Shi (Eds.). ACM Press, Virtual Event, Republic of Korea, 1118–1134. https://doi.org/10.1145/3460120.3484580

[339] Ming-Wei Shih, Mohan Kumar, Taesoo Kim, and Ada Gavrilovska. 2016. S-nfv: Securing nfv states by using sgx. In *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. 45–48.

[340] Leslie F Sikos. 2020. Packet analysis for network forensics: A comprehensive survey. *Forensic Science International: Digital Investigation* 32 (2020), 200892.

[341] Priyanka Singh and Hany Farid. 2019. Robust Homomorphic Image Hashing. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 11–18. http://openaccess.thecvf.com/content_CVPRW_2019/html/Media_Forensics/Singh_Robust_Homomorphic_Image_Hashing_CVPRW_2019_paper.html

[342] Mohit Singhal, Chen Ling, Nihal Kumarswamy, Gianluca Stringhini, and Shirin Nilizadeh. 2022. SoK: Content Moderation in Social Media, from Guidelines to Enforcement, and Research to Practice. *arXiv preprint arXiv:2206.14855* (2022).

[343] Trevor Smith, Luke Dickenson, and Kent E. Seamons. 2020. Let's Revoke: Scalable Global Certificate Revocation. In *ISOC Network and Distributed System*

Security Symposium – NDSS 2020. The Internet Society, San Diego, CA, USA.

[344] softonic. 2022. TrueCrypt. https://truecrypt.en.softonic.com/

[345] John Solis and Gene Tsudik. 2006. Simple and Flexible Revocation Checking with Privacy. In *PET 2006: 6th International Workshop on Privacy Enhancing Technologies (Lecture Notes in Computer Science, Vol. 4258)*, George Danezis and Philippe Golle (Eds.). Springer, Heidelberg, Germany, Cambridge, UK, 351–367. https://doi.org/10.1007/11957454_20

[346] Martin Steinebach, Huajian Liu, and York Yannikos. 2012. Forbild: Efficient robust image hashing. In *Media Watermarking, Security, and Forensics 2012*, Vol. 8303. International Society for Optics and Photonics, 83030O.

[347] Janet Sternberg. 2012. *Misbehavior in cyber places: The regulation of online conduct in virtual communities on the Internet.* Rowman & Littlefield.

[348] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. 2021. Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash. *arXiv preprint arXiv:2111.06628* (2021).

[349] Hao Sun, Jinshu Su, Xiaofeng Wang, Rongmao Chen, Yujing Liu, and Qiaolin Hu. 2017. Primal: Cloud-based privacy-preserving malware detection. In *Australasian Conference on Information Security and Privacy*. Springer, 153–172.

[350] Sandeep Tamrakar, Jian Liu, Andrew Paverd, Jan-Erik Ekberg, Benny Pinkas, and N. Asokan. 2017. The Circle Game: Scalable Private Membership Test Using Trusted Hardware. In *ASIACCS 17: 12th ACM Symposium on Information, Computer and Communications Security*, Ramesh Karri, Ozgur Sinanoglu, Ahmad-Reza Sadeghi, and Xun Yi (Eds.). ACM Press, Abu Dhabi, United Arab Emirates, 31–44.

[351] Arunabha Tarafdar, Hrittika Paul, Shubhojit Gupta, Labani Acharjee, Akash Sah, and Soham Ganguly. 2021. Spam Detection Using Threshold Method on Whatsapp Image Data. In *Advances in Smart Communication Technology and Information Processing*. Springer, 317–325.

[352] M Taufiq Nuruzzaman, Changmoo Lee, Mohd Fikri Azli bin Abdullah, and Deokjai Choi. 2012. Simple SMS spam filtering on independent mobile phone. *Security and Communication Networks* 5, 10 (2012), 1209–1220.

[353] The Conversation. 2016. How we moderate off-topic comments. https://theconversation.com/how-we-moderate-off-topic-comments-64155

[354] Kurt Thomas, Devdatta Akhawe, Michael Bailey, Dan Boneh, Elie Bursztein, Sunny Consolvo, Nicola Dell, Zakir Durumeric, Patrick Gage Kelley, Deepak Kumar, et al. 2021. SoK: Hate, harassment, and the changing landscape of online abuse. In *2021 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 247–267.

[355] Thorn. 2019. Benchmarking. https://perception.thorn.engineering/en/latest/examples/benchmarking.html

[356] Bohdan Trach, Alfred Krohmer, Sergei Arnautov, Franz Gregor, Pramod Bhatotia, and Christof Fetzer. 2017. Slick: Secure middleboxes using shielded execution. *arXiv preprint arXiv:1709.04226* (2017).

[357] Bohdan Trach, Alfred Krohmer, Franz Gregor, Sergei Arnautov, Pramod Bhatotia, and Christof Fetzer. 2018. Shieldbox: Secure middleboxes using shielded execution. In *Proceedings of the Symposium on SDN Research*. 1–14.

