A Novel Principle to Validate Digital Forensic Models

Dinesh Mothi[a], Helge Janicke[a] and Isabel Wagner[a]

[a]School of Computer Science and Informatics, De Montfort University, Leicester, UK

ARTICLE INFO

Keywords:
digital forensic models
validation principle
anti-forensics

ABSTRACT

Digital forensic models (DFMs) form the base for any digital investigation because they guide the investigators with necessary steps and procedures to be taken during the investigation. State-of-the-art DFMs assume that it is safe to proceed from one stage of the investigation to the next without taking into account the anti-forensic techniques that could be used to defeat the investigation process. However, the findings in the literature shows that common phases in the digital forensic process such as acquisition, examination, analysis, and reporting are affected by various anti-forensic (AF) methods. To fill this gap, we propose an abstract digital forensic framework and validate DFMs by factoring in AF techniques affecting various phases in a digital forensic process. This validation principle can be used to enhance state-of-the-art DFMs to enable principled detection and countering of AF techniques before being applied to a real-time investigation case.

1. Introduction

Digital forensic science is defined by Palmer (2017) as "the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations". Based on this definition of digital forensics, numerous digital forensic models (DFMs) have been proposed over the years that suit different investigative methodologies and organisational policies. Due to a significant number of DFMs being proposed, efforts to standardise these DFMs have been made by Reith et al. (2002),Valjarevic and Venter (2012), and Kohn et al. (2013). ‘ In recent years, this led to creating Standard Operating Procedures (SOPs) alongside DFMs. The Scientific Working Group on Digital Evidence (2019) (SWGDE) produced its SOP for computer forensics. Even law enforcement has been designing and implementing its own SOPs in the form of guidelines Ashcroft et al. (2019) Williams (2012). One of the challenges of digital forensics is anti-forensics. Anti-Forensics has the ability to delay the investigation or to prevent the investigation from taking place by deleting evidence. This poses a threat to the investigation process if proper measures are not taken as argued by Lindsey (2017). The state-of-the-art DFMs do not appear to take anti-forensic methodologies or techniques into account. To address this gap, in this paper we propose an abstracted framework, and a principle to validate DFMs with the help of which an investigator can validate a DFM by confirming anti-forensic attacks that defeat the forensic process. The main contribution in this paper is deducting a mathematical principle for validating digital forensic models for confirming the detection and countering anti-forensic techniques.The significance of this contribution is that investigators could use validation principle to confirm whether or not their digital forensic models or frameworks are under an influence or being affected by an anti-forensic attack. Therefore, letting the investigator know if their investigative model or framework is valid for the purpose of conducting an investigation. The paper is structured as follows: section 2 reviews digital forensic models and anti-forensic attacks on various phases of the digital forensic process. In section 3 an abstracted digital forensic framework is proposed and the importance along with the relationship between the four abstract layers DFM validation, DF tool validation, DF method validation, and legislation conformation is established. In section 4 we justify why is it necessary to validate DFMs. In Section 5 the validation principle is proposed, defined, explained, and proved with the help of propositions and theorems. Then in section 6 the validation principle is evaluated by showcasing how it can be applied to DFMs to validate or invalidate them. Section 7 concludes the paper along with future scope.

ORCID(s):

2. Related Work

In this section we review digital forensic models, and anti-forensic attacks affecting various phases of the digital forensic process. The findings show that only two models in the literature consider tackling anti-forensic techniques into their DFM. Also, the anti-forensic techniques affecting various phases on the digital forensic process is shown in table 1.

2.1. Review of Digital Forensic Models

The generic computer forensic investigative model (GCFIM) developed by Yusoff et al. (2011) groups common computer forensic phases from previous DFMs. It consists of five phases: pre-process, acquisition, analysis, presentation, and post-process. The authors conclude by stating that their model can serve as a high-level computer forensic investigation model and also could assist in creating new computer forensic investigative methodologies. Systematic Digital Forensic Investigation Model (SRDFIM) was proposed by Agarwal et al. (2011) by comparing and expanding over previous computer forensic models. SRDFIM consists of eleven phases namely preparation, securing the scene, sur-

vey and recognition, documenting the scene, communication shielding, evidence collection, preservation, examination, analysis, presentation, and result and review. This model helps in reconstructing events by realizing certain properties such as individuality, repeatability, reliability, performance, testability, scalability, quality and standards in analysis pertaining to computer frauds and cyber crimes. Whereas, Soltani and Seno (2019) design their event reconstruction model using formal methods such as temporal logic which is an automatic verification technique, and is evaluated on the file allocation table (FAT) file system. Eleven digital forensic models was assessed and evaluated by Montasari (2016) using the five criteria that is set in the Daubert Standard, a standard that is used to accept scientific evidence in the United States. None of the eleven models could satisfy all the conditions of the Daubert Standard as argued by Meyers and Rogers (2005). It was deduced that, Ciardhuáin et al. (2009) and Rogers et al. (2006) took the most scientific approach to develop their digital forensic models. Argument is also made that the DFMs did not include the full scope of the investigation, and only concentrate on a few aspects of the digital forensic investigation. The DFMs were based on personal experience and no model can be considered as a standard one, and the error rate could not be calculated. The existing models are not flexible, that is, they are not generalised and cannot be applied to different domains of digital forensics as pointed out by Montasari (2016).

During the last decade specific DFMs developed by Hitchcock et al. (2016), Kaur et al. (2018), Ali et al. (2017), Zia et al. (2017), and Lutui (2016) targeting certain areas of digital forensics such as digital triage, network forensics, mobile forensics, and Internet of Things (IoT) forensics respectively. To increase the effectiveness and efficiency of IoT investigations Oriwoh et al. (2013) propose 1-2-3 Zones in conjunction with Next-Best-Thing Triage (NBT) model where necessary by maximising the utilisation of time to ensure relevant evidence is identified and acquired. The NBT model and 1-2-3 zones was adopted by Harbawi and Varol (2017) into their digital forensic procedure flowchart and argue that their theoretical framework for IoT investigations is improved and copes with evidence acquisition issues, and to analyse potential digital evidence in an IoT ecosystem, Kebande et al. (2018) proposes an integrated digital forensic framework for IOT devices.

Various researchers also adopted International Standard Organization (ISO) standards into their digital forensic framework to either enhance or hope to standardise their framework. To improve speed and quality of investigations Kao and Wu (2016) propose a digital triage forensics framework of windows malware forensic toolkit based on ISO/IEC 27037, which provides guidelines with respect to common scenarios encountered during the digital evidence handling process and assists organisations in their methods and procedures, and also facilitates the exchange of digital evidence between various jurisdictions. A forensic-by-design framework to enhance the framework for digital forensic investigations pertaining to cyber-physical cloud system that aids organisations to recover from a cyber-physical attack based on ISO 27043:2015 was developed by and this provides guidelines on various investigation scenarios involving digital evidence. Similarly Kigwana et al. (2017) propose a digital forensic investigative framework which is also based on ISO/IEC 27043:2015 to develop a standard eGov forensic investigation procedure, and Karie et al. (2019) argue that key factors such as blockchain should be added to ISO/IEC 27043:2015 to support standardised digital forensic report generation process .

