How To Think About End-To-End Encryption and AI: Training, Processing, Disclosure, and Consent

Mallory Knodel[1], Andrés Fábrega[2], Daniella Ferrari[1], Jacob Leiken[1], Betty Li Hou[1], Derek Yen[1], Sam de Alfaro[1], Kyunghyun Cho[1], and Sunoo Park[1]

[1]New York University
[2]Cornell University

March 2025

Abstract

End-to-end encryption (E2EE) has become the gold standard for securing communications, bringing strong confidentiality and privacy guarantees to billions of users worldwide. However, the current push towards widespread integration of artificial intelligence (AI) models, including in E2EE systems, raises some serious security concerns.

This work performs a critical examination of the (in)compatibility of AI models and E2EE applications. We explore this on two fronts: (1) the integration of AI assistants within E2EE applications, and (2) the use of E2EE data for training AI models. We analyze the *security implications* of each, by (1) providing a new definitional framework and threat modeling approach for E2EE security ranging from theory to practice, including a new concept called *caveat features*, based on which (2) we offer a novel analysis to distinguish approaches to AI integration that are and are not compatible with E2EE. Then, we analyze *legal implications* of integrating AI models in E2EE applications, given how AI integration can undermine the confidentiality that E2EE promises. Finally, we offer a list of detailed *recommendations* based on our technical and legal analyses, including: technical design choices that must be prioritized to uphold E2EE security; how service providers must accurately represent E2EE security; and best practices for the default behavior of AI features and for requesting user consent. We hope this paper catalyzes an informed conversation on the tensions that arise between the brisk deployment of AI and the security offered by E2EE, and guides the responsible development of new AI features.

Contents

1 Introduction **2**

2 Background I: Encryption, AI, and Trusted Hardware **4**

3 Our Definitions and Taxonomy **19**

4 Security Implications of Integrating AI with E2EE **26**

5 Background II: Consent, Privacy, Users, and Areas of Law **32**

6 Legal Implications of Integrating AI with E2EE **35**

7 Recommendations **41**

8 Discussion **45**

9 Conclusion **48**

Related Shorter Resources: Blog-post summaries of this paper by some of the authors are available at the NYU DeTaIL Lab Blog [7] and Tech Policy Press [119]. For examples of others' commentary on our

work, see Matt Green's perspective in his blog A Few Thoughts on Cryptographic Engineering [96], and Stephen Thomas's succinct coverage on the NYU Center for Data Science blog [199].

1 Introduction

Widespread adoption of end-to-end encryption (E2EE)[1] has improved the confidentiality and integrity of data in various contexts, including messaging, video, and voice communication. Leading applications—such as iMessage [12], WhatsApp [212], and Signal [166]—have integrated E2EE into their systems by default, bringing strong confidentiality and privacy to billions of users' communications. This trend appears poised to expand further, with an increasing number of services committing to the adoption of E2EE in order to ensure "safer, more secure and private service" to users [81, 75].

In an era where so many forms of communication have moved online, widespread E2EE has become an essential underpinning for the security of communications, providing critical protections for journalists, dissidents, and activists, as well as the everyday family and social communications of billions. As underscored by independent experts from the UN Office of the High Commissioner on Human Rights, strong encryption is essential to fundamental human liberties including free expression, the right to hold opinions, the right to privacy, and the right to assembly [115]. Beyond individual privacy, the ubiquitous nature of E2EE offers *systemic* protections for particular at-risk users, preventing their activities from appearing exceptional: widespread deployment of E2EE has made privacy the norm. This normalization of secure communications, in turn, makes it harder to distinguish whether individuals are engaged in frivolous or sensitive activities, and deters the targeting of individuals merely for using privacy technologies, resulting in a critical "network effect" [100] that creates the conditions necessary for journalists, human rights defenders, activists, and others to communicate securely with less fear of surveillance and repercussions—protections that a special-purpose platform just for top-secret communication would not be able to provide. These "network effects" currently protect privacy and freedom of expression for at-risk groups and everyday users around the world.

The movement towards widespread E2EE communication stayed strong alongside the "big data" trends of recent decades: a recognition of the value of keeping private messaging data out of increasingly vast repositories of personal information, even as data was called the new oil by some [198]. Privacy technologies *gained* momentum as prominent examples of government and corporate intrusions on individual privacy came to light [97, 35, 94], and people became increasingly aware of their digital activity being tracked and profiled at large scale [25]. Those providing E2EE services were popularly seen as protectors of consumer privacy and freedom of speech [36]. Both corporate and government surveillance were key issues, but corporations often stepped into the "protector" role and deployed E2EE technologies that would permit nobody other than the users participating in a communication to access private communication content: a decision that made business sense at the time (e.g., [35]).

Following remarkable recent advances and an explosion of interest in large language models and generative artificial intelligence (AI) more broadly, however, we observe three trends that raise alarms for E2EE security.

First, the way people interact with AI models is changing. While they originally served as standalone applications, AI models are now increasingly incorporated into other everyday applications and throughout devices, including messaging applications, in the form of AI "assistants." Interacting with these assistants is often baked into the user experience by default, made readily available as part of the application client (e.g., within a messaging app). Such integration creates new systemic data flows at scale between previously separate systems, and accordingly raises security and privacy considerations not limited to E2EE.

Second, high-quality training data is becoming scarce. This has created a race between model developers, with tech companies under increasing pressure to tap any potential source of human-created content [206]. That which is publicly accessible online has already been harnessed, so privately held data is naturally the next resort, and indeed, companies have been quietly changing their terms of service to enable training AI models on more and more of the user data they hold [196]—a practice about which the U.S. Federal Trade Commission has raised concerns [4]. At the time of writing, E2EE messaging data on major existing platforms remains off limits for AI training, but companies may be incentivized to reconsider these policies in the near future. Apple and Samsung have very recently launched integrated AI features that can process E2EE content on their devices (see Section 2.5), and other companies may be considering similar options.

[1] We use "E2EE" to mean either "end-to-end encryption" or "end-to-end encrypted," depending on context.

Third, recent trends in AI appear to have shifted business incentives. Over the last decade, E2EE became a central part of the business model of online intermediaries. Business incentives to protect proprietary and user data were aligned with human rights protections. For example, in 2016, Apple and Facebook were lauded for their public resistance to government requests for access to E2EE data, including their refusal to comply with government requests to build systemic weaknesses into their encryption systems which would allow law enforcement access to messaging data [35]. However, in the last couple of years we have seen a shift towards prioritizing AI features as a component of modern applications, and a corresponding search for sources of data to power these, sometimes even at the cost of existing product features and profits [95, 125, 85]. At the same time, interestingly, research and reporting shows that consumers themselves can be much more skeptical of AI-powered products [117, 54].

Against the backdrop of these three concerning trends, the motivating question of our paper is as follows:

Is processing of E2EE content by integrated AI models compatible with end-to-end encryption?
(If so, to what extent and under what circumstances?)

E2EE is a context in which confidentiality is the core guarantee above all others—confidentiality is the essence of the product. Though E2EE may be relatively unusual in this respect, our inquiry in this paper resonates with themes in the broader debate about security, privacy, and trust in AI.

Summary of contributions. We systematically analyze the above question from both a technological and a legal perspective, combining the expertise of an multidisciplinary team across cryptography/security, AI/LLMs, and technology law. Our contributions and recommendations illustrate that technological solutions and legal frameworks must work in tandem in order to achieve effective and enforceable safeguards.

First, we present a new taxonomy and threat modeling of E2EE that combines qualitative system-level context with rigorous cryptographic definition. Our taxonomy spans stylized academic treatments to definitions that capture state-of-the-art deployments, and also specifies what we consider *not* to be E2EE. We specify where these definitions diverge, and note which aspects have been sources of community controversy. We identify key confidentiality and integrity properties E2EE offers for both individual and systemic privacy, essential properties that hold only when E2EE is used at scale or by default, and auxiliary properties that are widely associated with E2EE but not technically within the scope of formal definitions to date.

Second, we examine a range of technical configurations for AI processing of E2EE data, taking into consideration the state of the art in cryptography and privacy technologies, as well as the latest technical developments in AI (§4). We assess the capacity of each such technical configuration to uphold the guarantees of E2EE. We conclude that some configurations *cannot* uphold these guarantees, while some others can.

Third, we overview relevant areas of law, and provide an analysis of the circumstances under which E2EE service providers are likely to be able to offer AI features which use E2EE content, consistent with their legal obligations under current US and EU law (§§5–6). We highlight areas of legal uncertainty, and discuss pending legal (or quasi-legal) processes.

Finally, our analysis yields the four key recommendations below (§7; italic key terms defined in §3.1). Our analysis indicates that there would be serious risks—security risks and legal risks—involved in diverging from these recommendations and thus undermining the established operation of the current E2EE ecosystem.

1. **Training.** Using end-to-end encrypted content to train *shared AI models* is not compatible with E2EE.
2. **Processing.** Processing *E2EE content* for AI features (such as inference or training) may be compatible with end-to-end encryption only if the following recommendations are upheld:
   (a) Prioritize *endpoint-local* processing where possible.
   (b) No *third party* can see or use any *E2EE content*[2] without breaking encryption,[3]
   (c) A user's *E2EE content* is exclusively used to fulfill that user's requests.
3. **Disclosure.** Messaging providers should not make unqualified representations that they provide E2EE if the default for any conversation is that *E2EE content* is used (e.g., for AI inference or training) by any *third party*.

[2]Or any derivative of E2EE content (i.e., any non-trivial function of content). See Section 3.1 for full definitions.

[3]This may sound paradoxical; see Section 4.2 for detailed discussion of technical configurations that may achieve this condition.

4. **Opt-in consent.** AI assistant features, if offered in E2EE systems, should generally be off by default and only activated via opt-in consent. Obtaining meaningful consent is complex, and requires careful consideration including but not limited to: scope and granularity of opt-in/out, ease and clarity of opt-in/out, group consent, and management of consent over time.

Roadmap. In Section 2 (Background), we overview the properties of E2EE (§2.1), how AI assistants work (§2.2), how AI assistants can process E2EE data (§2.3), and some background on trusted hardware (§2.4). In Section 2.5, we discuss three examples of recently deployed AI assistants that can process data from E2EE environments. Section 3 provides definitions of key terms that we rely on throughout the paper (§3.1), including a taxonomy from "strict E2EE" to "not E2EE" (§3.2).

In Section 4 (Technical Implications), we analyze and present a range of technical configurations in which AI assistants might process E2EE data, and associated security/confidentiality implications. In Sections 5–6 (Legal Implications), we overview different areas of US and EU law and how they may constrain the use of E2EE data as input for AI models. In Section 7, we offer our detailed recommendations on how and when messaging data can be fed into AI models consistent with E2EE, taking into careful consideration both technical possiblities and limitations, and existing legal frameworks. Section 8 provides discussion and Section 9 concludes (and includes a detailed note on author contributions).

2 Background I: Encryption, AI, and Trusted Hardware

This section covers technical background: end-to-end encryption (Section 2.1), AI assistants (Section 2.2), how it is possible for AI assistants to process E2EE content (Section 2.3), and trusted hardware (Section 2.4). Section 2.5 gives a brief overview of three deployed systems in which AI assistants can process E2EE data: Apple Intelligence, Samsung's new AI-integrated devices, and the integration of MetaAI with WhatsApp.

For readers knowledgeable about E2EE, TEEs, and AI, we suggest skipping to Sections 2.4.2 and 2.5.

2.1 Background on End-to-End Encryption (E2EE)

Online messaging systems that allow communication between users (such as iMessage, WhatsApp, and Signal) are generally intermediaries to every communication. Although the user experience may make it seem like messages go directly from a sender S to a recipient R, in reality every message goes from S to the platform P and then from P to R. This means that, absent any additional guarantees, the platform has access to all communications of users on the platform.

End-to-end encryption (E2EE) is a standard of security for communication in which only the sender and an intended recipient can read communications between them. In particular, even the platform P cannot read these communications despite still acting as the intermediary for communication: that is, communications still go from S to P then P to R.

The key tool used in E2EE is *encryption*. Encryption is a method for a sender S to "scramble" their communications in a manner which is easy for their intended recipient R (and S) to unscramble, but practically impossible for anyone else. Using encryption, S will scramble (or *encrypt*) each message for R before sending it to platform P, which only handles messages in scrambled form (and does not have the ability to unscramble messages). When R receives a scrambled message from S via P, they are able to unscramble the message (or *decrypt* it) and read the original message that S wrote.

The additional security that E2EE provides is an important protection for the privacy of the communicating users, as well as a key protection against misuse of the message contents by or at the platform P: e.g., the messaging data is safely inaccessible if P suffers a data breach, if P is asked to share data with other organizations (e.g., business partners or the government), or if an insider at P wants to read the data.

We conclude this section with some terminology and follow it with a slightly more detailed and technically accurate explanation of how E2EE works.

Plaintext and Ciphertext. Message content that is unencrypted is described as *plaintext*. Normal, readable messages that you type into iMessage, Signal, or WhatsApp are plaintexts. Despite the name, plaintexts can include any type of content: audio files, images, spreadsheets, and any other digital information. Encryption

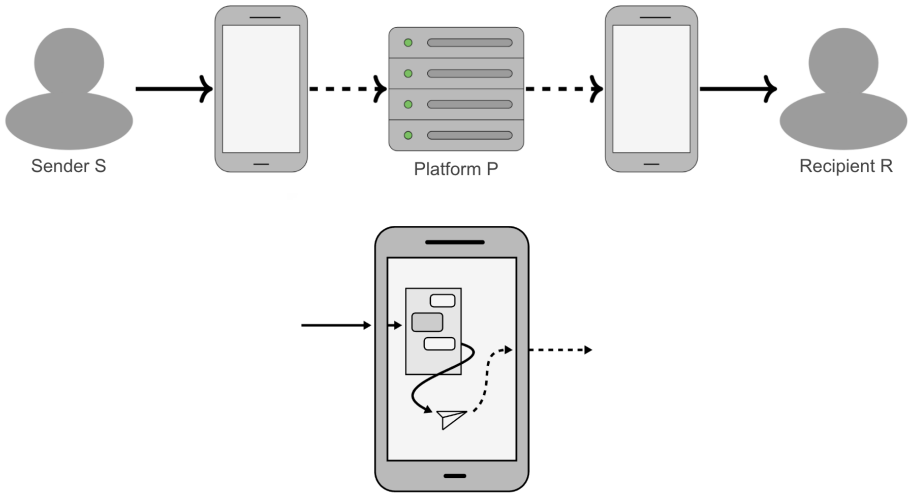

Figure 1: A sender S and receiver R communicate using an end-to-end encrypted application hosted by a company (middle). Solid lines represent plaintexts, and dashed lines represent ciphertexts. S and R can read their messages on their devices; however, while a message is "in transit" between their devices, it is encrypted so that it is not readable to the intermediary platform P handling it on its servers (or indeed to anyone but the intended recipient, whether a network eavesdropper, an employee at P, a hacker who compromises P, or a wrong recipient to whom the message was delivered by mistake).

provide a technique to convert (or scramble) plaintexts into *ciphertexts*, which are unreadable except to the designated recipient (and the sender who wrote it).[4]

Secret Key. Each user in an E2EE system has a *secret key*, a secret piece of information that should be known only to them (much like a password). A user's secret key is necessary to decrypt messages intended for them. Secret keys are created and stored on users' devices: platforms (and other users) do not have access to them.

Encryption and Decryption. The process of converting a plaintext into a ciphertext is called *encryption*, and the process of converting a ciphertext back into a plaintext is called *decryption*. Only the intended recipient R should be able to perform decryption to recover plaintexts from ciphertexts; accordingly, decryption requires the use of some secret information which only R knows, namely, R's *secret key*.

Putting it together. Equipped with more formal vocabulary, we restate the basic mechanism of E2EE. In E2EE messaging, each sending user S's device encrypts each plaintext message, then sends the resulting ciphertext to the platform P. The platform P passes the ciphertext on to the recipient user R's device, which uses its secret key to decrypt the ciphertext and get back the original plaintext message that S wrote.

2.1.1 The E2EE Confidentiality Guarantee

The core privacy guarantee of an E2EE system is called *confidentiality*.[5] Informally, E2EE confidentiality states that plaintext messages handled by an E2EE system can only be seen by the sender and intended recipients: no one else, including law enforcement or the service provider, can learn anything about the plaintext messages [120]. Importantly, this guarantee is stronger than saying that nobody else can *learn the plaintext messages*: E2EE confidentiality requires that nothing can be learned about the content of an E2EE message, including the message itself, but also any derivative of the message content.[6] Thus, E2EE

[4]The terms *plaintext* and *ciphertext* can each be used either as a noun or as an adjective.

[5]We will hereafter use the term "E2EE confidentiality" in this paper to stress when we are referring to this particular technical property, and not the common use of the word.

[6]The formal mathematical definition of security of encryption states that given the message length in bits, no computational adversary can guess any function of the message content with success non-negligibly better than random guessing.

confidentiality means that anyone besides the sender and intended recipient must not be able to learn, for example, what letter of the alphabet a message begins with, or whether the message contains a given word. See [99, 102, 120, 184] for more detailed definitions of end-to-end encryption.

E2EE confidentiality (or E2EE security) refers to a specific technical definition of confidentiality for messages, based on encryption. It is a term of art with a precise meaning, and does not ensure perfect "confidentiality," "privacy," or "security" for every colloquial meaning of the terms. In general use, these terms are very broad and mean different things in different contexts.[7]

2.1.2 Auxiliary Security Properties of E2EE Systems

Besides the core guarantees mentioned so far in this section, there are additional (cryptographic) security properties that E2EE applications often provide in practice. While distinct from basic E2EE confidentiality, these properties have become very commonly associated with, or expected of, modern E2EE applications. Notable examples include:

- *Integrity* [201]: Plaintext messages cannot be modified during transit in undetectable ways.
- *Forward secrecy* [98]: An adversary who has compromised the communication channel (e.g., by stealing the victims' secret session key) cannot recover messages that were transmitted before the session was compromised.
- *Post-compromise security* [61]: An adversary that has compromised the communicate channel cannot recover messages that are transmitted after the protocol "heals" itself (e.g., after session keys are rotated).
- *Deniability* [49]: No party can prove that a message was authored by any particular user of the system, even if the prover was one of the intended recipients of the message.
- *Unlinkability* [201]: Knowledge of the fact that a user authored a particular message reveals no information about whether they authored other messages.

State-of-the-art E2EE messaging protocols, such as the Signal protocol [166], meet these (and many additional) security properties.

2.1.3 Systemic Properties of E2EE Systems

Even if an E2EE application provides strong technical privacy guarantees (including and beyond E2EE confidentiality), the strength of the privacy guarantee it offers will also depend on its usage in practice. Messaging applications that are widely used offer additional protection because their use is *unremarkable*, and hence, the use of the application in itself does not draw attention. In economic terms, this is a *network effect* where the (privacy-related) utility of the application for any given user depends upon other users.

For example, suppose that a particular E2EE messaging app A is known to have a userbase which disproportionately represents a particular political community. Even if A provides perfect E2EE confidentiality, knowing that a particular person uses A can divulge their political views, if only by association. This type of example motivates the intuition that to realize the full privacy potential that E2EE can provide, the application should be commonplace and its use should be unremarkable in the relevant context. This "unremarkableness" is not a technical property: it is instead an emergent social and contextual property of the system's use as a whole considered over the entire userbase (relevant to a context of use[8]). Existing security and privacy literature has discussed similar network effects in privacy technologies not limited to encrypted messaging, and noted their importance to engineering effective privacy technologies (e.g., Tor) [80]. This key privacy property of E2EE systems, which cannot be achieved by technical means alone, *is* attained by popular E2EE applications such as iMessage and WhatsApp today—namely, the network effect gained from wide usage such that usage in itself is unremarkable.

[7] For example, E2EE technologies do not prevent your friend from choosing to disclose the sensitive information you shared with them to others, even though this might seem to "violate confidentiality" or "violate your privacy" in a broad sense. Indeed, no technology can prevent this. See also Section 5 for more discussion.

[8] E.g, widespread use in country X may not be helpful for a user in country Y.

In summary, there is a key privacy property of E2EE systems which cannot be achieved by technical means alone, but *is* attained by popular E2EE applications such as iMessage and WhatsApp today—namely, the network effect gained from wide usage, so that the use of the system in itself is unremarkable.

2.1.4 Necessary Conditions for E2EE Confidentiality

End-to-end encryption provides its promised confidentiality guarantee only under certain conditions: namely, when (A) the underlying encryption method is secure and (B) that encryption method is securely implemented in the application being used.

Cryptographic assumptions. Every practical encryption scheme relies upon a *hardness assumption*: an assumption that a particular mathematical problem is difficult for computers to solve. The hardness of the mathematical problems underlying today's widely used encryption schemes has held strong in the face of heavy scrutiny over decades by mathematicians, computer scientists, and others, leading to a strong community belief that it is acceptable to use cryptography that relies on these assumptions in security-critical contexts. When new encryption schemes that rely on new, less-tested assumptions are proposed, they are not considered suitable for deployment until they have been vetted in a multi-year process of scrutiny by mathematicians and computer scientists worldwide.

Practical assumptions. Even if the math underlying an encryption scheme is perfect in theory, the process of deploying the encryption scheme in software or hardware introduces avenues for security failures, and the use of the encryption scheme by fallible humans in practice introduces further potential for security failures. In order for E2EE confidentiality to hold: (1) the code implementing the E2EE application must be correct, (2) the hardware device that the application runs on must be secure, and (3) the users using the application must use it correctly and take adequate security precautions.[9] Undermining any one of these three could undermine E2EE confidentiality: for example, (1) the software fails to perform the encryption correctly; (2) the hardware device is infected with malware; or (3) a user fails to keep their secret key secret. Errors can be introduced in myriad ways: through simple mischance, human mistake, or intentional sabotage.

Typically, in current practice, a messaging application is implemented by an intermediary platform that provides the associated messaging service. Thus, the party with the most control over and visibility into the implementation process is the platform. Users therefore rely on platforms for (1), i.e., that the code implements E2EE correctly. Users rely on device manufacturers (and their suppliers) for (2), i.e., that the device running the app is secure. Finally, the users themselves determine (3), i.e., their usage of the application; that said, users often lack the expertise to confidently take adequate security precautions.

2.1.5 E2EE Messaging Features and Usability

Many academic definitions of E2EE messaging consider a simple model where the only activity that occurs is literal messaging—sending messages between senders and receivers and within group chats. However, modern messaging apps are more feature-rich, and the ways in which such additional features process message contents—sometimes in tension with E2EE confidentiality—has long been a subject of debate. Some common features are, strictly speaking, in violation of E2EE confidentiality: for example, *link previewing*, where the app displays a "preview" of a link contained in a message, technically violates E2EE confidentiality because a third party processes the link in order to fetch the preview [28]. Furthermore, some common features of E2EE applications *enhance* confidentiality despite not being part of the basic definition of E2EE: for example, *user blocking* helps protect users from harassment and spam, but has no direct relationship to E2EE confidentiality (or any of the other cryptographic security properties mentioned in Section 2.1.2).[10]

In short, any comprehensive treatment of E2EE messaging applications must consider the interaction of commonplace messaging features with the system. As discussed in Section 2.1.3, an E2EE application requires a sufficiently large and diverse user base to promise privacy in practice, and it is therefore pragmatically important for E2EE messaging apps to incorporate common usability features of messaging apps if doing so could make or break users' willingness to choose the app. Of course, this is not an argument for a kitchen-sink approach to messaging app development: some features may be too incompatible with E2EE, or

[9] In fact, these are essential considerations for the proper functioning of any software product, not limited to encryption.

[10] Other common features with potential consequences for confidentiality and E2EE confidentiality under some implementations include: importing contacts, autocorrect, and accessibility features such as text-to-voice.

can at least be made more palatable by choosing certain implementations over others. For example, Signal has link previewing, but also offers users the ability to disable link previewing in the settings.

This complicated relationship between messaging features, usability, and privacy in E2EE apps has led to practical controversy as well as academic debate. For example, Apple offers a feature where users can store encrypted backups of their iPhone data on Apple's cloud storage—a standard convenience. In February 2025, reports emerged that the UK government has ordered Apple to grant it backdoor access to these encrypted backups [170]. In this case, Apple providing (encrypted) backups—providing utility to the end user—also created a new resource for third parties (here, law enforcement) to target to gain access to end-to-end encrypted user communications, potentially without users' knowledge and without breaking encryption.

While it is out of our scope to fully characterize the wide spectrum of messaging app features and their privacy implications, it is useful to view AI assistants in the context of a broader constellation of other privacy-impacting messaging app features which have been met with controversy and debate.

2.2 Background on AI Assistants

AI assistants are applications that are designed to interpret everyday language from users and perform computational tasks. Today's AI assistants, such as ChatGPT, Gemini, Claude, DeepSeek, and Meta AI, can handle a wide range of tasks, including text analysis, content creation, code generation, and language translation. At their core, these systems rely on statistical models—programs trained on data to identify patterns. Most modern AI assistants are built using large language models (LLMs), which are trained on vast datasets of text to generate outputs based on predictions of the most likely next word. Due to the size of their training datasets and sophisticated architectures, LLMs can handle complex, previously unseen user inputs (*queries*) and provide contextually relevant responses.

2.2.1 Training and Inference

AI assistants undergo two key phases: *training*, when the model is built to identify patterns, relationships, and rules; and *inference*, when real-time responses are generated based on user inputs. Training itself often involves multiple stages, typically referred to as *pretraining* and *fine-tuning*. In pretraining, the model takes in large-scale general datasets to learn structures and patterns in the data. Fine-tuning then refines this pretrained model by training it on additional, more specialized datasets, improving its accuracy and relevance in particular contexts. We describe these processes in further detail below.

Training. Models have configurable *parameters* which are numerical values that determine how outputs are generated. These parameters are used in calculations to predict outputs when the model is queried with an input. During training, the model processes large datasets—such as books, websites, code repositories, and conversational datasets in the case of LLMs—which iteratively adjusts (or "tune") the parameters, capturing more and more refined patterns in the data with repeated passes. The finalized set of parameters constitutes the trained model, which can then be *prompted* with inputs (e.g., questions) to generate outputs (e.g., answers). Models at the scale of commercial AI assistants contain billions or even trillions of parameters, trained over the course of months on high-performance computers.

Inference. Once trained, the model can be used for inference, where it outputs real-time responses to user inputs. When a query is inputted to a model, it is processed with the model parameters to calculate an output. While inference is less resource-intensive than training, it still requires significant computational power and is typically handled by centralized servers.

Deployed models undergo continuous improvement through updates that often involve additional training on more diverse or recent datasets. Some approaches for this include reinforcement learning with human feedback, where collected user feedback is used to improve response quality [160]; patches to address issues such as specific problematic outputs, and model refinement with new techniques and architectures. As such, inference and training often overlap in a *feedback loop*, where data collected during inference is fed back into the training pipeline to fine-tune the model further. This process is shown in Figure 2.

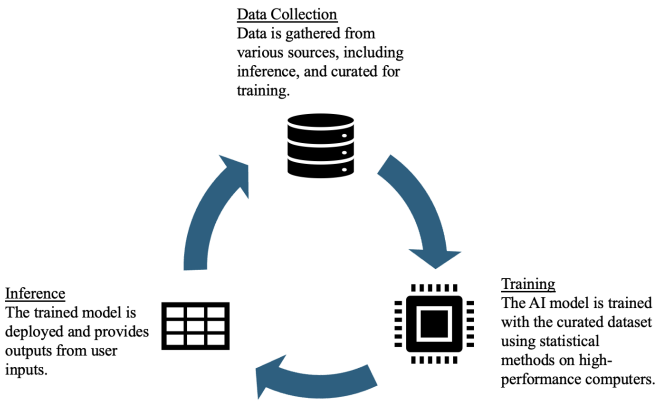

Figure 2: Inference and training of AI assistants happen continuously as a feedback loop. As users continue to query the AI assistant, this generates data that is stored and can be used to further train and improve the model.

2.2.2 Data Collection

Once the AI assistant is deployed, various types of data from the user interactions with the assistant may be collected. This can include *chat logs*, which record user inputs (text, voice, or image data) and the system's responses. These historical records, even if kept by the platform, may or may not be visible to users.[11] Additionally, *metadata*, such as timestamps, device type, language, and geographic region, as well as user preferences and settings, may also be collected and stored by the platform. This data serves multiple purposes, which can be broadly categorized as follows.

1. Ease of use: The data is used to enable certain features, such as maintaining conversation histories or continuity within a session/across sessions, for an individual user.

2. Personalization: The data is used to identify usage patterns, such as query frequency and commonly accessed features, to tailor the system's functionality for an individual user.

3. Analytics: The data and usage patterns is aggregated across users to steer further improvements and developments of the model.

4. Model refinement: The data is used as training data to fine-tune the model and continuously improve its performance.

Collecting user data is commonly enabled by default with collected user data transmitted and stored on centralized cloud servers. In some cases, some or all user data may remain local on the user's device, eliminating external transmission.