[358] Anton Troianovski and Adam Satariano. 2021. Apple and Google Remove 'Navalny' Voting App in Russia. *New York Times* (17 9 2021). https://www.nytimes.com/2021/09/17/world/europe/russia-navalny-app-election.html

[359] Nirvan Tyagi, Paul Grubbs, Julia Len, Ian Miers, and Thomas Ristenpart. 2019. Asymmetric Message Franking: Content Moderation for Metadata-Private End-to-End Encryption. In *Advances in Cryptology – CRYPTO 2019, Part III (Lecture Notes in Computer Science, Vol. 11694)*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer, Heidelberg, Germany, Santa Barbara, CA, USA, 222–250. https://doi.org/10.1007/978-3-030-26954-8_8

[360] Nirvan Tyagi, Ian Miers, and Thomas Ristenpart. 2019. Traceback for End-to-End Encrypted Messaging. In *ACM CCS 2019: 26th Conference on Computer and Communications Security*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, 413–430. https://doi.org/10.1145/3319535.3354243

[361] Nik Unger, Sergej Dechand, Joseph Bonneau, Sascha Fahl, Henning Perl, Ian Goldberg, and Matthew Smith. 2015. SoK: secure messaging. In *2015 IEEE Symposium on Security and Privacy*. IEEE, 232–249.

[362] US Department of Justice. 2016. National Strategy for Child Exploitation Prevention and Interdiction: A Report to Congress. (2016). https://www.justice.gov/psc/file/842411/download

[363] Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F Wenisch, Yuval Yarom, and Raoul Strackx. 2018. Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution. In *27th USENIX Security Symposium (USENIX Security 18)*. 991–1008.

[364] Ramarathnam Venkatesan, S-M Koon, Mariusz H Jakubowski, and Pierre Moulin. 2000. Robust image hashing. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, Vol. 3. IEEE, 664–666.

[365] VeraCrypt. 2022. VeraCrypt. https://veracrypt.fr/en/Home.html

[366] Rakesh Verma, Narasimha Shashidhar, and Nabil Hossain. 2012. Detecting phishing emails the natural language way. In *European Symposium on Research in Computer Security*. Springer, 824–841.

[367] Ryan Wails, Aaron Johnson, Daniel Starin, Arkady Yerukhimovich, and S. Dov Gordon. 2019. Stormy: Statistics in Tor by Measuring Securely. In *ACM CCS 2019: 26th Conference on Computer and Communications Security*, Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz (Eds.). ACM Press, 615–632. https://doi.org/10.1145/3319535.3345650

[368] Chih-Chien Wang and Sheng-Yi Chen. 2007. Using header session messages to anti-spamming. *Computers & Security* 26, 5 (2007), 381–390.

[369] Juan Wang, Shirong Hao, Yi Li, Zhi Hong, Fei Yan, Bo Zhao, Jing Ma, and Huanguo Zhang. 2019. TVIDS: Trusted virtual IDS with SGX. *China Communications* 16, 10 (2019), 133–150.

[370] Sicong Wang, Naveen Karunanayake, Tham Nguyen, and Suranga Seneviratne. 2020. Privacy-Preserving Spam Filtering using Functional Encryption. *arXiv preprint arXiv:2012.04163* (2020).

[371] William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*. 19–26.

[372] Wentao Wei, Jie Wang, Zheng Yan, and Wenxiu Ding. 2022. EPMDroid: Efficient and privacy-preserving malware detection based on SGX through data fusion. *Information Fusion* 82 (2022), 43–57.

[373] Li Weng and Bart Preneel. 2011. A secure perceptual hash algorithm for image content authentication. In *IFIP International Conference on Communications and Multimedia Security*. Springer, 108–121.

[374] WhatsApp. 2020. About Suspicious Links. https://faq.whatsapp.com/web/chats/suspicious-links

[375] Gang Xu, Leonardo Aguilera, and Yong Guan. 2012. Accountable Anonymity: A Proxy Re-Encryption Based Anonymous Communication System. In *18th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2012, Singapore, December 17-19, 2012*. IEEE Computer Society, 109–116. https://doi.org/10.1109/ICPADS.2012.25

[376] Hiroki Yamamuro, Keisuke Hara, Masayuki Tezuka, Yusuke Yoshida, and Tanaka Keisuke. 2021. Forward Secure Message Franking. https://journal-home.s3.ap-northeast-2.amazonaws.com/site/icisc2021/presentation/paper_25.pdf

[377] Jeff Yan and Pook Leong Cho. 2006. Enhancing collaborative spam detection with bloom filters. In *2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*. IEEE, 414–428.

[378] Rupeng Yang, Man Ho Au, Junzuo Lai, Qiuliang Xu, and Zuoxia Yu. 2017. Lattice-Based Techniques for Accountable Anonymity: Composition of Abstract Stern's Protocols and Weak PRF with Efficient Protocols from LWR. Cryptology ePrint Archive, Report 2017/781. https://eprint.iacr.org/2017/781.

[379] Andrew Chi-Chih Yao. 1986. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. IEEE, 162–167.

[380] Jing Yao, Xiangyi Meng, Yifeng Zheng, and Cong Wang. 2022. Privacy-Preserving Content-Based Similarity Detection Over in-the-Cloud Middleboxes. *IEEE Transactions on Cloud Computing* (2022).

[381] Asuhariet Ygvar and Cory Cornelius. 2021. AppleNeuralHash2ONNX. https://github.com/AsuharietYgvar/AppleNeuralHash2ONNX/issues/1

[382] Xingliang Yuan, Xinyu Wang, Jianxiong Lin, and Cong Wang. 2016. Privacy-preserving deep packet inspection in outsourced middleboxes. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.

[383] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Shen. 2015. PIF: A personalized fine-grained spam filtering scheme with privacy preservation in mobile social networks. *IEEE Transactions on Computational Social Systems* 2, 3 (2015), 41–52.

