Threat Models and Security of Phase-Change Memory

Gang Wang
Dept. of Computer Science and Engineering, University of Connecticut
g.wang.china86@gmail.com

***Abstract*—Emerging non-volatile memories (NVMs) have been considered promising alternatives to DRAM for future main memory design. Among the NVMs, Phase-Change Memory (PCM) can serve as a good substitute due to its low standby power, high density, and good scalability. However, PCM material also induces security design challenges mainly due to its interior non-volatility. Designing the memory system necessitates considering the challenges which may open the backdoor for attackers. A threat model can help to identify security vulnerabilities in design processes. It is all about finding the security problems, and therefore it should be done early in the design and adoption of manufacture. To our knowledge, this paper is the first attempt to thoroughly discuss the potential threat models for the PCM memory, which can provide a good reference for designing the new generation of PCM. Meanwhile, this paper gives security advice and potential security solutions to design a secure PCM to protect against these potential threats.**

I. Introduction

Given the grim prospect of technology scaling in DRAM, architects recently have had a growing interest in exploiting alternative memory technologies and integrating them into the main memory hierarchy of a computing system. Emerging non-volatile memories (NVMs), such as Phase-Change Memory (PCM), Resistive Random Access Memory (ReRAM), and Spin-Transfer Torque Random Access Memory (STT-RAM), have been considered as promising alternatives to DRAM for future main memory design [1] [2]. Among these NVMs, PCM is considered a competitive alternative for the design of main memory systems in the next few years. PCM has the advantages of low standby power, high density, good scalability, and non-volatility. Also, it can support multilevel cells (MLC), with which several bits can be installed in one PCM cell to further improve the density. Its non-volatility, however, induces security design challenges that data retained in memory after power-off need to be protected from malicious attacks.

Non-volatility makes the data on PCM more vulnerable to be attacked by malicious attacks, as data is kept in the PCM cells even when powered off. The security problem is more severe as mobile computing systems, such as tablets and smartphones, become more and more popular.

Most prior research focuses on improving write performance and lifetime of PCM as the main memory. Only a few of them address the security challenges caused by its non-volatility [1]. Non-volatility makes these data more vulnerable to be attacked by malicious attacks. One highly useful technique for analyzing security issues and designing defenses is the threat model. Threat models can help to identify vulnerabilities in various environments. The purpose of threat modeling is to organize system threats and vulnerabilities into general classes to be addressed with known memory protection techniques. It is all about finding security problems, and therefore it should be done early in the design and adoption of the services. For instance, a powerful attacker can physically remove the main memory and extract sensitive information from it through memory scanning [3]. A threat model helps in analyzing security problems, designing mitigation strategies, and evaluating potential solutions. Threat models are mainly used at the software level, not too much at the hardware level.

Data security is rightfully calling into question how physical memory, such as PCM, is protected. Literally, it suggests that PCM can be secured following the security models set forth in other domains of computing, such as network security and secure storage systems. However, these domains rely on the use of strong authentication mechanisms, ensuring the right authorization systems are in place, replication for availability, integrity detection mechanisms, and the use of encryption for confidentiality. Unfortunately, none of these methods alone, especially encryption, is a comprehensive solution for protecting PCM memories from threats. For instance, the use of encryption may provide storage confidentiality but may also hamper performance, usability and introduce denial-of-service vulnerabilities. This paper is the first thoroughly to discuss the threat models for PCM which can provide a good reference for designing the new generations of PCM.

The main objective of this paper is to provide an outlook of potential thread models in PCM cells. Meanwhile, we provide several state-of-the-art practical security solutions to prevent or reduce one or more these modeled threats at some extent. We consider both Single-Level Cell (SLC) PCMs and Multi-Level Cell (MLC) PCMs. Compared to SLC PCMs, one more major challenge for adopting MLC PCMs is the assistance drift issue.

The rest of this paper is organized as follows. Section II prepares the preliminary background of PCM cells and threat models. Section III provides the threat models for PCM from different perspectives. Section IV gives the several security design principles for PCM, which can partially help to prevent some threats. Section V concludes this paper.

II. Background

In this section, we first give basic information about PCM cell structure and its characteristics. Then we provide several vulnerability models that PCM main memory may face.

