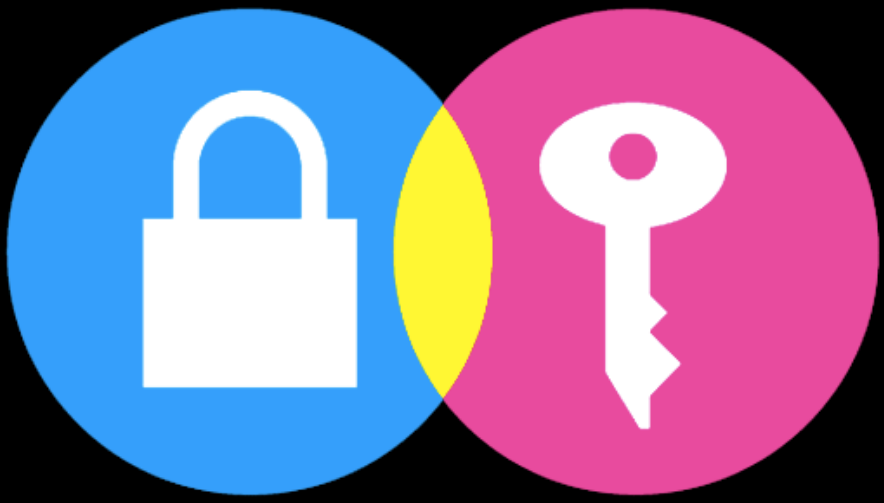


SECURE MESSAGING SCORECARD

Which apps and tools actually keep your messages safe?

In the face of widespread Internet surveillance, we need a secure and practical means of talking to each other from our phones and computers. Many companies offer "secure messaging" products—but are these systems actually secure? We decided to find out, in the first phase of a new EFF Campaign for Secure & Usable Crypto.

This scorecard represents only the first phase of the campaign. In later phases, we are planning to offer closer examinations of the usability and security of the tools that score the highest here. As such, the results in the scorecard below should not be read as endorsements of individual tools or guarantees of their security; they are merely indications that the projects are on the right track. For practical advice and tutorials on how to protect your online communication against surveillance, check out EFF's Surveillance Self-Defense guide.

