Efficient block cipher mode for NVM

A Khodjanov¹, F Rustamov¹ and J Yun¹

¹ Department of Computer and Information Security, Sejong University, 209 Neungdong-ro, Gawngjin-gu, Seoul 05006, South Korea

E-mail: jbyun@sejong.ac.kr (Corresponding author)

Abstract. The usage of non-volatile memory (NVM) storage devices is rapidly increasing in consumer and enterprise systems, providing high performance and low energy consumption compared to hard disk drives. Lately it is observed that NVM storage devices are widely used in cloud computing as well. However, existing disk encryption methods are designed for hard disk devices and an efficient block cipher mode of operation for NVM storage devices has not yet been studied in detail. In this work we have analyzed and evaluated performance of CBC, CTR and XTS-AES block cipher modes for whole disk encryption solutions and found that CTR and XTS-AES modes outperform CBC mode under the same scenarios.

1. Introduction

Storage technology is evolving very rapidly. The next generation of storage technology Non-Volatile Memory (NVM) has attracted more and more attention from both industrial and academic researchers [1]. NVM storage technology is just like HDDs, but more importantly has DRAM-like performance and byte-addressability. In NVM storage devices, the smallest write unit is one page (such as 4KB) and write operations can only be performed out-of-place, since data blocks have to be erased before new data can be written. Random writes especially can cause internal fragmentation of NVM storage devices and thus lead to a higher frequency of expensive erase operations. Besides performance degradation, the lifetime of NVM storage devices can also be dramatically reduced by random writes.

Security of data in storage devices is one of the main issues in computer security. Securing storage devices using encryption is a common method to deal with this problem. There are several disk encryption methods that are available, which include software-based encryption, controller based encryption and internal disk encryption. These methods provide solutions for encrypting data in motion and at rest. However, existing disk encryption methods are designed for hard disk devices, and a separate study on block cipher performance for NVM storage devices is needed.

Among many encryption/decryption standards, Advanced Encryption Standard (AES) is the most widely used symmetric cryptographic algorithm, which has been chosen as a secure encryption algorithm by the National Institute of Science and Technology (NIST) [2]. Existing encryption technologies like BitLocker [3] are using AES-CBC block cipher mode and cannot be applied to the NVM type of storage devices considering that it will impose security and performance degradation. Taking this into account, we have conducted an experimental study to measure the performance of different block cipher modes and tried to find the most efficient one for NVM.
2. Block ciphers for NVM

2.1. Block cipher

In cryptography, a block cipher is an algorithm that converts fixed-length groups of N bits, called blocks, to a cipher block of N bits with an unvarying transformation that is specified by a symmetric key [4]. Block ciphers operate as important elementary components in the design of many cryptographic protocols, and are widely used to implement encryption of bulk data. There are five modes of operation for AES. Each of these modes has different characteristics. The five modes are ECB (Electronic Code Book), CBC (Cipher Block Chaining), CFB (Cipher Feedback), OFB (Output Feedback), and CTR (Counter).

In this paper, we will cover CBC mode since it is well known and mostly used in disk encryption and proposed CTR mode. This mode is amenable to parallelization due to the fact that you can split the message into chunks, each chunk having a range of counter values associated with it, and encrypt (or decrypt) each chunk independently. In addition to this, we have proposed XTS-AES mode because this mode is the most common if you are encoding random accessible data (like a hard disk or RAM).

2.1.1. Cipher Block Chaining (CBC) mode

The Cipher Block Chaining (CBC) is a block cipher mode of operation that is known to be the most commonly used mode in disk encryption nowadays. This mode is a secrecy mode whose encryption procedure highlights the joining ("chaining") of the plaintext obstructs with past ciphertext squares [4]. Disk encryption in CBC mode is shown in Fig. 1, it may encrypt each sector with a different initialization vector (IV), which is an important constraint. In CBC mode, encryption is not conducted in parallel. This is a major reason for performance degradation compared with other modes.

![CBC mode (Encryption)](image1.png)

**Figure 1.** CBC mode (Encryption) [4].

![CTR mode (Encryption)](image2.png)

**Fig. 2.** CTR mode (Encryption) [4].
2.1.2. Counter (CTR) mode

Fig. 2 shows the encryption in CTR mode. Counter mode turns a block cipher into a stream cipher. It generates the next keystream block by encrypting successive values of a "counter". The counter can be any function that produces a sequence which is guaranteed not to repeat for a long time, although an actual increment-by-one counter is the simplest and most popular.

2.1.3. XTS-AES mode

The XTS-AES mode is based on the concept of a tweakable block cipher, introduced in [5][6]. Fig. 3 depicts the encryption of XTS-AES mode. XTS-AES mode assures that the same plaintext data unit will encrypt to two different ciphertext data units for two different data unit positions. This is because of the tweak of tweakable block cipher, so XTS-AES mode has better grade for security [7][8].

