Cold Boot Attack on Encrypted Containers for Forensic Investigations

Frimpong Twum[1*], Emmanuel Mawuli Lagoh[1], Yaw Missah[1], Najim Ussiph[1], and Emmanuel Ahene[1]

[1] Department of Computer Science, College of Science, Kwame Nkrumah University of Science and Technology,
Ghana, West Africa
[e-mail : ftwum.cos@knust.edu.gh, emma3lag@gmail.com, ymissah@gmail.com, nussiph@yahoo.com,
eahene@gmail.com]
*Corresponding author: Frimpong Twum

*Received February 22, 2022; revised June 2, 2022; accepted August 20, 2022;
published September 30, 2022*

Abstract

Digital Forensics is gaining popularity in adjudication of criminal cases as use of electronic gadgets in committing crime has risen. Traditional approach to collecting digital evidence falls short when the disk is encrypted. Encryption keys are often stored in RAM when computer is running. An approach to acquire forensic data from RAM when the computer is shut down is proposed. The approach requires that the investigator immediately cools the RAM and transplant it into a host computer provisioned with a tool developed based on cold boot concept to acquire the RAM image. Observation of data obtained from the acquired image compared to the data loaded into memory shows the RAM chips exhibit some level of remanence which allows their content to persist after shutdown which is contrary to accepted knowledge that RAM loses its content immediately there is power cut. Results from experimental setups conducted with three different RAM chips labeled System A, B and C showed at a reduced temperature of -25C, the content suffered decay of 2.125% in 240 seconds, 0.975% in 120 seconds and 1.225% in 300 seconds respectively. Whereas at operating temperature of 25°C, there was decay of 82.33% in 60 seconds, 80.31% in 60 seconds and 95.27% in 120 seconds respectively. The content of RAM suffered significant decay within two minutes without power supply at operating temperature while at a reduced temperature less than 5% decay was observed. The findings show data can be recovered for forensic evidence even if the culprit shuts down the computer.

Keywords: Cold Boot Attack, Forensic Investigation, Encryption Keys, Remanence Effect, Volatile Data.

http://doi.org/10.3837/tiis.2022.09.013 **ISSN : 1976-7277**

1. Introduction

The Random Access Memory, referred to as the RAM, is generally believed to be volatile and thus, it loses its data immediately the computer is cut from power supply [1]. Many people including digital forensic experts believed this as well. Recent studies have countered this widespread assumption that the content of the RAM doesn't vanish immediately but rather fades gradually [2], [3]. However, the integrity of the content of the RAM is only reliable when the capacitors are constantly refreshed. These capacitors lose their charges with time when they are not being refreshed which leads to loss or corruption of the content of the RAM. Dynamic RAM (DRAM), been the most common form of memory used by computers these days [4], consists of capacitors that store data in the form of charges as long as the computer is on and it is receiving power. In the DRAM, a transistor and a capacitor are paired to form a cell of memory that is capable of storing one bit of data. A capacitor with an electron is represented by "1" while an empty capacitor is represented by "0" [2].

Most systems have their drives encrypted and protected with keys, these encryption keys and other cryptographic materials are being stored in the RAM after the drive has been decrypted and mounted on the first instance [5]. Encryption has well been embraced by computer users; both the good and bad guys. Basically, encryption prevents unauthorized access to data in a digital medium. Cyber-attacks ranging from computer fraud, pedophilia, terrorism, organized crimes have given an increase to the use of encryption by culprits [6]. The complexity of the various encryption algorithms used to encrypt these data makes it difficult, if not impossible for Forensic Investigators or Cyber Security Analysts to decrypt or break while conducting investigations or during analysis. Evidence that has been encrypted is a major challenge to digital forensic investigations [7].

However, when these data have been decrypted and loaded into memory, the decryption keys are stored in the memory as long as the computer is on and the DRAM is being refreshed. If an investigator can gain physical access to the computer of a suspect in this state, these encryption keys and other volatile data can be obtained for investigation and analysis even on encrypted drives at a later time.

Cold Boot attack is a technique used by both forensic examiners and attackers to access the content of memory when there is a power cut or the computer has been shut down within a considerable period of time. This technique helps forensic examiners and attackers to access critical data like encryption keys and other cryptographic materials kept in memory while the computer is active and running. The volatility of RAM requires that cold boot attacks are carried out not so long after the shutdown of the computer. Prior to the knowledge of cold boot attacks, people committing crimes with the computer simply turn off the computer or cut its power supply in order to destroy evidence and clear traces of encryption keys purposely to hinder any form of forensic investigation that may be carried out.

The most common approach of performing a digital forensic investigation is to image the computer storage and perform an analysis of its content [8],[9]. This approach known as the post-mortem analysis is efficient if the content of the drive is not encrypted. The common approach used when encrypted data is encountered is to persuade the suspect to give out their passwords or encryption keys which most of them would decide not to or claim to have forgotten [10],[11]. Some suspects may use duress keys or hidden containers to conceal their crimes in a way that two keys can be used to decrypt the drive. One key gives access to the real data whiles the other or the duress key opens up prearranged content with no connections to the crime or with no criminal intent. However, the acquisition and analysis of live data largely addresses the presence of encrypted data as the encryption key is stored in the RAM while the computer is running. Modern investigators are delving more into live data or volatile data (data in RAM) analysis as it sometimes contains enough evidence to solve a case. Another major advantage that "live data analysis" has over post-mortem analysis is, the analyst or the forensic investigator can have access to encryption keys in plain text as the RAM stores encryption keys as such.

Post-mortem analysis is limited when the drive is encrypted. On-The-Fly (OTF) encryption mechanism is used in most computer systems and this poses a challenge to Forensic Investigators when they fail to get physical access to the live system with decrypted disks. The RAM which stores the encryption credentials is also not accessible when the computer is shut down. This study developed an approach to acquire forensic data from encrypted disk or containers even when the computer is shut down or screen locked by retrieving the encryption keys from the RAM. The outcome of this study is a great leap in the practice of Digital Forensics as it creates more avenue for investigators to collect evidence. It provides an efficient method of accessing the cryptographic materials in the RAM of a computer even after the power to the computer is cut. Investigators can also access or vital data deemed volatile from the memory dump using this method.

In this study, we practically explored the remanence of RAM chips for cold boot attacks by transplanting the chip into another computer system. We experimented with variety of computer systems with different hardware components including DDR, DDR2 and DDR3 memory chips. We provide measurements that infer the impact of temperature on the remanence of RAM chips. We also developed tools in C to automate the retrieval of AES keys from a memory dump. These tools are not limited to cold boot attacks but can also be used to retrieved AES keys from a RAM dump be it live data forensics.

