Amnesiac DRAM: A Proactive Defense Mechanism Against Cold Boot Attacks

Hoseok Seol, *Student Member, IEEE*, Minhye Kim, *Student Member, IEEE*, Taesoo Kim, *Member, IEEE*, Yongdae Kim, *Member, IEEE*, and Lee-Sup Kim, *Fellow, IEEE*

Abstract—DRAMs in modern computers or hand-held devices store private or often security-sensitive data. Unfortunately, one known attack vector, called a cold boot attack, remains threatening and easy-to-exploit, especially when attackers have physical access to the device. It exploits the fundamental property of current DRAMs: remanence effects that retain the stored contents for a certain period of time even after powering off. To magnify the remanence effect, cold boot attacks typically freeze the victim DRAM, thereby providing a chance to detach, move, and reattach it to an attacker's computer. Once power is on, attackers can steal all the security-critical information from the victim's DRAM, such as a master decryption key for an encrypted disk storage. Two types of defenses were proposed in the past: 1) CPU-bound cryptography, where keys are stored in CPU registers and caches instead of in DRAMs, and 2) full or partial memory encryption, where sensitive data are stored encrypted. However, both methods impose non-negligible performance or energy overheads to the running systems, and worse, significantly increase the hardware and software manufacturing costs. We found that these proposed solutions attempted to address the cold boot attacks *passively*: either by avoiding or by indirectly addressing the root cause of the problem, the remanence effect. In this article, we propose and evaluate a *proactive* defense mechanism, Amnesiac DRAM, that comprehensively prevents the cold boot attacks. The key idea is to discard the contents in the DRAM when attackers attempt to retrieve (i.e., power on) them from the stolen DRAM. When Amnesiac DRAM senses a physical separation, it locks itself and deletes all the remaining contents, making it *amnesiac*. The Amnesiac DRAM causes neither performance nor energy overhead in ordinary operations (e.g., load and store) and can be easily implemented with negligible area overhead in commodity DRAM architectures.

Index Terms—Cold boot attack, DRAM, hardware defense, self erasing memory

1 Introduction

DRAMs have been used as a main memory in modern computer systems. Conceptually, DRAM is volatile, meaning that it maintains stored contents only when power is supplied. Because of its volatility, many people tend to believe that the DRAM is a safe storage medium for security- and privacy-sensitive data; when it is physically disconnected from the motherboard (e.g., from a stolen laptop), the contents would not be recoverable as the power is cut off.

However, in practice, the contents stored in DRAMs are not immediately erased even when the power is gone, thanks to the *remanence effect*. In particular, under cold temperatures, the contents remain restorable for a certain duration, giving an attacker enough time to detach, move, and reattach the stolen DRAMs to the attacker's systems [1], [2]. For example, Halderman et al. could successfully restore 99.8 percent of the contents in a DRAM even after an hour without power supply at -50°C [2].

Such an attack, known as a *cold boot attack*, is typically conducted in three steps: 1) freezing a victim's DRAM module, 2) detaching, moving and installing it to the attacker's system, 3) and restoring sensitive data from the DRAM. Since the first public demonstration [2], numerous attacks have been shown to steal various kinds of sensitive data against real devices, for example, DDR2/3 memory [3], [4], [5], hand-held devices [6], [7], and recently, even against DDR3/4 memory with a data randomizing scheme called scrambling [8], [9].

There are broadly two types of defenses proposed to mitigate cold boot attacks. First, CPU-bound cryptography is a software solution that utilizes registers or caches of the processor for storing sensitive data [10], [11], [12], [13], [14], [15], [16], [17]. Although it provides sound protection on the carefully selected data (e.g., cryptographic keys), it fails to protect other valuable contents (e.g., recent e-mails, photos and visited websites [6]) and, worse, requires modification of the protected software. Second, full memory encryption schemes [9], [18] can also provide comprehensive protection (i.e., simply encrypted) against cold boot attacks and potentially support a larger amount of a user's data. Unfortunately, this approach has two fundamental limitations: performance and energy overheads (i.e., en/decryption on

- *H. Seol is with the DRAM Development Division, Samsung, Hwasung, Gyeonggido, South Korea. E-mail: seolhs@mvlsi.kaist.ac.kr.*
- *M. Kim is with the Law School, Seoul National University (SNU), Seoul 08826, South Korea. E-mail: scv9595@gmail.com.*
- *T. Kim is with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA 30332 USA. E-mail: taesoo@gatech.edu.*
- *Y. Kim and L. Kim are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea. E-mail: {yongdaek, leesup}@kaist.ac.kr.*

Manuscript received 6 Mar. 2019; revised 15 July 2019; accepted 28 Aug. 2019. Date of publication 8 Oct. 2019; date of current version 11 Mar. 2021. (Corresponding author: Hoseok Seol.)
Recommended for acceptance by P. Barreto.
Digital Object Identifier no. 10.1109/TC.2019.2946365

every memory access) and manufacturing costs (i.e., an encryption engine in the memory controller).

We argue that these solutions attempt to address the cold-boot attack *passively*; this means that they are either avoiding or indirectly addressing the root cause of the problem, the remanence effect. For example, CPU-bound cryprography avoids the problem simply by not putting the content to the memory, and the full memory encryption focuses on preventing the restoration of the sensitive data after an attacker has access to the stolen user's contents (i.e., the step 3).

In this paper, we propose a *proactive* yet practical defense scheme against cold boot attacks, called Amnesiac DRAM, that can deter attackers from obtaining a victim's data. The key idea is to provide the protection inside the DRAM itself, which can reliably sense either physical separation or reconnection events (e.g., a power-off or power-on) and protectively delete all data while locking attackers from accessing it. It is worth noting that Amnesiac DRAM attempts to eliminate the root cause of cold boot attacks, the remanence effect, and make it truly volatile again, thereby providing a much more comprehensive security guarantee.

There are several challenges that we must overcome to design the Amnesiac DRAM in a practical manner. First, we need a power- and cost-effective solution to reliably sense any physical separation from the DRAM module. Second, the erasing operation should be done either without any power source (i.e., between detaching and attaching) or by safely interposing over the course of the cold boot attack (i.e., after attaching)— no risk of leaking the stolen data. Lastly, these operations and required modifications should not impose any performance or energy overhead in performing ordinary memory operations.

To overcome these challenges, we leverage the commodity DRAM architecture as much as possible: we utilize an existing power detector logic for sensing on/off states, and embed a locking logic to safely prohibit the DRAM accesses while performing the erase operation. In addition, Amnesiac DRAM introduces a new DRAM component, called MemSweep, which can rapidly erase the contents by exploiting the inner structure of DRAMs— it takes 2.8 ms (351 times faster than the conventional WRITE command) for erasing the contents of a 8 Gb DRAM chip [19].

Compared to previous approaches, Amnesiac DRAM incurs no performance and energy overhead to the running system; it interposes only at the standard initialization phase when the power is supplied. It increases, however, at one time, the system booting time by at most 2.8 ms, which is negligible considering the boot time cost in modern devices. Amnesiac DRAM is also practical, as it requires minimal changes of the commodity DRAM architecture with little chip overhead for implementation. We, therefore, believe Amnesiac DRAM is an effective, practical way to fundamentally defend against the cold boot attack.

In summary, this paper makes the following contributions.

- To solve the cold boot attack, we introduce the first DRAM-side solution with negligible performance, energy, and chip area overhead.
- We design MemSweep, which can quickly erase the contents by utilizing the inner structure of DRAMs.
- We evaluate the operational energy and initialization latency/energy of Amnesiac DRAM. In the evaluation, Amnesiac DRAM does not consume operational energy, while the prior memory encryption scheme [9] increases the DRAM system energy by up to 25.2 percent. Also, MemSweep is 351 times faster and consumes 126 times less energy than initialization using WRITEs.
- We show that Amnesiac DRAM can be implemented with a negligible chip area overhead by designing the required circuits using the 65 nm process.

2 Background

In this section, we first describe the technical background [20], [21] of DRAM required to understand the operation of Amnesiac DRAM. We then review technical details about the cold boot attack.

2.1 DRAM Basic Operations

Storage Principle. DRAMs store information in capacitors in the form of charge. The charged capacitor means 1 and the discharged one means 0. DRAM cells have various charge leakage paths, so the amount of charge stored in cells gradually changes. To compensate for the charge leakage, DRAMs perform periodic REFRESHes. Without the power supply, DRAMs cannot perform REFRESH, so the data disappears. This is why DRAM is called volatile memory.

However, the cell data does not disappear immediately after the power-off. Since the REFRESH period is determined based on the retention time of leaky cells, a sufficient amount of charge remains in DRAM cells in general. As a result, DRAM cells can tolerate charge leakage for a while without REFRESH (i.e., without power), thereby showing the DRAM remanence effect. At cold temperatures, the amount of charge leakage is tiny so that DRAM cells can retain their contents for long periods of time.

Cell Array. As shown in Fig. 1a, a DRAM cell consists of a capacitor and a transistor acting as a switch. A cell transistor is turned on when accessing the cell. The word line is a control line for the cell transistors. We call a row a group of cells connected to a single word line. A bit line is a data line that transfers the cell data. When the cell transistor is turned on, cell capacitors and corresponding bit lines are connected, and data can be written to or read from the cells.

