Definition of End-to-end Encryption

Mallory Knodel[1], Sofía Celi[2], Olaf Kolkman[3], and Gurshabad Grover[4]

[1]New York University
[2]Brave
[3]Internet Society

23 December 2023

This document provides a definition of end-to-end encryption (E2EE). End-to-end encryption is an application of cryptographic mechanisms to provide security and privacy to communication between endpoints. Such communication can include messages, email, video, audio, and other forms of media. E2EE provides security and privacy through confidentiality, integrity, authenticity and forward secrecy for communication amongst people.

Contents

1 Introduction

End-to-end encryption is an application of cryptographic mechanisms to provide security and privacy to communication between endpoints. End-to-end encrypted systems provide security and privacy through confidentiality, integrity, authenticity and forward secrecy for communication amongst people. For the scope of this document, such communication can include messages, email, video, audio, and other forms of media.

Improvements to end-to-end encryption strive to maximize the user's security and privacy while balancing usability and availability.

1.1 End point

An "end" either sends messages or receives them, usually both. Other systems on the path are just that: other systems. Other systems MAY be used to facilitate the sending of messages between both "ends", but are not "ends" themselves.

It is, however, not trivial to establish the definition of an end point in isolation (Hale, 2022). Depending on the context, an "end" may be a device colocated with the user or a set of devices controlled by a user that want to simultaneously participate in the conversation.

1.2 End-to-end principle

The end-to-end principle is a core architectural guideline of the Internet (RFC3724).

The principle has evolved to an understanding that the "network's job is to transmit datagrams as efficiently and flexibly as possible", and the rest should be done at the ends (RFC1958). This principle can also be extended to the design of applications itself (Saltzer, 1984) (RFC3238).

1.3 Encryption

Encryption is the process of using cryptographic methods to convert plaintext to ciphertext that is decipherable only by authorized parties. Encryption can help extend the end-to-end principle in application design, where the function of the network is limited to efficiently transporting messages, but additionally the network cannot access any part of the message itself.

Encryption can be applied in an end-to-end context in many ways. For example, applications may use the double-ratchet algorithm (which uses an authenticated encryption scheme) and of an Authenticated Key Exchange (AKE). The usage of these algorithms (or variants of these) is present in many modern messenger applications such as those adopted in the IETF Messaging Layer Security working group, whose charter is to create a document that satisfies the need for several internet applications for group key establishment and message protection protocols (RFC9420). OpenPGP, mostly used for email, uses a different technique to achieve security and privacy. It is also chartered in the IETF to create a specification that covers object encryption, object signing, and identity certification (RFC9580). Both protocols rely on the use of asymmetric and symmetric encryption, and exchange long-term identity public keys amongst end points.

2 Formal definition of end-to-end encryption

An end-to-end-encryption system provides confidentiality, integrity, authenticity and forward secrecy between ends.

Confidentiality implies that a system that uses "end-to-end [...] encryption would conceal communications between one user's instant messaging application through any intermediate devices and servers all the way to the recipient's instant messaging application (Kahn Gillmor, 2015)." Thus confidentiality is broken if content can be decrypted at any intermediate point.

Integrity and authenticity requires that the application functions only for the end user and does not perform functions for any other entity coverly, nor overtly, say even if that entity claims to have obtained the consent of the end user. Thus, end point authenticity must be established as (sub-)identities of the end user, and end-to-end integrity must also be maintained by the system. There is considerable system design flexibility available in the mechanisms for authentication and integrity, specifically data authentication, that still meet this requirement. Thus integrity and authenticity, and by extention E2EE, are broken if data is manipulated or pseudo-identities created for third party access.

3 End-to-end encryption implementations

Below are fundamental properties that distinguish an end-to-end encrypted system from one that does not employ end-to-end encryption as well as the development challenges for improving the features of end-to-end encryption.

