Secure user Browser Activity using Hybrid Data Hiding Techniques

Mohd Fahmi Mohamad Amran, Yamunah Kathiravan, Noor Afiza Mat Razali, Raja Mohd Tariqi Raja Lope Ahmad, Zuraidy Adnan, Muhammad Fairuz Abd Rauf, Mohd Afizi Mohd Shukran

Abstract: ***Web browsers may delete some files but it doesn't delete everything. The purpose of private browsing is for users to browse private mode just as a standard browsing session would, but without storing any data such as log-in credentials or browsing history upon exit. A secure framework to secure the web browser artefacts is proposed to fulfil the requirements. In order to compare and contrast the different methods of artefacts encryption, a hybrid method was introduced; Base64 + AES on the prototype. The test systems were created by utilising virtual machines. The prototype was developed using C# language in Microsoft Visual Studio application that runs on Windows. To provide countermeasures, this research proposes an implementation of a third-party privacy application, called PRINDOW, to improve security in hiding a user's browsing activity. Every browsing session is recorded and scanned using the prototype. This method allows only the base requirements to be installed on the virtual machine for each file with the cryptographic method. This framework could theoretically enhance current practises by making slight changes to the web browser's application structure.***

Keywords: Browser Artefacts, Computer Forensic, Encryption, Data Hiding.

I. INTRODUCTION

It has been a common practise for web browsers to run on personal computers, whether for work, study or leisure purposes. Users utilise a web browser to conduct a range of tasks, such as checking e-mail, performing internet banking, looking for information, and much more. Therefore, web browsers have the ability to capture every piece of information related to user's activity performed on their machine. As for private browsing mode, it is found that browser doesn't delete all local data after each session of private browsing [1]. The purpose of private browsing is for users to browse private mode just as a standard browsing session would, but without storing any data such as log-in credentials or browsing history upon exit. While browser developers endeavour to keep users' browsing private as it should, most computer forensic tools have retrieved browser artefacts even if the user used private mode for the entire session. Therefore, this paper discusses on enhancing web browser security using a technique from Counter Digital Forensics (CDF) which comprises of a set of techniques and measures taken by those interested in the stagnation or interjection in the digital investigation process [2]. Not only are cyber criminals using the CDF concept, but also people involved in data security use it equally [3]. CDF's main concerns are that digital data may be modified, removed or rendered difficult by rendering it significant, redundant or almost impossible to analyse [4]. The idea is not new since many cyber criminals have been utilizing rootkits to exploit systems to disguise malware operations for years to come [5].

One part of data hiding is AES encryption is one of Rijndael's basic points as it is a symmetric block cipher algorithm. The 128-bit data blocks can then be interpreted using 128-bit, 192-bit and 256-bit encryption keys. AES needs very little RAM space and very quickly [6]. AES provides excellent Data Security.

II. RELATED WORK

The following research articles are selected for review, keeping in mind the traditional and conventional approaches of file cryptography and encryption in general.

A. Folder Lock

File lock authentication is reliable with 256-bit AES encryption. Upon encryption, the original data will be deleted, so no data remains intact, and software integrity will be preserved upon decryption.

It stops others from deleting user's records, whether deliberate or accidental. The user can choose whether to "lock" or "encrypt" files or directories with many context menu choices. Every solution guarantees data security. History cleaning will erase log history and simple clipboard data.

B. CCleaner

CCleaner is a freeware configuration, safety and cleaning application for user's device. It removes redundant data from the machine, enabling Windows

Revised Manuscript Received on April 15, 2020.
***** Correspondence Author
Mohd Fahmi Mohamad Amran*, Computer Science Department, Faculty of Science and Defense Technology, Universiti Pertahanan Nasional Malaysia. Email: fahmiamran@upnm.edu.my
Yamunah Kathiravan, Computer Science Department, Faculty of Science and Defense Technology, Universiti Pertahanan Nasional Malaysia.
Noor Afiza Mat Razali, Computer Science Department, Faculty of Science and Defense Technology, Universiti Pertahanan Nasional Malaysia.
Raja Mohd Tariqi Raja Lope Ahmad, Computer Science Department, Faculty of Science and Defense Technology, Universiti Pertahanan Nasional Malaysia.
Zuraidy Adnan, Computer Science Department, Faculty of Science and Defense Technology, Universiti Pertahanan Nasional Malaysia.
Muhammad Fairuz Abd Rauf, Computer Science Department, Faculty of Science and Defense Technology, Universiti Pertahanan Nasional Malaysia.
Mohd Afizi Mohd Shukran, Computer Science Department, Faculty of Science and Defense Technology, Universiti Pertahanan Nasional Malaysia.

to run more efficiently and free usable hard disk space.