[384] Kuan Zhang, Xiaohui Liang, Rongxing Lu, and Xuemin Sherman Shen. 2013. SAFE: A social based updatable filtering protocol with privacy-preserving in mobile social networks. In *2013 IEEE International Conference on Communications (ICC)*. IEEE, 6045–6049.

[385] Zhongjun Zhang, Jianfeng Wang, Yunling Wang, Yaping Su, and Xiaofeng Chen. 2019. Towards efficient verifiable forward secure searchable symmetric encryption. In *European symposium on research in computer security*. Springer, 304–321.

[386] Wei Zheng, Ying Wu, Xiaoxue Wu, Chen Feng, Yulei Sui, Xiapu Luo, and Yajin Zhou. 2021. A survey of Intel SGX and its applications. *Frontiers of Computer Science* 15, 3 (2021), 1–15.

[387] Zhenyu Zhou and Theophilus Benson. 2015. Towards a safe playground for HTTPS and middle boxes with QoS2. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*. 7–12.

[388] Wanzheng Zhu, Hongyu Gong, Rohan Bansal, Zachary Weinberg, Nicolas Christin, Giulia Fanti, and Suma Bhat. 2021. Self-Supervised Euphemism Detection and Identification for Content Moderation. In *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, San Francisco, CA, USA, 229–246. https://doi.org/10.1109/SP40001.2021.00075

A LITERATURE SEARCH DETAILS

This section contains additional details on our literature search beyond those in Section 3.

The initial sample for our literature search was created by running the following queries in August 2022 in computer science and cryptography-related academic venues: content moderation, CSAM, end to end, malware, misinformation, porn, pornography, and spam. The academic venues were ACM CCS,[15] CRYPTO,[16] NDSS,[16] PETS,[16] IEEE S&P,[17] Usenix Security,[18] arXiv CS,[19] and IACR ePrint,[20]. We additionally examined the top 200 results from Google Scholar for "encrypted content moderation" and "end to end encrypted content moderation." For the computer science venues, we iteratively refined query terms per venue if the search returned too many results to manually search and inspection revealed that the results were irrelevant (e.g. a search for "content moderation" would surface matches for "content"). The final queries to form the initial set, with their result counts, were as follows:

- ACM CCS: "content moderation" (4), CSAM (1), "end to end encryption" (61), misinformation (26), porn (16), pornography (42), "privacy-preserving" AND "malware-detection" (10), "private" AND "malware detection" (51), spam AND encryption
- CRYPTO: content moderation (1), CSAM (0), end to end (2), malware (0), misinformation (0), porn (0), pornography (0), spam (1)
- NDSS: content moderation (0), CSAM (0), end to end (3), malware (30), misinformation (0), porn (0), pornography (0), spam (7)
- PETS: content moderation (0), end to end (1), malware (1), misinformation (0), porn (0), pornography (0), spam (0)
- IEEE S&P: content moderation (1), CSAM (0), end to end encryption (17), misinformation (0), porn (0), pornography (0), child pornography (0), privacy malware (84), spam (9)
- Usenix Security: "content moderation" (1), CSAM (0), "end to end encryption" (8), "malware detection" (9), misinformation (0), porn (0), pornography (1), spam (27)
- arXiv CS: "content moderation" (74), CSAM (10), "end to end encryption" (72), encryption misinformation (5), porn (14), pornography (31), privacy-preserving malware detection (9), private malware detection (20), encryption spam (11)
- IACR ePrint: content moderation (428), CSAM (0), "end to end encryption" (31), "end to end" (173), malware (32), misinformation (3), porn (0), pornography (0), child pornography (7), spam (28)
- Google Scholar: encrypted content moderation (200 examined), end to end encrypted content moderation (200 examined)
- UK Safety Tech Challenge [163]: Details of the five winners obtained from the End of Programme Supplier Showcase event [120].
- The following documentation for E2EE services: the Apple child safety page on August 5, 2021,[21] Google Messages support,[22] Signal support,[23] WhatsApp Help Center[24]

As mentioned in Section 3, we manually examined papers to identify relevant works. To be considered relevant, works must:

1. Include at least one subsection on a content moderation method, system, implementation, or construction, and
2. Provide either partial client privacy with respect to a class corresponding to the content moderation problem, or full client privacy (Definitions 2.3 and 2.4). For systems that can be run client-side, they must mention that they are intended to be used in an encrypted or private setting.

Once the initial set of relevant papers was identified, further works were identified from those works via snowball sampling. Citations were scraped via the publisher's API when available, then via Semantic Scholar or arXiv if located there, or extracted from the PDF paper itself if no other version was available. Forward references were found via Google Scholar's "cited by" feature.

A.1 Excluded works

The researchers excluded papers on topics adjacent but not identical to content moderation under E2EE. The researchers excluded many papers on the topic of detection/blocking of misbehaving users in anonymous networks, certificate transparency and revocation checking, moderation of the blockchain, implementations of Digital Rights Management (DRM) or watermarking schemes, key escrow or access control schemes, traitor tracing, or measurement of E2EE systems without moderation.

In the case that multiple versions of the work were identified, we only included one version. We included journal papers over conference papers, and conference papers over preprints or manuscripts. We also excluded any work where we could not get access either publicly, via institutional login, or through a loan to the Princeton University Library, and works that were not in English.