3.1 Properties

This section defines the security properties of an end-to-end encrypted system. The properties of end-to-end encryption from an implementation perspective can be split into two categories: 1) the required core properties of confidentiality, integrity, authenticity and forward secrecy; and 2) recommended additional properties for improved security, such as availability, deniability and post-compromise security, which are desirable enhancements.

3.1.1 Necessary properties

Confidentiality A system provides message confidentiality if only the sender and intended recipient(s) can read the message plaintext, i.e. messages sent between participants can only be read by the agreed upon participants in the group and all participants share the identical group member list.

Integrity A system provides message integrity when it guarantees that messages have not been modified in transit. If a message has been modified, it must be detected in a reliable way by the recipient.

Authentication A system provides authentication if the recipient and sender can verify each other's identities in relation to the contents of their communications.

Forward secrecy Forward secrecy is a security property that prevents attackers from decrypting encrypted data they have previously captured over a communication channel before the time of compromise, if the attacker compromises one of the endpoints. Forward secrecy is usually achieved by regularly deriving new encryption/decryption keys, and destroying old keys that are no longer required to encrypt or decrypt messages.

3.1.2 Optional/desirable properties and features

There is a set of optional/desirable features that a end-to-end system can provide. These properties can be related to the network, to the user interface or specialized variants of the previous features.

Availability A system provides high availability if the user is able to decrypt the contents of the message when they so desire and potentially from more than one device. For example, a message can arrive to a recipient even after they have been offline for a long time. Note that applications that use this feature often implement a threshold for this property: number or aggregate size of messages; or messages from a month ago can be read by a user that has been offline, but not messages from a year ago.

Loss Resilience If a message is permanently lost by the network, sender(s) and/or recipient(s) should still be able to communicate.

Deniability Deniability ensures that anyone able to decrypt a record of the transcript, including message recipients, cannot cryptographically prove to others that a particular participant of a communication authored a specific message. This optional property must exist in conjunction with the necessary property of authentication, i.e. participants in a communication must be assured that they are communicating with the intended parties but this assurance cannot be transmitted to any other parties.

Post-compromise security Post-compromise security is a security property that seeks to guarantee future confidentiality and integrity in the face of a passive end-point compromise and consequently that communication sent post-compromise is protected with the same security properties that existed before the compromise. It is usually achieved by adding new ephemeral key exchanges, ie new randomness, to the derivation of encryption/decryption keys every 'x' amount of time or after

’n’ messages sent. Note that post-compromise security is not met in the face of active attackers that compromise an end-point. This property can add a level of complexity to a protocol as deriving new key material can be expensive, and, therefore, it has to be carefully balanced as part of a system’s design.

Metadata obfuscation Digital communication inevitably generates data other than the content of the communication itself, such as IP addresses, user identifiers, group memberships, date and time of messages and size of messages. Inferred metadata includes interaction between IP addresses, time of first contact and frequencies of contact, login, and messages. To enhance the privacy and security of end-to-end encryption, steps should be taken to minimize, obscure or delete metadata.

Disappearing messages For confidential conversations, deleting one-by-one sensitive messages typically depends on a level of client-side security that is unsustainable. For example, endusers can still copy text or screenshot images outside the secured client application. A certain level of trust among users of the system is required. That said, manual actions like "delete for me" and "delete for everyone", or time-based automated deletion of content with "disppearing messages" still provide a valuable defense amongst trusted parties in the event of a compromise of a device of one of the participants.

3.2 Challenges

Below is a list of some challenges currently faced by designers of end-to-end encrypted systems.