Only two frameworks were found in the literature that consider to detect anti-forensic attacks. A digital forensic process consisting of five phases was proposed by Rekhis and Boudriga (2012a) that takes detection of anti-forensic attacks into consideration. However, their framework can detect anti-forensic attacks only in the analysis phase of the digital forensic process. Their framework is supported by several complex propositions to develop hierarchical visibility theory to detect anti-forensic attacks as proposed in Rekhis and Boudriga (2012b). A case study is also provided explaining how their propositions are applicable to a case where an administrative account has been compromised, but they fail to explicitly show how their theory or proposition can detect an anti-forensic attack.

Another framework to detect anti-forensics attack in a cloud environment is proposed by Rani and Kumari (2017). Their framework is based on and is similar to Rekhis and Boudriga (2012a) framework to detect anti-forensic attacks. It consists of six phases, and the authors only mention to use attack graphs in order to detect the AF attacks, but do not explicitly show how graph theory or attack graphs can be used to detect AF attacks in a cloud investigative environment. In their framework also, the anti-forensic attacks are proposed to be detected in the analysis phase, but by reviewing the literature, we found that anti-forensic attacks can affect not only the analysis phase but also various important phases in a digital forensic framework such as collection, examination, and reporting as shown in Table 1.

After reviewing various DFMs to date it does not appear that authors have taken measures to consider or detect various anti-forensic methods that could affect various phases in the digital forensic process except for Rekhis and Boudriga (2012a) and Rani and Kumari (2017). These frameworks take only the analysis phase into consideration, but they do not consider other phases of the digital forensic process that could be affected by anti-forensic attacks as shown in table 1. In this paper, we address these gaps by proposing a novel validation principle that helps to ensure that DFM phases have not been affected by AF attacks.

2.2. Anti-Forensic Methods Affecting DF Phases

In this section we review anti-forensic methods to find which anti-forensic methods affect which phases in the digital forensic process. We focus on the four most common phases in the digital forensic process: Acquisition, Examination, Analysis, and Reporting.

2.2.1. Acquisition

When the investigator decides to acquire evidence off a machine they would follow certain procedures depending on the state of the machine i.e., whether it is ON (live acquisition) or the OFF state (dead acquisition). There are anti-forensic techniques that could defeat either of the acquisition procedures. For example, during the acquisition of memory if the drivers used in a forensic tool is dependent on the 'KDBG' string to resolve symbols, then this method could interrupt the memory acquisition process. Also, if the undocumented memory enumeration application peripheral interface (API) and memory mapping API such as MmGetPhysicalMemoryRanges() and MmMapMemoryDumpMdl() are patched to return a NULL value then this would return a modified version of the memory map thus resulting in incomplete acquisition of the memory as argued by Stüttgen and Cohen (2013). Malware researchers and investigators employ API call hooking techniques to study and know the behaviour of malware, but Shaid and Maarof (2015) found that this method could be prove to be futile as a malware can detect API calls and then evidence could be manipulated. Another approach undertaken by Zhang et al. (2018) is by implementing an anti-forensic technique known as Hidden in I/O Space (HIveS) and malicious enclave software to tamper with the acquisition process to prevent memory analysis. This technique is designed to operate outside the scope of the operating system and therefore making it harder to detect its presence in the memory. Even hard drive acquisition can be defeated if the attacker adopts techniques such as anti-forensics of data storage by alternative use of communication channels (AFAUC) [34]. In AFAUC, the storage device is accessed through its diagnostic interface to hide or even obfuscate data such that it will not be accessible to the investigator. Moreover, this hidden data will also be absent in hidden areas such as host protected area and device configuration overlay of the hard drive which the investigator might look into before making a forensic image of the device. Hence, defeating the forensic acquisition process.

2.2.2. Examination

This phase of the forensic process can be mainly defeated by encrypting the whole storage media or just certain partitions within the storage media. Or, even applying steganography techniques such as hiding a file system within a file system which can be accomplished by the encryption tool TrueCrypt. This is a useful technique for an attacker because they can disclose the encryption key or passphrase for the first encrypted file system, but not for the other hidden filesystem which could result in the investigators being unaware of the hidden filesystem. Due to this clever technique the suspects in the UK would be able to comply with section 49 of the Regulation of Investigatory Powers Act 2000 (RIPA 2000), but they would be able to avoid punishment mentioned in section 53 of the RIPA, thus deceiving the investigation and the legislation used to tackle encryption key issues. Other ways of hiding a file to prevent examination as mentioned by Dahbur and Mohammad (2011) is to manipulate certain registry key, hiding data in the HPA and DCO areas of the hard drive, or even using bootable USBs or DVDs, and compression bombs such as 42.zip.

2.2.3. Analysis

During the analysis phase the investigation certain files or areas of the storage device will be looked into to find relevant information pertaining to the case. This process can be defeated or prolonged depending on the anti-forensic technique. A kernel rootkit disk filter driver known as 'Ddefy.sys' can be used to hide a file from a New Technology File System (NTFS) filesystem. The driver can locate the filename, its directory entry position, the clusters containing data, and its disk position and with this information the data can be hidden. The same rootkit can also defeat memory analysis by performing a system service dispatch table (SSDT) hook which can be used to modify a process and thread list Bilby. Also, encryption of certain files with strong passphrases will consume lot of the investigation time, steganography techniques of hiding a file within a file may cause the investigator to overlook certain important files. Tools such as metasploit's slacker which can be used to delete data in the slack space which would make it impossible to recover deleted data, and metasploit's transmogrify can be used to change metadata such as modified, accessed, created parameters of a file thus defeating timeline analysis as argued by Garfinkel (2006), Dahbur and Mohammad (2011), and Gül and Kugu (2017).

2.2.4. Reporting

It is possible to inject malicious code into computer forensic tools (CFTs) to produce false forensic reports and also infect the computer. This kind of attack can be done by adopting hypertext markup language (HTML) code injection technique wherein a malware can be embedded into a report that can be used to attack web browsers. Now, when the report is viewed in to web browser the malware escapes the virtual environment and infects the machine which was demonstrated by Wundram et al. (2013).

After reviewing various anti-forensic techniques,the findings show that various phases in the digital forensic process are subjected to anti-forensic attacks as shown in Table 1.

3. Abstracted Digital Forensic Framework

In the literature of digital forensics (DF) we can find numerous digital forensic models as reviewed in section 2. One of the important aspects of DF research is tool testing. The National Institute of Standards and Technology (NIST) oversees this aspect of DF research and various DF tools are tested and then reported on their website National Institute of Standards and Technology. Then there is FSR-G-218 method validation in digital forensics in the United Kingdom (UK) and Daubert Standard in the United States (US) to validate digital forensic methods and procedures. We argue that DFMs, DF tool testing, DF Method validation, and the concerned legislation are inter-related and have their layer of abstraction. We propose a abstracted framework that shows

Collection	Examination	Analysis	Reporting
AFAUC	Data Hiding	Steganography	Code Injection
Kernel Debugger Block Hiding	Artefact Wiping	Data Manipulation	
Memory enumeration API Hooking	Cryptography	Code Injection	
Memory Mapping API hooking	Encryption	Adding known files	
Data Pooling	Data manipulation	String decoration	
	AFAUC	False Audit Trails	
	Compression Bombs	Hash Collisions	
	Denial of Service Attack	Loop References	
	ReDOS	Restricted Filenames	
	Transmogrification		

Table 1
AF techniques affecting various digital forensic phases

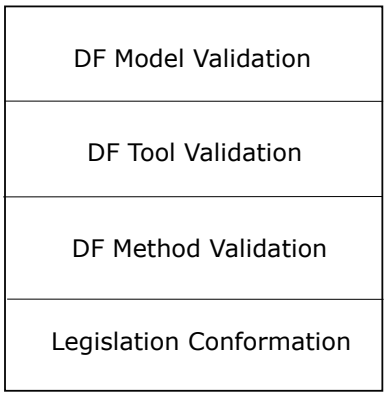

Figure 1: Levels of Abstraction for Testing Digital Forensic Models/Frameworks

the inter-relationship among the first three layers and their relationship to the fourth layer. The four layers of abstraction are described below:

3.1. DFM Validation

The first layer is the DFM. The investigation of digital crime is a step-by-step process. This process can be guided by a SOP or a DFM as to how to go by collecting, preserving, analysing and reporting digital evidence. Now, the procedure to collect digital evidence will be guided by a method. For example, during the collection phase if a computer is the ON state then specific procedures will be followed to gather relevant evidence Williams (2012). And when gathering digital evidence off a computer, this will entail a forensic software tool. During this process of collecting evidence, care will be taken to maintain the integrity of the evidence and this is done so as to conform to the legislation. Now, to maintain evidence integrity the method and the forensic tools used has to be validated. Method validation and tool validation are discussed below.

