(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 June 2015 (11.06.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/084344 A1

(51) International Patent Classification:
G06F 21/50 (2013.01)

(21) International Application Number:
PCT/US2013/073140

(22) International Filing Date:
4 December 2013 (04.12.2013)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **EMPIRE TECHNOLGY DEVELOPMENT, LLC** [US/US]; 2711 Centerville Road, Suite 400, Wilmington, DE 19808 (US).

(72) Inventor: **FINE, Kevin, S.**; Promenade Des Pins 44, Yverdon-les-bains, 1400 (CH).

(74) Agent: **TURK, Carl, K.**; Turk IP Law, LLC, 2885 Sanford Ave. S.W. #23998, Grandville, MI 49418 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: DETECTION OF SIDE CHANNEL ATTACKS BETWEEN VIRTUAL MACHINES

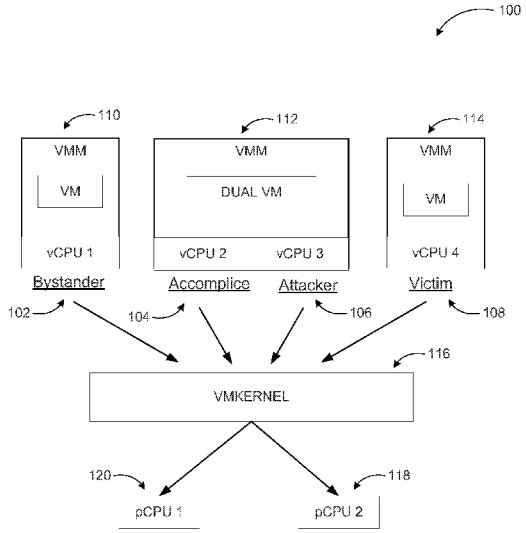


FIG. 1

(57) Abstract: Technologies are directed to a detector to identify a side channel attack between virtual machines. According to some examples, an inter-processor interrupt (IPI) rate of a first virtual machine (VM), a time stamp counter (TSC) rate of a second VM, and a cache miss ratio (CMR) of a third VM may be monitored. A side channel attack may then be detected based on the IPI rate, the TSC rate, and the CMR.

WO 2015/084344 A1

Published:

— *with international search report (Art. 21(3))*

DETECTION OF SIDE CHANNEL ATTACKS BETWEEN VIRTUAL MACHINES

BACKGROUND

[0001] Unless otherwise indicated herein, the materials described in this section are not prior art to the claims in this application and are not admitted to be prior art by inclusion in this section.

[0002] A cloud infrastructure provider may leverage economies of scale to provide dynamic and on-demand computing resources at lower cost. Virtualization is one of the tools that may be utilized by an infrastructure provider to increase efficiency and enhance performance. Virtual machines (VMs) from multiple customers may share physical resources such as servers. Shared resources may lead to multiple mutually distrusting customers simultaneously sharing physical resources from the same provider. Customers may be aware of the shared nature of cloud infrastructure. One obstacle preventing customers from migrating to cloud computing is the fear of spying from other users.

SUMMARY

[0003] The present disclosure generally describes techniques to operate a detector to identify side channel attacks between virtual machines.

[0004] According to some examples, a method is provided to detect a side channel attack between virtual machines. The method may include monitoring an inter-processor interrupt (IPI) rate of a first virtual machine (VM), monitoring a time stamp counter (TSC) rate of a second VM, monitoring a cache miss ratio (CMR) of a third VM, and detecting the side channel attack based on the IPI rate, the TSC rate, and the CMR.

[0005] According to other examples, a computing device may be provided to detect a side channel attack between virtual machines. The computing device may include a memory configured to store instructions, a controller coupled to the memory, where the controller executes a side channel attack detection module. The side channel attack detection module may be configured to monitor an inter-processor interrupt (IPI) rate of a first virtual machine (VM), detect the IPI rate to exceed a predetermined IPI rate threshold, label the first VM as at least one from a set of: an accomplice candidate and a bystander candidate, monitor a cache miss ratio (CMR) of a third VM, detect the CMR to exceed a predetermined CMR threshold, and label the third VM as a victim candidate.

[0006] According to other examples, a computer-readable medium may be provided to detect a side channel attack between virtual machines. The instructions may cause a method to be performed in response to execution, the method being similar to the methods described herein.

[0007] The foregoing summary is illustrative only and is not intended to be in any way limiting. In addition to the illustrative aspects, embodiments, and features described above, further aspects, embodiments, and features will become apparent by reference to the drawings and the following detailed description.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The foregoing and other features of this disclosure will become more fully apparent from the following description and appended claims, taken in conjunction with the accompanying drawings. Understanding that these drawings depict only several embodiments in accordance with the disclosure and are, therefore, not to be considered limiting of its scope, the disclosure will be described with additional specificity and detail through use of the accompanying drawings, in which:

FIG. 1 illustrates an example schema of virtual machines in an environment with side channel attacks;

FIG. 2 illustrates an example of a side channel attack between virtual machines;

FIG. 3 illustrates another example of a side channel attack between virtual machines;

FIG. 4 illustrates an example flow diagram of operating a detector to identify a side channel attack between virtual machines;

FIG. 5 illustrates an example of a detector used to detect a side channel attack between virtual machines;

FIG. 6 illustrates a general purpose computing device, which may be used to operate a detector to identify a side channel attack between virtual machines;

FIG. 7 is a flow diagram illustrating an example method to operation a detector to identify a side channel attack between virtual machines; and

FIG. 8 illustrates a block diagram of an example computer program product to operate a detector to identify a side channel attack between virtual machines,

all arranged in accordance with at least some embodiments described herein.

DETAILED DESCRIPTION

[0009] In the following detailed description, reference is made to the accompanying drawings, which form a part hereof. In the drawings, similar symbols typically identify similar components, unless context dictates otherwise. The illustrative embodiments described in the detailed description, drawings, and claims are not meant to be limiting. Other embodiments may be utilized, and other changes may be made, without departing from the spirit or scope of the subject matter presented herein. The aspects of the present disclosure, as generally described herein, and illustrated in the Figures, can be arranged, substituted, combined, separated, and designed in a wide variety of different configurations, all of which are explicitly contemplated herein.