- Making messaging applications interoperable is an important goal for a healthy and user-centric internet ecosystem, however it requires careful design of protocols and systems, such as content type negotiation; provisions of global services, such as discovery; and a great deal of cooperation amongst implementers.
- Public key verification is very difficult for users to manage. Authentication of the two ends is required for secure and private conversations. Therefore, solving the problem of verification of public keys is a major concern for any end-to-end encrypted system design. Some applications bind together the account identity and the key, and leave users to establish a trust relationship between them, assisted by public key fingerprint information.
- Users want to smoothly switch application use between devices, but this comes at a cost to security and privacy. Thus, there is a problem of availability in end-to-end encrypted systems because the account identity’s private key is generated by and stored on the end-user’s original device and moving the private key to another device can risk the compromise of the security of one of the end-points of the system, eg. by opening the door to key-impersonation attacks.
- Existing protocols are vulnerable to metadata analysis, even though metadata is often as sensitive as message content. Metadata is unencrypted (and sometimes unencryptable) information that travels through the network and includes delivery-relevant details that servers require such as the account identity of end-points, timestamps, message size or more. Metadata is difficult to eliminate or obfuscate entirely.
- Confidential and secure communications systems should also maintain the privacy of users but this is necessarily balanced with authentication and is related to the metadata problem for account identity. This can be particularly relevant for account recovery when the user has lost their credentials.
- Users need to communicate in groups, but this presents scalability problems for traditional end-to-end encryption systems. Message Layer Security protocol (RFC9420) is a modern end-to-end encrypted message protocol that addresses this scalability concern.
- The whole communication should remain secure if only one message is compromised. However, for encrypted communication, in some schemes, you must currently choose between forward secrecy or the ability to fully communicate asynchronously. This presents a problem for application design that uses end-to-end encryption for asynchronous messaging over email, RCS, etc.

- Users of end-to-end encrypted systems should be able to communicate with any medium of their choice, such as text, audio, video, or miscellaneous files. However, there is often a resource problem because there are no open protocols to allow users to securely share the same resource in an end-to-end encrypted system.
- Usability, accessibility and internationalisation features often need careful design and implementation with respect to security and privacy, such as message read status, typing indicators, URL/link previews, third-party input/output applications.
- End user security tools like anti-virus plugins, spam filters, fraud protections are in conflict with the security and privacy considerations of the end-to-end application.
- Deployment is notoriously challenging for any software application where maintenance and updates can be particularly disastrous for obsolete cryptographic libraries.

4 End-user expectations

While the formal definition and properties of an end-to-end encrypted system relate to communication security and privacy, they do not draw from a comprehensive threat model or speak to what users expect from end-to-end encrypted communication. It is in this context that some designs and architectures of end-to-end encryption may ultimately run contrary to user expectations of end-to-end encrypted systems (GEC, 2020). Although some system designs do not directly violate "the math" of encryption algorithms, they do so by implicating and weakening other important aspects of an end-to-end encrypted *system*.

4.1 A conversation is confidential

Users talking to one another in an end-to-end encrypted system should be the only ones that know what they are talking about (RFC7624).

4.2 Providers are trustworthy

Trustworthy A system is completely trustworthy if and only if it is completely resilient, reliable, accountable, and secure in a way that consistently meets users' expectations.

This definition is complete in its positive and negative aspects: what it is, e.g. "Worthy of confidence" and what it is not, e.g. in RFC 7258: "behavior that subverts the intent of communicating parties without the agreement of those parties" (RFC7258).

Therefore, a trustworthy end-to-end encrypted communication system is the provider of the set of functions needed by two or more parties to communicate among each other in a confidential, authenticated and integrity-preserving fashion without any third party having access to the content of that communication.

A proper implementation of end-to-end encryption significantly reduces the need of a user to trust a provider. However, this is contingent on users having some guarantee that the system actually works in conformance to the stated specification and security properties of end-to-end encryption. One way by which users can increase their trust in the system and confirm their system is performing in accordance to cryptographic protocols' specifications is using systems that are releasing their software as open source. Open source software allows technical users to analyse the system and be assured of its functioning. While most users will not be able to do so, as typical users lack the technological knowledge needed to analyse source code, technical communities can do so. It is vital that systems provide publicly accessible security analyses of their source code, enabling reproducible builds and audits and investigations that can be published and peer reviewed.