2. Related Work

The concept of cold boot attack employed for our study was first demonstrated in 2008 by Halderman et al. though it has been theoretically known years earlier [3], [12], [13]**.** In their paper, Halderman et al. demonstrated the remanence of DRAM chips which they exploited to retrieve volatile data from the RAM of a computer, specifically cryptographic keys. Partially decayed keys retrieved were also reconstructed using recovery algorithms and then used to break full disk encrypted containers (TrueCrypt, BitLocker, and FileVault) which implemented RSA, DES, AES encryption schemes [11], [12]. Their study developed a tool called AESfix which was used to reconstruct AES keys in decayed portions of the memory dump. Their study uncovered some security vulnerability with regards to confidentiality of data on the computer in encrypted containers. In attempt to patch these identified vulnerabilities, the Trusted Computing Group (TCG) introduced both software and hardware means to mitigate these attacks. TCG continues to address this vulnerability by introducing other means to preventing access to the data in the RAM after a reset, one of the measures is introducing a memory reset bit which causes the memory to reset all the bits to "0" whenever the CPU detects an improper shut down [14]. In 2013, Gruhn and Müller [14] carried out a study which demonstrated how to circumvent some of the software-based countermeasures implemented [10]. Some of the software-based countermeasures Gruhn dealt with are; DDR3 memory scrambling, RAM reset on boot, locking the boot process, and temperature detection. After these studies on cold boot, a number studies were carried out on retrieving and reconstructing keys from memory images [12], [16], [17].

In a study by Khorshed Alam et al. [6] on the recovery of encryption data from memory, they highlighted the recuperation of encryption and decryption keys from the memory dump of a live system. Those keys were later used to access the encrypted drive. The study focused on live data forensics and used a sequential search approach to look through the memory dump to retrieve the encryption keys present.

Most judicial courts around the world now accept electronic evidence as a form of evidence to judge cases, however there are certain modalities put in place to preserve the validity of the evidence [18], [19]. The admissibility and validity of digital evidence in court depends on the soundness of the forensic procedures used in the evidence collected [2], [20] and also the soundness of the evidence collected. The tools used in forensic data acquisition must also meet jurisdictional standards for evidence to be accepted [21].

The various works done on the concept of cold boot since 2008 have been an improvement in the practice of digital forensics and the process of data collection, however hardware and software manufacturers keep implementing mitigations to prevent access to data using cold boot techniques as criminals are also exploiting the concept for malicious gain.

3. The Setup Environment

The study experimentations were carried out with a 32-bit laptop running Windows 7 Enterprise Operating System as the **target system** with TrueCrypt encrypted containers. Four different DRAM chips with varying densities or memory capacities were used in the laptop. We used a laptop because it is the more exposed to physical access and likely to be a target for a cold boot attack due its mobility.

A 16GB SanDisk USB Pen drive was provisioned with the memory capture tool and also used as the destination for the memory dump. The provisioning of the USB Pen drive with the memory capture tool was done using the **host system** running on Linux. It is necessary for the USB Pen drive to be large enough to accommodate the dump that would be taken by the capture tool.

Canned nitrogen gas was used to reduce the temperature of the RAM. Nitrogen gas does not react with the RAM chips to cause corrosion. An Infrared Thermometer gun which is able to read the surface temperature from a distance of 5cm was used to take the temperature readings. **Table 1** outline the specifics of the computer systems used for the tests.

Table 1. Computer hardware systems used for the test

Systems	Manufacturer	Memory Type	System Make	Memory Size
A	Samsung	DDR	Lenovo	512
B	Hynix	DDR2	Lenovo	1024
C	Samsung	DDR2	HP	2048
D	Samsung	DDR3	Lenovo	2048

The processes involved in the experiment are discussed later in this section. The computer system that is been tested is referred to as the **target system** while the one used for the analysis is the **host system**.

3.1 The Sequence of the Experiment

The experiment was conducted following the steps below:

1. Deploy the imaging tool on the USB drive using the host system

2. Boot the target computer and log in with the user password.
3. Load a sample text as test data into memory of the target system.
4. Reduce the temperature of the RAM to *T* °C.
5. Cut the power supply to the target computer.
6. Wait for a specific time *s*.
7. Transplant the RAM to the host computer and reboot it to make a copy of the RAM.
8. Analyze the dump for its content as well as inconsistencies.

3.2 Memory Acquisition

The image of the RAM in a live target system could easily be captured using straight-forward and simple off-the-shelf tools like the Magnet RAM, MDD, Winen, FTK Imager from AccessData, etc. Some of these tools could as well be used to image hard disk for non-volatile data in a post mortem analysis [22]. The hard disk from a dead system could be extracted, imaged, and analyzed for evidential materials at an investigator's convenient time. However, this is not so for volatile data or RAM forensics. RAM data presents the state of the system at a particular point in time.

As stated earlier in this document, disk encryption or password protection hinders access to data on a system. To acquire the image of the RAM of a system with full disk encryption, we take advantage of the remanence of the RAM which allows data to persist for a while before decaying. It was exploited by transplanting the RAM into another computer for imaging. This was achieved with a tool we developed based on the cold boot attack concept of Halderman and his colleagues at Princeton University [3]. This tool was appropriate for the experiment because it is 10KB in size and hence leaves a relatively small footprint in memory. This tool is a simple boot image that copies the content of addressable memory of a device when booted. It could either be deployed with a USB disk or over a network (PXE utility) as most modern computers support booting over the network [15]. The PXE scraper is transferred over the network as part of the target's network bootstrap process, while the USB scraper is loaded from the USB disk. The pxedump utility runs on a remote system and requests a memory dump from the target using a simple UDP-based protocol. The usbdump utility on the other hand is used to recover a raw memory dump from a disk after it's been written out by the target.

3.2.1 Capturing a Memory Dump over the Network

Two pieces of software are needed to get the scraper loaded into a target computer: a DHCP server and a TFTP server. Most Linux systems include a TFTP server in their base installations, and a DHCP server is often also present by default or can be easily added as an option. TFTP and DHCP servers are also available for Windows. The DHCP server should be configured to hand out leases specifying the file path that can be used to access the scraper utility via TFTP. The simplest approach is to use a laptop with an Ethernet port. Install and configure the DHCP and TFTP servers, then connect the laptop to the target system using a crossover cable. Once this is done, reboot or reset the target computer. Ensure that the BIOS on the target system does not perform a destructive memory test when it restarts.

When the target system's BIOS starts up, ask it to boot via network instead of from disk. Exactly how this is done varies depending on the BIOS implementation. Some systems offer a simple hotkey override ("Press F12 to boot from network") while others may require you to enter the BIOS configuration utility to enable PXE support and specify the network interface as a boot device. In any case, once the target begins its PXE boot sequence, it will begin searching for a DHCP server. Once it obtains a response from the laptop, it should download the scraper binary via TFTP and launch it. The scraper will print some status messages and then wait for a handshake from the pxedump utility. The scraper will use the IP address obtained by PXE from the DHCP server. At this point, the pxedump utility can be run on the computer system as follows:

```
% pxedump [IP address of target system] > memorydump.dat
```

The dumper should begin copying the target's memory to the disk. This dump will include the 640KB of lower memory and all extended memory (up to 3.5GB). Once the dump completes, the scraper will attempt to power off the target system using APM. If this fails (i.e. no APM BIOS is present), it will reboot the target instead.