2.2. Efficient block cipher module for NVM

Based on our study analysis it was found that CTR mode has more advantages of all other chaining modes approved by NIST. Unlike other modes of encryption and decryption, CTR mode can be done in parallel on multiple blocks of plain or cipher data. The encryption/decryption throughput is only limited by the amount of parallelism that is achieved and this mode can be efficiently implemented on the hardware level. CTR mode makes it possible to use very low latency encryption since it allows users to pre-calculate the key streams used in XOR functions. With CTR mode the $i$th block of plain or cipher data can be processed in random-access fashion, which is a very common access pattern for disk read/writes.

3. Performance evaluation

To assess different block cipher modes we have setup a Linux 4.4 machine. The synthetic benchmark platform consists of an Intel i5-4460 CPU and an 8GB DDR3 memory. Since commercialy available NVM is limited at the time of conducting our experiments, we used DRAM as a storage device. A well-known benchmarking tool Flexible IO (FIO) [9] was used in our performance measurements.

3.1. Benchmark results

Experimental results show that CTR mode and XTS-AES mode well outperform CBC mode in most of the cases we were interested in. Fig. 4 illustrates the results of sequential read and write operations. It shows that there is not much difference in performance for sequential read but in sequential write in CTR mode, it dramatically outperforms CBC mode for blocks larger than 4KB.
The second set of experiments compare throughput rates for random read and write. Fig. 5 illustrates that there is not much difference in random read performance as well, but again random write is far ahead in CTR mode and XTS-AES mode.

Our final set of experimental analysis was aimed to prove that CTR brings less latency on the encryption and CPU utilization when parallel encryption and decryption was enabled. Fig. 6 demonstrates much lower latency rates for CTR mode than in CBC mode. XTS-AES mode has 7% more overhead than CTR mode on average. The CPU high utilization in CTR mode illustrates that during synthetic benchmark all cores were used for encryption operations and therefore boost the throughput rate.
3.2. Security strength
CTR mode works faster than XTS-AES mode but CTR mode is malleable – an attacker can flip plaintext bits by simply flipping the corresponding ciphertext bits. XTS-AES is less malleable and it does the XOR operation twice on the record, once before encryption and once after the encryption, therefore it provides high-grade security. Although XTS-AES mode has 7% more overhead than CTR mode, we recommend to use XTS-AES mode because of its security. With technology advances in NVM development this speed drawback is very negligible. In addition, in the absence of authentication or access control, XTS-AES mode provides more protection than other approved confidentiality-only modes against unauthorized manipulation of the encrypted data [10].

4. Conclusions
In this paper, we have evaluated the performance of AES encryption algorithm with different block cipher modes to find the most efficient mode for NVM storage encryption. It is clear that CTR mode has performance improvement with lower latency. The study also illustrates CTR mode outperforms CBC mode due to the support of parallel encryption and decryption operations. Results indicate that there is only a negligible difference between CTR and XTS-AES modes. They reached almost the same percentage and number. Therefore, XTS-AES is the most suitable block cipher mode for NVM storage given the efficiency and protection.

5. Acknowledgments
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No.2015R1C1A1A02036511).

6. References
[1] Chen C, et al. 2016 Fine-grained metadata journaling on NVM In 2016 IEEE 32nd Symposium on Mass Storage Systems and Technologies, pp. 1-12 (DOI: 10.1109/MSST.2016.7897077).
[2] NIST: Announcing the ADVANCED ENCRYPTION STANDARD (AES), Nov. 2001, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf
[3] Ferguson N 2006 AES-CBC + Elephant diffuser A Disk Encryption Algorithm for Windows Vista, Microsoft Corporation https://css.csail.mit.edu/ 6.858/2010/readings/bitlocker.pdf
[4] Stallings W 2010 Cryptography and Network Security Principles and Practice (Prentice Hall. Boston, 5th Edition, 192.
[5] Liskov M, et al. 2011 Tweakable Block Ciphers Journal of Cryptology 24(3), pp. 588-613 (DOI: 10.1007/s00145-010-9073-y).
[6] Minematsu K, et al. 2009 Generation and Extension of XEX Mode IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences E92-A 2 (DOI: 10.1587/transfun.E92-A.517).
[7] IEEE Computer Society: “IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices,” IEEE Std 1619™-2007 (2008), pp. 1-32, (DOI: 10.1109/IEEESTD.2008.4493450).
[8] Dworkin M 2010 Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices NIST Special Publication 800-38E (DOI:10.6028/NIST.SP.800-38E).
[9] Flexible I/O Tester, https://github.com/axboe/fio
[10] Rogaway P 2011 Evaluation of Some Blockcipher Modes of Operation, Technical report, Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan.