Journal of Cybersecurity and Privacy

MDPI

Article

Exploiting Online Services to Enable Anonymous and Confidential Messaging

Pedro Sousa [1,*,†], António Pinto [2,3,*,†] and Pedro Pinto [1,3,4,†]

[1] Instituto Politécnico de Viana do Castelo, 4900-347 Viana do Castelo, Portugal
[2] Centro de Inovação e Investigação em Ciências Empresariais e Sistemas de Informação, Escola Superior de Tecnologia e Gestão, Politécnico do Porto, 4610-156 Porto, Portugal
[3] Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência, 4200-465 Porto, Portugal
[4] Instituto Universidade da Maia, 4475-690 Maia, Portugal
* Correspondence: pguilhermesousa@ipvc.pt (P.S.); apinto@inesctec.pt (A.P.)
† These authors contributed equally to this work.

Abstract: Messaging services are usually provided within social network platforms and allow these platforms to collect additional information about users, such as what time, for how long, with whom, and where a user communicates. This information allows the identification of users and is available to the messaging service provider even when communication is encrypted end-to-end. Thus, a gap still exists for alternative messaging services that enable anonymous and confidential communication and that are independent of a specific online service. Online services can still be used to support this messaging service, but in a way that enables users to communicate anonymously and without the knowledge and scrutiny of the online services. In this paper, we propose messaging using steganography and online services to support anonymous and confidential communication. In the proposed messaging service, only the sender and the receiver are aware of the existence of the exchanged data, even if the online services used or other third parties have access to the exchanged secret data containers. This work reviews the viability of using existing online services to support the proposed messaging service. Moreover, a proof-of-concept of the proposed message service is implemented and tested using two online services acting as proxies in the exchange of encrypted information disguised within images and links to those images. The obtained results confirm the viability of such a messaging service.

Keywords: covert; anonymous; communication

Citation: Sousa, P.; Pinto, A.; Pinto, P. Exploiting Online Services to Enable Anonymous and Confidential Messaging. *J. Cybersecur. Priv.* **2022**, *2*, 700–713. https://doi.org/10.3390/jcp2030035

Academic Editor: Aniello Castiglione

Received: 27 July 2022
Accepted: 26 August 2022
Published: 31 August 2022

1. Introduction

Messaging services are commonly provided by social networks and other online services to allow users to exchange messages and other types of data between them. These services enable real-time data exchange between family, friends, or colleagues, even in times when physical distancing is necessary, such as during the COVID-19 pandemic [1].

However, when using these messaging services, users, their activities, and their messages are subject to being tracked, analyzed, scrutinized, and profiled. Cases of misuse of personal information by social network companies, such as the Facebook–Cambridge Analytica data scandal [2], have also brought to light some unlawful uses of the personal information gathered and processed by these platforms. Privacy has gained momentum with the implementation of the General Data Protection Regulation (GDPR) [3] by the European Union; however, users of these services still may not be fully aware of the extent to which information is being collected about them and the purposes for which it is being processed.

Currently, some messaging services adopt end-to-end encryption [4] to enable confidential communication between users, and with this, regain some of the users' trust in their services. However, even when using end-to-end encryption, it may be possible to identify the communicating parties by analysing their web traffic [5]. Some other services,

such as Ricochet Refresh [6], a peer-to-peer messenger application, offer anonymous and decentralised communication via The Onion Router (TOR). This achieves the main objectives behind this paper (anonymous and untraceable messaging); however, its usage of TOR is also its main drawback, as it makes the service vulnerable to censorship in certain countries [7,8].

The authors of [9] argue that communication services should be designed to enable parties to communicate in a way so that no one knows who said something. Extending this argument, it is important to design a communication system between parties in which the participants are the only ones who know the content of the messages, the existence of the messages, and the identity of the involved parties, hiding this information even from the platforms and servers being used for communication.

Cryptography techniques have been proposed to enable confidentiality, integrity, non-repudiation, authentication, etc. [10]. The use of steganography [11–14] for hiding secret messages in pictures can be used to reinforce anonymity [15]. One of the most known and more commonly used image steganography algorithms is Least-Significant-Bit (LSB) [16]. The combination of cryptography and steganography in a communication system would enforce confidentiality and would also assure that only the sender and the receiver are aware that a secret message exchange exists. Sharing secret information within images through online services that do not require authentication, such as image-sharing services or social network platforms, can also provide anonymity to users.

In the work herein, the authors propose a messaging service that uses steganography and online services, including social networks, to support anonymous and confidential communication. The reference scenario is presented in Figure 1. In this scenario, a given user, the Sender, wants to share a message with the Receiver. The Sender uses an application that encrypts the message, hides it in an image using steganography, and uploads the image to an online service. This online service generates a Uniform Resource Locator (URL) link for sharing the image, which the user uploads to another online service. Later, the Receiver obtains the shared URL, locates and downloads the image, and decrypts the message. In this scenario, two online services that do not require user authentication are used simultaneously: one for image sharing and another for sharing the links to these images between users.

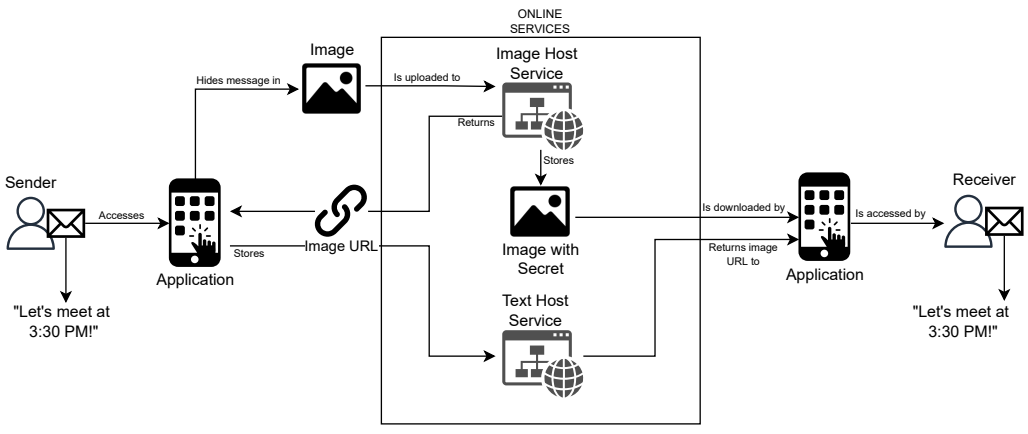


Figure 1. Reference scenario.

