ISSN 2007-9737

Overview of Methods to Avoid User Profiling

Abigail Nicolás Sayago, Sergio Gabriel Sánchez Valencia,
Olga Kolesnikova

Instituto Politécnico Nacional,
Escuela Superior de Cómputo,
Mexico

{gabrielsanv97, abigail3nic.say, kolesolga}@gmail.com

Abstract. Nowadays online communication services have obtained a stronger impact in our world. Some companies that provide communication services take advantage of them to create user profiles, which they further use to send undesired advertisements (Ads), or spam, to their users. This article explores several methods, developed based on different areas of knowledge including natural language processing, steganography, cryptography, which can help a user to avoid being profiled while sending and receiving texts over the internet.

Keywords. User profiling, natural language processing, cryptography, steganography, security.

1 Introduction

Privacy is an issue that has been compromised throughout technological advances and the use of the internet. Nowadays big companies try to know everything from their users by gathering information from their activity on social networks, email services, marketing websites, and other means of communication [9]. This is called *user profiling*.

Recently, with the outbreak of coronavirus, the use of online services has increased. For instance, it was mentioned in the *Focus on Powder Coatings journal* [1], that in pandemic situations and the like circumstances, people have to explore alternate means of information sharing. Online courses, messaging, webinars, Podcasts and YouTube shows have been growing in popularity. As online services increase, privacy issues also do. On the other hand, throughout recent years, methods have been developed to keep privacy in messages.

Some methods use cryptography or steganography techniques, which we explain below. This work is focused specifically on protecting users from profilers; we present most relevant methods that can be used to protect users from user profilers, we also analyze the complexity and effectiveness of each method to provide an overview of their specific features.

1.1 User Profiling

More formally, a user profile is a representation of the user's essential information in an online application that the user deals with. Most common contents of a user profile is interests, knowledge, background, skills, goals, behavior, interaction, and preferences [14]. Although it is true that profiling techniques bring certain benefits such as personalized user experience, they have also introduced a privacy problem.

One of the most common tools for user profiling is natural language processing (NLP), which is applied to identify tastes, preferences, and needs of users who share posts or messages on the web. NLP methods include such techniques as sentiment detection and keyword extraction.

The result provided by these techniques assist in classification, or assigning texts to some category. Most popular keyword extraction algorithms are Text Rank [7], RAKE [13], and TF-IDF [4].

Still more powerful text classification algorithms make use of trained neural networks [12]. In this paper, we call *a user profiler* a computer program that processes users' messages and extracts

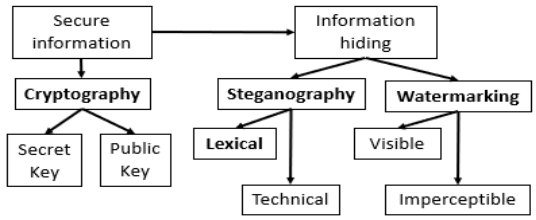

Fig. 1. Overview of methods to avoid user profiling

personal interests, preferences, or knowledge for classification purposes.

1.2 Techniques to Avoid User Profiling

One of the main attempts to keep privacy in messages has been cryptography, which consists in development of methods that aim at preventing unauthorized entities from accessing messages. Some cryptographic methods often use computationally complex and expensive algorithms, whose output generally is a text string unreadable by anyone. This leads to an easy identification of the fact that the text was encrypted, therefore, giving an opportunity to a possible adversary to perform cryptanalysis.

Another significant attempt to keep privacy in messages is steganography, which intends to hide information within a document or a file of any format, including images, video, audio, messages, among others.

However, a format which has caused trouble to researchers, who develop user profiling avoidance methods is plaintext in natural language. There exist still few algorithms and techniques, which can achieve an affective hiding of information within texts by combining several areas of knowledge such as cryptography, natural language processing, and steganography itself. Its main use and development focus have been on linguistic watermarking, whose intention is to provide a tool to identify the author of a particular text, thus protecting this text from copyright problems.

There are several techniques that try to avoid user profiling, like the ones we have already mentioned. Fig. 1 presents a general overview of current approaches. In this article, we review some of most effective techniques, which involve cryptography, steganography, watermarking, and their combinations.

