Qlink

Advanced Frequently Asked Questions

This document tries to address a set of more technical points about our service. There is no need of technical background for most of the answers, however some of them may require a higher level of knowledge for its full comprehension.

How can I be sure that the Qlink.it server does not save or read the messages I send or receive ?

The qlink server *cannot* read your message because of the way the the Qlink.it security protocol was designed. Your message reaches the server already encrypted with a key that you have it but is never sent to the server. This is one of the great values of Qlink.it! Let's explore this feature by describing in detail how Qlink.it works:

1. When you enter a message in qlink.it and click the "qlink it!" button, your browser runs a Javascript program which encrypts the message with a given random key, say for instance YYYYYY.
2. Afterwards, the encrypted message is sent through secure https protocol to the Qlink.it server.
3. At the server, the message (already encrypted with key YYYYY) is encrypted again to be stored, but now with another random key, say for instance XXXXXX.
4. Then, the server returns to you a preliminary qlink, in this case https://qlink.it/XXXXXX.
5. At that moment, your browser adds at the end of the preliminary qlink the key that only your browser knows to form the full qlink: https://qlink.it/XXXXXX#YYYYYY. Notice that the Qlink.it server didn't have access to the YYYYYY part of the qlink!
6. Then, you copy & paste the full qlink and send it to the intended recipient, either by email, chat, WhatsApp, or whatever.
7. When the recipient receives the full qlink and clicks on it, the browser *only requests* to the server the preliminary qlink, https://qlink.it/XXXXXX, because the special character hash mark (#) indicates that what follows should not be sent through the internet! (You can check this feature by using for instance the *inspect* option in some browsers as could be Chrome.) Therefore, the Qlink.it server never has access to the full key to read the true content of the message!
8. When the server receives the request with the preliminary qlink, the qlink has in it the key to look for the encrypted message and partially decipher it. The server then sends back through https secure protocol a message which is still encrypted with the unknown-to-the-server key YYYYYY. At that moment the server makes a secure delete on the encrypted message and is not available any more at the server.
9. When the recipient's browser gets the encrypted message, since it kept the last part of the full qlink YYYYYY, it runs a Java script to finally decipher the encrypted message using

this last part of the full qlink. Once the message is totally deciphered, the browser displays it on the recipient's screen.

Therefore, when you send or receive a qlinked message, only in the full qlink it is contained all the information to access and to decipher the message. Only if you have the full qlink can you read the content of the message, and the Qlink.it server only has one part of the full qlink!

In any case, it is worth noticing that the last thing we are interested is knowing what is being transmitted using our service. Our business is to provide security and privacy in a transparent way.

Finally, it is also interesting to point out that in the way that Qlink.it is designed, the qlink is sent and received by the sender and recipient's own e-mail, chat, WhatsApp or whatever agent, therefore the Qlink.it server can never establish who is the sender and who is the recipient of the encrypted message.

What's the point of knowing that my message was intercepted if it was already read? Isn't much more important to have a very-hard-to-crack encryption?!

No, in the practice of secure transmissions it is as important to have a very hard encryption as to know if the message was intercepted!

In Qlink.it the encrypted message is sent through https using a 4096-bits key, which is the latest web standard security. Therefore the encryption is as much, if not higher, than usual messages sent through the web.

In addition, Qlink.it has the auto-deletion feature which works as a trigger to tell if the message was intercepted. Therefore, reading the message is the confirmation that the message was not intercepted.

As a powerful example of how important is to have the knowledge if a message is intercepted, consider the World War II Third Reich's principal cryptography system, the *enigma*. This was an extremely sophisticated cryptography system, but the Allies did manage to partially crack it using scientific studies and some fortuitous events. As a result, the allies used the available information to predict movements which were decisive to the final outcome of World War II, as stated by Eisenhower itself. Had the Third Reich's known that their system was cracked the results could have turned out differently.

What for is the 'imprint IP' option ?

If you check the 'imprint IP' option box, then the Qlink.it server includes as a footnote to the message the original sender's full IP. This is useful for the –extremely unlikely– case in which you don't trust the email server (or WhatsApp, or Skype IM, or whatever) through which you are sending the qlink. In fact, in the unlikely case the email server would have a routine which reads the qlink and generates a new one with the same content, then this feature will catch up the cheating, because the IP that reaches the recipient will be that of the email server and not the one of the original sender.

Where and how is my confidential information stored in the Qlink.it server ?

Once you generate a qlink, the info that arrives to the server is treated differently depending if it's the encrypted typed text message or the encrypted attached file.

The encrypted text message is encrypted again and then is stored in a database that runs exclusively on the RAM of the server. That is, the info never touches a hard disk and is secure-deleted after the qlink is requested. To reach the location and to decipher the stored info, it can be only done using the qlink, which contains the required keys.

The encrypted attached file are treated with the same level of security of the text message but, due to technical limitations, this info is written to a hard disk and then is secure-deleted once the qlink is requested to the server.

We use Redis database running on the RAM to store text messages, and on the HD for attachments. We have modified the original source code to suppress all options that could become a potential security leak. Only the Qlink.it web server can access the stored information and only with the keys stored in the qlink. We have also modified the software in order to encapsulate the reading and secure-deleting of the message in only one request. In this way, the secure-delete is executed simultaneously with the reading request.

What is it "secure-delete" ?

Secure-delete is an advanced tool for completely deleting files from a hard drive. The working of secure-delete consists in writing over *every byte* of the file to be secure-deleted. In this way, after a file has been secure-deleted, there is no more access to any piece of it at all. (Notice that usual delete commands only delete a very small part of all the bytes in the file.)

What information does Qlink.it keep from me and my message ?

We are interesting in providing security and confidentiality, therefore we are interested in not keeping anything from you. That is, we secure-delete all the encrypted messages that we store. Moreover, we do not have access to know who is the sender nor the recipient of each qlink, since that info goes through different e-mail, chats, etc. services.

Only for analytic purposes we periodically run on the server Piwik, an open source statistic software which at the end is requested to keep only the first two numbers of the visitor's IP. However, user can avoid being surveyed by selecting the *no-track* option in their browser and in that case Piwik doesn't survey the visitor.

What is it better to use: Qlink.it or a public/private key encryption ?

Qlink.it is different, and eventually complementary, to a public/private key encryption. Let's go into the details of this fruitful comparison.

A public/private key encryption is a really great invention which is very secure as far as you keep secure your private key. But it can always happen that your private key gets stolen, or you

download by accident an application that sneaks in your device and gets your private key, which is not impossible at all. In that case, all the information you are sending could be read and you would never get to know it. And this is a real disaster for you.

The other problem with private/public key encryption is that you and also the recipient need to install special software to use it. This makes it not so useful in the practice, let's face reality: the invention is from ~1975 and still very few people use it. Only experts use it.

On the other hand Qlink.it is very simple to use and the info goes encrypted with a key that you share it in such a way that the server does not have access to it. Only in the case that your messenger company spies the links, your info would be stolen. However, you would get to know it right away! And since this is the case, and everybody knows it, then the messenger company would not touch the qlink, it is a *checkmate*.

Summarizing, both methods have their pros and cons:

Private/public key:
1) PRO: it is very secure and in principle cannot be read in the middle.
2) CON: if your private key is stolen, all your info is read and you never know it.
3) CON: it requires both parties to have special software installed in their devices.

Qlink.it:
4) PRO: it is very simple and secure, even if other application sneaks into your device looking for keys.
5) PRO: you get to know if somebody reads your messages.
6) CON: if your messenger company is an spy, it can read your message. But this is a *checkmate*, if the company would read it, it can do it only once, since after that everybody would know it and everybody would stop using that company. Therefore, the company doesn't want to do that move.

However, summarizing, we find that -as far as we know- the perfect mechanism for sending confidential info through the net would be to send a qlink through a public/private key encryption. In this perfect mechanism you get rid of the CONS 2) and 6), while keeping the PROS 1), 5) and half of 4). By the time being, and due to its complexity level for the user, Qlink.it is only providing the qlink piece of the perfect mechanism. The public/private key piece could be found elsewhere, as for instance in protonmail, mailvelope, cryptocat, etc.

How *secure* is Qlink.it ?

As we always insist, the is no 100% secure way of sending private information. If somebody asserts they have a 100% secure way of sending private info through the net, he or she is either lying to you or lacks the relevant knowledge to understand the problem.

Usually, a method for sending confidential info has a *theory* and an *implementation.* Many times the theory is quite robust and the potential security breaches come from its implementation. Even Quantum Cryptography - the most secure available encryption method - has potential security breaches in its implementation, which are of course sophisticated. (You can tell us

about a known encryption method and we'll tell you potential security problems it has … give it a shot!)

Concerning Qlink.it, we found the following sophisticated potential security breach: If an external agent could *simultaneously*

i) find out which is the messenger service you are using to send the qlink,
ii) crack/backdoor the messenger service to read the qlink,
Iii) crack/backdoor the Qlink.it server to read the encrypted message,

then it could have access to the message without noticing it. If one could avoid the external agent to accomplish *only one* of these points, then the security can be restored. Let's explore this situation.

Point *i)* is exclusively the user's responsibility. Point *ii)* depends on the messenger service, and is recommended to use a reliable one. Point *iii)* depends on Qlink.it. We put all our effort in having the most secure server, which we have it working as a black box, with a database running on RAM which gets practically destroyed if somebody attempts to touch it, among other security features. Besides these features, we are constantly thinking, brain-storming, discussing and improving the server security features.

However, it could be the extreme case that a user, or organization, would like to verify the server security or even have the server under their control. In this sense, we would be very glad to hear concerns about this point and talk about it. Upon request, and confidentiality commitments, we can open the server's source to verify its high level of security. For extremely high security organizations, as for instance National Security and highly compromised ONGs, we also offer the possibility of installing a new server service which could be under their own control. Moreover, for the case of non-profit organizations we could offer the rights of this service for free.

Do you still have more questions, comments or ideas concerning security at Qlink.it ?

Contact us through: security at qlink dot it, we reward good ideas and comments.