2.2.3 Architecture of AI Assistants in Messaging Platforms

Next, we overview the architecture of AI assistants integrated in messaging platforms. The architecture is how the application is structured technically, which has implications on how and where data is processed. First, it is important to distinguish between the messaging application, the AI assistant, and the model. The messaging application is the application installed on the user's device which facilitates communication between users and provides the interface for interacting with the AI assistant, e.g., WhatsApp, Messenger, and Signal. The AI assistant is the system that performs tasks such as answering queries, generating responses, or providing recommendations, e.g., Meta AI, which is built into WhatsApp and Messenger. At

[11] Search/query history can be surprisingly revealing when aggregated over time, involving family concerns, health issues, religious beliefs, and more. Research has shown that users can be reluctant to disclose all their search history even for better personalized search results [161].

its core is the model—in this case, an LLM—the underlying computational engine that processes inputs and generates outputs based on patterns learned during training.

When the messaging application gets an input from the user, this input is sent to the model in the form of a *request*. Once the request is processed and an output has been generated by the model, the output is sent back to the platform as a *response*. As such, while the messaging application itself will be installed on the user's device, the model need not also be directly on the device; network connections can facilitate this communication between the application and model.

We differentiate between two types of architectures: *cloud-based* and *local*, illustrated in Figure 3.

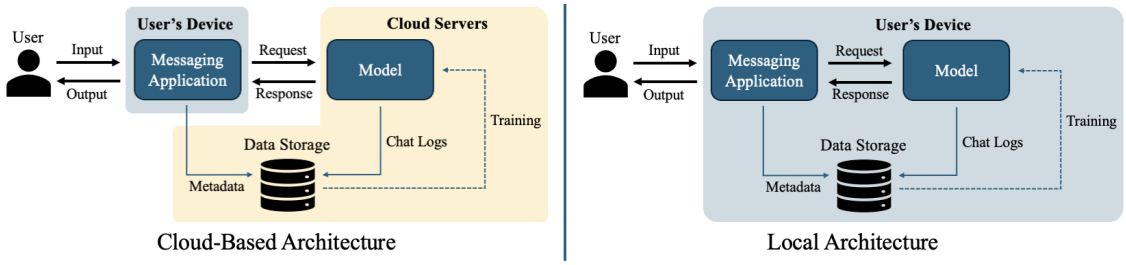

Figure 3: Generating an output from a user's input, differentiating between what is on-device and on cloud servers between a cloud-based and local architecture.

Cloud-Based. Cloud servers are designed to handle large computation and storage demands. Given the demands of an LLM—recall that there may be trillions of parameters to be processed simultaneously, requiring very high levels of computation power as well as memory to store the parameters and data—typically, models are hosted on cloud servers. Cloud servers can carry thousands of high-performance computers (Graphics Processing Units or Tensor Processing Units), exceeding the capabilities of a typical laptop by many magnitudes.

Local. In limited scenarios, computations can occur *locally*—that is, directly on the user's device using lightweight versions of the model or specific components. Local processing ensures that data does not leave the device, but is only possible with smaller, less powerful models due to reduced processing power and storage capacity of local hardware.

2.2.4 Standalone vs. Integrated AI Features in E2EE Applications

AI assistants are commonly available in major E2EE applications, as a standalone tool *separate* from user conversations (and any other E2EE content). For example, an E2EE messaging application might also include a search bar where users can type in queries to search the web and/or send to AI models.

In general, we do not consider such standalone tools to be *integrated* with E2EE applications in the way we are concerned about in this paper. Standalone AI features separated from E2EE content generally do not raise the security concerns that we argue *do* arise with integrated AI features, unless providers create the impression that these AI features are themselves E2EE. We consider standalone features out of our scope.

Our focus is, rather, on integrated AI models and their capacity to process E2EE content. Integrating AI assistants as a feature within E2EE conversations would often require the AI assistant to process E2EE content: for example, to allow users to invoke the AI assistant within an E2EE conversation or to provide contextually relevant support, responses, or summaries within E2EE conversations.

2.2.5 Tool Usage

Tool usage is increasingly built into deployed proprietary AI assistants, extending language models beyond text generation by enabling calls to resources such as search engines, calculators, databases, code interpreters, and application APIs. This serves to address several limitations of standalone language models, including lack of access to up-to-date information [53] and difficulty with multi-step computational tasks [84].

When a user makes a request, the assistant determines whether a tool is needed, selects an appropriate tool, generates a structured call, receives the tool's output, and incorporates the result into the response

[185]. This process may be hidden from the user, in which case the system can invoke tools internally and outputs only the final synthesized response to the user. For example, Anthropic's Claude documentation states that Claude can call developer-defined or Anthropic-provided tools, and that Claude decides when to call a tool based on the user request and tool description [8].

It is not understood how to have AI assistants access both tool use and private data while reliably preventing exfiltration of private data to external services, and numerous examples demonstrate such exfiltration in the wild [218].

2.3 How Is It Technically Possible for E2EE Data to Be Fed Into AI Assistants?

The definition of E2EE may make it sound like it should be impossible to feed E2EE content into a (cloud-based) AI assistant. How can an AI model have access to plaintext data, when E2EE guarantees that only the sender and recipient can access E2EE content in plaintext?

Certainly, the cloud-based service provider has access to the encrypted message—the ciphertext—and could input that ciphertext to an AI assistant. In this case, however, the AI assistant would not be able to read any meaningful information from the ciphertext, and thus would not be able to produce a useful answer, either.[12]

The current discourse around using E2EE data in AI assistants, therefore, is focused primarily on the idea of feeding readable *plaintext* versions of E2EE messaging data into AI assistants. Since only the sender and recipient devices in an E2EE system can read the plaintext messages, the sender or recipient devices would be the only ones who could provide the plaintext data to the AI assistants. This possibility appears to be primary focus of the ongoing discourse: that users' devices would be set up to send plaintext copies of E2EE content to cloud-based AI assistants for processing.

Nothing about E2EE technically prevents this from happening—indeed, no technical measure would both be able to allow senders and recipients to read their own messages, and prevent them from showing them to anyone else. This is a bit like no matter how secretively you encode and deliver your hand-written letter personally to your friend, nothing you can do can stop your friend from showing the decoded letter to other people once they have it. (This analogy falls short after a point. In the letter-writing example, your friend would likely have to take an affirmative, intentional action to show the letter to others; but in the encrypted messaging case, users' devices and applications can be set up by service providers to *automatically* send plaintext copies of messages to third parties without any action being taken by users.)

But then, what is the point of end-to-end encryption?. We have just explained that E2EE is a technology that allows transmission of messages in encrypted form so that nobody but the sender and intended recipient can read them, not even intermediary platforms. We have also explained that E2EE does not stop the sender and recipient devices from sending plaintext copies of the messages to anybody — and that devices can do this in a way that is very hard to detect for average users.

Then, *is there any point to using E2EE?* After all, our devices could be routinely sending plaintext copies of all our messages *to the messaging platforms themselves*, and E2EE would not stop this — yet such a practice would undermine precisely the benefit that E2EE purports to provide.

E2EE *does* still have a point, for a number of reasons. First, it provides important protections when service providers *choose* not to be able to read your messages, and implement E2EE in order to achieve this goal. Such a choice might be motivated by various reasons: it might provide a competitive edge, provide protection in the event of data breaches, or reduce compliance costs for companies. Once such a choice is made, the platform moreover has a reputational interest in making sure E2EE is achieving the stated goal that the platform not be able to read messages. Second, E2EE provides stronger protections in the case of popular messaging platforms that are widely used and therefore widely scrutinized by technical experts, including security and privacy experts. While subversions like sending plaintext copies of messages to the platform would be hard to detect for an average user, they would be much more easily detected under such scrutiny by experts — and from the experts' scrutiny, average users benefit too. Third, E2EE provides even stronger protections in the case of popular messaging platforms that choose to openly disclose the details

[12] This is mathematically guaranteed by the formal definition of secure encryption.

of how their system and security features work, for similar reasons. Balsa et al. [30] provide more detailed discussion of these and related issues around platforms implementing E2EE and other cryptography.

2.4 Background on Trusted Execution Environments (TEEs)

Trusted hardware refers to physical devices that are designed to ensure the confidentiality and integrity of computations, even if the owner of the hardware may be compromised. Trusted hardware is designed to create a secure area of computation, in which no information about the computation is released outside of the trusted hardware during execution of code, and that code will run exactly as specified. There are many types of technologies that fall under the broad umbrella of trusted hardware—such as hardware security modules (HSMs), trusted platform modules (TPMs), and trusted execution environments (TEEs)—which offer different capabilities. Of these technologies, TEEs are the most relevant for the implementation of AI functionalities, as they are designed to run arbitrary programs; other technologies currently have more limited, bespoke use cases, such as key generation and storage. We thus focus on TEEs from here on.

We can conceptualize a TEE as a physical "lockbox", which is designed to execute arbitrary code such that no outside entities (including, for example, the operating system, other device processes, and third parties) can observe nor manipulate the state of the code as it is running. During initial setup of the TEE, a computer program—call it Π—is loaded inside the TEE. The TEE then has a single input/output interface, through it which receives inputs x from outside the TEE, computes the program Π on the inputs, and outputs the result $\Pi(x)$ from the TEE. During execution of Π, the TEE maintains some internal ephemeral state in memory, which stores temporary values during computation. TEEs are also often equipped with persistent storage, on which they can store data longer-term. See [164] for a formal treatment of TEE abstractions.

TEE security goals. TEEs are intended to provide two core security guarantees. The first is *confidentiality*, which means that the internal state of the TEE should not be visible to any parties during execution of Π; only x and $\Pi(x)$ are revealed to third parties. Similarly, the persistent storage of the TEE should always be kept secret. The second goal of TEEs is *integrity*, which means that Π should be the exact program run by the TEE, which cannot be undetectably tampered with nor replaced by another program. Besides these two core guarantees, some TEE implementations have a third security goal, *remote attestation*, which means that the TEE should be able provide convincing evidence that a given program Π is the code that is loaded inside the TEE. Critically, these guarantees should hold even in the presence of a compromised TEE owner (e.g., a cloud provider): using a number of hardware-level protections, TEEs are designed to be tamper-resistant (for example, TEE chips often have physical seals that are designed to detect attempts to open them, upon which the TEE's storage is immediately erased).

The goal of TEE design is to enable outsourcing of computations over user data to potentially untrusted (but more computationally powerful) environments, by loading the program inside a server-side TEE, and processing user data inside of it. As described so far, the inputs to and outputs from the TEE can be seen by the owner of the TEE. However, inputs and outputs can be hidden from the TEE owner with an additional step: encrypt inputs and outputs so that they can only be decrypted using secret key that is stored inside the TEE. Then, computations inside the TEE can still happen in plaintext, within the "shield" of the hardware protections provided by the TEE. Figure 4 shows examples of architecture designs that use TEEs.

TEE performance. Modern TEEs (such as Intel SGX [73] and AMD SEV [114]) are capable of supporting complex computations with practical overheads. For example, TEE-based security has been deployed in Signal's contact discovery mechanism [133], and even entire blockchains [152].

A server-side TEE in practice is often deployed on shared infrastructure to serve multiple users,, e.g. Meta AI secure enclaves [212]. Modern TEEs (e.g., Intel SGX [73] and AMD SEV [114]) can support complex computations with practical overheads. But efficiency is not the only engineering consideration for TEEs: their design of strict isolation imposes limitations on the programs that TEEs can run. In particular, a TEE with strict isolation cannot natively support networking hardware, and thus cannot make direct internet requests. In practice, however, TEEs can establish TLS sessions with a remote party by using an intermediary service (e.g., the operating system) to relay messages between itself and the external party [190] — TLS encrypted, but not end-to-end encrypted. As such, this allows the TEE to outsource information or access external state in such a way that the intermediary service (e.g., the OS) and external parties can

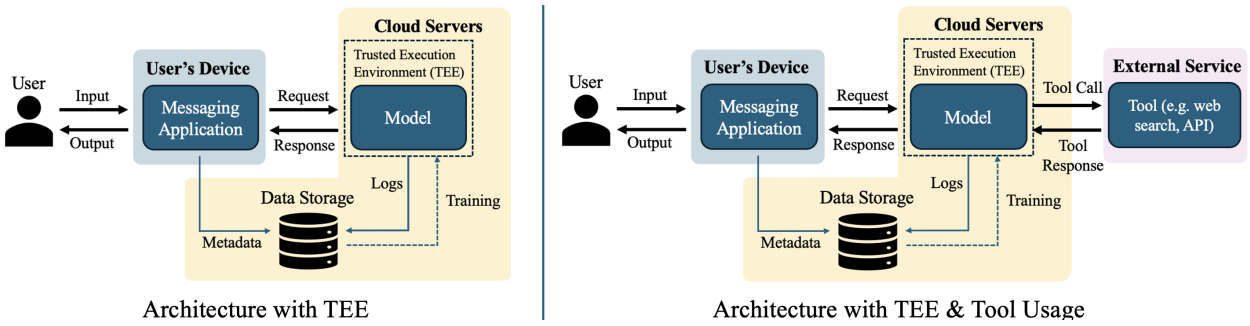

Figure 4: Architectures with TEEs. The left shows a cloud-based architecture in which the model is physically hosted within a TEE. The right shows tool usage with a TEE, where the model makes calls and receives responses from an external service. The messaging application, model, TEE, and external tool may each be operated by the same provider or distributed across different providers.

access plaintext content.[13]

2.4.1 Necessary conditions for TEE security

The security guarantees of TEEs naturally rely on the correct implementation of the TEE, and that the hardware has not been tampered with at any point in the TEE's life cycle (e.g., tamper-proof seals have not been removed, backdoors have not been inserted, etc). This requires trust in various actors across the hardware supply chain, from manufacturing, to assembly, to deployment. Further, users typically do not handle TEEs directly; rather, they are embedded in devices or handled server-side. As such, users rely on the application or device provider that a TEE is in fact being used for any given application.. While some TEEs have additional mechanisms to attest to their usage for certain computations (e.g., via their attestation service), this generally needs to leverage some trusted authority, such as the hardware provider.

Even if the hardware is correctly implemented according to specification, the security of TEEs additionally relies on the specification itself being free of vulnerabilities. Unfortunately, there is a history of practical attacks on various TEE implementations [88, 205, 187]. These attacks range across a variety of threat models—including adversaries that have physical access to the TEE, can launch co-located TEEs with arbitrary programs, control the system's boot process, etc.—and violate the confidentiality and integrity of the TEE, i.e., recover sensitive runtime state or tamper with the data in the TEE. Some of the attacks have been mitigated since their discovery, while others have not [205]. Furthermore, some attacks may not be fully patchable with known techniques, e.g., [110]. Notable examples of attack vectors include physical and micro-architectural side-channels, which exploit shared caches [45], branch predictors [122], page faults [220], and transient execution [203].

2.4.2 TEEs vs E2EE

While TEEs and encryption have similar security goals, there are some critical differences between the two technologies, which we review in this section. We first explain why they are not interchangeable security solutions, and then why TEEs introduce additional security concerns not present in the encryption context.

TEEs and encryption provide different security guarantees. E2EE security and TEE security each rely on a number of conditions holding true, as summarized in Section 2.4.1 just above (for TEEs) and in Section 2.1.4 (for E2EE). The practical guarantees provided by each technology depend on both the precise security goals they are designed to achieve, and the conditions (or the *threat model*) under which they are designed to achieve them. While TEEs and E2EE have overlapping security goals, the conditions necessary for E2EE security and for TEE security are *orthogonal*—qualitatively different and incomparable.

The security of E2EE and TEEs rely on two broad assumption types: (1) specification correctness, and (2) implementation correctness. The former states that the design of the technology/protocol is secure, while

[13]This capability is generally not modelled explicitly in existing TEE formalisms [164].

the latter states that a particular instance of the technology/protocol is correctly implemented according to the specification. TEEs and E2EE have important differences for both assumption types. Assumption type (1) generally hinges on a computational hardness assumption for E2EE; and on a robust hardware design for TEEs. On the other hand, assumption type (2) hinges on the security of software and hardware implementations for E2EE and TEEs, respectively. These entail a completely different set of considerations between E2EE and TEEs, since the software and hardware supply chains involve different entities, vendors, and processes.

Since the security of TEEs and E2EE depend on orthogonal considerations, their resulting security guarantees are *incomparable* in a technical sense. That is, the precise type of security you get, and the conditions under which you get it, are different and not directly comparable. TEEs rely upon a distinct set of conditions for their security from E2EE, and therefore cannot be substituted into an E2EE protocol without acknowledging that the nature of the security offered has changed, away from E2EE security. Thus, again, while the security goals are similar, they are not interchangeable technologies.

TEEs raise certain security concerns that encryption does not. The difference in the type of security offered by TEEs and by encryption mean that depending on the context, one may be more suitable for use than the other. That said, for an application which might admit either, TEEs raise certain kinds of security concerns not present for encryption. Widely used encryption schemes today have largely withstood decades of scrutiny without critical confidentiality breaks; indeed, new encryption schemes that have not yet been subject to such scrutiny are widely considered unsuitable for use in security applications. While TEEs may enable secure computation in promising new applications which would not be practical using cryptography alone, our understanding and scrutiny of TEE security is comparatively young, and the field has seen a stream of discoveries of security flaws with impacts on deployed systems (e.g. [88, 205, 187]).

There are also *differences in the types of attacks* that current encryption and TEEs can protect against. The most notable example is *post-quantum (PQ) security*, i.e., resistance against attacks performed with quantum computers. State-of-the-art E2EE protocols already offer post-quantum security [121, 70], and thus protect against "harvest-now-decrypt-later" attacks [37]. TEEs, on the other hand, are not yet resilient to PQ attacks [202]. In applications requiring PQ security, TEEs alone would not be suitable.

Finally, TEEs permit disclosure of the data that they protect, in ways that E2EE does not, due to the fact that TEEs are able to establish network connections with external services. In particular, TEEs can interact with external services in ways that *depend on the information that is being processed*, and thereby leak plaintext content (or functions thereof) to third parties. (Even the mere act of establishing a connection with an external service itself can be revealing, as in the case of link previews, discussed earlier.) Such disclosure is (provably) not possible when content is encrypted.

2.5 Deployments in Practice: Apple, Samsung, and Meta AI

In this section, we overview three deployed examples of AI assistants that may process E2EE data: Apple Intelligence, Samsung's Galaxy AI, and Meta AI in WhatsApp. Appendix B applies our recommendations to the case of Apple Intelligence; we use Apple as an illustrative example application as we found that the documentation currently available on Apple Intelligence supports a more detailed analysis than for the other two products. We leave the other two examples—as well as future releases of these and other products—as an exercise for the reader.

The information in this section is based on publicly available documentation from the relevant vendors, not direct testing of the products. We list key points that the currently available documentation leaves unclear, under the heading "What We Still Don't Know" at the end of each subsection. These lists are not comprehensive, but highlight notable points of ambiguity that hinder a complete evaluation of the E2EE compatibility of these AI integrations.

These products continue to evolve, so while our case-specific description and analysis remain illustrative, some details may soon be out of date. Readers seeking to understand the implications of our work for the latest technologies should check recent product releases and may conduct their own analysis based on our generalizable analytical framework and recommendations.

2.5.1 Apple Intelligence

Apple introduced a new product called Apple Intelligence in late 2024, which provides new OS-integrated AI capabilities across the Apple ecosystem.[14] These AI-powered features are capable of reaching out to third-party servers, such as to perform Google searches [14]. Apple Intelligence is able to process and perform AI tasks on E2EE content [18]. According to its online documentation, Apple's plan does not involve *training* any models on any "users' private personal data or user interactions," which presumably excludes E2EE content from use for training [19].

What data on Apple devices is E2EE?. Applications across the iOS ecosystem differ on the confidentiality guarantees they offer. Our work focuses only on data that is protected with E2EE, so in this section, we explain what data in Apple devices is E2EE. There are two broad classes of applications for Apple devices: native applications, which are managed by Apple and come built-in with the device; and third-party applications, which are downloaded from the App Store. A few native applications are E2EE by default, such as iMessage, Facetime, and Keychain. However, storage on iCloud is not E2EE by default, so most native applications that rely on cloud storage are not E2EE. For additional security, users have the option of enabling *Advanced Data Protection (ADP)* [16], a feature that enables E2EE for the majority of iCloud data, such as photos, notes, voice memos, etc. Some iCloud data is not E2EE even when ADP is activated, such as mail, contacts, and calendars.[15] In this section, we refer to native applications for which ADP enables E2EE as *ADP-E2EE applications*. In addition, ADP enables E2EE for certain Siri features, such as Siri shortcuts and other information (e.g., settings and personalization). It is not clear to us whether Siri transcripts are E2EE or not, even with ADP enabled.

The confidentiality guarantees of third-party applications[16] on Apple devices vary by application. Some third-party applications are E2EE, such as Signal [166], and WhatsApp [212].

What data can Siri and Apple Intelligence access?. Siri and Apple Intelligence will be able to process users' content drawn from any application where the application developer has "integrated" Siri and Apple Intelligence functionalities [17], *and* the user has not deactivated this integration in their settings [21]. That is, if an application supports the integration of Siri and Apple Intelligence, then integration/access is on by default unless a user turns it off in their settings.

Possible use cases involving E2EE data. We highlight two key contexts in which E2EE data is used for inference in Apple Intelligence. First, Apple Intelligence may be used to process user content produced inside an E2EE application. For example, long sequences of messages in iMessage chats may be processed by AI to produce a short "summary" (say, when a user has been away from a group chat, to help the user catch up) [18]. Using Apple Intelligence or Siri inside any third-party E2EE or (if users have ADP enabled) any ADP-E2EE application could involve interaction with E2EE data.

The second way in which Apple Intelligence may process E2EE data is through its content generation features, which may be accessible from within E2EE applications—this includes third-party E2EE and ADP-E2EE applications if ADP is enabled. For instance, image and emoji generation can be based on user-uploaded images and accessed through the iMessage app. The generated images and emojis may be based on E2EE content if ADP is enabled.

Three-Tiered Approach. As users interact with Apple applications on their iPhones, iPads, or Macs, these applications may submit queries to the Apple Intelligence system. These queries, often a function of user data—and sometimes a function of user E2EE data—are then processed, and responses are sent back to the user (e.g., producing short summaries of long text or updating a contact's information).

Apple employs a three-tiered approach to handling user queries: (I) *local models* which process simple queries entirely on-device; (II) a *server model*, which processes more complex queries in a cloud-based trusted execution environment hosted by Apple; and (III) sending data to a *third-party model*, OpenAI's ChatGPT, which processes outsourced queries that could benefit from "ChatGPT's broad world knowledge" [18]. Each time a query is sent to ChatGPT, user consent is requested.[17]

[14] Apple Intelligence's initial release was in October 2024; the ChatGPT integration was launched in December 2024 [18, 11].

[15] We refer readers to [16] for a complete description of the confidentiality guarantees for various data types and native applications in iOS, both in the default case and when ADP is activated.

[16] I.e., applications that run on Apple devices but are not developed by Apple.

[17] Our understanding is that permission prompts can be disabled with the exception of when files are sent to ChatGPT, in which case the user will always be asked [159].

These three cases are coordinated by a local *orchestration layer*, which examines every query and determines how to process it: under Case I, II, or III. The orchestration layer prioritizes these cases in the order presented above: "many of the models" used to process queries are on-device [18, 11], only delegating "more sophisticated requests" to the more advanced model [22], and ChatGPT is only summoned through Siri "when helpful" [18].

The picture is further complicated by Apple's collaboration with Google. Apple has announced that its next-generation Foundation Models are custom-built in collaboration with Google and its Gemini models; these models run both server-side and on-device [10]. As we discuss in Section 2.5.1, Apple has not clarified the general criteria used by the orchestration layer to decide how a given query should be processed. However, they have shared examples of how some particular use cases are handled. For example, we know that all images generated with Image Playground are created on device, as well as previews of emails, messages, and notifications [18, 9]. Users have the option to turn on request handling logs, which detail which queries were processed on device, by server models, or by ChatGPT [9]. We now discuss what we know about the three cases in more detail.

Case I (Local model). Apple Intelligence's first processing method consists of a series of small AI models stored entirely on a user's device [19]. While some of these models are bespoke for certain tasks (e.g., code completion), their core local model consists of an approximately three-billion-parameter general-purpose language model, which can be adaptively fine-tuned for particular language tasks (e.g., summarizing text).

Case II (Server model). Tasks that are "sophisticated" enough to fall in Case II are outsourced to a larger model, which is hosted on Apple's servers [22]. While the full details of this model have not been made public, we refer to [79] for an in-depth third-party analysis of possible architectures by Trail of Bits. Outsourcing queries to a cloud-based model places—including E2EE data—undermines E2EE confidentiality. Apple offers some discussion of this heightened risk and proposes to use a trusted execution environment (TEE)[18] to perform server-side inference more securely [22].

Apple has built custom trusted execution environments in their data centers—which they call *Private Cloud Compute (PCC)*—to host (and perform inference with) their server-side model. User devices communicate directly with Apple's PCC and supply plaintext inputs to the model, which are evaluated inside the PCC (i.e., inside Apple's TEE). Apple employs several additional measures to enhance the security of the PCC computation nodes; some of these measures appear to be designed partly in response to security considerations such as those outlined in Section 2.4.[19]

Case III (Third-party model). Finally, Apple Intelligence provides an extension which sends certain user queries to ChatGPT. This feature is turned off by default, but can be manually enabled in settings [24]. When the feature is enabled, Siri examines each request and determines if ChatGPT "might be helpful to get the information" to respond to it [23]. The types of requests for which Siri may invoke ChatGPT may include direct prompts to ChatGPT, text generation, and summarization [23]. If either ChatGPT is directly invoked or Siri deems ChatGPT could be "helpful" [18], the user is asked in a pop-up prompt whether they would like to send the request to OpenAI for processing.[20] By default, this prompt occurs every time a request could be sent to ChatGPT, but can be disabled in settings[21] [23]. Apple collects limited metadata about the queries sent to ChatGPT, including the number and size of requests [21].

Upon enabling the ChatGPT extension in Siri, users may choose to continue with or without logging in to their OpenAI account.[22] If the user doesn't log in, query processing is consistent with Apple Intelligence's privacy policies: OpenAI is required to process queries ephemerally and must not use data from requests to train its models [24].[23] However, without logging into their OpenAI account, the user would not have access

[18] See Section 2.4 for an introduction to TEEs.

[19] For example: the PCC nodes leverage many of the hardware security technologies used by Apple's end-user products (such as their Secure Enclave Processor (SEP) and Secure Boot); the nodes are stripped of most functionality (except for security guardrails and inference capabilities) to minimize potential attack vectors; and user data (which is kept exclusively inside the PCC during computation) is deleted immediately once a response is returned to the user [22]. Apple additionally relies on a number of techniques to address supply-chain attacks and other backdoors, such as publishing every PCC production build in append-only, tamper-proof logs [22], and X-ray imaging during assembly for anomaly detection [13].

[20] For an example of what this looks like, see this example from the Apple Intelligence press release [18].

[21] See footnote 17.

[22] For an example of what the log-in prompt looks like, see [159].

[23] When a conversation is continued in the ChatGPT app or website, queries are processed under OpenAI's privacy policy, regardless of whether the user was initially signed in through Apple Intelligence or not [157].

to any premium features linked to their OpenAI account [157]. If, on the other hand, the user chooses to log in to their OpenAI account, query processing is governed by ChatGPT's privacy policies and the user's OpenAI account settings [24]—not Apple's policies. As such, for logged-in users, OpenAI may log requests and use query data to train its models if the user's OpenAI account permits such processing (even if this is inconsistent with Apple's policies) [24].[24]

We note that ChatGPT is able to access E2EE content on Apple devices in ways that are separate from its integration with Apple Intelligence. The ChatGPT desktop application, which is unrelated to Apple Intelligence, supports an iMessage plugin that gives OpenAI access to encrypted messages [158]. This plug-in runs locally on Apple computers and uses files that are already stored on the device [158]. Users must also enable multiple options to activate the feature, so it may be difficult to enable accidentally [158]. Since this feature is independent of Apple Intelligence, it falls outside the current scope of our analysis.

What We Still Don't Know. Based on the available documentation at the time of writing, significant details remain unclear about Apple's system. Open questions, to which the answers could be essential to independently evaluate the system's security risks and compatibility with E2EE, include the following.

- Are users always aware of when Apple Intelligence is invoked?
- How is it determined whether any given query will fall under Case I, II, or III?
- How will queries be processed for users who opt into Advanced Data Protection? There does not appear to be documentation regarding whether the processing of queries from these services (when Advanced Data Protection is enabled) will be handled differently by Apple Intelligence.

After our analysis and recommendations, we will return to the Apple system in Appendix B.

2.5.2 Samsung's Galaxy AI

Samsung introduced a new series of AI-integrated phones in January 2025 [178]. Several new phones, such as the Galaxy S25 Ultra, Galaxy S25+, and Galaxy S25, come with integrated AI assistants [178]: Samsung's own Galaxy AI, some of whose features leverage Google's Gemini [178]. While some previous Samsung phones were able to tap into certain Galaxy AI features, these newer models provide significant additional use cases.

