mls
Internet-Draft
Intended status: Informational
Expires: January 13, 2022

M. Knodel
CDT
F. Baker

O. Kolkman
ISOC
S. Celi
Cloudflare
G. Grover
Centre for Internet and Society

July 12, 2021

Definition of End-to-end Encryption
draft-knodel-e2ee-definition-02

Abstract

End-to-end encryption (E2EE) is an application of cryptography in communications systems between endpoints. E2EE systems are unique in providing features of confidentiality, integrity and authenticity for users. Improvements to E2EE strive to maximise the system's security while balancing usability and availability. Users of E2EE communications expect trustworthy providers of secure implementations to respect and protect their right to whisper.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2022.

Table of Contents

1. Introduction

This document defines end-to-end encryption (E2EE) using three different dimensions that together comprise a full definition of E2EE, which can be applied in a variety of contexts.

The first is a formal definition that draws on the basic understanding of end points and cryptography. The second looks at E2EE systems from a design perspective, both its fundamental features and the direction of travel towards improving those features. Lastly we consider the expectations of the user of E2EE systems.

These dimensions taken as a whole comprise a generally comprehensible picture of consensus at the IETF as to what is end-to-end encryption, irrespective of application, from messaging to video conferencing, and between any number of end points.

2. Formal definition of end-to-end encryption

An end-to-end encrypted communications system, irrespective of the content or the specific methods employed, relies on two important and rigorous technical concepts: The end-to-end principle and what defines an end, according to the IETF because of its importance to internet protocols; and encryption, an application of cryptography and the primary means employed by the IETF to secure internet protocols. In the tradition of cryptography it's also possible to achieve a succinct definition of end-to-end encrypted security.

2.1. End point

Intuitively, an "end" either sends messages or receives them, usually both; other systems on the path are just that - other systems.

It is, however, not trivial to establish the definition of an end point in isolation, because its existence inherently depends on at least one other entity in a communications system. That is why we will now move directly into an analysis of the end-to-end principle, which introduces nuance, described in the following sub-section.

However despite the nuance for engineers, it is now widely accepted that the communication system itself begins and ends with the user [RFC8890]. We imagine people (through an application's user interface, or user agent) as components in a subsystem's design. An important exception to this in E2EE systems might be the use of public key infrastructure where a third party is often used in the authentication phase to enhance the larger system's trust model. Responsible use of of public key infrastructure is required in such cases, such that the E2EE system does not admit third parties under the user's identity.

We cannot equate user agent and user, yet we also cannot fully separate them. As user-agent computing becomes more complex and often more proprietary, the user agent becomes less of an "advocate" for the best interests of the user. This is why we focus in a later section on the E2EE system being able to fulfill user expectations.

2.2. End-to-end principle

We need first to answer "What constitutes an end?", which is an important question in any review of the End-to-End Principle [RFC3724]. However the notion of an end point is more fully defined within the principle of end-to-end communications.

In 1984 the "end-to-end argument" was introduced [saltzer] as a design principle that helps guide placement of functions among the modules of a distributed computer system. It suggests that functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level. It is used to design around questions about which parts of the system should make which decisions, and as such the identity of the actual "speaker" or "end" may be less obvious than it appears. The communication described by Saltzer is between communicating processes, which may or may not be on the same physical machine, and may be implemented in various ways. For example, a BGP speaker is often implemented as a process that manages the Routing Information Base (RIB) and communicates with other BGP speakers using an operating system service that implements TCP. The RIB manager might find itself searching the RIB for prefixes that should be advertised to a peer, and performing "writes" to TCP for each one. TCP in this context often implements a variant of the algorithm described in RFC 868 (the "Nagle algorithm"), which accumulates writes in a buffer until there is no data in flight between the communicants, and then sends it - which might happen several times during a single search by the RIB manager. In that sense, the RIB manager might be thought of as the "end", because it decides what should be communicated, or TCP might be the "end", because it actually sends the TCP Segment, detects errors if they occur, retransmits it if necessary, and ultimately decides that the segment has been successfully transferred.

