

Using our Encrypted Messaging Service

This document will guide you through the process of using our online secured messaging tool.

- [About](#)
- [Security](#)
 - [Self Destruction](#)
 - [Single View](#)
 - [AES-256 Encrypted with HMAC-SHA512](#)
 - [Private Passphrase](#)
 - [Hashcash Protected](#)
 - [Anonymous Fingerprints](#)
- [How to use](#)

About

It is a platform for sending secure messages that self-destruct immediately after being read. Unread messages are automatically deleted after 30 days. As such, this application is suitable for sending private information such as PINs, usernames, passwords, and other sensitive information over insecure channels.

Security

All notes are encrypted on the server, and decrypted when the recipient views the note. The note can only be viewed once, and the note is securely destroyed on the server by overwriting the note with random data before deletion. Private passwords can be used to encrypt the note, instead of using the default server password. Private passwords are never stored on disk. In addition to private passwords, duress keys can be generated which will immediately destroy the note without decrypting it, in case you are being forced to provide the private password that decrypts the note.

Self Destruction

Every note submitted to the server will eventually be destroyed. All unread notes are destroyed after 30 days. All notes that have been viewed are destroyed immediately after viewing, and the browser will be redirected away from the note after 3 minutes. When the note is destroyed, it is overwritten with random data. This application does not make any assumptions about the underlying filesystem. Better safe than sorry, we say.

Single View

The note can be viewed only once. If someone attempts to view the note after the initial viewing, the note has been destroyed, so the client is redirected to a standard 404 error.

AES-256 Encrypted with HMAC-SHA512

Before storing the note on disk, every note is compressed with ZLIB, then encrypted with 256-bit AES in CTR mode using HMAC-SHA512 encryption. The AES-256 and HMAC-SHA512 keys are never stored on disk and are part of the URL.

Private Passphrase

The application uses a per note random key that is stored in the URL and is part of the encryption and decryption of the notes. The application supports using a custom passphrase to encrypt and decrypt the note. This custom passphrase is created using a password-based key derivation function (PBKDF2). The private passphrase is not stored to disk in the application, but this does not prevent the system administrator from modifying the code to do so. Therefore, it is best that you host this application yourself, on your own server.

Hashcash Protected

To prevent spammers from abusing the service, note submission is protected by the client solving a proof-of-work puzzle by minting a valid [Hashcash](#) token. As such, JavaScript is required to post a note. Hashcash tokens are stored on the server to prevent double-spending of the same token. Unfortunately, minting Hashcash tokens does not work well with mobile browsers.

Anonymous Fingerprints

Part of Hashcash tokens is the ability to create a uniquely identifiable resource string. Typically, this is an email address. However, with this application, your browser fingerprint is used as the resource string. The browser fingerprint is a 32-bit [MurmurHash number](#). By itself, it cannot identify who it belongs to. However, the system administrator could tie that fingerprint to an IP address and user agent string by looking in his web server logs.

How to use

- Browse to <https://onetime.msg.quatchi.org>
Self destructing encrypted notes

Securely send PINs, messages, passwords, or other sensitive or private information. All notes are encrypted, and destroyed immediately after viewing.

Type or paste your text below:

The screenshot shows the web interface for creating a self-destructing note. It features a large rectangular text input area. Below the input area, there are two buttons: 'Generate One' and 'Submit Note'. The 'Generate One' button is highlighted with a red box. Above the 'Generate One' button, there is a label 'Enter optional passphrase / password:' followed by a question mark icon. Below the 'Submit Note' button, there is a label '(Please be patient)' followed by a question mark icon.

- Type in the details requested into the large text box
Type or paste your text below:

The screenshot shows the same web interface as before, but the large text input area now contains the placeholder text 'Your secure message here'.

- Now secure your message by clicking "Generate One" for randomly generating a passphrase

The screenshot shows the 'Generate One' button highlighted with a red box. Below it, the generated passphrase 'fbE1oHuSFbJPH0dDV_1jDb' is displayed. Above the passphrase, there is a label 'Enter optional passphrase / password:' followed by a question mark icon.

- You are now ready to submit your message, click "Submit Note"

The screenshot shows the 'Submit Note' button highlighted with a red box. Below it, the text '(Please be patient)' is displayed, followed by a question mark icon.

- Once the next screen is loaded copy the following URL and Passphrase and send them to us (support@aim4cloud.com):

- "Your private url"
- "Your passphrase"

Your Secret Note Details

If you submitted this note in error, you can [destroy the note here](#).

Your private url: https://onetime.msg.quatchi.org/dnote/EAGnij4Wcm2oPd2Uki_ZFA

Your private ID: EAGnij4Wcm2oPd2Uki_ZFA ?

Your passphrase: fbE1oHuSFbJPH0dDV_1jDb

Your duress key: 68Q1mAR35rP6tinwIWGERQ ?

Alternatively, you can scan this QR code, and "share" it using your mobile device via text message, email, etc. **NOTE:** [ZXing Barcode](#) users: disable "Retrieve more info" from the settings, otherwise your recipient will **NOT** be able to view the post.

[Show QR Code](#)

- You **must** send us the above details or we wont be able to access your note, the system does not send any notification to ensure the security of your message.