The main contributions of this work are:

- Evaluate existing online services and social network platforms to check the feasibility of exchanging images containing secret messages created using readily available and free-to-use steganography programs.
- Propose and implement a Proof of Concept (PoC) of a messaging service that exploits online services for secret messaging, testing its viability.

This paper is structured as follows. Section 2 presents related work. Section 3 provides an assessment of online services to be used by the proposed messaging service. Section 4

presents the details on implementing the proposed messaging service. Section 5 presents a discussion focused on security-related aspects of the proposed messaging service. Section 6 presents the final conclusions and points towards future work.

2. Related Work

The presented related work focuses on proposals that intend to exploit online services by combining the use of steganography techniques and tools to enable concealed communication between users without enabling server access to the exchanged messages.

In [17], a comparison between various messaging applications' security levels is presented. The paper goes into detail as to what security flaws or strengths each application provides and an overall result on whether each application should be considered trustworthy to manage a user's private data. Among the presented applications, this paper focuses only on the ones recommended: Signal [18], Threema [19], Wire [20], and Session [21]. The solution proposed in this paper is unique and has advantages over all these messaging applications.

Although Signal does not present major security flaws, it requires new users to offer their mobile phone number to register a new account (and also for recovery purposes). This information, sent to parent companies, is a breach of anonymity, linking a user's cyberspace presence to their real personage. Furthermore, the requirement to give a phone number relies on the user having a mobile phone.

Wire also has this issue of giving parent or third-party companies information, more specifically, e-mail and/or a phone number, which is required to register a new account to start using the service. Once again, this is a breach of anonymity.

As presented, both of these applications fail the anonymity check—it is impossible to use these applications without some link to the user's real persona.

Threema does not have registration as a mandatory requirement. It gives the user the choice to insert a phone number for recovery purposes, but this is optional. That said, their application API and server code are not open-source. Furthermore, it is a paid application.

Session makes use of blockchain technology to provide its own onion routing service. When a user sends a message, it is carried through three different nodes on the network before passing through a listening node and once again hopping through three different nodes. Since each node only has information regarding its adjacent nodes (i.e., the first node to receive the message only knows the IP of the sender and the second node—not the recipient's IP), privacy is ensured. The application relies, for the most part, on local storage, with the exception being temporary storage of messages in multiple nodes, designated a swarm, which get deleted after the messages Time To Live (TTL) are surpassed if the recipient is offline. However, this network requires nodes. In order for a node to be authorized and accepted to the system, a stake must be made, requiring the owner of the node to lock a certain monetary value to the node. Afterwards, the person responsible receives a reward for their node's usage. On the one hand, this makes it quite costly for anyone attempting to attack the network; however, the reward system may also create a conflict of interests between maintaining privacy vs. abusing this system.

The proposed solution is to be built as cross-platform, so it does not rely on a mobile phone for proper usage. It is currently planned to be fully open-source and free. No registration is required—each account has a unique ID, and a new account (or ID) can be created at any time to replace the older one (along with the old ID's data). All data are transferred through images that are uploaded anonymously and hosted on various image-hosting services. These images are only available through their ID and are deleted after a certain amount of time. Although an attacker may see that an image is being uploaded, the solution relies on the inconspicuousness of the scenario to avoid suspicion.

Table 1 presents an overview of the security level of each messaging application already available and the proposed solution.

Table 1. Messaging applications and proposed solution security comparison.

	Signal	Threema	Wire	Session	Proposed Solution
Provides Anonymity?	**No ***	**No ***	**Yes**	**Yes**	**Yes**
Data gathered	Contact Info	Contact Info, Diagnostics	Contact Info, Diagnostics, Usage Data	None	None
Supports self-destructing messages?	Yes	No	Yes	Yes	Yes
Can be used without a mobile phone?	No	No	Yes	Yes	Yes
Is a free application?	Yes	No	Yes	Yes	Yes
Is Open source?	Yes	No	Yes	Yes	Yes
Is resistant to conflicts of interest?	Yes	No	Yes	No	Yes
Requires cryptocurrency?	No	No	No	Yes	No

* A phone number required to register a new account and data are processed by parent or third-party companies.

A set of research works proposed the use of steganography to provide messaging services. The authors of [22] argued that steganography could be used for message exchange between users in a way that only the sender and receiver were able to decrypt the message and were the only ones aware that messages were being exchanged. They also considered its use on mobile phones, which led them to compare the performance of existing steganography solutions. They concluded that at the time, their method was feasible but highly influenced, in terms of execution times, by the image size.

In [23], the authors proposed a secure form of communication using steganography to hide the information exchanged between users. Their solution aimed at using existing online services such as social networks and photo-sharing services, which enabled them to identify and compare the processing operations performed by multiple online services. One of their conclusions was that these processing operations (compression, resizing, and metadata and file name changes) hamper the use of traditional image-based steganography; thus, they proposed the use of filename-based and tag-based steganography. Despite supporting message exchanges between users without servers being aware of them, their proposal did not encrypt the hidden messages and presented significant size constraints, as only 4 to 7 bytes of data could be stored in each filename.

The authors of [24] studied the possibility of using Facebook to upload and share images that included messages hidden using steganography. One of their findings was that image compression for Facebook to save storage space and bandwidth generally disrupted steganography. They tested multiple steganography programs and, for the most part, had little success in obtaining the information after the image was uploaded. Partial positive results were only obtained with preprocessing and multiple attempts.

In [25], the authors researched ways to bypass steganography disruption due to modification of the submitted images. Their solution first creates a compressed version of the image and then embeds the secret data into the compressed image, resulting in a Stego-image. Next, an intermediate image is created using adjustments based on the Stego-image and the original image. The resulting image can then be uploaded to a social network, and their compression does not impact the extraction of the secret data.

More recently, Lu et al. [26] presented another solution using steganography algorithms that are resistant to JPEG compression. In this work, the original image was first run through an auto-encoder, which also inserts the secret message, in an attempt to create an intermediate image which, by the predictions of the program, will generate the target image after image compression.

Table 2 presents the characteristics of the aforementioned research works. From the analysis of these works, it can be concluded that the topic, despite not being new, is still under active research. Further, none of these solutions allows user anonymity, as all require user authentication before using the service. To the best of our knowledge, there is no solution that allows anonymous and confidential messaging between users without user authentication or dedicated servers. The design of a novel proposal should not compromise performance when using different image sizes, all messages processed and hidden in

images should be encrypted beforehand, and the image compression imposed by using services should be circumvented to maintain the original images.

Table 2. Previous work and proposed solution features overview.