Another important question is "what statement exactly summarizes the end-to-end principle?". Saltzer answered this in two ways, the first of which is that the service implementing the transaction is most correct if it implements the intent of the application that sent it, which would be to move the message toward the destination address in the relevant IP header. Salzer's more thorough treatment, however, deals with end cases that come up in implementation: "Examples discussed in the paper", according to the abstract, "include bit error recovery, security using encryption, duplicate message suppression, recovery from system crashes, and delivery acknowledgement." It also notes that there is occasionally a rationale for ignoring the end-to-end arguments for the purposes of optimization. There may be other user expectations or design

features, some explained below, which need to be balanced with the end-to-end argument.

More concisely, suppose that an end user is the end identity. An E2EE system may run between potential end points at different network layers within the end identity's possession. These end points may then be considered acceptable sub-identities provided that no path between the end identity and sub-identity is accessible by any third party. This definition of end points accounts for potentially several devices owned by a user, and various application-specific forwarding or delivery options among them. It also accounts for E2EE systems running at different network layers. Regardless of the sub-identities allowed, the definition is contingent on that all end sub-identities are under the end identity's control and no third party (or their sub-identities, e.g. system components under third-party control) can access the end sub-identities nor links between the sub-identity and end identity. This creates a tree hierarchy with the end user as the root at the top, and all potential end points being under their direct control, without third party access. As an example, decryption at organizational network router before message forwarding (encrypted or unencrypted) to the end identity does not constitute E2EE. However, E2EE to a user's personal device and subsequent E2EE message forwarding to another one of the user's personal devices (without access available to any third party at any link or on device) maintains E2EE data possession for the user.

2.3. Encryption

From draft-dkg-hrpc-glossary-00, encryption is fundamental to the end-to-end principle. "End-to-End : The principal of extending characteristics of a protocol or system as far as possible within the system. For example, end-to-end instant message encryption would conceal communication content from one user's instant messaging application through any intermediate devices and servers all the way to the recipient's instant messaging application. If the message was decrypted at any intermediate point-for example at a service provider-then the property of end-to-end encryption would not be present."[dkg] Note that this only talks about the contents of the communication and not the metadata generated from it.

The way to achieve a truly end-to-end communications system is indeed to encrypt the content of the data exchanged between the endpoints, e.g. sender(s) and receiver(s). The more common end-to-end technique for encrypting uses a double-ratchet algorithm with an authenticated encryption scheme, present in many modern messenger applications such as those considered 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 [mls]. OpenPGP, mostly used for email, uses a different technique to achieve encryption. It is also chartered in the IETF to create a specification that covers object encryption, object signing, and identity certification [openpgp]. Both protocols rely on the use of asymmetric and symmetric encryption, and exchange public keys with amongst end points.

There are dozens of documents in the RFC Series that fundamentally and technically define encryption schemes. Perhaps interesting work to be done would be to survey all existing documents of this kind to define, in aggregate, their common features. The point is, the IETF has clear mandate and demonstrated expertise in defining the specifics of encrypted communications of the internet.

While encryption is fundamental to the end-to-end principle, it does not stand alone. As in the history of all security, authentication and data integrity properties are also linked, and contributed to the end-to-end nature of E2EE. Permission of data manipulation or pseudo-identities for third parties to allow access under the user's identity are against the intention of E2EE. 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 entity authentication mechanisms and data authentication that still meet this requirement.

2.4. Succinct definition of end-to-end security

A succinct definition for end-to-end security can describe the security of the system by the probability of an adversary's success in breaking the system. Example snippet:

The adversary successfully subverts an end-to-end encrypted system if it can succeed in either of the following: 1) the adversary can produce the participant's local state (meaning the adversary has learned the contents of participant's messages), or 2) the states of conversation participants do not match (meaning that the adversary has influenced their communication in some way). To prevent the adversary from trivially winning, we do not allow the adversary to compromise the participants' local state.

We can say that a system is end-to-end secure if the adversary has negligible probability of success in either of these two scenarios [komlo].