[0010] This disclosure is generally drawn, *inter alia*, to methods, apparatus, systems, devices, and/or computer program products related to operation of a detector to identify a side channel attack between virtual machines.

[0011] Briefly stated, technologies are directed to a detector to identify a side channel attack between virtual machines. According to some examples, an inter-processor interrupt (IPI) rate of a first virtual machine (VM), a time stamp counter (TSC) rate of a second VM, and a cache miss ratio (CMR) of a third VM may be monitored. A side channel attack may then be detected based on the IPI rate, the TSC rate, and the CMR.

[0012] FIG. 1 illustrates an example schema of virtual machines in an environment with side channel attacks, arranged in accordance with at least some embodiments described herein.

[0013] As shown in a diagram 100, a virtual machine kernel (vmKernel) 116 may manage multiple virtual machine managers (VMM)s 110, 112, and 114. A virtual machine may be a self-contained execution environment for applications and services. A virtual machine may execute high-level applications such as operating systems. A VMM may manage and monitor a VM and provide resources to allow the VM to execute. The VMM 110 may provide a virtual central processing unit (vCPU) 1 to execute operations associated with a VM. A vCPU may be a processor placeholder such as a thread. Each VM may be assigned a vCPU, which in turn may be assigned to a processor to execute operations associated with a VM. Alternatively, a VM and a vCPU may be synonymous terms used to describe a self-contained execution environment.

[0014] In an alternative example, the VMM 112 may manage dual VMs. The dual VMs may be assigned to a vCPU 2 and a vCPU 3. The VMM 114 may manage another VM assigned to a vCPU 4. The vmKernel 116 may manage execution of instructions from the

vCPU 1, the vCPU 2, the vCPU 3, and the vCPU 4 by allocating physical resources such as a physical CPU (pCPU) 1 (120) and a pCPU 2 (118) to execute the instructions.

[0015] An attacker 106 sending instructions through the vCPU 3 may take advantage of shared resources with a victim 108 to execute a side channel attack to capture instructions executed by the victim 108. The attacker 106 may use an accomplice 104 during the side channel attack. The accomplice 104 may issue IPIs to the attacker 106 to enable the attacker 106 to interrupt execution of instructions by the victim 108. A bystander 102 may or may not take part in the side channel attack. The bystander 102 may include the VMM 110 that may execute instructions before or after those instructions are executed by the victim 108 on one of the shared resources such as the pCPU 1 (120) or the pCPU 2 (118).

[0016] The attacker 106 may take advantage of commonly used code base of an application to execute a side channel attack. Commonly used code base may include open source code elements or libraries used to construct applications. The attacker 106 may determine a path followed by an application executing in a commonly used processor between the victim 108 and the attacker 106 such as the pCPU 1 (120) or the pCPU 2 (118). Knowledge of the path may be used to determine data input or output of the application (executed by the victim 108) such as secret keys.

[0017] The execution path may be determined by examining the instructions of the victim 108 loaded into an L1 instruction cache of a commonly used processor. The attacker 106 may use a prime probe attack protocol to examine the L1 instruction cache or an instruction cache (I-cache), for example. The attacker 106 using the prime probe attack protocol may completely load I-cache with dummy instructions (that is, imitation instructions). Next, the attacker 106 may allow the victim 108 to execute a few instructions using the I-cache. Followed by the attacker continuing to load the I-cache with dummy instructions. While continuing to load the I-cache, the attacker 106 may measure a time it takes to load each dummy instruction into the I-cache. The attacker 106 may measure the time it takes to load the I-cache using a read time stamp counter (rdtsc) instruction. The rdtsc instruction may read the time stamp counter (TSC). The TSC is a counter (for example, 64-bit) on a processor, which counts a total number of clock cycles since a reset. A variance in a returned value before and after execution of the rdtsc instruction may determine a time of instruction consumption. If the attacker 106 loads a previously evicted cache location, the time of instruction consumption may be a large value. The attacker 106 may be aware that the cache location was used for an executed instruction by the victim 108.

[0018] A direct relationship between the cache location and a system memory location may not be discernable. However, a cache is set-associative. That is, a location within the cache of instructions associated with an executed application may correspond only to a subset of locations within the system memory associated with the executed application. Therefore, monitoring used cache locations may allow the attacker 106 to determine the execution path of the victim 108.

[0019] In multiple VM environments, the attacker 106 may use an observation granularity, an observation noise, and a vCPU migration to execute the side channel attack. The observation granularity may encapsulate an observation period by the attacker 106. In an example scenario, a vmKernel may allow the victim 108 to execute instructions during a predetermined period of time. In some examples, a number of executed instructions by the victim 108 may be a relatively large number, which may prevent analysis of the instructions by the attacker 106. For example, if the process runs uninterrupted for 30 milliseconds while executing 3 billion instructions per second a total of approximately 90 million instructions may be executed. This may be much larger than the instruction cache implying that the instruction cache is overwritten multiple times, preventing analysis by the attacker. The attacker 106 may interrupt the execution of instructions by the victim 108 to lower the observation granularity. The attacker 106 may use IPIs to interrupt execution of instructions by the victim 108. The attacker 106 may also use a second attacker process such as the accomplice 104. An example of the second attacker process may include the accomplice 104 configured to execute an iteration that issues IPIs to the attacker 106. The iteration may prompt the vmKernel to instigate the attacker to manage the IPI and prevent execution by the victim 108. The victim 108's execution time may be shortened because the attacker 106 may interrupt an executed application by the victim 108 to manage the IPI. Shortened execution time of the victim 108 may allow the attacker 106 to trace the execution path of the victim 108's executed application.

[0020] The observation noise may encapsulate noise associated with manual analysis of the victim 108's instructions. In an example scenario, the attacker 106 may determine a measurement time of an execution by the victim 108 to include the observation noise that inhibits an analysis of the victim 108's instructions. The attacker 106 may use a machine learning algorithm that includes a support vector machine (SVM) based on a hidden Markov model (HMM) to overcome the observation noise, for example. The attacker 106 may infer executed operation with few errors using the machine learning algorithm.