Advertisers and websites actively track the online activity of users with cookies. Saved passwords, saved data, and Internet history allows user identity less secure. CCleaner removes these details to maintain the online experience secure, meaning that consumers are less likely to encounter information theft and/or cyber fraud.

III. PROPOSED SCHEME

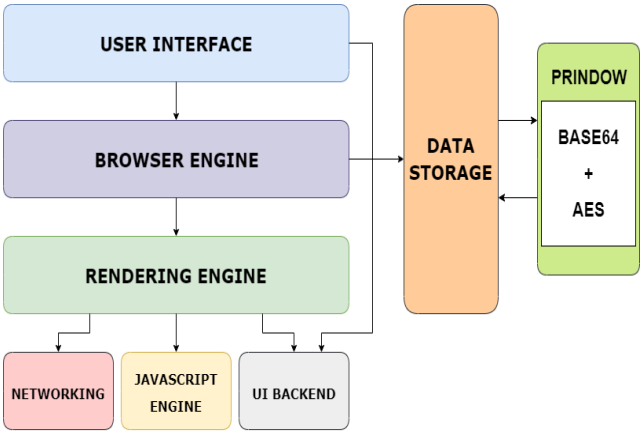

Figure 1 The "Secured Web Browser Artefacts with Encryption" Framework

Figure 1 shows the Web Browser Architecture Main Components with the added Proposed System; PRINDOW. PRINDOW is two-part system with combination of Base64 and AES. A more thorough flowchart in shown in Figure 2.

IV. PROPOSED FRAMEWORK

Current systems with AES implemented were all focused on complex mathematical functions to protect the privacy of the user. These systems will work well enough, i.e. removing user data but not complying with one or more requirements, including speed, trust, memory constraints, etc.

The framework proposed makes use of intermediate text files instead of system files which can be tricky to encrypt straight away [7]. This transforms the original SQLite file to a text file by minimising it to a Base64 version of the same text file. The text file is later sent to a single AES encryption to provide security. Compared to the original artefacts, the Base64 converted files is simple to encrypt. The overall web browser artefacts encryption process is as follows:

Step 1: The browser session exits upon request.
Step 2: PRINDOW scans the selected browsers and artefacts.
Step 3: Converts the browser artefacts to plaintext (Base64).
Step 4: Generate AES symmetric key.
Step 5: Encrypt the plaintext with AES symmetric key.
Step 6: Delete the final encrypted file along with the symmetric key

The browser artefacts will be converted to Base64 and then encrypted after writing on hard disk. This way, the data cannot be compromised by an outsider without the authority of the owner. The overall of the proposed system is shown in Figure 2.By default, each piece of artefacts that the browser submits is encrypted with a unique key, and the user's browsing data is generated random key at every use, allowing the browser to encrypt all the artefacts each time with different keys. This helps the user to perform better without being bound to a single key, and is more secure. Random symmetric key for each message is generated and that symmetric key is used to encrypt the browser artefacts.

