Proactive Biometric-Enabled Forensic Imprinting

Abdulrahman Alruban[1, 2] ✉, Nathan Clarke[1, 3], Fudong Li[1] and Steven Furnell[1, 3, 4]
[1]Centre for Security, Communications and Network Research, Plymouth University
Plymouth, UK
[2]Computer Sciences and Information Technology College, Majmaah University
Majmaah, Saudi Arabia
[3]Security Research Institute, Edith Cowan University
Perth, Western Australia
[4]Centre for Research in Information and Cyber Security, Nelson Mandela Metropolitan University
Port Elizabeth, South Africa
{abdulrahman.alruban, n.clarke, fudong.li, steven.furnell}@plymouth.ac.uk

Abstract—**Threats to enterprises have become widespread in the last decade. A major source of such threats originates from insiders who have legitimate access to the organization's internal systems and databases. Therefore, preventing or responding to such incidents has become a challenging task. Digital forensics has grown into a de-facto standard in the examination of electronic evidence; however, a key barrier is often being able to associate an individual to the stolen data. Stolen credentials and the Trojan defense are two commonly cited arguments used. This paper proposes a model that can more inextricably links the use of information (e.g. images, documents and emails) to the individual users who use and access them through the use of steganography and transparent biometrics. The initial experimental results of the proposed approach have shown that it is possible to correlate an individual's biometric feature vector with a digital object (images) and still successfully recover the sample even with significant file modification. In addition, a reconstruction of the feature vector from these unmodified images was possible by using those generated imprints with an accuracy of 100% in some scenarios.**

Keywords—Digital forensics; biometrics; grille cipher, data leakage; guilty identification.

I. INTRODUCTION

Insider threats are considered to be a significant security issue [1–3]. The recent decade has witnessed countless numbers of data loss and exposure incidents all over the world in which data has become publicly available and easily accessible [4]. The impact of losing or disclosing sensitive data or confidential intellectual property might cause substantial financial and reputational damage to the company. In particular, when the exposure is originated by an authorised individual (i.e. employee, contractor, etc.) who misuses their legitimate access, the opportunity for adverse impacts is typically greater. Since insiders are more likely to bypass some security controls compared to outsiders who might have limited knowledge about the internal infrastructure [5, 6]. Therefore, insiders pose significantly greater threats to organisations than the outsiders do.

One of the aims of the digital forensics process is to produce and test a hypothesis about who did what, where and how in relation to the incident under investigation. Indeed, existing methods and tools used by investigators to conduct examinations of a digital crime significantly help in collecting, analysing and presenting the digital evidence [7–9]. However, the question of who did the crime is crucial, especially if the digital forensics process leads to the presentation of findings in a court of law [10]. Therefore, digital forensics investigators have to link the identity of a digital object to a human as opposed to just using an electronic record or a log that indicates a user interacted with the questioned object (evidence). Indeed, this is a challenging task, because it is currently difficult for digital forensics professionals and investigators to prove beyond a reasonable doubt in a court of law that a specific human being has used the specific identity of a digital subject at a certain time [11–13]. As a possible solution to this problem—suggested by the authors of this paper—the use of transparent biometrics could provide such a link. Moreover, transparently capturing the user's biometrics and instantly generating a *biometric imprint* that correlates the user interaction with the used object could give rise to important information, which would help investigators answer the question "who?".

This paper introduces a proactive framework that uses transparent biometrics to aid digital forensic investigators in their analysis of electronic evidence. Furthermore, it examines the feasibility of linking a subject (computer user) with an object of interest such as images, documents, or emails. In addition, this investigation develops a set of experiments that employ a grille cipher to link embed the transparent biometric sample. Unlike most existing methods such as digital watermarking or null ciphers (form of encryption where the plaintext is mixed with a large amount of non-cipher material) the integrity of the object is modified [14, 15], a grille cipher employs a template that is used to cover the carrier message; the words that appear in the openings of the template are the hidden message. Further, the proposed approach only "imprints" the object with any given data (i.e. user's biometric feature vector). Therefore, the employed imprinting process can

GB Patent filing GB1609673.7 dated 2nd June 2016

be described as a correlation of the feature vector with the object.

The rest of the paper is organised as follows: Section II highlights the related work in the area of guilt identification and proactive digital forensics. Section III introduces the proposed approach, including the core process. Section IV explains the experimental methodology of different types of attack vectors to evaluate the robustness of the proposed method. Section V presents the experimental results. Section VI covers the discussion for the findings and possible future works. Finally, the paper concludes in Section VII.