[0021] The vCPU migration may encapsulate physical resource sharing. In an example scenario, the attacker may share the pCPU 1 (120) with the victim 108. Alternatively, the attacker 106 may not share the pCPU 1 (120). The attacker 106 may aggregate code fragment sequences associated with the victim 108 using the machine learning algorithm that includes the SVM and the HMM to determine time measurements associated with the victim 108.

[0022] FIG. 2 illustrates an example of a side channel attack between virtual machines, arranged in accordance with at least some embodiments described herein.

[0023] As shown in a diagram 200, a vmKernel may use a timeshare scheme to execute instructions by vCPUs 208 of VMMs on a pCPU 1 (204) and a pCPU 2 (206) through a time 202. A bystander may execute operations initially on the pCPU 1 (204). During the period of time in which the bystander executes instructions on the pCPU 1 (204), an attacker and a victim may execute instructions on the pCPU 2 (206). The attacker may be unable to interrupt the victim's execution of instructions during a period of time in which the bystander executes instructions on the pCPU 1 (204). Next, an accomplice may allow the attacker to monitor the executed instructions of the victim by interrupting an execution by the victim. Interrupted execution periods on the pCPU 2 (206) may allow the attacker to monitor an execution path of instructions by the victim through monitoring of instructions of cache of pCPU 2 (206) shared with the victim.

[0024] The accomplice may aid the attacker by transmitting an inter-processor interrupt 210 to allow the attacker to execute an instruction on the pCPU 2 (206) and interrupt the execution of instructions by the victim. During a period such as a timing attack 212, the accomplice may send multiple IPIs to the attacker to interrupt execution of instructions by the victim on the pCPU 2 (206). The attacker may be allowed to perform a prime probe attack to monitor the instructions executed by the victim in an iterative process. The iterative process may continue until the accomplice is replaced by the bystander at the pCPU 1 (204). During the timing attack 212 period, the attacker may record a timing data for analysis of the cache of pCPU 2 (206) shared with the victim.

[0025] FIG. 3 illustrates another example of a side channel attack between virtual machines, arranged in accordance with at least some embodiments described herein.

[0026] A diagram 300 shows components of a side channel attack. Three vCPUs may be involved in the side channel attack in a datacenter hosting VMs. The vCPUs may execute instructions on a pCPU 1 (302) and a pCPU 2 (308). An attacker 1 (304) may issue IPIs to the pCPU 2 (308). The attacker 1 (304) may be an accomplice. An attacker 2 (306) may

execute rdtsc instructions to the pCPU 2 (308). Next, the attacker 2 (306) may be allowed to change places with a victim 310 to execute instructions on the pCPU 2 (308). The attacker 2 (306) may monitor cache of pCPU 2 (308) to determine executed instructions by the victim 310 stored in the cache during a period of interruption as a result of IPIs issued by the attacker 1 (304). The victim 310 may have L1 cache or I-cache misses because of the IPIs interrupting execution of instructions by the victim 310.

[0027] Each action, described above, may occur separately during normal operations at a datacenter. However, the actions occurring together may be an indicator of the side channel attack. Prevention of the side channel attack may be accomplished by a detector which correlates commonly used metrics between multiple vCPUs to spot behavior that signals a cross VM timing attack.

[0028] FIG. 4 illustrates an example flow diagram of operating a detector to identify a side channel attack between virtual machines, arranged in accordance with at least some embodiments described herein.

[0029] As shown in a diagram 400, detection of a side channel attack may involve a number of operations. A detector may analyze metrics from monitors monitoring vCPUs executing instructions on pCPUs through a vmKernel at operation 402 (ANALYZE METRICS FROM MONITORS). The metrics may include an IPI rate, a TSC rate, and a CMR. The IPI rate of a vCPU_m (that is, a first VM) may be compared to a predetermined IPI rate threshold at operation 404 (IS IPI RATE OF Vcpu_m GREATER THEN IPI RATE THRESHOLD?). The IPI rate threshold may be determined by a dynamic system setting that is adjustable based on a number of parameters associated with the vmKernel, a pCPU, a vCPU, a VM, or a VMM. Alternatively, the IPI rate threshold may be set manually by an external or an internal source such as the datacenter. In response to determining the IPI rate of the vCPU_m to be higher than the IPI rate threshold, the vCPU_m may be labeled an accomplice at operation 406 (LABEL vCPU_m ACCOMPLICE). The vCPU_m may be part of an attacker pair.

[0030] Next, a vCPU_n (that is, a second VM) receiving the IPIs from the vCPU_m may be identified as an attacker at operation 408 (IDENTIFY vCPU_n ATTACKER RECEIVING IPIs). The vCPU_n may be interrupted by the vCPU_m. The vCPU_n may be the other part of the attacker pair. The interrupted vCPU_n may be allowed to execute on a pCPU used by a co-executor vCPU_k (that is, a third VM).

[0031] Execution of instructions such as an RDTSC by the vCPU_n on the pCPU may interrupt execution of instructions by the co-executor vCPU_k. The co-executor vCPU_k

may be identified as a victim at operation 410 (IDENTIFY CO-EXECUTOR vCPU_k VICTIM OF vCPU_n ATTACKER).

[0032] A CMR of the vCPU_k may be compared to a CMR threshold at operation 412 (IS CMR OF vCPU_k GREATER THAN CMR THRESHOLD?). The CMR may be a number of cache misses divided by a total cache access associated with a vCPU (or a VM). Caches may be designed to keep the CMR close to zero. A prime probe attack may force the vCPU_k to have the CMR near a value of one. A vCPU with a high CMR near to the value of one may further be examined to determine the side channel attack. The CMR threshold may be determined by a dynamic system setting adjustable based on number of parameters associated with the vmKernel, the pCPU, the vCPU, the VM, or the VMM. Alternatively, the CMR threshold may be set manually by an external or an internal source such as the datacenter. In response to determining the CMR higher than the CMR threshold, a suspicion value or an S_mnk associated with the side channel attack based on the three vCPUs may be computed.

[0033] The S_mnk may be computed based on the IPI rate of the vCPU_m, a TSC rate of the vCPU_n, and the CMR of the vCPU_k. The computed the S_mnk may be compared to a suspicion value threshold at operation 414 (IS S_mnk GREATER THAN S THRESHOLD?). The suspicion value threshold may be determined by a dynamic system setting adjustable based on a number of parameters associated with the vmKernel, the pCPU, the vCPU, the VM, or the VMM. Alternatively, the suspicion value threshold may be set manually by an external or an internal source such as the datacenter. In response to determining the suspicion value S_mnk higher than the suspicion value threshold, a system, such as a datacenter management, may be notified of the side channel attack associated with the vCPU_m, the vCPU_n, and the vCPU_k at operation 416 (NOTIFY SYSTEM).