3.2. Tool Validation

Ensuring reliability of digital forensic tools is of prime importance to the forensic community. The National Institute of Standards and Technology (NIST) in the United States (US) set up the Computer Forensic Tool Testing Program (CFTT) for this purpose National Institute of Standards and Technology. The CFTT projects evaluates digital forensic tools by adopting functional conformance testing i.e., a set of requirements are established by developing a set of test assertions and cases. And set of setting procedures are used to perform tests. Over the years numerous digital forensic tools have been tested and documented on their project website. And of October 2017 onwards in the UK, the FSR in its code of practice made it mandatory that all providers of digital forensic services to the criminal justice system must be accredited to ISO-17025, which provides general requirements for the competence of testing and calibration laboratories.

3.3. Method Validation

In the US a method or procedure adopted by a investigator has to be adhered to the Daubert Standard, which provides a set of objective guidelines for judges to determine the admissibility of scientific evidence in the courts. The Daubert standard applies to those digital forensic methods or procedures that were used to uncover evidence from digital devices, and must satisfy the following criteria Meyers and Rogers (2005)

1. "Testing: Can and has the scientific procedure been independently tested? Peer Review: Has the scientific procedure been published and subject to peer review?"
2. "Error rate: Is there a known error rate, or potential to know the error rate, associated with the use of this scientific procedure?
3. Standards: Are there standards and protocols for the execution of the methodology?"
4. "Acceptance: Is the scientific procedure generally accepted by the relevant scientific community?"

In the United Kingdom (UK), in 2005 the House of Commons Science Technology Committee (2019) in the paragraph 173 mentioned that validation of scientific methods are absent before they are admitted in the court, and judges are incompetent to determine the validity of scientific evidence and they recommended Forensic Science Advisory Council which belongs to Forensic Science Regulator. (2019) to develop a test for scientific evidence which should build

on the Daubert Test. In 2016, the Forensic Science Regulator (FSR) has produced a guidance on method validation in digital forensics FSR-G-218 as seen on Forensic Science Regulator (2016). It amalgamates essential information from International Standards Organisation (2017), FSR-G-201 validation guidance found in Forensic Science Regulator (2014), Scientific Working Group on Digital Evidence (2019), and Criminal Practice Directions as mentioned in Courts and Tribunals Judiciary (2014).

3.4. Legislation Conformation

If the investigator's methods, procedures, tools, process are validated, then the chances of the evidence being accepted in the court increases. For example, if the investigation method fits or satisfies the criteria and objectives of the daubert standard then it would be accepted or admissible under the Federal Rules of Evidence (FRE) 702 and 902. Upcounsel.

In the abstraction layers the digital forensic tools and methods can be validated, but a procedure to validate DFMs is not present in the literature. In the next section, we justify why validating DFMs is important, and in section 5 we propose a principle to validate DFMs.

4. Justifying the Validation of DFMs

In section 6.1.2 of FSR-G-218 it is mentioned that risk assessment must be conducted before performing validation testing. Risk assessment in the Criminal Justice System (CJS) usually includes: "a. the risk of wrongful conviction(s); b. the risk of wrongful acquittal(s); and c. the risk of obstructing or delaying investigation(s)."

The aforementioned three circumstances can occur if one is not aware of the anti-forensic methodologies and techniques as pointed out by Dahbur and Mohammad (2011), and Garfinkel (2006). Apart from this, if the anti-forensic techniques/methodologies as shown in Table 1 is not taken into consideration into the risk assessement procedures of FSR-G-218 or any guidance document that depends on it, then this can pose a threat to the investigation as far as reliability of evidence is concerned which is specifically mentioned in the direction 19A.6 of the Criminal Practice Directions.

Previously, according to section 69 of Police and Criminal Evidence Act 1984 The National Archives (2019), it was mandatory to prove that the computer was functioning properly, and was not wrongly used to produce a document that could be admitted as evidence. This rule has been repealed by section 60 of the Youth Justice and Criminal Evidence Act 1999 The National Archives. (2019). And now, the computer evidence adheres to the common law rule that a presumption exists that the computer producing evidence was functioning properly at the time of investigation. Therefore, the evidence is admissible in the court of law. However, this presumption can be rebutted if either of the parties (prosecution or defense) adduce evidence to the contrary. Digital Forensic Investigators often use forensic software tools to generate forensic reports. Wundram et al. (2013) demonstrated an anti-forensic technique that attacks a forensic tool to generate false forensic reports. If either of the party proves that the investigation was compromised, then according to rebuttable presumption the evidence might be adjudged inadmissible in the court of law. Most of the digital forensic models in section 2 or the FSR-G-218 guidance on method validation in digital forensics do not address anti-forensic issues. In this paper, we eliminate the aforementioned gaps by proposing a principle for validating DFMs.

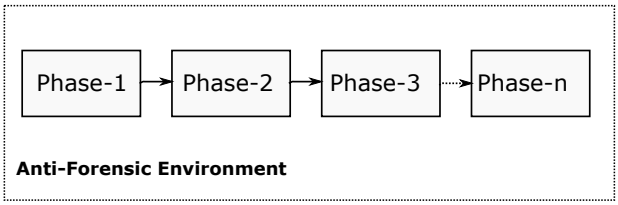

Figure 2: n-phase DFM in an AF environment

5. Validating Digital Forensic Models

In this section, we propose a principle for validating DFMs by counteracting anti-forensic techniques that affects each phase in the digital forensic process. The logic behind this principle is that for a DFM to be validated, every phase in the DFM must be validated before proceeding to the next phase, and when all the phases are validated then it can be concluded that a DFM is validated. For a DFM phase to be validated, the anti-forensic techniques in each and every phase of the digital forensic process must be accounted for, i.e., the AF techniques should be detected and countered. The process of detecting and countermeasuring anti-forensic (AF) techniques in each phase of the digital forensic process is the essence of the validation principle and is formalized using the concept of tensor products.

Consider a DFM with n phases in an anti-forensic environment as shown in Figure 2.

According to the validation principle, firstly individual phases need to be validated in their own AF environment. Now consider Phase 1 in Figure 2 in its anti-forensic environment. Its AF environment is defined by its respective AF techniques, for example those shown in Table 1. The first and the foremost step is to detect if an anti-forensic technique or method is acting upon or affecting a phase in the digital forensic process or not. To mathematically express this process certain definitions and axioms are first proposed, and then propositions are proved with the help of these definitions and axioms. Mathematical notations are presented in table 2.