Are past

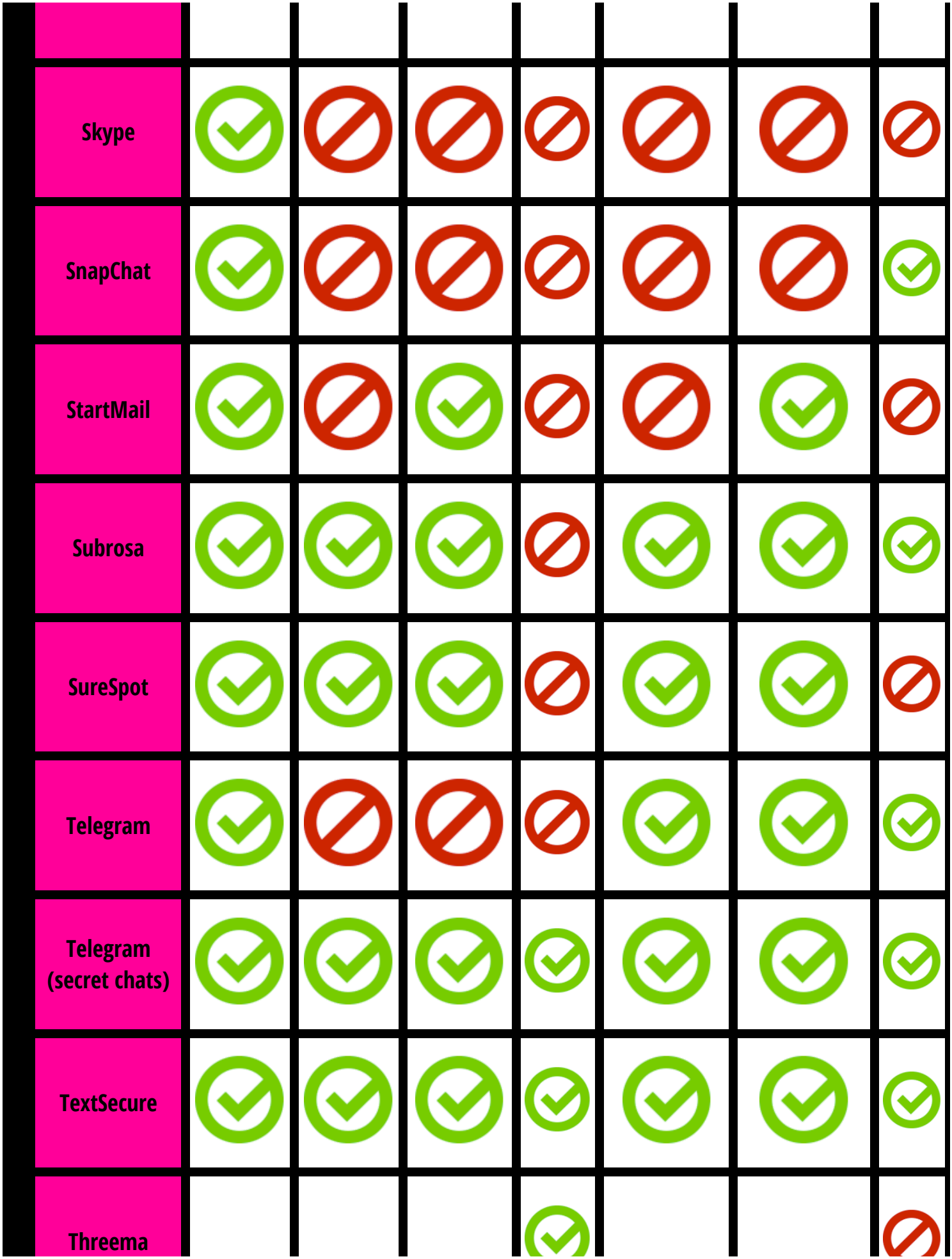
Has

All Tools

	Encrypted in transit?	Encrypted so the provider can't read it?	Can you verify contacts' identities?	Past comms secure if your keys are stolen?	Is the code open to independent review?	Is security design properly documented?	Has there been any recent code audit?
AIM							
BlackBerry Messenger							
BlackBerry Protected							
ChatSecure + Orbot							
CryptoCat							
Ebuddy XMS							
Facebook chat							
FaceTime							

Google Hangouts/Chat "off the record"							
Hushmail							
iMessage							
iPGMail							
Jitsi + Ostel							
Kik Messenger							
Mailvelope							
Mxit							
Off-The-Record Messaging for							

Mac (Adium)							
Off-The-Record Messaging for Windows (Pidgin)							
PGP for Mac (GPGTools)							
PGP for Windows Gpg4win							
QQ							
RetroShare							
Signal / RedPhone							
Silent Phone							
Silent Text							


Skype
SnapChat
StartMail
Subrosa
SureSpot
Telegram
Telegram (secret chats)
TextSecure
Threema

	✓	✓	✓		✗	✓	
Viber	✓	✗	✗	✗	✗	✗	✗
Virtru	✓	✗	✗	✗	✗	✓	✓
WhatsApp	✓	✗	✗	✗	✗	✗	✓
Wickr	✓	✓	✓	✓	✗	✗	✓
Yahoo! Messenger	✓	✗	✗	✗	✗	✗	✗

ABOUT

For years, privacy and security experts worldwide have called on the general public to adopt strong, open-source cryptography to protect our communications. The Snowden revelations have confirmed our worst fears: governments are spying on our digital lives, grabbing up communications transmitted in the clear.

Given widespread government surveillance, why don't

people routinely use tools to encrypt their communications? Wouldn't we all communicate a little more freely without the shadow of surveillance?

It boils down to two things: security and usability. Most of the tools that are easy for the general public to use don't rely on security best practices--including end-to-end encryption and open source code. Messaging tools that are really secure often aren't easy to use; everyday users may have trouble installing the technology, verifying its authenticity, setting up an account, or may accidentally use it in ways that expose their communications.

EFF, in collaboration with Julia Angwin at ProPublica and Joseph Bonneau at the Princeton Center for Information Technology Policy, are joining forces to launch a campaign for secure and usable crypto. We are championing technologies that are strongly secure and also simple to use.

The Secure Messaging Scorecard examines dozens of messaging technologies and rates each of them on a range of security best practices. Our campaign is focused on communication technologies -- including chat clients, text messaging apps, email applications, and video calling technologies. These are the tools everyday users need to communicate with friends, family members, and colleagues, and we need secure solutions for them.

We chose technologies that have a large user base--and thus a great deal of sensitive user communications--in addition to smaller companies that are pioneering advanced security practices. We're hoping our scorecard will serve as a race-to-the-top, spurring innovation around strong crypto for digital communications.

METHODOLOGY

Here are the criteria we looked at in assessing the security of various communication tools.

1. Is your communication encrypted in transit?
This criterion requires that all user communications are encrypted along all the links in the communication path. Note that we are not requiring encryption of data that is transmitted on a company network, though that is ideal. We do not require that metadata (such as user names or addresses) is encrypted.

2. Is your communication encrypted with a key the provider doesn't have access to?
This criterion requires that all user communications are end-to-end encrypted. This means the keys necessary to decrypt messages must be generated and stored at the endpoints (i.e. by users, not by servers). The keys should never leave endpoints except with explicit user action, such as to backup a key or synchronize keys between two devices. It is fine if users' public keys are exchanged using a centralized server.

3. Can you independently verify your correspondent's identity?
This criterion requires that a built-in method exists for users to verify the identity of correspondents they are speaking with and the integrity of the channel, even if the service provider or other third parties are compromised. Two acceptable solutions are:

- An interface for users to view the fingerprint (hash) of their correspondent's public keys as well as their own, which users can verify manually or out-of-band.
- A key exchange protocol with a short-authentication-string comparison, such as the Socialist Millionaire's protocol.

Other solutions are possible, but any solution must verify a binding between users and the cryptographic channel which has been set up. For the scorecard, we are simply requiring that a mechanism is implemented and not evaluating the usability and security of that mechanism.

4. Are past communications secure if your keys are stolen?

This criterion requires that the app provide forward

secrecy, that is, all communications must be encrypted with ephemeral keys which are routinely deleted (along with the random values used to derive them). It is imperative that these keys cannot be reconstructed after the fact by anybody even given access to both parties' long-term private keys, ensuring that if users choose to delete their local copies of correspondence, they are permanently deleted. Note that this criterion requires criterion 2, end-to-end encryption.

Note: For this phase of the campaign, we accept a hybrid forward-secrecy approach with forward secrecy on the transport layer (for example through TLS with a Diffie-Hellman cipher suite) and non-forward-secret end-to-end encryption, plus an explicit guarantee that ciphertexts are not logged by the provider. This is a compromise as it requires trusting the provider not to log ciphertexts, but we prefer this setup to having no forward secrecy at all.

5. Is the code open to independent review?

This criterion requires that sufficient source-code has been published that a compatible implementation can be independently compiled. Although it is preferable, we do not require the code to be released under any specific free/open source license. We only require that all code which could affect the communication and encryption performed by the client is available for review in order to detect bugs, back doors, and structural problems.

Note: when tools are provided by an operating system vendor, we only require code for the tool and not the entire OS. This is a compromise, but the task of securing OSes and updates to OSes is beyond the scope of this project.

6. Is the crypto design well-documented?

This criterion requires clear and detailed explanations of the cryptography used by the application. Preferably this should take the form of a white-paper written for review by an audience of professional cryptographers. This must provide answers to the following questions:

- Which algorithms and parameters (such as key sizes

or elliptic curve groups) are used in every step of the encryption and authentication process

- How keys are generated, stored, and exchanged between users
- The life-cycle of keys and the process for users to change or revoke their key
- A clear statement of the properties and protections the software aims to provide (implicitly, this tends to also provide a threat model, though it's good to have an explicit threat model too). This should also include a clear statement of scenarios in which the protocol is not secure.

7. Has there been an independent security audit?

This criterion requires an independent security review has been performed within the 12 months prior to evaluation. This review must cover both the design and the implementation of the app and must be performed by a named auditing party that is independent of the tool's main development team. Audits by an independent security team within a large organization are sufficient. Recognizing that unpublished audits can be valuable, we do not require that the results of the audit have been made public, only that a named party is willing to verify that the audit took place.

We've discussed this criterion in depth in a Deeplinks post: What Makes a Good Security Audit?

CHANGELOG

Entries in the Secure Messaging Scorecard were checked with the listed projects and companies when the Scorecard launched on 2014-11-06. Updates will be made if the listees or others inform us of changes or inaccuracies. A log of all such changes is below:

- **2015-06-12** :
  - We removed Secret as the service was shut down in May 2015.
- **2015-03-06** :
  - We credited QQ for completing an independent internal security audit (✓).
- **2015-02-17** :
  - We credited Telegram (both secret and regular mode) for undergoing a code audit in February 2015 (✓).
- **2015-01-29** :
  - We have credited QQ for encrypting messages in transit (✓). Though QQ does not use TLS/SSL, which is considered best practice, they have implemented a custom protocol for message encryption in transit.
  - We have clarified our scoring for forward secrecy (criterion #4).
- **2015-01-05** :
  - We have split the scoring for Telegram into two rows: baseline Telegram and Telegram secret chats. Baseline Telegram chats are not end-to-end encrypted so that the provider can't read them (✘). Telegram secret chats are, and in addition Telegram now supports perfect forward secrecy so that messages can be securely deleted (✓)
  - Wickr now provides the ability to verify contact's identities by exposing key fingerprints, which can be verified out-of-band or through in-band video. (✓)

- **2014-11-14** :
  - RIM has told us that BlackBerry Messenger Protected uses an out-of-band passphrase exchange and EC-SPEKE to perform identity verification. (✓) RIM has also told us that BBM Protected receives security reviews by an internal security team. (✓)
  - Viber has received a recent external security audit from EY Advanced Security Center. (✓)
  - Pidgin has documented a number of auditing processes including regular use of static analysis

tools and audits by a team at Cisco Talos. The Pidgin developers were clear that they do not know how thorough and complete these audits have been, but the audits nonetheless meet our criteria. (✓) Although auditing of Pidgin improves the security of the related Adium project (the projects share many components including libpurple, libotr, and libxml2), the Adium developers tell us that besides occasional static analysis by the developers themselves they are not aware of any independent auditing effort that addresses the Adium-specific code. So for the time being that project will not receive an audit checkmark.

- **2014-11-10** : Skype check mark for end-to-end encryption removed. (✘)
- **2014-11-04** : Snapchat app has audits from an internal security team. (✓)