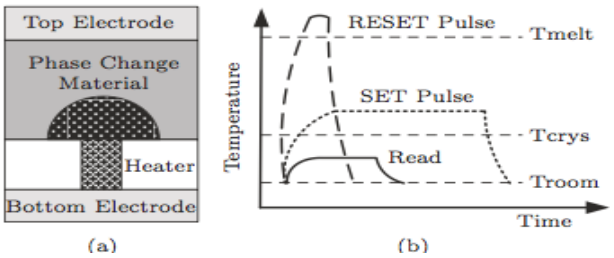

Fig. 1. (a) PCM cell (b) RESET and SET operations [7].

A. PCM

Phase change memory (PCM) is one type of non-volatile memory that exploits the phase change property of chalcogenide alloy to store bit information [4], [5], [6]. A PCM cell usually consists of a thin layer of chalcogenide material, such as $Ge_2Sb_2Te_5$ (GST), two electrodes adhered to the chalcogenide from each side, and a heating resistor extended from one of the electrodes to contact the chalcogenide layer. Figure 1(a) shows the structure of conventional PCM cell.

Phase change material, chalcogenide, has two stable states and can switch back and forth between two stages: the amorphous state that has high resistance(because of the disordered crystalline lattice) and polycrystalline state that has low resistance(because of the regular crystalline structure). PCM utilizes the resistance difference to store bit information.

Figure 1(b) shows the set and reset mechanisms of PCM technology [7]. The phase change of chalcogenide is induced by intense localized Joule heating. To $RESET$, writing bit "0", a PCM cell, a high but short voltage pulse is applied to the phase change material and switches it from the polycrystalline(low resistivity) state to the amorphous state(high resistivity). To SET, writing bit "1", a PCM cell, a low but sustained voltage pulse is applied to switch the material back to the polycrystalline state. Typically, RESET operation has short latency but consumes high power, while SET operation has long latency but consumes low power. So the energy consumption for the RESET operation is much higher than that of the SET operation. To read the state of phase change material, a low enough voltage pulse is applied to the material. The bit information is distinguished according to the current difference. Both the read latency and the read energy of PCM are low.

The large resistance difference between crystalline and amorphous states makes it possible to utilize intermediate resistive states to store multiple bits in one PCM cell, referred as Multiple Level Cell (MLC) PCM. The cell that stores one bit is referred as Single Level Cell (SLC) PCM [4]. Figure 2 shows a typical 2-bit MLC with 4 states. Here, "10" and "01" are the two intermediate state compared with the states of "11" and "00". For MLC PCM, the iterative programming technique (programming-and-verify) is used to write MLC cells. A sequence of precisely controlled SET and RESET operations adjusts the fraction of crystalline materials in a cell and can set the resistance to an intermediate value. In MLC PCM cells, the actual resistance of a written cell is a random variable whose distribution is lognormal [8] [9].

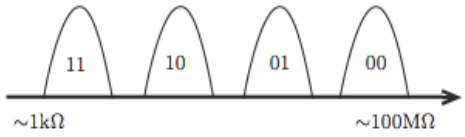

Fig. 2. 2-bit MLC with 4 states.

B. Vulnerability Models

PCM memories have high scalability. PCM's higher density compared to DRAM, coupled with the increasing need for memory capacity in computing, make it likely for future systems using PCM to be provisioned with much larger main memories. This potentially implies an even larger amount of data persisting on the main memory due to non-volatility and increasing the incentive to attack the system [3]. Its non-volatility opens many interesting new doors for system optimization.

The basic vulnerability is the one in which an attacker obtains physical access to the system, and extracts sensitive information from the storage system by reading it. All current computing systems store information in the main memory in plaintext form. There is no software solution that encrypts data in the main memory because the software itself must store its code, data and program variables in the main memory. The lack of secure main memory is acceptable to many users when DRAM is used as the main memory because once powered off, information is not retained in the main memory. PCM incurs a security vulnerability because information lingers on in the main memory long after the system is powered down [10].

Another vulnerability for PCM cells suffers from limited write endurance. Each PCM cell is projected to endure a maximum of about 10^7 to 10^9 writes. One solution to implement secure wear leveling is to perform a randomized remapping of memory lines [11].

While PCM has read power and delay in the same realm as DRAM, writes are very different [12], [13]. Compared to DRAM, PCM writes consume significantly more power, and take significantly more time to complete. The problem is that a write to a PCM cell is an inherently power-intensive operation. Delivering this power is a serious challenge, so PCM systems limit the total number of concurrent writes allowed.

Reports of scavenging sensitive data from discarded disks, snooping of the data bus, micro-probing, electromagnetic and power analysis, and other side channel attacks exist in literature [14]. If PCM is to become a viable main memory technique, it needs to overcome at least the following three challenges: inferior performance seen by both reads and writes, limited cell lifetime due to wear, and the higher power required per access(especially with respect to writes) [12].

III. Threat Models of PCM

This section presents several threat models in PCM main memory. The threat models provided in this section are not conceptually identical to traditional threat models from the adversary view which describes the ability of the adversaries; the threat models in this section show the vulnerabilities that the PCM model faces.

A. Non-volatility

A PCM main memory has both advantages and disadvantages over its volatile counterpart. An advantage of PCM, no refresh power is consumed to maintain code and data compared with DRAM, and resumption from sleep or hibernation can be made instantaneous. One disadvantage, non-volatility makes data more vulnerable to be attacked by malicious attacks since data stored in PCM main memory can be retained after power-off. For instance, an attacker can physically remove the main memory and extract sensitive information from it through the techniques of memory scanning [3], [15]. As the portable devices, such as smartphones and tablet computers, become more and more popular, this security problem is more severe which gives attackers more opportunities to get physical access to systems and to launch such physical attacks [3].

Prior methods used to solve the non-volatility security problem use data encryption, such as AES-based and counter-mode XOR based [3]. The overhead of AES algorithm cannot be avoided. Besides the non-trivial overhead induced by encryption schemes and storage requirements to the keys, these approaches are not optimized for PCM as main memory. The encryption process can further harm the write endurance of PCM cells due to the extra write operations caused by encryption and decryption. This leads the efficiency of write reduction approaches for PCM cells to be significantly degraded. To increase the write endurance, the wear-leveling is considered a practical solution, however, the address remapping process in wear-leveling can also be considered as a type of data encryption, which can be leveraged to reduce the design complexity of encryption.

To leverage the encryption schemes, it first needs to declare a feasible attack model to PCM main memory. Here one reasonable attack model can be based on the following assumptions to facilitate the potential attacks.

- The attacker can access PCM cells and has the right to write specific data into PCM as a normal user. The written data may be a malicious code, which can be used in the later attack.
- PCM main memory is not physically secure, so that the PCM main memory can be physically obtained by an attacker either after power-off or during runtime execution.
- PCM main memory is a universal memory. It can be plugged into an attack system in which all cipher-data (either plaintext or ciphertext) can be scanned out, using scanning techniques, for an attack.

The goal of the attacker is to find out sensitive plain data from cipher-data retained in PCM cells using statistical or computing methods.

To overcome the limitations of current encryption schemes, an encryption scheme must be found to satisfy several key requirements. *1) Security.* All data in memory must be encrypted during execution against potential runtime attacks; *2) Overhead.* Both design complexity and runtime timing and power overhead should be moderate; *3) Compatibility.* Encryption method should work well with other optimization techniques for PCM.

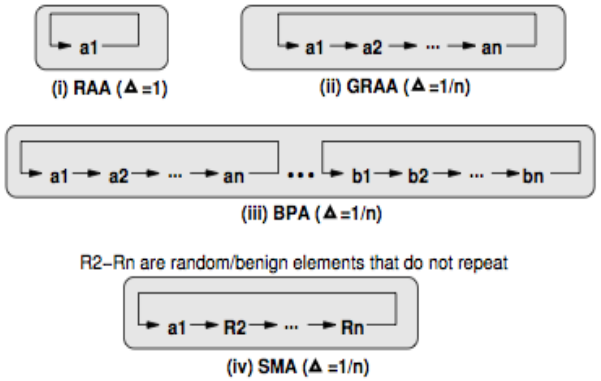

Fig. 3. Four types of attacks (i) Repeat Ad- dress Attack (RAA) (ii) Generalized RAA (iii) Birthday Paradox Attack (iv) Stealth Mode Attack [16].

B. Write Endurance

Besides the vulnerability of non-volatility in the secure consideration, another significant vulnerability is the limited write endurance. Each PCM cell is projected to endure a maximum of about 10^8 to 10^8 writes. Heavily written lines fail much faster than the rest of the lines, which may cause system failure much earlier than the expected lifetime.

Due to the limited write endurance, PCM memory is susceptible to malicious attack, which writes to certain PCM cells repeatedly, known as the selective attack. Without any protection, the PCM memory may fail in minutes under selective attack [2]. If the attacker simply writes to one address repeatedly, it only takes about 32 seconds to fail a memory cell [11]. The reason why selective attack could cause serious damage is that the program does not follow the application locality.

Three requirements are important for an attack to successfully cause failure in lifetime limited memories in a short time. It has to write to a *few lines*, *repeatedly* and at a *sufficiently high write bandwidth.*

Figure 3 shows the canonical form of typical four attacks for write endurance [16]. The symbol Δ represents *Attack Density* (AD), which is defined as the ratio of the number of writes to the most frequently written line to the total number of writes within a given time period. Figure 3(i) shows the Repeat Address Attack (RAA), which continuously writes to the same line. Figure 3(ii) shows the generalized RAA (GRAA), which continuously writes to n lines. Figure 3(iii) shows the Birthday Paradox Attack (BPA) (where $\Delta = \frac{1}{n}$), which changes the working set after every several million writes. Figure 3(iv) shows the Stealth Mode Attack (SMA), which attacks only one line but disguises it in other $(n-1)$ lines.

To defend the attacks, the ideal wear leveling scheme is required to concurrently satisfy at least four properties. *1) Secure Mapping.* The address mapping mechanisms should be invisible to the users. The robust way prefers the random changing mapping. *2) Efficiency.* The remapping scheme should make sure the mapped physical address of logical address is changed quickly in case that the logical address is attacked. *3) Sufficiency.* The potential physical address space to which the logical address can be mapped should be as large as possible so that the wear can be distributed sufficiently. *4) Low overhead.* The overhead induced by wear leveling scheme

should be as low as possible, since the induced overhead would potentially result in longer access latency and extra wear.

C. Side-Channel Attacks

Besides the vulnerabilities of non-volatility and write endurance, prior techniques did not consider the circumstances of a compromised Operating System (OS) and its security implication to the overall PCM design. A compromised OS allows adversaries to manipulate processes and exploit side channels to accelerate wear out.

Side Channel Attacks (SCA) have been known as a major threat to any unprotected cryptographic implementation in software and hardware. Lots of efforts have already been dedicated towards the development of corresponding countermeasures, in particular against Differential Power Analysis (DPA) [17]. However, a single and efficient countermeasure cannot provide complete protection against a large variety of SCA attacks.

The side channel attacks work when a signal difference exists between sensing "0" and "1". This difference creates a way to distinguish which signal/range represents value "1" and which signal/range represents "1". PCM as the main memory has asymmetrical write behavior compared with other kinds of memories; this causes PCM memory to be easily subject to SCA. Partially, this is because the PCM element itself or the write circuitry, which can be a source of side channel attacks on PCM. The writing asymmetry in the different write operations of PCM is a potential source of power or timing side channel information leakage and this leakage can be used as the source of vulnerability.

Based on the asymmetrical write behaviors of PCM, a compromised OS can allow a malicious process to obtain and assemble useful information leaked from side channels, such as timing attacks to deduce shuffling pattern in a wear leveling scheme [18], the wear leveling scheme will not stop adversaries from tracking, pinpointing and wearing out target PCM memory. Attacking a system can use various side channels, such as time, power, electromagnetic emission and architectural vulnerability.

Information can leak through side channels. A sufficient amount of leaked information, such as time and power, allows an adversary to assemble this useful knowledge and devise a dedicated side channel attack for target PCM locations. However, simple hiding internal memory addresses alone will not address this issue properly. Hence, the designed scheme should consider constantly updating the address mapping to obfuscate any relationship among information leaked from side channels.

To defend the side channel attacks, the ideal designed scheme for SCA must concurrently satisfy at least three requirements. *1)* To obfuscate the address information regarding the actual physical data placement from applications, the (compromised) OS, and the memory controller. *2)* To obfuscate potential side channel leakage. *3)* To prohibit any physical tampering, such as memory bus probing.

D. Other Models

Besides the three main threat models, non-volatility, write endurance and side-channel attacks, in this section, we present several not too common threat models to consider when designing a specific system.

1) Resistance Drift: A PCM cell is a controllable resistor whose resistance can be set at one of the several levels, but the resistance slowly changes over time. Compared to Single Level Cell (SLC) PCM, a Multi-level Cell (MLC) PCM has higher density and larger capacity, its programming latency is longer, and it has slow growth in cell resistance. The long programming latency further degrades memory system performance. Unfortunately, cell resistance is not constant over time; it drifts. Moreover, slow growth in cell resistance with time, resistance drift [19], can cause transient errors in MLC PCM, which could be a potential threat target.

