Communication-Encryption

> [!toc] Table of Contents

The encryption of all communication plays an essential role in our digital lives. In this article, we want to explain what communication encryption means, what different types of encryption exist, and what advantages and disadvantages they have.

We distinguish between transport encryption and end-to-end encryption (E2EE).

> [!info] TL;DR {static}
>
> While transport encryption is a nice-to-have, it is in no way sufficient for most use cases - we recommend using end-to-end encryption (E2EE) whenever possible.

Transport encryption

Transport encryption is generally implemented with SSL/TLS. Those are encryption-based Internet security protocols that provide privacy, authentication, and integrity to Internet communications You are using SSL/TLS everyday in your browser, for example, when a padlock appears next to the URL and `https` appears before the URI. If this is not used, only `http` appears (and in most cases a warning appears that the connection is not secure).

We will use the graphic below and a practical example to explain how transport encryption works.

Example: Mail with transport encryption

Anna wants to send Arthur a message, for example by email. The example also works with other services without E2EE, such as Telegram, Discord, or chats in games. However, then there would only be one server instead of two.

Here is the example with email:

Anna has an email address on the yellow server; in our example, it would be `systemli.org`. Her email address is therefore `anna@systemli.org`

Arthur has an email address on the red server, in this case `riseup.net`. His email address is therefore `arthur@riseup.net`

Because we are talking about transport encryption, neither of them uses E2EE such as PGP. This means that Anna does not have Arthur's PGP key, and vice versa!

The keys and locks in the graphic below symbolize so-called **certificates**. Each server has its own certificate with which communication to it can be encrypted. Only the server in possession of the certificate also has the corresponding key an can read the information that is sent to it.

If Anna now wants to write an email, she retrieves the certificate from Systemli (yellow lock) and uses it to encrypt her email. This is completely independent of who the email will ultimately be sent to! Arthur's receiving address (mailto:arthur@riseup.net) is then written on the "envelope", just like with normal mail. This email (yellow, sealed envelope with a lock) is then sent to the Systemli mail server (yellow box).

The Systemli mail server now opens the email encrypted with its own certificate and scans it for spam, for example. Above all, it looks at the recipient address on the envelope: `arthur@riseup.net`. The server recognizes the part after the `@` symbol as the mail server to which it must forward this email: `riseup.net` (red server). So it quickly goes over to Riseup, grabs a copy of their certificate, encrypts Anna's email again with it, and sends it (red, locked envelope with lock) to the Riseup mail server.

From here, this process repeats itself until the email reaches Arthur. The Riseup server unpacks the email, repacks it, and finally sends it to Arthur.

Transport encryption graphic

Problem

The problem here is obvious. Every participant in the communication chain can easily open and read the email. In addition, many applications (as listed above) store copies of the messages on their (email) servers. See more about network surveillance

End-to-end encryption

Once you understand the threat posed by transport encryption, the need for end-to-end encryption is almost self-explanatory.

1. Anna obtains Arthur's lock (public key). This point is very important; please note the [TOFU] section!
2. Same as in 1.
3. Anna encrypts her message with Arthur's public key.
4. The message remains encrypted in all steps of 4 (a-e). Only the metadata (e.g., sender/recipient address) is visible (at all possible points, including during transport!) and is read by the servers in order to forward the email.
5. Arthur receives his message. Because the message was encrypted with his padlock and he has taken good care of his key (private key), only he can decrypt the message with his key.

End-to-end encryption graphic

TOFU is bad

TOFU: Trust On First Use

The key must be verified “out of band.” An unencrypted (i.e., transport-encrypted) email makes the exchange of public keys vulnerable to interception. This is called a "machine-in-the-middle attack".

Graphic machine-in-the-middle attack

For more information on the dangers of transport encryption, see network surveillance.

Version #4
Erstellt: 2026-02-15 16:16:21 UTC von ESC-IT Migration Bot
Zuletzt aktualisiert: 2026-03-20 20:09:26 UTC von ESC-IT Migration Bot