Some Samsung AI features can access third-party servers to perform web searches. For instance, the Circle to Search feature is capable of performing Google searches, which may include Google's AI-generated recommendations in the results [174]. Samsung's AI assistant Bixby, which is included in the Samsung AI suite [176], is also able to perform searches using Perplexity [169].

What data on Samsung devices is E2EE?. E2EE applications on Samsung phones are either preinstalled on the device or can be downloaded from the Galaxy Store. Google Messages, for instance, comes with the phone and is E2EE by default between Google Messages users [175, 93]. Cloud storage on Samsung devices is not E2EE by default. However, users have the option of enabling *Enhanced Data Protection* (EDP), a feature which enables E2EE for call log, message, clock, settings, and app backups [182]. [25]

Possible use cases involving E2EE data. Galaxy AI can process content in E2EE applications, including via the Gemini integration which can perform tasks across Google, Samsung, and third-party apps [178, 180]. If users have Enhanced Data Protection enabled, Galaxy AI may also process additional content that is E2EE under EDP for purposes such as generating call transcripts and others [178].

Two-Tiered Approach. Galaxy AI queries are either processed on-device or outsourced to a cloud model. Samsung's documentation notes that Google Messages features, "most translations", and ambient wallpapers are processed on-device, whereas most other features, including auto-summarizing and image editing, are cloud-based [180].

On-device AI processes data locally with the Personal Data Engine, which is secured by hardware measures such as Knox Vault [179].

[24] ChatGPT accounts have model training on by default [156].

[25] Large amounts of data such as message attachments may not be backed up under E2EE, even with Enhanced Data Protection [182].

Samsung partnered with Google Cloud in 2024 to provide Gemini-powered AI tools [177]. Several aspects of Galaxy AI's cloud processing arrangement and security measures remain unclear, as further discussed below.

Galaxy AI is on by default on these newer phones, but some features require users to log in to their Samsung or Google accounts [180]. For example, Cross App Action (a feature that can perform tasks across different apps, including Google, Galaxy native, and third-party apps such as WhatsApp and Spotify) will require users to log into their Google account [181].[26] Users have the options to disable cloud-based features while maintaining on-device processing capabilities, and to disable all AI features, in their settings [180]. Samsung states that "personal data is never stored long-term or used for AI training—whether processed on-device or on the cloud" [183]; however, what qualifies as personal data remains unclear.

What We Still Don't Know.

- Does all Galaxy AI cloud processing take place on Google Cloud or just Gemini-based features?
- What security measures are used for Galaxy AI's cloud processing? How and when are encryption and trusted hardware employed?
- Are users always aware of when AI is invoked?
- Are users aware of how AI queries are being processed—locally or on the cloud?[27]

2.5.3 Meta AI in WhatsApp

Meta features an integrated AI assistant in the E2EE messaging platform WhatsApp [214]. This case, software integration instead of device integration, is slightly different than the previous two.

Meta AI can either be accessed in a standalone chat (between a user and the AI assistant) or it can join other conversations when it is summoned explicitly by a user sending a message containing the term "@MetaAI" [214]. In the latter case, Meta AI is only able to read messages that mention @MetaAI or explicit replies to these messages [214]. These boundaries are noted in the initial disclaimer the first time Meta AI is invoked, but do not appear afterward. Users are able to delete messages with Meta AI through the regular WhatsApp interface, though this does not delete the message from Meta AI [142]. In order to delete messages from Meta AI, users can use the "/reset-all-ais" command [142], though as mentioned below, it is not clear exactly what this does.

Some queries to Meta AI are processed using "Private Processing" [215], a TEE-based system for server-side inference. It is unclear if all AI features are handled via Private Processing, though we know that some features, such as Incognito Chat, do employ it [209]. When a user shares their messages for AI features that use Private Processing, the messages are end-to-end encrypted between the user and the server-side TEE [211]. The requests sent to Private Processing "only include data that's useful for processing the user's request", and are not stored once the request is completed [215]. The user data that "Meta receives and how Meta uses it depend on how you interact with the [AI] experiences" [213]. Private Processing is off by default [216], and, once activated, can be turned off at any time via the application settings [215]. The Private Processing whitepaper states that it is "clearly displayed to inform users" when Private Processing is invoked [211].

Users are also be able to turn on an "Advanced Chat Privacy" setting within individual chats, which prevents other users in the chat from exporting messages, and from "using messages for AI features" [210] in that chat. Meta AI is trained on a combination of publicly available online information, licensed information, and information shared on Meta products and services [140]. Queries sent from WhatsApp to Meta AI may be used for training, and this training data is kept by Meta "as long as [they] need it on a case-by-case basis" [140]; there is no option for users to opt out of this usage.

Meta AI in WhatsApp is capable of making web searches through third-party servers [209]. This feature is on by default, but can be turned off in incognito chats [209]. To search the web, Meta AI uses several search engine providers, such as Microsoft, Brave, and Finnhub [141].

[26] The Now Brief and Now Bar features, which provide users with relevant updates, will require users to log into their Samsung accounts [181]. And other features such as the Audio Eraser and AI Select also require a Samsung account [181].

[27] Users are able to select if they want to turn off AI features or turn off cloud-based AI features in their settings [180], but it remains unclear if users are told upfront how their queries are being processed.

What We Still Don't Know.

- Which features rely on Private Processing? How do features that do not rely on Private Processing operate?
- What exactly happens when users delete chats with AIs?

2.6 Confer

Confer is a new application proposing what it describes as "end-to-end encryption for AI chats" [131]. It was initially released as a standalone encrypted AI chatbot application by Signal's creator Moxie Marlinspike, and then acquired by Meta AI for "incognito chats" in WhatsApp (that is, secure chats between a user and Meta AI) [148]. Confer focuses on *adding encryption and other security protections to conversations with AI chatbots*—that is, Confer's focus is on strengthening the security of user interactions with AI chatbots, which at the time of writing, typically do not offer E2EE.

In contrast, our paper's focus is on *maintaining the existing guarantees of E2EE applications*, such as messaging where end-to-end encryption is on by default, when AI is introduced within the application environment. That is, we are concerned with preventing degradation of strong existing security guarantees.

These are different contexts and security design goals, and thus we consider Confer to be largely outside of our present scope. Our main point here is the importance of distinguishing these contexts clearly. Below, we briefly elaborate the relationship and differences between our focus and approach and those of Confer.

First, Confer relies both on encryption security and on TEE security, treating the server-side TEE where queries are processed as an "endpoint" of its "end-to-end encryption" [132]. These definitions diverge from our narrower definitions: we would not consider a cloud server an "endpoint" even if TEE-protected and thus we would not call this system "end-to-end encrypted." In conventional E2EE systems, endpoints correspond to user-controlled devices holding private keys. Even if a cloud server runs inside a TEE, we would not characterize it as an endpoint in the context of E2EE, and the security guarantee would not be E2EE as we define it: confidentiality against the service provider depends not only on cryptography but also on hardware isolation, vendor trust assumptions, and remote attestation mechanisms. We would instead describe Confer as promising example of *TEE-mediated private inference*.

Second, *if* Confer were integrated with an E2EE messaging application, then like most other AI assistant integrations, it would break the E2EE guarantee that only parties to the conversation could decrypt message contents, as discussed in Section 4. Introducing a server-side inference endpoint—even one protected by a TEE—changes the trust model and potentially expands the set of entities capable of processing plaintext.

To our knowledge, there is no suggestion that Confer be integrated with any E2EE applications; this is a hypothetical that we address for completeness. The currently available Confer application is a standalone web-based AI chat interface,[69] and current documentation does not discuss integration across apps and devices, how that would work, or whether such integration is expected or intended at all. Whether and how such integration would change the nature of the baseline security guarantee offered by the underlying apps or devices would depend on context. There is no equivalent issue in the context of standalone AI chat applications, as they are generally not E2EE to begin with, so no there is comparable downgrade risk.

TEE-based private inference may be a promising architectural pattern for certain AI applications, particularly where usability and server-hosted computation are primary design constraints. However, as discussed at length earlier in the paper, the security guarantees offered by TEE-based systems differ importantly from those of cryptographic E2EE. Conflating these models risks obscuring important differences in threat assumptions, trust boundaries, and user expectations.

3 Our Definitions and Taxonomy

Section 3.1 establishes definitions of key terms. Then, Section 3.2 introduces our taxonomy ranging from "Strict E2EE" to "No E2EE," consisting of five categories in decreasing order of the degree to which they maintain the integrity of E2EE. Finally, Section 3.3 introduces threat models for E2EE and AI.

3.1 Terminology

Next, we define nine key terms we use in our recommendations and analysis. In addition to the definitions in this section, we rely throughout on key definitions introduced in Section 2 (Background), such as *encryption*, *plaintext*, *ciphertext*, *model*, *inference*, *training*, and others. While Section 2 defines basic terms from established literature, the present section focuses on clarifying the terminology essential to make our analysis and recommendations precise within the scope of this paper. As such, our claim is not that the following are the universal definitions of these terms, but that these are the meanings we use in this paper.

Endpoint. An *end* is either the sender or a recipient of a message. An *endpoint* encompasses an "end" user and the user's device or devices (that the user has authenticated with a client application to send and receive messages[28]).

Endpoint-local. We say that an operation, computation, or process is *endpoint-local* if it takes place entirely on the user's physical device. Endpoint-local computation is often done with a client application. This definition excludes any forms of implicit or virtual locality (e.g., a user-specific private cloud container) as well as any transmission of data off the user's physical device.

Primary application content. For a given application, the *primary application content* is the content within the application that is part of the main service(s) that the application provides.

> For example, in a messaging app, the primary application content consists of communication contents (including text, images, voice messages, and any other content types supported by the app). In a videoconferencing app (like Zoom), the primary application content consists of the contents of video calls (including audio and video but also other in-call content such as polls or screen-sharing).

E2EE application. We use *E2EE application* broadly to mean any application that makes claims that it offers E2EE in any part of the application functionality—for primary application content or otherwise.

Metadata. Information that is auxiliary to the content offered in an application is called *metadata*.[29] In E2EE applications, metadata is generally not encrypted.

> For a messaging service, this includes who is communicating with whom, at what times and how frequently messages are exchanged, message lengths, and potentially many other types of data associated with a message—but not the message content.

E2EE content. We write *E2EE content* or *E2EE data* to refer to any data which a service provider explicitly or implicitly indicates is protected by E2EE in an application, and any derivatives of such data.[30] Many claims of E2EE by service providers are not very specific regarding which data are E2EE—in such cases, we treat a generalized claim of E2EE in an application as an implicit claim that the primary application content is end-to-end encrypted. We do not treat non-primary application content as E2EE content unless a provider has explicitly claimed that that type of content is E2EE.

> For example, if a messaging provider claims that its app is E2EE, then the text, audio, images, videos, and any other message content exchanged by users—and any derivatives thereof—are E2EE content. In contrast, messaging metadata would not be E2EE content in this scenario.)

Shared model. A shared model is an AI model that may be be used for inference by multiple users, and/or may be trained on multiple users' queries. In a given E2EE context, we describe a model as *shared* if its users may include people and/or AI models other than the sender or the recipients of E2EE content.

Per-user model. A per-user model is an AI model that can be used for inference exclusively by a single user, and that is continuously improved with queries from only that user. Note that the latter implies that data from a particular user is not used to train the per-user models of other users. As further discussed

[28] Endpoint identification is an important security consideration in practice, which is largely out of scope for this paper [102].
[29] The distinction between content and metadata is complex and context-specific [34].
[30] Derivatives include any mathematically correlated function of such data (excluding bit-length).

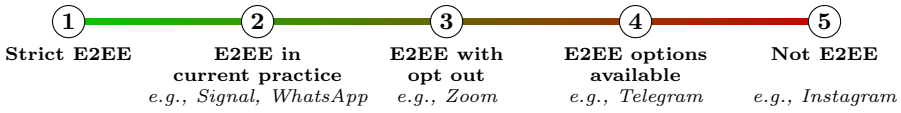

Figure 5: Our Taxonomy in a Nutshell. Examples are current as of August 2026.

in Section 4.2, this definition encompasses both "physical" and "logical" implementations, such as completely separate models for each user or a single foundational model dynamically refined during inference, respectively.

Third party. With respect to a particular piece of E2EE content, a *third party* is defined as any party which is neither the sender nor intended recipient of that content.

> For example, if Alice is sending a message to Bob using a messaging platform A, then A is a third party with respect to that message. If the platform A outsources some processing to another company B, then B is also a third party to users' conversations.
>
> If Alice is querying an AI assistant Hal directly, (e.g., "Hal, what will the weather be like this evening?") and Hal is a cloud-based AI assistant offered by platform A, then we do *not* consider A (and Hal) to be a third party with respect to that query.

3.2 A Taxonomy of E2EE

Our taxonomy, from "Strict E2EE" to "No E2EE," consists of five categories in decreasing order of the degree to which they maintain the integrity of E2EE, summarized in Fig. 5. Strict E2EE is a straightforward definition for a bare-bones application involving only messages; however, as discussed in Section 2.1, modern E2EE applications typically involve additional features and complexity beyond those explicitly modeled by formal definitions, which Categories 2–4 take into account. As such, E2EE applications in practice cover the full range of our first four categories.

Category 1 (Strict E2EE) refers to applications where *no E2EE content is seen or used by any third party at any point in the functioning of the application, for all available configurations of application settings and features.*

No major, widely used E2EE application meets Category 1's most stringent level of confidentiality at the time of writing.

Category 2 (E2EE in current practice) encompasses a range of applications that do not qualify as "Strict E2EE" due to caveat features — introduced below — but otherwise provide strong E2EE guarantees. To be in Category 2, *no E2EE content may be seen or used by any third party at any point in the functioning of the application,* except *for caveat features*, and applications must offer each user the option of disabling all caveat features in their own use of the application. (In other words, there must exist a configuration of application settings and features in which the application operates like a Category 1 application.) We place Signal[31] and WhatsApp[32] in Category 2.

Caveat features. Certain commonplace features that enhance user experience involve the disclosure of (often small amounts of) E2EE content to third parties.[33] As such, even with these features, it has become common to refer to and advertise the application as "end-to-end encrypted," both in industry and in academia, including among encryption experts.

Definition 1 (Caveat feature) Caveat features *refer to features in E2EE applications which involve third-party processing of strictly limited types or quantities of plaintext-dependent content to provide a user-experience-enhancing feature, and which are commonplace in applications that provide the strongest E2EE guarantees in current practice.*

[31] Showing the preview image of a link shared in Signal violates strict E2EE, for example [129, 28].
[32] WhatsApp's (non-E2EE) backup feature violates strict E2EE, for example [108].
[33] One example, already mentioned in Section 2.1.1, is *link previews.*

Caveat features definitionally depend upon societal practices; as such, they are an inherently sociotechnical concept, the precise boundaries of which may be debatable and are not susceptible to purely technical definition.

We give some examples of caveat features and argue for the utility of the term. Many caveat features are indispensible to the use of a modern messaging application, and it is unlikely that a user would use an application in their absence. We have already given the example of link previews; for similar reasons, many kinds of multimedia services, like gifs or stickers, involve fetching resources from a third party and are caveat features. Another example of a caveat feature is *message franking*, which is a technique (used, for example, in Facebook Messenger) for selectively revealing a user's messages in order to support e.g. reporting of abusive content [101]. Message franking is an illustrative example of a caveat feature in that its entire purpose consists in revealing E2EE content to a third party in order to support users; content moderation requires revealing content. The feature is seen as justifiable to the community because it supports a well-defined need, but in the strictest sense it violates the definition of E2EE.

It is also crucial that message franking is minimally invasive: only selected reported messages are revealed to the moderator. More broadly, we will insist in our definition that caveat features are "strictly limited" to reveal content only to the extent necessary to provide the service. By contrast, a hypothetical moderation technique which dumps the entire chat transcript would not be considered a caveat feature due to its invasiveness.

While major E2EE applications tend to have at least some caveat features (e.g., link previews and stickers are very common), some applications, such as Element [86], Session [186] and Threema [200], notably avoid them and aim to reduce or remove metadata, in a departure from this trend.

As alluded to before, we do not endeavor to draw a bright line separation between what features are "caveat features" and which features pose too great of a privacy leakage to include in an app described as "E2EE." This distinction is both societally determined and subjective and debatable at the margin, and expectations of what qualify as caveat features will evolve as uses of applications change. That said, we believe the examples given above can be safely described as "caveat features" (at least, at present) and hope they serve as a template for the type of reasoning deployed in making this distinction.

The current and prevailing use of the term "E2EE" generally refers to Categories 1 and 2, both in industry and in academia, including among encryption experts.

Category 3 (E2EE with opt out) refers to applications which offer strong E2EE features as an option, but either these features are off by default, or the application allows users to opt out of these features. Note, in contrast, that Categories 1 and 2 definitionally preclude E2EE being off by default and/or users turning E2EE off. A Category 3 application's E2EE features, *when enabled*, must be strong enough that *no E2EE content may be seen or used by any third party at any point in the functioning of the application, except for caveat features.* (That is, again, there must exist a configuration of settings and features in which the application operates like a Category 1 application.) We place Zoom [223] in Category 3, as it offers E2EE videoconferencing only when turned on in user settings. We consider Zoom to be in Category 3 since a user can set E2EE as their personal default in their account settings (which means calls that that user sets up will be E2EE unless the user explicitly indicates otherwise); what we focus on is whether E2EE is off by default *for any given user account.* Category 3 applications lack the important *systemic protections* discussed in §2.1.3.

Category 4 (E2EE options available) encompasses applications that offer E2EE as an option for limited features, but does not offer any configuration where all primary application content is protected by E2EE. That is, Category 4 applications are not E2EE because they *always involve processing of primary application content by third parties in the functioning of the application* unless data is protected by E2EE as a narrow exception. Telegram allows users to turn on E2EE on a per-chat basis and it is always off by default [197].

Category 5 (Not E2EE) describes applications that fall in none of the above categories. Category 5 does not preclude use of encryption: for example, applications that use *encryption* but not *end-to-end* encryption fall in Category 5. As later sections elaborate, Category 5 applications are outside the scope of this paper — unless service providers claim that they offer E2EE.

3.3 Threat Models

We begin by adopting a basic E2EE threat model, standard in the academic literature [102, 120], which assumes that messaging parties (endpoints) are honest, all server-side infrastructure can be compromised or malicious, and seeks to protect confidentiality and integrity of messages under these conditions.

Table 1: **Overview of threat models.** ✓ means a party is trusted, × means untrusted (and may collude with all others marked × within a column), and $\hat{\times}$ means untrusted but does not collude. – means a party is not present in a model. $*$ indicates we consider threat model variants with any of the other columns' trust assumptions on parties marked $*$. Yellow marks where a column differs from the preceding one. TM2 is the prevailing model in academia (e.g., [102, 120]). TM4 best captures how caveat features are treated in current mainstream industry practice. TM5 is shown because industry appears to favor it, and in red because we recommend strongly against it. See also Definitions 2, 3, 5 and 6.

Party	**Threat Model**							
	TM1	**TM2**	**TM3**	**TM4**	**TM5**	**TM6**	**TM7**	**TM8**
Sender	✓	✓	✓	✓	✓	✓	✓	✓
Recipient(s)	✓	✓	✓	✓	✓	✓	✓	$\hat{\times}$
E2EE provider — for client code	×	✓	✓	✓	✓	✓	✓	$*$
E2EE provider — as intermediary	×	×	×	×	×	×	×	$*$
Caveat feature parties (Def. 4)	–	–	$\hat{\times}$	×	×	×	×	$*$
TEE — for secure processing	–	–	–	–	✓	✓	×	$*$
Parties TEE may communicate to	–	–	–	–	✓	×	×	$*$
Security Guarantee	X	X	Y	Y	Z	Z	Z	Z

Upon this foundation, we extend our threat model to account for *caveat features*, with particular focus on AI assistants and their tool calls, which could involve communication with a third party beyond the E2EE service (e.g., in link previews or with a third-party AI assistant). Then, we further extend our threat model to account for TEE usage by E2EE providers.

Table 1 overviews all the threat models we discuss. The complete list of parties our threat models consider is given in the first column of Table 1; not all of our threat models take into consideration all of these parties. The threat models have the following relationship:

$$\text{TM1} \prec \text{TM2} \prec \text{TM3} \prec \text{TM4} \prec \text{TM7} \quad \text{and} \quad \text{TM4} \sim \text{TM5} \prec \text{TM6} \prec \text{TM7}\ ,$$

where $A \prec B$ means B is stronger than A, and $A \sim B$ denotes incomparability.

Definition 2 (E2EE Threat Models) *Take any Threat Model i in Table 1. Let $S \in \{X, Y, Z\}$ be the security guarantee in column i of the bottom row of Table 1, and let P_α be the set of parties marked α in column i, for $\alpha \in \{\checkmark, \times, \hat{\times}, -\}$. A protocol Σ is* E2EE secure (with parameters ω) under Threat Model i *if:*

> *In an execution of Σ, if all parties $p \in P_\checkmark$ behave honestly, all $p \in P_\times$ behave arbitrarily (possibly colluding with each other), and all $p \in P_{\hat{\times}}$ behave arbitrarily (not colluding), then the statement $S_\omega(\Sigma)$ holds.*

Our framework can naturally extend to model security evolving across time, capturing adaptive compromise by an adversary across multiple epochs.

Basic threat models (TM1–2). The strictest threat model (which we call Threat Model 1 (TM1)) is one in which not even the client code is trusted: users assume that the service provider may tamper with the code and, e.g., add a backdoor or keylogger. In this setting, users may code their client themselves according to a protocol specification, install code from an open-source repository they trust, download the application from their device's relevant app store and manually inspect it, or use a mechanism for verifying client code (e.g., binary transparency mechanisms [6]). Some academic literature (e.g., [222]) assumes this kind of model, but most modern academic work reflects prevailing industry practices in allowing for client code implemented by E2EE providers that is not realistically likely to be checked by users. We call this model TM2; it corresponds to Category 1 (Strict E2EE) in our taxonomy from Section 3.2.

Academic models generally elide software updates (e.g., [102, 120]), a critical security consideration in practice. We highlight the explicit examination of security definitions and detailed threat models for software updates in E2EE applications as a promising and underexplored direction for future work. We do not define a separate threat model for updates here, but rather, bundle software updates with "client code" in Table 1 and note that one reason updates may be overlooked is the prevalence of TM2 which trusts E2EE providers for client code.

Academic threat models generally explicitly consider the set of parties to be the messaging parties and the intermediary platform relaying messages. As such, they technically provide no guarantee about other third parties (indicated by – in Table 1) that commonly interact with E2EE systems in practice.

Definition 3 (X) *We define $X(\Sigma)$ to be the following statement:*

Let I_p denote the inputs of any party p participating in Σ, and $I = \bigcup_p I_p$. Let $I_\times = \bigcup_{p \in P_\times} I_p$ be the set of all inputs of parties in $P_\times$. For any PPT adversary $\mathcal{A}_\times$ that corrupts all k parties $\{p_1^\times, \ldots, p_k^\times\} = P_\times$; and for any PPT adversaries $\mathcal{A}_1, \ldots, \mathcal{A}_j$ that respectively corrupt each of $\{p_1^{\hat{\times}}, \ldots, p_j^{\hat{\times}}\} = P_{\hat{\times}}$, there are simulators $\mathcal{S}_\times, \mathcal{S}_1, \ldots, \mathcal{S}_j$ such that

$$\mathcal{A}_\times \left(\mathsf{View}_{p_1^\times}(I), \mathsf{View}_{p_2^\times}(I), \ldots, \mathsf{View}_{p_k^\times}(I) \right) \approx \mathcal{S}_\times \left(I_\times \right) \ , \tag{1}$$

and

$$\forall i \in [j], \quad \mathcal{A}_i \left(\mathsf{View}_{p_i^{\hat{\times}}}(I) \right) \approx \mathcal{S}_i \left(I_{p_i^{\hat{\times}}} \right) \ , \tag{2}$$

where $\approx$ denotes computational indistinguishability and $\mathsf{View}_p(I)$ denotes the view of party p in an execution of the protocol Σ with inputs I.

Introducing caveat features (TM3–4). Our threat model necessarily changes when we introduce caveat features, which by definition fail to satisfy TM1 or TM2 (see Definition 3) as they reveal E2EE content to non-endpoint parties. Caveat features and their context vary in ways significant for threat modeling. For example, link previews extend the threat model by allowing third party web servers (and intermediaries) to learn when URLs are sent in an E2EE chat.

Likewise, AI assistant features may invoke a model implemented and hosted by the E2EE provider, or a third party. For example, OpenAI's ChatGPT MacOS application has a plug-in that can access iMessage content [158]: Apple's iMessage is the E2EE messaging provider and OpenAI's ChatGPT is the AI assistant service. Caveat features may also involve communication with third parties that are not providing dedicated services for an E2EE application: e.g., link previews communicates E2EE content to web servers at the links shared.

Definition 4 *A* caveat feature party *is a third party who may receive E2EE content via caveat features.*

Typically, existing literature has considered caveat features in isolation from one another; however, in practice, popular E2EE applications offer a range of caveat features at once. Thus, TM3 treats third parties who may receive E2EE content via caveat features as permitted to receive a certain leakage function ℓ of E2EE content, and assumed not to collude with each other. For example, for link previews, $\ell(M)$ returns the message M if M is a URL, and returns $\perp$ otherwise.[34] We are not aware of prior work that considers a holistic threat model accounting for all caveat features and possible collusion between them, defined as TM4, and note this would be promising future work.

Definition 5 (Y) *For a tuple of* leakage functions $L = (\ell_1, \ldots, \ell_j)$*, we define $Y_L(\Sigma)$ to be identical to $X(\Sigma)$ as in Definition 3, except that* (2) *is replaced by:*

$$\begin{aligned} \forall \mathcal{A}_i \in \mathbf{A}, \quad & \mathcal{A}_i \left(\mathsf{View}_{p_i^{\hat{\times}}}(I) \right) \approx \mathcal{S}_i \left(I_{p_i^{\hat{\times}}} \right) \\ \textit{and} \quad \forall \mathcal{A}_i \in \mathbf{B}, \quad & \mathcal{A}_i \left(\mathsf{View}_{p_i^{\hat{\times}}}(I) \right) \approx \mathcal{S}_i \left(I_{p_i^{\hat{\times}}}, \ell_i(I) \right) \ , \end{aligned} \tag{3}$$

[34] In practice, link previews is more complex than this: e.g., it may also unfurl some URLs within messages, and how URL detection works can also be security critical.

where: $P_T = \{p_i^{\hat{\times}}\}_{i \in T} \subseteq P_{\hat{\times}}$ *is the set of third parties, other than the E2EE provider, that caveat features may communicate with; and* $\mathbf{A}, \mathbf{B} \subseteq \{\mathcal{A}_1, \ldots, \mathcal{A}_j\}$ *are the sets of adversaries that respectively corrupt parties in* $P_{\hat{\times}} \setminus P_T$ *and* P_T*.*

Bringing in TEEs (TM5–7). TEEs introduce a new attack surface not typically considered in E2EE threat models. As Section 2.4.2 mentioned, a strict threat model including TEEs might aim for resilience to various kinds of TEE compromise: whether at the point of production due to hardware components being compromised or backdoors in the OS, or after installation due to physical side channel attacks, for example. Another common approach in the academic literature on TEEs (e.g., [165, 221]) is to assume that the TEE is secure, and examine what guarantees are possible based on that assumption.

Importantly, *even with a TEE assumed secure*, the way in which the TEE is used by the E2EE provider can create new attack vectors with respect to E2EE content. Because TEEs process data in the clear, it is possible for them to exfiltrate data to third-party services; this is not possible with conventional E2EE systems, which only handle data under encryption (which would be useless if sent to a third-party service, assuming the security of encryption). For example, a TEE may determine the weather at a user's location by using a weather forecasting API, thus disclosing the user's current location to a third party, even if the E2EE provider hosting the TEE would not be able to learn that same information. Thus, TEEs, like caveat features, complicate the basic E2EE threat model by creating ways for data to flow to third parties, which may be distinct from the E2EE provider. TEEs may moreover combine with caveat features, such as in the case of an AI assistant running at the E2EE provider's server within a TEE which communicates with third parties via tool use.

Thus, introducing TEEs introduces two new types of party in our threat model: the TEE (or TEE provider), and parties the TEE may communicate to. As E2EE service providers have recently started deploying TEEs in their E2EE applications, they have appeared to both treat the TEE as fully trusted for secure computation, and to treat all parties the TEE may communicate to as fully trusted to handle E2EE content (see Section 2.5). We note that this set of parties is expansive — basically, anyone on the internet — in the common case of TEEs with general network connectivity. As also noted earlier, even if the TEE host aims to set up the TEE not to contact such an expansive set of parties or to send E2EE content to them, it is currently not understood how to reliably prevent such outcomes when generative AI is involved [218]. As such, we strongly recommend against this approach; such a trust assumption fundamentally incompatible with the promise of E2EE not to disclose E2EE content to third parties. Table 1 includes TM5 because industry appears to favor it, but colored red to indicate we do not consider within the scope of E2EE.

TM6 and TM7 treat the TEE as trusted and untrusted *for secure computation*, respectively, but treat third parties the TEE may contact as untrusted. We omit explicitly defining the variant threat models that would treat third parties the TEE may contact as untrusted but not colluding (i.e., $\hat{\times}$), for brevity.