Let the anti-forensic methods or techniques (A) acting upon a DF phase be represented by the vector $A = \{a_1, a_2, a_3, \dots, a_n\}$ such that $a_i \in A$, and the detection methods be represented as $D = \{d_1, d_2, d_3, \dots, d_n\}$ such that $d_i \in D$.

Definition 5.1. *The logical detector product (D_M) is defined*

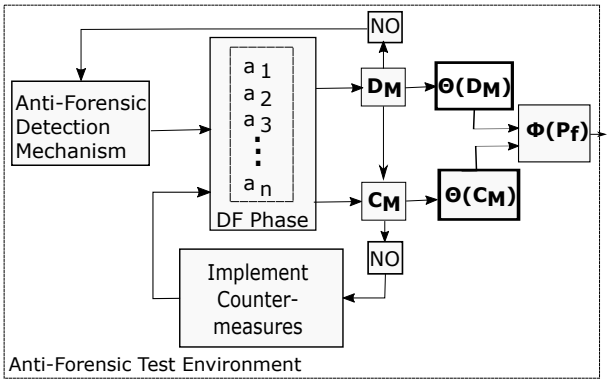

Figure 3: DF Phase Validation

as $\{a_j.d_j \in D_M | D_M = a_j \wedge {d_j}^T\}$*, which means an anti-forensic technique* ($a_j \in A$) *can be detected in 'n' unique number of ways using elements in D. This is analogous to inputting the individual vectors* $\mathbf{a}$ *and* $\mathbf{d}$ *to logical and-gates.*

Definition 5.2. *The counter tensor product* (C_M) *is defined as* $\{e_j.c_j \in C_M | C_M = e_j.{c_j}^T\}$*. The significance of* D_M *and* C_M *as shown in Figure 3 is that it shows us if an anti-forensic technique is detected or countered i.e., if the logical and of* $a_j.d_j = 1$ *then* a_j *is detected. Likewise,* a_j *is countered if* $e_j.c_j = 1$.

5.1. Vector Transformation Operators

The vector transformation operators play an important role in transforming D_M and C_M. If $m_{ij} \in \mathbb{R}^{p\times q}$ is a Matrix M, then $\Theta^n_{r,k}(M_{p\times q}) \longmapsto N_{p\times 1}$, which means $\Theta^n_{r,k}$ on $M_{p\times q}$ yields $n_{ij} \in \mathbb{R}^{p\times 1}$ matrix N a column vector of order $p \times 1$. Here, the subscript r means the transformation is only being applied to all the rows in M, and 'k' and 'n' means the transformation is from row 'k' to row 'n'. If only a specific row 'i' is considered in M, then the row vector transformation on row j in M is defined as or-ing of all elements in row j, i.e., $\Theta_{r,i}(R_i) = \mathbb{V}^n_{i=1}(r_i)$, and this operation yields a singleton set by transforming the row vector R_i of order $1 \times q$ to a - matrix of order 1×1. Now, if the row vector transformation is applied to all rows in M, beginning from the first row to the nth row, then $\Theta^n_{r,1} = \{\Theta_{r,1}(R_1), \Theta_{r,2}(R_2), \ldots, \Theta_{r,n}(R_n)\}$, and this transformation yields a matrix N of order p×1, i.e., $\Theta^n_{r,1}(M_{p\times q}) = N_{p\times 1}$. Similarly, $\Omega^n_{r,k}(M) = \wedge^n_{i=1}(r_i)$ and $\mathcal{Z}^n_{r,k}(M) = \Sigma^n_{i=1}(r_i)$ are logical ς, and summation over each row in Matrix M respectively.

5.2. Formalization of DFM Validation

We now formalize the notions of detecting AF attacks in DFM phases (Propositions 5.1 and 5.2) and of validating DFMs (Theorem 5.1). For in-depth mathematical proofs, please refer to appendix A. Before moving onto explaining the propositions, it is important to understand the function of the elements in Figure 3, which is the diagrammatic representation of DFM phase validation. Now, if a phase in a DFM is considered, say the first phase, then this phase must

Table 2
Mathematical notations

Maths Notation	Meaning
D_M	Detectability Matrix
C_M	Counterability Matrix
P_f	Phase Function
$\langle a_j.d_i \mid a_j \times d_i \rangle$	Dot Product or Cross Product
$\Theta^n_{r,1}(M)$	Vector–Or Transformation
$\Omega^n_{r,1}(M)$	Vector–And Transformation
${}_{r,1}\Phi^n_{c(r,1)}(M)$	Vector And–Or Transformation

be accounted for anti-forensic attacks. This is done with the help of the elements D_M and C_M. In order to detect AF attacks, the AF detection mechanism may consist of numerous techniques or methods to detect a single or numerous AF technique(s). These detection methods and AF techniques form a matrix (D_M). And when certain operations are applied to D_M, certain results can be deduced from those operations, such as, when do AF techniques are said to be detected, not detected, countered or not countered. This operation is mathematically formalised in proposition 5.1 and is proved in appendix A.1.

Proposition 5.1. *(a) If in a DFM phase,* $\Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [k]_{1\times 1}$*,* $k \in \mathbf{B}$*, and if k=1 then the AF techniques* $a \in A$ *are detected, and if k=0, then A is undetected.*
(b) If $\Omega_{c,1}[\Theta^n_{r,1}(C_M)] = [k]_{1\times 1}$ *then the detected AF technique* $a \in A$ *is countered if k=1, and is not countered if k=0.*

The purpose of D_M and C_M is not only to notify whether AF techniques are detected and countered as proved in proposition 5.1, but also one can specifically pin-point as to which detection method(s) has detected an AF attack(s). This is stated in proposition 5.2 and is proved in appendix A.2.

Proposition 5.2. *If* $[\Theta_{r,i}(a_j.d_i)] = [1]_{1\times 1}$ *then* a_j *[* $\pi^{N-1-i}_{i=0}(\bar{S}_i)$ S_{N-i} $\pi^{i}_{k=0}(\bar{S_{N-K}})$ S_{N-i}*] gives the element* $d_i \in R_i$ *that detects* a_j*.*

Now. after proposition 5.1 is satisfied it is imperative to calculate the pass function P_f, which indicates to the investigator whether the phase has been validated, and hence the process can be proceeded to the next stage. P_f addresses the issue of assuming that it is safe to proceed from one phase of the DFM without considering the effects of AF as seen in the finding of the review in Section 2. And, when all phases have been validated, then the validation principle is satisfied. The validation principle is stated in theorem 5.1 and is proved in appendix A.