	[22]	[23]	[24]	[25]	[26]	**Proposed Solution**
Covert Messages	Yes	Yes	Yes	Yes	Yes	Yes
Anonymity	N/A	No	No	No	N/A	Yes
Message Encryption	No	No	No	No	No	Symmetric encryption
Resource Requirements	Low	Low	Low	High	High	Low
Cross-Platform	No	No	No	No	No	Yes

3. Assessment of Online Content Sharing Services

Steganography and online content sharing services are central parts of the presented reference scenario. The proposed message service uses steganography to hide secret messages in an image, and online services to share the image, along with a link for the image. In particular, we assessed current steganography applications and online content-sharing services.

Concerning steganography applications, we assessed the latest versions of the following: OpenPuff [27], OpenStego [28], StegHide [29], and StegoShare [30]. These applications were chosen because of their availability as free versions and their open source code.

OpenPuff supports multiple formats and provides multiple layers of protection, including encryption of secret data, scrambling of data to make it more difficult to know where the data begins/ends, whitening to mix the scrambled data with noise, and finally, encoding the whitened data through a non-linear function. OpenStego allows data to be encrypted and then hidden in cover files, as well as watermarking files in order to identify ownership. StegHide allows data to be encrypted and hidden in JPEG and BMP formats and also in some audio formats. StegoShare can also encrypt and hide data in images.

Current online services are known to change image/photo characteristics after upload, modifying compression, resolution, metadata, and file name, and may impact steganography. To assess this impact, the following online services were selected: Facebook (https://facebook.com (accessed on 30 September 2021)), Twitter (https://twitter.com (accessed on 30 September 2021)), LinkedIn (https://linkedin.com (accessed on 30 September 2021)), Imgur (https://imgur.com (accessed on 1 October 2021)), Flickr (https://flickr.com (accessed on 1 October 2021)) and ImgBox (https://imgbox.com (accessed on 1 October 2021)). Facebook, Twitter, and LinkedIn were selected given their popularity among social networks, while the remainder were selected as social networks focused on image hosting.

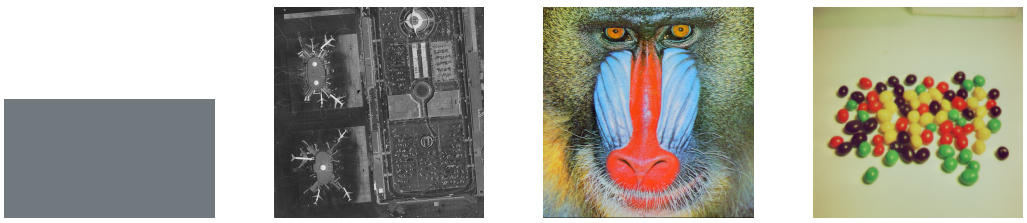
The assessment included two tests. The first test was designed to check the changes introduced by the selected online services. This test sent a random image with a resolution of 3840 $\times$ 2160 pixels to a set of steganography applications (OpenPuff, OpenStego, StegHide, and StegoShare) to generate an input stego image. Each input stego image was then uploaded to the selected online services to determine the changes introduced in the output image regarding image size, resolution, and format.

Table 3 presents the results of the first test, conducted in September 2021, using the selected steganography applications and online services. The results show that Facebook converts images to JPEG format with a maximum resolution of 2048 $\times$ 1152 pixels using a high compression rate. Flickr does not compress the uploaded images nor change their resolution. ImgBox maintains image resolution, but the resulting file size indicates that a low-compression algorithm is used.

Table 3. Results of the first test.

		Input	Output Image					
		Stego Image	**Facebook**	**Linkedin**	**Twitter**	**Imgur**	**Flickr**	**ImgBox**
OpenPuff	Size	6.72 MB	312 KB	186 KB	546 KB	467 KB	6.72 MB	6.54 MB
	Res.	3840 × 2160	2048 × 1152	2048 × 1152	3840 × 2160	3840 × 2160	3840 × 2160	3840 × 2160
	Format	PNG	JPG	JPG	JPG	JPG	PNG	PNG
OpenStego	Size	7.8 MB	312 KB	186 KB	545 KB	467 KB	7.8 MB	4.49 MB
	Res.	3840 × 2160	2048 × 1152	2048 × 1152	3840 × 2160	3840 × 2160	3840 × 2160	3840 × 2160
	Format	PNG	JPG	JPG	JPG	JPG	PNG	PNG
StegHide	Size	791 KB	277 KB	187 KB	770 KB	749 KB	791 KB	877 KB
	Res.	3840 × 2160	1920 × 1080	2048 × 1152	3840 × 2160	3840 × 2160	3840 × 2160	3840 × 2160
	Format	JPG	JPG	JPG	JPG	JPG	JPG	JPG
StegoShare	Size	19.2 MB	339 KB	190 KB	647 KB	467 KB	19.2 MB	N/A
	Res.	3840 × 2160	2048 × 1152	2048 × 1152	3840 × 2160	3840 × 2160	3840 × 2160	N/A
	Format	PNG	JPG	JPG	JPG	JPG	PNG	N/A

The second test was designed to (1) determine if the compression algorithms used by the different services impacted the steganography; (2) assess if the systems maintain their behaviour when presented with images with different characteristics, such as lower resolutions or monochrome images; and (3) assess if the systems work when hiding bigger text messages. Thus, this test consisted of two specific text messages of different lengths encrypted with AES256 and hidden by a steganography application in four base test images of different colours and sizes. The first text message was "The quick brown fox jumps over the lazy dog", and the second is a random 256-character message. The adopted base test images for the second test are presented in Figure 2 and consisted of a solid grey image with a resolution of 1920 × 1080 pixels plus three other images from Volume 3 of the database of standard test images of the University of Southern California [31]. In particular, images "4.1.08—Jellybeans" (with a resolution of 256 × 256 pixels), "4.2.03—Baboon" (with a resolution of 512 × 512 pixels), and "5.3.02—Airport" (with a resolution of 1024 × 1024 pixels) were selected due to their different resolutions and the need to have both coloured and grayscale images. All images were then converted to both PNG and JPEG formats due to the fact that some steganography programs only support one of these formats. To check the differences between the images before and after the upload to the online services, secure hashes (MD5 and SHA) were obtained and compared.



(**a**) (**b**) (**c**) (**d**)

Figure 2. Adopted base images: (**a**) solid colour; (**b**) airport; (**c**) baboon; (**d**) jellybeans.