[0034] In some examples, the S_mnk may be computed using a formula of S_mnk =α * (IPI rate of vCPU_m / IPI rate threshold) + β * (TSC rate of vCPU_n / TSC rate threshold) + γ *(CMR of vCPU_k / CMR threshold). The formula may heuristically increase the suspicion value of the S_mnk in response to an increase in a suspicious behavior. The factors α, β, and γ may be included to allow datacenters to configure or adjust the formula to compute the S_mnk to a real world behavior. A default value for each of α, β, and γ may be 1.0, for example.

[0035] In response to determining the S_mnk value higher than the suspicion threshold, the vCPU_m and the vCPU_n may be identified as attackers (or the attacker pair). Next, an event identifying the vCPUs as the attackers may be recorded in a data store. The event may

be generated by clustering the IPI rate, the TSC, the CMR and data associated with the vCPU_m, the vCPU_n, and the vCPU_k into the event. To prevent the attack, the vCPU_n may be forced to execute alone on the pCPU. In addition to α, β, and γ, the IPI rate threshold, the TSC rate threshold, the CMR threshold, and the suspicion threshold may be configured by the datacenter in some examples.

[0036] According to some embodiments, CMR of an L1 data cache may also be monitored instead of CMR of an L1 instruction cache (L1i). Side channel attacks on the L1 data cache may be similarly likely as on the L1i. The side channel attack on an L2 and an L3 cache may be unlikely because of increased size of the L2 and the L3 cache, mixed instructions in the L2 and L3 cache, and increased distance of the L2 and the L3 cache from the pCPU.

[0037] Other interrupts may also be monitored because the accomplice VM may use alternate interrupts including input/output (I/O) interrupts to interrupt the attacker VM and force the victim VM to stop executing instructions. A detector monitoring and analyzing interrupts to detect the side channel attack may monitor IPIs as well as other interrupts.

[0038] According to other embodiments, the attacker may use an instruction other than rdtsc to time events. In an example scenario where the attacker may use a different instruction to time events, the detector may monitor CMR associated with the victim and the IPI rate associated with the accomplice to determine the side channel attack. The CMR above the CMR threshold and the IPI rate above the IPI rate threshold may indicate the side channel attack.

[0039] FIG. 5 illustrates an example of a detector used to detect a side channel attack between virtual machines, arranged in accordance with at least some embodiments described herein.

[0040] As shown in a diagram 500, a bystander vCPU 1 may be managed by a VMM 502, an accomplice vCPU 2 and an attacker vCPU 3 may be managed by a VMM 504, and a victim vCPU 4 may be managed by a VMM 506. A vmKernel 508 may allocate resources to allow vCPUs 1, 2, 3, and 4 to execute instructions. The vmKernel 508 may manage the resources pCPU 1 (512) and pCPU 2 (514).

[0041] The vmKernel 508 may host a detector 510 analyzing metrics transmitted from monitors in the VMMs 502, 504, and 506. The monitors may monitor metrics including an IPI rate, a TSC rate, and a CMR associated with the vCPU 1, the vCPU 2, the vCPU 3, and the vCPU 4. The detector may analyze the IPI rate, the TSC rate, and the CMR to determine a suspicion value associated with the vCPU 2, the vCPU 3, and the vCPU 4. The suspicion

value may be compared to a suspicion value threshold to identify the vCPU 2 and the vCPU 3 as an attacker pair.

[0042] FIG. 6 illustrates a general purpose computing device, which may be used to operate a detector to identify a side channel attack between virtual machines, arranged in accordance with at least some embodiments described herein. The computing device 600 of the FIG. 6 may be one or more of the vmKernel 508 executed by a datacenter, or some other device that is not shown in FIG. 1, 2, 3, 4, and 5. In a configuration 602, computing device 600 typically includes one or more processors 604 and a system memory 606. A memory bus 608 may be used for communicating between processor 604 and system memory 606.

[0043] Depending on a particular configuration, processor 604 may be of any type including but not limited to a microprocessor (μP), a microcontroller (μC), a Digital Signal Processor (DSP), or any combination thereof. Processor 604 may include one more levels of caching, such as a level cache memory 612, a processor core 614, and registers 616. Example processor core 614 may include an Arithmetic Logic Unit (ALU), a floating point unit (FPU), a Digital Signal Processing core (DSP Core), or any combination thereof. An example memory controller 618 may also be used with processor 604, or in some implementations, memory controller 614 may be an internal part of processor 604.

[0044] Depending on the particular configuration, system memory 606 may be of any type including but not limited to volatile memory (such as RAM), non-volatile memory (such as ROM, flash memory, etc.) or any combination thereof. System memory 606 may include an operating system 620, a detector application 622, and program data 624. The detector application 622 may include a side channel module 626 that is arranged to detect a side a channel attack between VMs (or vCPUs). Program data 624 may include one or more of metrics data 628 and similar data as discussed above in conjunction with at least FIG. 1, 2, 3, 4, and 5. This data may be useful for detecting the side channel attack between VMs as described herein. This described basic configuration 602 is illustrated in FIG. 6 by those components within the inner dashed line.

[0045] Computing device 600 may have additional features or functionality, and additional interfaces to facilitate communications between basic configuration 602 and any required devices and interfaces. For example, a bus/interface controller 630 may be used to facilitate communications between basic configuration 602 and one or more data storage devices 632 via a storage interface bus 634. Data storage devices 632 may be removable storage devices 636, non-removable storage devices 638, or a combination thereof. Examples of removable storage and non-removable storage devices include magnetic disk

devices such as flexible disk drives and Hard-Disk Drives (HDDs), optical disk drives such as Compact Disk (CD) drives or Digital Versatile Disk (DVD) drives, Solid State Drives (SSDs), and tape drives to name a few. Example computer storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data.

[0046] System memory 606, removable storage devices 636 and non-removable storage devices 638 are examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVDs) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which may be used to store information and which may be accessed by computing device 600. Any such computer storage media may be part of computing device 600.

[0047] Computing device 600 may also include an interface bus 640 for facilitating communication from various interface devices (for example, output devices 642, peripheral interfaces 644, and communication devices 666 to basic configuration 602 via bus/interface controller 630. Example output devices 642 include a graphics processing unit 648 and an audio processing unit 650, which may be configured to communicate to various external devices such as a display or speakers via one or more A/V ports 652. Example peripheral interfaces 644 include a serial interface controller 654 or a parallel interface controller 656, which may be configured to communicate with external devices such as input devices (for example, keyboard, mouse, pen, voice input device, touch input device, etc.) or other peripheral devices (for example, printer, scanner, etc.) via one or more I/O ports 658. An example communication device 666 includes a network controller 660, which may be arranged to facilitate communications with one or more other computing devices 662 over a network communication link via one or more communication ports 664.

[0048] The network communication link may be one example of a communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and may include any information delivery media. A "modulated data signal" may be a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media may include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF),

microwave, infrared (IR) and other wireless media. The term computer readable media as used herein may include both storage media and communication media.

[0049] Computing device 600 may be implemented as a portion of a physical server, virtual server, a computing cloud, or a hybrid device that include any of the above functions. Computing device 600 may also be implemented as a personal computer including both laptop computer and non-laptop computer configurations. Moreover, computing device 600 may be implemented as a networked system or as part of a general purpose or specialized server.

[0050] Networks for a networked system including computing device 600 may comprise any topology of servers, clients, switches, routers, modems, Internet service providers, and any appropriate communication media (for example, wired or wireless communications). A system according to embodiments may have a static or dynamic network topology. The networks may include a secure network such as an enterprise network (for example, a LAN, WAN, or WLAN), an unsecure network such as a wireless open network (for example, IEEE 802.11 wireless networks), or a world-wide network such (for example, the Internet). The networks may also comprise a plurality of distinct networks that are adapted to operate together. Such networks are configured to provide communication between the nodes described herein. By way of example, and not limitation, these networks may include wireless media such as acoustic, RF, infrared and other wireless media. Furthermore, the networks may be portions of the same network or separate networks.

[0051] FIG. 7 is a flow diagram illustrating an example method to operation a detector to identify a side channel attack between virtual machines that may be performed by a computing device 710, such as the computing device 600 in FIG. 6, arranged in accordance with at least some embodiments described herein.

[0052] The computing device 710 may be embodied as computing device 600, or similar devices executing instructions stored in a non-transitory computer-readable medium 720 for performing the method. A process to operate a detector to identify a side channel attack between virtual machines may include one or more operations, functions or actions as is illustrated by one or more of blocks 722, 724, 726, and/or 728.

[0053] Some example processes may begin with an operation 722, "MONITOR AN INTER-PROCESSOR INTERRUPT (IPI) RATE OF A FIRST VIRTUAL MACHINE (VM)." At the operation 722, the computing device 600 may identify a VM having an IPI rate higher than a predetermined IPI rate threshold as the accomplice 104.

[0054] The operation 722 may be followed by an operation 724, "MONITOR A TIME STAMP COUNTER (TSC) RATE OF A SECOND VM." At the operation 724, the computing device 600 may identify a VM having a TSC rate higher than a predetermined TSC rate threshold as the attacker 106.

[0055] The operation 724 may be followed by an operation 726, "MONITOR A CACHE MISS RATIO (CMR) OF A THIRD VM." At the operation 726, the computing device 600 may identify a VM having a CMR higher than a predetermined CMR rate threshold as the victim 108.

[0056] FIG. 8 illustrates a block diagram of an example computer program product to operate a detector to identify a side channel attack between virtual machines, arranged in accordance with at least some embodiments described herein.

[0057] In some examples as shown in FIG. 8, computer program product 800 may include a signal bearing medium 802 that may also include machine readable instructions 804 that, in response to execution by, for example, a processor, may provide the functionality described above with respect to FIG 1 through FIG. 7. Thus, for example, referring to computing device 600, one or more of the tasks shown in FIG. 8 may be undertaken in response to instructions 804 conveyed to the computing device 600 by medium 802 to perform actions associated with a detector to identify side channel attacks between virtual machines. Some of those instructions may include monitoring an inter-processor interrupt (IPI) rate of a first virtual machine (VM), monitoring a time stamp counter (TSC) rate of a second VM, monitoring a cache miss ratio (CMR) of a third VM, and detecting the side channel attack based on the IPI rate, the TSC rate, and the CMR.

[0058] In some implementations, signal bearing medium 802 depicted in FIG. 8 may encompass a non-transitory computer-readable medium 806, such as, but not limited to, a hard disk drive, a Compact Disc (CD), a Digital Versatile Disk (DVD), a digital tape, memory, etc. In some implementations, signal bearing medium 802 may encompass a recordable medium 808, such as, but not limited to, memory, read/write (R/W) CDs, R/W DVDs, etc. In some implementations, signal bearing medium 802 may encompass a communications medium 810, such as, but not limited to, a digital and/or an analog communication medium (for example, a fiber optic cable, a waveguide, a wired communications link, a wireless communication link, etc.). Thus, for example, computer program product 800 may be conveyed to the processor 804 by an RF signal bearing medium 802, where the signal bearing medium 802 is conveyed by a wireless communications

medium 810 (for example, a wireless communications medium conforming with the IEEE 802.11 standard).

[0059] According to some examples, a method is provided to detect a side channel attack between virtual machines. The method may include monitoring an inter-processor interrupt (IPI) rate of a first virtual machine (VM), monitoring a time stamp counter (TSC) rate of a second VM, monitoring a cache miss ratio (CMR) of a third VM, and detecting the side channel attack based on the IPI rate, the TSC rate, and the CMR.

[0060] According to other examples, detecting the side channel attack may include detecting the IPI rate to exceed a predetermined IPI rate threshold, and labeling the first VM as at least one from a set of: an accomplice candidate and a bystander candidate. Detecting the side channel may also include detecting the TSC rate to exceed a predetermined TSC rate threshold, and labeling the second VM as an attacker candidate. Detecting the side channel may further include detecting the CMR to exceed a predetermined CMR threshold, and labeling the third VM as a victim candidate.

[0061] According to further examples, the method may further include detecting one or more IPI sent by the first VM and received by the second VM, and determining the second VM to be interrupted by the first VM. The CMR may be determined through a first number of cache misses divided by a second number of total cache accessed by the third VM. A suspicion value associated with the side channel attack may be computed by: determining a first value based on a quotient of the IPI rate and a predetermined IPI rate threshold, determining a second value based on a quotient of the TSC rate and a predetermined TSC rate threshold, determining a third value based on a quotient of the CMR and a predetermined CMR threshold, and adding the first value, the second value, and the third value to compute the suspicion value. The suspicion value may be detected to exceed a predetermined suspicion value threshold, and the first VM and the second VM may be identified as an attacker pair. One or more datacenter operator may be allowed to adjust one or more from a set of: a predetermined IPI rate threshold, a predetermined TSC threshold, a predetermined CMR threshold, and a predetermined suspicion value threshold.

[0062] According to some examples, the method may further include clustering the IPI rate, the TSC, the CMR and data associated with the first VM, second VM, and third VM into an event in response to determining the side channel attack, and recording the event in a data store. One or more datacenter operator may be notified in response to determining the side channel attack.

[0063] According to other examples, a computing device may be provided to detect a side channel attack between virtual machines. The computing device may include a memory configured to store instructions, a controller coupled to the memory, where the controller executes a side channel attack detection module. The side channel attack detection module may be configured to monitor an inter-processor interrupt (IPI) rate of a first virtual machine (VM), detect the IPI rate to exceed a predetermined IPI rate threshold, label the first VM as at least one from a set of: an accomplice candidate and a bystander candidate, monitor a cache miss ratio (CMR) of a third VM, detect the CMR to exceed a predetermined CMR threshold, and label the third VM as a victim candidate.

[0064] According to some examples, the side channel attack detection module may be further configured to monitor a time stamp counter (TSC) rate of a second VM, and detecting the side channel attack based on the IPI rate, the TSC rate, and the CMR. The TSC rate may be detected to exceed a predetermined TSC rate threshold, and the second VM may be labeled as an attacker candidate. One or more IPI sent by the first VM and received by the second VM may be detected, and the second VM may be determined as interrupted by the first VM. The CMR may be computed through a first number of cache misses divided by a second number of total cache accessed by the third VM.

[0065] According to further examples, the side channel attack detection module may be further configured to compute a suspicion value associated with the side channel attack through an operation to: determine a first value based on a quotient of the IPI rate and a predetermined IPI rate threshold, determine a second value based on a quotient of a TSC rate if a second VM and a predetermined TSC rate threshold, determine a third value based on a quotient of the CMR and a predetermined CMR threshold, and add the first value, the second value, and the third value to compute the suspicion value.

[0066] According to other examples, the side channel attack detection module may be further configured to compute the suspicion value for the first VM, the second VM, and the third VM, detect the suspicion value to exceed a predetermined suspicion value threshold, and identify the first VM and the second VM as an attacker pair. The IPI rate, the TSC rate, the CMR and data associated with the first VM, a second VM, and the third VM may be clustered into an event in response to determining the side channel attack, and the event may be recorded in a data store. One or more datacenter operators may be allowed to adjust one or more from a set of: the predetermined IPI rate threshold, the predetermined TSC rate threshold, the predetermined CMR threshold, and a predetermined suspicion value threshold.

One or more datacenter operators may be notified in response to determining the side channel attack.

[0067] According to other examples, a computer-readable medium may be provided to detect a side channel attack between virtual machines. The instructions may cause a method to be performed in response to execution, the method being similar to the methods described above.

[0068] There is little distinction left between hardware and software implementations of aspects of systems; the use of hardware or software is generally (but not always, in that in certain contexts the choice between hardware and software may become significant) a design choice representing cost vs. efficiency tradeoffs. There are various vehicles by which processes and/or systems and/or other technologies described herein may be effected (for example, hardware, software, and/or firmware), and that the preferred vehicle will vary with the context in which the processes and/or systems and/or other technologies are deployed. For example, if an implementer determines that speed and accuracy are paramount, the implementer may opt for a mainly hardware and/or firmware vehicle; if flexibility is paramount, the implementer may opt for a mainly software implementation; or, yet again alternatively, the implementer may opt for some combination of hardware, software, and/or firmware.

[0069] The foregoing detailed description has set forth various embodiments of the devices and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood by those within the art that each function and/or operation within such block diagrams, flowcharts, or examples may be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. In one embodiment, several portions of the subject matter described herein may be implemented via Application Specific Integrated Circuits (ASICs), Field Programmable Gate Arrays (FPGAs), digital signal processors (DSPs), or other integrated formats. However, those skilled in the art will recognize that some aspects of the embodiments disclosed herein, in whole or in part, may be equivalently implemented in integrated circuits, as one or more computer programs running on one or more computers (for example, as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (for example as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination

thereof, and that designing the circuitry and/or writing the code for the software and or firmware would be well within the skill of one of skill in the art in light of this disclosure.

[0070] The present disclosure is not to be limited in terms of the particular embodiments described in this application, which are intended as illustrations of various aspects. Many modifications and variations can be made without departing from its spirit and scope, as will be apparent to those skilled in the art. Functionally equivalent methods and apparatuses within the scope of the disclosure, in addition to those enumerated herein, will be apparent to those skilled in the art from the foregoing descriptions. Such modifications and variations are intended to fall within the scope of the appended claims. The present disclosure is to be limited only by the terms of the appended claims, along with the full scope of equivalents to which such claims are entitled. It is to be understood that this disclosure is not limited to particular methods, systems, or components, which can, of course, vary. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to be limiting.

[0071] In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as a program product in a variety of forms, and that an illustrative embodiment of the subject matter described herein applies regardless of the particular type of signal bearing medium used to actually carry out the distribution. Examples of a signal bearing medium include, but are not limited to, the following: a recordable type medium such as a floppy disk, a hard disk drive, a Compact Disc (CD), a Digital Versatile Disk (DVD), a digital tape, a computer memory, etc.; and a transmission type medium such as a digital and/or an analog communication medium (for example, a fiber optic cable, a waveguide, a wired communications link, a wireless communication link, etc.).

[0072] Those skilled in the art will recognize that it is common within the art to describe devices and/or processes in the fashion set forth herein, and thereafter use engineering practices to integrate such described devices and/or processes into data processing systems. That is, at least a portion of the devices and/or processes described herein may be integrated into a data processing system via a reasonable amount of experimentation. Those having skill in the art will recognize that a typical data processing system generally includes one or more of a system unit housing, a video display device, a memory such as volatile and non-volatile memory, processors such as microprocessors and digital signal processors, computational entities such as operating systems, drivers, graphical user

interfaces, and applications programs, one or more interaction devices, such as a touch pad or screen, and/or control systems including feedback loops.

[0073] A typical data processing system may be implemented utilizing any suitable commercially available components, such as those typically found in data computing/communication and/or network computing/communication systems. The herein described subject matter sometimes illustrates different components contained within, or connected with, different other components. It is to be understood that such depicted architectures are merely exemplary, and that in fact many other architectures may be implemented which achieve the same functionality. In a conceptual sense, any arrangement of components to achieve the same functionality is effectively "associated" such that particular functionality is achieved. Hence, any two components herein combined to achieve a particular functionality may be seen as "associated with" each other such that the particular functionality is achieved, irrespective of architectures or intermediate components. Likewise, any two components so associated may also be viewed as being "operably connected", or "operably coupled", to each other to achieve the particular functionality, and any two components capable of being so associated may also be viewed as being "operably couplable", to each other to achieve the particular functionality. Specific examples of operably couplable include but are not limited to physically connectable and/or physically interacting components and/or wirelessly interactable and/or wirelessly interacting components and/or logically interacting and/or logically interactable components.

[0074] With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

[0075] It will be understood by those within the art that, in general, terms used herein, and especially in the appended claims (for example, bodies of the appended claims) are generally intended as “open” terms (for example, the term “including” should be interpreted as “including but not limited to,” the term “having” should be interpreted as “having at least,” the term “includes” should be interpreted as “includes but is not limited to,” etc.). It will be further understood by those within the art that if a specific number of an introduced claim recitation is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases "at least one" and "one or more" to introduce claim recitations. However, the use of such phrases should not be

construed to imply that the introduction of a claim recitation by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim recitation to embodiments containing only one such recitation, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an" (for example, “a” and/or “an” should be interpreted to mean “at least one” or “one or more”); the same holds true for the use of definite articles used to introduce claim recitations. In addition, even if a specific number of an introduced claim recitation is explicitly recited, those skilled in the art will recognize that such recitation should be interpreted to mean at least the recited number (for example, the bare recitation of "two recitations," without other modifiers, means at least two recitations, or two or more recitations).

[0076] Furthermore, in those instances where a convention analogous to “at least one of A, B, and C, etc.” is used, in general such a construction is intended in the sense one having skill in the art would understand the convention (for example, “ a system having at least one of A, B, and C” would include but not be limited to systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together, etc.). It will be further understood by those within the art that virtually any disjunctive word and/or phrase presenting two or more alternative terms, whether in the description, claims, or drawings, should be understood to contemplate the possibilities of including one of the terms, either of the terms, or both terms. For example, the phrase “A or B” will be understood to include the possibilities of “A” or “B” or “A and B.”

[0077] As will be understood by one skilled in the art, for any and all purposes, such as in terms of providing a written description, all ranges disclosed herein also encompass any and all possible subranges and combinations of subranges thereof. Any listed range can be easily recognized as sufficiently describing and enabling the same range being broken down into at least equal halves, thirds, quarters, fifths, tenths, etc. As a non-limiting example, each range discussed herein can be readily broken down into a lower third, middle third and upper third, etc. As will also be understood by one skilled in the art all language such as “up to,” “at least,” “greater than,” “less than,” and the like include the number recited and refer to ranges which can be subsequently broken down into subranges as discussed above. Finally, as will be understood by one skilled in the art, a range includes each individual member. Thus, for example, a group having 1-3 cells refers to groups having 1, 2, or 3 cells. Similarly, a group having 1-5 cells refers to groups having 1, 2, 3, 4, or 5 cells, and so forth.

[0078] While various aspects and embodiments have been disclosed herein, other aspects and embodiments will be apparent to those skilled in the art. The various aspects and

embodiments disclosed herein are for purposes of illustration and are not intended to be limiting, with the true scope and spirit being indicated by the following claims.

CLAIMS

WHAT IS CLAIMED IS:

1. A method to detect a side channel attack between virtual machines, the method comprising:
monitoring an inter-processor interrupt (IPI) rate of a first virtual machine (VM);
monitoring a time stamp counter (TSC) rate of a second VM;
monitoring a cache miss ratio (CMR) of a third VM; and
detecting the side channel attack based on the IPI rate, the TSC rate, and the CMR.

2. The method of claim 1, wherein detecting the side channel attack comprises:
detecting the IPI rate to exceed a predetermined IPI rate threshold; and
labeling the first VM as at least one from a set of: an accomplice candidate and a bystander candidate.

3. The method of claim 1, wherein detecting the side channel attack comprises:
detecting the TSC rate to exceed a predetermined TSC rate threshold; and
labeling the second VM as an attacker candidate.

4. The method of claim 1, wherein detecting the side channel attack comprises:
detecting the CMR to exceed a predetermined CMR threshold; and
labeling the third VM as a victim candidate.

5. The method of claim 1, further comprising:
detecting at least one IPI sent by the first VM and received by the second VM; and
determining the second VM to be interrupted by the first VM.

6. The method of claim 1, further comprising:
determining the CMR through a first number of cache misses divided by a second number of total cache accessed by the third VM.

7. The method of claim 1, further comprising:
computing a suspicion value associated with the side channel attack by:
determining a first value based on a quotient of the IPI rate and a predetermined IPI rate threshold;

determining a second value based on a quotient of the TSC rate and a predetermined TSC rate threshold;

determining a third value based on a quotient of the CMR and a predetermined CMR threshold;

multiplying the first value, the second value, and the third value with a first coefficient, a second coefficient, and a third coefficient, respectively; and

adding products of the first value and the first coefficient, the second value and the second coefficient, and the third value and the third coefficient to compute the suspicion value.