For each technique, we discuss its characteristics, advantages, disadvantages, and complexity.

2 Current Methods

2.1 Cryptography

In the beginning, cryptography was used for secret communications, nowadays it has been used in various online services: bank transactions, online shopping, digital signatures, IT (Internet of Things), among others. The Concise Oxford Dictionary (2020) defines cryptography as the art of writing or solving codes. The services that cryptography provide are privacy, integrity, authentication, and non-repudiation.

Cryptography methods can be divided into two important categories: symmetric and asymmetric ciphers, which we consider in separate sections that follow, showing how these techniques can be used to avoid user profiling and discussing their advantages and disadvantages.

2.1.1 Symmetric Cryptography

The main feature of symmetric cryptography is that all the algorithms have a unique secret key. This key allows for encrypting and decrypting plaintext. Some encryption algorithms that form a part of symmetric cryptography are stream ciphers, block ciphers, and MAC (Message Authentication Code). Fig. 2 shows the scheme of symmetric cryptography where a unique key is used to encrypt the message giving an encrypted text as a result. The key is sent through a secure channel to the receiver who will decrypt the encrypted text. One of the problems of symmetric cryptography is the distribution of the key.

The fundamental idea of stream ciphers is to divide the text into small blocks, for each block a key is generated and the encoding of each block depends on the previous block.

One-time pad is a stream cipher that has the property of *perfect secrecy* where no adversary can get any information about the plaintext by observing the ciphertext, no matter if the adversary has unlimited resources. We can note here that perfect secrecy allows us to avoid user profiling.

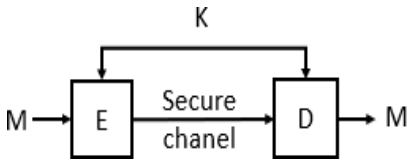

Fig. 2. Symmetric encryption, where M is message, E is its encryption, D is the decrypted message, and K is the key

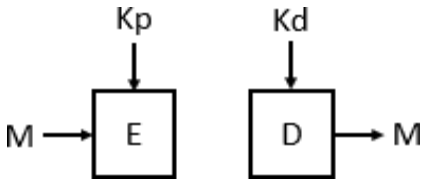

Fig. 3. Asymmetric scheme, where M is a message, E is its encryption, D is the decrypted message, Kp is the public key, and Kd is the private key

One-time pad encryption is defined as:

$$E = M \oplus K,$$

where *M* is the message and *K* is a truly random generated key and is used only once, |*K*| = |*M*|. Decryption is defined as:

$$M = E \oplus K.$$

Block ciphers divide the text into large blocks, for each block the same key is used and the encoding of each block does not depend on the previous one. Commons block ciphers are DES (Data Encryption Scheme), Triple-DES, and AES (Advanced Encryption Standard). The result of these cryptographic algorithms is a sequence of characters that do not make sense even for a computer program.

2.1.2 Asymmetric Cryptography

Different from symmetric ciphers, modern cryptography introduced a new important scheme, the concept of an asymmetric key, which is used in RSA (Rivest-Shamir-Adleman) algorithm, ECC (Elliptic Curve Cryptography), and DH (Diffie-Hellman). The purpose of RSA and ECC is the signature, while in the case of DH the purpose is the exchange of keys. Fig. 3 gives the encryption scheme of an asymmetric algorithm. Every entity that wants to receive a message must have its own private and public key.

For the sake of a better explanation, the two communicating parties are named Alice and Bob. Alice wants to send a message to Bob. In order to encrypt the message, Alice uses the public key of Bob. Bob receives the message from Alice and decrypts it with his private key.

As we mentioned before, modern cryptography introduced a new scheme using a public-key algorithm. RSA is a representative example of this approach.

RSA consists of three procedures: key generation, encryption, and decryption [6].