Finally, we note that TM6 and TM7 are not currently achievable unless using TEEs incapable of external network communication [218] (see also Section 2.2).

Definition 6 (Z) *For a tuple of* leakage functions $L = (\ell_{\mathsf{TEE}}, \ell_1, \ldots, \ell_j)$*, define* $Z_L(\Sigma)$ *to be identical to* $Y_{(\ell_1,\ldots,\ell_j)}(\Sigma)$ *as in Definition 5, except* (1) *is replaced by:*

$$\mathcal{A}_\times \left(\mathsf{View}_{p_1^\times}(I), \mathsf{View}_{p_2^\times}(I), \ldots, \mathsf{View}_{p_k^\times}(I) \right) \approx \mathcal{S}_\times \left(I_\times \, , \, \ell_{\mathsf{TEE}}(I) \right) \ .$$

Some leakage functions are not acceptable within the scope of E2EE because they reveal "too much" E2EE content to third parties; regardless, precisely specifying leakage functions for caveat features and external communication of TEEs is a useful security modeling exercise.

Finally, not all users are perfectly aligned (TM8). As we have seen, there are a range of conceivable threat models for E2EE applications which vary in important and subtle ways. Different threat models may be appropriate for different user contexts and preferences. For example, Alice and Bob may be members of the same group chat but differ in which third parties they are willing to trust to handle leakage via caveat features (or indeed, whether to tolerate caveat features at all). More concretely, Alice may not want AI assistants to process any of her E2EE content whereas Bob might find AI assistant features useful. While E2EE threat models have traditionally suggested that messaging parties behave honestly in their collective interest, we therefore introduce the new idea that endpoints' interests may not be aligned: Bob's choices

might undermine the E2EE confidentiality that Alice seeks. We leave it as a fruitful direction for future work to explore this threat model variant in depth.

Note on UC. Universal composability [48] is an important framework that aims to account for security in interactions between systems while modeling systems separately, aligning with our aims in this paper. A UC definition of messaging systems with caveat features would require an ideal functionality that models each caveat feature explicitly. We do not adopt a UC definition, for the sake of generality: we present a general simulation-based framework for thinking about realistic threat model variants for E2EE, agnostic to specific caveat features. UC models for specific E2EE system designs would be valuable future work.

Note on third parties. One of our contributions lies in exposing the ways E2EE threat models change with the introduction of caveat features. Some caveat features invoke a service, such as AI assistants, which is integrated into messaging applications. Threat models differ incomparably when these service providers are the same or different from the E2EE provider. The E2EE provider is the primary adversary in the standard E2EE threat model; an external service provider is yet another third party that could gain access to E2EE content through caveat features. Whether an additional service provider is seen as a distribution of trust (beneficial for security), or an increase in threat exposure (harmful for security) may depend on deployment context and specific caveat features.

4 Security Implications of Integrating AI with E2EE

Building an AI assistant into an E2EE application involves several layers of configuration which affect the ways in which content may be used, processed, and stored. Building an AI assistant into an E2EE application involves many choices around how content is used, processed, and stored. Here, we analyze characteristics of these configurations and their security implications. Our analysis concludes that existing cryptography andprivacy technologies are *insufficient* to provide E2EE confidentiality for AI assistants processing E2EE content. Section 4.1 discusses how integrating AI assistants into E2EE applications raises significant security concerns due to the mechanisms by which AI assistants use and process data. Section 4.2 provides an overview of existing cryptography and privacy technologies that may (seem to) mitigate some of the aforementioned security concerns, and explain how these technologies are *insufficient* to provide E2EE confidentiality for AI assistants processing E2EE content.

4.1 Security Considerations for AI assistants

As explained in Section 2, AI assistants interact with data during both training and inference. For an implementation of an AI assistant to be compatible with E2EE, it must uphold the guarantees of E2EE during both training and inference.

4.1.1 AI Assistants Interacting with E2EE Data

As discussed in Section 2.2, a model's training data consists of vast amounts of information which can be continuously supplemented by data collected from users (e.g., queries, usage patterns) after the model is deployed and used for inference. This data would necessarily be plaintext as a model cannot be trained on encrypted text. As such, if user data is collected and incorporated into the training process, the model is inherently itself a derivative of E2EE content.

Additionally, models have been observed to reproduce portions of the training data in their outputs, at times producing significant portions of training data, sometimes verbatim or nearly verbatim [71, 167, 137]. The reasons for this phenomenon are not fully understood, but it is believed by many experts to be inherent to current machine learning and AI development techniques [51]. Training on E2EE content thus raises the security threat of models exposing sensitive information from training data even during regular, non-adversarial use.

The memorization and reproduction of training data in models also introduces the risk of individuals exploiting this behavior to extract certain data points and patterns. Prior work has shown that adversarial queries can expose information in the training data, which would directly compromise the confidentiality of E2EE data [50, 146]. Individuals may potentially perform *adversarial attacks*, involving using the model

(often in a black-box manner) as a means to acquire private data. Adversarial attacks are an active research area, and come in various forms, including *membership inference attacks*, where an adversary can determine whether a particular data point is part of the model's training set [191, 109]; and *training data extraction attacks*, where an adversary can recover elements of the model's training data often by generating candidate data points and using membership inference to verify whether the items are in the training set [50].

4.1.2 Key Considerations

Given that integrated AI assistants may process potentially significant volumes of E2EE data, they are *not* an example of caveat features (as defined in Section 3.2). How integrated AI assistants are implemented in an E2EE application directly impacts how strongly an application preserves E2EE, and the addition of integrated AI may demote an application's categorization within our taxonomy (Section 3.2). The risks outlined above underscore the following critical considerations for the implementation of AI assistants.

1. *Which users can query the models?* To preserve the guarantees of E2EE, user messages should only be used to train models that can be queried by the senders and intended recipients of the messages.

2. *Which parties have access to the inputs and outputs from the model?* Intermediaries that process plaintext user queries before relaying them to the model provider may have access to inputs and outputs from the model, in the form of chat logs. To preserve the guarantees of E2EE, inputs and outputs that depend on user messages should only be visible to the intended recipients of the messages.

3. *Are the inputs to the model used for anything else besides inference?* Even if the no parties and no other users can query or access logs of the model, inputs to the model may be used for other purposes—for example, the model provider may collect aggregate statistics on user queries. To be compatible with E2EE, queries should be used exclusively to fulfill users' requests. In other words, inference should be a *stateless* process.

4.2 How Privacy Technologies Can and Cannot Help

The privacy and confidentiality concerns outlined in the preceding subsection are by now well known to the machine learning, privacy, and security communities, and there exist various techniques to enhance privacy (according to various metrics and definitions) during inference and training.

Next, we evaluate whether the types of privacy enhancements that existing techniques offer meet E2EE confidentiality, considering a range of possible implementations. We find that existing privacy-enhancing techniques *cannot* address the security concerns we raise around integrating AI assistants with E2EE applications. We consider the following privacy-enhancing techniques: *endpoint-local training and inference*, *fully homomorphic inference (FHE)*, *trusted execution environments (TEEs)*, *per-user models*, *data sanitization*, *differential privacy*, *federated learning*, and *multi-party computation (MPC)*. While some of the techniques and implementations that we consider reflect existing practices, others represent novel configurations.

Underlying our evaluation is a generalizable analytical framework designed to *identify the key design considerations for implementing AI assistants in E2EE environments*, structured around four key considerations which are detailed below. Our framework can be used to evaluate and compare different implementations of AI assistants with (or without) diverse privacy technologies, and the degree to which they uphold E2EE security, generalizing beyond the specific technologies that we examine in this paper. The four key framework considerations are interdependent: at each consideration point, design choices can either affirm the configuration's compliance with E2EE or breach it (thereby rendering further considerations irrelevant). Figure 6 summarizes our evaluation framework as a flowchart, and Table 2 summarizes our analysis of various AI assistant implementations' compatibility with E2EE under our framework. In Appendix A, we present an extended framework for evaluating the compatibility of arbitrary features (as opposed to just AI assistants) with E2EE, which generalizes the framework from this section and encompasses three levels: technical implementation, data inputs, and user control.

Consideration #1: Can the model interact with the external world?. The first design parameter is whether the model is capable of interacting with the external world, such as through tool use. As discussed in Section 2.2.5, tools enhance the agent's capabilities on a variety of tasks. Some tools can be used purely

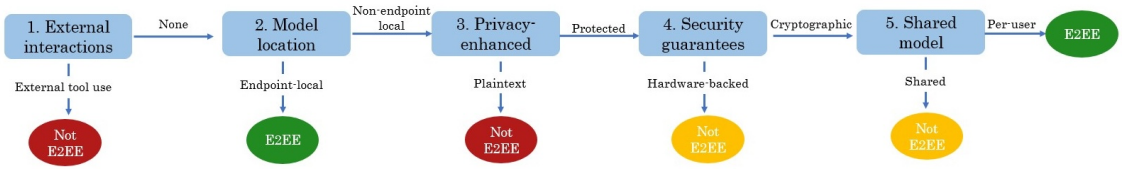

Figure 6: Evaluation framework for implementations of AI assistants.

locally, such as calculators and code interpreters. Other tools involve connecting with a third-party service, whether to supply the model with additional information present on the internet (e.g., search queries or checking the weather or news) or to interface with APIs (e.g., booking a flight or restaurant reservation). Such interactions necessarily communicate plaintext information that is a function of the content present in the user's plaintext query, and therefore leak information about that content to the third-party service(s) involved. Even if actual text from the user's query is not sent to the third party, the mere act of contacting a server for information can reveal E2EE content: for example, if a user performs a query asking about health concerns, and the model accesses websites on a particular medical condition.

This consideration is largely orthogonal to later considerations such as #2 and #4, in that both local and TEE-hosted models are capable of establishing connections with external services (and thus undermining confidentiality guarantees that would otherwise hold). In other words, external interaction is a privacy consideration for AI assistants independent of whether encryption is involved.

As the above examples illustrate, a model's interaction with external services may be to a user's benefit: indeed, these examples are all ones in which the model is doing so to serve a user's needs. However, such external interactions can undermine E2EE security, and more broadly, can disclose query content to third parties in ways that may be opaque to users. This highlights a fundamental limitation of E2EE-compatible AI assistant integrations: they cannot incorporate this kind of useful functionality — even if they use local models, and even in a hypothetical world where FHE is practical.

Consideration #2: Where is the model located? The next design parameter concerns where data is processed, i.e., where the model is hosted. We distinguish between *endpoint-local* and *non-endpoint-local* designs (as defined in Section 3).

Endpoint-local models can be fully compatible with E2EE, and are the safest way to perform inference and training, since no AI assistant queries are ever handled by an external server. This holds true provided that E2EE messages are only used to train endpoint-local models of the sender and recipients, and not used to fine-tune endpoint-local models in other devices.

Furthermore, recall from Consideration #1 that local models may still perform E2EE-undermining external interactions. Local models without external interactions, which are isolated within the E2EE application may be compatible with E2EE. However, even in these conditions, the model could leak E2EE content from one conversation into a response generated for another. Such isolation may not be common in practice, e.g., Apple Intelligence [20].

Using endpoint-local models involves significant challenges, as most consumer grade devices are unable to support the hefty computations involved in running a feature-rich modern AI assistant. For some limited types of computations, local processing is feasible: e.g., in Apple Intelligence, as discussed in Section 2.5. But even with the benefits of Apple's custom hardware, not all AI assistant queries can be supported with the limited computational faculties of a smartphone, and certain features require sending the message in plaintext off the device (see §2.5).

When *non-endpoint-local models* are used to answer queries, compatibility with E2EE depends on design Considerations #3 and #4 below.

Consideration #3: If the model is non-endpoint-local, is its processing of user queries privacy-enhanced, or in plaintext?. For non-endpoint-local models, the confidentiality of data depends on whether processing of queries occurs in *plaintext* or in some *privacy-enhanced* form (i.e., encrypted or contained in secure hardware) on the servers. Plaintext inference and training is not compatible with E2EE, as they would expose user data to third parties. Privacy-enhanced processing provides additional protection. There are various technologies that could implement some kind of "privacy enhancement"—such as fully homomorphic encryption (FHE), and trusted execution environments (TEEs)—which differ on their confidentiality guarantees, performance,

and other practical considerations. The compatibility of privacy-enhanced processing with E2EE depends on the nature of the confidentiality offered by the technology being used, which is the focus of Consideration #4.

Consideration #4: If the model is non-endpoint-local and performs privacy-enhanced processing, what is the type of the confidentiality guarantees?. As discussed in Section 2.4.2, different security technologies—such as TEEs and encryption—provide different types of confidentiality under different conditions. Established *cryptographic* techniques tend to provide the same type of security as E2EE under types of conditions, and are therefore more likely to be compatible with E2EE.

Fully homomorphic encryption (FHE) provides an instructive example. FHE enables arbitrary computations to be performed on encrypted data without the need to decrypt the data, and yields the result of the computation still in encrypted form. FHE makes it possible to "outsource" a computation on private data without revealing anything about the data to the outsourcing provider. FHE would thus allow non-endpoint-local models to process user messages for inference without decrypting them—at least in theory.

FHE schemes are far less efficient than typical encryption schemes like those used in E2EE applications. Consequently, computationally intensive operations like inference are prohibitively slow under FHE, and would not result in practical turnaround times for users. For example, Zama, a company working on applications of FHE in blockchain and AI, estimates that, at present, running an "average-sized LLM" under FHE costs about $5,000 per word [106]. While FHE is not a presently viable solution, it is an active area of research; substantial breakthroughs in the state of the art in the machine learning, hardware, and cryptography fronts will be necessary to eventually make this approach practical.

Hardware-backed security is the second major class of confidentiality guarantees, which involve using trusted hardware, such as TEEs (Section 2.4), to process user data. In this case, the AI model would be hosted inside a non-endpoint-local TEE. Processing inside the TEE would happen in plaintext, while still being shielded from outside entities by the hardware protections of the TEE.

As described in Section 2.4.2, hardware-backed security is different from E2EE security. Hardware solutions represent a substantial security/privacy improvement over plaintext inference, which may well be a fruitful avenue for exploration, beyond the scope of this paper. Our scope is to stress that the importance of clearly distinguishing them from E2EE and cryptographic security guarantees.

Finally, we reiterate that a model hosted inside a TEE may communicate data that is protected inside the TEE to external services in plaintext (Sections 2.2.5 and 2.4). Tools that provide access to encrypted data to third parties can be used to facilitate exfiltration of confidential and private information [218], a security threat (Section 3.3). Any tool usage of E2EE data requires plaintext communication, violating Consideration #1.

Consideration #5: If the model is non-endpoint local, is it a shared or per-user model?. Any model training introduces an additional critical consideration: who can query models trained with whose E2EE data? Since a model is a derivative of its training data, it is important that queries over this training data are consistent with the guarantees of E2EE.

There are two possible configurations for this design choice: *shared* or *per-user* models, as defined in Section 3.1. Shared models are not compatible with E2EE, since users would be able to receive inference outputs that are a function of other users' E2EE data, which violates E2EE confidentiality. In practice, this means that E2EE data may be exposed to entities that can query the model (e.g., due to the possibility of leakage of training data and adversarial attacks). As we discuss in more detail in Section 4.2.1, mitigations for adversarial attacks (e.g., differential privacy) are insufficient to fully address this concern. Shared models are incompatible with training on E2EE content, irrespective of the implementation.

On the other hand, per-user models can be compatible with E2EE, provided that applications enforce strict isolation between user data, and that any components that depend on E2EE data must themselves be adequately privacy-enhanced in the sense outlined in Considerations #3 and #4. There are a number of ways an application could support per-user models. For example, a service could maintain a separate copy of a foundation model for each user in the system, which is fine-tuned with that user's data. This approach would maintain strict isolation, but require large amounts of resources for applications with many users.

A more practical approach is to use *retrieval augmented generation* (RAG) techniques: the application hosts a single foundation model which is not trained on user data, but which is dynamically refined during

inference using user-dependent data sources [135].[35] The application could store E2EE data for every user locally (e.g., [135]), and retrieve and incorporate it locally during inference—temporarily fine-tuning the model. After the query is processed, the base model should revert to its prior state, which was not dependent on any E2EE data, resulting in a *stateless* augmentation process. While this reversion or statelessness would largely address the specific concern of Consideration #5—namely, which users can query the models trained with the E2EE data of which users—the storage and processing of the per-user E2EE data and the temporarily fine-tuned model (which is itself E2EE data) would, in turn, need to be scrutinized under Considerations #1–#4 and comply with the recommendations we put forward throughout this paper.

Applying this framework. A specific implementation of an AI assistant can be evaluated with respect to each of the four design choices in our framework. From the results of this evaluation, we can determine the implementation's compatibility with E2EE. For example, the analysis of local inference with no external tool use would be as follows:

(1) None (2) local, (3) N/A, (4) N/A, (5) N/A, therefore fully compatible with E2EE.

Example analyses of a range of possible implementations are presented in Table 2.

Table 2: Example analyses of different implementations of AI assistants under our framework. Green tags denote an implementation compatible with E2EE, yellow tags denote an implementation that is not compatible with E2EE but has additional protections, and red tags denote implementations that are most vulnerable.

Implementation	External interactions	Location	Protected	Guarantees	Shared or per-user
On-device, fine-tuned model, no tools	None	Endpoint local	No	N/A	N/A
On-device, fine-tuned model, with tools	Possible	Endpoint local	No	N/A	N/A
Server-side plaintext inference w/o fine-tuning	Possible	Non-endpoint local	No	N/A	N/A
Server-side inference using FHE w/o fine-tuning, no tools	None	Non-endpoint local	Yes	Cryptographic	N/A
Server-side inference on separate per-user fine-tuned models on TEEs	Possible	Non-endpoint local	Yes	Hardware-backed	Per-user
Server-side inference using FHE on a global model fine-tuned with DP	Possible	Non-endpoint local	Yes	Cryptographic	Shared
Server-side inference using FHE on a global model	Possible	Non-endpoint local	Yes	Cryptographic	Shared
Apple Intelligence	Possible	Non-endpoint local	Yes	Hardware-backed	N/A

Connection to our taxonomy. Recall that Section 3.2 describes a taxonomy from "Strict E2EE" to "No E2EE." The (in)compatibility of an AI assistant implementation with E2EE, as assessed under the evaluation framework just presented, directly impacts an application's categorization under our taxonomy. An application's original features, before integrating an AI assistant, determine what we call its *initial* categorization; if AI features are added, a fresh evaluation and categorization are necessary, and the new categorization will depend on the implementation details highlighted in our evaluation framework.

The addition of a non-E2EE-compatible AI assistant implementation to an application that is initially in Category 1 or 2 will result in the application being "demoted" out of those two categories, as the added AI feature will render E2EE content no longer fully protected in the way that E2EE promises. In more detail:

- If the AI assistant implementation is fully compatible with E2EE (green tags in Table 2), the application maintains its initial category (from before the AI feature was added).
- If the AI assistant implementation is *not* fully compatible with E2EE (yellow or red tags in Table 2), the application is demoted to Category 3, 4, or 5, as follows: If the AI assistant is *off by default* or

[35] See [219] for more on retrieval augmentation.

provides an option to turn the AI assistant off application-wide, the application is demoted to Category 3. If the AI assistant is *on by default* and only *provides an option to turn off in specific contexts* (e.g., for individual chats), the application is demoted to Category 4. If the AI assistant *cannot be turned off*, the application is demoted to Category 5.

4.2.1 Discussion on Privacy-Preserving Training

This section considers a range of privacy-preserving training methods—differential privacy, data sanitization, federated learning, and multi-party computation—and discusses how the types of privacy guarantees they provide are different from E2EE confidentiality. None of these techniques can guarantee E2EE confidentiality for training shared models on E2EE content.

Differential privacy (DP) is a method for designing algorithms to compute statistics over datasets such that the results do not depend "too much" on any individual data entry. In contrast with encryption's binary security guarantee, DP is parameterized by an "acceptable" amount of "privacy loss," representing dependence upon individual data entries (encoded via a parameter ε) [83]. Applying DP to machine learning is an active area of research; training with DP incurs a computational overhead and loss in accuracy compared to a non-DP model [162, 168]. Though DP is often discussed alongside cryptographic techniques, DP offers a categorically different security guarantee than encryption, applicable under different conditions.

Data sanitization aims to remove "sensitive" information from data that is aggregated into training datasets. This presents the difficulty of determining what information is "sensitive;" sensitivity depends on context and there is no generally correct heuristic for this procedure [47]. In contrast, the strict confidentiality of encryption is agnostic to message contents, and provides systemic security benefits by protecting mundane communications alongside sensitive ones (as discussed in Section 2.1.3).

Federated learning is another approach intended to enhance the privacy of training data, by having end-users locally train local models with their personal data. The resulting local model information is then sent to and aggregated into a single model by a central entity. The privacy benefit is that the raw personal data does not leave users' devices. But the local models, which are dependent on that personal data, *do* leave users' devices. In turn, the final centralized model is still dependent on users' personal data as it depends on these local models. Thus, federated learning does not address the issues of training data leakage through model use or adversarial attacks.

Multi-party computation (MPC) is a technique that allows multiple parties (e.g., users or companies), each with a private data set, to collectively compute a function (e.g., model training) on all of their data sets combined and access the result (e.g., a trained model), without directly sharing their data sets with each other. To use MPC, all parties must agree on a function they want to compute. MPC provides the guarantee that parties will learn the desired function output (e.g., the trained model) and *nothing more than that output*.[36] MPC does not constrain the function parties choose to compute, and in particular, allows for the function output to be dependent on, and reveal information about, the input data sets—such as when the output is a trained model.[37] A trained model resulting from an MPC computation is just as dependent on its training data as a model trained without MPC. Thus, MPC does not address the issues of training data leakage through model use or adversarial attacks.

All of the above techniques provide valuable privacy protections. However, their privacy goals are different in kind from the privacy goals of E2EE systems. As such, they cannot achieve E2EE confidentiality for training shared AI models on E2EE content. Future advances in the technologies discussed will likewise be valuable but will not bring these technologies closer to E2EE, because the type of privacy protection they provide is different in kind from E2EE confidentiality. Put differently, private training techniques create a *spectrum* of privacy—as determined by, for example, the level of noise introduced, or how many sensitive words are removed from the message—whereas E2EE is a *binary* kind of guarantee.

Private training techniques themselves provide a lens to illustrate the tension between E2EE and AI training: we can frame the guarantees of E2EE as "extreme" forms of various private training techniques,

[36] Necessarily, by learning the output, parties may also learn information that can be inferred from that output.

[37] The confidentiality guarantee offered by MPC thus depends heavily on the function that the parties choose to compute, which they can choose freely. For example, if the function they choose to compute is the identity function—i.e., the output is simply a copy of all of the parties' data sets—then MPC provides no confidentiality at all. If the chosen function consists of aggregate statistics—e.g., the average salary of the many employees in an industry—then MPC provides a stronger guarantee.

which correspond to degraded, impractical instantiations of each technique. For example, the guarantees of E2EE can be cast as differential privacy with $\varepsilon = 0$ (i.e., when the statistical distance between training data with or without a particular item included is 0). However, this value can only be achieved when private data is removed from training to begin with [219], at which point differential privacy is vacuous. Similarly, the guarantees of E2EE can be described as data sanitization in the case when all parts of an input are considered sensitive. Thus, sanitization techniques would remove E2EE messages entirely, which is equivalent to not training at all to begin with. As these examples show, to meet E2EE security, private training techniques essentially degrade to cases where training data is removed altogether.

5 Background II: Consent, Privacy, Users, and Areas of Law

Next, we turn to legal and normative considerations around clear disclosures about and meaningful consent to the use of AI integrated with E2EE systems, complementing our preceding analysis of when and how such systems can be technically compatible with the confidentiality promised by E2EE. The legal landscape around E2EE and AI is rapidly evolving, raising questions about user protection and data practices that, in turn, have implications on security and confidentiality. This section provides preliminary background on data practices (Section 5.1), notice and consent (Section 5.2), and relevant areas of law (Section 5.3).

5.1 What Users Understand About How Their Data is Used

Users are typically unaware of data practices and don't often know what data is being collected about them, by whom, and for what use [204]. Privacy issues are often context-dependent, and decisions may be made based on how safe or trustworthy a company is perceived to be [123].

TOS and privacy policies in practice. When evaluating whether to try a new product, users cannot easily understand terms of service (TOS) and privacy policies, which are often incomprehensible in practice, exceeding reasonable standards of length and readability [128]. Research has shown that very few people read privacy policies and terms of service, and that reading these for all the apps that a person uses would take hundreds of hours a year [138]. Even if users did somehow read all the available information about how their data is being handled, it would be unrealistic to expect them to assimilate this information and act in their best interest based on it, given the vast amounts of data generated over an extended period of time [2].

Deceptive patterns. Another barrier to users' ability to make informed decisions about their data is the use of deceptive design. These deceptive patterns (or "dark patterns") are sets of design choices that coerce users to do things that aren't in their best interest. Dark patterns can take the form of trick wording, comparison prevention, hidden costs, and nagging. These techniques can cause financial harm, loss of control (for instance over data usage), and discrimination [46]. In the case of privacy policies, dark patterns may be employed to coerce users into making choices that are beneficial to the given service, but not necessarily beneficial to the user. Deceptive patterns are often used to make users part with more data than they may be aware of or be willing to disclose [150]. Documenting and regulating dark patterns is challenging: they are deceptive, so users often are not aware of them, and users may also feel embarrassed about having been tricked, which complicates reporting [194].

Data can be surprisingly revealing. It is possible to infer surprisingly invasive information from someone's shopping habits; even to predict a consumer's pregnancy before close family members are made aware [82] [105]. Time-stamped record of an individual's cell phone location may be stored by mobile network operators for extended periods of time without the user's knowledge.[38].

Data brokers. Companies that collect data from users often sell user information to data brokers who combine the data and build incredibly detailed dossiers about individuals' profiles that are available for sale to the general public [189]. Users may not know that data brokers even exist.

The "Privacy Paradox". Consumer actions tend to undervalue their privacy. This social phenomenon is captured by the "privacy paradox," which identifies that people's stated concerns about privacy are often not reflected in their online behaviors [3]. Thus, giving consumers a choice to opt in or out will often not effectively translate their preferences. Additionally, the sheer volume and time constraints of decision-making

[38] *Carpenter v. United States*, 585 U.S. 296 (2018).

with regards to privacy has been known to cause "consent fatigue" [192]. Traditionally, consent has been provided in a binary format, like an on/off switch. However, as services offer wider ranges of settings and process increasingly more types of user data, privacy choices have become more granular. These consent requests compete for consumers limited time and attention, making it less likely for each request to be considered appropriately [89].

5.2 The Failure of Consent

5.2.1 Legal Consent

The concept of user *consent* plays a significant role in nearly all laws that regulate privacy [192]. From contract law to tort law to privacy statutes, notions of consent are relied upon to legitimize a wide range of data collection and processing activities that would otherwise not be permissible. Legal definitions describe consent idealistically as an individual's voluntary agreement that is wilfully and freely manifested, in a manner that is not coerced, fraudulent, or erroneous [208]. In practice, the situations in which courts consider a user to have consented to a condition often fail to meet this idealistic description. Normatively, the legal framework of consent has been heavily criticized for insufficiently reflecting the realities of consumers and for ignoring the underlying social dynamics of data privacy.

Legal consent presupposes a prior *disclosure* or legal *notice*. In digital services, legal notice may come in the form of a privacy statement, terms of service, or banner displayed by websites and apps. In contract law, privacy notices are regarded by a great majority of courts as a valid contracts, despite extensive research (see Section 5.1) showing that these notices are not read in practice [31]. Terms of service and privacy policies are generally upheld as legally binding contracts even if users did not consent by an affirmative action like signing a document or checking a box labeled "I agree" while creating an account—American courts often treat a user's continued use of a website or app as "implicit consent" to its terms and conditions.

The "notice and consent" (or "notice-and-choice") approach is the prevailing framing of consent in the United States. The underlying rationale is that consumers have the choice to stop using the service entirely or to opt out of specific features if they are not in agreement with the provided terms or notice [43]. Laws like the Gramm-Leach-Bliley Act, the Children's Online Privacy Protection Act, and the Health Insurance Portability and Accountability Act, which respectively regulate financial information, children's data, and health data, require organizations to provide individuals with a privacy notice and an opt-out mechanism.

Another legal approach to consent is to require the "express consent" of users. This approach requires consumers to opt in by taking an affirmative action to indicate consent, such as by checking a box or clicking an "Accept" button. The European Union's General Data Protection Regulation requires "informed," "specific," and "unambiguous" consent, where the term "unambiguous" is understood to reject notions of implied consent through inaction.[39] Due to the European Union's regulatory strength, the so-called Brussels effect, and the GDPR's exterritorial scope, many privacy laws across the globe have adopted similar standards for consent [44]. In the United States, although the notice-and-choice approach is still more prevalent, some U.S. state privacy laws require express consent in specific circumstances.[40]

5.2.2 Normative Critiques

Regardless of the consent regime adopted by privacy laws, the conditions for meaningful consent are widely regarded as fictional in normative terms. Meaningful consent would be informed, not coerced or unduly manipulated, and require that individuals have the capability to make an appropriate risk assessment (e.g., based on a cost-benefit analysis) [192].