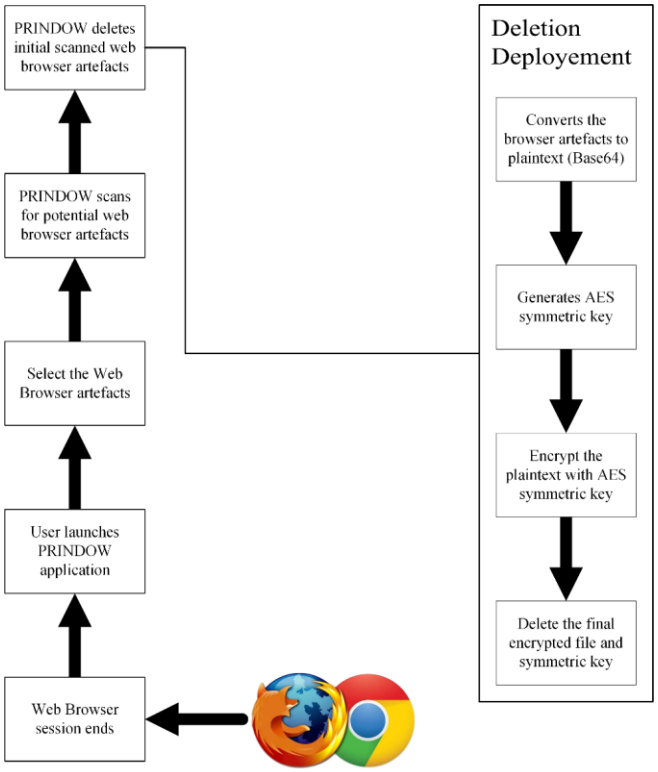

Figure 2 The Flowchart of PRINDOW

A stimulated software has been developed with C # code to incorporate a realistic encapsulation framework with each specific understudy and assess its efficacy on these aspects. A set of artefacts for investigations uses a comprehensive and accurate collection of produced web browser data. A common PC with 8GB RAM and Intel i7 2.3GHz was used to host the test program. The main interface of the prototype is as shown in Figure 3.

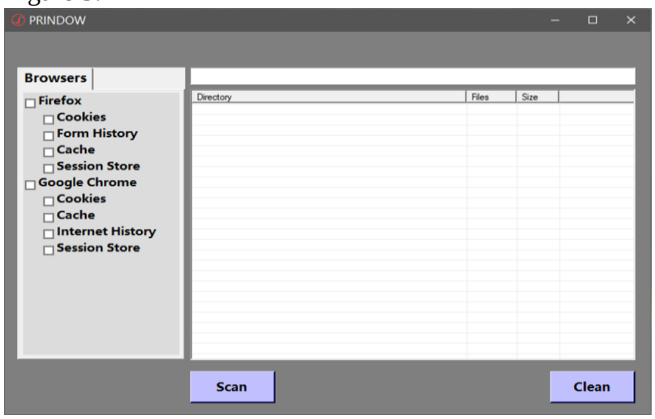

Figure 3 PRINDOW Interface

V. STRUCTURE OF PRINDOW

PRINDOW, a hybrid program of encoding and encryption algorithm built with a purpose of making the web browser artefacts secured. This hybrid prototype includes an implementation of Base64 and AES 128. Base64 involvement was to transform the browser artefacts into text files before encryption takes place. The converted Base64 files encrypt really quick as compared to original web browser artefacts. A thorough explanation is discussed in the following:

A. Base64 Encoding

This section will analyse the Base64 algorithm. The plaintext will be encoded into ciphertext. Let's assume the plaintext is "PRINDOW". It has seven characters. The calculation below shows the explanation of Base64 example.

Index	1	2	3	4	5	6	7
Char	P	R	I	N	D	O	W
Decimal	80	82	73	78	68	79	87

The decimals are later converted to binaries. The concatenation of the binaries is

01010000 01010010 0100100101001110 01000100 01001111 01010111

Table 1 shows the final binary bit of the previous example. The amount of plaintext is seven characters. The total bit is 7 x 8 bits = 56 bits. The 56-bits is divided into ten parts of 6-bits characters. There will be ten characters in Base64 format. The ciphertext is **UFJJTkRPVw** based on Base64 character table.

Table 1 Binary 6-Bit

Index	Binary 6-bit	Char
1	010100	20
2	000101	5
3	001001	9
4	001001	9
5	010011	19
6	100100	36
7	010001	17
8	001111	15
9	010101	21
10	110000	48

B. AES Encryption

Now convert the Base64 ciphertext **UFJJTkPVw** to hexadecimal form which will translate to the following:

Plaintext : **UFJJTkPVw**
Hexadecimal : **55 46 4A 4A 54 6B 50 56 77**

Key size used will 128-bits. Consists of 10 rounds with 16-byte key size. Input key is used as the first key, then a key is generated for each of the next 10 rounds.

Let's give the key as "You stay private" and translate it to hexadecimal.

Key : **You stay private**
Hexadecimal : **59 6F 75 20 73 74 61 79 20 70 72 69 76 61 74 65**

C. Key Schedule

Key Words

```
w0  = 59 6f 75 20
w1  = 73 74 61 79
w2  = 20 70 72 69
w3  = 76 61 74 65

w4  = w0 ⊕ z1 = w1 = B7 FD 38 18
w5  = w4 ⊕ w1 = w2 = C4 89 59 61
w6  = w5 ⊕ w2 = w3 = E4 F9 2B 08
w7  = w6 ⊕ w3 = 92 98 5F 6D

w8  = w4 ⊕ z2 = w5 = F3 32 04 57
w9  = w8 ⊕ w5 = w6 = 37 BB 5D 36
w10 = w9 ⊕ w6 = w7 = D3 42 76 3E
w11 = w10 ⊕ w7 = 41 DA 29 53
.
.
.
.
.
w36 = w12 ⊕ z9 = w33 = A5 50 09 62
w37 = w36 ⊕ w33 = w34 = 23 3A AC 55
w38 = w37 ⊕ w34 = w35 = 80 18 1A 3C
w39 = w38 ⊕ w35 = 2B 17 46 02

w40 = w36 ⊕ z10 = w37 = 63 0A 7E 93
w41 = w40 ⊕ w37 = w38 = 40 30 D2 C6
w42 = w41 ⊕ w38 = C0 28 C8 FA
w43 = w42 ⊕ w39 = EB 3F 8E F8
```