Bit lines are connected to corresponding sense amplifiers. The sense amplifier performs the following functions simultaneously: 1) senses the cell data and 2) restores the cell charge. For the memory controller to access the cell, the cell data must first be loaded into the sense amplifiers. Near the sense amplifier is a switching transistor that connects the bit line to the bit line equalization power (BLEQ power). These switches are called precharge switches and set the bit line voltage to VBLEQ (voltage level of BLEQ power) to prepare the next operation of the sense amplifiers.

Primary Operations. There are three steps to read or write data to a DRAM cell. The first sequence is to move the data stored in cells into sense amplifiers, called ACTIVATE. The ACTIVATE operation enables the word line of the target row and turns on the sense amplifiers. Then, the sense amplifiers read the data by sensing the amount of charge

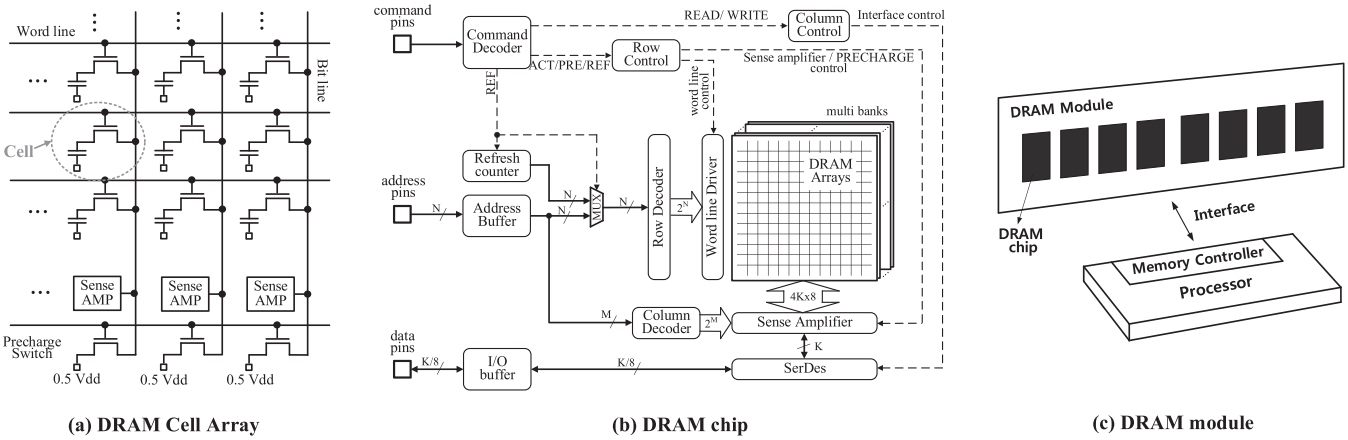


Fig. 1. DRAM basic structure.

stored in cell capacitors (more than half: 1, less than half: 0). At the same time, the sense amplifiers fully restore the cell charge.

The second step is to read/write data from/to sense amplifiers, called READ/WRITE. In READ/WRITE, data travels on- and off-chip data bus, between the sense amplifiers of the DRAM and the data queue of the memory controller.

The third step is to reset sense amplifiers and prepare the next ACTIVATE, called PRECHARGE. In the PRECHARGE operation, the word line is set to a low voltage to separate cells from a bit line. After that, the sense amplifiers are turned off and the PRECHARGE switches are turned on to make the voltage of bit lines VBLEQ.

In addition, DRAM needs periodic REFRESH operations to maintain its contents. Modern DRAMs keep contents through 8 K (8,192) REFRESH commands for 64 ms [22]. Therefore, DRAM refreshes $\frac{\text{density}}{8192}$ cells whenever one REFRESH is executed. The internal operation of REFRESH is the same as the continuous sequence of ACTIVATE and PRECHARGE operations. As a result, the data in DRAM cells is fully restored by using the sense amplifiers. The required time to execute a REFRESH operation is called tRFC.

Chip Structure. The overall structure of the DRAM chip consists of DRAM arrays and peripheral circuits to operate it, as shown in Fig. 1b. The DRAM array consists of several banks. Peripheral circuits can be divided into the address, data, and control path.

The address path provides the row and column addresses to the cell array. The memory controller sends the row address with the ACTIVATE command and the column address with the READ/WRITE command to the address pins of DRAM. Addresses coming into the address pins are converted to digital information through the Address Buffer and sent to the Row and Column Decoders.

Modern DRAMs determine the order of rows to be refreshed by themselves. To do this, DRAMs have a REFRESH counter that points to the address of the row to be refreshed. The value of the REFRESH counter increments with each REFRESH operation. At REFRESH, the address path transfers the value of the REFRESH counter to the Row Decoder instead of the externally transmitted address.

The data path transfers data between the sense amplifiers and the memory controller. The memory controller reads/writes 32 bit (x4 DRAM), 64 bit (x8 DRAM), or 128 bit (x16 DRAM) of data from/to a DRAM chip in a READ/WRITE. To reduce the number of pins, SerDes (Serializer/Deserializer) converts 128/64/32 bit internal data into the series of 16/8/4 bit off-chip data or vice versa.

The control path is represented by the dotted line in Fig. 1b and acts to operate the cell array and the peripheral circuits at an appropriate time. Command Decoder generates internal micro-operations (ACTIVATE, PRECHARGE, REFRESH, etc.) depending on the input of command pins. The Row Control generates the array control signals required for ACTIVATE, PRECHARGE, and REFRESH command execution. These row control signals enable the word line and activate the sense amplifier/equalizer at the appropriate time. The Column Control generates the control signals required for READ/WRITE command execution.

Module Structure. One DRAM module is composed of several DRAM chips (Fig. 1c). For example, a 16 GB module can consist of sixteen 8 Gb DRAM chips. In the general purpose system, the DRAM module can be attached/detached on the motherboard, and the user can freely change the DRAM module. The DRAM module attached on the motherboard is connected to the memory controller. The memory controller is typically built into the processor in recent years.

2.2 Cold Boot Attack

A cold boot[1] attack is a physical attack that steals memory contents on screen-locked computers or hand-held systems. For a cold boot attack, the attacker reboots the target system and executes malicious memory dump code instead of a regular booting sequence. Then, the memory dump code transfers the DRAM contents to the attacker; thus the attacker accomplishes her purpose.

Halderman et al. [2] proposed three rebooting methods for the cold boot attack. The first one uses the warm boot, a rebooting procedure of operating systems, in the victim computer system. To perform a memory dump procedure, the attacker can use USB or network booting.

The second method executes a cold boot in the victim computer system through a power on/off switch. For the second method, the power supply of the memory module

1. The definition of cold boot is a booting method to remove the power to the system and resupply it.

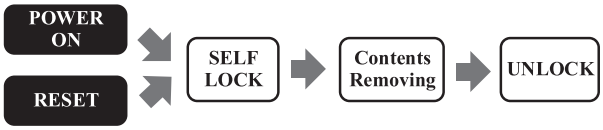


Fig. 2. Basic operation of Amnesiac DRAM .

disappears for a while, but the DRAMs have a high probability of retaining the contents because of the remanence.

The third method physically moves the DRAM module to the attacker's system and then steal the contents. Specifically, this approach opens the case of the victim system and physically separates the memory module from the motherboard. Then, it connects the memory module to the attacker's system and executes a procedure to dump the memory contents. With this method, the power in the DRAM module is lost while moving the module. To minimize data loss during the module transfer, this method uses compressed air or liquid nitrogen to freeze the DRAM module.

Halderman et al. noted that there might be countermeasures for the first and second methods. Defense against the first method is to update the operating system to erase the memory contents before the warm boot. Defense against the second method is to set BIOS to remove the contents at booting time.[2] However, the third method physically separates the DRAM module from the system suddenly, so it does not give OS or BIOS the chance to delete contents.[3] Therefore, this third attack method is the hardest to defend against.

3 Overview

This section introduces our threat model and describes the overall operation as well as the design of Amnesiac DRAM.

3.1 Threat Model

The primary goal of Amnesiac DRAM is to prevent a cold-boot attack. We assume that the attacker uses one of the three booting scenarios in Section 2 to steal contents in DRAM: 1) warm-boot in the victim system, 2) cold boot in the victim system, and 3) freezing and transferring the module to the attacker's system. In this paper, we try to defend against these three attack methods. Access to memory content due to OS compromise is out of scope for this paper.

3.2 Key Operations

To defend three cases of the boot attack mentioned in the threat model, the DRAM itself must have a mechanism to protect its contents. The key idea of Amnesiac DRAM is to erase all contents inside the DRAM before the attacker steals information. To do this, Amnesiac DRAM locks itself in a situation where a cold boot attack can occur, as shown in Fig. 2. When the DRAM is in a self-lock state, the command that can access the contents (ACT, PRE, READ, WRITE, REFRESH) is internally blocked, preventing the attacker from stealing the contents. Then, the built-in initialization block of Amnesiac DRAM automatically erases the contents stored in DRAM. After all the contents have been cleared, Amnesiac DRAM unlocks itself.

Amnesiac DRAM senses the two situations shown in Fig. 2: 1) power on and 2) reset. If either of two cases occur, the Amnesiac DRAM enters a self-lock state. The following paragraph describes these cases.