There are many reasons why users are unable to meaningfully consent, many of which are related to the considerations already discussed in Section 5.1. Notoriously, notice and consent is inadequate in informing individuals about an organization's privacy practices because consumers do not read privacy notices [134].

Moreover, understanding data privacy implications often requires technical expertise. This is especially true for AI/ML systems, which are frequently opaque and referred to as "black box" models. Even the developers of many such systems do not fully understand the privacy implications of training their models on user data [104]. In general, consumers cannot be expected to understand the implications of disclosing particular

[39] Regulation (EU) 2016/679 ("GDPR"), Article 7 and Recital 32.

[40] E.g., the Colorado Privacy Act requires opt-in consent for processing sensitive data. (CPA § 6-1-1308 (7))

pieces of their data, especially in aggregate forms. Consumers overestimate baseline legal protections [107] and underestimate the risk seemingly inconsequential disclosures pose over time [193].

Regarding the contractual relationship between consumers and online platforms, consumers do not have meaningful bargaining power or choice [58]. The monopolistic tendencies of online social networks are exacerbated by the low costs of scalability and increased value as the number of users grow, resulting in lack of real choice between services [59]. Even if alternative services exist, privacy practices and terms will often be the same among competitors. Additionally, the asymmetric nature of the relationship means that individual opt-outs require scale in order to exercise compelling power against large corporations.

Another often overlooked condition of consent is its validity across time. Viewing consent as a one-time deal does not match the ongoing nature of data processing activities. Consent may expire or may simply not be enough to capture an activity that happens too far into the future, where circumstances may have changed for either party. Attempting to solve this, some laws like the GDPR and the Colorado Privacy Act explicitly acknowledge that consent may be withdrawn at any time and that consumers have a right to request for data deletion.[41] Although these are technically two different user rights, in the case of recurring data processing activities, the withdrawal of consent will often require deleting personal data in order to honor the consumer's wishes. However, considering the ongoing nature of AI/ML system-training, the deletion of data is, in general, infeasible. [72]. Thus, opting out or withdrawing consent after the data has already been used for training is rendered pointless and ineffective to stop undesired results, such as a model outputting the personal data on which it was trained.

Finally, legal consent largely elides the social dimensions of privacy. Although privacy can easily be understood to have a social connotation that highlights its value as a shared good, legal consent (in its current form) cannot, and is typically linked to an individual permission. A person is unable to consent for another unless in some legally accepted representation capacity. Multi-party messaging data raises complex issues outside the scope of existing legal consent frameworks, as legal consent does not account for the fact that shared data is extra-individual by nature. Genetic data is another example of information in this category, as it reveals information about a person's entire family.

5.2.3 Social Dimensions of Information Privacy

In his influential book *Code and Other Laws of Cyberspace*, Larry Lessig argued that laws are only one kind of the several regulations that affect the way people behave [124]. Architecture (whether of buildings or code), markets, and social norms can regulate individuals' behavior just as much, if not more than, law.

Defining social norms precisely may be an impossible task in its own right. Helen Nissenbaum's theory of contextual integrity argues that the term "privacy" must be understood depending on the context [149]. The norms around privacy in a health context may be completely different than in a school context, and from group to group. For example, if a child is diagnosed with ADHD by a private psychiatrist, it would be a severe violation of the child's privacy for the psychiatrist to disclose that information outside of the child's family. However, if the psychiatrist is employed by the child's school, disclosure to the child's teacher and the school's accommodation staff may be expected. Blanket consent fails to account for these norms and undercuts privacy—addressing the flow of data itself is the only way to fully address privacy concerns.

Linnet Taylor, Luciano Floridi, and Julie Cohen are three of many academics who argue for notions of "group privacy." The motivating thought behind this concept is that giving each individual the last word on privacy concerning their personal data may not capture all the interests at stake. There are different flavors: Taylor and Floridi argue that informed consent can only be given by a data subject, so if the subject is a group, the consent of individuals is meaningless [127]. The definition of the relevant group greatly affects the manner in which informed consent can be given. For example, acquiring the consent of a group chat would look very different than acquiring the consent of an ethnic group. Cohen argues that reduced privacy harms the normal functioning of society in a way that manifests far beyond the individual [60]. For example, individualized search engine results distort citizens' perception of the world and make participation in the democratic process less likely to result in compromise. Both of these approaches acknowledge that an individual's grant or withholding of consent is unlikely to capture the full set of issues involved.

[41] Regulation (EU) 2016/679 ("GDPR"), Article 7(3) and Article 17(1)(b) and CPA ("Colorado Privacy Act") §6-1-1306 (1)(a)(IV)(C).

Several scholars have proposed changes to consent based on the idea of group privacy. August Bourgeus and Laurens Vandercruysse suggest a form of semi-automated consent, which would allow individuals to set their preferences accurately once and carry them throughout their online life [42]. Daniel J. Solove advocates for "murky consent," a baseline model which would rely on data holders accepting a duty of loyalty, duty to avoid thwarting reasonable expectations, and a duty to avoid unreasonable risk [192]. Jack Balkin, and many scholars since, argues that holders of sensitive information should act as fiduciaries to data subjects [29].

5.3 Areas of Law

We refer to several substantive areas of law that affect E2EE service providers. They are summarized below.

Contract law. Contracts are agreements between private entities. "Contract law" is a legal framework that enables enforcement of contracts by the judicial system. With few exceptions (for illegal and unconscionable conduct), contract law does not place limits on what may be included in a contract. This "freedom to contract" is especially strong in the United States.

Consumer protection law. Consumer protection law places limits on certain practices that corporations might undertake that could harm consumers. Thus, consumer protection law can be seen as one limitation on the freedom to contract. For example, one particularly voluminous subfield of consumer protection law encompasses all the safety regulations regarding car design.

Data protection law. Data protection laws are regulations that govern how personal data is handled and used. The definition of personal data may vary between jurisdictions, but generally means information that can identify individuals. These laws establish rules and guidelines for lawful processing of personal data.

Antitrust law. Antitrust laws are regulations that promote competition and prevent monopolies by regulating the conduct and organization of businesses.

6 Legal Implications of Integrating AI with E2EE

The integration of AI assistants into applications that process any form of E2EE data raises important considerations under current legal regimes, which are related to the security considerations we have laid out thus far, as well as broader legal issues around disclosure and consent.

In this section, we begin with a brief survey of privacy terms in several E2EE messaging service providers' Terms of Service and Privacy Policies (Section 6.1). We then overview relevant legal obligations under data protection and consumer protection law, considering how AI service providers are currently regulated in the US (Section 6.2) and in the EU (Section 6.3). The specific issue of AI processing of E2EE data has not yet come up in legal proceedings, but our discussion provides background on relevant existing law, as well as ongoing controversies and uncertainties, that will inform how AI processing of E2EE data is treated under US and EU law. Finally, we note regulatory trends across other jurisdictions (Section 6.4) and provide a brief note on how competition law may affect data-driven business operations (Section 6.5).

6.1 Terms of Service and Privacy Policies of E2EE Services in Practice

When drafting terms of service and privacy policies, international companies must account for several significant regulations. In general, companies are incentivized to write overly broad terms of service that cover any possible future business development, because courts generally uphold broad contract terms, especially in the United States [134]. In the case of an E2EE messaging service, for example, a company might be incentivized to include a term in its fine print that says that it may stop offering E2EE at any time, even if all of its promotional material touts its E2EE, and even if it does currently offer strong E2EE and has no plans to cease providing it. Similarly, while E2EE messaging services may have high standards with regards to the third parties they contract with, they regularly attempt to disclaim liability for the actions of any other entity. Third party liability disclaimers are found in all four of the terms of service summarized below.[42] So, while E2EE messaging services may advertise lofty ideals, and even if they really do promote

[42]This does not prevent companies from including quality assurances in contracts with vendors. In the event of a data breach, vendors can be liable to both direct customers and end consumers. *In re Blackbaud, Inc., Customer Data Breach Litigation*, 567 F. Supp. 3d 667 (D.S.C. 2021).

those ideals by deploying strong privacy technologies, sometimes their terms of service may not make the same promises. For example, WhatsApp's terms of service would allow it to share user data for marketing purposes, a function that would conflict with the E2EE that it *does*, elsewhere, commit itself to providing. Such inconsistencies are common in terms of service because of the incentive to over-include.

The contractual privacy terms of four messaging services which offer E2EE features are summarized in Table 3. Notably, the three services (WhatsApp, Meta's Messenger, and Telegram) that do not promise use of E2EE do not use it for messages with bots or businesses. Bots or businesses could store user messages in plaintext if they chose to—in other words, these three services have decided not to dictate third-party message handling practices, instead allowing the third-party's privacy policy to govern these interactions. Finally, WhatsApp, Telegram, and Signal broadly market themselves as "secure" messaging platforms, and occasionally use the term "end-to-end encryption" in marketing. Meta's Messenger, on the other hand, does not mention security on its splash page, at the time of writing.

Of the four services discussed, WhatsApp's privacy policy notably promises less security than its marketing materials promise, and than its product actually provides. To our knowledge, the service as provided at the time of writing falls into Section 3.2's Category 2, consistent with marketing materials, while the privacy policy describes a Category 3 system.

Table 3: Comparing Privacy Policies of Select E2EE Messaging Services[43]

Service	**Legible policy in single document?**	**E2EE Promised?**	**Circumstances of transfer of consumer data to third parties**	**Limitations on third parties mentioned in ToS**
Facebook Messenger	No	"On some Products, you can use end-to-end encrypted messages."	For marketing and to advertisers on Facebook, normal operations	None
iMessage	Yes	Yes	N/A	N/A
Signal	Yes	Yes	For text message verification	N/A
Telegram	Yes	For "secret chats"	For on-platform purchases, translation, and transcription	Bots can only get username, language, IP, and shared messages
WhatsApp	Yes	"Offered"	For marketing, normal operations	None

6.2 Overview of US Legal Considerations for E2EE & AI

Data protection statutes provide consumers with relatively limited protections as well. The United States does not have a federal general data protection law and sector-specific laws would only cover messaging services in the rarest edge cases. While some states, like California, have stronger general data protection laws, services can and do vary their products from state to state to comply with these laws. The GDPR may have more significant implications, although its scope is also limited, and many general data protection laws outside of the EU are modeled after the GDPR and suffer from the same limitations.

In the US, consumer protection law, a broad hammer mostly wielded by the FTC and state attorneys general, is likely to be provide stronger protections than data protection law, and has the flexibility to provide strong checks against the most egregious business practices, as discussed next.

6.2.1 Consumer Protection and the US Federal Trade Commission

Consumer protection law in the United States would constrain some implementations of AI in E2EE systems. At the federal level, the Federal Trade Commission Act of 1914 (15 U.S.C. § 45) created the Federal Trade Commission (FTC) and enabled it to investigate "unfair" or "deceptive" trade practices.

[43]This review is based on active versions of privacy policies as of December 18, 2025.

Deceptive trade practices—the most relevant category here—are defined as those which constitute "a misrepresentation, omission, or other practice, that misleads the consumer acting reasonably in the circumstances, to the consumer's detriment" [143]. The FTC investigates security-related incidents as deceptive practices. Such investigations are often prompted by data breaches and result in the FTC discovering misleading statements made to consumers about security. The FTC has discretion over which investigations to pursue—and must make choices, as it does not have the capacity to pursue every lead it encounters.

For example, increased public use and scrutiny of Zoom in 2020 led to the discovery that Zoom was not protecting all video calls with the "end-to-end, 256-bit encryption" it advertised. The company settled with the FTC, agreeing constructively to implement a comprehensive security program and cease misrepresentations related to encryption [126]. A dental software company advertising its use of "encryption," unqualified, was found in an FTC proceeding to be misleading customers because the level of encryption used did not meet industry standards [113]. A mortgage company settled with the FTC for advertising that the data it collected was encrypted, when in reality it was encrypted in transit but not after receipt [56].

There is no precise technical specification that companies must meet to avoid being prosecuted for deceptive practices.[44] That said, under FTC precedent, there are clear lines that an E2EE messaging app could not cross without being found deceptive. If an app is marketed as using "end-to-end encryption," established FTC precedents demonstrate that the app must use end-to-end encryption throughout the processing of the relevant data [126, 56], have E2EE on by default [126], and meet industry standards [113]. In other words, the marketing of a service in Category 5 (No E2EE) defined in Section 3.2 with an unqualified claim that the product uses "end-to-end encryption" would be considered deceptive under FTC precedent. We stress, again, that the FTC has discretion over which cases it pursues and prioritizes—so despite the above, an app in violation of the above would not necessarily be investigated or suffer consequences.

The FTC is not the only governmental agency capable of investigating deceptive practices. Many state attorneys general have similar powers granted by local versions of the FTC Act.[45] State attorneys general are important consumer protection enforcers because they can address smaller or more local concerns that the FTC would miss or lack the bandwidth to address. They can also be significant regulators in their own right, and collaborate between states to take on the most sophisticated corporations. With uncertainty around the enforcement objectives of the FTC during the second Trump administration [91], state attorneys general are likely to take on even more importance as enforcers of consumer protection laws.

6.3 Overview of EU Legal Considerations for E2EE & AI

6.3.1 EU Legal Background

In many ways, legal obligations for data-driven services in the EU extend beyond the disclosure and consent model described above. Next, we overview key relevant EU legislation.

GDPR. The GDPR sets out key principles that impose additional constraints upon collecting, storing, and using personal data.[46] Under the GDPR, obtaining consent is not the only basis to legitimize the processing of user data—it is just one among six. Many user data-related activities within companies can rely on the broad alternative basis of "legitimate interests," that is, where processing is necessary for a "legitimate interest" pursued by the organization.[47] For instance, analyzing customer data for fraud prevention and signs of criminal activity or engaging in direct marketing with preexisting clients are typically deemed activities within a company's legitimate interests [155]. The legal basis of "legitimate interests" is often seen by organizations as more flexible and less cumbersome than obtaining user consent.[48]

This is because the GDPR sets higher standards for user consent: it must be "freely given," "specific," "informed," "unambiguous," and may be withdrawn (or opted out) at any time. Employing dark patterns

[44] But there is a roughly defined floor, below which companies can be investigated for *unfair* practices. This comes up often when companies are subjects of a data breach compromising unencrypted credit card, passport, or social security numbers [116].

[45] For example, the New York State Attorney General investigates deceptive practices under the Deceptive Practices Act, N.Y. Gen. Bus. Law § 349.

[46] For example, the "lawfulness, fairness and transparency", "purpose limitation" and "data minimization" principles are important guardrails enshrined in the GDPR. Regulation (EU) 2016/679 ("GDPR"), Article 5(1).

[47] Regulation (EU) 2016/679 ("GDPR"), Article 6(1)(f).

[48] Despite the requirement to ensure that organizational interests do not override users' fundamental rights (known as the "balancing test"). Regulation (EU) 2016/679 ("GDPR"), Article 6(1)(f).

to obtain user consent or to hide mechanisms to opt out may violate the GDPR and other overlapping regulations, such as the 2022 Digital Services Act (DSA) and the Unfair Commercial Practices Directive ("UCP" Directive) [64]. The European Data Protection Board (EDPB) provided examples of deceptive techniques used by social media platforms that do not qualify as "freely-given" consent [38]. These include "forced" consent, where refusing to consent leads to a denial of the service, and "bundled" consent, when consent is bundled with the acceptance of the platform's terms and conditions. These heightened protections are valuable; at the same time, the GDPR still offers an imperfect regime for protecting user rights and falls short of the ideals of freely given, unambiguous, informed consent based on mutual assent (e.g., [57]).

The GDPR is just one piece of data protection law in the EU. Other data protection legislation does mandate obtaining user consent in certain situations, as discussed next.

ePrivacy Directive. The ePrivacy Directive (Directive (EU) 2002/58 on Privacy and Electronic Communications) requires obtaining user consent for the surveillance of private communications[49] and the storing of non-essential browser cookies and other tracking technologies in user devices—a requirement that has reshaped the appearance of websites with the display of consent-collecting "cookie banners".[50] Article 5 of the ePrivacy Directive establishes a general prohibition on storing information or gaining access to stored information on users' end devices without obtaining prior consent, including listening, tapping, or other kinds of interception or surveillance of communications and related metadata.

As the ePrivacy Directive was originally designed to regulate traditional telecom operators, it is not completely clear whether it applies to "over-the-top" (OTT) communication services such as WhatsApp, Telegram and Zoom. In 2017, the European Commission adopted a proposal to replace the Directive with an updated ePrivacy Regulation that would explicitly cover OTT services [163]. Attempts to advance the reform received significant pushback from industry players, and were undercut by the passing of a temporary derogation that allows the scanning of private communications to detect online Child Sexual Abuse Material [112]. After being blocked for several years due to complicated trilogue negotiations in the EU Parliament, the ePrivacy Regulation proposal was withdrawn by the EU Commission in February 2025 [163] [52].

Despite the uncertainty facing potential reforms, the European Data Protection Board (EDPB) recommends that a new ePrivacy Regulation must under no circumstances lower the level of protection offered by the current ePrivacy Directive, as the confidentiality of electronic communications is a fundamental right that requires specific protection [41]. The EDPB further recommends that any exceptions must be narrowly tailored, and should not lead to the systematic monitoring of electronic communication content, nor allow providers or third parties to circumvent any encryption.

AI Act. The EU's recently enacted AI Act reinforces the applicability of both the ePrivacy Directive and the GDPR to AI systems, highlighting that these laws establish the basis for sustainable and responsible data processing.[51] Recital 69 of the AI Act notes that the rights to privacy and to protection of personal data—which are fundamental rights in the EU[52] —must be "guaranteed throughout the entire lifecycle of [an] AI system," and that measures taken by providers to ensure compliance may include anonymization, encryption, and "the use of technology that permits algorithms to be brought to the data and allows training of AI systems without the transmission between parties or copying of the raw or structured data themselves".

6.3.2 Training AI in the EU and the UK: Ongoing Controversies

Conditions for processing user data to train AI have been a contentious topic in the European Union and the United Kingdom—even for user data in technically public posts, let alone in more private contexts. Throughout 2024, tech companies like Google, Meta, X and LinkedIn have delayed or suspended plans to launch generative AI models in the EU due to regulatory inquiries into their training practices [55]. So far, these companies have been aiming to harness user-generated content to train their AI assistants or large language models and have not been openly targeting private messaging. Nevertheless, AI training practices have already been raising substantial privacy concerns in the EU.

Ireland's Data Protection Commissioner (DPC) has actively monitored the release of several AI-powered virtual assistants fueled by EU user data. The Irish authority is the European Union's lead regulator for

[49] Directive (EU) 2002/58 on Privacy and Electronic Communications ("ePrivacy Directive"), Article 5(1).
[50] Directive (EU) 2002/58 on Privacy and Electronic Communications ("ePrivacy Directive"), Article 5(3).
[51] Regulation (EU) 2024/1689 ("EU AI Act"), Recital 10 and Article 2(7).
[52] Charter of Fundamental Rights of the European Union (2000), Articles 7 and 8.

many US tech companies due to their EU headquarters being based in Dublin. In June 2024, the DPC raised concerns about Meta's privacy policy update that allowed Meta AI to be trained on public posts of EU users on Facebook and Instagram [63]. The DPC acted in response to a complaint filed by the advocacy group NOYB - European Center for Digital Rights. NOYB filed complaints in eleven countries urging regulators to act against Meta's "abuse of personal data for AI", drawing attention to the feature's lack of opt-in consent and hidden opt-out form [151].

Similarly, in August 2024, the DPC initiated proceedings against X (formerly Twitter) in the Irish High Court for concerns related to the training of X's AI assistant Grok [62]. A few months prior, users noticed a default setting on the app, collecting user consent via a pre-ticked box, allowing X to use posts, interactions, inputs and results for AI training [144]. The proceeding was dismissed after X agreed to suspend the use of publicly posted data on X to train Grok. In September, the DPC opened an inquiry into Google's use of European Union users' personal data to help develop its foundational AI Model, Pathways Language Model 2 (PaLM 2) [103]. The DPC also publicly confirmed that it has been in contact with Apple regarding its plans for Apple Intelligence AI [207]. At the broader industry level, an open letter coordinated by Meta and signed by executives from more than two dozen companies has criticized the EU's regulatory approach for generating uncertainty and for blocking innovation in the region [130] [5].

In late 2026, the Irish DPC issued a report on the "insights gathered" from "supervision of Artificial Intelligence products and services between 2021 and 2025," including an overview of their approach and a range of case studies, which directly opined on training on private message data [111]. The DPC report indicates that users cannot generally be expected to anticipate that their private messages might be used for training, and that relying on "legitimate interests" (GDPR Art. 6) as a legal basis for training on private messages would be "very difficult to achieve" [111, p. 51]. This accords with our analysis in this paper.

Other regulators have taken slightly different approaches. In contrast to the EU, plans to launch Meta AI were resumed in the United Kingdom after Meta's engagement with the UK Information Commissioner Office's (ICO) [139]. Under the UK GDPR,[53] Meta confirmed it will be relying on "legitimate interests"—not user consent—as a legal basis to train Meta AI on the public posts and comments of users above the age of 18. The ICO mandated certain measures from Meta, including making it simpler for users to object to the data processing and providing them with a longer window to do so, as well as providing more transparency about how the data is being used, and highlighted that it did not provide final regulatory approval and that Meta is bound to demonstrate "ongoing compliance" [154]. Meta confirmed it will be sending in-app notifications to inform users about the feature and maintains that private messages are not within the scope of training data [90].

Regulator inquiries and guidelines from EU member states have typically operated on a case-by-case basis, taking into consideration each AI system's business model and operational specifications. The European Data Protection Board (EDPB), which oversees the consistent application of data protection rules across the EU, began harmonizing the GDPR's interpretation regarding AI training after the launch of ChatGPT. Noting that the Italian regulator took enforcement action against OpenAI due to several GDPR violations, the EDPB launched a dedicated task force to investigate privacy issues in the ChatGPT service, which issued a report in 2024 [40]. The main findings concerning AI training were based on OpenAI's practice of web scraping (the automatic extraction of data from third-party platforms). This practice is different from obtaining data posted by users on AI companies' own platforms, as Meta AI and Grok's processing operations currently do. Nevertheless, some of the EDPB's recommendations may be more broadly applicable beyond the specific case of OpenAI and ChatGPT, such as the requirement that users be clearly and demonstrably informed that any prompts, including text queries and file uploads, may be used for training purposes.

Aiming to trigger more comprehensive discussions at the supranational level, the DPC requested the EDPB for a supervisory opinion on relevant issues when processing personal data to develop and train AI models [62]. The EDPB's Opinion 28/2024 was published in December 2024, focusing specifically on (1) when and how an AI model can be considered as "anonymous"; (2) how organizations can demonstrate the appropriateness of legitimate interest as a legal basis in the development and (3) deployment phases; and (4) consequences of the unlawful processing of personal data in the development phase of an AI model on subsequent processing operations [39]. With regard to balancing organizations' "legitimate interests" and user rights, the EDPB highlighted the importance of transparency and meeting users' reasonable expectations

[53]Following Brexit, the UK retained a copy of the GDPR in domestic law, which is known as the UK GDPR.

considering the complexity of the technologies and the variety of potential uses of personal data. The EDPB did not address user consent; the EDPB noted that the opinion does not cover all EU data protection issues in the context of AI development and deployment.

In November 2025, amid increasing industry pressure, the European Commission submitted two "Digital Omnibus" regulatory proposals for legislative review, proposing targeted amendments to the GDPR, e-Privacy rules, and the AI Act [67]. The proposals aim to ease development of AI models by allowing companies to use personal data (including certain sensitive data) to train AI based on business interests rather than user consent, simplifying consent-heavy rules, and delaying or lightening certain AI Act obligations [173].

Importantly, while the EU is openly debating the lawfulness of AI training under different conditions, none of the regulator inquiries or opinions to date address the issue of using data obtained from messaging or end-to-end encrypted services. While data posted on social media platforms under public settings may generally be considered public, messaging data is private by nature.[54] Legal regimes typically recognize that private communications must be treated to a higher standard of privacy protection because of their more intimate and vulnerable qualities, as embodied by the EU's ePrivacy directive.[55] Thus, the use of private messages shifts the paradigm and necessarily calls for more comprehensive debates around user consent. Moreover, training on what is advertised or commonly thought to be E2EE data may frustrate users' legitimate expectations as to the privacy and security of their data.

Although the regulation of AI training in the EU is still in flux and conclusions are uncertain, the recent inquiries in the EU and the UK have indicated that simply updating the company's privacy policy to allow AI training is insufficient to comply with GDPR standards. Besides providing users and regulators with proactive transparency and disclosure, the substance of how companies are collecting personal data, the types of data harnessed and how they are being used also matter. If private messages are used as training data, questions related to the indispensability of user consent, adequate interfaces for collecting consent and options to exercise control over data become substantially more pressing.

6.4 Other Jurisdictions

The EU GDPR is seen by some as the gold standard for online privacy protection [195]. Multiple jurisdictions have enacted new regulations or updated existing privacy laws, many of which emulate the GDPR's provisions, indicating a global trend to increase data regulation. Due to the GDPR's influential role, European regulators' stance on AI training issues will likely trigger ripple effects across the globe [44].

For instance, following the Irish regulator's reaction to Meta AI, Brazil's data privacy regulator (National Agency for Data Protection (ANPD)) similarly suspended Meta's privacy policy update allowing the training of Meta AI on the data of Brazilian users [77]. Brazil's first comprehensive data protection regulation (the "LGPD") was enacted in 2020 and is heavily inspired by the GDPR. Brazil is reported to be WhatsApp's second biggest market worldwide, with more than 124 million active users [188]. The ANPD raised concerns as to Meta's legal basis for processing sensitive and non-sensitive data, transparency in how the changes in the privacy policy were communicated, and accessibility to exercise the right to opt out.

The regulator's technical report also noted discrepancies in Meta's standards for EU and Brazilian users, despite similarities between the GDPR and the LGPD. Unlike EU users, Brazilians were required to go through eight steps to access an opt-out form hosted in a complex interface [26]. Additionally, while EU users were notified about the privacy policy update via email and app notifications, Brazilian users were not directly informed and were only able to access the update via Meta's Privacy Policy Center. After several rounds of negotiations and subsequent fixes carried out by Meta, the ANPD lifted the suspension [78]. Meta AI was launched in Brazil in October 2024, although the feature is still undergoing full regulatory inspection.

Meta has recently faced severe enforcement actions for privacy violations in other jurisdictions that have updated their privacy laws to be more robust. In November 2024, South Korea's privacy regulator fined Meta 21.6 billion won (roughly 14 million euros) for illegally collecting sensitive personal information from Facebook users, including data about their political views and sexual orientation, and sharing it with

[54] There are also nuances regarding the public nature of publicly posted data: "(...) the mere fact that personal data is publicly accessible does not imply that 'the data subject has manifestly made such data public'. In order to rely on the exception laid down in Article 9(2)(e) GDPR, it is important to ascertain whether the data subject had intended, explicitly and by a clear affirmative action, to make the personal data in question accessible to the general public." See [40].

[55] In the US, the Electronic Communications Privacy Act of 1986 protects wire, oral, and electronic communications from unauthorized interception, disclosure, or use, including email, phone calls, and data stored electronically.

thousands of advertisers without users' express consent [87]. In July 2023, Meta subsidiaries were fined by the Australian Competition and Consumer Commission a total of $20 million Australian dollars for misleading consumers about the privacy protections in Meta-owned VPN app Onavo Protect [68]. Although the app was marketed as a privacy tool in app stores, it gathered extensive user data for Meta's commercial purposes.

These enforcement actions show a trend of increased regulatory oversight over digital corporations' data practices stemming from legal privacy obligations. As corporations are faced with multiple cumulative regulatory standards, they may be encouraged to conform to the combination of the most stringent regulations provided by different jurisdictions [44]. However, different jurisdictions will exercise varying degrees of leverage over tech companies based on the company's market share and number of users by country or region, as well as the local law's penalty provisions and the value of the currency.

6.5 Privacy, Antitrust and Consumer Data

Big tech corporations are increasingly at the center of discussions about both antitrust and data protection regulation. Access to vast amounts of consumer data is a strategic asset that may reinforce online platforms' dominant positions by creating barriers to entry for competitors, reducing consumer choice, or unfairly manipulating markets [59]. Mergers and acquisitions between tech corporations may involve the merging of databases and data sharing practices in violation of privacy laws [153], at times blurring the lines between privacy and antitrust concerns. We briefly provide two illustrative examples.

At the time of Meta's acquisition of WhatsApp in 2014, the European Commission approved the acquisition [65] with the understanding that Facebook would be unable to do automated matching between Facebook users' accounts and WhatsApp users' accounts [66]. However, in August 2016 WhatsApp announced updates to its terms of service and privacy policy including the possibility of linking user data across WhatsApp and Facebook. The aftermath resulted in a €110 million fine on Meta (formerly Facebook) for providing misleading information under EU merger regulation [66]. Had a separation between WhatsApp and Facebook data been enforced beyond levying a fine, this antitrust measure could have had far-reaching implications for the integration of certain operations across WhatsApp and Facebook.