Finally, as discussed in Section 2.1, we excluded works that did not meet Definition 2.4 for results not in the problematic class C, in particular this means we excluded works that sent raw cryptographic or bloom filter hashes of all content to the server.

B MODERATION OF TLS TRAFFIC AT MIDDLEBOXES

Tables 2 and 4 show our detailed examination of each work we examined for our full literature search. Table 2 in Section 3 shows all non-TLS middlebox works. Here, Table 4 shows the works on middleboxes.

B.1 On the inclusion of privacy-preserving deep packet inspection

Prior analyses of content moderation in end-to-end encryption [187, 239, 301] have either been content-agnostic or considered content moderation only in the setting of social media or a general

[15] https://dl.acm.org/conference/ccs/proceedings
[16] searched via DBLP https://dblp.org/ and filtered by venue
[17] https://ieeexplore.ieee.org/Xplore/home.jsp filtered by venue
[18] https://www.usenix.org/publications/proceedings
[19] https://arxiv.org/search/cs
[20] https://eprint.iacr.org/search
[21] https://web.archive.org/web/20210805200549/https://www.apple.com/child-safety/
[22] https://support.google.com/messages/
[23] https://support.signal.org/hc/en-us
[24] https://faq.whatsapp.com/

Problem context		Detection											Response		Transparency	
Work	Goal	Exact matching	Rules/patterns	Machine learning	General Crypto/MPC	Trusted Hardware	Searchable Encryption	Size of largest ruleset	Latency	Communication	Security Against Server	Security Against Client	Server privacy	Client privacy	Transparency	Details
Yao et al. [380]	General middlebox	●				●		5M	0.5ms	×	Mal.	×	●	◐	○	
Grubbs et al. [150]	General middlebox		●		●			2M	3.1s		S.H.	Mal.	○		◔	Mention
Ren et al. [310]	General middlebox		●		●		●	3k	287s	0.4 MB	S.H.(NC)	×	●	◐	◔	Mention
Lai et al. [215]	General middlebox		●		●			24k	0.85ms	5x	S.H.		●	◐	○	
Fan et al. [108]	General middlebox		●		●		●	×	Asymp.	×	×	Mal.	●	◐	○	
Desmoulins et al. [94]	General middlebox		●		●		●	6k	2.3ms	64x	Mal.	S.H.	○	◐		
EndBox [137]	General middlebox		●			●		377	1.67ms	16%	S.H.	Mal.	●	◐	◐	Attestation
SPABox [107]	General middlebox		●		●			3k	0.5ms		S.H.		●	◐	◔	Mention
Zhou and Benson [387]	General middlebox	●						×	10s	×	S.H.			★[25]	○	
Li et al. [225]	Intrusion detection		●		●		●	1.6k	6.5ms	×	S.H.	Mal.	●	◐	◔	Mention
Chen et al. [59]	Intrusion detection		●		●		●	3k	103ms	407 KB	S.H.	Mal.	●	◐	◔	Honesty assumption
Jia and Zhang [182]	Intrusion detection	●	●		●		●	3k	267s	82 MB	×	Mal.	●	◐	◔	Honesty assumption
Chuchotage [271]	Intrusion detection		●			●		×	0.07s	×	TEE	Mal.	●	◐	◐	Attestation
Canard and Li [55]	Intrusion detection		●		●		●	3k	1.5us	×	S.H.	Mal.	●	◐	○	
Guo et al. [154]	Intrusion detection		●		●		●	1.6k	5ms	400 B	S.H.	Mal.	●	◐	○	
Bkakria et al. [36]	Intrusion detection		●		●		●		36ms	640 KB	S.H.	Mal.	●	◐	◔	Honesty assumption
Pine [273]	Intrusion detection		●		●		●	6k	665ms	350 KB	S.H.	Mal.	●	◐	◔	Mention
Ren et al. [309]	Intrusion detection		●		●		●	3k	3.82s	50 MB	S.H.	Mal.	●	◐	◔	Honesty assumption
Han et al. [157]	Intrusion detection		●			●		24k	×	×	TEE	Mal.	●	◐	◐	Attestation
Ren et al. [311]	Intrusion detection		●		●			3k	2s	×	S.H.	Mal.	●	◐	○	
TVIDS [369]	Intrusion detection		●			●		×	×	×	TEE		●	◐	◐	Attestation
LightBox [104]	Intrusion detection		●			●		3.7k	20us		TEE	Mal.	●	◐	○	
Guo et al. (A) [153]	Intrusion detection		●		●		●	1.6k	4ms	2x	S.H.	S.H.	●	◐	◔	Honesty assumption
ShieldBox [357]	Intrusion detection		●			●		×	40us		TEE	Mal.	●	◐	◐	Attestation
SafeBricks [294]	Intrusion detection		●			●		18k	×	16%	TEE	Mal.	●	◐	◐	Third party audit, Attestation
Snort [213]	Intrusion detection		●			●		3.4k	×		TEE	Mal.	●	◐	◐	Third party audit, Attestation
Alabdulatif et al. [8]	Intrusion detection			●	●				1.3s		S.H.		●	◐	○	
BlindIDS [54]	Intrusion detection		●		●		●	3k	74s	×	S.H.	Mal.	●	◐	○	
Trusted Click [78]	Intrusion detection		●			●		×	×	2x	TEE	Mal.	●	◐	◐	Attestation
SGX-Box [158]	Intrusion detection		●			●		26k	×	11.9%	TEE		●	◐	◐	Attestation
Slick [356]	Intrusion detection		●			●		10	18us	88%	TEE		●	◐	◐	Attestation
Melis et al. [246]	Intrusion detection		●		●		●	10	250ms	119 B	S.H.	×	●	★[26]	◔	Honesty assumption
S-NFV [339]	Intrusion detection		●			●		×	27us	8.79x	TEE	Mal.	●	◐	◐	Attestation
Shi et al. [337]	Intrusion detection		●		●			×	×	×	S.H.	Mal.	●	◐	◔	Mention
P2DPI [199]	Data exfiltration, Intrusion detection		●		●		●	2k	0.037ms	840 KB	S.H.	S.H.	●	◐	○	
PrivDPI [274]	Data exfiltration, Intrusion detection		●		●			3k	0.15s	49 B	S.H.	Mal.	●	◐	◔	Honesty assumption
CloudDPI [226]	Data exfiltration, Intrusion detection		●		●			96.6k	3ms	28.5 MB	S.H.	S.H.	●	◐	◔	Honesty assumption
Lin et al. [228]	Data exfiltration, Intrusion detection	●	●		●			1k	10us	1 MB	S.H.	S.H.	●	◐	○	
Yuan et al. [382]	Data exfiltration, Intrusion detection		●		●		●	3.2k	10us	3x	S.H.	S.H.	●	◐	◔	Mention
Embark [216]	Data exfiltration, Intrusion detection		●		●		●	100k	50ms	4.3x	S.H.	Mal.	●	◐	○	
PRI [328]	Data exfiltration, Intrusion detection		●			●		×	×	×	TEE	Mal.	●	◐	◐	Attestation
BlindBox [336]	Data exfiltration, Intrusion detection, Parental control		●		●		●	3k	33us	2.5x	S.H.	Mal.	●	◐	◔	Mention
Ramezanian et al. [304]	Parental control	●	●	●	●			100	1s	57.5 KB	S.H.	S.H.	●	●	◔	Mention