4.3 Access by a third-party is impossible

No matter the specifics, any methods used to access to the content of the messages by a third party would violate a user's expectations of end-to-end encrypted messaging. "[T]hese access methods scan message contents on the user's [device]", which are then "scanned for matches against a database of prohibited content before, and sometimes after, the message is sent to the recipient" (GEC, 2020). Third party access also covers cases without scanning – namely, it should not be possible for any third-party end point, even those under the user's identity as per Section 2.1, to access the data regardless of reason.

If a method makes secure and private communication, intended to be sent over an encrypted channel between end points, available to parties other than the sender and intended recipient(s), that method violates the understood expectation of that security property.

4.4 Pattern inference is minimised

Analyses such as traffic fingerprinting or other encrypted or unencrypted data analysis techniques, outside of or as part of end-to-end encrypted system design, allow third parties to draw inferences from communication that was intended to be confidential. "By allowing private user data to be scanned via direct access by servers and their providers," the use of these methods should be considered an affront to "the privacy expectations of users of end-to-end encrypted communication systems" (GEC, 2020).

Not only should an end-to-end encrypted system value user data privacy by not explicitly enabling pattern inference, it should actively be attempting to solve issues of metadata and traceability (enhanced metadata) through further innovation that stays ahead of advances in these techniques.

4.5 The end-to-end encryption is not compromised

RFC 3552 talks about the Internet Threat model such as the assumption that the user can expect any communications systems, but perhaps especially end-to-end encrypted systems, to not be intentionally compromised (RFC3552). Intentional compromises of end-to-end encryption are usually referred to as "backdoors" but are often presented as additional design features under terms like "key escrow" or "exceptional access". Users of end-to-end encryption would not expect a front, back or side door entrance into their confidential conversations and would expect a provider to actively resist – technically and legally – compromise through these means.

5 Conclusions

From messaging to video conferencing, there are many competing features in an end-to-end encrypted implementation that is secure, private and usable. The most well designed system cannot meet the expectations of every user, nor does an ideal system exist from any dimension. End-to-end encryption is a technology that is constantly improving to achieve the ideal as defined in this document.

Features and functionalities of end-to-end encryption should be developed and improved in service of end user expectations for privacy preserving communications.

6 Acknowledgements

Fred Baker, Stephen Farrell and Richard Barnes contributed to the early strategic thinking of this document and whether it would be useful to the IETF community.

The folks at Riseup and the LEAP Encryption Access Project have articulated brilliantly the hardest parts of end-to-end encryption systems that serve the end users' right to whisper.

Ryan Polk at the Internet Society has energy to spare when it comes to organising meaningful contributions, like this one, for the technical advisors of the Global Encryption Coalition.

Reviews, comments and questions from Paul Wouters, Adrian Farrel, Eric Rescorla and David Black led to improvement of this document.

Chelsea Komlo and Britta Hale have contributed their deep expertise and consice and rigourous writing to this draft.

7 References

[GEC] Global Encryption Coalition, “Breaking Encryption Myths,” Global Encryption Coalition. Accessed: Dec. 18, 2024. [Online]. Available: https://www.globalencryption.org/2020/11/breaking-encryption-myths

[Kahn Gillmor] D. Kahn Gillmor, N. ten Oever, and avri doria, “Human Rights Protocol Considerations Glossary,” Internet Engineering Task Force, Internet Draft draft-dkg-hrpc-glossary-01, Oct. 2015. Accessed: Dec. 18, 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-dkg-hrpc-glossary

[Hale] B. Hale and C. Komlo, “On End-to-End Encryption,” 2022, 2022/449. Accessed: Dec. 18, 2024. [Online]. Available: https://eprint.iacr.org/2022/449

[Saltzer] J. H. Saltzer, D. P. Reed, and D. D. Clark, “End-to-end arguments in system design,” ACM Trans. Comput. Syst., vol. 2, no. 4, pp. 277–288, Nov. 1984, doi: 10.1145/357401.357402.