II. RELATED WORK

A variety of studies have examined the possibility of identifying the person that leaked data [16–19]. In [16] the authors investigated the feasibility of inserting fake objects into data of interest for the purpose of distributing these data to third party agents. The idea was to add a unique object into the data prior to handing it out to those agents. However, adding these fake objects is not always possible. For example, in the case of medical records, manipulating the data or injecting invalid information could lead to huge risk and consequences on the patients' life. The examination of the feasibility of their method found that it is better in identifying the source of the data leakage compared to the simple data allocation algorithms. Moreover, 95% of confidence was obtained via their experiments.

Subsequent practical implementations of the guilt model published in [17–19] resulted in the development of several prototype models. All of these models use the same concept introduced in [16] by inserting unique fake objects or digital watermarks to the data prior to the distribution. In general, the data creator (in this case the distributer) is responsible for generating and embedding the fake objects. However, in many cases the data can be created by an insider who leaks the sensitive data by himself. In additional, the fake object creation process could be a complicated task.

From a forensic prospective, [20] proposed a system that proactively and continuously collects evidence by creating and storing file signatures that are deleted, edited, or copied within computers on the local network. The system uses a centralized database to store the generated objects' signatures, which provide significant information, such as user identifier, object time stamp, and type of the event. For instance, events like file creation or deletions, user identifier, file name, file path, a time stamp for the event, and a machine identifier. This is helpful especially when conducting a forensic activity. The generated fingerprints are equal to ~1.06 percent of the original file size, which is a huge reduction in terms of storage space. Further, the system supports several file types, such as Microsoft Word documents and Portable Document Formats (PDF). For the deployment, the system requires patching the system kernel in order to intercept system calls. Unfortunately, such low level kernel hardcoding is limited to open source operating systems. In contrast, our proposed approach does not require any modification on the kernel level.

III. THE PROPOSED APPROACH

The proposed framework consists of two engines: biometrics and a grille cipher engine. The biometrics engine transparently captures and extracts the user's biometric samples (e.g. facial features, keystroke analysis, behavioural profiling) and stores them in a database on the user's computer. The grille cipher engine retrieves the object metadata and its Hex representations and requests the latest user's biometric feature vector from the biometric engine to be used in the imprinting process. Finally, these generated imprints are stored in a centralized database for later analysis when required. Fig. 1 illustrates the framework architecture for the proactive biometric-enabled forensic imprinting system.

Upon the detection of data leakage, the object (whether it be posted on a public website or captured by the network) can be analysed for the biometric imprint. The sample is extracted and then processed by a biometric system in order to determine the last user who interacted with the object as presented in Fig. 2.

The generation process of the imprints is inspired by the benefits of employing the grille cipher technique. Grille ciphers has been used in the past (prior to the modern null ciphers) as a means for transferring/exchanging secret messages between two parties. It was originally used to extract hidden messages from plain text by mapping the text throughout a pierced sheet or such a cardboard. For instance, the words "secret" and "plan" can be extracted from a letter puzzle by applying appropriate cardboards that map the desired locations of the letters, as Fig. 3 illustrates. Therefore, the embedded secret message can be retrieved by mapping specific locations. Hence, applying the same technique to imprint the biometric feature vector to an object file is possible, where the object can be an image file, document, video, or any digital file types. In order to adapt the grill cipher technique to the proposed approach, it involves several consecutive steps, as follows:

1) Preparation of Feature Vector and Object:

The preparation step converts both feature vector and object into its Hex representations for the mapping purpose. In addition, the index of each character is preserved during this conversion, which begins with '0' for the first character and ascendingly continues until the last one. Furthermore, the process of conversion is not necessarily achieved by transforming each character, since reading the whole object in binary mode allows for low-level representations of both Hex and Binary. However, still character-by-character (or byte-by-byte) indexing is required in order to generate the object index list.

2) Mapping the Feature Vector with the Object:

After obtaining the Hex representations of the feature vector and object, each Hex value in the feature vector are mapped with its equivalent positions in the object's Hexes to retrieve the possible positions where both are match.