The EU's Digital Markets Act (DMA), applicable since May 2023, is a landmark antitrust regulation aimed at curbing Big Tech's market dominance and ensuring fair competition between digital services. The DMA recognizes privacy and data protection as key principles and specifically regulates the prevention of cross-service data merging in Article 5(2).[56] This provision states that gatekeepers cannot combine personal data from different services unless they obtain explicit opt-in consent from users.

Outside the EU, in November 2024, India's antitrust regulator targeted Meta for anti-competitive behavior related to data sharing [145]. The Competition Commission of India (CCI) imposed a penalty of roughly 25.4 million dollars on Meta after WhatsApp users were forced to to share their data with other Meta-owned companies by means of a 2021 privacy policy update with no opt-out mechanism [172]. The regulator also ordered WhatsApp not to share user data for advertising purposes with other Meta-owned applications for five years, highlighting that the data sharing resulted in denial of market access for competitors.

These examples illustrate how data privacy decisions can lead to broader antitrust concerns beyond the scope of privacy and vice versa. While companies often have monetary incentives to access larger pools of data and integrate features throughout a broad range of products, this may come at the expense of higher scrutiny by regulators. This is a particularly timely observation as the past four years have seen a wave of antitrust lawsuits against Big Tech initiated by the US and the EU [136]. That said, the 2025 government administration changes in the US may change priorities at the FTC, which is a prominent enforcer of both antitrust and privacy laws.

7 Recommendations

This section presents our recommendations for integrating AI assistants into E2EE applications.

Our recommendations address both compatibility with E2EE from a *security perspective* (Recommendations 1 and 2, discussed in Section 7.1), and disclosure and consent from a *legal and normative perspective*

[56] Regulation (EU) 2022/1925 ("Digital Markets Act - DMA"), Article 5(2).

(Recommendations 3 and 4, discussed in Section 7.2). Appendix B applies our recommendations to Apple Intelligence as a case study.

As a recap, the bullet-point summary of our recommendations is as follows.[57]

1. **Training.** Using end-to-end encrypted content to train *shared AI models* is not compatible with E2EE.
2. **Processing.** Processing *E2EE content* for AI features (such as inference or training) may be compatible with end-to-end encryption only if the following recommendations are upheld:
   (a) Prioritize *endpoint-local* processing where possible.
   (b) No *third party* can see or use any *E2EE content*[58] without breaking encryption,[59]
   (c) A user's *E2EE content* is exclusively used to fulfill that user's requests.
3. **Disclosure.** Messaging providers should not make unqualified representations that they provide E2EE if the default for any conversation is that *E2EE content* is used (e.g., for AI inference or training) by any *third party*.
4. **Opt-in consent.** AI assistant features, if offered in E2EE systems, should generally be off by default and only activated via opt-in consent. Obtaining meaningful consent is complex, and requires careful consideration including but not limited to: scope and granularity of opt-in/out, ease and clarity of opt-in/out, group consent, and management of consent over time.

7.1 Compatibility with E2EE Security

Recommendation 1 (Training) *Using end-to-end encrypted content to train* shared AI models *is not compatible with E2EE.*

As discussed in Section 4.2, training a non-endpoint-local *shared* model with E2EE data is incompatible with the security guarantees of E2EE encryption; regardless of how the data reaches the model or how the model is stored, the fact that other users can query a model that is a function of other users's E2EE data is not compatible with E2EE confidentiality. In practice, this can lead to leakage of sensitive data or malicious extraction via adversarial attacks.

It may be helpful to consider the following simpler model, based on an analogy. We can think of AI models as a "bundle" of the training data that went into them, which is somehow encoded in the values of its parameters—not necessarily perfectly encoded,[60] but captured well enough that the model's outputs are highly dependent on this bundle of data. Then, an AI assistant trained on E2EE messaging data can be thought of as a *bundle of private messages*. The model receives queries from users, performs a computation on this bundle of messages, and outputs a response. Through this lens, we can see that entities with access to the model itself—for example, the service provider, or if the model is stored directly in end-user devices—now have access to (a bundle of) messages not intended for them—a bundle of E2EE content—violating E2EE confidentiality. Further, even if no entity has direct access to this bundle (for example, if the model is somehow encrypted), any user can still *perform computations* on this bundle of messages, and receive outputs that depend on it—which are E2EE content, being derivatives of E2EE plaintext messages—with just query access to the model. In other words, model outputs can be thought of as being "tainted" by any E2EE content in the underlying training data, and thus models trained on E2EE content produce outputs that are necessarily dependent on these training data and thus are themselves E2EE content. So, training shared AI assistants with E2EE data *definitionally* undermines the confidentiality guarantees of E2EE technologies.

Recommendation 2 (Processing) *Processing* E2EE content *for AI features (such as inference or training) may be compatible with end-to-end encryption only if the following recommendations are upheld:*

[57] This bullet-point summary of our recommendations is identical to that in Section 1.

[58] Or any derivative of E2EE content (i.e., any non-trivial function of content). See Section 3.1 for full definitions.

[59] This may sound paradoxical; see Section 4.2 for detailed discussion of technical configurations that may achieve this condition.

[60] E.g., the bundle may be disorganized, and it may be difficult to extract every single piece of training data on demand.

1. *Prioritize* endpoint-local *processing where possible.*

2. *No* third party *can see or use any* E2EE content[61] *without breaking encryption,*[62]

3. *A user's* E2EE content *is exclusively used to fulfill that user's requests.*

Application developers should prioritize local inference whenever possible, as this option is fully E2EE-compatible. To maintain E2EE confidentiality in non-endpoint-local inference, it is critical that no third party (including the model provider) handles any E2EE data unencrypted.[63] AI assistants hosted on a TEE typically retain the ability to interact with third party servers, including conveying plaintext information (e.g., through tool use), so do not meet this standard.

If exploring TEE-based approaches, service providers should not describe these as E2EE (see also Recommendation 3), and exercise great care around user expectations, consent, and unanticipated harms around transitioning a system from one type of security guarantee to another (see also Recommendation 4).[64]

7.2 Towards Meaningful Disclosure and Consent

In practice, legal consent and disclosure are often analyzed together. For the purposes of this section, we distinguish disclosure as the set of texts in advertising and user interfaces that the user is likely to read, despite formally not being part of a service's privacy policy or ToS. This allows us to recommend changes to the set of communications that may actually change some individuals' behavior. This kind of disclosure can also be described as one-way communication from the service to the user.

Consent, then, encompasses assent to the formal terms of service and privacy policies that bind the user and service in a contractual agreement. Opt-in consent mechanisms are those which allow users to take an affirmative step to agree to legal terms, such as pop-up banners for clicking a button or checking a box. These mechanisms may display brief disclosures informing users of the most important terms and may link to the pages where the complete terms and policies are displayed. Conversely, opt-out mechanisms are interfaces that provide consumers with a streamlined process to opt out of certain data processing activities after consent has been given, or as an alternative to the initial opting-in. Opt-out can therefore be considered withdrawal of consent. While users may not read all the fine print (see Section 5.1), the content contained in terms of service meaningfully affects users' rights.[65]

Next, we present our Recommendations 3 and 4, which are respectively focused on disclosure and consent.

Recommendation 3 (Disclosure) *Messaging providers should not make unqualified representations that they provide E2EE if the default for any conversation is that* plaintext-dependent *versions of* E2EE content *is used (e.g., for AI inference or training) by any* third party*.*

"Unqualified representation" is a term often used in consumer protection law. It generally refers to a statement made without caveats, and, in this paper, excludes caveats made in fine print, such as footnotes or terms of service. So, a splash page for an E2EE messaging service which boldly states "E2EE Used for All Messages*" would be making an unqualified representation, even if the asterisk points to a footnote that says "Unless opted out by user." or "Except when AI features are used."

Put more simply, Recommendation 3 says that messaging services should not make unqualified claims that they are using E2EE when their implementation does not achieve E2EE confidentiality.

[61] Or any derivative of E2EE content (i.e., any non-trivial function of content). See Section 3.1 for full definitions.

[62] This may sound paradoxical; see Section 4.2 for detailed discussion of technical configurations that may achieve this condition.

[63] The one cryptographic tool that might permit encrypted processing, FHE, is not currently practical for this use case. See Section 4.2 for more detail.

[64] As discussed in Section 2.4.2, TEE security is different from encryption security: a system that relies on TEE security for message confidentiality does not achieve E2EE security and cannot accurately be described as offering E2EE.

[65] Using these definitions, the line between disclosure and consent can be blurry in some situations. We include terms of service and privacy policies under the scope of consent and not disclosure. Elements of a service's chat interface, like pop-up notifications, are included in disclosure, but not necessarily consent. We treat advertising, which is generally not considered binding on a service, as a disclosure. Cookie banners, ubiquitous since the passage of GDPR, overlap both disclosure and consent: although easily dismissed by many consumers, they are prominently displayed on websites and contain non-legal language designed to be legible, a common characteristic of disclosures. Consumers then select their cookie preferences, making a two-way communication characteristic of consent.

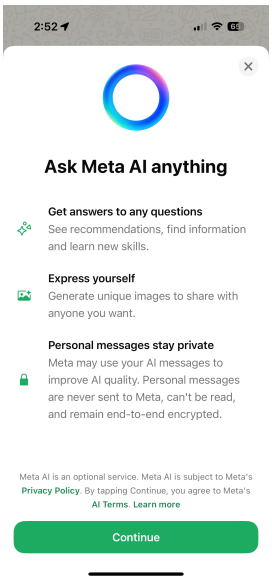

Figure 7: WhatsApp's disclosure upon a user's first use of Meta AI as of 15 November 2024

Moreover, as discussed in Section 6, to avoid regulator scrutiny for deceptive practices, messaging services hoping to make AI available to or train AI in E2EE contexts should err on the side of over-disclosure.

The requirements are clear for many possible implementations. A service with a configuration which falls into Category 1 (as defined in Section 3.2) would be able to advertise itself as using E2EE with no qualifications or caveats. On the other hand, services with configurations which fall into Categories 3–5 would need to make *qualified* claims regarding their use of E2EE. Most Category 2 services will require additional, specific disclosures based on their features.

For example, in WhatsApp, many points of the advertising and app interface indicate that "text and voice messages" are E2EE. Many users may see messages invoking Meta AI, which are not encrypted, as a "text or voice message," and thus expect that their requests to Meta AI *are* E2EE. Meta reinforces this false, but natural, misunderstanding. As shown in Figure 7, when entering a Meta AI chat for the first time, WhatsApp users are shown a screen which states "Personal messages stay private. Meta may use your AI messages to improve AI quality. Personal messages are never sent to Meta, can't be read, and remain end-to-end encrypted." The scope of "personal messages" is critical, and ambiguous, in this disclosure. A sufficient disclosure would explicitly state that the messages in the following conversation are not encrypted and are sent to Meta, and as such should not include sensitive information.

Besides clarity and accuracy, another important component of disclosure is giving users notice of said disclosures. In other words, services should ensure that their disclosures are actually reaching users. Since E2EE services often do not require an email address to sign up, relevant disclosure updates should be sent to all users via the service itself.

Recommendation 4 (Consent, opt-in and opt-out) *AI assistant features, if offered in E2EE systems, should generally be off by default and only activated via opt-in consent. Obtaining meaningful consent is complex, and requires careful consideration including but not limited to: scope and granularity of opt-in/out, ease and clarity of opt-in/out, group consent, and management of consent over time.*

Despite normative critiques highlighting the insufficiency of consent (see Section 5.2.2), consent is still the main legal tool currently available for giving users greater control over their privacy rights. Implementing consent interfaces well can make an important difference.

We recommend that opt-in consent be the standard mechanism for allowing messaging services to train AI on user data. Consent for using messaging services must not be conflated with consent for additional features such as AI assistants. Consent for separate features should be collected in separately, each with a specifically tailored opt-in banner or other form of disclosure. Although the banner should refer to a more

detailed privacy policy and/or terms of service, there should be enough information on the banner to meet previously mentioned disclosure requirements.

AI tool use is often opaque to users, but notice and consent for tool use is important for security and privacy as it may involve disclosure of E2EE or otherwise "protected" information to third parties.

Messaging services should recognize the inherently private nature of communications, even (or especially) in environments that mix E2EE and non-E2EE data. Opt-in consent banners should serve the dual purpose of (i) clearly informing users about the exact scope of data that will be used for training purposes, including whether the data is protected by E2EE or not (or by some other security mechanism such as a TEE); (ii) providing users with a real choice, making sure to disclose how their choices may affect E2EE protection. It should be made clear to users which data is kept under E2EE (or TEE) protection, including any differences between individual and group settings. Additionally, terms of service and privacy or security policies should accurately and adequately describe security-relevant implementation details and associated risks of enabling the AI assistant feature.

Beyond legal considerations, opt-in consent for AI features better preserves the application's systemic E2EE security properties (Section 2.1.3): user inaction results in the unaltered functioning of E2EE protections, while only users who explicitly want the E2EE-violating feature will choose to have it enabled.

Finally, digital services should also provide clear opt-out mechanisms upon signing up and when new features are added, including a clear disclosure of what features still work if a user opts out. Each option should carry its own opt-out mechanism. A clear opt-out mechanism is one that is not concealed by dark patterns or unnecessary obstructions.[66] As a rule of thumb, opting out or withdrawing consent for services to which users have previously accepted must be made as easily accessible as the initial consent for using the feature. When possible, opting out from allowing AI training on one's data should not be a precondition to the use of an entire app or to the use of AI features in the app. Relatedly, services should consider separating the opt-in/opt-out for use of an AI feature and for training based on such use. Especially regarding AI training, it must be clearly disclosed that the opt-out is forward-looking: data which the company has already used for training will remain in their models, regardless of any opt-out exercised.[67] Finally, consent should be subject to periodic renewals and opt-in banners should be re-validated after reasonable periods of time.

We apply our recommendations to the case of Apple Intelligence in Appendix B, as we found that the documentation currently available on Apple Intelligence supports a more detailed analysis than for the other two products. We leave the other two examples—as well as future releases of these and other products—as an exercise for the reader.

8 Discussion

8.1 Crypto Wars 3.0

Law enforcement agencies and the global intelligence community have sought over decades to both limit the use of strong encryption and exploit the naturally occurring weaknesses of encrypted systems [68] [33]. These efforts are in response to the technical reality that end-to-end encryption provides confidentiality to anyone who uses it, including criminals. The so-called "Crypto Wars" can be described as the confrontation across regulatory, corporate, and research settings between state power and encryption proponents to determine whether there exists a technical compromise that would facilitate "lawful access" to confidential data [147]. The Crypto Wars have evolved over the years with encryption, and technical proposals have evolved from pushing for weak cryptographic standards to various forms of key escrow [1]. The latest proposed technological solutions are framed as sophisticated alternatives to backdoors, which would preserve strong encryption algorithms while accessing E2EE content in other ways, namely on the device (the "client") and in the cloud using cryptographic computational techniques not requiring decryption.

What do backdoors, client-side scanning and homomorphic encryption have to do with AI assistants? Like data-driven algorithms, lawful access is incompatible with the privacy and confidentiality of end-to-end encryption, especially when users are uninformed of the risks. Furthermore, both are extensions or features of private messaging applications, not meant to enhance privacy and confidentiality; rather, they are designed

[66] For instance, requiring users to navigate an eight-step process to access the opt-out form is not a clear mechanism [26].
[67] Implementing retroactive withdrawal is currently practically infeasible [92].
[68] Very recent developments suggest an interesting possible shift, though it is too soon to be clear [76, 118]. See also [27].

to undermine it by enabling access to data that would not otherwise be accessible. A recurring concern in the Crypto Wars has been the risk of large-scale, hard-to-detect abuse and compromise: Even if we all agreed that AI assistants are beneficial, if E2EE data can be accessed by AI assistants for good, that method of access could also be used to track and target users, undermining the core of the security guarantee that E2EE provides.

8.2 Business Incentives to Integrate AI (and Fundamental Rights)

Companies with data-driven business models may see messaging data as a valuable resource for refining AI algorithms, and the temptation to use this data will likely grow as AI assistants become increasingly embedded in everyday devices. Messaging data is a source of largely untapped human-authored text at a time when concerns are mounting about the scarcity of such data; providing E2EE hinders the platform's ability to collect and analyze this data.

We are concerned that this new business incentive may eclipse the existing business incentives to protect user data, which are better aligned aligned with human rights protections. In what appears to be the new business model for Apple—which for years built its brand around privacy—and its competitors, services are seeking out and leveraging large sources of user data at a pace that does not permit thorough community deliberation over associated risks and externalities, even in the absence of clear user demand, and in the face of user concerns [54, 117].

8.3 But Can't I Just Paste That Into ChatGPT?

Of course, a user can always take their E2EE messages and paste them into an AI assistant. That doesn't violate E2EE confidentiality. So why are we making such a big deal about the idea of E2EE content being sent from a user's device, where the user has them decrypted anyway, to an integrated AI assistant?

At a glance, integrating AI features in an E2EE messaging app might seem like the same thing as pasting one's own messages into an AI assistant—just streamlined, which might be a convenient benefit for the user. In fact, we believe it is different in many ways, though a full exposition is beyond this paper's scope; interested readers may refer to the theories of privacy briefly discussed in Section 5.2.3 for additional context.

In a nutshell, the two key differences are in the *service provider's promise* about an E2EE application—that it does not access or process E2EE content, that it designs its systems to uphold E2EE confidentiality—and in the *systemic nature of the breach of confidentiality*. If a user pastes their E2EE messages into an AI assistant, or indeed if they post them publicly online, that is a intentional act undertaken by an individual that may be a breach of social norms (depending on the context) but neither breaches the E2EE *service provider's promise* or creates a *systemic breach* of confidentiality by which similar processes can happen as a matter of course, as part of a regular app feature that is easily accessible and whose frequent use is encouraged by the very service provider who offers the E2EE.

8.4 Confer

Confer, released by Moxie Marlinspike and acquired by Meta AI for "incognito chats" in Whatsapp,[148] proposes what it describes as "end-to-end encryption for AI chats." [131] Confer focuses on *adding E2EE to AI chats*, which currently are not E2EE by default—that is, using cryptographic protections to strengthen existing security practices of AI systems, which typically are weaker than E2EE.

In contrast, our paper's focus is on *maintaining the existing guarantees of E2EE applications*, such as messaging where end-to-end encryption is on by default, when AI is introduced within the application environment. That is, we are concerned with preventing degradation of strong existing security guarantees.

These are different contexts and security design goals, and thus we consider Confer to be largely outside of our present scope. Our main point here is the importance of distinguishing these contexts clearly. Below, we briefly elaborate the relationship and differences between our focus and approach and those of Confer.

First, Confer relies both on encryption security and on TEE security, treating the server-side TEE where queries are processed as an "endpoint" of its "end-to-end encryption" [132]. These definitions diverge from our narrower definitions: we would not consider a cloud server an "endpoint" even if TEE-protected and thus we would not call this system "end-to-end encrypted." In conventional E2EE systems, endpoints correspond

to user-controlled devices holding private keys. Even if a cloud server runs inside a TEE, we would not characterize it as an endpoint in the context of E2EE, and the security guarantee would not be E2EE as we define it: confidentiality against the service provider depends not only on cryptography but also on hardware isolation, vendor trust assumptions, and remote attestation mechanisms. We would instead describe Confer as promising example of *TEE-mediated private inference*.

Second, *if* Confer were integrated with an E2EE messaging application, then like most other AI assistant integrations, it would break the E2EE guarantee that only parties to the conversation could decrypt message contents, as discussed in Section 4. Introducing a server-side inference endpoint—even one protected by a TEE—changes the trust model and potentially expands the set of entities capable of processing plaintext.

To our knowledge, there is no suggestion that Confer be integrated with any E2EE applications; this is a hypothetical that we address for completeness. The currently available Confer application is a standalone web-based AI chat interface, and current documentation does not discuss integration across apps and devices, how that would work, or whether such integration is expected or intended at all. Whether and how such integration would change the nature of the baseline security guarantee offered by the underlying apps or devices would depend on context. There is no equivalent issue in the context of standalone AI chat applications, as they are generally not E2EE to begin with, so no there is comparable downgrade risk.

TEE-based private inference may be a promising architectural pattern for certain AI applications, particularly where usability and server-hosted computation are primary design constraints. However, as discussed at length earlier in the paper, the security guarantees offered by TEE-based systems differ importantly from those of cryptographic E2EE. Conflating these models risks obscuring important differences in threat assumptions, trust boundaries, and user expectations.

8.5 Additional Security Concerns: Message Integrity

This work primarily focuses on the interplay between AI assistants and the confidentiality and systemic guarantees of E2EE. Yet, as discussed in Section 2.1.2, modern E2EE applications aim to provide additional security properties beyond these two. An important direction for future work is to perform an analysis of the (in)compatibility of AI assistants and these other guarantees. For any security property that an E2EE application claims to provide, it is critical that the implementation of AI assistants upholds this guarantee.

Integrity is another standard security guarantee of E2EE applications, which states that a message received by the recipient should be equal to the one sent by the sender. AI *summary features* could be in tension with integrity: incorrect summaries of messages could be considered a violation of this guarantee, as they display an inaccurate representation of the message to the recipient. While the original (unmodified) messages are still visible in the chat history, incorrect summaries may still cause unintended consequences akin to if the messages were modified in transit. Incorrect summaries may occur accidentally, as a result of imperfect AI features. For example, in February 2025, the BBC issued a research report finding that language models of the time were unable to accurately summarize news articles, generating answers with "significant inaccuracies and distorted content," such as incorrectly claiming that the NHS did not recommend vaping as an aid for quitting smoking [171, 32]. A user relying on an AI summary of an article could be completely misled as to the contents of the article, representing a violation of integrity. Issues with AI summaries could therefore be exploited as a vector for deliberate attacks.

Whether and in what contexts incorrect AI summaries should be cast as a violation of integrity is an interesting question outside the scope of this work. The analysis is more subtle than for confidentiality, since integrity violations can in principle be detected by users by inspecting the original messages in the full chat history—in contrast, once E2EE content has been disclosed to third parties, it is not possible to backtrack. Exploring this question may require a broader socio-technical treatment that analyzes normative expectations of integrity. We anticipate that such tensions between integrity and AI assistants may become more relevant as AI summary features become more ubiquitous. As AI assistants improve and missummarizations become less frequent, the possibility of integrity violations only magnify in gravity, as users may come to expect faithfulness in AI summaries.

8.6 Usable Security for E2EE and AI Assistants

Tensions between usability and security have long been known (e.g., [74, 217]) and the field of *usable security* studies how to reconcile them to build secure systems that are also usable [74]. Even the best security in the world is not practically useful if poor usability will prevent adoption — and adoption is an especially important consideration in contexts such as E2EE where wide usage in practice contributes to the strength of security guarantees (as discussed in §2.1.3). We recall from Section 2.1.5 that (1) strict E2EE is realized almost nowhere in practice because many expected and necessary features of messaging applications are not possible without violating E2EE; (2) pragmatically, it is valuable for E2EE messaging applications to include such features to entice users to use the applications (over, say, less secure alternatives); and that (3) the practically realized privacy of an E2EE application depends, in part, upon having a sufficiently large userbase to make use of the app non-deanonymizing, thus making usability a privacy-relevant consideration.

In the context of E2EE and AI assistants, usable security requires a delicate balance, enabling enough convenient features (possibly including AI features) to support widespread usage in practice, even if such features are not ideal for security, while still enforcing high security standards for as many users as possible.

We anticipate that similar considerations may arise in the context of AI tool use: many tools involve interaction with third parties and therefore necessarily break E2EE guarantees (see §4.2).[69] Different tools have different security implications. For example, an agent with access to an API for retrieving restaurant reviews may not be capable of leaking any sensitive information in using the API. In contrast, an agent which may browse the entire internet could potentially leak sensitive information through its browsing behavior. This tradeoff must also be negotiated in light of users' views on the utility of a model having access to tools.

AI assistants and tool use are still relatively new and expectations around them are still evolving. Depending on how societal and user expectations develop, it may become an important exercise in usable security to weigh the usability benefits of a range of AI assistant features, including tool use, against the security degradation they entail, in order to design E2EE or other secure applications to provide realistic security and privacy protection to as many users as possible in practice. Meanwhile, while such expectations are still developing, we recommend the conservative approach of protecting systems against degradation of existing security guarantees by these emerging technologies.

9 Conclusion

AI models are being developed rapidly and integrated across devices and applications, including in contexts where they may process end-to-end encrypted or otherwise private content. In this work, we performed a detailed examination of complementary technical and legal tensions inherent to integrating AI models with E2EE applications, identifying some serious risks and concerns. Perhaps the greatest risk is in eroding E2EE as a systemic guarantee—a condition of any application or service that is held out to be E2EE—and turning it instead into merely a bonus feature that is routinely compromised in exchange for convenience.

Development of AI features in E2EE environments should be accompanied by informed multidisciplinary discussions on potential harms. Our work represents a first step towards this; we hope this conversation continues, involving free speech, privacy, and human rights advocates, civil society representatives, and others outside academia, to develop frameworks and standards for the responsible deployment of AI and E2EE. It is critical for these debates to occur promptly, before unilateral industry rollout.

Author contributions

MK had the initial idea and co-led the overall structuring of the paper and recommendations with SP. AF, BLH, and DY contributed equally to designing and writing technical background, implications, and recommendations. DF and JL contributed equally to designing and writing legal background, implications, and recommendations. SA led the case studies and provided research support throughout other aspects of the paper. SP was faculty lead and KC provided consultation.

[69] Even outside of the E2EE security model, the security benefits of an AI model being hosted locally or on trusted hardware can be undone if the agent has access to tools.

Acknowledgments

We are grateful to Joseph Bonneau, Daniel Genkin, Peter Hall, Daji Landis, Julia Len, Megan Richards, Nathalie Smuha, and Michael A. Specter for helpful discussion.

This work was supported in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) with a grant funded by the Ministry of Science and ICT (MSIT) of the Republic of Korea in connection with the Global AI Frontier Lab International Collaborative Research Agreement (No. RS-2024-00469482 & RS-2024-00509258).

References

[1] Harold Abelson et al. "Keys under doormats: mandating insecurity by requiring government access to all data and communications". In: *Journal of Cybersecurity* (2015).

[2] A. Acquisti and J. Grossklags. "Privacy and rationality in individual decision making". In: *IEEE Security & Privacy* 3.1 (2005), pp. 26–33. DOI: 10.1109/MSP.2005.22.

[3] Alessandro Acquisti and Jen Grossklags. "What Can Behavioral Economics Teach Us About Privacy?" In: *Digital Privacy: Theory, Technologies and Practices* (2007). PERMALINK: https://perma.cc/56BA-YXXQ.

[4] *AI (and other) Companies: Quietly Changing Your Terms of Service Could Be Unfair or Deceptive*. https://www.ftc.gov/policy/advocacy-research/tech-at-ftc/2024/02/ai-other-companies-quietly-changing-your-terms-service-could-be-unfair-or-deceptive. PERMALINK: https://perma.cc/A2ZP-DWV7. 2024.

[5] Meta et. al. "Europe needs regulatory certainty on AI". In: *Open Letter* (2024).

[6] Mustafa Al-Bassam and Sarah Meiklejohn. "Contour: A practical system for binary transparency". In: *International Workshop on Data Privacy Management*. 2018.

[7] Sam de Alfaro et al. *AI Assistants: Breaking the Promise of End-to-End Encryption?* NYU DeTaIL Lab Blog. Mar. 2025. URL: https://nyudetail.substack.com/e2ee-ai-broken-promise.

[8] Anthropic. *Tool Use Overview*. https://platform.claude.com/docs/en/agents-and-tools/tool-use/overview. PERMALINK: https://perma.cc/TD54-UASX. 2026.

[9] Apple. *Apple Intelligence and Privacy*. PERMALINK: https://perma.cc/P2KS-QJ8L. 2025. URL: https://www.apple.com/legal/privacy/data/en/intelligence-engine/.

[10] Apple. *Apple Intelligence brings powerful AI capabilities into everyday experiences*. PERMALINK: https://perma.cc/8B47-3YFQ. 2026.

[11] Apple. *Apple Intelligence now features Image Playground, Genmoji, Writing Tools enhancements, seamless support for ChatGPT, and visual intelligence*. https://www.apple.com/newsroom/2024/12/apple-intelligence-now-features-image-playground-genmoji-and-more/. PERMALINK: https://perma.cc/N6XN-T4FQ. 2024.

[12] Apple. *Apple Platform Security*. Tech. rep. PERMALINK: https://perma.cc/LSK7-6ASF. 2024. URL: https://help.apple.com/pdf/security/en_US/apple-platform-security-guide.pdf.

[13] Apple. *Hardware Integrity*. https://security.apple.com/documentation/private-cloud-compute/hardwareintegrity. PERMALINK: https://perma.cc/8WZH-CZE2. 2024.

[14] Apple. *How to get Apple Intelligence*. PERMALINK: https://perma.cc/7MLJ-2L5T. 2026. URL: https://support.apple.com/en-us/121115.