3.2.2 Capturing a Memory Dump on a Disk

To use the USB-based memory scraper, we needed a 16GB disk which was large enough to hold a dump from the RAM of the target system. The storage medium can be any storage device as long as the target system's BIOS supports it as a boot device. USB mass storage devices are good candidates: these include USB thumb drives, USB SD cards, compact flash card readers (with appropriate media attached), and ordinary hard disks in USB disk enclosures. Once the selected storage device is connected to the host system, the scraper binary can be dumped onto it. On UNIX/Linux systems, this can be done with the dd command as follows:

```
# dd if=scraper.bin of=/dev/diskdev
```

No special formatting of the disk is needed: the disk will now function as a standalone memory capture device. To use it, connect it to the USB port of a target system, res it, and then set the system to boot from USB. As soon as the target boots, it will load the scraper program and begin dumping all available RAM to the USB device. Once it completes, it will turn off the system or reset it if APM power off is not supported.

To recover the memory dump, connect the disk to the host system again, and use the usbdump utility to extract the dump image with this command:

```
# usbdump /dev/diskdev > memorydump.dat
```

3.3 RAM Remanence

The outcome of this study is largely dependent on the remanence effect of the memory hardware used in the study. It was thoroughly studied as it is the core of the study. It turned out from the experimentations that most RAM exhibit data persistence. Thus, the contents of RAM survive for brief periods even after it's been powered off. This phenomenon can be exploited for various purposes, both good and evil. RAM persistence can be exploited using both hardware and software mechanisms. One major challenge to hardware exploits is that they require a certain amount of specialized expertise and a willingness and/or opportunity to disassemble or possibly damage the system being exploited. With many systems, there is no alternative to this, particularly with computers that perform destructive memory tests or ECC (Error-Correcting Code) scrubbing at startup. However, there are a surprising number of machines where the contents of RAM survive undamaged well after the system BIOS or boot code has finished running [10]**,** and these can be exploited much more easily using only software.

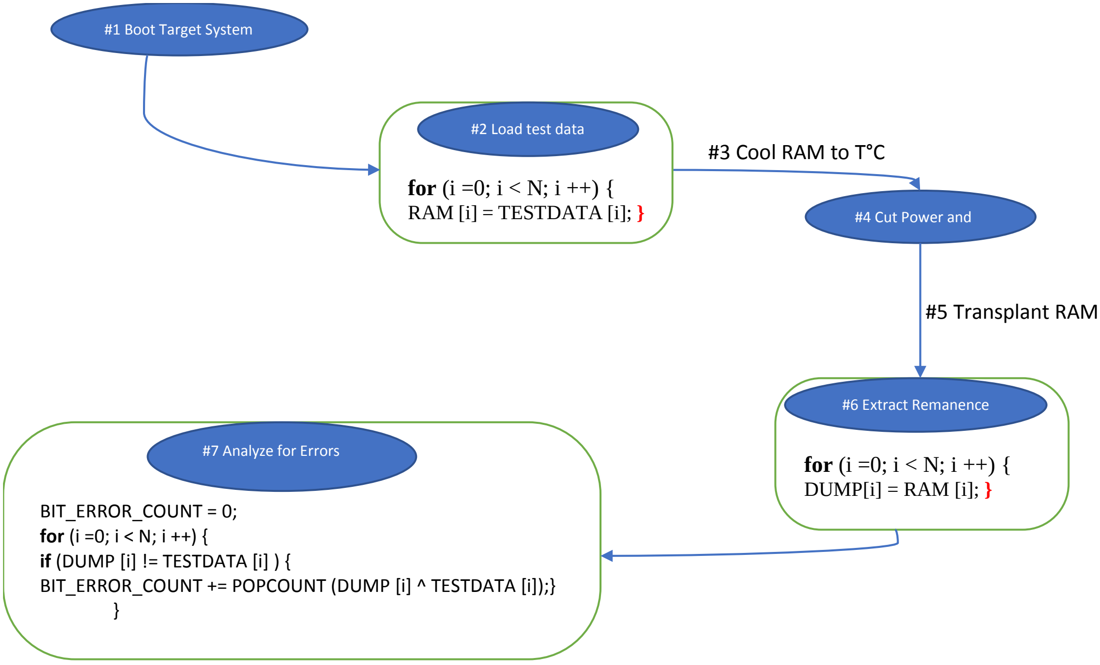
3.3.1. Cooling and Temperature Measurements

The temperature of the RAM at room temperature read 25°C on an infrared non-contact thermometer gun. The compressed air or canned nitrogen gas was then sprayed evenly on the surface of the RAM chip. The orientation of the RAM in the laptop provided access to only the top surface to be sprayed. Nitrogen gas was a good choice for the experiment because it is able to adequately cool the RAM and also, it is not reactive and hence does not cause rusting of the chips. The infrared thermometer was able to read the temperature on the surface of the chip from about 5cm distance.

The temperature reading was recorded immediately after applying the cooling agent but before turning the computer off. The region of the RAM closer to the socket was always hotter than the region further away from the socket. Applying the cooling agent while the computer is still on makes the RAM tend to heat up again. At the instance of RAM transplant, we applied the cooling agent again to maintain the temperature low enough to slow decay as much as possible.

3.3.2. Data Decay

When the power to the computer is cut for a particular period of time, we expected a level of decay to occur to the content of the RAM. To observe and measure this, we had to load a certain finite data which was our test data into memory whiles the computer is on and active. The test data was loaded into the RAM using a placer tool developed by Gruhn [2]. This tool copies the data to a location in memory and issues a WBINVD instruction (WBINVD – Write Back and Invalidate Cache) to ensure the data is written to the RAM but not the CPU caches that might available. The tool loops to load the bits of the test data into memory one after the other until the last bit is loaded. After loading the test data into memory, the power to the computer is cut and the RAM transplanted into our host system. The host system boots the memory capture tool in less than a minute because of its miniature size. We now take the image of the RAM and make a bitwise check; comparing the bits that were loaded into memory to those that were retrieved. The variation in any of the bits implied there is a decay. This can be interpreted mathematically to state the rate of decay at varying temperatures and time intervals. The percentage of the total bits that changed their state (ie from “1” to “0”) represents the percentage of decay that has taken place. Fig. 1 is the flow diagram and pseudocode of our experimental design to determine the rate of bits that decay.



```
for (i =0; i < N; i ++) {
RAM [i] = TESTDATA [i]; }
```

```
for (i =0; i < N; i ++) {
DUMP[i] = RAM [i]; }
```

```
BIT_ERROR_COUNT = 0;
for (i =0; i < N; i ++) {
if (DUMP [i] != TESTDATA [i] ) {
BIT_ERROR_COUNT += POPCOUNT (DUMP [i] ^ TESTDATA [i]);}
        }
```

Fig. 1. Flow diagram of our experimental setup

The following are terminologies and variables used in the pseudocode:

BIT_ERROR_COUNT: number of bit errors.
DUMP: dump of the RAM stored on the boot medium.
N: number of bytes in test data
POPCOUNT: hamming weight of a byte
RAM: physical RAM being analyzed
TESTDATA: finite data loaded into memory.