The second test was conducted during the months of September and October of 2021, and a total of 192 image uploads was processed and analysed (6 online services, 4 steganography applications, and 8 images—the 4 selected images with the first text message and the same 4 with the second one). Table 4 shows the percentage of secret messages that were recovered after the respective image had been uploaded to and downloaded from an online service. The results show that when using Facebook, LinkedIn, and Twitter, no messages were recovered. The other services had success rates of 0%, 87.5%, and 100%.

The Facebook, Twitter, and LinkedIn compression algorithms do not appear to be the same, since the hash values of all images were different. Further, all images, regardless of resolution, file size or format, were converted into JPG format, and the resulting file size was smaller. Images having a resolution of 3840 × 2160 pixels were reduced to 2048 × 1152 pixels on Facebook and LinkedIn. The metadata of the image was also changed. These social network platforms use image compression, directly hampering steganography. Similar behaviour was assumed to exist in all social network platforms.

Table 4. Percentage of successfully retrieved messages.

	Facebook	LinkedIn	Twitter	Imgur	Flickr	ImgBox
OpenPuff	0%	0%	0%	100%	100%	87.5%
OpenStego	0%	0%	0%	100%	100%	100%
StegHide	0%	0%	0%	100%	100%	0%
StegoShare	0%	0%	0%	100%	100%	100%

Imgur had a 100% success rate when using images with a resolution of 1920 × 1080 pixels or less. Images with higher resolutions or in PNG format were converted to JPEG and compressed in such a way that their size became too small to retain the secret message. However, if the original image was already in JPEG format, the image file changes did not disrupt the steganography. Further, the secure hash results of the retrieved images were different from those of the uploaded pictures. After additional analysis, we concluded that Imgur stripped metadata from all images, resulting in different hash values for the same images. These conclusions were drawn by extracting the core data of the images and then comparing the secure hash values of these.

Flickr had a 100% success rate due to the fact that Flickr allows the user to download the original images without any changes. Even image metadata and resolutions are preserved. The secure hash values of the downloaded images confirmed that no modifications were made to the uploaded images.

ImgBox was able to retrieve information for all steganography programs except StegHide. In our testing, StegHide processed only JPEG images, which, in this case, underwent enough changes to disrupt steganography-based message retrieval.

Table 5 summarizes the comparison of the tested online services with regard to file modification, image characteristic modification, and the capability of retrieving the hidden text from the downloaded images. Imgur alters the image files but does not alter the image data, as proven by different hash values of the files and equal hash values of the data section of the image. Imgur also enables the retrieval of hidden text unless the image resolution exceeds 1920 × 1080 pixels.

Table 5. Online service comparison.

	Facebook/Linkedin/Twitter	ImgBox	Imgur	Flickr
Hash	Changed	Changed	Changed	Unchanged
Image	Changed	Changed	Partially Changed	Unchanged
Message	Unretrievable	Partially Retrievable	Partially Retrievable	Retrievable

As a remark, Flickr and Imgur appear to adequately demonstrate the viability of the use of steganography. Image resolution should be 1920 × 1080 pixels or lower. The PNG format offers more space to hide information but is also more likely to suffer compression and format change. The smallest image size used, 256 × 256 pixels, was still able to hide a 256-character message.

In particular, Imgur allows images to be uploaded anonymously, and images are also not listed on its website, so only users that have the URL or know the ID of an image can access it. Such behaviour, despite favouring anonymous operations, raised one problem: How would the receiver know the location of and get access to the image? The users should avoid at all costs sending the image links to each other, as this would lead to their identification. On the one hand, a specific service could be developed to exchange the image links. On the other hand, searching and evaluating existing online services is also an option. Multiple possibilities were identified, such as Advanced DontPad [32], Dontfile [33], Dontpad++ [34], and Dontpad [35]. Of these, only the latter operates solely in HyperText Markup Language (HTML), while the remaining ones require the execution of JavaScript (JS) in their text area. The use of JS was considered a limitation since it requires a client application to first render the page before being able to work with the data in it, increasing the use of resources and its runtime. Being an online text-file editor, DontPad is

a lightweight service and does not require user authentication or previous setup; thus, it appears to be the most-adequate online service for the current context.

4. Implementing the Proposed Messaging Service

The proposed messaging service should enable end-to-end, anonymous, confidential, and covert message exchange between users. End-to-end means messages are encrypted at the sender and only decrypted at the destination. Anonymous means only the users that are exchanging messages know their identity. Confidential refers to exchanged messages being encrypted end-to-end. Covert means the exchange of messages remains hidden. Therefore, the solution should not require dedicated servers and should use existing online services while disguising its use. Moreover, in order to better promote user anonymity, the proposed experiment should only use online services that do not require user authentication.

Multiple online services were analysed in order to verify their adequacy for the identified requirements. Our search for an adequate online image-sharing website led to the selection of Imgur because it maintains image format (except for the case of images with resolutions above 1920 $\times$ 1080 pixels), but mainly because it also allows images to be uploaded anonymously. As an alternative channel to exchange image URLs, the DontPad online service was selected since it was the only one found that did not require JavaScript, and because it allows anonymous use. JavaScript was considered a potential avenue for user identification, motivating its dismissal.

When two users want to exchange messages, the proposed messaging service requires that they first exchange a channel identifier and agree on a password and salt to use with a Key Derivation Function (KDF) [36,37] to create a channel key (C_i) shared by all users in the channel. A secure procedure to exchange the identifier, key, and salt is assumed. This exchange can also occur offline, preferably in person for the sake of online anonymity. These values are generated per conversation using a secure random number generator. The proposed solution adopts the Universally Unique Identifier (UUID) format for the channel identifier. The URL used to store the links to the images with hidden information is obtained by concatenating the generated UUID with the text-sharing service URL. For instance, assuming Channel 1 has the UUID_1, then the group's shared online folder in DontPad will be available at: http://dontpad.com/UUID_1. This would then be used to pass the image's URL, with the URL being deleted as soon as the receiver reads the secret message.

