Performance Analysis of Encryption in Securing the Live Migration of Virtual Machines

Yaohui Hu Sanket Panhale Tianlin Li Emine Kaynar Danny Chan Umesh Deshpande Ping Yang Kartik Gopalan
Computer Science, State University of New York at Binghamton
Email: {yhu15,spanhal1,tli16,ekaynar1,dchan20,udeshpa1,pyang,kartik}@binghamton.edu

***Abstract*—Virtual machine (VM) migration is a technique for transferring the execution state of a VM from one physical host to another. While VM migration is critical for load balancing, consolidation, and server maintenance in virtualized datacenters, it can also increase security risks. During VM migration, an attacker with sufficient privileges can compromise a VM by modifying its memory contents during transit to subvert its applications or the guest operating system. One could maintain dedicated, and presumably more secure, control networks to carry the migration traffic, but at significant hardware and administrative complexity. Alternatively, one could encrypt the migration traffic, which eliminates the need for dedicated control networks, but might introduce performance overheads. To date, there has been no systematic study of how encryption affects VM migration, especially in high-bandwidth low-delay networks that are common within datacenters. In this paper, we present a study of the impact of AES and 3DES encryption algorithms on two widely used live VM migration approaches – pre-copy and post-copy. Our key findings are as follows. The encryption algorithm used can have a significant impact on the total migration time. The impact of encryption on downtime varies with the type of the migration technique. The overhead of encryption also depends upon the relative speeds of source and target machines. Finally, an application's performance within a VM during encrypted migration varies with the type of the application and the migration mechanism.**

I. Introduction

Virtual machine (VM) migration is used in virtualized datacenters and cloud computing environments for various administrative tasks such as load balancing, consolidation, and server maintenance. VM migration refers to the transfer of a VM's execution state from one physical machine (called "source machine") to another physical machine (called "destination machine"). There are two types of VM migration approaches: stop-and-copy and live migration. The stop-and-copy migration suspends the VM's execution on the source machine, copies the VM to the destination machine, and then resumes the VM on the destination machine. When the size of the VM is large, stop-and-copy results in high downtime, during which the VM does not make any progress. Live VM migration, on the other hand, allows the VM to continue running during the state transfer. Most hypervisors such as VMware [1], Hyper-V [2], KVM [3], Xen [4], and Virtualbox [5] support live VM migration.

Despite its benefits, VM migration also gives rise to security challenges. Attackers may modify the memory contents of a VM during migration in order to subvert its applications and the operating system (OS). For example, Oberheide et al. [6] demonstrated how attackers can manipulate the object code of sshd's authentication routines during VM migration and gain root access to the guest OS after the migration.

To secure live VM migration, VMware recommends [6] using a separate network (or a VLAN) dedicated for migration. A major drawback of this approach is the growth in complexity and administrative costs as the VM population grows [7]. Alternatively, some products such as Oracle VM [8], HP VM [9], and Proxmox VM [10] use encryption or SSL to protect migration traffic. Authors in [11] considered the problem of enabling secure VM-vTPM (Virtual Trusted Platform Module) migration in private cloud environments by encrypting the migration traffic using the vTPM keys. Encryption eliminates the requirement for dedicated networks or additional hardware, and can be used to protect migration traffic in different types of networks, such as LAN, metropolitan area networks(MAN), campus area networks (CAN), and wide area networks (WAN). A drawback of encryption is that it may slow down the VM migration in high-bandwidth low-delay networks, if the processing of encryption/decryption is slow.

Encryption can be used with different live VM migration techniques such as pre-copy [12], [13] and post-copy [14]. Most of the existing literature on securing VM migration assumes the use of pre-copy migration with encryption. However, the performance impact of using encryption can differ with different migration mechanisms due to the manner in which content is transmitted to the destination. To the best of our knowledge, existing research has not considered the impact of encryption on the performance of different VM migration approaches. For instance, it is an open question whether the fastest VM migration approach without encryption will still remain the fastest approach with encryption, and how different applications and server workloads affect the performance of VM migration with encryption.

Main contributions: In this paper, we study the impact of Advance Encryption Standard (AES) [15] and Triple Data Encryption Standard (3DES) [16] encryption on KVM pre-copy and post-copy migration for different application workloads and networks. We chose AES and 3DES because both are commonly used and are considered secure. AES is much faster than 3DES, which enables us to evaluate how different encryption speeds affect the performance of VM migration. Our experiments illustrate the following findings that are not reported in prior literature.

2159-6190/15 $31.00 © 2015 IEEE
DOI 10.1109/CLOUD.2015.87

- When using 3DES encryption with KVM pre-copy, the migration of a 128MB VM running a memory-write-intensive application does not terminate even after 30 minutes. Our analysis shows that the number of dirty pages decreases initially and then becomes stable after certain number of iterations. It is not clear whether KVM pre-copy would ever terminate in this case. On the other hand, when using AES encryption or no encryption, migration completes within 20 seconds.
- The affect of an encryption algorithm on VM downtime varies with the type of the migration technique. For instance, we find that 3DES significantly increases the downtime of KVM pre-copy, but has no effect on the downtime of KVM post-copy. We also found that the estimated downtime computed in the KVM pre-copy implementation was inaccurate; the actual downtime is significantly higher than the estimated downtime.
- When a VM is migrated between a fast machine (i.e. having a fast CPU) and a slow machine, the overhead of encryption differs depending upon the direction of the migration. For instance, with both pre-copy and post-copy, AES imposes higher overhead when migrating a VM running a memory-write-intensive application from a fast to a slow machine than in the reverse direction.
- The performance impact of encryption during migration on a VM's applications varies with the type of the application and the migration mechanism. For example, network-intensive applications, such as Netperf, experience a greater performance degradation when using encryption with post-copy than with pre-copy. On the other hand, CPU-intensive applications, such as Kernbench, show greater performance degradation when using encryption with pre-copy whereas no observable degradation is seen with post-copy.

The rest of this paper is organized as follows. Section II provides an overview of the live migration approaches on KVM. Sections III, IV, V, and VI present the evaluation of AES and 3DES in securing KVM live migration on various workload and applications. The related work is described in Section VII and Section VIII concludes the paper.

II. Preliminaries

A. Pre-copy Live Migration

Pre-copy [12], [13] is the most widely used live VM migration approach, which has been implemented in Xen, KVM, and VMware ESX server. In the KVM pre-copy implementation, the source machine first sends all memory pages of a VM to the destination machine, and then iteratively sends pages modified (dirtied) in the previous iteration to the destination. When the estimated downtime is less than a threshold (30 millisecond in KVM pre-copy by default), the source machine suspends the VM and transfers the remaining dirty pages, the hardware device state, and the CPU state to the destination. The VM is then resumed on the destination machine. During the iterative pre-copy rounds, the guest OS and all applications inside the migrated VM continue execution at the source.

The time between suspending the VM on the source machine and resuming the VM on the destination machine is called *downtime*. The *total migration time* includes the time when migration starts till the time when migration ends. Existing research aims to balance the migration time and the downtime to reach an optimal migration performance.

To estimate the downtime, KVM pre-copy first computes the available bandwidth as the number of bytes transferred in the previous iteration divided by the duration of the previous iteration. The estimated downtime is then computed as the number of remaining dirty bytes divided by the estimated bandwidth. KVM pre-copy has also implemented an optimization that avoids transferring a page if all bytes in the page are the same. In such case, only 8 bytes (64-bit), instead of the entire page, are sent to the destination.

B. Post-copy Live Migration

In post-copy migration [14], [17], the source machine first transmits the minimal execution context to the target machine where the VM starts execution immediately. The source machine then actively pushes pages to the target machine and predicts the next page that the VM may access well before it faults on the page. If the VM at the target faults upon a missing page, the target will request the source for the page. Post-copy ensures that each memory page is transferred at most once, thus avoiding the overhead of transmitting duplicate pages. The downtime of post-copy migration is technically very small, equal to the time to transmit the VCPU and I/O device state to the target machine. However, the performance of the VM at the target right after resumption can be significantly slow until the working set of the VM is received from the source, either through active push or demand paging. The compression optimizations used in KVM pre-copy can also be applied to KVM post-copy.

III. Evaluation Setup

This section describes the setup used to evaluate the performance of AES and 3DES in securing KVM pre-copy and post-copy migration using various workloads and applications.

We chose QEMU/KVM hypervisor because it supports both pre-copy and post-copy migration techniques. We used pre-copy implementation in qemu-kvm version 0.12.3 and Yabusame QEMU/KVM post-copy implementation [17].

Migration is performed through SSH tunneling using OpenSSH version 5.9 (SSH2 protocol). The ciphers we used are 128-bit AES and 3DES with cipher block chaining mode. SSH2 protocol does not support unencrypted channels. To enable fair comparison, we modified the implementation of SSH2 protocol to add an option for unencrypted channels.

Our test environment consists of three machines connected through a Gigabit Ethernet switch with 1Gbps full-duplex ports. Virtual disks are accessed by each VM over the network from a NFS server; this allows a VM to access its storage from both source and destination machines without the need for migration. The configurations of the three machines used are given below.

- **Machines F_1 and F_2:** Host system with 3.30GHz Intel dual-core i3-2120 CPU, 2GB of RAM, and running Ubuntu 12.04 (precise) 3.5.0-23-generic.
- **Machine S:** Host system with 2.60GHz Pentium dual-core CPU E5300 processor, 4GB of RAM, and running Ubuntu12.04 (precise) 3.5.0-23-generic.

Note that machines F_1 and F_2 have the same configurations and are faster than machine S. The size of the VM migrated is 1GB and the OS installed in the VM is Ubuntu 12.04. VM workloads consist of three benchmarks.

- A network-intensive application Netperf [18].
- A CPU-intensive application Kernbench [19].
- A memory-intensive synthetic benchmark that either reads or writes to a large region of main memory.

For all benchmarks, we measured the total migration time, the total number of pages transferred from the source to the destination, the downtime, and the migration bandwidth (computed by dividing the number of bytes transferred during migration by the total migration time)[1]. In addition, we also measured how pre-copy and post-copy migrations (with and without encryption) affect the network bandwidth reported by Netperf and the time for compiling a Linux kernel source tree using Kernbench. Below, we list notations used in various figures that follow in subsequent sections.

- **NONE :** Migration without encryption.
- **AES/3DES:** Migration with AES/3DES encryption.
- **Pre-copy/Post-copy M_1 M_2:** Pre-copy/Post-copy migration from machine M_1 to machine M_2.
- **Net/Minor:** Network/minor page faults in post-copy.

IV. Memory-intensive Workload

This section evaluates the performance of AES and 3DES encryption in securing KVM pre-copy and post-copy migration using memory-intensive workloads. We wrote a synthetic benchmark using C that either repeatedly writes or reads random numbers to/from a large region of memory. The size of the working set (i.e., the size of the memory written/read) ranges from 64MB to 768MB. The benchmark starts as soon as the VM starts and the VM is migrated when the program is running within the VM.

A. Write-Intensive Application

Figures 1 and 2 compare the performance of pre-copy and post-copy when migrating a VM running a write-intensive benchmark with and without encryption, respectively.

Migration Bandwidth: Our experimental results show that the relative speeds of the source and destination machines affect the overhead of encryption on VM migration. As shown in Figure 1(c), AES imposes almost no overhead on migration bandwidth when the VM is migrated from F_1 to F_2 using pre-copy. This indicates that encrypting a page with AES on F_1 and F_2 is as fast as transmitting a page from F_1 to F_2. AES imposes 12.7%–14.9% overhead when the VM is migrated from F_1 to S and from S to F_1. This is because encrypting/decrypting one page on S (the slower machine) with AES is slower than transferring one page from F_1 to S.

3DES imposes high performance overhead on KVM pre-copy when the size of the working set is 64MB. In addition, with 3DES, migration using KVM pre-copy does not complete even after 30 minutes when the size of the working set is 128MB or larger. Our analysis shows that the number of dirty pages decreases initially and then becomes stable after certain number of iterations. It is not clear whether pre-copy would ever complete unless we impose a hard limit on the number of pre-copy rounds, in which case downtime would be significantly high.

Post-copy is faster than pre-copy. As shown in Figure 2(c), AES imposes almost no overhead when migrating the VM from F_1 to F_2 using post-copy. AES imposes the highest overhead (12%–15.1%) when migrating the VM from F_1 to S. This is because, (1) decrypting a page on S is slower than transmitting a page from S to F, and (2) the memory write-intensive application is running on S during pre-copy, which further slows down AES decryption on S. In addition, AES imposes 2.3%–10.9% overhead when migrating the VM from S to F_1 because AES encryption is slow on S. With 3DES, the migration bandwidth of post-copy is significantly lower than that without encryption. Figure 2(i) shows that pre-copy migration using 3DES between F_1 and S is significantly worse than between F_1 and F_2.

Downtime: The affect of an encryption algorithm on VM downtime varies with the type of the migration technique. The KVM pre-copy iteratively copies dirty pages from the source to the destination until the estimated downtime is less than a given threshold (30 milliseconds by default). Figure 1 (d) shows that the downtimes of pre-copy without encryption and with AES are close (0.2–0.6 seconds) and is significantly higher than the threshold. In addition, 3DES results in significantly higher downtime (1–2.5 seconds). This shows that the downtime estimated in pre-copy is inaccurate due to incorrect accounting of bytes sent for compressed uniform pages.

With Post-copy, the downtime of migration is always less than 0.01 seconds. AES and 3DES do not impose observable overhead on downtime.

Network and minor page faults of post-copy: A network page fault occurs when a page needed in the destination has not been transmitted from the source. If the VM faults upon a missing page that is already transmitted to the destination, then the page fault is a minor page fault. When a network page fault occurs, the destination machine requests the source machine to send the page. As a result, network page faults affect the migration time more than minor page faults. Figure 3 shows that the number of network page faults is the highest when migrating the VM from S to F_1, especially with encryption. This is because, (1) AES and 3DES encryption are slower on S, which results in pages reaching the destination more slowly, and (2) the VM runs faster on F_1 than S, which results

[1] Note that the total number of bytes transferred could be less than the VM size, due to runtime compression of uniform pages, or more due to retransmission of dirtied memory in pre-copy. Further, different runs could yield different number of bytes transferred for the same configuration, depending upon the number of zero (or uniform) pages in VM's memory.

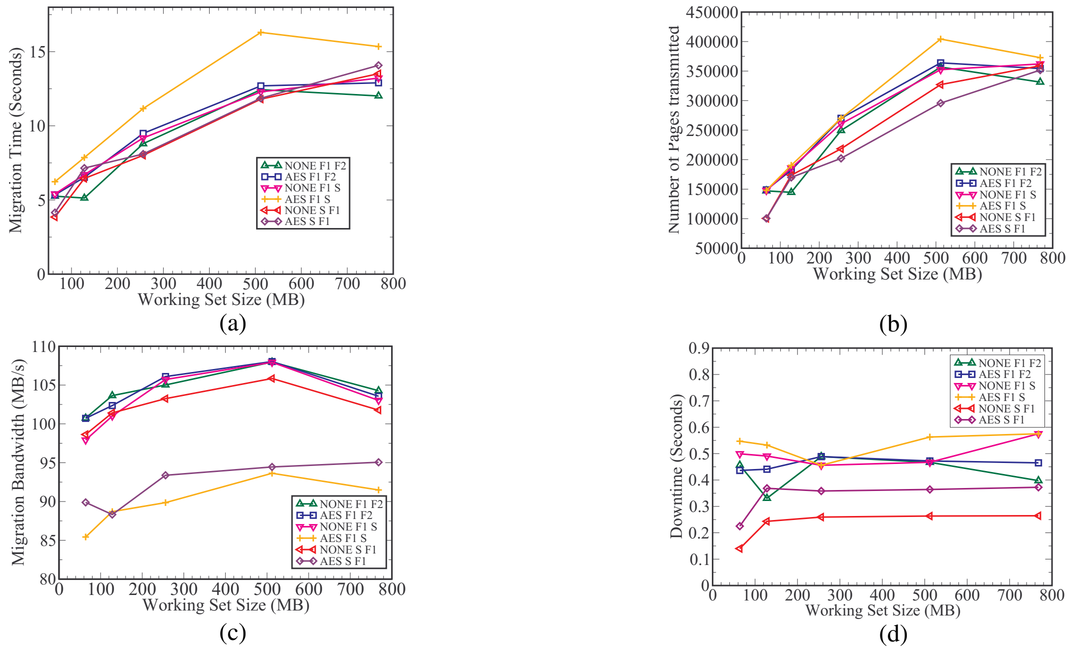


Fig. 1. Pre-copy migration of a VM running a memory-write-intensive application (AES, NONE): (a) total migration time, (b) total number of pages transferred, (c) migration bandwidth, and (d) downtime.

in more missing pages. The number of minor page faults is the highest when migrating the VM from F_1 to F_2 (with and without encryption).

B. Read-Intensive Application

Figure 3 shows the results of migrating a VM running the read-intensive benchmark using pre-copy. For post-copy, the experimental results for the memory-read-intensive application are very close to those for the write-intensive benchmark and hence are not plotted.

Our results show that, with KVM pre-copy, the total migration time for the memory-read-intensive program grows much more slowly than that for the write-intensive program when the size of the working set increases. This is because the memory-read-intensive application does not write data to the memory and hence results in fewer dirty pages than the memory-write-intensive application. Also, similar to the memory-write-intensive application, AES does not impose any overhead when migrating the VM from F_1 to F_2 using pre-copy, and imposes the highest overhead when migrating the VM from F_1 to S.

Finally, as with write-intensive application, the downtime of pre-copy with AES encryption is almost the same as that without encryption, and is significantly higher than the estimated downtime. With 3DES, the downtime of pre-copy is significantly higher than that of post-copy.

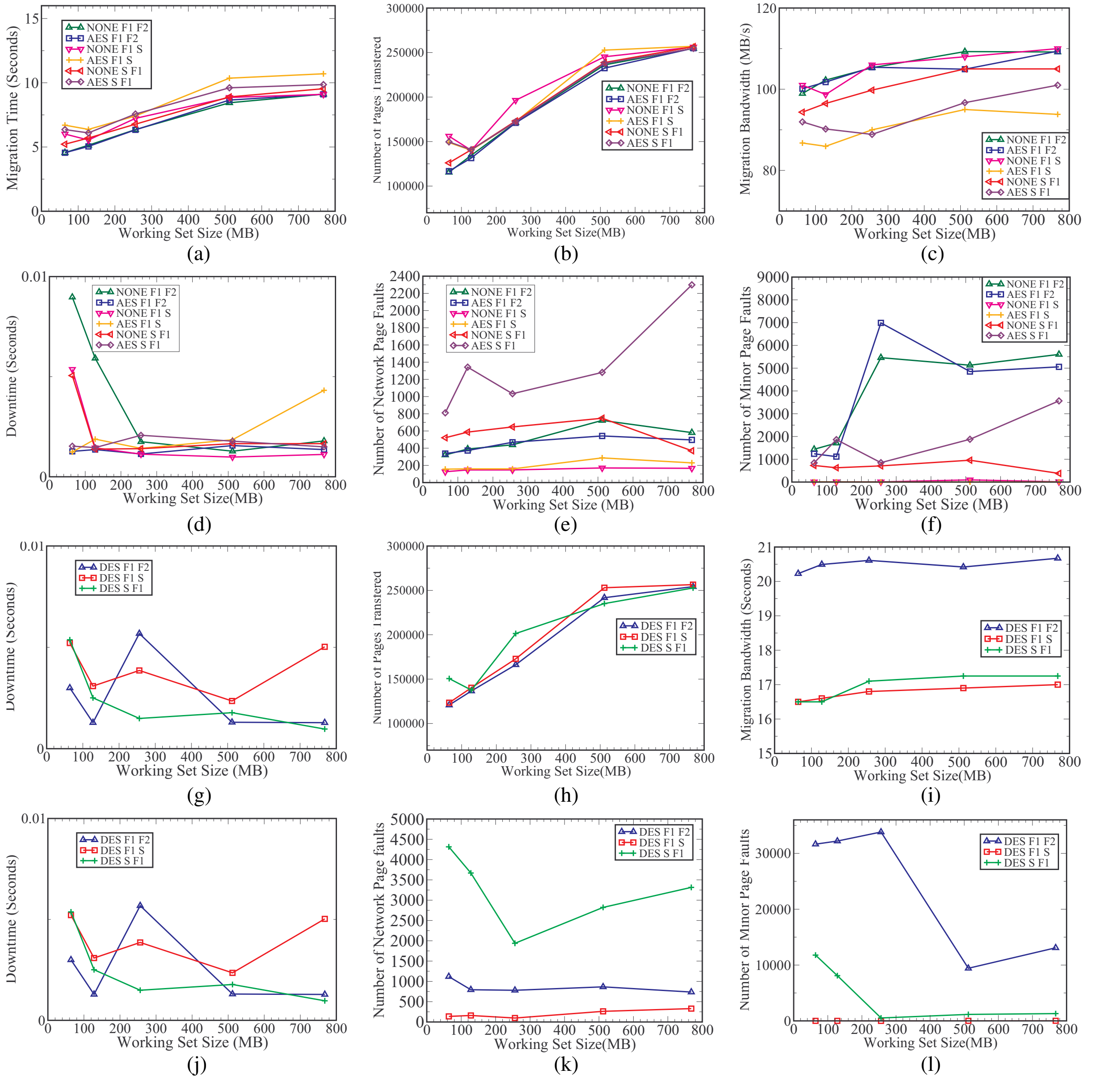
V. Network-intensive Workload

This section evaluates the impact of encryption when migrating a VM running a network-intensive application. We use the Netperf TCP benchmark which generates high-bandwidth traffic and measures the performance of different types of network. We migrate the VM running a Netperf server to which a Netperf client sends/receives packets from an external machine (neither source nor destination).

Figure 4 presents the total migration time, the downtime, the total number of pages transferred, and the migration bandwidth when migrating the VM running Netperf that receives packets from an external client using pre-copy and post-copy. The results for the case of Netperf sending packets to the external client are similar. Netperf does not perform high rate of memory write operation and hence the total number of pages transmitted with pre-copy is significantly less than that for the memory-write-intensive application. In addition, with pre-copy and AES encryption, the migration bandwidth is 5%–10% lower than that without encryption. This is because, Netperf is a network-intensive application and running Netperf on the source machine slows down AES encryption on the source machine. With post-copy, AES does not have an observable impact on migration bandwidth. 3DES significantly slows down the migration in both pre-copy and post-copy due to its slow encryption speed.

Figure 4(e) shows the number of network and minor page faults on the destination when migrating the VM using KVM post-copy. Similar to the write-intensive case, there are more network page faults when migrating the VM from S to F_1 than from F_1 to F_2 or from F_1 to S.

Impact of AES and 3DES on network bandwidth: We also measured the impact of encrypted migration on the network bandwidth computed by a Netperf server running inside the VM in both pre-copy and post-copy. Netperf reports the number of bytes received every 0.01 seconds most time, but sometimes report the number of bytes received in a longer



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 2. Post-copy migration of a VM running a memory-write-intensive application: (a) total migration time (AES, NONE), (b) total number of pages transferred (AES, NONE), (c) bandwidth (AES, NONE), (d) downtime (AES, NONE), (e) network page faults (AES, NONE), (f) minor page faults (AES, NONE), (g) total migration time (3DES), (h) total number of pages transferred (3DES), (i) bandwidth (3DES), (j) downtime (3DES), (k) network page faults (3DES), (l) minor page faults (3DES).

duration (ranging from 0.02 seconds to few seconds) when it does not receive enough bytes every 0.01 seconds.

Bandwidth computed by Netperf server receiving data from the external client: With pre-copy, the VM is running on the source machine, and the outgoing migration traffic does not interfere with the incoming data received by Netperf. As a result, when migrating the VM from F_1 to F_2 and from F_1 to S, there is no observable degradation of Netperf bandwidth with and without encryption, except during downtime when network bandwidth becomes zero. When migrating the VM from S to F_1 using pre-copy, there is no degradation of the network bandwidth when encryption is not performed; AES and 3DES reduce the bandwidth from 900-1000MB/second to 300-600MB/second. This is because AES and 3DES slow down the execution of Netperf on the slow machine S.

Figure 5 gives a visual representation of the reduction in bandwidth of Netperf in post-copy. Our experimental results show that the network bandwidth is reduced to 600-800MB/second during migration when no encryption is performed. AES and 3DES significantly reduce the network

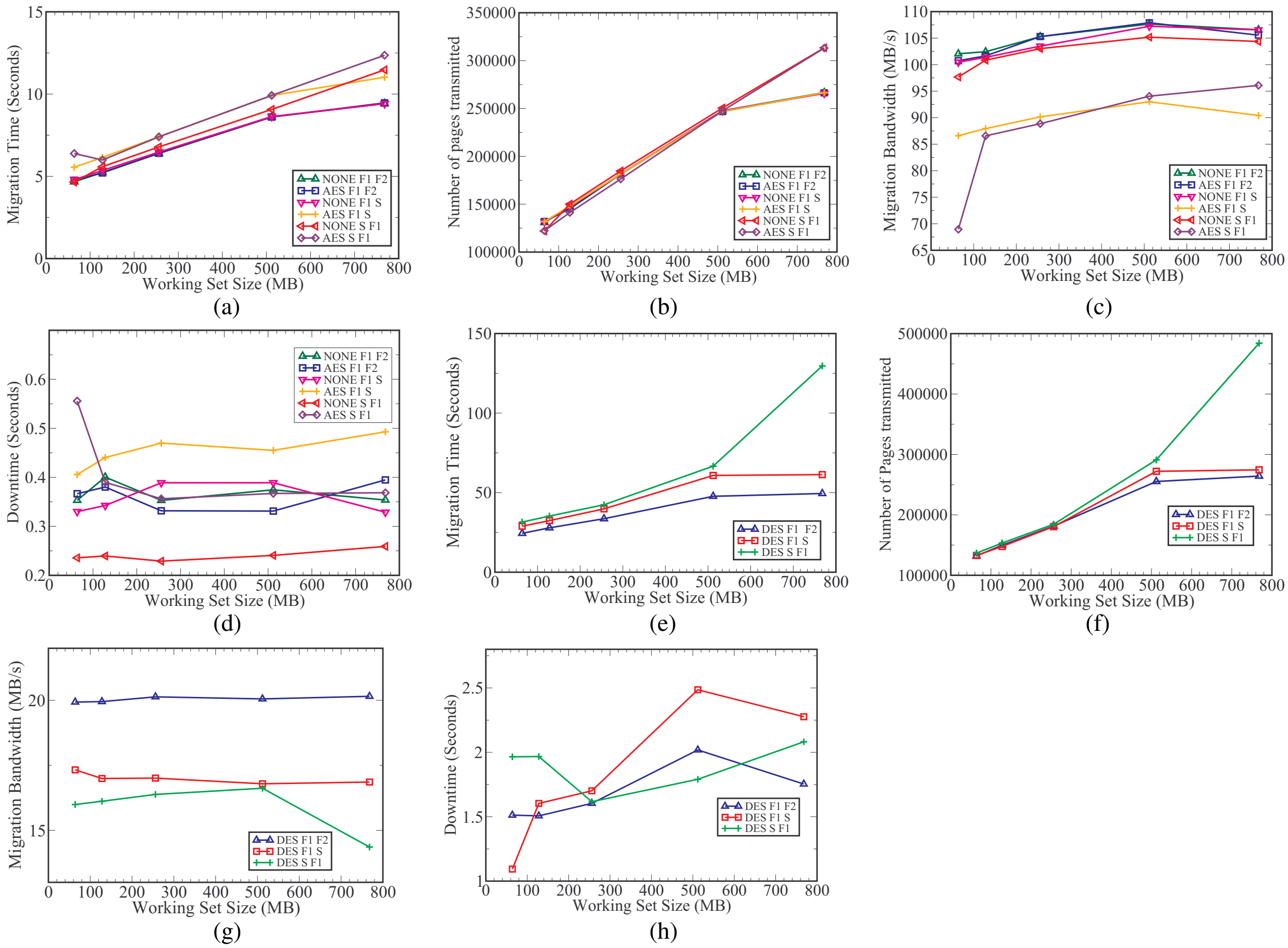


Fig. 3. Pre-copy migration of a VM running a memory-read-intensive application: (a) total migration time (NONE, AES), (b) total number of pages transferred (NONE, AES), (c) bandwidth (NONE, AES), and (d) downtime (NONE, AES), (e) total migration time (3DES), (f) total number of pages transferred (3DES), (g) bandwidth (3DES), and (h) downtime (3DES).

bandwidth when the migration starts. For example, when migrating the VM from F_1 to F_2 using AES, Netperf receives less than 200MB bytes in the first 5 seconds. This is because, (1) with post-copy, the incoming migration traffic interferes with the incoming Netperf traffic, which affects the network bandwidth computed by Netperf, and (2) the performance of the VM at the target right after resumption is significantly slow until the working set of the VM is received from the source.

Netperf client sending data to an external server: With pre-copy, the outgoing migration traffic interferes with the outgoing Netperf traffic. As a result, the network bandwidth is reduced to 400-800MB/second during migration with and without encryption. With post-copy, the page requests sent from the destination to the source interferes with the outgoing Netperf traffic. The network bandwidth is reduced to 500-900MB/second most time with and without encryption when migrating the VM from F_1 to F_2. When migrating the VM from F_1 to S and from S to F_1, the network bandwidth is reduced to 300-900MB/second without encryption and with AES; with 3DES, netperf sends less than 100MB bytes in the first 15 seconds because 3DES is slow which slows down the speed of transferring pages to the target and the performance of the VM at the target is slow until the working set is received.

VI. CPU-intensive Workload

The total migration time, downtime, and the number of pages transmitted when migrating a VM running Kernbench – a Kernel compilation benchmark– are similar to those when migrating a VM running Netperf. With both pre-copy and post-copy, AES does not impose observable overhead when migrating the VM from F_1 to F_2. When migrating the VM from S to F, AES reduces the migration bandwidth by 11.87% for pre-copy and 12.49% for post-copy.

We have also measured how migration and encryption affect the kernel compilation time. Without migration, it takes 69.96 seconds and 127.39 seconds to compile the kernel on F_1 and on S, respectively. Our experimental results show that, when migrating a VM from F_1 to F_2, post-copy does not increase kernel compilation time, while pre-copy without encryption and with AES slightly increases kernel compilation time (less than 1 second). Pre-copy with 3DES increases the

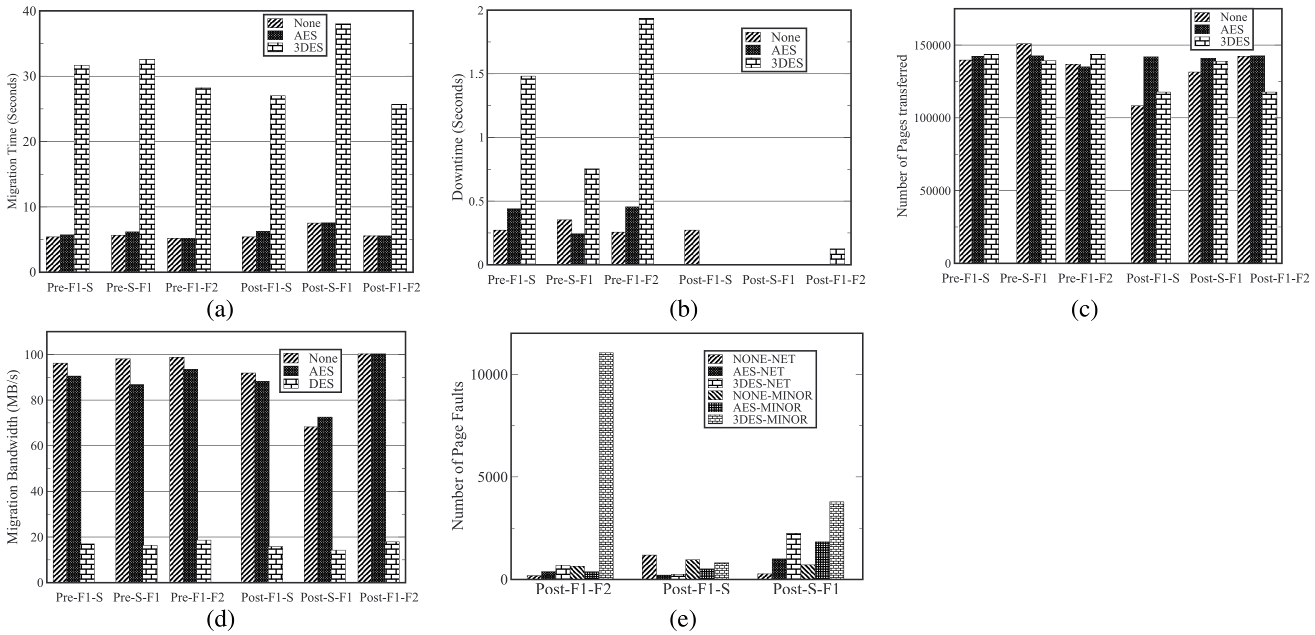


Fig. 4. Live migration of a VM running network-intensive Netperf benchmark: (a) Total migration time, (b) Downtime, (c) The number of pages transferred, (d) Bandwidth, and (e) Number of page faults.