Power-on. In a cold boot situation, the power of the DRAM chip is also removed and then supplied again. Therefore, it is possible to detect a cold boot attack by sensing that power is removed and then supplied back to the DRAM.

RESET. When the DRAM reset pin is set high, the DRAM initializes the state of internal circuits and starts the initialization sequence. According to the DRAM specification [21], the memory controller should set the DRAM RESET pin high during a cold boot for a particular period. Also, DRAM vendors recommend that users reset DRAM in a warm boot case [24]. Therefore, by entering the DRAM into a self-lock state when a high input is applied to the RESET pin, physical attacks using warm/cold boot can be protected.

3.3 Defense Against Each Attack Scenario

In the following, we describe how to defend against the three types of boot attacks mentioned in the attack model. Amnesiac DRAM can efficiently defend against boot attacks in the victim system without OS or BIOS support.

1) Warm Boot in the Victim System. The victim system's memory controller can be designed to reset the DRAM during a warm boot. When the memory controller sets the RESET pin high, Amnesiac DRAM enters the self-lock state. The attacker cannot acquire the memory contents because s/he can access the memory after the contents are erased.

2) Cold Boot in the Victim System. In the case of a cold boot, the power inside the DRAM chip is discharged and then supplied again. As a result, Amnesiac DRAM enters a self-lock state. Also, after the stable power is supplied to the DRAM, the memory controller sets the RESET pin high to start the initialization sequence. Therefore, Amnesiac DRAM will face two cases for entering the self-lock state in a cold boot situation. When the Amnesiac DRAM enters the self-lock state, the contents cannot be read until the data is completely erased, and thus, the attack fails.

3) Freezing and Transferring the DRAM Module to the Attacker's System. If the attacker disconnects the DRAM module from the victim's system, the power connection is disconnected. Since the idle current (i.e., IDD2P, IDD2N) caused by the analog circuits and the leakage of transistors continues to flow in the DRAM chip, without the continuous power supply, the power in the DRAM chip is discharged rapidly.[4] When the attacker attaches the DRAM module to the attacker's system and performs the cold boot, power is supplied to a DRAM chip. As a result, the Amnesiac DRAM enters the self-lock state. Also, if a commercial memory controller is installed in the attacker's system, it will set the RESET pin to high following the DRAM specification, which also causes Amnesiac DRAM to enter the self-lock state.

2. This method can be neutralized by the attack method to reset BIOS password by discharging the CMOS battery [23]

3. Halderman et al. also noted the physical defense to lock the DRAM module with soldering, but this defense method prevents users from updating the DRAM module as needed.

4. We can confirm this with the simple calculation. Assume that the power capacitor is 1.5 [uF] in the DRAM and the DRAM idle current (IDD2P) is 15 [mA] [19]. In this case, the power drops to ground level for 100 ns, ('Q = CV = It', C = 1.5u, V = 1, I = 15m)

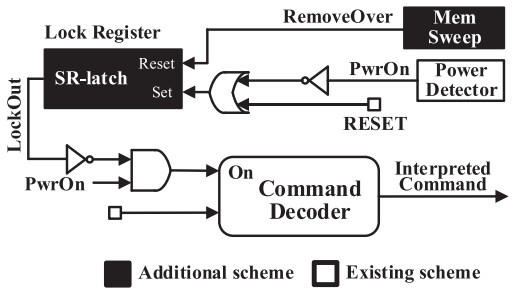


Fig. 3. Block diagram of MemLock.

With these two sensing mechanisms, Amnesiac DRAM can adequately protect against cold-boot attacks.

3.4 Key Mechanisms

Below, we outline key mechanisms for Amnesiac DRAM.

MemLock. MemLock is a scheme that locks the DRAM to prevent the physical attacker from accessing it. The conditions for locking DRAM are 1) power-off/on, 2) RESET. After the DRAM is locked, Amnesiac DRAM automatically removes the contents. After all the contents are cleared, the lock is automatically released to allow user access.

MemLock is implemented using a method that disables the control path in DRAMs. As described in Section 2, the Command Decoder is a circuit for converting the externally transmitted command to the micro-operation at the begin of the control path. Therefore, disabling it prevents the execution of the commands. MemLock locks the DRAM by blocking the input to the Command Decoder.

MemSweep. Modern personal electric devices requires short booting time (general computing system: around 20 sec [25], [26], embedded system: around 3 sec [27]) and the system developers are constantly striving to further reduce the booting time. However, Amnesiac DRAM increases the system booting time because it erases the contents in the initialization sequence. Thus, the long contents removing time can be a major obstacle to the spread of Amnesiac DRAM. Therefore, we propose MemSweep to accelerate the content-removing speed.

Conventionally, the WRITE command was the only way to remove DRAM data. In a x4 DRAM chip, the WRITE command writes 4 bytes of content at a time. It takes about 1 second to erase all the contents of an 8 Gb DRAM chip using the WRITE commands (See Section 5 for calculation). Therefore, using the WRITE command to Amnesiac DRAM causes significant additional booting latency.

To reduce the data-removing time, we propose MemSweep. MemSweep deletes plenty of content (128 KB for the 8 Gb DRAM chip) at once. MemSweep operates by exploiting the existing REFRESH circuits. The REFRESH operation activates several word lines at a time and uses sense amplifiers to restore the cell contents. MemSweep operates identically to REFRESH except that it uses the PRECHARGE switch to charge the cell to the bit line equivalent voltage (VBLEQ) without turning on the sense amplifier. VBLEQ is set to 1/2 VEXT (external voltage level) when the DRAM operates normally. However, for MemSweep, we set VBLEQ to 0 V to make the cell data zero.

MemSweep initializes the 8 Gb DRAM chip within 2.8 ms. Therefore, MemSweep increases the memory content-removing speed by 351 times compared to using the WRITE commands. Please refer to Sections 4 and 5 for implementation and evaluation details.

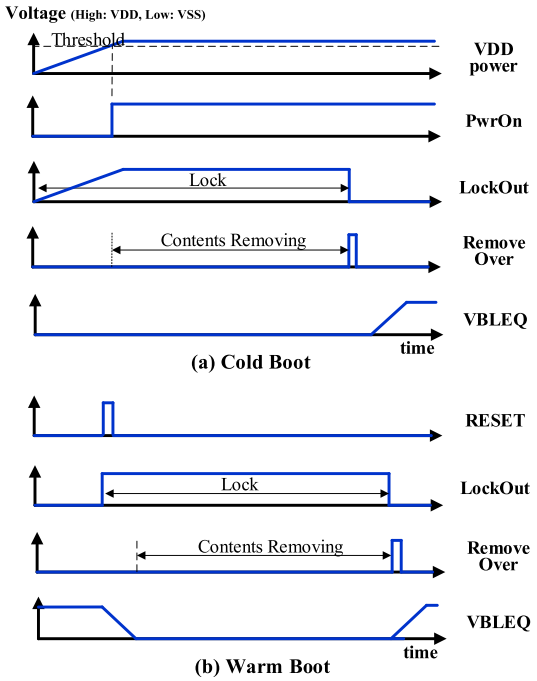


Fig. 4. Timing diagram of MemLock.

Sensing Power Off/On. Since there is no power left in the DRAM module in a cold boot situation, it is necessary to slowly provide power (i.e., ramp up the power voltage level) from the outside. Most of the circuits inside the DRAM are designed to turn on after the power voltage level is ramped up sufficiently. To this end, the DRAM designers embed a power detector inside the DRAM to monitor the power voltage level. Most circuits in the DRAM receive the result of the power detector as a control signal and only operate after a stable power ramp up. To sense the power off/on, Amnesiac DRAM utilizes the existing power detector without adding a new mechanism. When the output of the power detector changes from low (before ramp-up) to high (after ramp-up), the Amnesiac DRAM locks itself and waits for the MemSweep command.

4 Implementation

This section describes the detailed design and implementation of MemLock and MemSweep in Amnesiac DRAM.

4.1 MemLock

Fig. 3 shows a block diagram of MemLock. The uncolored box is circuits existing in conventional DRAMs, and the box colored with black is added circuits for MemLock.

In conventional DRAMs, when the PwrOn signal becomes high, the command decoder operates normally. The PwrOn signal is generated by the power detector when a stable power-on is completed. Fig. 4a shows the operation of the power detector. Since DRAMs have a large number of power capacitors, when DRAMs are supplied with power, the voltage level gradually rises (within 200 ms, [21]), as shown in Fig. 4a. The power detector monitors the increase of this voltage level and makes the PwrOn signal high if the voltage rises

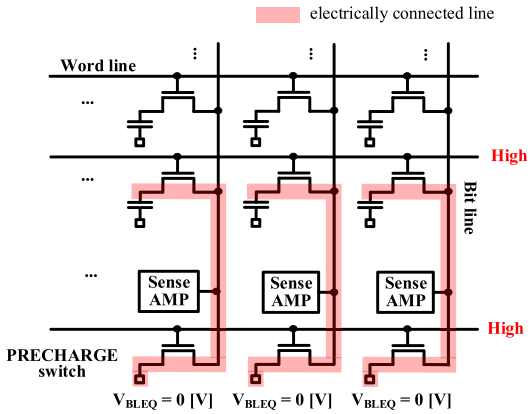


Fig. 5. Array operation of MemSweep.

above a certain threshold level. Therefore, a high PwrOn signal means that stable power is supplied to the DRAM. As shown in Fig. 3, the Command Decoder of conventional DRAM also generates command only when the power is stable.