3.4 Encryption Software

The encryption software used to encrypt the drive used in the setup was TrueCrypt which implements AES in LRW (Liskov, Rivest & Wagner) mode [23].

3.4.1 TrueCrypt

This is a type of software that not only creates but also manages encrypted disks while it is still running or on-the-fly (OTF). OTF encryption implies the data gets encrypted automatically when it is saved on the disk and decrypted when it is loaded. Data on the encrypted disk can only be accessed when one has the right password or the right keys for encryption. TrueCrypt functions by encrypting the entirety of the file system without leaving out files and folder names, existing metadata, available contents of all files, and also the free spaces on the disk. Copying files from a mounted volume in TrueCrypt to and from other sources follows the same digital process as other drives, one simply has to drag and drop. During the transfer of files from an encrypted TrueCrypt drive either by copying or reading, the files in memory or RAM are spontaneously decrypted. The operation is true in the reverse scenario. That is, files being moved to TrueCrypt are also spontaneously encrypted. This however does not mean the complete file that is to be encrypted or decrypted needs to be kept in RAM before the encryption or decryption can take place. It is worth noting that, TrueCrypt needs no additional memory allocation [24].

To illustrate, if you have a video file with *.avi* extension saved on a TrueCrypt drive, that entire video file is encrypted. When a user provides the correct password or keyfile, the drive hosting TrueCrypt is mounted. When the user attempts to play the video file by double-clicking the file icon, the Operating System opens and runs the associated program that corresponds with the file type, which is most likely a media player. In a bit to play the file, the media player will load a tiny fraction of the file from the encrypted TrueCrypt drive into memory. This causes TrueCrypt to instantly decrypt that portion of the file as it loads in the memory. The media player is then able to play the section of the decrypted video which is kept in memory. With this segment being played, the media player proceeds to load another tiny bit of the video file from the TrueCrypt encrypted into memory. This process is repeated over and over again in what is known as on-the-fly encryption or decryption. This procedure can be applied to all file formats.

Strictly speaking, TrueCrypt does not save any decrypted data on the disk, rather, it tends to keep it temporarily in RAM [24]. Even at the point when its volume is mounted, the data contained in it remains encrypted. Also, when the operating system is shut down or restarted, the disk is unmounted with the data contained in it encrypted and unavailable. As is common with computers, the power supply to the machine can suddenly be cut off, without allowing for a complete system shutdown. In this instance, the files saved on the disk are still encrypted and unavailable [25]. To make these files available again, the disk needs to be mounted and the correct password or keyfile provided.

3.4.2 Encryption Schemes

To successfully mount a TrueCrypt volume (suppose no cached passwords/keyfiles exist) or you are to perform a pre-boot authentication, these steps have to be carried out:

1) The initial 512 bytes of the volume represent the standard volume header. Out of these 512 bytes, the first 64 bytes are the salt. The salt is usually used in encrypting the system.
2) The final 512 bytes that belong to the first logical drive track are also read into RAM (the Boot Loader of TrueCrypt is stored on the TrueCrypt Rescue Disk and/ or the first track of the system drive).
3) The bytes ranging from 65536-66047 of the volume are also read into RAM. For system encryption to take place, the bytes from 65536-66047 of the initial partition which is located behind the active partition are read. If the possibility of a hidden volume within this volume (or within the partition that comes after the boot partition), its header is considered to have been read at this point, if not, random data is what has just been read (whether a hidden volume exists or not can only be determined by attempting to decrypt this data).
4) At this stage, TrueCrypt will now attempt to decrypt the volume header which was read in Step (1). Any data generated and used during this process of decryption are stored in RAM (At no time does TrueCrypt does save these data onto the disk). Certain parameters are usually unknown, and they will have to be determined by the trial and error process, by running and testing all the existing combinations of the following;

    a. PRF used by the header key derivation function (which can be one of the following:

       HMAC-SHA-512, HMAC-RIPEMD-160, HMAC-Whirlpool.

       When a user enters a password that may have been applied to one or more keyfiles and the salt is read in step (1) which is passed to the header key derivative function, subsequently produces a sequence of values that are used to form the header encryption key and the secondary header key (XTS mode). These keys are what are applied in decrypting the volume header.

    b. The encryption algorithms: AES-256, Serpent, Twofish, AES-Twofish-Serpent, etc.
    c. Mode of operation: XTS, LRW (*deprecated/legacy*), CBC (*deprecated/legacy*)
    d. Key size(s)

5) For decryption to be successful, the first 4 bytes of the decrypted data must contain the ASCII string “TRUE”, and also the CRC-32 checksum of the concluding 256 bytes of the decrypted data, the volume header must match the value which is located at byte #8 of the decrypted data (this value remains unknown to an adversary because it is encrypted). Unless these conditions are met, the process continues from step (3) again, but this time around, the data read in step (2) will be used and not the data read in step (1) like previously (that is, possible hidden volume header). If this process is undertaken and the condition remains unmet, the mounting is terminated, possibly

because of a wrong password, a corrupted volume, or the volume is not a TrueCrypt volume.

6) At this point, we can assume with a very high probability that we possess all of the following; a correct password, the right encryption algorithm, the mode, the key size, and the correct header key derivation algorithm. If the data read in step (2) is successfully decrypted, we can know for sure that we are mounting a hidden volume and its size is retrieved from data read in step (2) decrypted in step (3).
7) The routine of encryption is reinitialized with both the primary master key and the secondary master key, which have been retrieved from the decrypted volume header. These primary and secondary keys can be deployed to decrypt any sector of the volume, excluding the volume header area (or the key data area, for the system encryption), which is usually encrypted using the header keys. The volume is mounted.

Provided there is a situation where the size of the active partition is less than 256MB, the data will then have to be read from the second partition behind the active one. For Windows 7 and later, by default, there is no need to boot from the partition on which they are installed. These parameters are kept secret. In keeping these parameters secret, the aim is not to make attacks more complex, but rather to ensure that TrueCrypt volumes remain primarily unidentifiable, this will not be easily achieved, if the parameters were allowed to be stored within the volume header without any encryption. It is also worth noting that, once a non-cascaded encryption algorithm is used for system encryption purposes, that algorithm is known. This can be determined on the Rescue Disk of TrueCrypt or by also analyzing the contents stored within the first logical drive track of the TrueCrypt Boot Loader which is unencrypted [26]. The master keys will remain unchanged because they were generated when the volume was being created. To change the volume password, we would have to re-encrypt the volume header with the use of a new header key that must also be derived from a new password

4. Results

All the systems used in the experiment demonstrated some level of remanence with or without cooling. There was no uniform decay pattern observed in all the test systems, however, the decay increases as the system stayed longer without power. At operating temperature, which we recorded averagely to be 25°C, decay was rapid in all the systems as compared to the results we obtained when the temperature was reduced.

The test data we experimented with had approximately the same number of 1’s and 0’s. A complete decay would mean all the ones have been zeroed which would evaluate to 50% of the bits changing. A fully decayed memory would mean that all the bits have been zeroed. The error rate is the number of bit errors or BIT_ERROR_COUNT divided by the total bits in the pseudorandom test data.

Table 2 shows the summary of the overall percentage of decay recorded by the DDR and DDR2 systems with respect to time.

Table 2. Summary of decayed bits at different time intervals

Systems	**Memory Type**	**Temperature / °C**	**Time without power / s**	**% decay**
A	DDR	25	60	82.33
		-25	240	2.125
B	DDR2	25	60	80.31
		-25	120	0.975
C	DDR2	25	90	95.27
		-25	300	1.225

The DDR1 and DDR2 chips used in Systems A, B, and C had remanence which could easily be exploited. After extracting the remanence of the RAM in the systems, we compared the bits of the remanence to that of the test data to determine the number of bits that decayed. The fraction of bits recovered correctly is computed by dividing the recovered bits by the total bits in the test data. We found over 97% of the bits in systems B and C to be recovered correctly. Performing the experiment several times increased the efficiency and accuracy by a small margin on the subsequent attempts. The worst result that we attained due to a delay in the transplant gave us 94% of all the bits recovered correctly. This result would as well be considered successful as it is in line with the study of Ref. [3] which developed a tool called *AESfix* which could reconstruct AES keys when 7% of the bits are decayed in a matter of a few seconds.

Fig. 2. is a graphical representation of the percentage decay of systems A, B, and C at operating temperature without any cooling effects applied. There is not much difference in the curves, it was observed there was a constant increase in the decay as the duration prolongs and eventually there was total decay in all three systems. The remanence of the RAM after 4 minutes of power cut without cooling in all the systems experienced total decay and hence the data could not be reconstructed by any known tool as at the time of the study. As stated earlier, there was no specific pattern that could be spotted.

Fig. 3. however, is the graphical representation of the results obtained when the RAM chips were cooled to a temperature averagely recorded as -25°C. It is observed from the diagram that the rate of decay has drastically reduced and hence would afford the investigator more time to carry out his investigation with a more accurate outcome. The results in this diagram shows a high rate of success of our approach, encryption keys were retrieved successfully without reconstruction.

With the results displayed in **Fig. 3**, we hypothesis that when RAM is cooled to lower temperatures and maintained, the data could be preserved for hours with a small amount of decay.

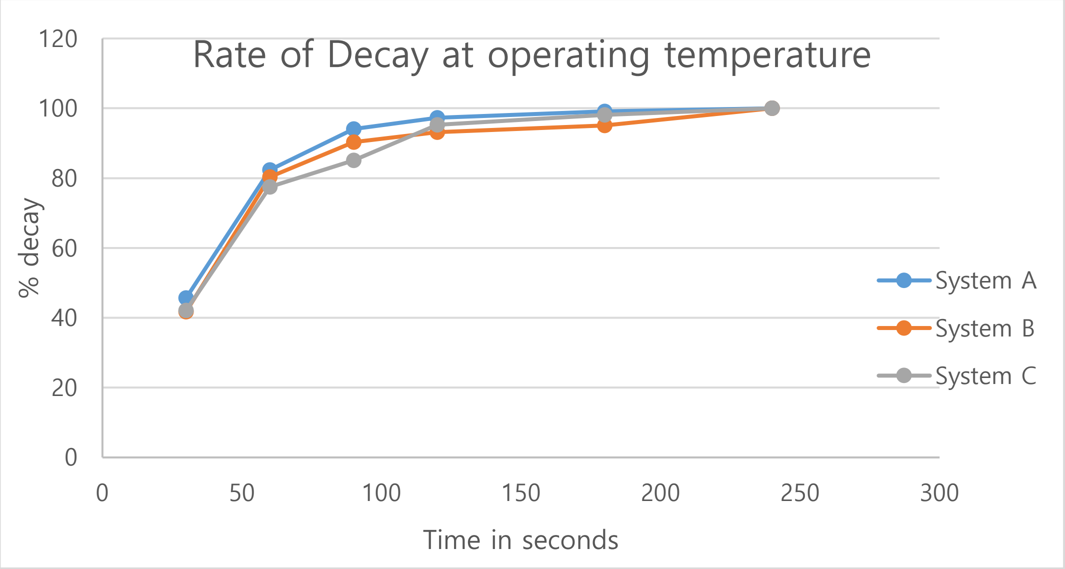


Fig. 2. Rate of decay at operating temperature

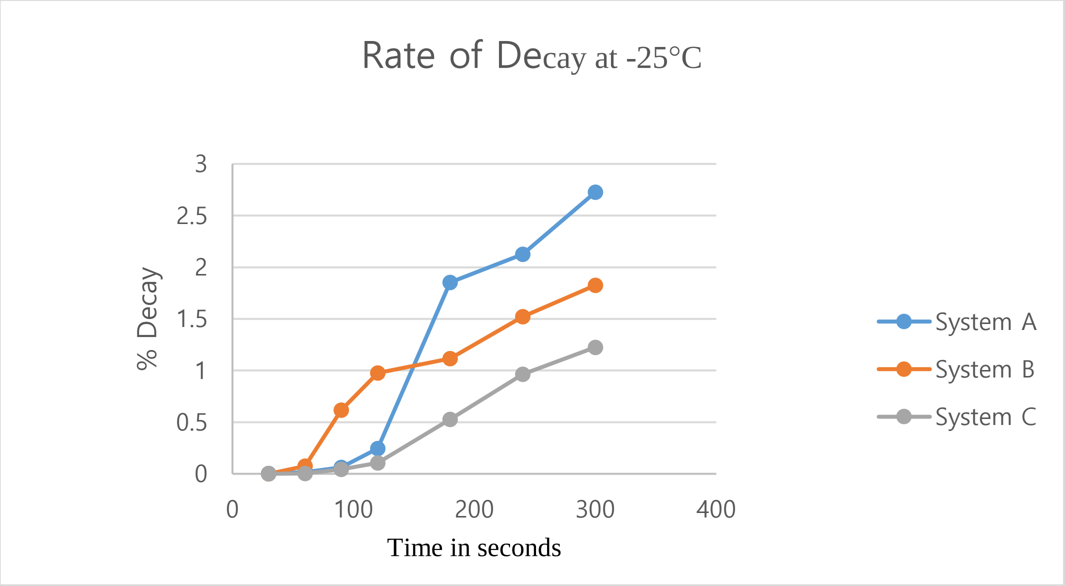


Fig. 3. Rate of decay at a reduced temperature

Table 3 and **Table 4** presents detailed results obtained for the rate of decay at different time intervals at standard operating temperature of 25°C and reduced temperature of -25°C respectively. From these results in **Table 3**, there was total decay in all the system within 4 minutes of power cut at operating temperature. For the first 30 seconds, almost half of the bits were decayed in all the systems, the decay slowed down after 60 seconds as the memory approaches its ground state. On the other hand, at a reduced temperature of -25C, there was less than 1% of decayed bits after 2 minutes in all the systems as indicated in **Table 4**.

Table 3. detailed result of decayed bits at 25°C with varying time.

TIME	System A	System B	System C
30	45.670	41.752	42.125
60	82.325	80.305	77.525
90	94.125	90.325	95.267
120	97.267	93.175	98.15
180	99.115	95.075	100.000
240	100.000	100.000	100.000
300	100.000	100.000	100.000

Table 4. Detailed result of decayed bits at -25°C with varying time.

TIME	System A	System B	System C
30	0.002	0.000	0.000
60	0.015	0.075	0.002
90	0.061	0.618	0.045
120	0.244	0.975	0.105
180	1.854	1.115	0.528
240	2.125	1.520	0.965
300	2.726	1.825	1.225

4.1 Key Identification

To retrieve the encryption keys from the memory dump, tools were developed in C to automate the process and enhance the rate of locating the keys. These tools work based on two proposed algorithms; retrieving keys using key schedules in memory and retrieving keys from memory using linear scan. The containers used in this study were encrypted with TrueCrypt which implemented the AES encryption scheme.

4.1.1 Retrieving Keys Using Key Schedules

This tool uses the key schedule in memory to locate the AES keys from the dump, it does this by testing a series of bytes sequentially to check if it decrypts a known text correctly. The tool takes the memory image as its input and works based on the algorithm below:

i. Loop through all the bytes of the memory image. Treat either the block having 176 bytes or 240 bytes as that of the AES key schedule.
ii. Compute the Hamming distance for every word in the potential key schedule, from that word to the key schedule word that should have been generated from the surrounding words.
iii. If the total number of bits violating the constraints on a correct AES key schedule is sufficiently small, output the key.

For efficiency, it also performs a simple entropy test to filter out blocks that are not keys. It counts the number of repeated bytes and skips blocks that have too many repeats; Returns true if the 176 bytes starting at location bmap[i] contain more than 8 repeats of any byte. This is a primitive measure of entropy, but it works well enough. The function keeps track of a sliding window of byte counts.

This method works even if few bits of the key schedule have been corrupted due to memory decay.

4.1.2 Retrieving Keys Using Linear Scan

This tool uses pattern matching technique to identify keys in memory. When an encrypted container is created during the initial setup, the TrueCrypt graphical interface displays portions of the keys used to encrypt the container. TrueCrypt's open-source nature facilitated the development of pattern matching. The tool scans the whole memory, treating each 48-byte block as Master Keys and Tweak Keys in a predetermined sequence. The tool only decrypts the first block of the container's data region because it is all that is required to ascertain whether they are the correct keys or not, allowing for substantially faster implementation.

The approach used by this tool can be categorized into three distinct stages:

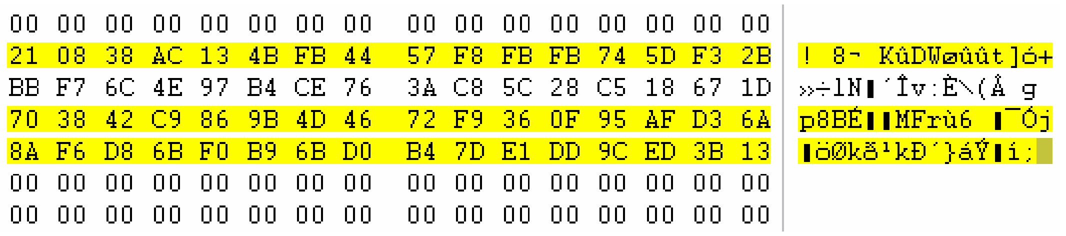
i. Determine the pattern of the way in which the keys are stored in memory.
ii. Identify known plaintext in the container in order to easily find the correct keys for decryption.
iii. Decrypt the container from only the master key.

The tool repeats the process automatically for all possible keys in the memory dump. It basically scans the memory image linearly checking every position from the image and recovering possible keys using an identified pattern and trying to decrypt the container.

4.2 Recovery of Text and Encryption Keys

The tools developed recovered the keys successfully from the memory dump they were fed with. The tool that implemented the linear scan algorithm retrieved the key from a 2GB dump within 5 minutes whiles the tool using the key schedule algorithm retrieved the key from the same dump within 8 minutes. The linear scanning tool was provided a dump that does not contain an encryption key, it scanned the whole dump and returned there was no key found with at an approximate rate of 23MB per seconds.