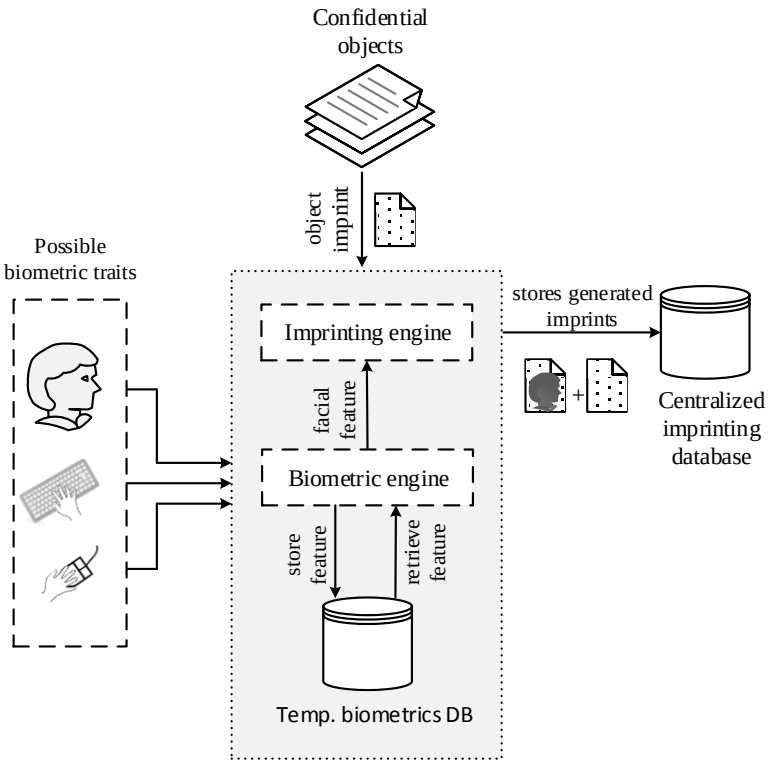

Fig. 1. The Proposed framework architecture

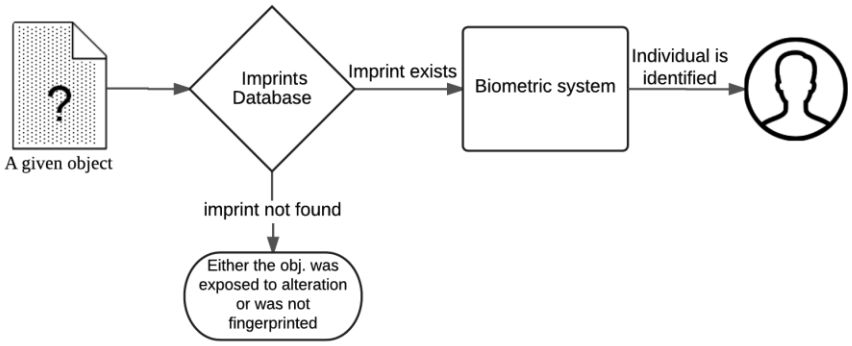

Fig. 2. The process of identifying an individual

Accordingly, the mapping process returns lists of indexes for those matched Hexes.

3) Generating the Feature Vector Imprints:

By retrieving the positions of each character of the feature vector with the object, now it is possible to generate the imprints based on the list of indexes, which means that multi-imprints of the whole feature vector can be generated by combining those positions.

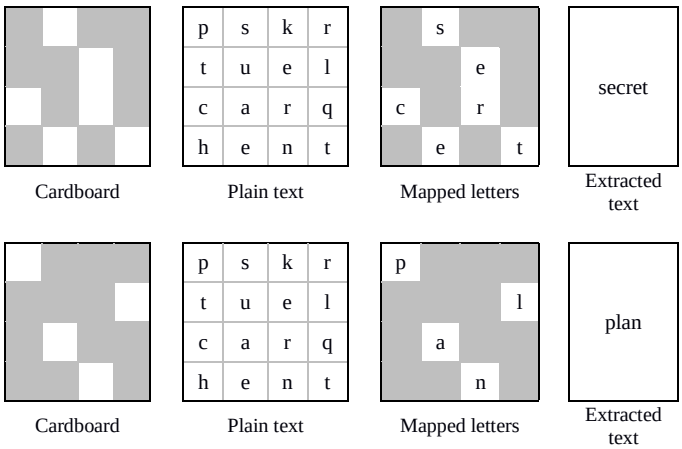

Fig. 3. Grille cipher

The pseudocode of the imprinting process starting from the preparation is illustrated below in Algorithm 1.

Algorithm 1: imprinting algorithm:

Input: Feature Vector (*FV*), Object (*O*)
Output: Imprints
function PREP (*FV, O*)
    for *each value in FV & O:*
        Convert *FV, O* into its HEX representations
        Retrieve the index of each value
    Return $FV_{HEX,\ index}, O_{HEX,\ index}$
function MAPPING ($FV_{HEX,\ index}, O_{HEX,\ index}$)
    for each value in $FV_{HEX}, O_{HEX:}$
        $index\,(O_{index}) \leftarrow FV_{HEX} \cap O_{HEX}$
    Return $index\,(O_{index})$
function IMPRINTING (*indexes*)
    imprint ← Combine unique indexes from the retrieved index list
    Return *imprints*

The following example explains how this algorithm works in practice. For the demonstration purpose, assume that the following feature vector needs to be mapped into an object as presented in Fig. 4.