MemLock disables the Command Decoder by monitoring the LockOut signal in addition to the PwrOn signal. To do this, logic gates to turn on the Command Decoder are included and LockOut signal is connected to the input of the logic gates. The LockOut signal is the information (1:lock, 0: unlock) stored in a 1-bit register, called the Lock Register. The Lock Register is made of an SR (Set-Reset) latch. When the set and reset input comes in, the stored bit is set to 1 (lock) and 0 (unlock), respectively. The set input is linked to the OR gate output of $\overline{PwrOn}$ and RESET signal. The reset input is connected to the RemoveOver signal that is generated when MemSweep completes successfully.

Fig. 4a shows the operation of MemLock in the cold boot case. Before the power is sufficiently charged to the DRAM, the power detector makes the PwrOn signal 0. If PwrOn is 0, the set input of the Lock Register becomes 1, which makes the Lock Register automatically initialize to 1. Therefore, the DRAM moves to a lock state before the power-on is completed. Since the SR latch keeps the set state before the reset signal comes in, the DRAM remains in the lock state after power-on. After the DRAM enters the lock state, Amnesiac DRAM automatically performs MemSweep. After removing the DRAM contents, MemSweep produces a RemoveOver pulse to indicate that it is done. This pulse is transferred to the reset input of the Lock Register and makes the Lock Register reset (unlock).

Fig. 4b shows the operation in the warm boot case. In this case, the memory controller sets the RESET pin to start an initialization process. The RESET signal line is connected to the set input of the Lock Register, which makes the bit stored in the Lock Register to 1 (lock). Then, Amnesiac DRAM automatically performs MemSweep. When MemSweep finishes, it produces the RemoveOver pulse, and the lock is released.

4.2 MemSweep

MemSweep is implemented by modifying the REFRESH circuit. REFRESH restores all DRAM cells through 8 K (8,192) commands. MemSweep similarly removes the contents through 8 K Unit MemSweep (UMS) operation. UMS erases DRAM contents by making the cell voltage 0 V.

Array Operations. Fig. 5 shows the cell array operation of UMS. As mentioned in Section 2, the word line is enabled at REFRESH to couple the cells to the bit line. Then, the precharge switch is turned off and the sense amplifier is turned on to restore the cell data. The cell array operation at UMS is identical to that at REFRESH except for the sense amplifier and the PRECHARGE switch. UMS keeps the PRECHARGE switch on and keeps the sense amplifier off, similar to the idle state (i.e., PRECHARGE state). As a result, the cell is connected to the bit line equalization power (BLEQ) via the bit line and the PRECHARGE switch, so the cell is charged to VBLEQ. The red line represents the electrically connected path during UMS. The UMS array operation can be implemented by modifying the input signals of the DRAM array, and these input signals are generated in the peripheral area of DRAM. Thus, this implementation does not require modification in the DRAM core area (i.e, bank) and thus causes negligible area overhead.

Bit Line Equalization Power Control. BLEQ power is the power to precharge the bit line and its voltage level is 1/2 VEXT in normal conditions. Since MemSweep makes the cell 0 V, BLEQ power is set to 0 V during the MemSweep. To do this, the MemSweep operation changes the VBLEQ.

Fig. 4a shows the change of VBLEQ during a cold boot. In the case of a conventional DRAM, when VEXT is ramped up, the internal power generators ramp up DRAM powers, including a BLEQ power. In the case of Amnesiac DRAM, other powers ramp up like a conventional DRAM, but the ramping up of VBLEQ is postponed until after the MemSweep. Therefore, during the MemSweep, VBLEQ maintains 0 V, the initial voltage level. After MemSweep, VBLEQ ramps up to 1/2 VEXT for normal operation.

Fig. 4b shows the change of VBLEQ during warm boot. When the memory controller reset DRAMs, Amnesiac DRAM discharges BLEQ power to 0 V before the MemSweep. After the MemSweep, the Amnesiac DRAM ramps up the BLEQ power back to normal level (1/2 VEXT).

We calculated the ramp up and discharge time of the BLEQ power. According to the reference [28], the BLEQ power of 1 Gb SDRAM has a capacitance of 120 nF and the BLEQ power generator can supply 30 mA of current. We assume that 8 Gb DDR4 has eight times the capacitance of 1 Gb SDRAM. Therefore, 2.4 us is required to charge the BLEQ power to 0.6 V, which is 1/2 VDD (t = CV / I).

Control Logic. Fig. 6a shows the blocks required to implement MemSweep. The uncolored box is the circuit that formerly existed in DRAMs, and the box colored with black is the new circuit for MemSweep.

The array control signals for UMS are generated in Row Control. As mentioned in Section 2, Row Control is a block that generates array control signals at the proper timing. We connected the UMS signal to the Row Control via an OR gate so that the Row Control outputs the same control signals during UMS and REFRESH. Then, we connected AND gates to the output of the Row Control so that the 'Sense Amplifier On' and 'Precharge SW Off' do not occur during the MemSweep (i.e., when LockOut is high). As a result, the DRAM array operates as shown in Fig. 5 for UMS.

MemSweep control, which is surrounded by a dashed line, controls the overall MemSweep operation. MemSweep control generates 8 K UMS operations at appropriate times

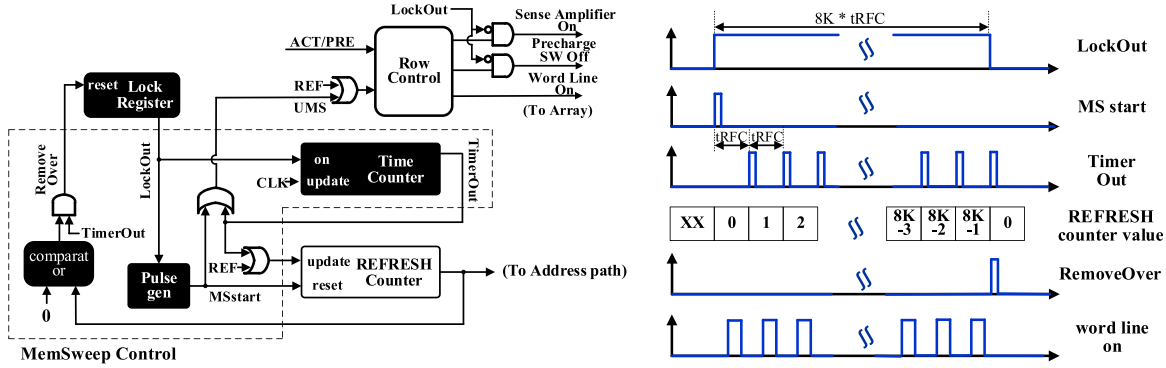


Fig. 6. Control block of MemSweep.

and transfers them to the DRAM control path. In each UMS, the target row address is also transferred to the address path. To this end, MemSweep consists of the REFRESH counter, the Time Counter, and a comparator.

The REFRESH counter is a circuit to provide the address of a row for the REFRESH operation. For MemSweep, the existing REFRESH counter is used and provides the row address for the UMS operation. Time Counter is a clock-based counter for determining proper UMS timing. MemSweep generates 8 K periodic UMS operations. Time Counter counts the time required for a single UMS operation (tRFC: time for a REFRESH). Time Counter operates only when the DRAM is in the lock state. A comparator is a circuit that compares two digital values. It compares the output of the REFRESH counter with address 0 to determine the end of MemSweep. The REFRESH counter resets its contents to 0 at the beginning of MemSweep and increments the value by 1 for each UMS operation. Therefore, the value of REFRESH counter becomes 0 after all DRAM contents are removed through 8 K UMS operations. The comparator generates a RemoveOver signal when the value of the REFRESH counter is 0, which resets Lock Register and releases the lock.

Fig. 6b shows the overall timing diagram of the major control signals for MemSweep. When the DRAM is locked, the LockOut signal will become high. At the same time, the MSstart (MemSweep Start) pulse, which represents the start of MemSweep, is generated from a pulse generator. The MSstart signal sets the value of the REFRESH counter to 0 and triggers UMS array operations. The time counter generates a periodic TimerOut pulse when LockOut is high. Similar to the MSstart signal, this TimeOut pulse 1) increases the REFRESH counter value and 2) triggers UMS array operations. After the time counter generates 8 K UMS array operations, the value of the REFRESH counter exceeds the maximum value and becomes 0. Then, the comparator monitors it and generates the RemoverOver signal, which ends the MemSweep operation.

TABLE 1
Evaluation Environment

Processor	4 core, 3.2 GHz, Out of Order, Reorder Buffer Size: 128 Retire Width: 4, Fetch Width:4, Pipeline depth: 10
Cache	Private 512 KB Last Level Cache per core, Cache line size: 64B
DRAM	DDR4-2133, 8 GB, 1 channel, 1 rank, 16 banks
Workload	Commercial: comm1-5 / Biobench: mummer SPEC: leslie, libq / PARSEC: black, face, ferret, fluid, freq, stream, swapt

4.3 Memory Controller Level Support

In Amnesiac DRAM, the memory controller should wait until the contents are removed in the initialization sequence. To this end, the DRAM specification should include the exact clock-based execution time required for MemSweep. By referring this execution time,[5] the memory controller can know when Memsweep has finished.