Afterwards, users can exchange messages. A sequence diagram is depicted in Figure 3. The sender starts by securely deriving a channel key (C_i) from the password and the salt (Step 1). In Step 2, an initialization vector IV_i is generated with a secure random generator. Afterwards, in Step 3, the key C_i and the IV_i are used to encrypt the plaintext message using symmetric encryption (e.g., AES), generating the ciphertext E_i. In Step 4, the sender generates a Hash-Based Message Authentication Code (HMAC) [38–41] H_i from the ciphertext E_i and IV_i. The ciphertext E_i, HMAC H_i, and a nonce are encrypted with the key C_i and then concatenated with IV_i in Step 5—these data shall be referred to as M_i. In Step 6, the secret data M_i is hidden through steganography, being embedded into an image file F_i, which has had its EXIF data scrubbed clean, resulting in the stego-image SF_i. The sender then, anonymously, uploads SF_i to the image-hosting service (Step 7), obtains its URL (Step 8), and writes the URL to a folder on the text-hosting service (step 9). Next, the receiver can read the shared folder on the text-hosting service, obtain the new URL (Step 10), and proceed to download SF_i (Step 11). Then, he/she extracts the embedded data M_i (Step 12) to retrieve IV_i and the encrypted E_i and H_i (Step 13). In Step 14, E_i and H_i are retrieved after decryption, and afterwards, a new HMAC H_{igen} is generated from the retrieved E_i and IV_i (Step 15). H_i and H_{igen} are compared to ensure the integrity of the message (Step 16). Finally, in Step 17, the plaintext message is retrieved by decrypting E_i with the key C_i and the initialization vector IV_i.

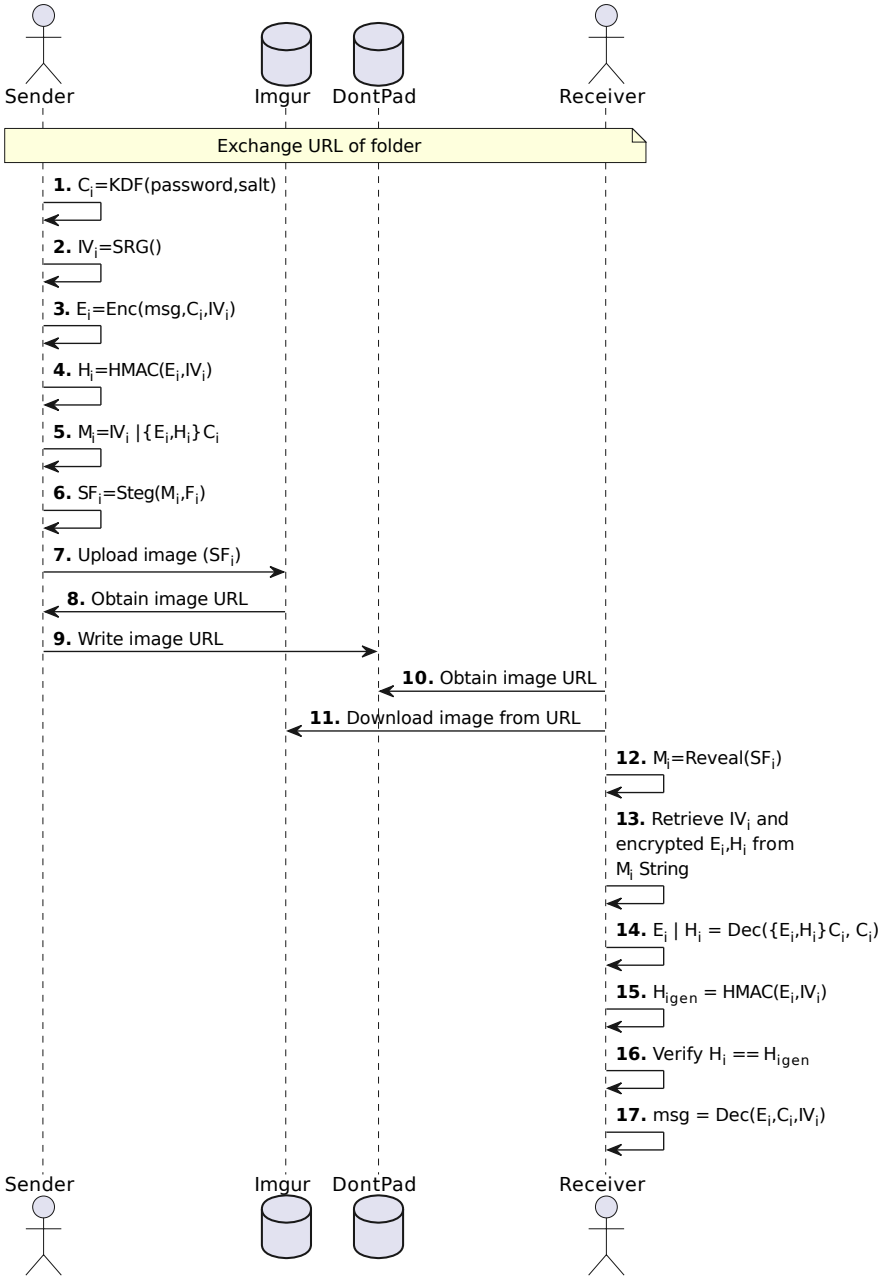


Figure 3. Sequence diagram of the proposed message service.

In order to demonstrate messaging service feasibility and evaluate its performance, a PoC implementing the procedures presented in Figure 3 was developed and tested. The tests were executed between 27 December 2021 and 9 January 2022, and each test ran three simultaneous instances per execution, each one uploading and retrieving secret messages through one of three images of different sizes (small, medium, and large). The smallest image, "jellybeans", had a size of 83.4 kB; the medium size image, "baboon" had a size of 611 kB; and the larger image, "airport", had a size of 703 kB. The secret message embedded in the images had a size of 6 bytes. Elapsed times for each execution were recorded.

In order to obtain the duration of the different steps that compose a full execution, the following five steps were considered and their execution durations measured. The first step comprises the time taken to encrypt the original message and embed it into an image. The second step comprises the time taken to upload the image to Imgur and to obtain the image URL within the Imgur website. The third step comprises the time taken to store the

image URL in a specific area on DontPad by the sender plus URL download from DontPad by the receiver. The fourth step comprises the time taken by the receiver to download the image file from Imgur. Lastly, the fifth step comprises the time taken by the receiver to extract and reveal the secret message.

The chosen KDF was Argon2 [42,43], which was used to create hashes with a length of 32 characters. The hashes were used as a key for AES-256-CBC encryption, with CBC mode chosen due to its lesser resource requirements. An Initialization Vector (IV) with a length of 32 characters was generated for encryption. An HMAC-SHA256 of the encrypted message was also generated for authentication and integrity purposes.

Figure 4 presents the PoC runtimes with standard deviations (in milliseconds) of each step per image size normalized to 100%. For all image sizes, upload takes the most time (between 60% and 70% of the total time elapsed), followed by download (around 20% of the total time elapsed). Shorter durations were obtained for small images.