Table 4: Details of all TLS middleboxes found in our literature search. Latencies and communication corresponds to processing one packet on all rules. See caption of Table 2 for legend.

E2EE messaging service. The inclusion of privacy-preserving deep packet inspection in the modern content moderation debate is rare, in part because the focus as of late has been on applications rather than moderation of the general public by Internet Service Providers (ISPs), though internet infrastructure services still perform both voluntary and mandated content removal (e.g. [48, 299, 322]).

Privacy-preserving deep packet inspection meets all our criteria of E2EE content moderation: it occurs over an E2EE channel (web traffic encrypted with Transport Layer Security, or TLS [314]), it seeks to identify specific problematic content (e.g. unauthorized data exfiltration or intrusion detection) and it achieves at least partial client privacy, revealing no additional information about non-problematic content to the moderator (aside from false positives). It often also provides server privacy for the specific rules, patterns, or data being to be matched against the encrypted traffic.

These methods are also used in privacy-preserving parental control systems, occupy a curious space in between corporate moderation and social media moderation: The goals of parental control tend to be more aligned with social media goals: blocking content concerning specific topics, e.g. sexual or drug-related. However the technology used tends to be more highly related to the corporate network monitoring methods—we found many papers for which the technical scheme was self-described for use in both settings [54, 55, 108, 150, 182, 215, 216, 226, 274, 336, 336].

Criticisms of deep packet inspection—privacy-preserving or otherwise—also echo the current content moderation debate for E2EE. A 2017 survey by O'Neill et al. [277] investigated the general public's support of TLS proxies (without privacy preservation) and found that more than 70% were at least somewhat concerned that TLS proxies could be used by hackers or governments. 90% agreed that browsers should notify users of TLS proxies. However, there was support for uses in many specific settings: with notification, 80% believed the use of TLS proxies was permissible by companies for company devices, 69% said the same for elementary schools, and 64% for universities. This makes the problem context extremely important: Not all content moderation is for social media conversations among adults.

Thus, we unite these two areas of the literature. The current debate over child safety and misinformation in E2EE has lessons to learn—both helpful and cautionary—from privacy-preserving content moderation of TLS-encrypted network traffic.

C PROBLEM CONTEXTS: THREE EXAMPLES

This appendix contains three brief examples of how the problem context changes the landscape of feasible detection, response, and transparency mechanisms.

Child safety protections. Child safety incorporates a host of different topics that broadly seek to protect children from online harms, especially those of a sexual nature. Levy and Robinson identify seven "harm archetypes," distinguishing between offender-to-offender CSAM sharing, offender-to-victim grooming, live streaming of child abuse, non-CSAM communication between offenders (individual or group), consensual child-to-child indecent image sharing, and viral image sharing (in which CSAM is sent to shock or offend) [223]. Child trafficking is also frequently included in the discussion of child safety, e.g. [44, 114, 224]. Different interventions apply to different harms: Comparing images in messages against a list of known CSAM held by a clearinghouse such as the National Center for Missing and Exploited Children (NCMEC) will be much more effective for catching offender-to-offender image sharing and viral image sharing [139, 170, 223, 266, 324, 362], whereas textual and metadata analysis is more suited for detecting offender-to-child grooming or enticement [47, 234, 257]. The tension between privacy and child safety has been recognized for years, though no resolution has emerged (e.g. [1, 113, 163, 188, 223, 227, 244]). The same privacy-safety tradeoff occurs many forms of content moderation under encryption, but child safety is an especially important topic due to the horrific scale and type of the problem: NCMEC received more than 29 million reports of child sexual abuse material, in 2021 alone [265], and the magnitude of online sexual harms to children have grown significantly in the last several years [298]. For more information on content moderation for child safety, we direct the reader to [52, 152, 220, 223].