5 Evaluation

In this section, we evaluate Amnesiac DRAM in terms of energy consumption, initialization latency, correctness and chip area. We also compare the prior memory encryption approach [9] with Amnesiac DRAM regarding energy consumption. We also compare the conventional WRITE method and the previous DRAM bulk initialization approaches [29], [30] with MemSweep concerning initialization overhead.

5.1 Methodology

To show the energy overhead of Amnesiac DRAM, we used USIMM, which has been widely adopted to evaluate DRAM systems [31], [32], [33], [34], [35]. For the DRAM chip energy parameter used in USIMM, we used the IDD parameter from within the DDR4 DRAM specification [19]. We used the benchmarks from MSC [36]. For the off-chip data bus energy parameter for USIMM, we used the Micron DDR4 power calculator [37].[6] We modeled a 4-core system operating at 3.2 GHz connected to an 8 GB DDR4-2133 module. The specific system configuration is mentioned in Table 1. Also, to measure the energy and chip area overhead consumed by the additional circuits for Amnesiac DRAM, we performed SPICE simulation and layout, with a 65 nm process.[7] We composed the DRAM cell array based on the DRAM core model [38].

5. In general, the execution times of DRAM commands are described in the specification, and the memory controller can know when the specific command ends by referring to the execution time.
6. Termination: RTTu1/c=120, RTTu2=infinite, Rz=48, Rs=10.
7. Since the DRAM model parameters are not publicly available, we have carried out simulations with the logic model parameter.

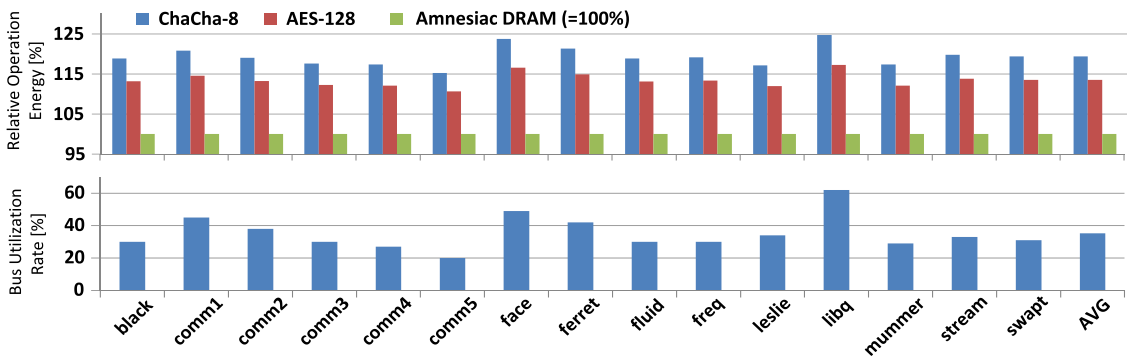


Fig. 7. Operation energy overhead comparison.

5.2 Operation Energy

Full memory encryption, like Amnesiac DRAM, can protect all the contents (i.e., except the hardware encryption key) stored in the DRAM. However, Amnesiac DRAM does not require any energy overhead in normal operation, while full memory encryption consumes energy to generate a secret key during data READ / WRITE. The prior approach [9], which implemented full memory encryption on a memory system, analyzed power consumption for two proposed cipher modules (AES-128, ChaCha-8). According to this study, AES-128 consumes 10 mW and ChaCha-8 consumes 15 mW when transmitting data per DRAM channel.[8] Utilizing these values and the USIMM, we evaluated the proportion of full memory encryption that increases DRAM system energy.

Fig. 7 shows the DRAM system energy consumed by previous approaches (AES-128, ChaCha-8) and Amnesiac DRAM for various workloads. Specifically, Fig. 7a shows the normalized value, and the baseline is the DRAM system energy except for the defense mechanism. Since Amnesiac DRAM does not consume operation energy, it shows 0 percent energy overhead in all workloads. In contrast, the energy overhead of AES-128 is 10 $\sim$ 17.6 percent and ChaCha-8 is 14.3 $\sim$ 25.2 percent. This energy overhead shows similar tendency with the data bus utilization rate of the workload. As shown in Fig. 7b, comm5, which has the lowest data bus utilization rate (20 percent), shows the lowest energy overhead (AES-128: 10 percebt, ChaCha: 14.3 percent and libq, which has the highest data bus utilization rate (61 percent), shows the highest energy overhead (AES-128: 17.6 percent, ChaCha-8: 25.2 percent). This is because, as more data is transferred, more secret keys must be generated, and therefore the energy consumption of the cipher block increases. On average, the energy overhead of AES-128 is 13.3 percent and ChaCha-8 is 19.1 percent.

Fig. 8a shows the analysis of the energy consumed by the defense mechanism (average values of all workloads) depending on the number of cores. As Fig. 8a shows, as the number of cores increases, the energy overhead of the cipher engine increases. The energy overhead of the cipher engine is AES-128: 8.3, 10.3, and 13.3 percent, and ChaCha-8: 11.8, 14.8, and 19.1 percent when the number of core is 1, 2, and 4. This is because as the number of cores increases, more workloads are simultaneously operated and the data bus utilization rate becomes higher.[9] As shown in Fig. 8b, the data bus utilization rate shows an average of 12, 19, and 35 percent when the number of core is 1, 2, and 4.

5.3 Initialization Overhead

Latency. We calculated the time required for MemSweep. MemSweep deletes all the DRAM contents through 8 K UMS. Therefore, the time required for MemSweep is $8192\times$ 'time required for a single UMS.'

To determine the time required for UMS, two factors must be considered. The first factor is the number of word lines that can be turned on per unit time. DRAM receives 1 $\sim$ 1.8 V power from the outside, but the voltage around 3 V is required to activate the word line[10] [20], [28]. To this end, there are voltage generators inside DRAMs to generate a voltage higher than the externally supplied voltage. Since the voltage generator takes up a large area, DRAM vendors place as many voltage generators as needed on the chip, taking into account the maximum number of word lines enabled per unit time. In modern DRAMs, the operation when the maximum number of word lines are turned on per unit time is the REFRESH [39]. Therefore, considering only the first factor, the minimum period of UMS operation is tRFC.

The second factor is the time required for a cell to be precharged to 0 V. We confirmed this through SPICE simulation. Fig. 9 shows the voltage of the DRAM cell node during UMS. We can see that the cell data that had 1.2 V is changed to 0 V as a result of the UMS operation. Since we gave process variation to the cell transistors [40], the rate at which the cell is charged to 0 V differs from cell to cell. As shown in Fig. 9, charging the cell data ends in 7 ns even for the cell with the worst process variation.[11]

As density increases, DRAM has a larger tRFC because it needs to refresh more cells (2 Gb: 160 ns, 4 Gb: 260 ns, 8 Gb: 350 ns). The smallest tRFC of the commodity DDR4 is 160 ns (2 Gb DRAM). Since the time it takes for a cell to precharge to 0 V is 7 ns, it can be seen that tRFC determines the operating time of UMS. Therefore, 8 K x tRFC is required for the overall MemSweep operation. Fig. 10 shows the time required for the MemSweep operation with varying DRAM

8. We extracted this value from Figure 7 in the paper [9].

9. As mentioned before, more secret keys must be generated, and the energy is consumed in the cypher engine.

10. The physical size of the cell switch transistor is too small, so it requires a high gate voltage to reduce the on-resistance of the switching transistor.

11. The time at which the cell voltage entered the 1% error range.

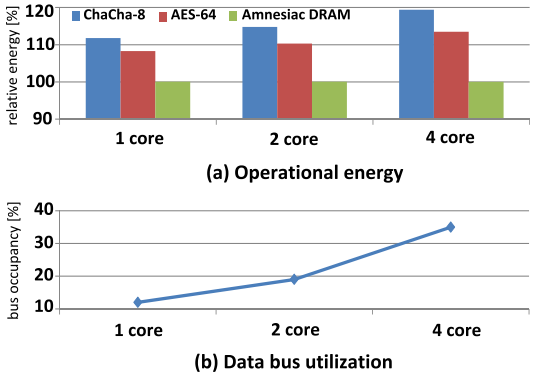


Fig. 8. Operation energy versus the number of core.

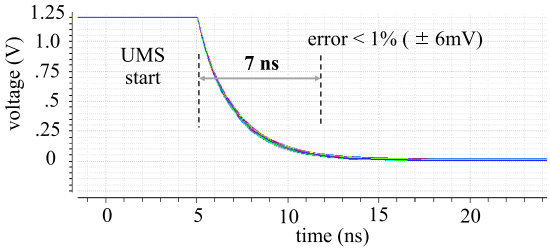


Fig. 9. Cell node voltage during UMS(SPICE).

density. As shown in Fig. 10a, the MemSweep operation can be completed within 2.8 ms even in the case of the 8 Gb DRAM, the largest commercial DRAM currently available. In the case of a smaller capacity DRAM, the MemSweep operation requires less time.