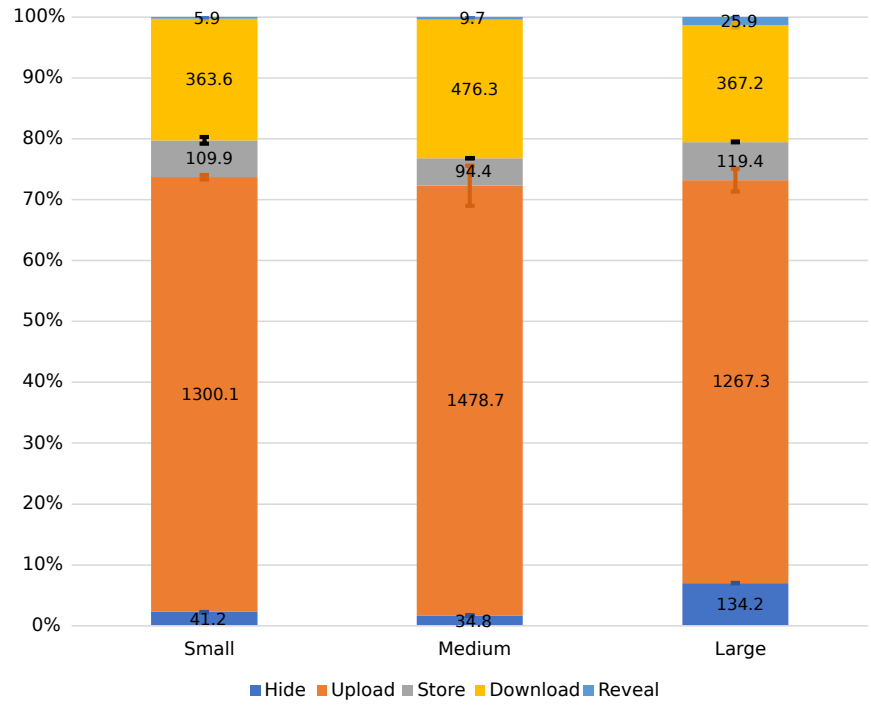


Figure 4. Runtimes (in milliseconds) per step and image size, normalized to 100%.

5. Discussion

The objective of the proposed messaging service is to enable covert communication guaranteeing the confidentiality, integrity, and authentication of the exchanged messages while using images as containers for the message exchange. Considering that the envisioned concept makes use of online services, the Dolev–Yao intruder model [44] was considered adequate for this security analysis. The Dolev–Yao intruder model states that the intruder has full control over the network, being capable of reading, altering, or deleting data in transit.

In the designed PoC, the KDF, Secure Random Generator (SRG) [45–47], and HMAC are assumed to be secure, and the channel's password and salt are assumed to be previously exchanged in a secure manner, preferably in-person, and generated using a SRG. Moreover, it is also assumed that the UUID is securely generated and that the used images are either chosen randomly from online galleries or from the user's gallery accordingly to user preference.

Following these assumptions, a discussion can be made about user anonymization, message confidentiality, message integrity, and system availability.

User anonymity is assured with the usage of generally available online services that accept unauthenticated usage, combined with the inability to distinguish between users that take part in a channel. Only users having the password and salt of a channel can read messages exchanged in that channel, and it is also assumed that user equipment is secure in a way that the used services do not collude or pursue user reidentification based on their access patterns or IP address (such as a non-rooted phone). Aside from using multiple services, users can also rely on Virtual Private Network (VPN) [48–52] services or a custom-built service on the TOR [53] network to enhance user anonymity. Furthermore, the Exchangeable Image File Format (EXIF) data is removed from all images used so that no location data, for instance, is left.

Message confidentiality is assured with the use of secure functions for KDF, SRG, and encryption. The channel key (C_i) is derived using a KDF such as Argon2 [42,43]. A unique, random, and securely generated IV (IV_i) is also used for each message.

Integrity is guaranteed by calculating a hash value (H_i) for each message using a secure HMAC function that receives the encrypted message (E_i) and the current IV (IV_i) as parameters. The hash value will always be unique due to the freshness of the IV_i, even if the message (*msg*) is the same. The *msg* and H_i are only exchanged while encrypted, making them unable to be changed outside of the channel without detection.

The availability of the proposed PoC is considered to be partially assured since the user is reliant on multiple, externally controlled online services. The PoC is based on currently existing technology and services; however, an advanced user may create his/her own image-hosting service as an alternative in case the available hosting services change operation procedures or experience a shutdown. Further, while an attacker may not be able to easily delete the image from the chosen image-hosting service, he/she can, however, delete the URL from the text-hosting service if he/she knows the used filename (UUID_i). Because the used online text-sharing service does support Transport Layer Security (TLS), the use of a VPN connection is assumed. If a URL is deleted from the text-hosting service prior to being read by the recipient, the recipient will not even realize that a new message had been sent. We argue that the risk of such behaviour is outweighed by the eventual loss of privacy associated with numbering messages and keeping a record of which ones are read by whom. The proposed PoC also allows protection against specific attacks such as replay attacks, Chosen-Ciphertext Attacks (CCAs) [54–58], and Chosen-Plaintext Attacks (CPAs) [54,55,59–61]. Although an attacker without access to the channels' *password* and *salt* is unable to insert new messages in a channel, the Dolev–Yao intruder model assumes that previous messages can be sent, performing a replay attack. The proposed PoC adopts the per-message use of a securely generated random value that can only be used once (*nonce*). Each user stores previously exchanged *nonce* values, rejecting messages for which this value is repeated. Moreover, the *nonce* is encrypted when exchanged, and, thus, it is assumed that it is impracticable for the attacker to alter the *nonce*, making replay attacks infeasible. In a CCA, the attacker must be able to request the decryption of ciphertext of their own choosing. In a CPA, the attacker must obtain the encryption of plaintexts of their own choosing. The only way to generate a valid ciphertext, or to decrypt it, is by having the correct *password* and *salt*, which are assumed to have been securely exchanged between users. Moreover, the used KDF and SRG are assumed secure, plus the adoption of a fresh IV for each per message ensures that these attacks do not break the system.

6. Conclusions and Future Work

Currently, mainstream messaging services are provided within social network platforms, and these platforms are able to collect information from users, even when these platforms use end-to-end encryption.

The work presented herein proposes a message service to support anonymous and confidential communication without requiring dedicated online servers but exploiting existing ones, including social networks. The current work reviewed the viability of using existing online services to support the proposed messaging service. Steganography was

chosen as the technique to enable covert communication, and existing online services were assessed in terms of their support for the use of steganography. Despite most social networks disrupting steganography through image compression, a selected list of online services can be used. The message service was implemented as a PoC to confirm its feasibility.