3. End-to-end encrypted systems design

When looking at E2EE systems from a design perspective, the first consideration is the list of fundamental features that distinguish an E2EE system from one that does not employ E2EE. Secondly one must consider the direction of travel for improving the features of E2EE systems. In other words, what challenges are the designers, developers and implementers of E2EE systems facing?

The features and challenges listed below are framed holistically rather than from the perspective of their design, development, implementation or use.

3.1. Features

Defining a technology can also be done by inspecting what it does, or is meant to do, in the form of features. The features of end-to-end encryption from an implementation perspective can be inspected across several important categories: 1) the necessary features of E2EE of authenticity, confidentiality, and integrity, whereas features of 2) availability, deniability, forward secrecy, and post-compromise security are enhancements to E2EE systems.

3.1.1. Necessary features

Authenticity A system provides message authenticity if the recipient is certain who sent the message and the sender is certain who received it.

Confidentiality A system provides message confidentiality if only the sender and intended recipient(s) can read the message plaintext, i.e. messages are encrypted by the sender such that only the intended recipient(s) can decrypt them.

Integrity A system provides message integrity when it guarantees that messages has not been modified in transit, i.e. a recipient is assured that the message they have received is exactly what the sender intended to send.

3.1.2. Optional/desirable features

Availability A system provides high availability if the user is able to get to the message when they so desire and potentially from more than one device, i.e. a message arrives to a recipient even if they have been offline for a long time.

Deniability Deniability ensures that anyone with a record of the transcript, including message recipients, cannot cryptographically

prove to others that a particular participant of a communication authored the message. As demonstrated by the Signal and OTR protocols, this optional property must exist in conjunction with the necessary property of message authenticity, i.e. participants in a communication must be assured that they are communicating with the intended parties but this assurance cannot be proof to any other parties.

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, even if they have compromised one of the endpoints. Forward secrecy is usually achieved by updating the encryption/ decryption keys, and older ones are deleted periodically.

Post-compromise security Post-compromise security is a security property that seeks to guarantee a way to recover from an 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 ephemeral key exchanges to the derivation of encryption/decryption keys.

3.2. Challenges

Earlier we defined end-to-end encryption using formal definitions assumed by internet protocol implementations. Also because "the IETF is a place for state-of-the-art producing high quality, relevant technical documents that influence the way people design, use, and manage the Internet" we can be confident that current deployments of end-to-end encrypted technologies in the IETF indicate the cutting edge of their developments, yet another way to define what is, or ideally should be, how a technology is defined.

Below is an exhaustive, yet vaguely summarised, list of the challenges currently faced by protocol designers of end-to-end encrypted systems. In other words, in order to realise the goals of end-to-end encrypted systems, both for users and implementers (see previous section), these problems must be tackled. Problems that fall outside of this list are likely 1) unnecessary feature requests that negligibly, or do nothing to, achieve the aims of end-to-end encrypted systems or are 2) in some way antithetical to the goals of end-to-end encrypted systems.

Public key verification is very difficult for users to manage. Authentication of the two ends is required for confidential 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 the security of user data. 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 to move the private key to another device compromises the security of one of the end-points of the system.

Existing protocols are vulnerable to meta-data analysis, even though meta-data is often much more sensitive than content. Meta-data is plaintext information that travels across the wire and includes delivery-relevant details that central servers need such as the account identity of end-points, timestamps, message size. Meta-data is difficult to obfuscate efficiently.

Users need to communicate in groups, but this presents major problems of scale for end-to-end encryption systems that rely on public key cryptography.

The whole of a user's data should remain secure if only one message is compromised. However, for encrypted communication, you must currently choose between forward secrecy or the ability to communicate asynchronously. This presents a problem for application design that uses end-to-end encryption for asynchronous messaging over email, RCS, etc.

Users of E2EE systems should be able to communicate with any medium of their choice, from text to large 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. Client-side, e.g. end-point, activities like URL unfurling scanning.

Usability considerations are sometimes in conflict with security considerations, such as message read status, typing indicators, URL/link previews.

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 E2EE system relate to communication security, they do not draw from a comprehensive threat model or speak to what users expect from E2EE communication. It is in this context that some E2EE designs and architectures may ultimately run contrary to user expectations of E2EE systems [GEC-EU]. 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 E2EE _system_.

4.1. A conversation is confidential

Users talking to one another in an E2EE system should be the only ones that know what they are talking about [RFC7624]. People have the right to data privacy as defined in international human rights law and within the right to free expression and to hold opinions is inferred the right to whisper, whether or not they are using digital communications or walking through a field.

4.2. Providers are trustworthy

While "trustworthy" can be rigourously defined from an engineering perspective, for the purposes of this document we choose a definition of Trustworthy inspired by an internal workshop by Internet Society staff:

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. The opposite of trustworthy is untrustworthy.

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 set of functions needed by two or more parties to communicate among each other in a confidential and authenticated fashion without any third party having access to the content of that communication where the functions that offer the confidentiality and authenticity are trustworthy.

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 E2EE 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-EU]. Third party access also covers cases without scanning - namely, it should be possible for any third-party end point to access the data regardless of reason.

If a method makes private communication, intended to be sent over an encrypted channel between end points, available to parties other than the sender and intended recipient(s), without formally interfering with channel confidentiality, 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 should be considered outside the scope of an E2EE system's goals of providing secure communications to end users.

Such methods of analyses, outside of or as part of E2EE 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-EU].

Not only should an E2EE system value user data privacy by not 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 E2EE system 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 E2EE systems, to not be intentionally compromised [RFC3552]. Intentional compromises of E2EE systems are often referred to as "backdoors" but are often presented as additional design features under terms like "key escrow." Users of E2EE systems 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 E2EE system that is secure and usable. The most well designed system cannot meet the expectations of every user, nor does an ideal system exist from any dimension. E2EE is a technology that is constantly improving to achieve the ideal as defined in this document.

Features and functionalities of E2EE systems should be developed and improved in service of end user expectations for privacy preserving communications.

6. Acknowledgements

Fred Baker, Stephen Farrell, Richard Barnes, Olaf Kolkman all 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.

7. Security Considerations

As this draft concerns an informational document, there are no security considerations.

8. IANA Considerations

This document has no actions for IANA.

9. Informative References

[dkg] Gillmor, D., "Human Rights Protocol Considerations Glossary", 2015, <https://tools.ietf.org/html/draft-dkg-hrpc-glossary-00>.

[GEC-EU] Global Encryption Coaliation, ., "Breaking encryption myths: What the European Commission's leaked report got wrong about online security", 2020, <https://www.globalencryption.org/2020/11/breaking-encryption-myths/>.

[komlo] Chelsea Komlo, ., "Defining end-to-end security", 2021, <https://github.com/chelseakomlo/e2ee/blob/master/e2ee_definition.pdf>.

[mls] IETF, ., "Messaging Layer Security", 2018, <https://datatracker.ietf.org/doc/charter-ietf-mls>.

[openpgp] IETF, ., "Open Specification for Pretty Good Privacy", 2020, <https://datatracker.ietf.org/doc/charter-ietf-openpgp>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC3724] Kempf, J., Ed., Austein, R., Ed., and IAB, "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture", RFC 3724, DOI 10.17487/RFC3724, March 2004, <https://www.rfc-editor.org/info/rfc3724>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <https://www.rfc-editor.org/info/rfc7624>.

[RFC8890] Nottingham, M., "The Internet is for End Users", RFC 8890, DOI 10.17487/RFC8890, August 2020, <https://www.rfc-editor.org/info/rfc8890>.

[saltzer] Saltzer, et al, J., "End-to-end arguments in system design", 1984, <https://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf>.

Authors' Addresses

Mallory Knodel
CDT

Email: mknodel@cdt.org

Fred Baker

Email: fredbaker.IETF@gmail.com

Olaf Kolkman
ISOC

Email: kolkman@isoc.org

Sofia Celi
Cloudflare

Email: cherenkov@riseup.net

Gurshabad Grover
Centre for Internet and Society

Email: gurshabad@cis-india.org