Removing the DRAM contents also can be performed by using the WRITE commands. Fig. 10a also shows the required content removal time in the case of using the WRITE commands. x4 DRAM, which is widely used, write 32-bit to a single write command in one chip. A single WRITE requires a time of tCCD.[12] Therefore, assuming that the contents of the 8 Gb DRAM are removed at a speed of 2133 Mbps (DDR4-2133), x4 DRAM needs 1.007 s[13] for contents removal. This is 351 times slower than MemSweep.

Fig. 10a also shows the content removal latency of the prior in-DRAM bulk initialization methods [29], [30]. A prior approach, called RowClone, uses back-to-back ACTIVATE commands to copy data from a row to an another row, within a subarray.[14] For data initialization, RowClone writes a single preserved row per sub-array with zero and copies the zero data to other rows. Another prior approach, called RowReset, writes zero to a single row by using a modified ACTIVATE command.[15] As shown in Fig. 10a, RowClone and RowReset takes 28.1 ms and 26.2 ms to initialize the 8 Gb DRAM data, respectively.[16] This is 9.7 times

12. Minimum time between two column access (reads or writes).
13. The latency can be calculated by using this formula: $\frac{8G}{32}$ x tCCD
14. One DRAM bank is divided into several sub-arrays, and rows belonging to one sub-array share the same bit lines.
15. The power control signal of sense amplifiers is modified.
16. The 8 Gb DRAM has 2 M rows in all banks. ACTIVATE on different banks must wait for tRRD. Since RowClone requires two ACTIVATEs to initialize one row, 'tRRD x 2M x 2' is required for data copy. Also, '8G / 32 x tCCD / 512' is necessary to write zero data to one row per sub-array (assuming 512 word per subarray [38]). Also, RowReset requires two ACTIVATEs (i.e., one modified-ACTIVATE and one normal ACTIVATE) to initialize one row.

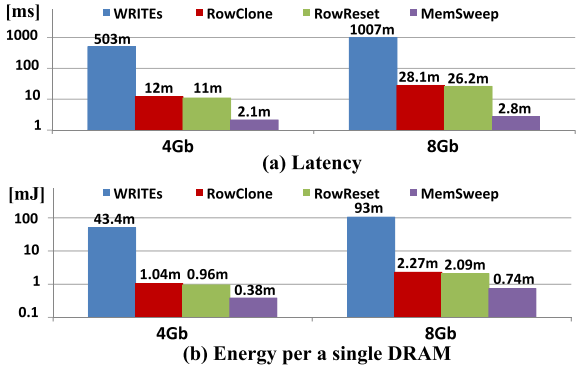


Fig. 10. Comparison between MemSweep and prior initialization approaches.

and 9 times slower than MemSweep, respectively. This is because the REFRESH (utilized by MemSweep) can access more cells per unit time than the back-to-back ACTIVATEs (RowClone) and modified ACTIVATEs (RowReset) [39].

Energy. We predicted the energy consumed by MemSweep using IDD parameters described in the 4 Gb and 8 Gb DDR4-2133 specifications [19], [41].[17] MemSweep is the same operation as REFRESH except that sense amplifiers do not work. Therefore, the energy required for MemSweep is equal to 'the REFRESH energy - the energy required to operate sense amplifiers.' Since it is not easy to extract the energy required for the sense amplifiers, we conservatively estimate that the energy required for MemSweep is equal to the energy required for REFRESH.

Fig. 10b shows the required energy for each density of DRAM. The results show that 0.7 mJ of energy is needed to erase all the contents stored in the 8 Gb DRAM. This is the same amount of energy as when the DRAM is in the standby state for 29.4 ms. Considering the fact that MemSweep only occurs once at system booting, the energy consumption of MemSweep is negligible.

Fig. 10b also shows the energy needed to erase contents using the WRITEs, RowClone, and RowReset. The results show that the data-removing energy for 8 Gb DRAM (x4) using WRITEs is 93.1 mJ, which is 126 times greater than MemSweep. Also, the data-removing energy using Rowclone and Row Reset is 2.3 mJ and 2.1 mJ, which is 3.7 times and 2.8 times greater than MemSweep, respectively.

5.4 Correctness

We verified the logical operation of Amnesiac DRAM during a cold-boot attack through SPICE simulation (Fig. 11). To set the cold-boot situation in this simulation, we stopped supplying power to the DRAM model for a while (i.e., floating the DRAM power node) and then supplied power again. We set the temperature to -50 degree Celsius to match the environment of the cold-boot attack. As shown in Fig. 11 (See *Power* line), when the power supply to the DRAM is cut off, the potential of the power node inside the DRAM rapidly falls to ground level. That is, because the various circuits in the DRAM consume energy, the power level will be lowered unless the system continuously supplies the power.

17. The energy consumed by the additional hardware for MemSweep was negligible (8-Gb DRAM: 7 nJ in SPICE simulation).

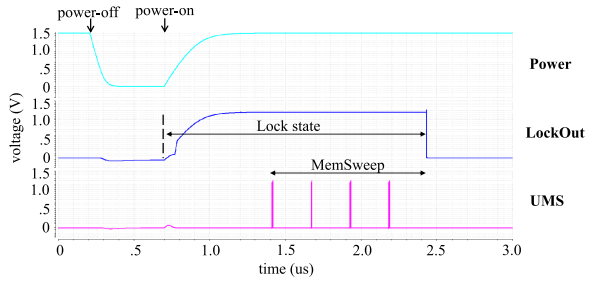


Fig. 11. Control signal of Amnesiac DRAM (SPICE).

Fig. 11 (See *LockOut* line) shows the information stored in the Lock Register. Before turning off the power, the DRAM is unlocked, and thus '0' is stored in the Lock Register. When the power is disconnected from the DRAM, the potential of the power node becomes ground, so the value of the Lock Register keeps its value, '0' (ground level). As mentioned in Section 4, the Lock Register is designed to be initialized to a value of '1,' so it changes its value to '1' at power-on.[18]

Fig. 11 (See *UMS* line) shows the operation of the primary control signals during MemSweep. We changed the REFRESH counter to 2-bit in this simulation to show the overall MemSweep operation in a limited simulation time. As shown in Fig. 11, the UMS signal starts to be generated when the power enters a stable state (PwrOn = 1). Until the REFRESH counter becomes zero, UMS operations are periodically created. Since we set the REFRESH counter to 2-bit, MemSweep ends after four UMS operations and the value of Lock Register becomes '0' (Unlock). We also confirmed that UMS makes cell contents zero (Fig. 9).

5.5 Chip Area Overhead

We measure the area overhead by laying out the additional circuits needed for MemLock and MemSweep by using the 65 nm logic process. As mentioned in Section 4, the additional circuits required for MemLock are a 1-bit SR latch and a small number of logic gates. The area required to implement MemLock is 5 um^2. Additional circuits required for MemSweep are a 10-bit counter, an 18-bit comparator, and some logic gates. To make this, an area of 505 um^2 is required. Amnesiac DRAM requires a total area of 510 um^2.

The 20 nm 8 Gb DRAM reported in a prior study has an area of 26.4 mm^2 [42]. Therefore, the area overhead required to implement Amnesiac DRAM is 0.002% of the conventional DRAM, which is almost negligible.

6 Discussion

Steps Toward Deploying the Amnesiac DRAM. The key criteria for adoption of a new DRAM feature is its manufacturing cost. Since Amnesiac DRAM costs near none (i.e., 0.002 percent chip area) and its mechanism are comprehensive to understand its benefits, we believe it is not unrealistic to make a formal proposal to the standardization consortium (JEDEC). As the security of the DRAM is one of emerging interests and demands of the DRAM community, Amnesiac DRAM can be discussed and included in the next generation DRAM specifications (DDR6, LPDDR6/7, GDDR7). If that happens, we believe end users will benefit from our security enhancement with negligible performance overheads.

Defeating Advanced Attacks. Amnesiac DRAM's focus is to defend against conventional cold boot attacks (i.e., cases described in Section 2.2). However, more determined attackers may attempt to deceive Amnesiac DRAM, so in this section, we discuss the feasibility of new attack vectors and potential directions to mitigate them.

One direct way for attackers to bypass the current Amnesiac DRAM is to deceive the sensing mechanisms of Amnesiac DRAM because it relies on two signals, namely, power off/on and RESET. To maintain the power supply when transferring the DRAM module, competent adversaries, in theory, can solder an electric wire to the power node of the DRAM module and connect that wire to the battery or power supply before disconnecting the DRAM module. Similarly, the RESET signal can be deceived by grounding the signal before physical disconnection of the DRAM module; disconnection may set the RESET level high, forcing Amnesiac DRAM into the lock state, which makes further attack impossible.

One feasible way to defeat the proposed advanced attacks, if one wishes to thwart them under the threat model, is to modify the I/O training logic of the DRAM. However, such modification would be minimal since modern DRAMs (e.g., DDR4, LPDDR4/5, and GDDR5/6) already require the I/O training logic in the initialization sequence for high data bandwidth. Specifically, the I/O training is a process of tuning the I/O interface circuits of the memory controller to reflect the process variation of DRAM chip/package/module-PCB, motherboard, and memory controller chip/package.[19] Since this I/O training requires close coordination of the DRAM (i.e., Write Leveling in DDR4), the memory controller cannot perform training unless the DRAM permits the training.