compilation time by 7.7% when migrating the VM from F_1 to F_2. When migrating the VM from S to F_1 and from F_1 to S, the kernel compilation time depends on when the migration starts and downtime, which makes it hard to measure the performance impact of migration and encryption. For example, when migrating a VM from S to F, the earlier the pre-copy migration starts, the better the compilation time.

VII. Related Work

To secure live VM migration, VMware recommends [6] using a separate network (or a VLAN) dedicated for migration. A major drawback of this approach is the growth in complexity and administrative costs as the number of VMs grows [7]. Oracle VM [8], HP VM [9], and Proxmox VE [10] use encryption or SSL to protect migration traffic. Danev et al. [11] proposed a vTMP (Virtual Trusted Platform Module) key hierarchy that introduced an intermediate layer of keys between the TPM and vTPM, and then based on this key hierarchy, proposed a VM-vTMP secure migration protocol. Wang et al. [20] proposed to leverage Intel vPro and TPM to improve the security of live migration. They also proposed a role-based access control mechanism for VM migration and used remote attestation to perform platform measurement before migration. Anala et al. [21] discussed the attack model on the virtualization system and proposed to apply role-based access control, network intrusion detection techniques, firewall, and encryption for secure live migration. [22] proposed to detect live migrations inside a compromised VM, delay the live migration procedure, and use the delayed time to secure live VM migration. Other researchers have also considered security issues in virtual machines, including checkpointing [23], [24], [25], [26] and side channel attacks [27]. However, none of the above works studied the impact of encryption on the performance of VM migration.

VIII. Conclusion

In this paper, we analyzed the performance of pre-copy and post-copy live VM migration techniques when the VM's memory contents are encrypted during migration using two different encryption algorithms – AES and 3DES. Our experimental evaluations over different network and application workloads have resulted in the following findings. First, the type of encryption algorithm used can have a significant impact on the total migration time. Secondly, the affect of an encryption algorithm on VM downtime varies with the type of the migration technique. Thirdly, when a VM is migrated between a fast and a slow machine, the overhead of encryption differs depending upon the direction of the migration. Finally, the performance impact of encryption during migration on a VM's applications varies with the type of the application and the migration mechanism.

Acknowledgment This work is supported in part by the National Science Foundation through grant CNS-1320689.

References

[1] VMware Inc, http://www.vmware.com/.

[2] Microsoft Corp, "Hyper-v server 2008 r2," http://www.microsoft.com/hyper-v-server/en/us/overview.aspx.

[3] "Kernel based virtual machine," http://www.linux-kvm.org/.

[4] Xen Hypervisor, http://http://www.xen.org/.

[5] Oracle Corp, "Virtualbox," www.VirtualBox.org.

[6] J. Oberheide, E. Cooke, and F. Jahanian, "Exploiting live virtual machine migration," in *Black Hat*, 2008.

[7] Juniper Networks Inc., "Alternatives for securing virtual networks: A different network requires a different approach: Extending security to the virtual world, white paper 1000220-012-en," 2011.