Key generation is as follows:

1 Choose two prime numbers *p* and *q* (1024 or 2048 bits) randomly.

2 Compute n = p * q.

3 Choose e such that gcd(e, φ(n)) = 1, where φ(n) = (p – 1)(q – 1).

4 Find d such that e * d mod φ(n) = 1.

5 Finally, return e and d, the public key and the private key, respectively.

Imagine we have two entities: Alice and Bob. Alice wants to send a message to Bob. Alice encrypts her message in the following way:

$$E = M^{eB} mod\ n,$$

where *M* is the message, *eB* is defined as a public key of Bob, and $n = p * q$.

Bob decrypts the message:

$$M = E^{dB} mod\ n,$$

where *E* is the encrypted text, *dB* is defined as a private key of Bob, and $n = p * q$.

It can be noticed that with the algorithm of a public key, everyone knows the public key of a person. If we want to send a message to someone, we only need the public key to encrypt the message. To decrypt it, it is necessary to have the private key of the receiver.

The public key algorithm solves the problem of sharing keys in a secure channel, i. e., it resolves the issue that the symmetric scheme has. In the case of the asymmetric scheme, the problem is the public key infrastructure, that is, the generation and authenticity of the keys.

The result of using cryptography will be a sequence of characters or digits without sense,

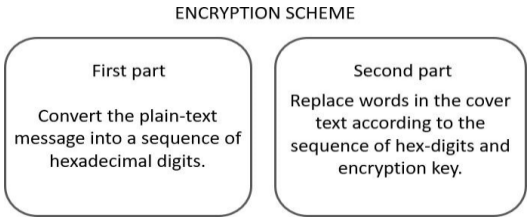

Fig. 4. Lunabel encryption scheme

and if it is accessed by a profiler, the latter will be unable to fulfill its objective.

2.2 Lexical Steganography

2.2.1 Lunabel

Lunabel [3] converts a plaintext into a new semantically and syntactically reasonable text using word replacement.

An important issue of this method is that Lunabel intends to create a natural text that can pass human inspection, as well as be processed and analyzed by an automatic user profiler.

Lunabel uses a so-called cover text to hide the original text. A highly recommended category of cover text is software installations readme.

During the procedure of encryption, a word list is built from the cover text, the list contains 16 words with similar frequencies, syntactic subcategories, and other selected characteristics. Lunabel encryption requires a key that helps to replace words in the cover text, see Fig. 4.

An advantage of Lunabel is that after word replacement, the encrypted text is read as a text that makes sense. Key encryption is realized by a pseudo-random sequence of integers. Decryption is executed by means of the word list and the key.

2.2.2 Steganography using Emoticons

This method replaces each of the characters of the original message by an emoticon based on a table of the corresponding mappings.

A set of predefined keys is used to hide and reveal the hidden message [10]. After encryption, the resulting message is a string containing, for example, the message “Today I felt” followed by the corresponding emoticons and timestamps, the purpose of the latter is to inform of the order of these emoticons at the moment of original text extraction. Since this method uses only a set of specific keys to embed and extract the initial message, it is not secure enough: a brute force algorithm can be applied easily to uncover the hidden message.

2.2.3 UniSpaCH

UniSpaCH considers a mixture of inter-sentence, inter-word, end-of-line, and inter-paragraph spacings to embed a secret message into the cover text [11].

To avoid suspicion, this method embeds payload in segments of two bits in each eligible space or combination of multiple types of spacings. The selected Unicode space characters for this method are *En Quad*, *Em Quad*, *Three-Per-Em*, *Six-Per-Em*, *Figure*, *Punctuation*, *Thin*, and *Hair*.

Since this method uses spaces to hide the secret message, the cover text has to be big enough to generate the necessary amount of spaces.

2.3 Selective Encryption

The main purpose of selective encryption algorithms is to reduce complexity involved in a regular encryption process by encrypting some parts of the message instead of encrypting the whole text. Thus, this technique generates two groups of text: one including the encrypted segments and the other consisting of fragments in the plaintext format. This selection of segments for encryption must be done in such a way that the message will be unreadable by any unauthorized entity, in our particular case, by user profilers.

2.3.1 Toss a Coin

The fundamental principle of this method consists in selecting 50% of the bits (only odd bits or only even bits) in a message to be encrypted and leaving the rest of it as plaintext. This technique implements a fair coin tossing process, where it is expected to obtain 50% heads and 50% tails [8].