Resistance drift may cause errors to happen in PCM cells. Once a cell is programmed to a certain state through the heating process, the cell resistance increases over time. However, the cell resistance may drift into the next state region, and then the sensing (read) circuit will read a different value other than the one that was written to that cell. This phenomenon is analogous to charge leakage in the DRAM cells.

In MLC PCM cells, the actual resistance of a written cell is a random variable whose distribution is lognormal: the logarithm of the resistance of the written cell is normally distributed, with a mean at or close to a nominal value, and with some standard deviation [9]. Let the cell be programmed at time $t = 0$, and let the cell resistance be sensed as R_0 after a very small amount of time t_0. Then resistance $R(t)$, the cell resistance at time t ($t > t_0$), can be modeled as the following Equation [20]:

$$R(t) = R_0 * (\frac{t}{t_0})^{\alpha}.$$

The exponent α ($\alpha < 1$) determines the drift rate. Due to process variation, every cell experiences different drift rates. Cells with higher drift rate (α) will suffer errors more quickly. Typically, drift errors occur only in intermediate states.

Given high drift rates, managing drift-induced errors is the key to enabling practical MLC PCM. To mitigate drift errors, it needs to apply a set of optimization techniques to conventional design, including smart cell encoding and optimal state mapping. For example, in paper [19], the authors proposed a new three-level cell (3LC) design that substantially reduces drift error rates to replace the conventional four-level cell (4LC) design.

2) Tamper Assisted Attack: In addition to non-invasive vulnerabilities discussed above, the adversary can deliberately alter the PCM memory content through non-invasive tampering. Its purpose is to set as many invalid (or dirty) bits as possible to increase the chances of the miss for the attack. There typically exist three kinds of tampers. *1) Magnetic Tampering.* The PCM memory can be exposed to external DC magnetic field in a direction that will flip the bits to "1" [21]. This may also corrupt some of the data bits. *2) Thermal Tampering.* The adversary can deliberately modulate the operating temperature with the intention to prolong the retention time to increase the number of persistent bits that can be compromised through unauthorized access at power-ON. *3) Miscellaneous Tampering.* There exist other tampering

techniques such as micro probing, radiation imprinting, and optical probing.

3) Denial of Service: Denial of service attacks corrupt the stored data or force the system into an unstable state. Data may also be corrupted by environmental effects such as heat and gamma rays. Environmental effects can be mitigated by error-correction codes and/or physical shielding. PCM memory is also susceptible to permanent damage by application of force. This can only be prevented by increased physical security.

IV. Security of PCM

In this section, we discuss the security issues of PCM as the main memory. First, we give four security requirements when designing a secure PCM main memory, then we discuss the potential security solutions based on the threat models in Section III and security requirements.

A. Security Requirements

One of the goals of this paper is to find a feasible solution to the security vulnerability of lingering data in the PCM main memory and mitigating write endurance and side channel attacks. A satisfactory solution to achieve the goal must satisfy at least four requirements.

1) Data Retention: This requirement means that we must preserve the original instant-on benefit of PCM main memory, which means that data should be retained in memory so that it can be recovered when the processor starts up or wakes up from hibernation. This requirement precludes flushing out data from the PCM cells when the system is powered off. Ideally, the data must be retained, but in an unintelligible or encrypted form so that attackers cannot recover any useful information from PCM main memory. This requirement can clearly be satisfied through encrypting data in memory, rather than discarding or flushing the data.

2) Self-contained: The self-contained property means that the encryption ability for PCM main memory should not depend on a particular processor platform, instruction set architecture, or require specific changes to the processor architecture. Since PCM main memory should be universal for high volume memory commodity, an encrypted PCM should comparable to a wide range of processor platforms it is attached to, such as servers, mobile devices and embedded systems. Also, its effectiveness should not be predicated on a specific instruction set or processor architecture changes. This requirement precludes the use of secure processor technology, such as Trusted Platform Module (TPM), that requires the processor-side engine to encrypt the main memory and necessitates the solution to have a memory-side cryptographic engine embedded in the memory module itself. Placing the cryptographic engine on the memory module, instead of the memory controller, avoids requiring changes to the processor chip as the memory controller can be integrated with the processor chip.

This requirement can be satisfied by architecting the solution entirely in the memory system, which means it allows the solution to be used in many processor systems on various platforms. The encryption engine must be located in the main memory module or device.