In 2021-22, the U.K. ran the "Safety Tech Challenge" (UK STC, [163]) in which companies built various kinds of child safety-focused content moderation systems for E2EE environments; these are described alongside the rest of our literature search.

Moderation of hate and harassment. Hate and harassment is broadly defined as persistent action toward an individual or group that is meant to cause emotional harm to the target, including causing fear of physical or sexual violence [66, 354]. This category itself incorporates a wide variety of behaviors; the recent work of Thomas et al. [354] identifies seven categories of hate. The category most relevant to this work is *toxic content*, e.g. hate speech, sexual harassment, or threats of violence.

For hate specifically, word filters for hate speech have been criticized both for being easy to evade [149] and for misunderstanding the context when a naively problematic word is being used in a positive way [115, 236, 371]. Many ML classifiers aim to detect hate speech, however they tend to have low accuracies and low agreements (and moreover, humans also have low agreement when it comes to identifying hate speech) [6, 321]. As such, automatically detecting hate speech under E2EE would have a high false positive rate. Moreover, for the kinds of toxic content under discussion, a user is in the loop who does not want the content and could report it. Automated classification may still have a role to avoid placing the "burden of responsibility [on] individual, isolated users" [126] (see also [79, 259]), however, in this case it may be more appropriate to perform the classification in a fully client private way.

Detecting data exfiltration. Many corporations use "TLS middleboxes" to perform various services, including detection of intruders and attempts at data exfiltration. Although the majority of companies today do this in a non-privacy preserving manner [89], there are some middleboxes which perform the detection in a partially client private way (see Section 1). These usually use searchable encryption or trusted hardware to ensure that only positive detections are revealed to the middlebox; non-matches remain as private as the underlying cryptography provides. The exact privacy and detection properties of these middleboxes vary, although partial client privacy is near-universal.

D ATTACKS ON PERCEPTUAL HASH FUNCTIONS

Recent works have demonstrated effective attacks on PHFs. In addition to evasion attacks, these works have also demonstrated partial inversion attacks, as well as targeted collision attacks. We describe each attack in turn:

Evasion. For the setting of matching via PHFs in content moderation, one of the most serious attacks is evading detection by creating an image that is highly similar to one on the list of problematic content, but has a different hash [159, 175, 209, 348]. This setting is concerning because the entire motivation for using PHFs rather than CHFs is the reduced false negative rate, thus, effective evasion attacks significantly lower the benefits of using PHFs while retaining the cost of a high FPR. Jain et al. [175] show both white-box and black-box evasion attacks on a variety of PHFs including PDQ and pHash. Hao et al. [159] show black-box attacks on pHash and a more robust variant known as Blockhash. Struppek et al. [348] demonstrate evasion attacks against NeuralHash along with prototype code. And Krawetz [209] shows a proof-of-concept evasion attack against PhotoDNA. The black-box attacks are especially concerning since they do not require any knowledge about how the algorithm actually works. This implies that these attacks are viable even in server-private settings. In settings that require some interactivity in order to compute the function, the server may be able to rate-limit clients who are computing the function too much, but given the frequency with which images and messages are sent in online communication, this form of rate-limiting will also likely interfere too much with normal communication. We expect black-box attacks to be feasible in the vast majority of content moderation scenarios.

Finding hash preimages. Another attack is inversion of hashes. Two independent proof-of-concept attacks have shown the feasibility of inverting hashes in PhotoDNA [22, 209], yielding a somewhat blurry and distorted version of an image which hashes to a particular known value. Although these attacks have not been demonstrated yet within the more formal research literature, the initial results demonstrate the importance of server privacy with respect to highly illegal content.

Target hash collisions. Finally, another line of attacks on PHFs create collisions with target hashes, e.g. [99] for pHash and other open-source hashes, and [348, 381] for NeuralHash. In the setting of Apple's CSAM detector, researchers and others posed concerns that if a hash in a CSAM list becomes known, target collision attacks could plant innocuous images on someone's device that would trigger a CSAM detection [1, 67, 187, 212, 252]. Attackers or protestors could also attempt to overwhelm Apple's human content moderation resources by triggering many adversarially-created matches that match CSAM hashes but are not themselves CSAM. Note that for matching via lists, adversarially induced false positives presuppose that a client has knowledge of at least one hash on the list. It is unclear whether this assumption is reasonable in practice. Adversarial attacks against ML classifiers are in some sense easier; they often require only black-box access to the classifier, rather than knowledge of a confidential list.

We suggest that the privacy loss for an induced collision is not as impactful as a "true" false positive – the sender could choose not to do so and avoid the detection with significantly higher probability. However, this does not extend to users who might unwittingly receive or forward adversarially-modified messages. This could, for instance, be used to plant matching material on a target device that would flag the images and potentially open up the target to future investigation. Beyond privacy, adversarially induced false positives could also be used by malicious actors, activists, or others to overwhelm the detection system or make it useless. Platforms should be prepared to decide how to detect whether particular users are sending adversarially-induced false positives, determine whether it is feasible to separate these from standard false positives, and potentially modify terms of service to prevent attempts at sending massive amounts of adversarially-created false positives.