The toss-a-coin method reduces the encryption time to a half; thus, it is one of the fastest algorithms to encrypt data.

However, encrypting only a half of the bits makes this method vulnerable to cryptanalysis, which may result in decrypting the hidden message.

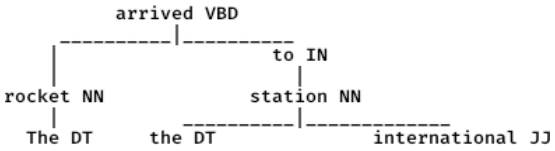

Fig. 5. TMR tree of the sentence “The rocket arrived to the international station”

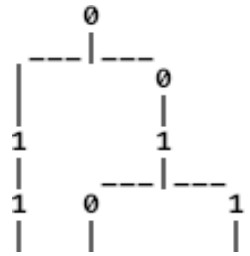

Fig. 6. Tree to create B_i = 1101100

2.3.2 Using Natural Language Processing

This framework introduces a novel technique, which applies Natural Language Processing (NLP) algorithms over the original text. The idea is to select words, which will be encrypted taking as a criterion how meaningful each word is to the text.

Particularly, the first step of the procedure discards stopwords (highly frequent words with abstract grammatical meaning like articles, prepositions, and the like) using a respective wordlist. At the second step, most meaningful words (keywords) are selected from the original text via corresponding NLP techniques [15].

After these steps, the keywords are encrypted using the Blowfish encryption method. Finally, the encrypted message is sent in two portions, the first one contains the encrypted keywords and the second one are stopwords in plaintext [5].

Since this method uses NLP to select most meaningful words, it is guaranteed that the encrypted message does not share sensitive data that could be used in cryptanalysis. Another advantage of this approach is the fact that it does not encrypt the whole text, this makes it competitive in terms of speed.

2.4 Natural Language Watermarking

Applying a watermark to a text is basically the same as applying watermarks to images or videos, the latter being the most popular media where watermarks are used. The intention of this procedure is to provide a tool to identify the author of some object (in our case, it is text), thus protecting the objects from copyright problems.

When applied to text in particular, watermarking is termed linguistic watermarking, it involves more interesting challenges, since one of the objectives is to preserve the meaning of the text, where the message is going to be embedded. The other challenges are quite similar to the ones found in linguistic steganography.

2.4.1 Embedding Information within a Tree Structure of Sentences

In 2001, Atallah *et al*. proposed an algorithm to hide a watermark within text meaning representation trees of sentences (TMR trees, see an example in Fig. 5) instead of doing it directly within the text. TMR trees are constructed with the help of quadratic residues and a large prime number, which is the key to hide and afterwards reveal the hidden message [2].

The tree for encryption is generated as follows, where H is a one-way hash function and T_i is the TMR tree for the i-th sentence.

1 Assign numbers to the nodes of T_i according to a pre-order traversal of that tree (such that the root gets 1, the root’s leftmost child gets 2, etc).

2 Replace every number j at a node by a bit 1 if (j + H(p)) is a quadratic residue modulo p, 0 otherwise.

3 B_i is a listing of the above-obtained bits at the nodes of T_i according to a post-order traversal of that tree (such that the root’s bit is at the end of B_i, the leftmost leaf’s bit is at the beginning of B_i, etc.), see an example in Fig. 6.

The hidden bits of the watermark are inserted into the TMR tree by applying linguistic transformations to the cover text. These transformations must not change the meaning of the cover text, some of such linguistic transformations to sentences are passivization, clefting, preposing, among others. The problem here is that, at this time, there is no ideal method

to do this kind of transformations automatically. Therefore, human interaction might be required.

2.4.2 ANiTW

The ANiTW technique consists of two main algorithms: embedding and extraction [16].

The embedding algorithm consists of three stages:

1. Generate a W_{bits} string for all the letters of the watermark string (W) according to an encoding lookup table.
2. Produce a hidden binary string (HBS) which includes the number of words (nw) of the cover text (CT) and W.
3. Replace the HBS based on a successive 3-bit by zero-width characters (ZWCs) into a W_h using a classification pattern created for this purpose, and embed the W_h after all sentences into the CT.

The extraction algorithm discovers the vector μ based on existing marked words and decodes the contractual 3-bit of each two ZWCs to generate the HBS. Then, it extracts the nw and W by considering the positions of the exclamation mark (!) [16].

This method is one of the most robust and recent developments in the field of natural language watermarking.

3 Method Comparison

Table 1 presents the features of each of the methods discussed in this paper. Each method meets the requirements for the task that it has been developed for.

With strong cryptographic algorithms that provide privacy, integrity, authentication and non-repudiation, we can guarantee that profilers will not be able to read the real contest of the message, however, it is relatively simple to detect whether a message has been encrypted or not, since encrypted messages usually end up being a weird combination of several symbols. Hence, this can lead to potential cryptanalysis attacks.

Furthermore, these methods either require the distributions of key when using secret-key methods or a complicated infrastructure to generate and authenticate keys when using public-key methods.

As well as traditional cryptographic methods, selective encryption is another technique which intends to provide the four cryptography requirements, with an exception that this method tries to make the encryption process more efficient and less costly when messages are transferred using network services. This is achieved by encryption of certain parts of the message and not the whole message. This method possesses some of the important features one needs to avoid user profiling like efficiency, security, and privacy. However, it does not provide invisibility, i.e., the original text can be easily detected as an encrypted message also leading to undesired tries to apply cryptanalysis over the message.

With a similar intention as for the cryptography task but conceptually different, steganography allows users to hide a message in images, video, and text. Focusing on lexical steganography, it can be noticed that these methods permits cryptography engineers to make an analysis of sentences and to create a specific program to avoid user profiling, depending on the requirements of a certain profiling technique. The main characteristic of lexical steganography is that a cover text is required. This method might be useful since its intention is to make a message invisible; however, in our study of steganography techniques we found out that most of such methods do not provide enough security.

Finally, for this overview, we examined such Natural Language Processing methods as Natural Language Watermarking and ANiTW. Both appear to be most promising methods for avoiding user profiling among all methods we considered in this article, since they provide most significant features to fight against user profiling techniques, which are invisibility and security.

However, we noticed that one of these methods (Natural Language Watermarking) requires human interaction to work and a long cover text to hide a small message within it. On the other hand, ANiTW does not require human interaction or a long cover text, which in our opinion makes it ideal to serve as an efficient and secure method to avoid user profiling.

This paper reviews and discusses several methods to protect users from human and machine user profilers. For this purpose, a variety of methods from different areas of knowledge has

Table 1. Method comparison chart

Name	Description	Pros	Cons
		Steganography	
UniSpaCH	Hides the message within a cover text using spaces in this text.	Easy to implement and efficient.	Security is exposed and requires a lot of spaces in the cover text.
Steganography with emoticons	Hides the message within emoticons by applying a mapping to each ASCII character.	Easy to implement and efficient.	Security is exposed and requires a lot of emoticons.
Lunabel	Hides the message in a cover text using word replacement.	Can pass human inspection and, consequently, computer programs as user profilers.	Requires a word list.
		Cryptography	
Caesar Encryption	Classic encryption with a symmetric scheme that returns an unreadable text.	Easy to implement.	It can be detected easily as an encrypted text.
RSA	Encryption with a public key algorithm that returns an unreadable text.	Has strong security.	It can be detected easily as an encrypted text.
		Selective Encryption	
Selective Encryption using NLP	Encrypts only certain fragments of the text using NLP to select the fragments.	Efficient and secure.	It can be detected easily as an encrypted text.
Toss a Coin	Encrypts 50% of the message.	Efficient.	Poor security and it can be detected easily.
		Natural Language Processing	
NL Watermarking	Applies natural language transformations to embed message into a cover text.	Invisible and secure.	Requires a long cover text and human interaction.
ANiTW	Applies math formulas to embed the bits of the message into a cover text.	Invisible and secure.	Only employed in Latin-based texts (English, German, Italian, etc.) and not applicable to Persian/Arabic and Urdu languages.

been developed, each of them has its own advantages and disadvantages.