3) Secure: The encrypted PCM main memory should be as secure as its volatile predecessor, such as DRAM. The retention time of DRAM can be served as a limit for how long it takes to complete memory encryption after power down. It is preferable to keep much of the memory encrypted at all times, and only encrypt a small amount of data upon power events.

There exist at least two options to satisfy this requirement: *a)* Encrypting the entire memory on power-down; this does not incur execution time overheads during regular execution, but cannot match DRAM's retention time; *b)* Keeping the entire memory encrypted at all times. This approach has zero retention time at power down, but suffers from high-performance overheads because every memory access by the processor must incur decryption latency.

4) Low Overheads: The low overheads, such as performance and energy, requirement states that the security solution should not incur substantial performance or energy overheads for applications running on the system. This requirement partly conflicts with the third requirement since encryption and decryption have lots of overheads. It needs the feasible coordination between secure and low overheads.

To design a feasible solution for PCM main memory, it should consider all the four requirements together. None of these methods alone or parts is a comprehensive solution for protecting PCM memories from threats.

B. Potential Security Solutions

Based on the security requirements of PCM, this section will discuss some potential hardware security solutions that leverage PCMs for better resilience and performance of security solutions. Instead of discussing how to secure PCM cells itself, we will discuss the potential security solution using PCM as the main memory.

1) PCM-based PUF: We intend to use Physical Unclonable Function (PUF) as the primitive, which is one of the dominant topics in the hardware security domain. PUF is a physical implementation of a function that maps an input *challenge* to an output *response*. The physical implementation needs to rely on a physical disorder so that each instance of the PUF creates a unique challenge -response mapping and thus cannot be cloned.

Digital keys are traditionally stored in non-volatile memory, such as PCM, for cryptographic applications. However, these keys in PCMs are vulnerable to invasive physical attacks. To secure these keys, PUF is receiving increased attention because PUFs offer a simple alternative to generating unique volatile digital keys in a very small hardware device without the need for tamper-sensing mechanisms. PUFs are easy to build but practically impossible to duplicate since PUFs rely on uncontrollable physical parameter variations that occur during hardware device manufacture. Also, PUF derives from inherent complexity in a given physical system. Thus, PUFs can thwart the physical attack.

PCM-based PUF structures exploit abundant process variation, have a small footprint and lower energy consumption of PCM technologies together with a programming sensitivity feature to formulate the basis for PCM-based reconfigurable PUF (rPUF), which has the ability to change its response to

the same challenge. There already exist several practical PCM-based rPUF [22], [23] to provide the security solution for the memory systems.

2) Error Recovery: With the limited write endurance, repeating writes to a PCM cell causes the cell to be expanded and contracted repeatedly. This leads to mechanical stress and eventually incurs a permanent stuck-at-fault (SAF) failure. Furthermore, with scaling down technology, more PCM cells are subject to SAF failure. Thus, the error recovery scheme is needed to correct multiple stuck-at faults.

One important requirement for error recovery technique is that it must operate in the presence of existing wear leveling algorithms. Otherwise, error recovery makes the memory system vulnerable to malicious attacks, especially when the OS is compromised. Also, error recovery scheme should be lightweight enough to be embedded inside a chip [24].

Since PCM cells with a stuck-at value are still readable, this property can be exploited to reuse the faulty cell with stuck-at value to provide hardware efficient multi-bit stuck at fault error recovery. Paper [25] proposed a stuck-at fault error recovery technique (SAFER), which enables a hardware efficient multi-bit error recovery by dynamically partitioning the data blocks to ensure that each partition has at most one fail bit.

There exist other potential hardware security solutions that leverage PCMs for better resilience and performance, which need to be exploited further.

V. Conclusions

Although PCM has recently attracted attention, various security concerns continue to arise over PCM main memory. Sensitive data written to PCM cells persist even when the system is powered down and can be invaded easily. This paper systematically provided several kinds of threat models in PCM main memory. Meanwhile, this paper proposed several security requirements which can help to design a secure PCM. Based on security principles, two potential security solutions are proposed using PCM as main memory.

References

[1] X. Zhang, C. Zhang, G. Sun, J. Di, and T. Zhang, "An efficient runtime encryption scheme for non-volatile main memory," in *Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems*. IEEE Press, 2013, p. 24.

[2] N. H. Seong, D. H. Woo, and H.-H. S. Lee, "Security refresh: prevent malicious wear-out and increase durability for phase-change memory with dynamically randomized address mapping," in *ACM SIGARCH computer architecture news*, vol. 38, no. 3. ACM, 2010, pp. 383–394.

[3] S. Chhabra and Y. Solihin, "i-nvmm: a secure non-volatile main memory system with incremental encryption," in *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*. IEEE, 2011, pp. 177–188.

[4] F. Xia, D.-J. Jiang, J. Xiong, and N.-H. Sun, "A survey of phase change memory systems," *Journal of Computer Science and Technology*, vol. 30, no. 1, pp. 121–144, 2015.

[5] H. Yu and Y. Du, "Increasing endurance and security of phase-change memory with multi-way wear-leveling," *IEEE Transactions on Computers*, no. 1, p. 1, 2012.

[6] C. J. Xue, G. Sun, Y. Zhang, J. J. Yang, Y. Chen, and H. Li, "Emerging non-volatile memories: opportunities and challenges," in *Hardware/Software Codesign and System Synthesis (CODES+ ISSS), 2011 Proceedings of the 9th International Conference on*. IEEE, 2011, pp. 325–334.

[7] H.-S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010.

[8] D. H. Yoon, J. Chang, R. S. Schreiber, and N. P. Jouppi, "Practical nonvolatile multilevel-cell phase change memory," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, p. 21.

[9] M. Boniardi, D. Ielmini, S. Lavizzari, A. L. Lacaita, A. Redaelli, and A. Pirovano, "Statistical and scaling behavior of structural relaxation effects in phase-change memory (pcm) devices," in *Reliability Physics Symposium, 2009 IEEE International*. IEEE, 2009, pp. 122–127.

[10] S. Kannan, N. Karimi, O. Sinanoglu, and R. Karri, "Security vulnerabilities of emerging nonvolatile main memories and countermeasures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 1, pp. 2–15, 2015.

[11] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali, "Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2009, pp. 14–23.

[12] A. Hay, K. Strauss, T. Sherwood, G. H. Loh, and D. Burger, "Preventing pcm banks from seizing too much power," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2011, pp. 186–195.

[13] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali, "Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling," in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2009, pp. 14–23.

[14] M. Steil, "17 mistakes microsoft made in the xbox security system," in *22nd Chaos Communication Congr.*, 2005.

[15] J. Kong and H. Zhou, "Improving privacy and lifetime of pcm-based main memory," in *2010 IEEE/IFIP International Conference on Dependable Systems&Networks (DSN)*. IEEE, 2010, pp. 333–342.

[16] M. K. Qureshi, A. Seznec, L. A. Lastras, and M. M. Franceschini, "Practical and secure pcm systems by online detection of malicious write streams," in *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*. IEEE, 2011, pp. 478–489.

[17] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Cryptology Conference*. Springer, 1999, pp. 388–397.

[18] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," in *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2. ACM, 2007, pp. 494–505.

[19] D. H. Yoon, J. Chang, R. S. Schreiber, and N. P. Jouppi, "Practical nonvolatile multilevel-cell phase change memory," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. ACM, 2013, p. 21.

[20] D. Ielmini, D. Sharma, S. Lavizzari, and A. L. Lacaita, "Reliability impact of chalcogenide-structure relaxation in phase-change memory (pcm) cellspart i: Experimental study," *IEEE Transactions on Electron Devices*, vol. 56, no. 5, pp. 1070–1077, 2009.

[21] J.-W. Jang, J. Park, S. Ghosh, and S. Bhunia, "Self-correcting sttram under magnetic field attacks," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, p. 77.

[22] L. Zhang, Z. H. Kong, and C.-H. Chang, "Pckgen: A phase change memory based cryptographic key generator," in *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*. IEEE, 2013, pp. 1444–1447.

[23] L. Zhang, C.-H. Chang, A. Cabrini, G. Torelli, and Z. H. Kong, "Leakage-resilient memory-based physical unclonable function using phase change material," in *Security Technology (ICCST), 2014 International Carnahan Conference on*. IEEE, 2014, pp. 1–6.

[24] N. H. Seong, "A reliable, secure phase-change memory as a main memory," Ph.D. dissertation, Georgia Institute of Technology, 2012.

[25] N. H. Seong, D. H. Woo, V. Srinivasan, J. A. Rivers, and H.-H. S. Lee, "Safer: Stuck-at-fault error recovery for memories," in *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2010, pp. 115–124.