E COMMON CRYPTOGRAPHIC TOOLS FOR CONTENT MODERATION UNDER E2EE

In this section we briefly describe the cryptographic tools frequently used by the detection mechanisms and provide references for further reading.

Private Set Intersection. The typical setting for Private Set Intersection (PSI) [125, 290–293] is for two parties Alice and Bob to hold secret sets A and B respectively. PSI allows either Alice, Bob, or both to learn the intersection $A \cap B$ without Bob learning $(A \setminus B)$ or Alice learning $(B \setminus A)$.

PSI is frequently used in exact or perceptual matching to find the intersection of the server's private list C with a client's message $\{m\}$. The latest PSI schemes are quite fast and PSI schemes specialized for membership testing have low communication complexity.

Searchable Encryption. Searchable Encryption (SE) [25, 40, 45, 82] is an umbrella term combining searchable symmetric encryption (SSE) [82] and Public-key Encryption with Keyword Search (PEKS) [25, 40]. The specifics of the scheme vary widely, but SE typically allows Alice to encrypt a list of documents L, where each document $D \in L$ is a list of words, in such a way that a designated keyholder Bob can perform a search over a ciphertext to identify or reveal documents $D \in L$ that contain Bob's word w. SE schemes typically have pre-specified leakage in the form of either *index leakage* (leaking information about L), *search pattern leakage* (which leaks information about w to Alice), or *access pattern leakage*, leaking information about the relationship between multiple queries w_1, w_2, and L.

The leakage inherent to searchable encryption schemes requires careful evaluation for each scheme to ensure it is compatible with server privacy and partial client privacy; there are known classes of attacks on searchable encryption [57, 172, 193].

Homomorphic and Functional Encryption. Both homomorphic encryption and functional encryption consider a "data owner" Alice, and an "evaluator" Bob. In both, Alice has the keys necessary to encrypt and decrypt ciphertexts, and Bob has a separate "evaluation key" that allows him to manipulate the ciphertext in specific ways, without (necessarily) learning Alice's underlying plaintext.

Homomorphic encryption [3, 127] allows Bob to perform limited computation on ciphertexts, without learning the result himself: Partially homomorphic encryption allows the addition of ciphertexts, that is, if c_1 and c_2 are ciphertexts for x_1 and x_2 respectively, then there is an addition protocol Add such that $\texttt{Add}(c_1, c_2)$ yields a ciphertext for $(x_1 + x_2)$. Somewhat homomorphic encryption has a similar protocol Mult for multiplication that may be used a limited number of times; fully homomorphic encryption allows unlimited use of Mult. Depending on the scheme, the evaluation key necessary to compute Add and Mult may be a "key" as we normally think of them, or it may be the case that ciphertexts can be added and multiplied without having any key at all.

Functional encryption [41] takes this idea a step further: the evaluation key e was generated from a specific secret value k and function f. Bob can use e to compute $f(x, k)$ with access only to e and a ciphertext of x. Among other uses, this allows functional encryption to emulate homomorphic encryption, but also allows specialized decryption (for example revealing to Bob whether $x = k$, and revealing no other information).

Homomorphic encryption (especially fully homomorphic encryption) and functional encryption tend to be slower operations (seconds rather than milliseconds) but are still practical for some settings.

[25] Reveal public but not private content
[26] Client MB separate from cloud MB gets processed info

Multi-party computation. Multi-party computation (MPC) [42, 105, 189] generically refers to any method that allows multiple parties $P_1, \ldots, P_N$ to compute a function output $f(x_1, \ldots, x_N)$ without learning anything aside from the output (in particular, without learning anything about the inputs $x_1, \ldots, x_N$. Sometimes it is used to refer to specific techniques such as secret-sharing [31, 85, 136], garbled circuits [379], or similar frameworks [91]; other times it can also refer to generic public-key and symmetric-key protocols run between at least two parties. In this work we use MPC to refer to any cryptographic protocol that is not one of the other specialized techniques described here.

Trusted Execution Environments. Trusted execution environments (TEEs) [75, 258, 386] like Intel Software Guard eXtensions (SGX) [243] provide an isolated encrypted area of memory known as an *enclave*, such that the data within that region cannot be accessed by other software running on that hardware, and SGX can attest that the correct software is running.

Mainly in corporate network monitoring, a common paradigm for the client or gateway forwards the decryption key for the encrypted channel directly to the SGX enclave—out of reach by the service provider itself—and all the desired network functions (e.g. traffic analysis or detection of exfiltrated secrets) take place within the enclave (see e.g. [78]). If the content moderation detection code is running within the enclave and all other information remains unaltered, this exactly meets our definition of partial client privacy (assuming one believes the guarantees of TEEs in general). TEEs can also enact server-private code, since the client can be denied access to read any encrypted rules.

TEEs have two major downsides: the first is that they require specialized hardware that may not be an option in most content moderation settings. The second, and more serious, is that TEEs have been heavily criticized for having privacy-crippling side channel attacks via timing, cache, energy, and speculative execution that are capable of recovering encryption keys [60, 61, 144, 229, 272, 363].

F CLIENT PRIVACY

In this section we elaborate on the privacy issues inherent to different settings of client privacy under different detection paradigms.