Our proposal is to modify the DRAM to conduct I/O training modes *only after* MemSweep. When the advanced attackers move the DRAM module to the attacker's system and initiate the training of the memory controller, the MemSweep phase is already performed and so the memory content should already be wiped by our protection.

Comparing with SGX. Existing defenses based on memory encryption, such as Intel SGX, can defeat cold boot attacks in certain circumstances [9], but have a few fundamental limitations, as they are designed to be general purpose. For example, SGX provides such a defense only on the enclave regions (i.e., encrypted memory), not on full system memory, and imposes non-negligible performance overheads on larger memory. However, we should note that it is not an apple-to-apple comparison as SGX provides beyond the simple memory encryption, namely, attestation and isolation, to cover much broader attack surfaces in a general-purpose manner. Unlike SGX, the goal of Amnesiac DRAM is to focus only on one type of attacks, the cold boot attack, but in a proactive and practical manner, aiming to be deployed in the real world.

18. The voltage of LockOut gradually rises as the power is charged.

19. Recent DRAMs (i.e., DDR4) only support high bandwidth operation (i.e., DDR4: from 1.6 Gbps) and future DRAMs operate at the higher data rate. At high data rate, the system must require I/O training for the errorless data transfer.

7 Related Works

We review various cases of cold boot attacks and their countermeasures that have been studied so far.

7.1 Cold Boot Attack

Gutmann initially claimed that the remanence nature of memory could pose a security threat even before a cold boot attack was proposed [1]. After that, Halderman et al. experimentally demonstrated that DRAM had remanence in 2009 and popularized memory remanence issues by offering a cold boot attack [2]. Since then, many studies have proven the effectiveness of cold boot attacks in various environments. Carbone et al. [3], Lindenlauf et al. [4], and Gruhn et al. [5] succeeded in a cold boot attack on X86 based computer systems. Müller et al. [6] and Huber et al. [7] succeeded in a cold boot attack in ARM based hand-held smartphone environments. Bauer et al. [8] and Yitbarek et al. [9] successfully performed a cold boot attack in modern Intel DDR3/4 memory controllers that have a data scrambler.

7.2 Countermeasures

Several prior studies have proposed countermeasures to protect sensitive information from cold boot attacks.

CPU Bound Cryptography. Until now, the primary targets of cold boot attacks have been to find cryptographic keys stored in memory. Therefore, many cryptographic solutions have studied how to protect their keys [10], [11], [12], [13], [14], [15], [17]. Their basic idea is not to use DRAM in the cryptography operation, but to use only the storage space (register or cache) inside the processor. With this method, there is no sensitive information inside the DRAM, so it is useless for the attacker to acquire the DRAM contents.

To store keys, TRESOR [11] used debug registers, which originally stored the debugging information. LoopAmnesia [12] stored the key using model-specific registers that serve as performance counters. AESSE [10] utilized the streaming SIMD extensions (SSE) registers. Like these, the approaches storing keys in the registers can prevent cold boot attacks, but they have some drawbacks. These methods cause a performance impact due to the lack of storage space. Since all the round keys cannot be stored in a few registers at once, these methods should create a round key before encryption, use it, and delete it. Besides, the registers used to store the key cannot be used for the original purpose.

Frozen cache [13] sets the CPU's L1 cache to no-fill mode and stores the round key.[20] Copker [14] also stores the key in the cache hierarchy and sets the shared cache to no-fill cache mode to prevent the data in the cache from eviction into DRAM. CaSE [16] does not evict the key to DRAMs by using *cache lockdown*[21] in the ARM architecture. Even though these prior approaches can prevent a cold boot attack, they prevent other tasks performed concurrently with cryptographic computation from using the cache properly. As a result, they cause performance degradation of the system. Mimosa [15] uses Intel's TSX (Transactional memory) to store the key only inside the cache. However, like other CPU bound cryptography, Mimosa cannot protect valuable information other than the cryptography key and has a performance issue for the cryptographic application.

Full Memory Encryption. This protects the contents by encrypting when writing the contents in DRAMs and decrypting when reading the contents [9], [18], [43], [44], [45], [46], [47]. Even if the attacker obtains the DRAM contents, he cannot understand the contents without correct decryption.

Modern Intel memory controllers use data scrambling [46]. The main objective of it is to prevent irregular power noise due to data patterns rather than security protection. Even though its main objective is not content protection, people have thought that scrambling can be a defense against cold boot attacks. However, recent studies have shown examples of successful cold boot attacks on the Intel DDR3/DDR4 memory controllers, which embed scramblers [8], [9].

Since then, Yitbarek et al. claim to have created a scrambling key stream using a cryptography engine rather than a simple mechanism [9]. This idea, however, requires additional energy consumption because the memory controller must generate a cryptographic key stream each time it accesses the DRAMs, as shown in Section 5. In addition, a massive dedicated hardware for the key stream generation in the memory controller is needed to implement this system.

8 Conclusion

A dangerous attack vector, called cold boot attack, exploits the DRAM's remanence effect that retain contents alive after power-off. To defend against cold boot attacks, we proposed a DRAM side solution, Amnesiac DRAM. The basic concept of Amnesiac DRAM is to detect a situation in which an attack occurs (i.e., system booting): 1) power off/on or 2) reset. Then, the built-in initialization mechanism deletes all the contents of DRAM. Amnesiac DRAM is an economical solution compared to existing countermeasures because it requires only a small amount of time (under 2.8 ms) during system booting and does not cause any other performance degradation or energy consumption to the running system. The implementation cost of Amnesiac DRAM is negligible because it can be implemented by slightly modifying the control path of DRAMs. In conclusion, Amnesiac DRAM is an economical way to popularize countermeasures for cold boot attacks.

Acknowledgments

This paper is under review process of IEEE Transactions on Computers.

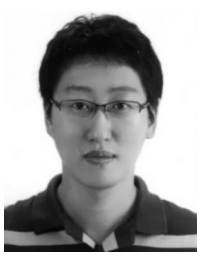
References

[1] P. Gutmann, "Data remanence in semiconductor devices," in *Proc. 20th USENIX Conf. Security*, 2001.
[2] J. A. Halderman *et al.*, "Lest we remember: Cold-boot attacks on encryption keys," *Commun. ACM*, vol. 52, May 2009.
[3] R. Carbone, C. Bean, and M. Salois, "An in-depth analysis of the cold boot attack," DRDC Valcartier, Defence Research and Develop., Canada, Tech. Rep., 2011.
[4] S. Lindenlauf, H. Höfken, and M. Schuba, "Cold boot attacks on DDR2 and DDR3 sdram," in *Proc. 10th Int. Conf. Availability Rel. Security*, 2015, pp. 287–292.