In future work, other online services could be analysed to create a list of similarly operating services for photo and text sharing between users. Such a list would increase the levels of anonymity and resilience. Further, rolling filenames in the text-hosting service through seeds may be explored, since this would make it more difficult for third parties to find the filenames being used in the text-sharing services. The development of a fully featured mobile application that operates as described can be also envisioned.

Author Contributions: Conceptualization, P.S., A.P., and P.P.; methodology, P.S., A.P., and P.P.; software, P.S.; validation, P.S., A.P., and P.P.; investigation P.S.; writing—original draft preparation, P.S.; writing—review and editing, P.S., A.P., and P.P.; supervision, A.P., and P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Norte Portugal Regional Operational Programme (NORTE 2022) under the PORTUGAL 2022 Partnership Agreement through the European Regional Development Fund (ERDF) within project "CybersSeCIP" (NORTE-01-0145-FEDER000044).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No other data available from external sources.

Acknowledgments: This study was developed in the context of the Master in Cybersecurity Program at the Instituto Politécnico de Viana do Castelo, Portugal.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

GDPR	General Data Protection Regulation
LSB	Least-Significant Bit
URL	Uniform Resource Locator
JS	JavaScript
HTML	HyperText Markup Language
UUID	Universally Unique Identifier
HMAC	Hash-Based Message Authentication Code
KDF	Key Derivation Function
SRG	Secure Random Generator
TOR	The Onion Router
VPN	Virtual Private Network
CPA	Chosen Plaintext Attack
CCA	Chosen Ciphertext Attack
IV	Initialization Vector
PoC	Proof of Concept
EXIF	Exchangeable Image File Format
TLS	Transport Layer Security
PFS	Perfect Forward Secrecy
TTL	Time To Live

References

1. Gong, W.J.; Wong, B.Y.M.; Ho, S.Y.; Lai, A.Y.K.; Zhao, S.Z.; Wang, M.P.; Lam, T.H. Family E-Chat Group Use Was Associated with Family Wellbeing and Personal Happiness in Hong Kong Adults amidst the COVID-19 Pandemic. *Int. J. Environ. Res. Public Health* **2021**, *18*, 9139. [CrossRef] [PubMed]
2. Confessore, N. Cambridge Analytica and Facebook: The Scandal and the Fallout So Far. 2018. Available online: https://www.nytimes.com/2018/04/04/us/politics/cambridge-analytica-scandal-fallout.html (accessed on 13 January 2022).
3. European Parliament and Council Regulation (EC) No 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). *Off. J. Eur. Union* **2016**, *119*, 1–88.
4. Ermoshina, K.; Musiani, F.; Halpin, H. End-to-End Encrypted Messaging Protocols: An Overview. In Proceedings of the Third International Conference, INSCI 2016—Internet Science, Florence, Italy, 12–14 September 2016; Lecture Notes in Computer Science (LNCS); Bagnoli, F., Satsiou, A., Stavrakakis, I., Nesi, P., Pacini, G., Welp, Y., Tiropanis, T., DiFranzo, D., Eds.; Springer: Florence, Italy, 2016; Volume 9934, pp. 244–254. [CrossRef]
5. Herrmann, D.; Gerber, C.; Banse, C.; Federrath, H. Analyzing characteristic host access patterns for re-identification of web user sessions. In Proceedings of the Nordic Conference on Secure IT Systems, Espoo, Finland, 27–29 October 2010; pp. 136–154.
6. Refresh, R. Ricochet Refresh. Available online: https://www.ricochetrefresh.net/ (accessed on 10 June 2022).
7. Winter, P.; Lindskog, S. How the Great Firewall of China is Blocking Tor. In Proceedings of the USENIX Security Symposium, Bellevue, WA, USA, 8–10 August 2012.
8. Woollacott, E. Russia Doubles Down on Censorship with Expanded Block on Tor. 2022. Available online: https://www.forbes.com/sites/emmawoollacott/2021/12/09/russia-doubles-down-on-censorship-with-expanded-block-on-tor/?sh=1a81407e19bc (accessed on 14 July 2022).
9. Rogaway, P. The Moral Character of Cryptographic Work. *IACR Cryptol. EPrint Arch.* **2015**, *2015*, 1162.
10. Ferguson, N.; Schneier, B. *Practical Cryptography*, 1st ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2003.
11. Morkel, T.; Eloff, J.H.; Olivier, M.S. An overview of image steganography. In Proceedings of the ISSA, Sandton, South Africa, 29 June–1 July 2005; Volume 1, pp. 1–11.
12. Hamid, N.; Yahya, A.; Ahmad, R.B.; Al-Qershi, O.M. Image steganography techniques: An overview. *Int. J. Comput. Sci. Secur. (IJCSS)* **2012**, *6*, 168–187.
13. Zielińska, E.; Mazurczyk, W.; Szczypiorski, K. Trends in steganography. *Commun. ACM* **2014**, *57*, 86–95. [CrossRef]
14. Hussain, M.; Hussain, M. *A Survey of Image Steganography Techniques*; The Pennsylvania State University: State College, PA, USA, 2013.
15. Mishra, R.; Bhanodiya, P. A review on steganography and cryptography. In Proceedings of the 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 19–20 March 2015; pp. 119–122.
16. Gupta, S.; Gujral, G.; Aggarwal, N. Enhanced least significant bit algorithm for image steganography. *IJCEM Int. J. Comput. Eng. Manag.* **2012**, *15*, 40–42.
17. Secure Messaging Apps Comparison. Available online: https://www.securemessagingapps.com/ (accessed on 7 July 2022).
18. Signal Messenger. Available online: https://signal.org/pt_PT/ (accessed on 7 July 2022).
19. Threema. Available online: https://threema.ch/ (accessed on 7 July 2022).
20. Wire. Available online: https://wire.com/ (accessed on 7 July 2022).
21. Session. Available online: https://getsession.org/ (accessed on 7 July 2022).
22. Stanescu, D.; Stangaciu, V.; Stratulat, M. Steganography on new generation of mobile phones with image and video processing abilities. In Proceedings of the 2010 International Joint Conference on Computational Cybernetics and Technical Informatics, Timisoara, Romania, 25–27 May 2010; pp. 343–347. [CrossRef]
23. Castiglione, A.; D'Alessio, B.; De Santis, A. Steganography and Secure Communication on Online Social Networks and Online Photo Sharing. In Proceedings of the 2011 International Conference on Broadband and Wireless Computing, Communication and Applications, Barcelona, Spain, 26–28 October 2011; pp. 363–368. [CrossRef]
24. Hiney, J.; Dakve, T.; Szczypiorski, K.; Gaj, K. Using Facebook for Image Steganography. In Proceedings of the 2015 10th International Conference on Availability, Reliability and Security, Toulouse, France,24–28 August 2015; pp. 442–447. [CrossRef]
25. Tao, J.; Li, S.; Zhang, X.; Wang, Z. Towards Robust Image Steganography. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 594–600. [CrossRef]
26. Lu, W.; Zhang, J.; Zhao, X.; Zhang, W.; Huang, J. Secure Robust JPEG Steganography Based on AutoEncoder With Adaptive BCH Encoding. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 2909–2922. [CrossRef]
27. Oliboni, C. OpenPuff (Version 4.01). Available online: https://embeddedsw.net/OpenPuff_Steganography_Home.html (accessed on 25 September 2021).
28. Vaidya, S. OpenStego (Version 0.82). Available online: https://www.openstego.com/ (accessed on 25 September 2021).
29. Hetzl, S. Steghide (Version 0.5.1). Available online: http://steghide.sourceforge.net/ (accessed on 25 September 2021).
30. Foundation, D.E. StegoShare (Version 1.01). Available online: http://stegoshare.sourceforge.net/ (accessed on 25 September 2021).
31. Available online: https://sipi.usc.edu/database/database.php?volume=misc (accessed on 20 October 2021).
32. Huynh, H. Advanced Dontpad. Available online: https://dontpad.herokuapp.com/ (accessed on 9 December 2021).