Using the Linear Scanning tool which employ pattern matching, the known encryption keys were found in the memory dump and displayed a clearly recognizable pattern in memory, as illustrated in **Fig. 4**. Experiments conducted revealed that the lower 256-bit block is the container's Master Key and the first 128-bit block is the Tweak Key.



```
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
21 08 38 AC 13 4B FB 44  57 F8 FB FB 74 5D F3 2B  ! 8¬ KûDWøûût]ó+
BB F7 6C 4E 97 B4 CE 76  3A C8 5C 28 C5 18 67 1D  »÷lN▌´Îv:È\(Å g
70 38 42 C9 86 9B 4D 46  72 F9 36 0F 95 AF D3 6A  p8BÉ▌▌MFrù6 ▌¯Ój
8A F6 D8 6B F0 B9 6B D0  B4 7D E1 DD 9C ED 3B 13  ▌öØkð¹kÐ´}áÝ▌í;
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Fig. 4. Portion of memory indicating primary and secondary master keys

It was observed that the header key was not located in memory. This was because TrueCrypt only needs the header key to decode the container header and extract the Master Keys, which are then used to decrypt the rest of the container. As a result, once the Master Keys are in memory, the header key may be deleted. This validated that the Master Keys were fully stored in the system's memory and, more crucially, detected the offset between the two keys. However, it was required to first develop a mechanism to decrypt the container using keys recovered from memory, and then test for successful decryption. During a typical TrueCrypt operation, the string 'TRUE' is used to indicate that the header has been correctly decrypted. The Master and Tweak keys are taken from the header and are known to be valid, and they are used to decrypt the contents in the container [27]. Offsets 3-7 of a decrypted 10 Megabyte FAT formatted container, decoded to ASCII 'MSDOS'. These correspond to the encrypted container offsets 515-519. (Skipping 0-512 which is the TrueCrypt header encrypted by the Header Key and not accessible). Because the string 'MSDOS' was found at offsets 3-7 and AES decrypts in 128-bit blocks, the known plaintext sits in the first block of the encrypted container's data area and was only decrypted.

5. Discussion

Investigation of digital media is gaining much popularity in the adjudication of some criminal cases. The admissibility of digital evidence in court requires that the forensic process is repeatable and the outcome verifiable. Investigation of digital media faces a major hurdle when an encrypted drive is encountered or when the content of the media is encrypted. However, the idea of cold boot discussed by Halderman et. al. [3] in their research gave us the idea to exploit the remanence of the RAM to enhance the digital forensic process. Access to encryption keys or passphrases is very essential in digital forensic investigations. These keys are sometimes stored on the hard drive or other storage media, or even written on materials in the physical surrounding of the computer. Even though computer users in an attempt to increase the security of their computer systems employ encryption, some still choose weak keys or passphrases for easy remembrance. Leveraging the balance between the strength and usability of a password plays to the advantage of the investigator. Most users tend to choose usability over strength, some users write their passwords on pieces of paper or other materials in order not to lose them. Other users, to make the password easy to remember, use passwords that are easy to guess; passwords that have personal meaning such as date of birth, names, phone numbers, etc. When passwords are carefully chosen and dictionary words are avoided, it becomes difficult to access passphrases for digital forensic investigation to be done [28].

Some encryption software writes the keys to the RAM which could get swapped onto the disk or sometimes to a temporary file on the disk. To retrieve the keys in such an instance, some automated tools such as AccessData FTK. Passware Forensic Kit could be used to generate a list of all keywords from the drive. This approach is limited when full disk encryption is used.

Being able to overcome encryption when it is encountered in a digital forensic investigation is an advancement to the digital forensic process. In the quest to defeat full disk encryption when conducting investigations, the RAM was transplanted from the target computer system into the host system and a dump of the memory was acquired in order to capture “the state” of the target system before the transplant. This technique was developed to exploit the remanence of the RAM chips and also evade the Platform Reset Attack Mitigation Specification by Trusted Computing Group (TCG).

The proposed exploit method of transplanting the RAM module is very efficient in cases the investigator is not able to boot the target machine with the imaging tools because the target computer might have implemented some anti-forensic techniques that erase or destroy the content of the RAM while booting. Most current computer systems, manufacturers have implemented the attack mitigation specifications mandated by TCG in their 2019 release [14] which does not allow for most cold boot attack techniques to be carried on them.

Decay was slower when the memory modules were cooled before turning off the computer or transplanting the memory. Cold boot attacks generally are ineffective when the computer system is properly shut down and the memory reset [29].

Contrary to the popular notion of the volatility of RAM that it loses its data or content immediately after the computer losses power, this study established that the content persists in RAM for seconds, or even minutes before it decays gradually. The duration with which data persist in the RAM after power cut can be prolonged by reducing the temperature of the RAM. From this study, it is noted that when the temperature is reduced, the content can be retrieved by transplanting the RAM into another computer by following the method proposed. Encryption keys and other volatile data could as well be retrieved from the RAM dump which can helps with forensic investigations.

Retrieving the content from a system’s memory by cooling it is possible because the content of modern memory is refreshed after every millisecond and the data stays in the memory between those refreshes. To retrieve this data, the memory should be cooled significantly into the negatives. The refreshes take longer when the RAM is chilled.

Even though we retrieved the encryption keys from a TrueCrypt encrypted container, we believe similar encryption software would also be vulnerable to this attack as they all store the encryption keys in memory. TCG technologies are not able to protect keys residing in memory

6. Limitations

One of the limitations of the technique we proposed and implemented is that the investigator must have physical access to the target computer while it is still on or immediately it has been turned off. When the target system is dead for a long period before the arrival of the investigator, he/she may not be able to retrieve any data from the RAM as the system is not running and hence does not have “a current state”. Also, when a culprit prevents the investigator from physical access to the computer, the investigator would have no means of implementing the proposed technique to retrieve volatile data from the computer memory.

Another limitation to the implementation of the proposed key recovery method is that the implemented approach was limited to AES in LRW mode which was used by TrueCrypt and it has only been demonstrated with the default mode of encryption for TrueCrypt. However, it will be possible to add the algorithm to decrypt containers or volumes that are encrypted with other encryption algorithms.

Finally, to be able to successfully implement the proposed technique of transferring the RAM from the target system to the host system, the memory controllers in both systems must be compatible.

7. Conclusion

In as much as computer manufactures implement various counter measures to mitigate cold boot attacks, the content of the RAM is still vulnerable to the cold boot attack technique of transplanting the RAM as we demonstrated in this study. This could be exploited by the digital forensic community to enhance the access to digital evidence when conducting investigations. When the memory chip is cooled before transplant, there's a greater chance of recovering cryptographic materials without decay.