Most attractive features of the methods are invisibility, unreadability, efficiency, security. These characteristics can be observed in some of the methods we considered, however, we found that at present, there is no single method that possesses all of these features, i.e., methods that provide unreadability might be poor in terms of efficiency, some other methods might have invisibility but not enough security.

We hope that in future, new efficient and robust techniques to avoid user profiling will be developed.

Acknowledgments

The work was done under partial support of Mexican Government: SNI, COFAA-IPN, BEIFI-IPN, and SIP-IPN grant 20201948.

References

1. **Biller, K. (2020).** Moving forward with the new normal. *Focus on Powder Coatings*, Vol. 2020, No. 6, pp. 1. DOI:10.1016%2Fj.fopow.2020.05.001.
2. **Atallah, M.J., Raskin, V., Crogan, M., Hempelmann, C., Kerschbaum, F., Mohamed, D., & Naik, S. (2001).** Natural language watermarking: Design, analysis, and a proof-of-concept implementation. *International Workshop on Information Hiding*, pp. 185–200.
3. **Chand, V. & Orgun, C.O. (2006).** Exploiting linguistic features in lexical steganography: Design and proof-of-concept implementation. *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, pp. 126b–126b, DOI:10.1109/HICSS.2006.175.
4. **Jones, K.S. (1972).** A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, Vol. 28, No. 1, pp. 11– 21.
5. **Kushwaha, A., Sharma, H., & Ambhaikar, D.A. (2017).** Selective encryption using natural language processing for text data in mobile ad hoc network. *Modeling, Simulation, and Optimization*, pp. 5–26. DOI:10.1007/978-3-319-70542-2_2.
6. **Mahajan, P. & Sachdeva, A. (2013).** A study of encryption algorithms AES, DES and RSA for security. *Global Journal of Computer Science and Technology*, Vol. 13, No. 15E, pp. 14–18.
7. **Mihalcea, R. & Tarau, P. (2004).** Text rank: Bringing order into text. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 404–411.
8. **Nithyapriya, J., Anandha-Jothi, R., & Palanisamy, V. (2020).** Securing Data with Selective Encryption Based DAC Scheme for MANET. *International conference on Computer Networks, Big data and* IoT, Vol. 31, pp. 133–139. DOI:10.1007/978-3-030-24643-3_15.
9. **Paik, W., Yılmazel, S., Brown, E., Poulin, M., Dubon, S., & Amice, C. (2003).** Applying natural language processing (NLP) based metadata extraction to automatically acquire user preferences. *Proceedings of the 1st International Conference on Knowledge Capture*, pp. 116–122. DOI:10.1145/500737.500757.
10. **Patiburn, S., Iranmanesh, V., & Teh, P. (2017).** Text steganography using daily emotions monitoring. *International Journal of Education and Management Engineering*, Vol. 7, pp. 1–14. DOI: 10.5815/ijeme.2017.03.01.
11. **Por, L.Y., Wong, K., & Chee, K.O. (2012).** Unispach: A text-based data hiding method using unicode space characters. *Journal of Systems and Software*, Vol. 85, No. 5, pp. 1075–1082. DOI: 10.1016/j.jss.2011.12.023.
12. **Prasanna, P. & Rao, D. (2018).** Text classification using artificial neural networks. *International Journal of Engineering and Technology (UAE)*, pp. 603– 606.
13. **Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010).** Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, pp. 1–20. DOI:10.1002/9780470689646. ch1.
14. **Schiaffino, S. & Amandi, A. (2009).** *Intelligent User Profiling*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 193–216. DOI:10.1007/978-3-642-03226-4_11.
15. **Sidorov. G. (2019).** *Syntactic n-grams in computational linguistics*. Springer, 2019, 98 p.
16. **Taleby-Ahvanooey, M., Li, Q., Zhu, X., Alazab, M., & Zhang, J. (2020).** Anitw: A novel intelligent text watermarking technique for forensic identification of spurious information on social media. *Computers Security*, Vol. 90, pp. 1–16. DOI: 10.1016/j.cose. 2019.101702.

Article received on 14/07/2020; accepted on 29/09/2020. Corresponding author is Olga Kolesnikova.