FV: [012345]
O: [012345012345012345]

Fig. 4. Feature vector and an object

Regardless of the file type of the object, since any file type can be transformed and treated as a Hex representation. The first step in the proposed algorithm is to convert both of the feature vector and object into its Hex representations. According to the ASCII table, Fig. 5 shows the converted characters as Hex alongside the position of each value (index). In this example, each value of the feature vector exists in more than one location within the object. For example, "30" (the Hex representation of "0") is located in positions 0, 6, and 12. In the same manner, the mapping process continues for all subsequent feature vector's values until all possible positions are retrieved.

In addition, Table 1 presents the retrieved positions for each value of the feature vector.

TABLE 1: FEATURE VECTOR VALUES POSITIONS IN THE OBJECT

Original Value	*Hex Representation*	*Positions in the object*
0	30	0, 6, 12
1	31	1, 7, 13
2	32	2, 8, 14
3	33	3, 9, 15
4	34	4, 10, 16
5	35	5, 11, 17

The last step in this example is to generate all possible imprints from those retrieved positions. Since each feature vector value is located in three different locations, the total unique imprints that can be generated from these positions are three as listed in Table 2. Therefore, using any value of these imprints, it is possible to reconstruct the original feature vector from the object by reversing the mapping process. After explaining how the imprinting technique works through the given example, the next section investigates the feasibility of imprinting biometric feature vectors with images and later recovering they (even after object modification).

TABLE 2: POSSIBLE IMPRINTS

Imprint number	*Imprint*
1	0, 1, 2, 3, 4, 5
2	6, 7, 8, 9, 10, 11
3	12, 13, 14, 15, 16, 17

IV. EXPERIMENTAL METHODOLOGY

The main goal of the experiment is to assess the feasibility of the proposed hypothesis where the subject's feature vector can be forensically linked and retrieved from an object of interest. Therefore, it is critical to evaluate its performance in a complex, subject-related manner. In total, four experiments were conducted as follows:

- The first experiment retrieves the feature vector from the original imprinted image.
- The second experiment examines the situation where the image is modified in one area with an increasing proportion of modification.
- The third experiment verifies the case where the image was modified in several areas.
- The final experiment investigates when only parts of the original image are available, while the rest is missing.

In these experiments, the used feature vector presents a real facial feature vector sample with a length of 57 numeric characters, as illustrated in Fig. 6. The length of the vector relies upon the used feature extraction algorithm to compute the feature vector. In this study, Fisherfaces algorithm is used to compute the feature vector for the captured users' faces images [21]. In addition, the algorithm performs a Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for dimensionality reduction [22].

Regarding the used objects in the performed experiments, the UCID image dataset version 2 is used [23]. It contains a total of 1,300 images with two sizes, either (1,234 x 1,858) or (1,858 x 1,234) width, height in pixels respectively. For the purpose of this study, only the first 100 images are used from this dataset, since it is assumed that this number is enough for the purpose of evaluation. The implementation of the proposed algorithm was developed in Python due to its flexibility in terms of list comprehension and image processing. Moreover, Python's built-in library has several useful functions, such as *map* and *zip* which facilitate many relevant operations [24]. As regards the deployment, these tests have been conducted on a machine with

FV	**0**	**1**	**2**	**3**	**4**	**5**
$Hex(FV)$	30	31	32	33	34	35
$Hex(FV)index$	0	1	2	3	4	5

O	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**	**0**	**1**	**2**	**3**	**4**	**5**
$Hex(O)$	30	31	32	33	34	35	30	31	32	33	34	35	30	31	32	33	34	35
$Hex(O)index$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Fig. 5. The Hex representation of the feature vector and the object

[1679.2235398,-1555.40390834,-1140.07728186,-1999.85500108]

Fig. 6. Facial feature vector

Microsoft Windows 7, Intel Core i5 2.70GHz and RAM 4.00 GB.

A. Retrieving the Feature Vector from the Original Imprinted Image

The aim of this experiment is to imprint the feature vector as many times as possible with each image in the dataset. The first experiment examines the possibility of generating the imprints between the feature vector and the used object. Since there is a high probability that the subject or other party (for intentional or unintentional reasons) somehow could modify the questioned object after it is imprinted, therefore, in the subsequent experiments investigates the accuracy of retrieving the feature vector from the object under several situations.

B. Modification in One Area

Experiment two evaluates the imprinting mechanism after the image is modified by a different percentage. The simulation of this is performed by randomly choosing a section of the image as a rectangle box at a growing size to reflect an increasing proportion of modification. In addition, equation 1 is used to determine the size and the random position of the modified section. The equation takes three variables, which are:

- *w*: image width,
- *l*: image height,
- *s*: the desired modification percentage.

The equation gives four values; x and y are random values between (0, image width) and (0, image height) respectively. These set the top left pixel position of the modified rectangle (as presented in red colour in Fig. 7). The third and fourth values are for the right down corner of the rectangle (as presented in blue colour).

$$P_{(w,l,s)} = \sum_{x=0}^{w-1} \sum_{y=0}^{l-1} (x, y,\ x + \frac{w}{10 \cdot \sqrt{s}}, y + \frac{l}{10 \cdot \sqrt{s}}) \qquad (1)$$

In this experiment, the imprinted images have been modified by 5% increments, which means that the first alteration rate is 5% then 10%, 15% and so forth, until reaching 100%. Fig. 8 demonstrates some samples of an image modified in different rates. The upper left image is modified by 5% of its original size, where the rest are modified at rates of 35%, 65%, and 95% respectively.

C. Modification in Multi Areas

The third experiment is similar to the previous one, except that the modifications occur in several parts of the image instead of an increasing proportion of one area. This type of attack is more influential since various and random parts of the image are affected by such alterations. In order to simulate such modifications, the dataset images are altered using multiple rectangle boxes, each of which is equal to 1% of the total image size. Therefore, simulating 5% randomly locations alteration, it would need five of these boxes among an image. In addition, this experiment assesses the proposed technique with an alteration size on the objects by 5% increments of its original size. Fig. 9 illustrates four sample images modified by 5%, 35%, 65%, and 95% respectively.

D. Image Partial

Further investigation was needed to better understand the effects of different attack vectors on retrieving the imprinted feature vector. Therefore, the last experiment in this study is interesting in terms of the obtained results. It simulates the scenario, where only part of the imprinted image is available and the rest is missing; for instance, the imprinted image could be resized or cropped. To simulate such alterations, a random section of the images in the dataset was cropped in different sizes, starting from 5% of the original size, and then in each subsequent test, again a random section was cropped with an increment of 5%. Fig. 10 illustrates some of these cropped samples.

V. RESULTS

In this study, the aim is to critically assess the hypothesis of linking a subject's biometric feature vector to an object of interest using the grille cipher technique. In average, it takes only ~3 milliseconds to generate an imprint with size average of those imprints is less than ~472 bytes per imprint. The result of experiment one shows that the average number of the generated imprints are 854 per image. While the minimum number of imprints in a single image was 244, and the maximum is 1,815. This means that the mapped feature vector could be retrieved and reconstructed from any of these imprints. This achieved number of imprints is not surprising, since the feature vector always contains numerical values (0-9). Therefore, there are many matches between the feature vector and those images' Hex values. In addition, a reconstruction

Fig. 7. Sample of a modified area

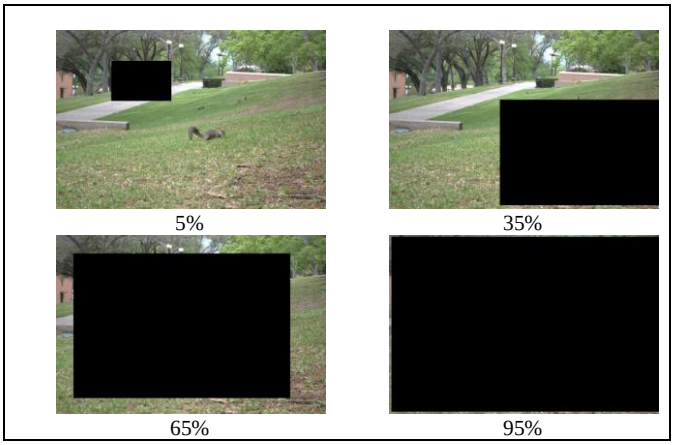

Fig. 8. Sample of a modified part of an image

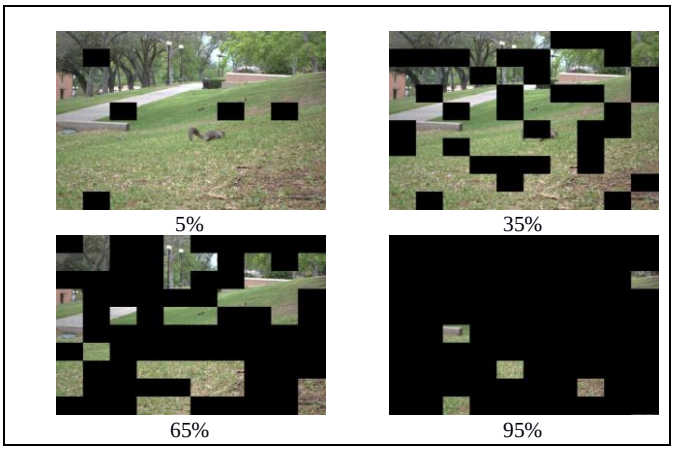

Fig. 9. Sample of a modified multiple parts of an image

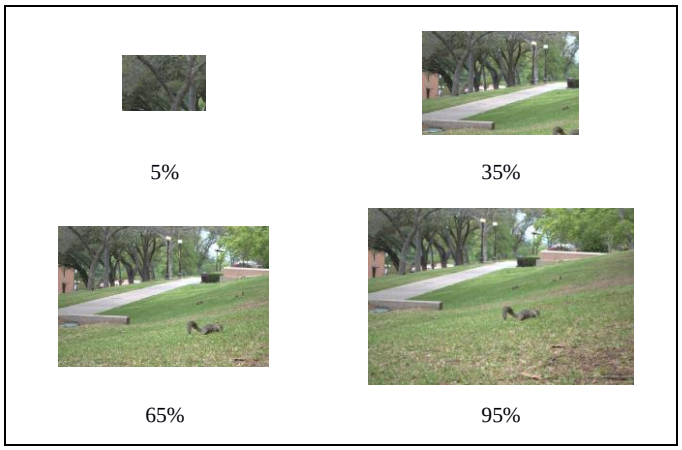

Fig. 10. Samples of a cropped image in certain percentage

of the feature vector from these unmodified images were possible by using those generated imprints with an accuracy of 100%. This was achieved easily by reversing the imprinting processes.

In the second experiment, it was found that this imprinting technique is very effective, since the imprinted feature vector is successfully retrieved from an average of 97 out of 100 images even when the modification percentage is 80%, as Fig. 11 illustrates. However, after a modification of 80% on the images, the number of valid retrieved feature vector significantly drops due to the loss of most of the imprints values across those images. This decline occurred for the reason that critical set of mapped indexes values are changed after such high modification rate. Yet, it is clearly illustrated that it is feasible to reconstruct the feature vector from the imprinted objects even though the huge destruction to its original values.

In the third experiment where the modification took a place in multiple areas, the result shows that the imprinted feature vector are successfully retrieved, even when the images are altered in more sophisticated way than the one area modification attack (experiment two). Fig. 12 exhibits the percentage of images where the feature vector was successfully retrieved. Since changing certain pixels' values-by printing those black boxes- after the imprinting process with the feature vector consequently affects the mapped indexes' values. Therefore, many of the imprints became useless after such attack. Despite massive destruction on the image visualisation with the increased rate of the modification, it is possible to recapture the feature vector from some of those images, even under enormous alteration such as when the object is changed by 95%. At the same time, this attack caused a major loss of the mapped indexes values comparing to the modification in one area experiment. Where the latter is less vandalism than the former in terms of impacting the interested pixels.

Finally, in the last experiment the most striking finding to emerge from the results is that among all these tests in this experiment, the feature vector is retrieved and reassembled 100% among all the tested images. This means that by giving

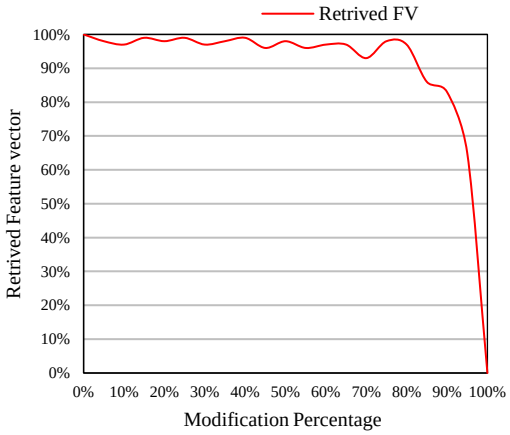

Fig. 11. % of images with successful retrieved feature vectors under one area modification attack

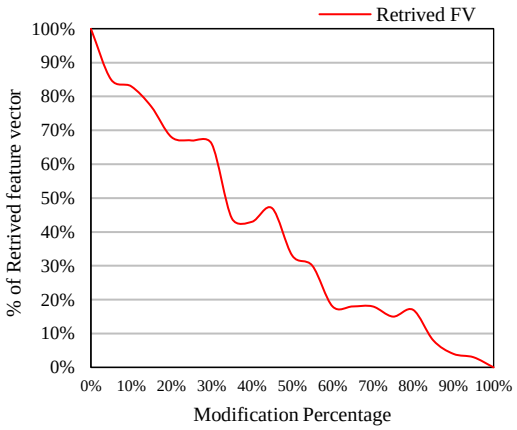

Fig. 12. % of images with successful retrieved feature vectors under multiple parts modification attack