Acknowledgement

We acknowledge Mr. Nathaniel Frimpong Danquah, Mr. Benjamin Lartey and Mr. Benjamin Partey, all of whom are Lab Technicians at the Department of Computer Science, College of Science, Kwame Nkrumah University of Science and Technology, Ghana.

References

[1] Y. Shah, "Forensic Analysis of Volatile Memory for Non-string Data," 2017.

[2] M. Gruhn, "Forensically sound data acquisition in the age of anti-forensic innocence," 2016, [Online]. Available: https://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/7938.

[3] J. Alex Halderman *et al.*, "Lest we remember: Cold boot attacks on encryption keys," in *Proc. of 17th USENIX Secur. Symp.*, pp. 45–58, 2008.

[4] K. Hausknecht, D. Foit, and J. Burić, "RAM data significance in digital forensics," in *Proc. of 2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc.*, no. May, pp. 1372–1375, 2015. Article (CrossRef Link)

[5] Periyadi, G. A. Mutiara, and R. Wijaya, "Digital forensics random access memory using live technique based on network attacked," in *Proc. of 2017 5th Int. Conf. Inf. Commun. Technol. ICoIC7 2017*, vol. 1, no. c, 2017. Article (CrossRef Link)

[6] K. Alam, J. Sang, H. Hu, A. Rahman, and M. Alam, "Encryption Data Recover from Memory," *United Int. J. Res. Technol.*, vol. 02, no. 06, pp. 58–66, 2021.

[7] D. Forte, "Do encrypted disks spell the end of forensics?," *Comput. Fraud Secur.*, vol. 2009, no. 2, pp. 18–20, 2019. Article (CrossRef Link).

[8] A. Case and G. G. Richard, "Memory forensics: The path forward," *Digit. Investig.*, vol. 20, pp. 23–33, 2017. Article (CrossRef Link).

[9] C. Hilgers, H. Macht, T. Muller, and M. Spreitzenbarth, "Post-mortem memory analysis of cold-booted android devices," in *Proc. of 8th Int. Conf. IT Secur. Incid. Manag. IT Forensics, IMF 2014*, pp. 62–75, 2014. Article (CrossRef Link).

[10] C. Hargreaves and H. Chivers, "Recovery of encryption keys from memory using a linear scan," in *Proc. of ARES 2008 - 3rd Int. Conf. Availability, Secur. Reliab. Proc.*, no. March 2008, pp. 1369–1376, 2008. Article (CrossRef Link).

[11] C. Maartmann-Moe, S. E. Thorkildsen, and André Årnes, "The persistence of memory: Forensic identification and extraction of cryptographic keys," *Digit. Investig.*, vol. 6, no. SUPPL., pp. 132–140, 2009. Article (CrossRef Link).

[12] I. Zimerman, E. Nachmani, and L. Wolf, "Recovering AES Keys with a Deep Cold Boot Attack." 2021. Article (CrossRef Link).

[13] R. Carbone, C. Bean, and M. Salois, "An in-depth analysis of the cold boot attack - Can it be used for sound forensic memory acquisition ?," *Memory*, no. January, 2011.

[14] Trusted Computing Group, "TCG PC Client Platform Reset Attack Mitigation Specification," 2019.

[15] M. Gruhn and T. Müller, "On the practicability of cold boot attacks," in *Proc.of 2013 Int. Conf. Availability, Reliab. Secur. ARES 2013*, pp. 390–397, 2013. Article (CrossRef Link).

[16] N. Mainardi, A. Barenghi, and G. Pelosi, "Plaintext recovery attacks against linearly decryptable fully homomorphic encryption schemes," *Comput. Secur.*, vol. 87, p. 101587, 2019. Article (CrossRef Link).

[17] B. Kaplan, “RAM is Key: Extracting Disk Encryption Keys From Volatile Memory,” p. 20, 2017.

[18] N. Syazwani and A. Kahar, “THE ADMISSIBILITY OF DIGITAL DOCUMENT AS EVIDENCE UNDER MALAYSIAN CIVIL COURT,” vol. 2021, no. ICoMM, pp. 248–257, 2021.

[19] S. Abdullah Kahar, A. F. Wan Ismail, A. S. Baharuddin, and L. Abdul Mutalib, “Requirement That Needed To Admit The Digital Document As Evidence In Syariah Court,” in *Proc. of 8th Int. Conf. Manag. Muamalah 2021 (ICoMM 2021)*, vol. 2021, no. ICoMM, pp. 2756–8938, 2021.

[20] F. M. Granja and G. D. R. Rafael, “The preservation of digital evidence and its admissibility in the court,” *Int. J. Electron. Secur. Digit. Forensics*, vol. 9, no. 1, pp. 1–18, 2017. Article (CrossRef Link).

[21] F. M. Ghabban, I. M. Alfadli, O. Ameerbakhsh, A. N. Abuali, A. Al-Dhaqm, and M. A. Al-Khasawneh, “Comparative analysis of network forensic tools and network forensics processes,” in *Proc. of 2021 2nd Int. Conf. Smart Comput. Electron. Enterp. Ubiquitous, Adapt. Sustain. Comput. Solut. New Norm. ICSCEE 2021*, pp. 78–83, 2021. Article (CrossRef Link).

[22] J. Seo, S. Lee, and T. Shon, “A study on memory dump analysis based on digital forensic tools,” *Peer-to-Peer Netw. Appl.*, vol. 8, no. 4, pp. 694–703, 2015. Article (CrossRef Link).

[23] M. A. Alomari, K. Samsudin, and A. R. Ramli, “A study on encryption algorithms and modes for disk encryption,” in *Proc. of 2009 Int. Conf. Signal Process. Syst. ICSPS 2009*, pp. 793–797, 2019. Article (CrossRef Link).

[24] Truecrypt Foundation, “TrueCrypt User Guide,” *System*, 2016.

[25] M. V. Ball, C. Guyot, J. P. Hughes, L. Martin, and L. C. Noll, “The XTS-AES Disk Encryption Algorithm and the Security of Ciphertext Stealing,” *Cryptologia*, vol. 36, no. 1, pp. 70–79, 2012. Article (CrossRef Link).

[26] M. Broz and V. Matyas, “The trueCrypt on-disk format - An independent view,” *IEEE Secur. Priv.*, vol. 12, no. 3, pp. 74–77, 2014. Article (CrossRef Link).

[27] L. Wilke, J. Wichelmann, M. Morbitzer, and T. Eisenbarth, “SEVurity: No security without integrity: Ng integrity-free memory encryption with minimal assumptions,” in *Proc. of IEEE Symp. Secur. Priv.*, pp. 1483–1496, 2020. Article (CrossRef Link).

[28] J. Aumasson, *Serious Cryptography*, No Starch Press, Inc., 2018.

[29] P. McGregor and T. Hollebeek, “Braving the cold: New methods for preventing cold boot attacks on encryption keys,” *Black Hat Secur. …*, 2014, [Online]. Available: http://www.crazylazy.info/cons/bh08/attach/BH_US_08_McGregor_Cold_Boot_Attacks.pdf.

Twum Frimpong received his Ph.D. degree in Computer Science from the Kwame Nkrumah University of Science and Technology (KNUST). Prior to that he received M.Sc. degree in Information Systems from Roehampton University, U.K. and also M.Sc. and B.Sc. degrees in Internet Engineering and Electrical and Electronic Engineering respectively from London South Bank University, U.K. He is currently a Senior Lecturer with the Department of Computer Science, KNUST. His research interests include Computer Networks Security, Machine Learning and Computer Vision.

Emmanuel Mawuli Lagoh studies Cybersecurity and Digital Forensics (MPhil) in Kwame Nkrumah University of Science and Technology in Ghana, after he attained BSc. Degree in Computer Science in the same institution. He worked as a Teaching and Research assistant for two years whiles on the Graduate programme. He currently works as a security analyst in a banking institution. His research interests include computer and network security, computer forensics and cyber criminology.

Yaw Marfo Missah obtained his Ph.D. in Computer Science from Colorado Technical University, USA and his MSc. degree in Information Technology from Clark University, USA. He is currently a senior lecturer at the Department of Computer Science KNUST. Dr. Missah's research interest includes Network systems and AI applications.

Najim Ussiph received the B.Sc. and M.Sc. degrees in computer science from the University of Ibadan, Ibadan, Nigeria, in 1987 and 1993, respectively, and the Ph.D. degree from the University of Salford, Manchester, in 2015. He has over 15 years of experience in teaching and research at Polytechnic and University level. He has been a Faculty Member with the Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, since April 2001, and has taught several courses at both undergraduate and graduate levels. His research interests include information systems and e-learning and learning

Emmanuel Ahene received the M.Eng. and Ph.D. degrees in computer science and technology from the University of Electronic Science and Technology of China. He is currently a Lecturer with the Department of Computer Science, Kwame Nkrumah University of Science and Technology, Ghana, and also a Co-Founder of Cyberpassconsult, an international cybersecurity consultancy firm. His research interests include information security and machine learning.