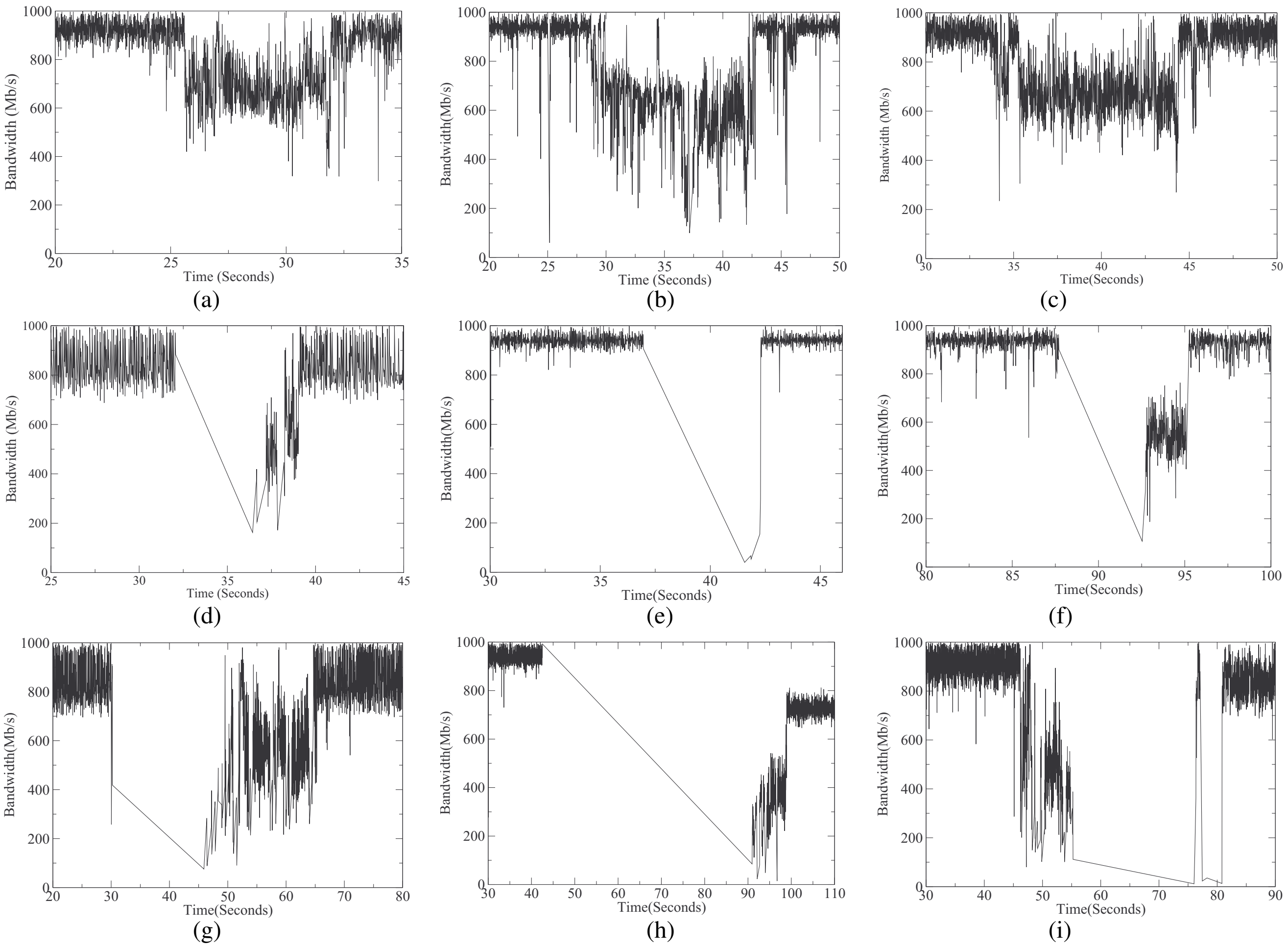


Fig. 5. The bandwidth reported by Netperf when migrating the VM using post-copy from (a) F_1 to F_2 (without encryption), (b) F_1 to S (without encryption), (c) S to F_1 (without encryption), (d) F_1 to F_2 (AES), (e) F_1 to S (AES), (f) S to F_1 (AES), (g) F_1 to F_2 (3DES), (h) F_1 to S (3DES), (i) S to F_1 (3DES).

[8] Oracle, "Oracle VM," http://www.oracle.com/us/026951.pdf?ssSourceSiteId=.

[9] HP VM, http://h18000.www1.hp.com/products/quickspecs/13375_div/13375_div.PDF.

[10] Proxmox, "Proxmox VE," pve.proxmox.com/.

[11] B. Danev, R. J. Masti, G. O. Karame, and S. Capkun, "Enabling secure vm-vtpm migration in private clouds," in *Proceedings of the 27th Annual Computer Security Applications Conference*, 2011, pp. 187–196.

[12] M. Nelson, B. Lim, and G. Hutchins, "Fast transparent migration for virtual machines," in *Proceedings of the annual conference on USENIX Annual Technical Conference*, 2005, pp. 25–25.

[13] C. Clark, K. Fraser, S. Hand, J. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Symposium on Networked Systems Design & Implementation*, 2005, pp. 273–286.

[14] M. R. Hines, U. Deshpande, and K. Gopalan, "Post-copy live migration of virtual machines," *SIGOPS Operating System Review*, vol. 43, no. 3, pp. 14–26, 2009.

[15] Federal Information Processing Standards Publication 197, "Announcing the advanced encryption standard (AES)," 2001.

[16] W. C. Barker and E. Barker, "Recommendation for the triple data encryption algorithm block cipher," in *National Institute of Standard and Technology Special Publication 800-67*, 2012.

[17] Takahiro Hirofuchi, Isaku Yamahata, "Postcopy live migration for qemu/kvm," http://grivon.apgrid.org/quick-kvm-migration.

[18] Netperf, http://www.netperf.org/netperf/.

[19] Linux Foundation, "The linux kernel archives," https://www.kernel.org.

[20] W. Wang, X. Wu, B. Lin, K. Miao, and X. Dang, "Secured VM live migration in personal cloud," in *poster, ACM Conference on Computer and Communications Security (CCS)*, 2010.

[21] M. R. Anala, J. Shetty, and G. Shobha, "A framework for secure live migration of virtual machines," in *International Conference on Advances in Computing, Communications and Informatics*, 2013.

[22] S. Biedermann, M. Zittel, and S. Katzenbeisser, "Improving security of virtual machines during live migrations," in *Eleventh Annual Conference on Privacy, Security and Trust (PST)*, 2013.

[23] M. I. Gofman, R. Luo, P. Yang, and K. Gopalan, "SPARC: A security and privacy aware virtual machine checkpointing mechanism," in *ACM Workshop on Privacy in the Electronic Society*, 2011, pp. 115–124.

[24] T. Ristenpart and S. Yilek, "When good randomness goes bad: Virtual machine reset vulnerabilities and hedging deployed cryptography," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2010.

[25] Y. Hu, T. Li, P. Yang, and K. Gopalan, "An application-level approach for privacy-preserving virtual machine checkpointing," in *the 6th IEEE International Conference on Cloud Computing*, 2013, pp. 59–66.

[26] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: security challenges in virtual machine based computing environments," in *Hot Topics in Operating Systems*, 2005.

[27] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-vm side channels and their use to extract private keys," in *ACM conference on Computer and communications security*, 2012, pp. 305–316.