[15] Apple. *How to get Apple Intelligence on iPhone*. PERMALINK: https://perma.cc/KBC3-AMFX. 2024. URL: https://support.apple.com/en-us/121115.

[16] Apple. *iCloud data security overview*. https://support.apple.com/en-us/102651. PERMALINK: https://perma.cc/EP89-TPED. 2024.

[17] Apple. *Integrating your app with Siri and Apple Intelligence*. https://developer.apple.com/documentation/appintents/integrating-your-app-with-siri-and-apple-intelligence. PERMALINK: https://perma.cc/W7D2-ZPBM. 2024.

[18] Apple. *Introducing Apple Intelligence, the personal intelligence system that puts powerful generative models at the core of iPhone, iPad, and Mac*. https://www.apple.com/newsroom/2024/06/introducing-apple-intelligence-for-iphone-ipad-and-mac. PERMALINK: https://perma.cc/YRT8-QPPA. 2024.

[19] Apple. *Introducing Apple's On-Device and Server Foundation Models*. https://machinelearning.apple.com/research/introducing-apple-foundation-models. PERMALINK: https://perma.cc/UB4P-BLRJ. 2024.

[20] Apple. *iOS & iPadOS 18.3 RC Release Notes*. https://developer.apple.com/documentation/ios-ipados-release-notes/ios-ipados-18_3-release-notes. PERMALINK: https://perma.cc/BK24-5QP8. 2024.

[21] Apple. *Legal — Ask Siri, Dictation & Privacy*. PERMALINK: https://perma.cc/83H3-ZFA7. 2024. URL: https://www.apple.com/legal/privacy/data/en/ask-siri-dictation.

[22] Apple. *Private Cloud Compute: A new frontier for AI privacy in the cloud*. https://security.apple.com/blog/private-cloud-compute/. PERMALINK: https://perma.cc/XU5M-DAS2. 2024.

[23] Apple. *Use Apple Intelligence with Siri on iPhone*. https://support.apple.com/en-gb/guide/iphone/iph17bafe0f6/ios. PERMALINK: https://perma.cc/R9XW-VYGL. 2024.

[24] Apple. *Use ChatGPT with Apple Intelligence on iPhone*. https://support.apple.com/en-mo/guide/iphone/iph00fd3c8c2/ios. PERMALINK: https://perma.cc/UC3Q-WRCX. 2024.

[25] Brooke Auxier et al. *Americans and Privacy: Concerned, Confused and Feeling Lack of Control Over Their Personal Information*. Pew Research Center, 2019.

[26] Maria Badillo. "Processing of Personal Data for AI Training in Brazil: Takeaways from ANPD's Preliminary Decisions in the Meta Case". In: *Future of Privacy Forum* (2024). PERMALINK: https://perma.cc/24GE-ABNN.

[27] Jim Baker. *Rethinking Encryption*. https://www.lawfaremedia.org/article/rethinking-encryption. PERMALINK: https://perma.cc/H2XZ-3SU7. 2019.

[28] Talal Haj Bakry and Tommy Mysk. *Link Previews: How a Simple Feature Can Have Privacy and Security Risks*. PERMALINK: https://perma.cc/32QY-US99. 2020. URL: https://www.mysk.blog/2020/10/25/link-previews/.

[29] Jack M. Balkin. "Information Fiduciaries and the First Amendment". In: *UC Davis Law Review* (2016). PERMALINK: https://perma.cc/7MD4-HV79.

[30] Ero Balsa, Helen Nissenbaum, and Sunoo Park. "Cryptography, Trust and Privacy: It's Complicated". In: *CSLAW*. Ed. by Daniel J. Weitzner, Joan Feigenbaum, and Christopher S. Yoo. 2022.

[31] Oren Bar-Gill, Omri Ben-Shahar, and Florencia Marotta-Wurgler. "Searching for the Common Law: The Quantitative Approach of the Restatement of Consumer Contracts". In: *University of Chicago Law Review: Vol. 84: Iss. 1, Article 2* (2017). PERMALINK: https://perma.cc/5MF2-E7WP.

[32] BBC. *Representation of BBC News content in AI Assistants*. https://www.bbc.co.uk/aboutthebbc/documents/bbc-research-into-ai-assistants.pdf. PERMALINK: https://perma.cc/6DA6-XRSZ.

[33] Steven M. Bellovin et al. "Going Bright: Wiretapping without Weakening Communications Infrastructure". In: *IEEE Security & Privacy* (2013).

[34] Steven M Bellovin et al. "It's Too Complicated: How the Internet Upends Katz, Smith, and Electronic Surveillance Law". In: *Harv. JL & Tech.* (2016).

[35] Katie Benner and Paul Mozur. *Apple Sees Value in Its Stand to Protect Security*. https://www.nytimes.com/2016/02/21/technology/apple-sees-value-in-privacy-vow.html. PERMALINK: https://perma.cc/8QR5-KWLD. 2016.

[36] Katie Benner and Nicole Perlroth. *How Tim Cook, in iPhone Battle, Became a Bulwark for Digital Privacy*. https://www.nytimes.com/2016/02/19/technology/how-tim-cook-became-a-bulwark-for-digital-privacy.html. PERMALINK: https://perma.cc/F3AF-N6UM. 2016.

[37] Karthikeyan Bhargavan et al. "Formal verification of the PQXDH Post-Quantum key agreement protocol for end-to-end secure messaging". In: *USENIX Security*. 2024.

[38] European Data Protection Board. "Guidelines 03/2022 on Deceptive design patterns in social media platform interfaces: how to recognise and avoid them". In: (2022). PERMALINK: https://perma.cc/RLY5-FN2X.

[39] European Data Protection Board. "Opinion 28/2024 on certain data protection aspects related to the processing of personal data in the context of AI models". In: (2024). PERMALINK: https://perma.cc/MB7M-KMHF.

[40] European Data Protection Board. "Report of the work undertaken by the ChatGPT Taskforce". In: (2024). PERMALINK: https://perma.cc/J6UQ-ZM82.

[41] European Data Protection Board. "Statement 03/2021 on the ePrivacy Regulation". In: (2021). PERMALINK: https://perma.cc/92MS-2HXJ.

[42] August Bourgeus and Laurens Vandercruysse. "Reinvigorating Consent: Exploring New Paradigms for Privacy and Data Sharing". In: *Proceedings of 22nd European Conference on Computer-Supported Cooperative Work*. European Society for Socially Embedded Technologies (EUSSET), 2024. DOI: 10.48340/ecscw2024_ep08. URL: https://doi.org/10.48340/ecscw2024_ep08.

[43] Andrea J. Boyack. "The Shape of Consumer Contracts". In: *Denver Law Review* (2023). PERMALINK: https://perma.cc/T4LR-6CR5.

[44] Anu Bradford. "The Brussels Effect: How the European Union Rules the World". In: *Oxford University Press* (2020). PERMALINK: https://perma.cc/4AND-6QJB.

[45] Ferdinand Brasser et al. "Software Grand Exposure: SGX Cache Attacks Are Practical". In: *WOOT*. 2017.

[46] H Brignull et al. *Deceptive patterns – user interfaces designed to trick you*. https://www.deceptive.design/. PERMALINK: https://perma.cc/KQ4T-F7QD. Apr. 2023.

[47] Hannah Brown et al. "What does it mean for a language model to preserve privacy?" In: *FAccT*. 2022.

[48] R. Canetti. "Universally composable security: a new paradigm for cryptographic protocols". In: *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.

[49] Rein Canetti et al. "Deniable encryption". In: *CRYPTO*. 1997.

[50] Nicholas Carlini et al. "Extracting Training Data from Large Language Models". In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021.

[51] Nicholas Carlini et al. "The secret sharer: Evaluating and testing unintended memorization in neural networks". In: *28th USENIX security symposium (USENIX security 19)*. 2019, pp. 267–284.

[52] Foo Yun Chee. "EU ditches plans to regulate tech patents, AI liability, online privacy". In: *Reuters* (2025). URL: https://www.reuters.com/technology/eu-ditches-plans-regulate-tech-patents-ai-liability-online-privacy-2025-02-12/.

[53] Jeffrey Cheng et al. "Dated data: Tracing knowledge cutoffs in large language models". In: *arXiv preprint arXiv:2403.12958* (2024).

[54] Mesut Cicek, Dogan Gursoy, and Lu Lu. "Adverse impacts of revealing the presence of "Artificial Intelligence (AI)" technology in product and service descriptions on purchase intentions: the mediating role of emotional trust and the moderating role of perceived risk". In: *Journal of Hospitality Marketing & Management* (2024). DOI: 10.1080/19368623.2024.2368040.

[55] Sam Clark and Pieter Haeck. *Europe's privacy patrol is spoiling Big Tech's AI party*. https://www.politico.eu/article/europe-privacy-patrol-vengeance-block-ai-artificial-intelligence/. PERMALINK: https://perma.cc/A8YF-MM3U. 2024.

[56] Jessica Rich Claudia Bourne Farrell. *Mortgage Company Settles Information Security Charges*. https://www.ftc.gov/news-events/news/press-releases/2005/09/mortgage-company-settles-information-security-charges. PERMALINK: https://perma.cc/GM74-8VZS. 2005.

[57] Ignacio Cofone. "The Privacy Fallacy". In: *Cambridge University Press* (2023). PERMALINK: https://perma.cc/L4D8-TSFV. URL: https://www.cambridge.org/core/books/privacy-fallacy/547578F2A1AE0C40963105CE066B412E.

[58] Julie E. Cohen. "Examined Lives: Informational Privacy and the Subject as Object". In: *52 Stan. L. Rev. 1373-1438* (2000). PERMALINK: https://perma.cc/VR6L-87AS.

[59] Julie E. Cohen. *Law for the Platform Economy*. https://scholarship.law.georgetown.edu/facpub/2015. PERMALINK: https://perma.cc/59AP-9F32. 2017.

[60] Julie E. Cohen. "What Privacy Is For". In: *Georgetown Law Faculty Publications and Other Works* (2013). PERMALINK: HTTPS://PERMA.CC/WQU4-PAJV.

[61] Katriel Cohn-Gordon, Cas Cremers, and Luke Garratt. "On post-compromise security". In: *IEEE CSF*. 2016.

[62] Data Protection Commission. *Data Protection Commission welcomes conclusion of proceedings relating to X's AI tool 'Grok'*. https://www.dataprotection.ie/en/news-media/press-releases/data-protection-commission-welcomes-conclusion-proceedings-relating-xs-ai-tool-grok. PERMALINK: https://perma.cc/L6F3-LGMP. 2024.

[63] Data Protection Commission. *The DPC's Engagement with Meta on AI*. https://www.dataprotection.ie/en/news-media/latest-news/dpcs-engagement-meta-ai. PERMALINK: https://perma.cc/Y65M-U72C. 2024.

[64] European Commission. *Consumer protection: manipulative online practices found on 148 out of 399 online shops screened*. https://ec.europa.eu/commission/presscorner/detail/en/ip_23_418. PERMALINK: https://perma.cc/J24T-LCEH. 2023.

[65] European Commission. "Mergers: Commission approves acquisition of WhatsApp by Facebook". In: *Press Release* (2014). PERMALINK: https://perma.cc/5EU4-KNWE.

[66] European Commission. "Mergers: Commission fines Facebook €110 million for providing misleading information about WhatsApp takeover". In: *Press Release* (2016). PERMALINK: https://perma.cc/V9MC-YHGQ.

[67] European Commission. "Simpler EU digital rules and new digital wallets to save billions for businesses and boost innovation". In: (2025). PERMALINK: https://perma.cc/U7VL-GA85. URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_25_2718.

[68] Australia Competition and Consumer Commission. *$20m penalty for Meta companies for conduct liable to mislead consumers about use of their data*. https://www.accc.gov.au/media-release/20m-penalty-for-meta-companies-for-conduct-liable-to-mislead-consumers-about-use-of-their-data. PERMALINK: https://perma.cc/2WPA-LQ5A. 2023.

[69] *Confer — Private AI. Speak freely.* PERMALINK: https://perma.cc/3HAB-4X9W. URL: https://confer.to/ (visited on 06/19/2026).

[70] Graeme Connell and Rolfe Schmidt. *Signal Protocol and Post-Quantum Ratchets*. PERMALINK: https://perma.cc/V2N2-HK6E. 2025. URL: https://signal.org/blog/spqr/.

[71] A Feder Cooper and James Grimmelmann. "The Files are in the Computer: Copyright, Memorization, and Generative AI". In: *arXiv preprint arXiv:2404.12590* (2024).

[72] A. Feder Cooper et al. "Machine Unlearning Doesn't Do What You Think: Lessons for Generative AI Policy, Research, and Practice". In: *arXiv preprint arXiv:2412.06966* (2024). URL: https://arxiv.org/pdf/2412.06966.

[73] Victor Costan and Srinivas Devadas. "Intel SGX Explained". In: *Cryptology ePrint Archive* (2016).

[74] Lorrie Faith Cranor and Simson Garfinkel. *Security and usability: designing secure systems that people can use*. O'Reilly Media, 2005. ISBN: 0596514891.

[75] Loredana Crisan. *Launching Default End-to-End Encryption on Messenger*. https://about.fb.com/news/2023/12/default-end-to-end-encryption-on-messenger. PERMALINK: https://perma.cc/9HSW-WNFG. 2023.

[76] Cybersecurity and Infrastructure Security Agency (CISA) et al. *Enhanced Visibility and Hardening Guidance for Communications Infrastructure*. https://www.cisa.gov/resources-tools/resources/enhanced-visibility-and-hardening-guidance-communications-infrastructure. PERMALINK: https://perma.cc/K54W-WMB4. Dec. 2024.

[77] Autoridade Nacional de Proteção de Dados. *ANPD determina suspensão cautelar do tratamento de dados pessoais para treinamento da IA da Meta*. https://www.gov.br/anpd/pt-br/assuntos/noticias/anpd-determina-suspensao-cautelar-do-tratamento-de-dados-pessoais-para-treinamento-da-ia-da-meta. PERMALINK: https://perma.cc/WWA4-NQ9K. 2024.

[78] Autoridade Nacional de Proteção de Dados. *Meta cumpre exigências da ANPD e poderá retomar, com restrições, o uso de dados pessoais para treinamento de inteligência artificial*. https://www.gov.br/anpd/pt-br/assuntos/noticias/meta-cumpre-exigencias-da-anpd-e-podera-retomar-com-restricoes-o-uso-de-dados-pessoais-para-treinamento-de-inteligencia-artificial. PERMALINK: https://perma.cc/3M5Q-ZCJN. 2024.

[79] Artem Dinaburg. *Understanding Apple's On-Device and Server Foundation Models release*. https://blog.trailofbits.com/2024/06/14/understanding-apples-on-device-and-server-foundations-model-release/. PERMALINK: https://perma.cc/AFW8-LXND. 2024.

[80] Roger Dingledine and Nick Mathewson. "Anonymity Loves Company: Usability and the Network Effect". In: *5th Annual Workshop on the Economics of Information Security, WEIS 2006, Robinson College, University of Cambridge, England, UK, June 26-28, 2006*. 2006. URL: https://econinfosec.org/archive/weis2006/docs/41.pdf.

[81] Discord. "Discord's audio/video end-to-end encryption (DAVE) protocol". In: (2024).

[82] Charles Duhigg. *How Companies Learn Your Secrets*. https://www.nytimes.com/2012/02/19/magazine/shopping-habits.html. 2012.

[83] Cynthia Dwork. "Differential Privacy: A Survey of Results". In: *Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings*. Ed. by Manindra Agrawal et al. Vol. 4978. Lecture Notes in Computer Science. Springer, 2008, pp. 1–19. DOI: 10.1007/978-3-540-79228-4_1. URL: https://doi.org/10.1007/978-3-540-79228-4_1.

[84] Nouha Dziri et al. "Faith and fate: Limits of transformers on compositionality". In: *Advances in neural information processing systems* 36 (2023), pp. 70293–70332.

[85] Benj Edwards. *So far, AI hasn't been profitable for Big Tech*. https://arstechnica.com/information-technology/2023/10/so-far-ai-hasnt-been-profitable-for-big-tech. PERMALINK: https://perma.cc/B3R3-GA2Y.

[86] Element. URL: https://element.io.

[87] Euronews. "South Korea fines Meta €14 million for illegally collecting information on Facebook users". In: *Euronews Tech News* (2024). PERMALINK: https://perma.cc/8MY9-X4MP.

[88] Shufan Fei et al. "Security Vulnerabilities of SGX and Countermeasures: A Survey". In: *ACM Computing Surveys (CSUR)* (2021).

[89] Geoffrey A. Fowler. "I tried to read all my app privacy policies. It was 1 million words." In: *The Washington Post* (2022).

[90] Stefano Fratta. *Building AI Technology for Europeans in a Transparent and Responsible Way*. https://about.fb.com/news/2024/06/building-ai-technology-for-europeans-in-a-transparent-and-responsible-way/. PERMALINK: https://perma.cc/6377-A64J. 2024.

[91] Jody Godoy. "Trump picks Andrew Ferguson to chair FTC". In: *Reuters* (2024).

[92] Shashwat Goel et al. "Corrective machine unlearning". In: *arXiv preprint arXiv:2402.14015* (2024).

[93] Google. *Use end-to-end encryption in Google Messages*. PERMALINK: https://perma.cc/WP25-WD89. 2025. URL: https://support.google.com/messages/answer/10252671.

[94] Jennifer Stisa Granick and Daniel Kahn Gillmor. *The Vital Role of End-to-End Encryption*. https://www.aclu.org/news/privacy-technology/the-vital-role-of-end-to-end-encryption. PERMALINK: https://perma.cc/RN4P-K94X.

[95] Nico Grant. *Google's A.I. Search Errors Cause a Furor Online*. https://www.nytimes.com/2024/05/24/technology/google-ai-overview-search.html. PERMALINK: https://perma.cc/NG3P-JFM5.

[96] Matthew Green. *Let's talk about AI and end-to-end encryption*. A Few Thoughts on Cryptographic Engineering. Jan. 2025. URL: https://blog.cryptographyengineering.com/2025/01/17/lets-talk-about-ai-and-end-to-end-encryption.

[97] Matthew Green. *Looking back at the Snowden revelations*. https://blog.cryptographyengineering.com/2019/09/24/looking-back-at-the-snowden-revelations. PERMALINK: https://perma.cc/F6PV-SW9H.

[98] Matthew D Green and Ian Miers. "Forward secure asynchronous messaging from puncturable encryption". In: *IEEE S&P*. 2015.

[99] Andy Greenberg. *Hacker Lexicon: What Is End-to-End Encryption?* https://www.wired.com/2014/11/hacker-lexicon-end-to-end-encryption/. PERMALINK: https://perma.cc/JET4-3P8T. 2014.

[100] Andy Greenberg. *Signal Is More Than Encrypted Messaging. Under Meredith Whittaker, It's Out to Prove Surveillance Capitalism Wrong*. https://www.wired.com/story/meredith-whittaker-signal/. PERMALINK: https://perma.cc/6JMM-537E. 2024.

[101] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. *Message Franking via Committing Authenticated Encryption*. Cryptology ePrint Archive, Paper 2017/664. 2017. URL: https://eprint.iacr.org/2017/664.

[102] Britta Hale and Chelsea Komlo. "On end-to-end encryption". In: *Cryptology ePrint Archive* (2022).

[103] Padraic Halpin. *Top EU privacy regulator opens probe into Google's AI compliance*. http://archive.today/2024.09.12-004248/https://www.reuters.com/technology/artificial-intelligence/top-eu-privacy-regulator-opens-probe-into-googles-ai-compliance-2024-09-11/. PERMALINK: https://perma.cc/2LUC-U7V8. 2024.

[104] Melissa Heikkilä. *Nobody knows how AI works*. https://www.technologyreview.com/2024/03/05/1089449/nobody-knows-how-ai-works/. PERMALINK: https://perma.cc/XTN9-J5DA. 2024.

[105] Kashmir Hill. *How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did*. https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/. PERMALINK: https://perma.cc/D944-EEFW. 2022.

[106] Rand Hindi. *Homomorphic Encryption and LLM : Is ChatGPT end to end encrypted ?* PERMALINK: https://perma.cc/4TQZ-7ADT. 2023. URL: https://www.zama.ai/post/chatgpt-privacy-with-homomorphic-encryption (visited on 10/21/2024).

[107] Chris Jay Hoofnagle and Jennifer M. Urban. "Alan Westin's Privacy Homo Economicus". In: *49 Wake Forest Law Review 261 (2014), UC Berkeley Public Law Research Paper No. 2434800* (2014). PERMALINK: https://perma.cc/2EWE-7DU9.

[108] *How to back up your chat history | WhatsApp Help Center*. https://faq.whatsapp.com/481135090640375/?cms_platform=android. PERMALINK: https://perma.cc/AP3F-9X6S. (Visited on 01/15/2025).

[109] Hongsheng Hu et al. "Membership inference attacks on machine learning: A survey". In: *ACM Computing Surveys (CSUR)* 54.11s (2022), pp. 1–37.

[110] Ryan Hurst. *Trusted Execution Environment Vulnerability Taxonomy*. PERMALINK: https://perma.cc/KJ3K-MQ3G. 2026. URL: https://docs.google.com/document/d/1vlkwCJPEFyC2vIt9PgTkf-SuUvsELaFDH50GcTdd-bA/.

[111] Irish Data Protection Commission. *Responsible Artificial Intelligence Innovation: Insights from the Data Protection Commission's Supervision of AI (2021–2025)*. PERMALINK: https://perma.cc/SF9P-UZ9Q. 2026. URL: https://www.dataprotection.ie/sites/default/files/uploads/2026-09/DPC-AI-Insights-Report-AC.pdf.

[112] Ella Jakubowska. *A beginner's guide to EU rules on scanning private communications: Part 1*. https://edri.org/our-work/a-beginners-guide-to-eu-rules-on-scanning-private-communications-part-1/. PERMALINK: https://perma.cc/63ZP-2D44. 2021.

[113] Kristin Madigan Jay Mayfield Jessica Lyon. *Dental Practice Software Provider Settles FTC Charges It Misled Customers About Encryption of Patient Data*. https://www.ftc.gov/news-events/news/press-releases/2016/01/dental-practice-software-provider-settles-ftc-charges-it-misled-customers-about-encryption-patient. PERMALINK: https://perma.cc/7TC3-UJD8. 2016.

[114] David Kaplan, Jeremy Powell, and Tom Woller. *AMD memory encryption*. https://www.amd.com/content/dam/amd/en/documents/epyc-business-docs/white-papers/memory-encryption-white-paper.pdf. PERMALINK: https://perma.cc/WY82-655P. 2021.

[115] David Kaye. *Report on encryption, anonymity, and the human rights framework*. https://www.undocs.org/A/HRC/29/32. PERMALINK: https://perma.cc/QN7A-R9V5. 2015.

[116] Rebecca Kern. "FTC Takes Action Against Marriott and Starwood Over Multiple Data Breaches". In: (2024).

[117] Alec Tyson and Emma Kikuchi. *Growing public concern about the role of artificial intelligence in daily life*. https://www.pewresearch.org/short-reads/2023/08/28/growing-public-concern-about-the-role-of-artificial-intelligence-in-daily-life/. PERMALINK: https://perma.cc/3BDG-5MPL. 2023.

[118] Mallory Knodel. *The FBI Wants You to Know it Has Not Changed its Position on Encryption. And That's a Problem.* Tech Policy Press. PERMALINK: https://perma.cc/7ZHL-54WU. Dec. 2024.

[119] Mallory Knodel and Andrés Fábrega. *Can Bots Read Your Encrypted Messages? Encryption, Privacy, and the Emerging AI Dilemma*. Tech Policy Press. Feb. 2025. URL: https://www.techpolicy.press/can-bots-read-your-encrypted-messages-encryption-privacy-and-the-emerging-ai-dilemma.

[120] Mallory Knodel et al. *Definition of End-to-end Encryption*. https://datatracker.ietf.org/doc/html/draft-knodel-e2ee-definition-11. PERMALINK: https://perma.cc/33J2-TKNE. 2023.

[121] Ehren Kret. *Quantum Resistance and the Signal Protocol*. PERMALINK: https://perma.cc/CUF6-EHK5. 2023. URL: https://signal.org/blog/pqxdh/.

[122] Sangho Lee et al. "Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing". In: *USENIX Security*. 2017.

[123] Maeve Duggan Lee Rainie. *Privacy and Information Sharing*. 2016. URL: https://www.pewresearch.org/internet/2016/01/14/privacy-and-information-sharing/.

[124] Lawrence Lessig. *Code and Other Laws of Cyberspace*. Basic Books, 1999. ISBN: 0-465-03912-X.

[125] Tom Lewis. "How Generative AI Is Making Customer Experience Worse". In: *Forbes* (2024). PERMALINK: https://perma.cc/44FY-QYFV.

[126] Caroline Schmitz Linda Holleran Kopp Ryan Mehm. "FTC Requires Zoom to Enhance its Security Practices as Part of Settlement". In: (2020). PERMALINK: https://perma.cc/24RA-7NB5.

[127] Bart van der Sloot Linnet Taylor Luciano Floridi. *Group Privacy, New Challenges of Data Technologies*. Springer Cham, 2017. ISBN: 978-3-319-46606-4.

[128] Kevin Litman-Navarro. *We Read 150 Privacy Policies. They Were an Incomprehensible Disaster.* https://www.nytimes.com/interactive/2019/06/12/opinion/facebook-google-privacy-policies.html. PERMALINK: https://perma.cc/5XJJ-4CQV. 2019.

[129] Joshua Lund. *I link therefore I am*. https://signal.org/blog/i-link-therefore-i-am/. PERMALINK: https://perma.cc/9DZY-2BHG. 2019.

[130] Kim Mackrael. "Meta to European Union: Your Tech Rules Threaten to Squelch the AI Boom". In: *The Washington Post* (2024).

[131] Moxie Marlinspike. *Confessions to a data lake*. en. PERMALINK: https://perma.cc/7GKR-GAGD. Dec. 2025. URL: https://confer.to/blog/2025/12/confessions-to-a-data-lake (visited on 01/07/2026).

[132] Moxie Marlinspike. *Private inference*. en. PERMALINK: https://perma.cc/P89Y-M5FE. Jan. 2026. URL: https://confer.to/blog/2026/01/private-inference (visited on 01/07/2026).

[133] Moxie Marlinspike. *Technology preview: Private contact discovery for Signal*. https://signal.org/blog/private-contact-discovery/. PERMALINK: https://perma.cc/DR7X-3G2E. 2017.

[134] Florencia Marotta-Wurgler. "Will Increased Disclosure Help? Evaluating the Recommendations of the ALI's Principles of the Law of Software Contracts". In: *University of Chicago Law Review: Vol. 78: Iss. 1, Article 9* (2011). PERMALINK: https://perma.cc/BP4K-VPTQ.

[135] Kim Martineau. *What is retrieval-augmented generation?* https://research.ibm.com/blog/retrieval-augmented-generation-RAG. PERMALINK: https://perma.cc/48JR-WJ2U. Aug. 2023. (Visited on 12/11/2024).

[136] Elaine McArdle. "(Anti)Trust Issues". In: *Harvard Law Bulletin* (2024). PERMALINK: https://perma.cc/HVX2-742Z.

[137] R Thomas McCoy et al. "How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven". In: *Transactions of the Association for Computational Linguistics* 11 (2023), pp. 652–670.

[138] Aleecia M McDonald and Lorrie Faith Cranor. "The cost of reading privacy policies". In: *I/S: A Journal of Law & Policy for the Information Society* 4 (2009), pp. 543–897.

[139] Meta. "Building AI Technology for the UK in a Responsible and Transparent Way". In: *Newsroom for Europe, Middle East and Africa* (2024). PERMALINK: https://perma.cc/VGZ6-QGHK.

[140] Meta. *How Meta uses information for generative AI models and features*. PERMALINK: https://perma.cc/57AG-YUNX. 2025. URL: https://www.facebook.com/privacy/genai.

[141] Meta. *How Meta works with select partners to improve AI responses*. PERMALINK: https://perma.cc/F72P-FGSU. URL: https://www.facebook.com/privacy/dialog/ai-partners.

[142] Meta. *Resetting or correcting an AI*. PERMALINK: https://perma.cc/YL2N-3XEH. 2025. URL: https://privacycenter.instagram.com/dialog/correcting-ai-information/.

[143] James C. Miller III. *FTC Policy Statement on Deception*. https://www.ftc.gov/system/files/documents/public_statements/410531/831014deceptionstmt.pdf. Letter. PERMALINK: https://perma.cc/XD4D-AML3.

[144] Dan Milmo. *Elon Musk's X under pressure from regulators over data harvesting for Grok AI*. https://www.theguardian.com/technology/article/2024/jul/26/elon-musks-x-under-pressure-from-regulators-over-data-harvesting-for-grok-ai. PERMALINK: https://perma.cc/SPP5-QD5S. 2024.

[145] Dipak Mondal. "Anti-trust regulator levy Rs 213 crore penalty on Meta for 'forcing' WhatsApp users to share data". In: *The New Indian Express* (2024). PERMALINK: https://perma.cc/863A-8AVC.

[146] Milad Nasr et al. "Scalable extraction of training data from (production) language models". In: *arXiv preprint arXiv:2311.17035* (2023).

[147] National Academies of Sciences, Engineering, and Medicine. *Decrypting the Encryption Debate: A Framework for Decision Makers*. PERMALINK: https://perma.cc/NX82-EMD8. The National Academies Press, 2018. URL: https://nap.nationalacademies.org/catalog/25010/decrypting-the-encryption-debate-a-framework-for-decision-makers.

[148] Lily Hay Newman and Matt Burgess. "Signal's Creator Is Helping Encrypt Meta AI". en-US. In: *Wired* (Mar. 2026). Section: tags. ISSN: 1059-1028. URL: https://www.wired.com/story/signals-creator-is-helping-encrypt-meta-ai/ (visited on 05/29/2026).

[149] Helen Nissenbaum. *Privacy in Context: Technology, Policy and the Integrity of Social Life*. Stanford Law Books, 2010. ISBN: 0804752370.

[150] Midas Nouwens et al. "Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. ACM, Apr. 2020. DOI: 10.1145/3313831.3376321. URL: http://dx.doi.org/10.1145/3313831.3376321.

[151] NOYB. "noyb urges 11 DPAs to immediately stop Meta's abuse of personal data for AI". In: *Forced Consent & Consent Bypass* (2024). PERMALINK: https://perma.cc/92R9-XX5D.

[152] Oasis Labs. *Smart Privacy for Web3 & AI*. PERMALINK: HTTPS://PERMA.CC/9CXH-8964. URL: \url{https://oasisprotocol.org/}.

[153] Information Commissioner's Office. "Due diligence when sharing data following mergers and acquisitions". In: *For Organizations* (). PERMALINK: https://perma.cc/WMJ8-W8AJ.

[154] Information Commissioner's Office. *ICO statement in response to Meta's announcement on user data to train AI*. https://ico.org.uk/about-the-ico/media-centre/news-and-blogs/2024/09/ico-statement-in-response-to-metas-announcement-on-user-data-to-train-ai/. PERMALINK: https://perma.cc/JWT8-6RC4. 2024.

[155] Information Commissioner's Office. *When can we rely on legitimate interests?* https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/lawful-basis/legitimate-interests/when-can-we-rely-on-legitimate-interests/. PERMALINK: https://perma.cc/PQZ4-CM76.

[156] OpenAI. *Data Controls FAQ*. https://help.openai.com/en/articles/7730893-data-controls-faq. WEBARCHIVE: https://web.archive.org/web/20250109132809/https://help.openai.com/en/articles/7730893-data-controls-faq. 2024.

[157] OpenAI. *How your data is handled when you use ChatGPT through Apple's integrations*. https://help.openai.com/en/articles/9737562-how-your-data-is-handled-when-you-use-chatgpt-through-apple-s-integrations. WEBARCHIVE: https://web.archive.org/web/20250119144511/https://help.openai.com/en/articles/9737562-how-your-data-is-handled-when-you-use-chatgpt-through-apple-s-integrations. 2024.

[158] OpenAI. *Learn ChatGPT: Plugins*. PERMALINK: https://perma.cc/58WV-DKU2. URL: https://learn.chatgpt.com/docs/plugins?surface=app#app-use-apple-messages-from-codex.

[159] OpenAI. *Setting up ChatGPT with Apple Intelligence*. https://help.openai.com/en/articles/10269382-setting-up-chatgpt-with-apple-intelligence. PERMALINK: https://perma.cc/V8M6-68XK. 2024.

[160] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in neural information processing systems* 35 (2022), pp. 27730–27744.

[161] Saurabh Panjwani et al. "Understanding the privacy-personalization dilemma for web search: a user perspective". In: *2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013*. Ed. by Wendy E. Mackay, Stephen A. Brewster, and Susanne Bødker. ACM, 2013, pp. 3427–3430. DOI: 10.1145/2470654.2466470. URL: https://doi.org/10.1145/2470654.2466470.

[162] Nicolas Papernot and Abhradeep Guha Thakurta. *How to deploy machine learning with differential privacy*. https://www.nist.gov/blogs/cybersecurity-insights/how-deploy-machine-learning-differential-privacy. PERMALINK: https://perma.cc/39ZU-CGBV. 2021.

[163] European Parliament. *Proposal for a regulation on privacy and electronic communications*. https://www.europarl.europa.eu/legislative-train/theme-connected-digital-single-market/file-jd-e-privacy-reform. PERMALINK: https://perma.cc/796G-R3J7. 2024.

[164] Rafael Pass, Elaine Shi, and Florian Tramer. "Formal abstractions for attested execution secure processors". In: *Eurocrypt*. 2017.

[165] Rafael Pass, Elaine Shi, and Florian Tramèr. "Formal Abstractions for Attested Execution Secure Processors". In: *Advances in Cryptology – EUROCRYPT 2017*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Cham: Springer International Publishing, 2017, pp. 260–289. ISBN: 978-3-319-56620-7.

[166] Trevor Perin and Moxie Marlinspike. *The Double Rachet Algorithm*. PERMALINK: https://perma.cc/H26Z-TYD2. 2016. URL: https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf.

[167] Fabio Petroni et al. *Language Models as Knowledge Bases?* https://arxiv.org/abs/1909.01066. 2019. arXiv: 1909.01066 [cs.CL].

[168] Natalia Ponomareva et al. "How to DP-fy ML: A Practical Guide to Machine Learning with Differential Privacy". In: *Journal of Artificial Intelligence Research* 77 (July 2023), 1113–1201. ISSN: 1076-9757. DOI: 10.1613/jair.1.14649. URL: http://dx.doi.org/10.1613/jair.1.14649.

[169] Samsung Mobile Press. PERMALINK: https://perma.cc/V9TD-LUHL. 2026. URL: https://www.samsungmobilepress.com/articles/bixby-one-ui-8-5-conversational-device-agent.

[170] Dominic Preston. *Apple ordered to open encrypted user accounts globally to UK spying*. PERMALINK: https://perma.cc/NM5B-77PN. 2025. URL: https://www.theverge.com/news/608145/apple-uk-icloud-encrypted-backups-spying-snoopers-charter.

[171] Imran Rahman-Jones. *AI chatbots unable to accurately summarise news, BBC finds*. https://www.bbc.com/news/articles/c0m17d8827ko. PERMALINK: https://perma.cc/79UE-LJY6. 2025.

[172] Reuters. "India restricts WhatsApp sharing data with other Meta entities, imposes $25.4 mln fine". In: *Reuters - Boards, Policy & Regulation* (2024).

[173] European Digital Rights. *Why the Digital Omnibus puts GDPR and ePrivacy at risk*. PERMALINK: https://perma.cc/Z52Q-CZ85. 2025. URL: https://edri.org/our-work/why-the-digital-omnibus-puts-gdpr-and-eprivacy-at-risk/.

[174] Samsung. *Around the World (Aug 2025)*. PERMALINK: https://perma.cc/7QMK-DASB. 2025. URL: https://news.samsung.com/global/around-the-world-aug-2025.

[175] Samsung. *Change the default messaging app on your Galaxy phone*. PERMALINK: https://perma.cc/XZV9-N9QR. 2025. URL: https://www.samsung.com/us/support/answer/ANS10001947/.

[176] Samsung. *Galaxy AI*. PERMALINK: https://perma.cc/Q76P-KRES. URL: https://www.samsung.com/us/galaxy-ai/.

[177] Samsung. *Samsung and Google Cloud Join Forces to Bring Generative AI to Samsung Galaxy S24 series*. PERMALINK: https://perma.cc/9FTR-MCVY. 2024. URL: https://news.samsung.com/us/samsung-google-cloud-bring-generative-ai-to-samsung-galaxy-s24-series/.

[178] Samsung. *Samsung Galaxy S25 Series Sets the Standard of AI Phone as a True AI Companion*. PERMALINK: https://perma.cc/RJ7U-28WN. 2025. URL: https://news.samsung.com/global/samsung-galaxy-s25-series-sets-the-standard-of-ai-phone-as-a-true-ai-companion.

[179] Samsung. *Samsung Knox Documentation: Knox Vault*. PERMALINK: https://perma.cc/7QK3-UA9C. 2024. URL: https://docs.samsungknox.com/admin/fundamentals/whitepaper/samsung-knox-for-android/core-platform-security/knox-vault/.

[180] Samsung. *Use features with Galaxy AI on your Galaxy phone and tablet*. PERMALINK: https://perma.cc/E6YA-9ZWR. 2025. URL: https://www.samsung.com/us/support/answer/ANS10000753/.

[181] Samsung. *What are the Galaxy AI Features on the Galaxy S25 Series?* PERMALINK: https://perma.cc/VD4H-P7G4. 2025. URL: https://www.samsung.com/uk/mobile-phone-buying-guide/samsung-galaxy-s25-galaxy-ai-features/.

[182] Samsung. *What is the 'Enhanced Data Protection' function, and when can I use it?* PERMALINK: https://perma.cc/2B37-KUBA. 2023. URL: https://www.samsung.com/ae/support/mobile-devices/what-is-the-enhanced-data-protection-function-and-when-can-i-use-it.

[183] Samsung. *Your Privacy, Secured: How Galaxy AI Empowers You to Take Control of Your Data.* PERMALINK: https://perma.cc/MCT4-MGUG. 2025. URL: https://news.samsung.com/us/your-privacy-secured-galaxy-ai-empowers-you-take-control-your-data/.

[184] Sarah Scheffler and Jonathan Mayer. *SoK: Content Moderation for End-to-End Encryption.* https://arxiv.org/abs/2303.03979. 2023. arXiv: 2303.03979 [cs.CR].

[185] Timo Schick et al. "Toolformer: Language models can teach themselves to use tools". In: *Advances in neural information processing systems* 36 (2023), pp. 68539–68551.

[186] Session. URL: https://getsession.org.

[187] Alex Seto et al. "WireTap: Breaking Server SGX via DRAM Bus Interposition". In: *ACM CCS*. 2025.

[188] Gadjo Sevilla. *WhatsApp outage results in a scramble for messaging alternatives.* https://www.emarketer.com/content/whatsapp-outage-results-scramble-messaging-alternatives. PERMALINK: https://perma.cc/LN6D-8DQ4. 2022.

[189] Justin Sherman. *Data Brokers Are a Threat to Democracy.* https://www.wired.com/story/opinion-data-brokers-are-a-threat-to-democracy/. PERMALINK: https://perma.cc/AH8Z-JAR7. 2021.

[190] Aaditya Shidham. *Trusted Execution Environments (TEEs): A primer.* PERMALINK: https://perma.cc/U9WX-668S. 2025. URL: {https://a16zcrypto.com/posts/article/trusted-execution-environments-tees-primer/}.

[191] Reza Shokri et al. "Membership inference attacks against machine learning models". In: *2017 IEEE symposium on security and privacy (SP)*. IEEE. 2017, pp. 3–18.

[192] Daniel J. Solove. "Murky Consent: An Approach to the Fictions of Consent in Privacy Law". In: *Boston University Law Review* (2024). PERMALINK: https://perma.cc/8ZGU-4M3F.

[193] Daniel J. Solove. "Privacy Self-Management and the Consent Dilemma". In: *126 Harvard Law Review 1880 (2013), GWU Legal Studies Research Paper No. 2012-141, GWU Law School Public Law Research Paper No. 2012-141* (2013). PERMALINK: https://perma.cc/4THR-HS4F.

[194] Jasmine E. McNealy Stephanie T. Nguyen. *I, Obscura — Illuminating deceptive design patterns in the wild.* https://pacscenter.stanford.edu/wp-content/uploads/2021/07/I-Obscura-Zine.pdf. PERMALINK: https://perma.cc/2UAD-ZAFN. 2021.

[195] European Data Protection Supervisor. *Data Protection.* https://www.edps.europa.eu/data-protection/data-protection_en. PERMALINK: https://perma.cc/H8KV-SQXN.

[196] Eli Tan. *When the Terms of Service Change to Make Way for A.I. Training.* https://www.nytimes.com/2024/06/26/technology/terms-service-ai-training.html. PERMALINK: https://perma.cc/NJL3-U22V. 2024.

[197] Telegram. *End-to-End Encryption FAQ.* https://tsf.telegram.org/manuals/e2ee-simple. PERMALINK: https://perma.cc/68ST-YUFQ. (Visited on 01/15/2025).

[198] The Economist. *The world's most valuable resource is no longer oil, but data.* https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data. PERMALINK: https://perma.cc/YF85-8QLS. 2017.

[199] Stephen Thomas. *AI Assistants in Encrypted Messaging: Moving Too Fast in the Dark?* NYU Center for Data Science on Medium. Mar. 2025. URL: https://nyudatascience.medium.com/ai-assistants-in-encrypted-messaging-moving-too-fast-in-the-dark-8680ab195834.

[200] Threema. URL: https://threema.com.

[201] Nik Unger et al. "SoK: Secure Messaging". In: *IEEE S&P*. 2015.

[202] Filippo Valsorda. *A Cryptography Engineer's Perspective on Quantum Computing Timelines.* PERMALINK: https://perma.cc/3HUF-3M8F. 2026. URL: https://words.filippo.io/crqc-timeline/.

[203] Jo Van Bulck et al. "LVI: Hijacking Transient Execution through Microarchitectural Load Value Injection". In: *IEEE S&P*. 2020.

[204] Max Van Kleek et al. "Better the Devil You Know: Exposing the Data Sharing Practices of Smartphone Apps". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017.

[205] Stephan Van Schaik et al. "SoK: SGX.Fail: How Stuff Gets eXposed". In: *IEEE S&P*. 2024.

[206] Pablo Villalobos et al. "Position: Will we run out of data? Limits of LLM scaling based on human-generated data". In: *Forty-first International Conference on Machine Learning*.

[207] Adrian Weckler. *Meta 'pauses' AI data collection in EU following Irish DPC request*. https://www.independent.ie/business/technology/meta-pauses-ai-data-collection-in-eu-following-irish-dpc-request/a2000443736.html. PERMALINK: https://perma.cc/XDG6-ZH5H. 2024.

[208] Wex. "Consent". In: *Legal Information Institute* (2022). PERMALINK: https://perma.cc/RB7X-UL4A.

[209] WhatsApp. *About Meta AI web search in Incognito chats with Meta AI*. PERMALINK: https://perma.cc/G4J6-U6G9. URL: https://faq.whatsapp.com/1462468411932488.

[210] WhatsApp. *Introducing Advanced Chat Privacy: Enhanced Protection for Your Most Sensitive Conversations*. PERMALINK: https://perma.cc/8DWQ-D3N8. 2025.

[211] WhatsApp. "Private Processing for WhatsApp Overview". In: (2025).

[212] WhatsApp. "WhatsApp Encryption Overview, v.8". In: (2024). PERMALINK: https://perma.cc/GQ5W-RUGX. URL: https://whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf.

[213] WhatsApp. *WhatsApp FAQ: About AI experiences available through WhatsApp*. PERMALINK: https://perma.cc/JC8G-K754. 2025. URL: https://faq.whatsapp.com/1111412106858632.

[214] WhatsApp. *WhatsApp FAQ: About AIs from Meta*. PERMAINK: https://perma.cc/8D6T-F3CQ. 2025. URL: https://faq.whatsapp.com/2257017191175152.

[215] WhatsApp. *WhatsApp FAQ: About Private Processing*. PERMALINK: https://perma.cc/SE26-9VBQ. 2025. URL: https://faq.whatsapp.com/2089630958184255.

[216] WhatsApp. *WhatsApp FAQ: How to use Meta AI to summarize messages*. PERMALINK: https://perma.cc/TH3R-2MFK. 2025. URL: https://faq.whatsapp.com/9004571389643912.

[217] Alma Whitten, J Doug Tygar, et al. "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0." In: *USENIX security symposium*. Vol. 348. 1999, pp. 169–184.

[218] Simon Willison. *The lethal trifecta for AI agents*. Simon Willison's Weblog. PERMALINK: https://perma.cc/LN7J-844J. 2025. URL: https://simonw.substack.com/p/the-lethal-trifecta-for-ai-agents.

[219] Lukas Wutschitz et al. "Rethinking privacy in machine learning pipelines from an information flow control perspective". In: *arXiv preprint arXiv:2311.15792* (2023).

[220] Yuanzhong Xu, Weidong Cui, and Marcus Peinado. "Controlled-Channel Attacks: Deterministic Side Channels for Untrusted Operating Systems". In: *IEEE S&P*. 2015.

[221] Fan Zhang et al. "Town Crier". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM Conferences. Oct. 2016, pp. 270–282. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978326. URL: https://dl.acm.org/doi/10.1145/2976749.2978326 (visited on 09/17/2026).

[222] Philip Zimmermann. *Why I Wrote PGP*. 1999. URL: https://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html.

[223] Zoom. *Security Guide: Zoom*. https://explore.zoom.us/docs/doc/Zoom-Security-White-Paper.pdf. PERMALINK: https://perma.cc/P4MV-KWLE. 2021.

A E2EE Features Extended Evaluation Framework

In the course of evaluating the many ways in which AI features might be introduced in E2EE applications, we created a framework that describes necessary considerations for determining whether a specific feature

of an E2EE messaging application can be implemented such that it is still "E2EE," which was helpful in guiding our analysis. While we applied the framework largely to the context AI assistants in E2EE messaging applications, we believe there are many other features already widely deployed in mainstream E2EE applications whose implementations would benefit from use of this framework. AI has been a catalyst for our thinking about the role of data collection and use in today's communications apps. While simple, we hope the framework facilitates a principled discussion of the implications of features in E2EE environments.

Our framework distinguishes between three core axes that must be considered in the context of application features: (1) the *technical implementation* of the feature, and its confidentiality guarantees (or lack thereof). For considering AI assistants, we have performed this layer of analysis in Section 4.2; (2) the *data inputs* to the feature, i.e., what type of user data will interface with it; and (3) the level of *user control* over the feature. We discuss these components in more detail below.

Layer #1: Technical implementation. Does the underlying implementation of the AI assistant preserve the confidentiality guarantees of E2EE?

We do so by presenting an abstract taxonomy of possible implementations, and labeling their compatibility with E2EE confidentiality. This captures the intuition that there are better and worse implementations of the same feature from the perspective of confidentiality, and that certain implementations do not admit any configuration that is confidentiality-respecting. For our purposes, there are two main confidentiality considerations for the implementation of an AI feature:

- *Does the training of the AI system preserve the confidentiality of user data?* Training on personal data is fundamentally incompatible with E2EE, and thus no implementation of training on message data can guarantee confidentiality.
- *Is inference performed in a way that preserves the confidentiality of user data?* Inference is in principle compatible with E2EE confidentiality, whether done locally (on-device) or through central servers.

Layer #2: Data inputs. The next layer of our framework considers the data that is used as input to the AI features (for both training and inference). This layer is agnostic to a particular technical implementation, and is concerned exclusively with the data being used. There are four main considerations regarding the data that is fed to an AI features:

- *What type of data is being used?* Types of data may include, for example, messages, links, images, user profile information, metadata, etc. Different AI features may be considered more or less sensitive due to the type of data it interacts with.
- *Who is the intended recipient of the data?* The intended endpoint of communication may be either other users of the application, or the service provider itself. If it is the latter, then the data is not sensitive from the service provider's point of view.
- *What is the "semantic" value of the data?* The semantic value of data relates to the information content of the message. Data with higher semantic value is more sensitive.
- *Is the data referential?* Relevant to the prior point, some data may have "intrisic" meaning (e.g., messages), while other data may have "referential" meaning (e.g., URLs). Intrinsic data is generally more sensitive than referential data.

Layer #3: User control. The last layer of our framework considers the level of control that the user has over an AI feature in regards to data privacy. There are four main considerations related to user control:

- *Can the user opt out?* If the feature is on by default, does the user have the ability to turn the feature off? Being able to turn a feature off is more compatible with E2EE.
- *What is the default configuration for the AI feature?* The AI feature can be either on or off by default within a user's application, with the latter being more compatible with E2EE than the former.
- *What is the scope of the authorization?* Some features have *full scope* of authorization, i.e., turning on the feature authorizes its usage across the entire application state. Conversely, features may have *contextual scope* of authorization, in which case the feature must be turned on for individual application contexts. Contextual scope gives users more fine-grained control for how the AI assistant interacts with their data. For example, in a messaging application, it may be the case that the AI assistant must be individually authorized for each chat. Contextual scope is more compatible with E2EE.

- *What is the barrier to toggling off?* Disabling an AI assistant can involve a range of friction and effort for the user, and require different levels of technical fluency. For example, an easy-to-find toggle button in the user settings requires less effort from the user and presents less friction than one that is buried in the settings, or that requires sending a request to the service provider. Low barriers to toggling off are more aligned with the principles of E2EE.

These layers are roughly ordered from most technical and objective to the most context-dependent and normative. For example, the underlying implementation of a feature (Layer #1) may reveal a user's data in plaintext, and therefore immediately preclude that feature from being compatible with E2EE just on the basis of the desired cryptographic guarantees. Conversely, arguing the acceptability of certain user control setups (Layer #3) necessitates reference to normative principles.

B Evaluating Apple Intelligence

Having established our recommendations, we return to discuss the example of Apple (introduced in Section 2.5.1). Analyzing the strengths and shortcomings of Apple's system, though based on the limited information available, provides a helpful illustration of our recommendations in practice.

Many parts of Apple's system are compliant with our recommendations, and provide a positive illustration of what can be done to preserve E2EE security when deploying AI assistants. Some parts of Apple's system raise significant concerns related to our recommendations, and other parts are not well enough documented to support a detailed analysis of compliance with our recommendations.

Apple's system has three modes of processing, as described in Section 2.5.1. We proceed case by case.

Case I (Local Models). In Case I, no user data is used for training [19], which is compliant with Recommendation 1. Further, processing of queries—including, importantly, queries involving E2EE content—occurs locally on device [18]. However, whether tooling with external interactions may be used in Case I processing is not mentioned in any available materials. Therefore, Case I appears to be compliant with E2EE security according to Recommendation 2, unless it permits such tool use. If Case I processing does not permit (non-local) tool use, features relying solely on Case I may be publicized as E2EE without qualifications according to Recommendation 3. Apple claims to prioritize Case I; "many of the models" run fully on-device and it is the "more complex requests" that require using Cases II or III [18]—though it is not clear exactly what this means, the statement appears to broadly align with Recommendation 2(a). Lastly, during its first release in iOS 18.1, Apple Intelligence was off by default, and had to be manually activated by users in their device settings [15], which is in line with Recommendation 4. As of iOS 18.3, Apple Intelligence is on by default and "integrated across features in your apps" [20]. This is in violation of our Recommendation 4.

Case II (Server Model). Just like Case I, no user data is used to train Apple's PCC models [19], which is compliant with Recommendation 1. However, notwithstanding security-conscious design choices within Apple's PCC, processing inference queries inside trusted hardware provides a different kind of security from E2EE security, and thus is not compatible with E2EE under Recommendation 2 (even without considering the possibility of tool use). Therefore, in line with Recommendation 3, Apple should not advertise features relying on Case II as E2EE without qualifications; rather, Apple should clearly articulate the limitations introduced by the TEE processing, and explicitly state the contexts in which E2EE applications (both native and third-party) no longer meet E2EE guarantees. Such notices should be prominently displayed. Importantly, even users who have Apple Intelligence off should be aware of these limitations, due to the systemic E2EE properties mentioned in Section 2.1.3. Finally, during its first release in iOS 18.1, Apple Intelligence was off by default, and had to be manually activated by users [15], which is in line with Recommendation 4. However, as of iOS 18.3, Case II no longer adheres to Recommendation 4, as Apple Intelligence is on by default [20].

Case III (Third-Party Model). On Recommendation 1, OpenAI states that if users do not sign in to their ChatGPT accounts, then OpenAI does not use their queries to train its models [24]—thus, Case III processing is compliant with Recommendation 1 for users who are not logged in. However, if a user *is* logged into their OpenAI account and has not configured their settings otherwise[70], queries outsourced to ChatGPT may be used for training OpenAI models [157], directly opposing Recommendation 1. Thus, overall, Case III does not uphold Recommendation 1.

On Recommendation 2, if Siri is integrated within an application that is E2EE, then processing queries with ChatGPT via Siri would be in violation of Recommendation 2(b), as E2EE content would be used by a third party. This is also true if ChatGPT can respond to queries involving third-party E2EE or ADP-E2EE application content. Furthermore, if a user has logged into their OpenAI account and has not configured their settings otherwise, ChatGPT can log requests and use query data to train its models. Using ChatGPT to process queries on E2EE content would therefore be in violation of Recommendation 2(c), since the user's E2EE content would not solely be used for the purpose of fulfilling that user's requests.

Hence, according to Recommendation 3, Apple should not advertise features relying on Case III as E2EE without qualifications; it should clearly disclose the contexts in which E2EE data could be processed by third parties and how those third parties could use it.

[70] ChatGPT accounts have model training on by default [156].

Finally, since the ChatGPT extension in Siri is off by default and must be explicitly enabled in settings [24], Case III abides by Recommendation 4. Furthermore, it seems users are always made aware of when ChatGPT is being used to respond to a query [23]. Even if a user chooses not to be asked for permission before each query is sent to OpenAI, a small banner remains at the bottom of the response stating that it has been generated via ChatGPT.[71] An additional disclosure that content sent to ChatGPT is no longer E2EE could further clarify security guarantees to users.

[71] For an example of what this looks like, see this example from the Apple Intelligence press release [18].

C Meta AI Disclaimers in WhatsApp

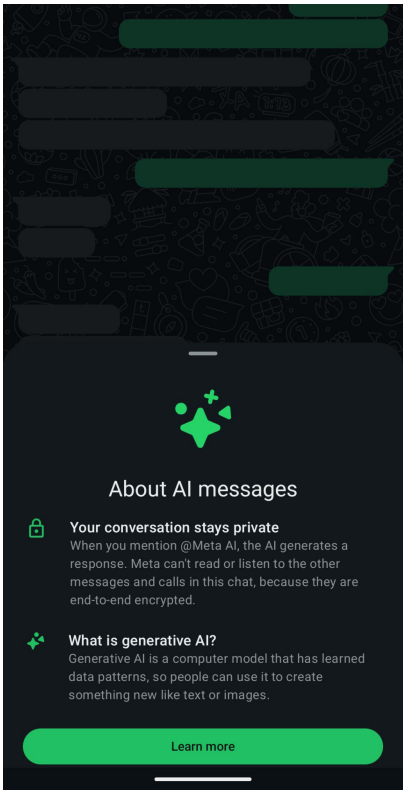

Figure 8: Meta AI's disclaimer when typing "@MetaAI" on WhatsApp for Android the first time, as of 02/14/25.

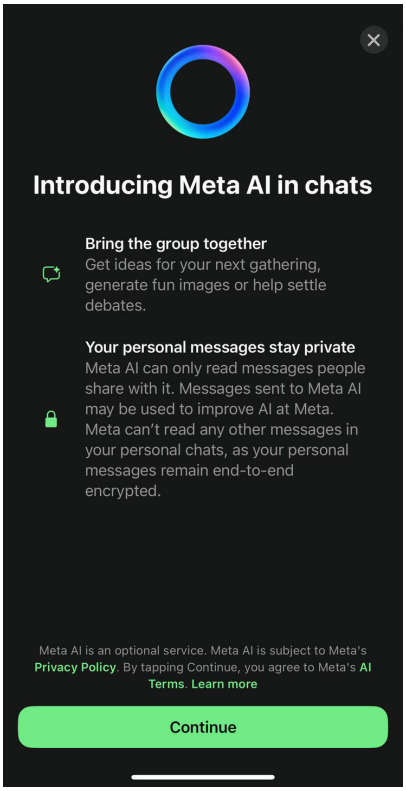

Figure 9: Meta AI's disclaimer when typing "@MetaAI" on WhatsApp for iOS for the first time, as of 03/15/25.

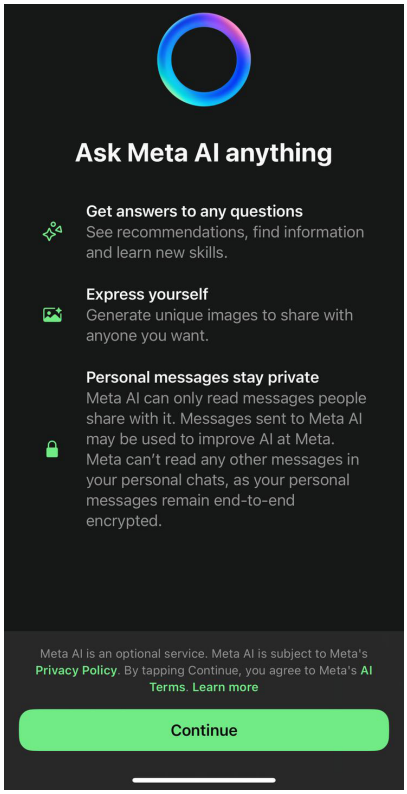

Figure 10: Meta AI's disclaimer when direct messaging with Meta AI on WhatsApp for iOS for the first time, as of 03/15/25.