33. Richard, M. Dontfile. Available online: http://www.dontfile.com/ (accessed on 9 December 2021).
34. Caio, V. Dontpad++. Available online: https://dontpad-plus-plus.firebaseapp.com/ (accessed on 9 December 2021).
35. de Toledo, R. Dontpad. Available online: http://dontpad.com/ (accessed on 4 April 2022).
36. Yao, F.F.; Yin, Y.L. Design and analysis of password-based key derivation functions. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 14–18 February 2005; pp. 245–261.
37. Krawczyk, H. Cryptographic extraction and key derivation: The HKDF scheme. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 2010; pp. 631–648.
38. Krawczyk, H.; Bellare, M.; Canetti, R. *HMAC: Keyed-Hashing for Message Authentication*; Technical Report; The Pennsylvania State University: State College, PA, USA, 1997.
39. Bellare, M.; Canetti, R.; Krawczyk, H. Message authentication using hash functions: The HMAC construction. *RSA Lab. CryptoBytes* **1996**, *2*, 12–15.
40. Turner, S.; Chen, L. *Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms*; Technical Report; The Pennsylvania State University: State College, PA, USA, 2011.
41. Bellare, M. New proofs for NMAC and HMAC: Security without collision resistance. *J. Cryptol.* **2015**, *28*, 844–878. [CrossRef]
42. Biryukov, A.; Dinu, D.; Khovratovich, D. Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS P), Saarbruecken, Germany, 21–24 March 2016; pp. 292–302. [CrossRef]
43. Biryukov, A.; Dinu, D.; Khovratovich, D. phc-Winner-Argon2. Available online: https://github.com/P-H-C/phc-winner-argon2 (accessed on 27 April 2022).
44. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
45. Yu, F.; Li, L.; Tang, Q.; Cai, S.; Song, Y.; Xu, Q. A survey on true random number generators based on chaos. *Discret. Dyn. Nat. Soc.* **2019**, *2019*. [CrossRef]
46. Sathya, K.; Premalatha, J.; Rajasekar, V. Investigation of strength and security of pseudo random number generators. In Proceedings of the IOP Conference Series: Materials Science and Engineering, Erode, India, 15 December 2021; Volume 1055, p. 012076.
47. Özkaynak, F. Cryptographically secure random number generator with chaotic additional input. *Nonlinear Dyn.* **2014**, *78*, 2015–2020. [CrossRef]
48. Braun, T.; Günter, M.; Kasumi, M.; Khalil, I. Virtual private network architecture. In *Charging and Accounting Technology for the Internet (Aug. 1, 1999) (VPNA)*; VPNA: New York, NY, USA, 1999.
49. Zhang, Z.; Zhang, Y.Q.; Chu, X.; Li, B. An overview of virtual private network (VPN): IP VPN and optical VPN. *Photonic Netw. Commun.* **2004**, *7*, 213–225. [CrossRef]
50. Ezra, P.J.; Misra, S.; Agrawal, A.; Oluranti, J.; Maskeliunas, R.; Damasevicius, R. Secured communication using virtual private network (VPN). In *Cyber Security and Digital Forensics*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 309–319.
51. Babkin, V.; Stroganova, E. Evaluation and optimization of virtual private network operation quality. In Proceedings of the 2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), Yaroslavl, Russia, 1–3 July 2019; pp. 1–4.
52. Iqbal, M.; Riadi, I. Analysis of security virtual private network (VPN) using openVPN. *Int. J. Cyber-Secur. Digit. Forensics* **2019**, *8*, 58–65. [CrossRef]
53. Huang, H.Y.; Bashir, M. The onion router: Understanding a privacy enhancing technology community. *Proc. Assoc. Inf. Sci. Technol.* **2016**, *53*, 1–10. [CrossRef]
54. Rao, B.S.; Premchand, P. A Review on Combined Attacks on Security Systems. *Int. J. Appl. Eng. Res.* **2018**, *4562*, 16252–16278.
55. Jana, B.; Chakraborty, M.; Mandal, T.; Kule, M. An Overview on Security Issues in Modern Cryptographic Techniques. In Proceedings of the Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), Jaipur, India, 26–27 March 2018; pp. 26–27.
56. Rackoff, C.; Simon, D.R. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 11–15 August 1991; pp. 433–444.
57. Cramer, R.; Shoup, V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.* **2003**, *33*, 167–226. [CrossRef]
58. Jia, D.; Lu, X.; Li, B. Constructions secure against receiver selective opening and chosen ciphertext attacks. In Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, CA, USA, 14–17 February 2017; pp. 417–431.
59. Yu, N.; Darling, K. A low-cost approach to crack python CAPTCHAs using AI-based chosen-plaintext attack. *Appl. Sci.* **2019**, *9*, 2010. [CrossRef]
60. Bard, G.V. The vulnerability of SSL to chosen plaintext attack. In *Cryptology ePrint Archive*; ESORICS: Guildford, UK, 2004.
61. Qin, Y.; Wan, Y.; Gong, Q. Learning-based chosen-plaintext attack on diffractive-imaging-based encryption scheme. *Opt. Lasers Eng.* **2020**, *127*, 105979. [CrossRef]