Theorem 5.1. *If (a)* $\{\forall P_i \in P_n | P_i = 1 \vee {}_{r,1}\Phi^n_{c(r,1)}(P_n) = 1\}$ *then the DFM is validated.*
(b) $\{\exists P_i \in P_n | P_i = 0 \vee {}_{r,1}\Phi^n_{c(r,1)}(P_n) = 0\}$ *then the DFM is invalidated.*

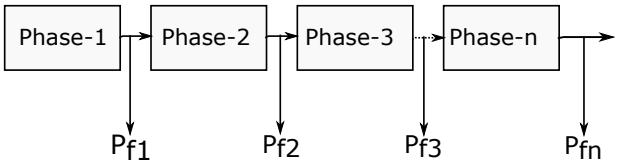

Figure 4: DF Phase Validation

6. Evaluation

This section demonstrates the feasibility of the validation principle by evaluating three DFMs. The first DFM is based on a hypothetical scenario to show a DFM that is successfully validated, and the other two DFMs selected from the literature are Rekhis and Boudriga (2012b) Rani and Kumari (2017), to prove how these DFMs fail validation using the validation principle. All DFMs are evaluated based on the validation principle as mentioned in section 5. Firstly, a hypothetical situation is explained, and then the evaluation process of the DFM phases is processed in four steps: 1) The AF column vector **a** and the detection row vector **d** are determined. 2) The vector-or transformation $\Theta(D_M)$ is computed to detect anti-forensic techniques in a DF phase. 3) The vector-or transformation $\Theta(C_M)$ is computed to counter anti-forensic techniques in a DF Phase. 4) Then finally, the phase function P_f is computed. And based on the value of P_f we validate or invalidate a DFM.

6.1. Validation Case

The application of this principle is shown by using a hypothetical situation where in a defendant is accused of DDoS attacking an organisation's network from their laptop, and the defendant claims that they're unaware of it and that their laptop might be hacked to perpetrate the crime. Now, let us assume the investigator follows a SOP and applies the principle to test for the validation of the DFM (before proceeding with the investigation and making a report to the court) which comprises of 4 phases collection, examination, analysis, and reporting. The objectives for each phase are as follows:

1. Collection: To collect or gather evidence from physical memory and hard drive.
2. Examination: To examine the content of the hard drive such as what is the size of the hard drive, volumes, hidden files, file types etc.
3. Analysis: To answer questions pertaining to investigation or support the investigative hypothesis/counter-hypothesis.
4. Report: To make a report of the findings.

To keep this case study simple, we assume that each phase in the DFM has two AF techniques a_1 and a_2 (except for reporting phase), and a_1 and a_2 have only d_1 and d_2 as their detecting methods and C_1 and C_2 as their countermeasures respectively.

6.1.1. Collection

During the collection phase the investigator uses a forensic tool to image the physical memory, and since the principle is being used the AF techniques in the collection phase (a_1 and a_2) will be looked up. Let us assume the AF techniques "Memory enumeration API hooking" and "Memory Mapping API hooking" is acting up on the collection phase, and also assume these two AF techniques can be detected and countered. The following four steps shows the application of the principle to validate the collection phase:

Step1: Determine column vectors **a** and **d**. The two AF techniques affecting the collection phase are:
a_1 = Memory enumeration API hooking
a_2 = Memory Mapping API hooking, and
$d_1 = d_2$ = A forensic tool that can detect a_1 and a_2.
Step 2: Compute the detector matrix once the vectors **a** and **d** are determined. By definition, $D_M = \begin{pmatrix} a_1.d_1 & a_2.d_2 \end{pmatrix}$. Since, for a_1 can be detected by d_1, and a_2 can be detected by d_2. Therefore, the logical *and* operation of $a_1.d_1 = 1$, and $a_2.d_2 = 1$. This implies, $D_M = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [1]_{1\times1}$

Step 3: Compute C_M once the AF techniques a_1 and a_2 have been detected. Now, a_1 and a_2 can be countered by, $c_1 = c_2$ = DumpIt, a memory acquisition tool resistant to a_1 and a_2, which means $ap_2.c_2$ = logical 1. And, by definition $C_M = \begin{pmatrix} ap_1.c_1 & ap_2.c_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(C_M)] = [1]_{1\times1}$

Step 4: Compute phase function (P_f) once the values of D_M and C_M have been determined. Therefore, $P_{f_{Collection}} = \begin{pmatrix} 1 & 1 \end{pmatrix} \Rightarrow \Omega_{c,1}[\Theta^n_{r,1}(P_1)] = 1$ the collection phase is validated since proposition 5.1 is satisfied, and the investigator can proceed to the next phase, Examination.

6.1.2. Examination

The process of validating the examination phase is shown in the following four steps:
Step1: Determine column vectors **a** and **d**. The two AF techniques affecting the examination phase are:
a_1 = Encryption
a_2 = Compression Bombs
d_1 = Forensic Software (example EnCase 8)
d_2 = Intelligent Decompression Libraries
Therefore, $a_1.d_1 = 1$ and $a_2.d_2 = 1$, since a_1 and a_2 are detected by d_1 and d_2
Step 2: After determining the values of **a** and **d**, compute the detector matrix D_M. By definition, $D_M = \begin{pmatrix} a_1.d_1 & a_2.d_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [1]_{1\times1}$
Step 3: Compute C_M once the AF techniques a_1 and a_2 have been detected, that is, D_M has been determined. Let's say a_1 can be countered by c_1 = Encryption Key, and $c_2 = 1$, since it has been detected in step 2 (see special case in section III). Therefore, $ap_1.c_1$ = logical 1 and $ap_2.c_2$ = logical 1. By definition, $C_M = \begin{pmatrix} ap_1.c_1 & ap_2.c_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(C_M)] = [1]_{1\times1}$
Step 4: Compute phase function (P_f) once D_M and C_M for the examination phase are determined. From steps 2 and 3, $P_{f_{Examination}} = [1\ 1] \Rightarrow \Omega_{c,1}[\Theta^n_{r,1}(P_2)] = 1$. Therefore, the

examination phase is validated since proposition 5.1 is satisfied, and the investigator can proceed to the next phase, Analysis.

6.1.3. Analysis

Step1: The validation of analysis phase is shown in the following four phases:
Step1: Determine vectors **a** and **d**. The two AF techniques affecting the analysis phase are:
a_1 = Hash Collisions
a_2 = Loop References
d_1 = linear probing
d_2 = file name resolving error
Therefore, $a_1.d_1 = 1$ and $a_2.d_2 = 1$, since a_1 and a_2 are detected by d_1 and d_2. Step 2: Compute the detector matrix D_M when the values of the vectors **a** and **d** have been determined. By definition, $D_M = \begin{pmatrix} a_1.d_1 & a_2.d_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [1]_{1\times 1}$.
Step 3: Compute C_M once the AF techniques a_1 and a_2 have been detected, that is, DM has been computed. By definition, $C_M = \mathrm{a}* \ C_M$. Now, **a** can be countered by $c_1 = c_2 = 1$, since it has been detected in step 2 (see special case in section). Therefore, $ap_1.c_1$ = logical 1 and $ap_2.c_2$ = logical 1. Therefore, $C_M = \begin{pmatrix} ap_1.c_1 & ap_2.c_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(C_M)] = {}_{c,1}\Phi^n_{c(r,1)}(P_3) = [1]_{1\times 1}$
Step 4: Compute phase function (P_f) once the values of D_M and C_M have been determined. From steps 2 and 3,, $P_{f_A nalysis} = [1\ 1] \Rightarrow \Omega_{c,1}[\Theta^n_{r,1}(P_3)] = 1$. Therefore, the analysis phase is validated since proposition 5.1 is satisfied, and the investigator can proceed to the next phase, Reporting.

6.1.4. Reporting

Step1: The process of validating the reporting phase is shown in the following four steps:
Step1: First determine vectors **a** and **d**. The AF technique is affecting this phase is a_1 = Code Injection
$d_1 1$ = PsInfo Volatility Plugin
Therefore, $a_1.d_1 = 1$, since a_1 is detected by d_1.
Step 2: Compute D_M once **a** and **d** are determined. Therefore, $D_M = \begin{pmatrix} a_1.d_1 \end{pmatrix} = \begin{pmatrix} 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [1]_{1\times 1}$
Step 3: Compute C_M once the AF technique a_1 is detected, that is, D_M has been computed. Now, a_1 can be countered by c_1 = execute the software with elevated privileges. Therefore, $ap_1.c_1$ = logical 1, and by definition $C_M = \begin{pmatrix} a_1.c_1 \end{pmatrix} = \begin{pmatrix} 1 \end{pmatrix}$. Therefore, $\Omega_{c,1}[\Theta^n_{r,1}(C_M)] = [1]_{1\times 1}$
Step 4: Compute phase function (P_f) once the values of D_M and C_M are determined. Therefore, by definition we have $P_{f_R eporting} = [1] \Rightarrow \Omega_{c,1}[\Theta^n_{r,1}(P_4)] = 1$. Therefore, the reporting phase is validated, and the investigator can stop here as reporting is the final phase of the DFM.
And, now since all the individual phase functions in the respective phases are logical 1. Mathematically, the P_f is an all ones unitary row vector i.e.,
$P_f = \begin{pmatrix} P_{f collection} & P_{f examination} & P_{f analysis} & P_{f reporting} \end{pmatrix}$. And since $P_f = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow {}_{r,1}\Phi^n_{c(r,1)}(P_n) = 1$. Therefore, by theorem 5.1 we can conclude that this four-phase DFM is validated.

6.2. Invalidation Case

The DFMs Rekhis and Boudriga (2012b) and Rani and Kumari (2017) which consists of five and four phases respectively will also be evaluated because these are the two DFMs in the literature that consider to detect anti-forensic techniques in the DF process. For the Rekhis Rekhis and Boudriga (2012b) model, only the third phase function is $\Omega_{c,1}[\Theta^n_{r,1}(P_3)] = 1$, and the remaining phase functions of the individual phases are $\Omega_{c,1}[\Theta^n_{r,1}(P_1)] = \Omega_{c,1}[\Theta^n_{r,1}(P_2)] = \Omega_{c,1}[\Theta^n_{r,1}(P_4)] = \Omega_{c,1}[\Theta^n_{r,1}(P_5)] = 0$. This means the anti-forensic techniques are detected and countered only in the third phase, but the other phases are affected by anti-forensic attacks resulting in the overall phase function, ${}_{r,1}\Phi^n_{c(r,1)}(P_n) = 0$. Therefore by theorem **??**(b), the model is invalid.
For the Rani's model Rani and Kumari (2017), the third phase function is also $\Omega_{c,1}[\Theta^n_{r,1}(P_3)] = 1$ the anti-forensic attacks are detected only in phase 3, but not during the other phases as the phase functions are $\Omega_{c,1}[\Theta^n_{r,1}(P_1)] = \Omega_{c,1}[\Theta^n_{r,1}(P_2)] = \Omega_{c,1}[\Theta^n_{r,1}(P_4)] = 0$ leading to the overall phase function $\Rightarrow {}_{r,1}\Phi^n_{c(r,1)}(P_n) = 0$. Therefore according to theorem **??** (b), the model is invalid.
After evaluating the hypothetical model, and the other two models in the literature it has been found that the hypothetical model is valid for investigation purposes because it satisfies the validation principle. The other models only consider to detect anti-forensic attacks in the analysis phase, and not in other phases of the models. Therefore, by validation principle they are invalidated for investigation purposes.The advantage of this principle is, it just take only four steps to validate a phase in the DFM. The principle systematically computes the AF techinques, detection methods, and countermeasure methods thereby facilitating a top-level view of the validation process which would be helpful to an investigator to validate or invalidate a DFM.

7. Conclusion and Future Scope

In this paper, we proposed a novel validation principle to validate existing Digital Forensic Models (DFMs). This principle can be used to assess or negate any risks prior to the investigation that are caused by anti-forensic techniques. This will be useful to anyone making a validation plan or report according to FSR-G-218 which is under the umbrella of Forensic Science Regulator (2016) where in any risks must be assessed before proceeding to validate a method. Also, if the rebuttal presumption of correct working of computers rule is invoked by citing anti-forensics as the reason, then validation principle could be used to prove or disprove the reason for invocation of that rule—that is, whether that rule is applicable to a particular case or not. The validation principle falls on the higher layer of abstraction as shown in Figure 1. The reason for this is because the validation principle can only validate a DFM if it has prior knowledge of an anti-forensic technique. It does not have the ability to predict a new anti-forensic technique such as zero-day attacks if it is

not present in the database.

In the future, we hope to address the aforementioned shortcoming by creating a database for AF processes for specific digital forensic phases, which could be installed on a global server, so that the database could be updated for various emerging AF techniques, and investigators around the world could test their digital forensic models for AF techniques to ascertain whether a model could be used for investigation purposes or not. This would facilitate further research in the area of detecting anti-forensic methods, and testing forensic tools against anti-forensic techniques as argued by Wundram et al. (2013). Also, the validation principle can be incorporated into machine learning and deep learning algorithms that could be used to predict anti-forensic methods affecting a digital forensic process.

References

Agarwal, A., Gupta, M., Gupta, S., Chandra Gupta, S., 2011. Systematic Digital Forensic Investigation Model. Gupta International Journal of Computer Science and Security doi:10.1149/1.2992231.

Ali, A., Razak, S.A., Othman, S.H., Mohammed, A., Saeed, F., 2017. A metamodel for mobile forensics investigation domain. PLoS ONE doi:10.1371/journal.pone.0176223.

Ashcroft, J., Daniels, D., Hart, S., 2019. Forensic examination of digital evidence: A guide for law enforcement. URL: https://www.ncjrs.gov/pdffiles1/nij/199408.pdf.

Bilby, D., . Low down and dirty: Anti-forensic rootkits. URL: https://www.blackhat.com/presentations/bh-jp-06/BH-JP-06-Bilby-up.pdf.

Ciardhuáin, S., Beebe, N.L., Clark, J.G., Palmer, G., Perumal, S., Selamat, S.R., Yusof, R., Sahib, S., 2009. An extended model of cybercrime investigations. the First Digital Forensic Research Workshop (DFRWS) doi:10.1504/IJESDF.2010.033780.

Courts and Tribunals Judiciary, 2014. Criminal practice directions. URL: https://www.judiciary.uk/wp-content/uploads/JCO/Documents/Practice+Directions/Consolidated-criminal/criminal-practice-directions-2013.pdf.

Dahbur, K., Mohammad, B., 2011. The anti-forensics challenge, in: Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications - ISWSA '11. doi:10.1145/1980822.1980836.

Forensic Science Regulator, 2014. Fsr guidance validation. URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/375285/FSR-G-201_Validation_guidance_November_2014.pdf.

Forensic Science Regulator, 2016. Method validation in digital forensics. URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/528123/FSR_Method_Validation_in_Digital_Forensics_FSR-G-218_Issue_1.pdf.

Forensic Science Regulator., 2019. Forensic science regulator. URL: https://www.gov.uk/government/organisations/forensic-science-regulator/about/membership#forensic-science-advisory-council.

Garfinkel, S., 2006. Anti-Forensics : Techniques , Detection and Countermeasures. Security doi:10.1.1.109.5063.

Gül, M., Kugu, E., 2017. A survey on anti-forensics techniques, in: Artificial Intelligence and Data Processing Symposium (IDAP), 2017 International, IEEE. pp. 1–6.

Harbawi, M., Varol, A., 2017. An improved digital evidence acquisition model for the Internet of Things forensic I: A theoretical framework. 2017 5th International Symposium on Digital Forensic and Security, ISDFS 2017 , 1–6doi:10.1109/ISDFS.2017.7916508.

Hitchcock, B., Le-Khac, N.A., Scanlon, M., 2016. Tiered forensic methodology model for digital field triage by non-digital evidence specialists. Digital investigation 16, S75–S85.

House of Commons Science Technology Committee, 2019. Forensic science on trial. URL: https://publications.parliament.uk/pa/cm200405/cmselect/cmsctech/96/96i.pdf.

International Standards Organisation, 2017. Iso 17025:2017 general requirements for the competence of testing and calibration laboratories. URL: https://www.iso.org/standard/66912.html.

Kao, D.Y., Wu, G.J., 2016. A Digital Triage Forensics framework of Window malware forensic toolkit: Based on ISO/IEC 27037:2012. Proceedings - International Carnahan Conference on Security Technology 2015-January, 217–222. doi:10.1109/CCST.2015.7389685.

Karie, N.M., Kebande, V.R., Venter, H., Choo, K.K.R., 2019. On the importance of standardising the process of generating digital forensic reports. Forensic Science International: Reports 1, 100008.

Kaur, P., Bijalwan, A., Joshi, R., Awasthi, A., 2018. Network forensic process model and framework: An alternative scenario, in: Intelligent Communication, Control and Devices. Springer, pp. 493–502.

Kebande, V.R., Karie, N.M., Michael, A., Malapane, S., Kigwana, I., Venter, H.S., Wario, R.D., 2018. Towards an integrated digital forensic investigation framework for an IoT-based ecosystem. Proceedings - 2018 IEEE International Conference on Smart Internet of Things, SmartIoT 2018 , 93–98doi:10.1109/SmartIoT.2018.00-19.

Kigwana, I., Kebande, V.R., Venter, H.S., 2017. A proposed digital forensic investigation framework for an eGovernment structure for Uganda. 2017 IST-Africa Week Conference, IST-Africa 2017 , 1–8doi:10.23919/ISTAFRICA.2017.8102348.

Kohn, M.D., Eloff, M.M., Eloff, J.H., 2013. Integrated digital forensic process model. Computers and Security doi:10.1016/j.cose.2013.05.001, arXiv:arXiv:1507.07739v1.

Lindsey, T., 2017. Current cyber investigation challenges. URL: https://www.dfrws.org/sites/default/files/session-files/pres-current_cyber_investigation_challenges_in_digital_forensics.pdf.

Lutui, R., 2016. A multidisciplinary digital forensic investigation process model. Business Horizons 59, 593–604.

Meyers, M., Rogers, M., 2005. Digital forensics: Meeting the challenges of scientific evidence, in: IFIP International Conference on Digital Forensics, Springer. pp. 43–50.

Montasari, R., 2016. Review and Assessment of the Existing Digital Forensic Investigation Process Models. International Journal of Computer Applications doi:10.5120/ijca2016911194.

National Institute of Standards and Technology, . Computer forensics tool testing program. URL: https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt.

Oriwoh, E., Jazani, D., Epiphaniou, G., Sant, P., 2013. Internet of Things Forensics: Challenges and Approaches. 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing , 608–615doi:10.4108/icst.collaboratecom.2013.254159.

Palmer, G., 2017. Digital forensic research working group. URL: http://dfrws.org/sites/default/files/session-files/a_road_map_for_digital_forensic_research.pdf.

Rani, D.R., Kumari, G.G., 2017. A framework for detecting anti-forensics in cloud environment. Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016 , 1277–1280doi:10.1109/CCAA.2016.7813913.

Reith, M., Carr, C., Gunsch, G., 2002. An examination of digital forensic models. International Journal of Digital Evidence doi:10.1109/SADFE.2009.8.

Rekhis, S., Boudriga, N., 2012a. A system for formal digital forensic investigation aware of anti-forensic attacks. IEEE Transactions on Information Forensics and Security 7, 635–650. doi:10.1109/TIFS.2011.2176117.

Rekhis, S, S., Boudriga, N., 2012b. A Hierarchical Visibility theory for formal digital investigation of anti-forensic attacks. Computers and Security 31, 967–982. URL: http://dx.doi.org/10.1016/j.cose.2012.06.009, doi:10.1016/j.cose.2012.06.009.

Rogers, M., Goldman, J., Mislan, R., Wedge, T., Debrota, S., 2006. Computer Forensics Field Triage Process Model. The Journal of Digital Forensics, Security and Law doi:10.15394/jdfsl.2006.1004.

Scientific Working Group on Digital Evidence, 2019. Model standard operation procedures for computer forensics. URL: https:

//www.swgde.org/documents/Current%20Documents/SWGDE%20QAM%20and%20SOP%20Manuals/SWGDE%20Model%20SOP%20for%20Computer%20Forensics.

Shaid, S.Z.M., Maarof, M.A., 2015. In memory detection of windows api call hooking technique, in: 2015 international conference on computer, communications, and control technology (i4CT), IEEE. pp. 294–298.

Soltani, S., Seno, S.A.H., 2019. A formal model for event reconstruction in digital forensic investigation. Digital Investigation .

Stüttgen, J., Cohen, M., 2013. Anti-forensic resilient memory acquisition. Digital investigation 10, S105–S115.

The National Archives, 2019. Police and criminal evidence act 1984. URL: http://www.legislation.gov.uk/ukpga/1984/60/contents.

The National Archives., 2019. Youth justice and criminal evidence act 1999. URL: http://www.legislation.gov.uk/ukpga/1999/23/contents.

Upcounsel, . The federal standard of expert testimony reliability before daubert. URL: https://tinyurl.com/yxa5gpdx.

Valjarevic, A., Venter, H.S., 2012. Harmonised digital forensic investigation process model, in: 2012 Information Security for South Africa - Proceedings of the ISSA 2012 Conference. doi:10.1109/ISSA.2012.6320441.

Williams, J., 2012. Acpo good practice guide acpo good practice guide for digital evidence. URL: http://library.college.police.uk/docs/acpo/digital-evidence-2012.pdf.

Wundram, M., Freiling, F.C., Moch, C., 2013. Anti-forensics: The next step in digital forensics tool testing, in: Proceedings - 7th International Conference on IT Security Incident Management and IT Forensics, IMF 2013. doi:10.1109/IMF.2013.17.

Yusoff, Y., Ismail, R., Hassan, Z., 2011. Common Phases of Computer Forensics Investigation Models. International Journal of Computer Science and Information Technology doi:10.5121/ijcsit.2011.3302, arXiv:0607373v1.

Zhang, N., Zhang, R., Sun, K., Lou, W., Hou, Y.T., Jajodia, S., 2018. Memory forensic challenges under misused architectural features. IEEE Transactions on Information Forensics and Security 13, 2345–2358.

Zia, T., Liu, P., Han, W., 2017. Application-Specific Digital Forensics Investigative Model in Internet of Things (IoT), in: Proceedings of the 12th International Conference on Availability, Reliability and Security - ARES '17. doi:10.1145/3098954.3104052.

A. Appendix

Proposition A.1. *(a) If in a DFM phase,* $\Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [k]_{1\times1}$*,* $k \in \mathbf{B}$*, and if k=1 then the af techniques* $a \in A$ *are detected, and if k=0, then A is undetected.*
(b) If $\Omega_{c,1}[\Theta^n_{r,1}(C_M)] = [k]_{1\times1}$ *then the detected af technique* $a \in A$ *is countered if k=1, and is not countered if k=0.*

Proof: Consider any arbitrary DF phase, and if $a \in A$ are the n anti-forensic vectors acting upon the phase and if each vector in A can be detected by 'n' number of ways, then the detector tensor product of A and D is defined as $D_M = a_j.d_i^T$, where $a_j \in A$, $d_i \in D$. Let's assume D_M is a square matrix of order $n \times n$ consisting of 'n' number of rows defined by the set $R_n = \{R_1, R_r, \dots, R_n\}$, where any arbitrary row R_j row in R_n is defined as $\{R_j = a_j.d_1, a_j.d_2, \dots, a_j.d_n\}$. If $\{\exists a_j.d_i \in R_j | a_j.d_i = 1\}\ \forall R_n$ then a_j is detected by d_i. Now, if a row vector transformation is applied on D_M, then it yields an all ones column vector E of order $n \times 1$ i.e., $\Theta^n_{r,1}(D_M)] = E_{n\times1}$. Therefore, the column transformation of E, $\Omega_{c,1}(E) = \Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [1]_{1\times1}$, therefore $\forall a \in$ A is detected in a DFM phase. Now, if $\{\exists a_j.d_i \in R_j | aj.d_i = 0\}\ \forall R_n$ then $\Omega_{c,1}[\Theta^n_{r,1}(D_M)] = [0]_{1\times1}$, and the DFM phase is still under the effect of an anti-forensic attack. Proposition 1(a) is also known as the sufficiency principle. After the anti-forensic technique a_j is detected, it is passed to the counter product (C_M) as shown in Figure 3. The detected anti-forensic methods $a_j \in A$ in D_M can be countered by C=$\{c_1, c_2, c_3, \dots, c_j, \dots, c_n\}$ in 'n' number of ways, and thus the counter product is defined as $C_M = (a_j.d_j).c_j^T = (e_j).c_j^T = E.C^T$. Now in C_M if there exist at least one c_j such that axiom of counterability is satisfied then a_j is countered by c_j i.e., if $\{\exists e_j.c_i \in R_j | e_j.c_i = 1\}\ \forall R_n$ then a_j is countered by c_i, and if all the values on R_n are 0 then the anti-forensic technique is not countered i.e., if $\{\forall e_j.c_i \in R_j | e_j.c_i = 0\}$ $\forall R_n$ then a_j is not countered by c_i.

Proposition A.2. *If* $[\Theta_{r,i}(a_j.d_i)] = [1]_{1\times1}$ *then* $a_j\ [\ \pi_{i=0}^{N-1-i}(\bar{S}_i)\ S_{N-i}\ \pi^i_{k=0}(\bar{S_{N-K}})\ S_{N-i}]$ *gives the element* $d_i \in R_i$ *that detects* a_j*.*

Proof: Since detector product D_M is defined as $\{a_j.d_j \in D_M | D_M = a_j.d_j{}^T\} = \{R_j \in R_n | R_n = a_j.d_n\}$. Therefore, D_M is a set of rows R_1 to R_n, that is, $\{R_1, R_2, R_3, \dots, R_n\}$. Now consider any arbitrary row $R_j \in R_n$ defined as R_j=$a_j.d_1$+ $a_j.d_2 + a_j.d_3 + \cdots + a_j.d_n$. Now, if $a_j \in A$ forms the inputs to a n:1 multiplexer, and d forms the select lines $S = \{S_i \in S_n | S_i = (\pi_{i=0}^{N-1-i}\bar{S}_i.S_{N-i})(\pi^i_{k=0}\bar{S_{N-k}})\}$. Now if d_i and S are bijective, i.e., f:$d_i \rightarrow$ S. This can be represented as

$$\begin{pmatrix} d_0 \\ d_1 \\ \dots \\ d_{n-1} \\ d_n \end{pmatrix} \rightarrow \begin{pmatrix} \bar{S}_0 & \bar{S}_1 & \bar{S}_2 & \dots & \bar{S_{N-1}} & S_N \\ \bar{S}_0 & \bar{S}_1 & \bar{S}_2 & \dots & S_{N-1} & \bar{S_N} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \bar{S}_0 & S_1 & \bar{S}_2 & \dots & \bar{S_{N-1}} & \bar{S_N} \\ S_1 & \bar{S}_0 & \bar{S}_2 & \dots & S_{N-1} & \bar{S_N} \end{pmatrix}$$

Therefore, R_j=$a_j(\bar{S}_0\bar{S}_1 \dots S_N)$+$\cdots$+$a_j(S_0\bar{S}_1 \dots \bar{S_N})$. This equation shows what d_i in R_j detects a_j. For example, if R_1=$a_j(\bar{S}_0\bar{S}_1 \dots S_N)$, this means a_1 is detected by d_1 in row 1 (R_1) of D_M.

Theorem A.1. *If (a)* $\{\forall P_i \in P_n | P_i = 1 \vee_{r,1} \Phi^n_{c(r,1)}(P_n) = 1\}$ *then the DFM is validated.*
(b) $\{\exists P_i \in P_n | P_i = 0 \vee_{r,1} \Phi^n_{c(r,1)}(P_n) = 0\}$ *then the DFM is invalidated.*

Proof: By the definition of validation principle a DF phase is validated if all the anti-forensic techniques of the respective phases in a DFM are detected and countered. Mathematically, the validation principle is defined as
$P_{fj} = [\Omega_{c,1}[\Theta^n_{r,1}(a_j.d_i)]]$. $[\Omega_{c,1}[\Theta^n_{r,1}((a_j.d_i).c_i)]] = \Omega_{c,1}[\Theta^n_{r,1}(a_j.d_i).((a_j.d_i).c_i)] = \Omega_{c,1}[\Theta^n_{r,1}(a_j.d_i).(e_j.c_i)] = \Omega_{c,1}[\Theta^n_{r,1}(D_M).(C_M)] \Rightarrow \Omega_{c,1}[\Theta^n_{r,1}(P_i)]$.

Also, according to the validation principle if the phase function of the individual phases P_i and $i \in N$ in DFM is $[1]_{1\times1}$, then its respective phases are validated. Consider a DFM with its respective phase functions as shown in Figure 4. Now, the individual phase functions is P_n=$\{P_1, P_2, \dots, P_n\}$ $\Rightarrow \{\Omega_{c,1}[\Theta^n_{r,1}(P_1)], \Omega_{c,1}[\Theta^n_{r,1}(P_2)], \dots, \Omega_{c,1}[\Theta^n_{r,1}(P_n)]\}$, which is a row vector and represented as ${}_{r,1}\Phi^n_{c(r,1)} = \Omega_{r,1}[\Omega_{c,1}[\Theta^n_{r,1}(P_i)]]$ and after the application of proposition A.1 (a) and (b) to ${}_{r,1}\Phi^n_{c(r,1)}$ if all phase functions in a digital forensic model is

$[1]_{1x1}$ i.e., $\{\forall P_i \in P_n | P_i = 1\}$, then ${}_{r,1}\Phi^n_{c(r,1)}(P_n) = 1$ and the DFM is validated. Else if at least one of the elements in P_n or if at least one of phase functions in a digital forensic model is 0 i.e., $\{\exists P_i \in P_n | P_i = 0\}$, then ${}_{r,1}\Phi^n_{c(r,1)}(P_n) = 0$ and the DFM is invalidated.