All RoundKeys

Round 0 : **59 6F 75 20 73 74 61 79 20 70 72 69 76 61 74 65**
Round 1 : **B7 FD 38 18 C4 89 59 61 E4 F9 2B 08 92 98 5F 6D**
Round 2 : **F3 32 04 57 37 BB 5D 36 D3 42 76 3E 41 DA 29 53**
Round 3 : **A0 97 E9 D4 97 2C B4 E2 44 6E C2 DC 05 B4 EB 8F**
Round 4 : **25 7E 9A BF B2 52 2E 5D F6 3C EC 81 F3 88 07 0E**
Round 5 : **F1 BB 31 B2 43 E9 1F EF B5 D5 F3 6E 46 5D F4 60**
Round 6 : **9D 04 E1 E8 DE ED FE 07 6B 38 0D 69 2D 65 F9 09**
Round 7 : **90 9D E0 30 4E 70 1E 37 25 48 13 5E 08 2D EA 57**
Round 8 : **C8 1A BB 00 86 6A A5 37 A3 22 B6 69 AB 0F 5C 3E**
Round 9 : **A5 50 09 62 23 3A AC 55 80 18 1A 3C 2B 17 46 02**
Round 10 : **63 0A 7E 93 40 30 D2 C6 C0 28 C8 FA EB 3F 8E F8**

The results for each round are displayed in Table 2.

Table 2 Results based on each round in AES 128-bit

Round	Results
0	RoundKeys: 0C 29 3F 6A 27 1F 31 2F 57 70 72 69 76 61 74 65
1	SubBytes: FE A5 75 02 CC C0 C7 15 5B 51 40 F9 38 EF 92 4D ShiftRows: FE C0 40 4D CC 51 92 02 5B EF 75 15 38 A5 C7 F9 MixColumns: B1 E8 69 03 E0 C1 A4 88 FC 14 61 5D BA C2 18 C3 XOR RoundKey: 06 15 51 1B 24 48 FD E9 18 ED 4A 55 28 5A 47 AE
2	SubBytes: 6F 59 D1 AF 36 52 54 1E AD 55 D6 FC 34 BE A0 E4 ShiftRows: 6F 52 D6 E4 36 55 A0 AF AD BE D1 1E 34 59 54 FC MixColumns: 1A 4E BD E6 9C C8 D2 EA 57 BC 88 BF 2B 86 DA B2 XOR RoundKey: E9 7C B9 B1 AB 73 8F DC 84 FE FE 81 6A 5C F3 E1
3	SubBytes: 1E 10 56 C8 62 8F 73 86 5F BB BB 0C 02 4A 0D F8 ShiftRows: 1E 8F BB F8 62 BB 0D C8 5F 4A 56 86 02 10 73 0C MixColumns: F5 35 EF FD D7 D0 80 9B B0 B7 28 EA 4B BB E0 7D XOR RoundKey: 55 A2 06 29 40 FC 34 79 F4 D9 EA 36 4E 0F 0B F2
4	SubBytes: FC 3A 6F A5 09 B0 18 B6 BF 35 87 05 2F 76 2B 89 ShiftRows: FC B0 87 89 09 35 2B A5 BF 76 6F B6 2F 3A 18 05 MixColumns: 26 9C D9 21 C3 BB 9E 54 26 54 D6 B4 0D 76 2A 59 XOR RoundKey: 03 E2 43 9E 71 E9 B0 09 D0 68 3A 35 FE FE 2D 57
5	SubBytes: 7B 98 1A 0B A3 1E E7 01 70 45 80 96 BB BB D8 5B ShiftRows: 7B 1E 80 5B A3 45 D8 0B 70 BB 1A 01 BB 98 E7 96 MixColumns: 0F 87 93 A5 41 51 50 75 2D 32 FC 33 AF 34 57 9E XOR RoundKey: FE 3C A2 17 02 B8 4F 9A 98 E7 0F 5D E9 69 A3 FE
6	SubBytes: BB EB 3A F0 77 6C 84 B8 46 94 76 4C 1E F9 0A BB ShiftRows: BB 6C 76 BB 77 94 0A F0 46 F9 3A B8 1E EB 84 4C MixColumns: 14 42 ED A1 B3 AA FC FC 1E 59 18 62 D2 08 32 D5 XOR RoundKey: 89 46 0C 49 6D 47 02 FB 75 61 15 0B FF 6D CB DC
7	SubBytes: A7 5A FE 3B 3C A0 77 0F 9D EF 59 2B 16 3C 1F 86 ShiftRows: A7 A0 59 86 3C EF 1F 3B 9D 3C FE 0F 16 5A 77 2B MixColumns: 71 91 24 1C 76 E3 A0 C2 94 F3 57 60 9E 10 DF 41 XOR RoundKey: E1 0C C4 2C 38 93 BE F5 B1 BB 44 3E 96 3D 35 16
8	SubBytes: F8 FE 1C 71 07 DC AE E6 C8 EA 1B B2 90 27 96 47 ShiftRows: F8 DC 1B 47 07 EA 96 71 C8 27 1C E6 90 FE AE B2 MixColumns: C8 31 DB 5A CC 18 49 97 18 44 E6 AF 3E 2C E4 84 XOR RoundKey: 00 2B 60 5A 4A 72 EC A0 BB 66 50 C6 95 23 B8 BA
9	SubBytes: 63 F1 D0 BE D6 40 CE E0 EA 33 53 B4 2A 26 6C F4 ShiftRows: 63 40 53 F4 D6 33 6C BE EA 26 D0 E0 2A F1 CE B4 MixColumns: A1 E2 82 45 30 BA E4 59 95 2D 4C 08 26 2E 9B 32 XOR RoundKey: 04 B2 8B 27 13 80 48 0C 15 35 56 34 0D 39 DD 30

10	SubBytes: F2 37 3D CC 7D CD 52 FE 59 96 B1 18 D7 12 C1 04 ShiftRows: F2 CD B1 04 7D 96 C1 CC 59 12 3D FE D7 37 52 18 XOR RoundKey: 91 C7 CF 97 3D A6 13 0A 99 3A F5 04 3C 08 DC E0

Ciphertext : **91 C7 CF 97 3D A6 13 0A 99 3A F5 04 3C 08 DC E0**

VI. CONCLUSION

This main objective of this paper is to develop a new technique of securing browser data written on the local machine to fulfil user privacy which many major browsers failed to so. This paper gives an overview of the proposed framework for the prototype, which includes the structure of the prototype, as well as the operations. The hybrid prototype named PRINDOW, includes an implementation of Base64 and AES 128. Base64 involvement was to transform the browser artefacts into text files before encryption takes place. The converted Base64 files encrypt really quick as compared to original web browser artefacts. This prototype therefore answers the increased demands by users for the lack of security faced in private browsing.

ACKNOWLEDGMENT

The authors honourably appreciate National Defence University of Malaysia for the financial sponsorship and continuous support.

REFERENCES

1. Bunting, S., & Wei, W. (2006). EnCase Computer Forensics: The Official EnCE: EnCase Certified Examiner Study Guide: Wiley.
2. Kwak, J., Kim, H. C., Park, I. H., & Song, Y. H. (2016). Anti-Forensic Deletion Scheme for Flash Storage Systems.
3. Kessler, G. C. (2006). Anti-Forensics and the Digital Investigator. Proceedings of the 2014 47th Hawaii International Conference on System Sciences, 1–7.
4. Dewald, A. (2015). Characteristic Evidence, Counter Evidence and Reconstruction Problems in Forensic Computing. Proceedings - 9th International Conference on IT Security Incident Management and IT Forensics, IMF 2015, 77–82.
5. Rao, S. (2013). Hacktivism Trends, Digital Forensic Tools and Challenges: A Survey, (Ict), 138–144.
6. Ramesh, G., & Umarani, R. (2012). UR5: A Novel Symmetrical Encryption Algorithm with Fast Flexible and High Security Based on Key Updation. European, 77(2), 275–292.
7. Iyer, S. C., Sedamkar, R. R., & Gupta, S. (2016). A novel idea of video encryption using hybrid cryptographic techniques. Proceedings of the International Conference on Inventive Computation Technologies, ICICT 2016, 2016.