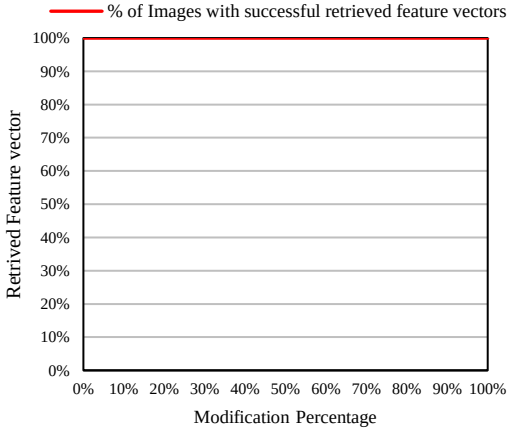

Fig. 13. % of images with successful retrieved feature vectors under partial image attack

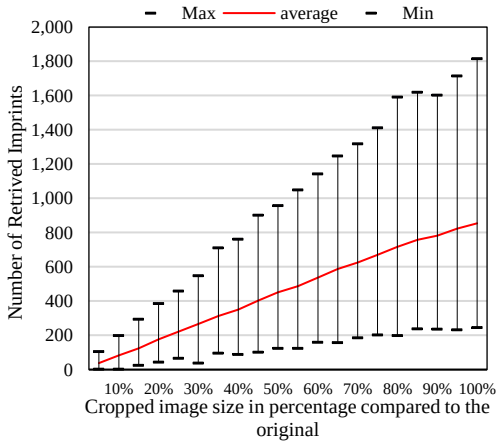

Fig. 14. Performance under partial image attack

only part of the original imprinted image, it is possible to restore the feature vector to its original values. Fig. 13 shows the percentage of the successful retrieved feature vector under the partial image attack. In addition, the results in Fig. 14 also show that the average, maximum, and minimum numbers of a retrieved feature vector cross on all examined images (i.e. 100 images). However, these results were obtained by assuming that the preserved indexes of the hexes of interest are not changed after the cropping process. This means that all of the imprints in the database are correlated with the questioned samples as a part of the original images. In practice this is not always possible since the original object might not be accessible or available after the imprinting process took a place. Therefore, more research is needed to find a link between such parts of an object and the original.

VI. DISCUSSION

The nature of the imprinting process reveals no information about where to locate the imprinted object–thereby making it particularly challenging to recover or modify–as illustrated in the experiments that have been conducted. In addition, the results have evidently shown that by mapping the Hex representations of a feature vector with the Hex representations of an image of interested, it is feasible to generate one or more imprints of this feature vector. The first conducted experiment results revealed an 'expected' outcome by imprinting the feature vector from the original imprinted image. Since 100% of the imprinted feature vector is retrieved using only the generated imprints that contains the indexes of the corresponding positions, it is expected because the mapped objects (images in this case) have not been exposed to any kind of alteration and, therefore, were tested based on their original status. The explanation of being able to score those high results is attributed to the nature of the examined object. Since images are a set of pixels that range from 0 to 255, changing one pixel's value does not affect other pixels' values, or their position. Thus, altering part of the image is not necessary, as it affects all imprinted indexes' values. Therefore, generating as many imprints as possible in various positions of the image, this in turn will increase the probability of successfully retrieving the imprinted values.

It is worth to mention that the approach introduced in this paper could be applied to other types of objects such as Office Word documents and PDF files. Nevertheless, the results do not necessarily reflect a robust success rate, since those types of objects are considered a binary format for storing a document. In addition, an initial experiment was carried out where a small set of Office Word and PDF files are examined using the same imprinting technique that was conducted on images. The results showed that unlike images, the dynamic nature of binary files makes changing a small value in the document/file content require recompiling the whole binary file, which consequently leads to adjustment of the whole indexes sequence. For that reason, many attacks would considerably impact the accuracy of retrieving the imprinted feature vector from such objects. Therefore, further work needs to be undertaken to ensure the biometric capturing, processing and imprinting systems need to be hardened against attack and modification in order for the approach to remain valid.

Nevertheless, these findings provide interesting insights for future research, where other techniques could be investigated for robust object alteration. In addition, a possible solution could tackle such issue is instead of mapping the feature vector with the object at a Hex level, a higher level of representation could be used. For instance, in the case of documents, mapping the feature vector with static representations of the document's text possibly will become less vulnerable to such alteration attack, especially when the generated imprints preserve more static values related to the object.

VII. CONCLUSIONS

This paper proposed a proactive framework that uses transparent biometrics to inextricably link the use of information (e.g., images, documents) to the individual users who use and access them rather than intermediate controls. Such approach aids digital forensic investigators in their analysis of an electronic evidence and could increases the likelihood of the evidence to be admissible in a court of law. The results of all the conducted experiments show that even when the object is altered in a sophisticated way, there is still a chance to retrieve and reconstruct the imprinted feature vector. From a privacy prospective, such an approach would require modifications to the relevant employee computer use policy, thus to make them aware that such a system was operating and what was happening to their biometric information.

Despite these promising results, the use of transparent biometrics to monitor and acquire subject's traits introduces several challenges that need to be considered when developing such a system. For instance, in the case of facial detection, the environmental and external factors such as light, subject distance from the camera and face orientation significantly affect the accuracy of the obtained samples. Even with extensive research being undertaken in this field, such issues cannot be overcome very easily, especially when the operation of transparent biometric monitoring is meant to be unobtrusive and unsupervised.

REFERENCES

[1] C. L. Huth, "The insider threat and employee privacy: An overview of recent case law," *Comput. Law Secur. Rev.*, vol. 29, no. 4, pp. 368–381, 2013.

[2] R. F. Collins, M. L., Spooner, D., Cappelli, D., Moore, A. P., & Trzeciak, "Spotlight On : Insider Theft of Intellectual Property inside the U . S . Involving Foreign Governments or Organizations," 2013.

[3] A. Shabtai, Y. Elovici, and L. Rokach, "A Survey of Data Leakage Detection and Prevention Solutions," no. 2008, pp. 5–11, 2012.

[4] Verizon, "2015 Data Breach Investigations Report," *Verizon Business Journal*, 2015. [Online]. Available: https://msisac.cisecurity.org/whitepaper/documents/1.pdf. [Accessed: 06-Jan-2016].

[5] W. Hurst, M. Merabti, and P. Fergus, "A Survey of Critical Infrastructure Security," in *Critical Infrastructure Protection VIII*, vol. 441, 2014, pp. 127–137.

[6] T. S. Perimeter, "People Are the Perimeter," in *Managing Risk and Information Security*, Apress, 2013, pp. 57–69.

[7] F. Carbone, *Computer Forensics with FTK*, 1st ed. Birmingham: Packt Publishing Ltd., 2014.

[8] S. Widup, *Computer Forensics and Digital Investigation with EnCase Forensic v7*, 1st ed. McGraw-Hill Osborne, 2014.

[9] SANS Institute, "SANS Investigative Forensics Toolkit Documentation," 2016. [Online]. Available: https://media.readthedocs.org/pdf/sift/latest/sift.pdf. [Accessed: 12-Feb-2016].

[10] A. Valjarevic and H. Venter, "Towards Solving the Identity Challenge," in *The 7th International Workshop on Digital Forensics and Incident Analysis (WDFIA 2012)*, 2012, pp. 129–138.

[11] C. S. D. Brown, "Investigating and prosecuting cyber crime: Forensic dependencies and barriers to justice," *Int. J. Cyber Criminol.*, vol. 9, no. 1, pp. 55–119, 2015.

[12] E. A. Vincze, "Challenges in digital forensics," *Police Pract. Res.*, vol. 4263, no. February, pp. 1–12, 2016.

[13] B. Shavers, *Placing the suspect behind the keyboard: using digital forensics and investigative techniques to identify cybercrime suspects*. Newnes, 2013.

[14] M. D. Nelson and M. Xie, "DATA LEAK PROTECTION," US Patent 2014/0007246 A1, 2014.

[15] S. R. D. J. Charbonneau and E. J. Simon, "Method and system for generating trusted security labels for electronic documents," US Patent 8,869,299 B2, 2014.

[16] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 51–63, Jan. 2011.

[17] S. A. Kale and S.V.Kulkarni, "Data Leakage Detection," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 1, no. 9, pp. 668–678, 2012.

[18] R. Jadhav, "Data leakage detection," *Int. J. Comput. Sci. Commun. Networks*, vol. 03, no. 01, pp. 37–45, 2012.

[19] J. Chavan and P. Desai, "Relational Data Leakage Detection using Fake Object and Allocation Strategies," *Int. J. Comput. Appl.*, vol. 80, no. 16, pp. 15–21, 2013.

[20] C. Shields, O. Frieder, and M. Maloof, "A system for the proactive, continuous, and efficient collection of digital forensic evidence," *Digit. Investig.*, vol. 8, no. SUPPL., pp. S3–S13, Aug. 2011.

[21] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, 1997.

[22] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data — with application to face recognition," *Pattern Recognit.*, vol. 34, no. 10, pp. 2067–2070, 2001.

[23] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *SPIE 5307, Storage and Retrieval Methods and Applications for Multimedia 2004*, 2003, pp. 472–480.

[24] "2. Built-in Functions — Python 2.7.11 documentation." [Online]. Available: https://docs.python.org/2/library/functions.html. [Accessed: 06-Feb-2016].