20. In the no-fill mode, read misses do not cause cache replacement and write misses access RAM directly.

21. Cache lockdown prevents CPU cores from evicting data allocated by a different core from the shared L2 cache.

[5] M. Gruhn and T. Müller, "On the practicability of cold boot attacks," in *Proc. Int. Conf. Availability, Rel. Security*, 2013, pp. 390–397.
[6] T. Müller and M. Spreitzenbarth, "FROST: Forensic recovery of scrambled telephones," in *Proc. 11th Int. Conf. Appl. Cryptography Netw. Security*, 2013, pp. 373–388.
[7] M. Huber, B. Taubmann, S. Wessel, H. P. Reiser, and G. Sigl, "A flexible framework for mobile device forensics based on cold boot attacks," *EURASIP J. Inf. Security*, vol. 2016, no. 1, 2016, Art. no. 17.
[8] J. Bauer, M. Gruhn, and F. C. Freiling, "Lest we forget: Cold-boot attacks on scrambled DDR3 memory," *Digit. Investigation*, vol. 16, pp. S65–S74, 2016.
[9] S. F. Yitbarek, M. T. Aga, R. Das, and T. Austin, "Cold boot attacks are still hot: Security analysis of memory scramblers in modern processors," in *Proc. 23rd IEEE Symp. High Perform. Comput. Architecture*, 2017, pp. 313–324.
[10] T. Müller, A. Dewald, and F. C. Freiling, "AESSE: A cold-boot resistant implementation of AES," in *Proc. 3rd Eur. Workshop Syst. Security*, 2010, pp. 42–47.
[11] T. Müller, F. C. Freiling, and A. Dewald, "TRESOR runs encryption securely outside ram," in *Proc. 20th USENIX Conf. Security*, 2011, pp. 17–17.
[12] P. Simmons, "Security through amnesia: A software-based solution to the cold boot attack on disk encryption," in *Proc. 27th Annu. Comput. Security Appl. Conf.*, 2011, pp. 73–82.
[13] J. Pabel, "Frozencache mitigating cold-boot attacks for full-disk-encryption software," in *Proc. 27th Chaos Commun. Congr.*, 2010.
[14] L. Guan, J. Lin, B. Luo, and J. Jing, "Copker: Computing with private keys without ram," in *Proc. IEEE Trans. Dependable Secure Comput.*, 2014, pp. 23–26.
[15] L. Guan, J. Lin, B. Luo, J. Jing, and J. Wang, "Protecting private keys against memory disclosure attacks using hardware transactional memory," in *Proc. IEEE Symp. Security Privacy*, 2015, pp. 3–19.
[16] N. Zhang, K. Sun, W. Lou, and Y. T. Hou, "CaSE: Cache-assisted secure execution on ARM processors," in *Proc. IEEE Symp. Security Privacy*, 2016, pp. 72–90.
[17] B. Garmany and T. Müller, "Prime: Private rsa infrastructure for memory-less encryption," in *Proc. 29th Annu. Comput. Security Appl. Conf.*, 2013, pp. 149–158.
[18] M. Henson and S. Taylor, "Memory encryption: A survey of existing techniques," *ACM Comput. Surv.*, vol. 46, Mar. 2014, Art. no. 53.
[19] "8gb B-die DDR4 SDRAM samsung," 2015.
[20] B. Jacob, S. W. Ng, and D. T. Wang, *Memory Systems (Cache, DRAM, Disk)*, 1st ed. San Mateo, CA, USA: Morgan Kaufmann, 2008.
[21] "JEDEC standard : DDR4 SDRAM specification," 2012.
[22] I. Bhati, M.-T. Chang, Z. Chishti, S.-L. Lu, and B. Jacob, "Dram refresh mechanisms, penalties, and trade-offs," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 108–121, Jan. 2016.
[23] T. Miller and F. C. Freiling, "A systematic assessment of the security of full disk encryption," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 5, pp. 491–503, Sep. 2015.
[24] "Micron technical note, TN-41–07: DDR3 power-up, initialization, and reset," 2008. [Online]. Available: https://www.micron.com/resource-details/30a90caf-7074-45d2-9d8a-9051233eafe5
[25] "Android vs iphone boot times tested: which one is the fastest?," May 2015. [Online]. Available: https://www.phonearena.com/news/Android-vs-iPhone-boot-times-tested-which-one-is-the-fastest_id69582
[26] "iPhone boot-up time comparison: 6s vs. 6 vs. 5s vs. 5 vs. 4 vs. 3gs," Sep. 2015. [Online]. Available: https://www.ifixyouri.com/blog/iphone-6s-bootup-time-comparison-vs.-6-vs.-5s-vs.-5-vs.-4-vs.-3gs/
[27] G. Lim and M. Ham, "BB: Booting Booster for consumer electronics with modern OS," in *Proc. 11th Eur. Conf. Comput. Syst.*, 2016, Art. no. 17.
[28] O. Weinfurtner, D. Storaska, and L. Hsu, "Advanced controlling scheme for a DRAM voltage generator system," *IEEE J. Solid-State Circuits*, vol. 35, no. 4, pp. 552–563, Apr. 2000.
[29] V. Seshadri *et al.*, "Rowclone: Fast and energy-efficient in-dram bulk data copy and initialization," in *Proc. 46th Annu. IEEE/ACM Int. Symp. Microarchitecture*, 2013, pp. 185–197.
[30] H. Seol, W. Shin, J. Jang, J. Choi, J. Suh, and L.-S. Kim, "In-dram data initialization," *Trans. Very Large Scale Integration Syst.*, vol. 25, vol. 11, pp. 3251–3254, Nov. 2017.
[31] P. Nair, C. C. Chou, and M. K. Qureshi, "A case for refresh pausing in DRAM memory systems," in *Proc. IEEE 19th Int. Symp. High Perform. Comput. Architecture*, 2013, pp. 627–638.
[32] W. Shin, J. Yang, J. Choi, and L. S. Kim, "NUAT: A non-uniform access time memory controller," in *Proc. IEEE 20th Int. Symp. High Perform. Comput. Architecture*, 2014, pp. 464–475.
[33] S. H. Pugsley *et al.*, "Comparing implementations of near-data computing with in-memory MapReduce workloads," *IEEE Micro*, vol. 34, no. 4, pp. 44–52, Jul. 2014.
[34] J. Choi *et al.*, "Multiple clone row DRAM: A low latency and area optimized DRAM," in *Proc. ACM/IEEE 42nd Annu. Int. Symp. Comput. Architecture*, 2015, pp. 223–234.
[35] X. Zhang, Y. Zhang, B. R. Childers, and J. Yang, "Restore truncation for performance improvement in future DRAM systems," in *Proc. IEEE Int. Symp. High Perform. Comput. Architecture*, 2016, pp. 543–554.
[36] "Memory Scheduling Championship (MSC)," 2012. [Online]. Available: http://www.cs.utah.edu/ rajeev/jwac12/
[37] "Micron DDR4 SDRAM system-power calculator," 2016. [Online]. Available: https://www.micron.com/support/tools-and-utilities/power-calc
[38] Rambus, "Dram power model," 2010.
[39] I. Bhati Z. Chishti, S.-L. Lu, and B. Jacob,"Flexible auto-refresh: Enabling scalable and energy-efficient dram refresh reductions," *ACM SIGARCH Comput. Architecture News*, vol. 43, no. 3, pp. 235–246, 2015.
[40] D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[41] "4gb D-die DDR4 SDRAM samsung," 2014.
[42] C. K. Lee *et al.*, "23.2 a 5 gb/s/pin 8 Gb LPDDR4X SDRAM with power-isolated LVSTL and split-die architecture with 2-die ZQ calibration scheme," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2017, pp. 390–391.
[43] D. L. C. Thekkath, M. Mitchell, P. Lincoln, D. Boneh, J. Mitchell, and M. Horowitz, "Architectural support for copy and tamper resistant software," in *Proc. 9th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, 2000, pp. 168–177.
[44] C. Yan, D. Englender, M. Prvulovic, B. Rogers, and Y. Solihin, "Improving cost, performance, and security of memory encryption and authentication," in *Proc. 33rd Annu. Int. Symp. Comput. Architecture*, 2006, pp. 179–190.
[45] S. Chhabra and Y. Solihin, "i-NVMM: A secure non-volatile main memory system with incremental encryption," in *Proc. 38th Annu. Int. Symp. Comput. Architecture*, 2011, pp. 177–188.
[46] "Intel atom processor s1200 product family for microserver," 2012.
[47] J. Yang, L. Gao, and Y. Zhang, "Improving memory encryption performance in secure processors," *IEEE Trans. Comput.*, vol. 54, no. 5, pp. 630–640, May 2005.



Hoseok Seol received the BS degree in electrical and electronics engineering from Kyoungpook National University, Daegu, Korea, in 2006, and the MS and PhD degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2008 and 2018, respectively. From 2008, he has developed sense amplifier, refresh circuits and I/O interface circuits for commercial DRAM in Samsung Electronics, Hwaseong, Korea. His research interests include memory-domain architecture. He is a student member of the IEEE.

Minhye Kim received the BS and MS degrees in electrical engineering from the Korea Advanced Institute of Science and Technology(KAIST), Daejeon, Korea, in 2014 and 2016, respectively. She is now a second-year JD candidate in the three year JD Program with Seoul National University(SNU) Law School. Her research interest include interaction between law and technology. He is student member of the IEEE.

Taesoo Kim received the SM and PhD degrees from MIT EECS, in 2011 and 2014, respectively. He is a Catherine M. and James E. Allchin Early Career, an assistant professor with the School Computer Science, Georgia Tech. He also serves as the director of the Georgia Tech Systems Software and Security Center (GTS3). His research interests lie in building a system that has underline principles for why it should be secure. Those principles include the design of the system, analysis of its implementation, ellimination of certain classes of vulnerabilities, and clear separation of its trusted components. His thesis work, focused on detecting and recovering from attacks on computer systems known as undo computing. He is a member of the IEEE.

Yongdae Kim received the BS and MS degrees in mathematics from Yonsei University, in 1991 and 1993, and the PhD degree from the Computer Science Department, University of Southern California under the guidance of Gene Tsudik. He is a professor with the Department of Electrical Engineering, KAIST. From 2002 to 2012, he was an associate/assistant professor with the Department of Computer Science and Engineering, University of Minnesota - Twin Cities. Before joining University of Minnesota, he worked as a research staff for two years in Sconce Group in UC Irvine. Before coming to the US, he worked six years with the ETRI for securing Korean cyberinfrastructure. He received the NSF Career Award on storage security and McKnight Land-Grant Professorship Award from the University of Minnesota, in 2005. Currently, he is serving as a steering committee member of Network and Distributed System Security Symposium (NDSS). His current research interests include security issues for various systems such as social networks, cellular networks, P2P systems, medical devices, storage systems, mobile/ad hoc/sensor/cellular networks, and anonymous communication systems. He is a member of the IEEE.

Lee-Sup Kim (M'89–SM'05–F'19) received the BS degree in electronics engineering from Seoul National University, Seoul, Korea, in 1982, and the MS and PhD degrees in electrical engineering from Stanford University, Stanford, California, in 1986 and 1990, respectively. He was a postdoctoral fellow with Toshiba Corporation, Kawasaki, Japan. Since March 1993, he has been with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea where he is a professor. His research interests include vision processors, memory controllers, and memory and display I/O circuits. He was co-recipient of the best paper runner up award at the 2014 IEEE International Symposium on High Performance Computer Architecture (HPCA) and the best paper award at the 2014 IEEE International Conference on Computer Design (ICCD). He has served on the technical committee of the IEEE International Solid-State Circuits Conference (ISSCC) (2004∼2009). He is a fellow of the IEEE.