A significant part of the modern debate on child safety content moderation in E2EE concerns the definition of E2EE and to what extent its guarantees are or are not violated by various detection and response mechanisms. After analyzing the literature, we see several approaches with conflicting privacy guarantees:

Full client privacy. The most privacy-preserving approach is to perform the entire pipeline, detection *and response*, on the client's device, with no automated message sent to the server or a moderator. Any detection mechanism that preserves full client privacy, from matching to machine learning, avoids the problem of leaking false positives to the server. Many client-side E2EE spam filters meet this requirement as do many misinformation "tiplines" in WhatsApp [26, 192, 237, 245, 248], and Apple's nudity classifier for underage accounts in iMessage [18]. User reporting with message franking removes one part of the deniability guarantee (see Section 7.1) but the confidentiality of the message holds unless one of the ends of the message deliberately reveals it.

Full client privacy does not remove "slippery slope" questions of whether the scheme could be altered in the future; a small tweak to client-side code would, for most applications, allow the detection to be sent to the server instead. However, a similar (though more obvious) tweak would allow most E2EE applications to exfiltrate *all* user data to the server; we rely on a variety of technical and non-technical means to detect such a change (see Section 7).

If one is to perform content moderation in E2EE, this is the most privacy-preserving option.

Exact matching (partial client privacy). In exact matching, one party—often the server—has a list of problematic content (usually stored as cryptographic hashes). The server learns whether any of the client's content matches with the list exactly (see Section 5.1), often accomplished by using Private Set Intersection or other multi-party computation. In principle, the match could also be performed on the client side, however, the literature mostly contains works that achieve full client privacy in that setting.

The exact matching paradigm carves out an important exception to the E2EE confidentiality guarantee: it only holds against non-matches. However, this method avoids the tricky issue of false positives: Although false positives are theoretically possible using exact matching, common cryptographic hash functions would only expect to reach a collision with probability $2^{-128} \approx 10^{-38}$, meaning if 7.5 billion WhatsApp messages are sent per day, even for a list of a billion elements with distinct hashes, it would take longer than the age of the universe to reach a single false positive in expectation. This category is still vulnerable to the surveillance and slippery slope concerns described in Section 2.3, but it avoids the privacy issues inherent to schemes with a higher false positive rate.

Predicate/policy exact matching (partial client privacy). Some systems, especially seen in corporate monitoring and parental control, use Searchable Encryption to perform exact matching anywhere in a packet (e.g. it would find a match for "nana" in the word "banana"). The technologies used to achieve this vary in their leakage. Schemes based on order-preserving encryption have known attacks revealing message content and typically do not meet the standard confidentiality guarantee of E2EE even on non-matches. Other schemes, based on searchable encryption, have different specified leakage. These schemes must be examined for privacy leakage on a case-by-case basis.

The privacy issues present in these schemes typically do not involve false positives, but rather involve the cryptosystem itself.

Perceptual matching (partial client privacy). In these schemes, one party (typically the server) holds a list of *perceptual* hashes of problematic content. Similar to exact matching, the server and client perform a protocol to determine whether the perceptual hash of the client's content appears on the server's list of problematic content, and, in partially client private systems, the server learns the result of the match.

This setting begins to significantly degrade the privacy guarantees of E2EE: the false positive rates of modern perceptual hash functions are in the range from 10^{-3} [175] to 10^{-8} [21]. The cryptographic tools for these systems typically increase the false positive rate only a negligible amount (on par with the amount for exact matching); nearly all of the false positive rate arises from the perceptual hash itself. Unlike exact matching, this does begin to erode the privacy guarantees of E2EE severely: using the same number of 7.5 billion messages per day, this corresponds to between 4.5 million and 135 false positives per day.

Furthermore, in addition to the false positive problem, the surveillance problems remain. The false positive problem adds an additional difficulty: if a PHF-based matching system was to be deployed, we believe the approximate false positive rate should be disclosed as a matter of transparency to allow users to make informed choices on the privacy properties of the chat services they use. We also suggest research into means of verifying the aggregate detection rate in Section 7. In that section we also suggest methods for cryptographically (and non-cryptographically) addressing appeal and redress.

To our knowledge, no research has been done on the distribution of false positives, but naively we would expect the false positives to be unevenly distributed in the distribution of sent messages. We call for more research on perceptual hash functions, both to develop more accurate and precise PHFs and also to understand the distribution of false positives.

This setting was precisely the matter at issue in Apple's automated CSAM detector [33]. Weighing the tradeoff between the significant privacy loss of these systems, their surveillance risk, and the horrific acts of child abuse they aim to stop is a policy tradeoff that is informed, but not determined, by this analysis.

ML classification (partial client privacy). The accuracy of ML classification varies widely based on the context-specific task and the classifier itself. Classifiers for content moderation tasks like nudity detection, misinformation, and child enticement achieve accuracies between 70%-97% [39, 124, 165, 168, 242, 257, 260, 287] The common consensus seems to be that at least for now, machine learning approaches have higher false positive rates than perceptual hash functions for the most serious categories of problematic content like CSAM [223].

The privacy impacts on E2EE are extreme, potentially leaking one in every 10-100 benign messages to the server or moderator, potentially leaking hundreds of millions of false positives per day if deployed on the scale of WhatsApp. This remains true even if the classification is performed client-side on the plaintext of the sender or receiver's device. For ML classification to be compatible with E2EE, we strongly recommend that either full client privacy be used, or the that significant improvements be made in the classification methods.