8. The method of claim 7, further comprising:

detecting the suspicion value to exceed a predetermined suspicion value threshold; and

identifying the first VM and the second VM as an attacker pair.

9. The method of claim 7, further comprising:

allowing a datacenter operator to adjust at least one from a set of: a predetermined IPI rate threshold, a predetermined TSC threshold, a predetermined CMR threshold, the first coefficient, the second coefficient, the third coefficient, and a predetermined suspicion value threshold.

10. The method of claim 1, further comprising:

clustering the IPI rate, the TSC, the CMR and data associated with the first VM, second VM, and third VM into an event in response to determining the side channel attack; and

recording the event in a data store.

11. The method of claim 1, further comprising:

notifying at least one datacenter operator in response to determining the side channel attack.

12. A computing device to detect a side channel attack between virtual machines, the computing device comprising:

a memory configured to store instructions;

a controller coupled to the memory, wherein the controller executes a side channel attack detection module, the side channel attack detection module configured to:

monitor an inter-processor interrupt (IPI) rate of a first virtual machine (VM);

detect the IPI rate to exceed a predetermined IPI rate threshold;

label the first VM as at least one from a set of: an accomplice candidate and a bystander candidate;

monitor a cache miss ratio (CMR) of a third VM;

detect the CMR to exceed a predetermined CMR threshold; and

label the third VM as a victim candidate.

13. The computing device according to claim 12, wherein the side channel attack detection module is further configured to:

monitor a time stamp counter (TSC) rate of a second VM; and

detecting the side channel attack based on the IPI rate, the TSC rate, and the CMR.

14. The computing device according to claim 13, wherein the side channel attack detection module is further configured to:

detect the TSC rate to exceed a predetermined TSC rate threshold; and

label the second VM as an attacker candidate.

15. The computing device according to claim 12, wherein the side channel attack detection module is further configured to:

detect at least one IPI sent by the first VM and received by the second VM; and

determine the second VM as interrupted by the first VM.

17. The computing device according to claim 12, wherein the side channel attack detection module is further configured to:

compute the CMR through a first number of cache misses divided by a second number of total cache accessed by the third VM.

18. The computing device according to claim 12, wherein the side channel attack detection module is further configured to:

compute a suspicion value associated with the side channel attack through an operation to:

determine a first value based on a quotient of the IPI rate and a predetermined IPI rate threshold;

determine a second value based on a quotient of a TSC rate if a second VM and a predetermined TSC rate threshold;

determine a third value based on a quotient of the CMR and a predetermined CMR threshold;

multiply the first value, the second value, and the third value with a first coefficient, a second coefficient, and a third coefficient, respectively; and

add products of the first value and the first coefficient, the second value and the second coefficient, and the third value and the third coefficient to compute the suspicion value.

19. The computing device according to claim 18, wherein the side channel attack detection module is further configured to:

compute the suspicion value for the first VM, the second VM, and the third VM;

detect the suspicion value to exceed a predetermined suspicion value threshold; and

identify the first VM and the second VM as an attacker pair.

20. The computing device according to claim 18, wherein the side channel module is further configured to:

cluster the IPI rate, the TSC rate, the CMR and data associated with the first VM, a second VM, and the third VM into an event in response to determining the side channel attack; and

record the event in a data store.

21. The computing device according to claim 18, wherein the side channel module is further configured to:

allow a datacenter operator to adjust at least one from a set of: the predetermined IPI rate threshold, the predetermined TSC rate threshold, the predetermined CMR threshold, the first coefficient, the second coefficient, the third coefficient, and a predetermined suspicion value threshold.

22. The computing device according to claim 12, wherein the side channel module is further configured to:

notify at least one datacenter operator in response to determining the side channel attack.

23. A computer-readable storage medium with instructions stored thereon to detect a side channel attack between virtual machines, the instructions in response to execution causing a method to be performed, wherein the method comprises:

monitoring an inter-processor interrupt (IPI) rate of a first virtual machine (VM);

detecting the IPI rate to exceed a predetermined IPI rate threshold;

labeling the first VM as at least one from a set of: an accomplice candidate and a bystander candidate;

monitoring a time stamp counter (TSC) rate of a second VM;

detecting the TSC rate to exceed a predetermined TSC rate threshold;

labeling the second VM as an attacker candidate;

monitoring a cache miss ratio (CMR) of a third VM; and

detecting the side channel attack based on the IPI rate, the TSC rate, and the CMR.

24. The computer-readable storage medium according to claim 23, wherein the third VM is labeled as a victim candidate in response to a detection that the CMR exceeds a predetermined CMR threshold.

25. The computer-readable storage medium according to claim 23, wherein at least one IPI is detected as sent by the first VM and received by the second VM and the second VM is determined as interrupted by the first VM.

26. The computer-readable storage medium according to claim 23, wherein the CMR is computed through a first number of cache misses divided by a second number of total cache accessed by the third VM.

27. The computer-readable storage medium according to claim 23, wherein a suspicion value associated with the side channel attack is computed by:

multiplying a first value, a second value, and a third value with a first coefficient, a second coefficient, and a third coefficient, respectively; and

adding products of the first value and the first coefficient, the second value and the second coefficient, and the third value and the third coefficient, wherein the first value is determined by dividing the IPI rate by a predetermined IPI rate threshold, the second value is determined by dividing the TSC rate by the predetermined TSC rate threshold, the third value is determined by dividing the CMR by a predetermined CMR threshold.

28. The computer-readable storage medium according to claim 27, wherein the first VM and the second VM are identified as an attacker pair in response to a detection that the suspicion value exceeds a predetermined suspicion value threshold.

29. The computer-readable storage medium according to claim 23, wherein the IPI rate, the TSC, the CMR and data associated with the first VM, second VM, and third VM are clustered into an event in response to determining the side channel attack and the event is recorded in a data store.

30. The computer-readable storage medium according to claim 23, wherein at least one datacenter operator is notified in response to determining the side channel attack.

31. The computer-readable storage medium according to claim 23, wherein at least one datacenter operator is enabled to adjust one or more of: the predetermined IPI rate threshold, the predetermined TSC threshold, a predetermined CMR threshold, the first coefficient, the second coefficient, the third coefficient, and a predetermined suspicion value threshold.

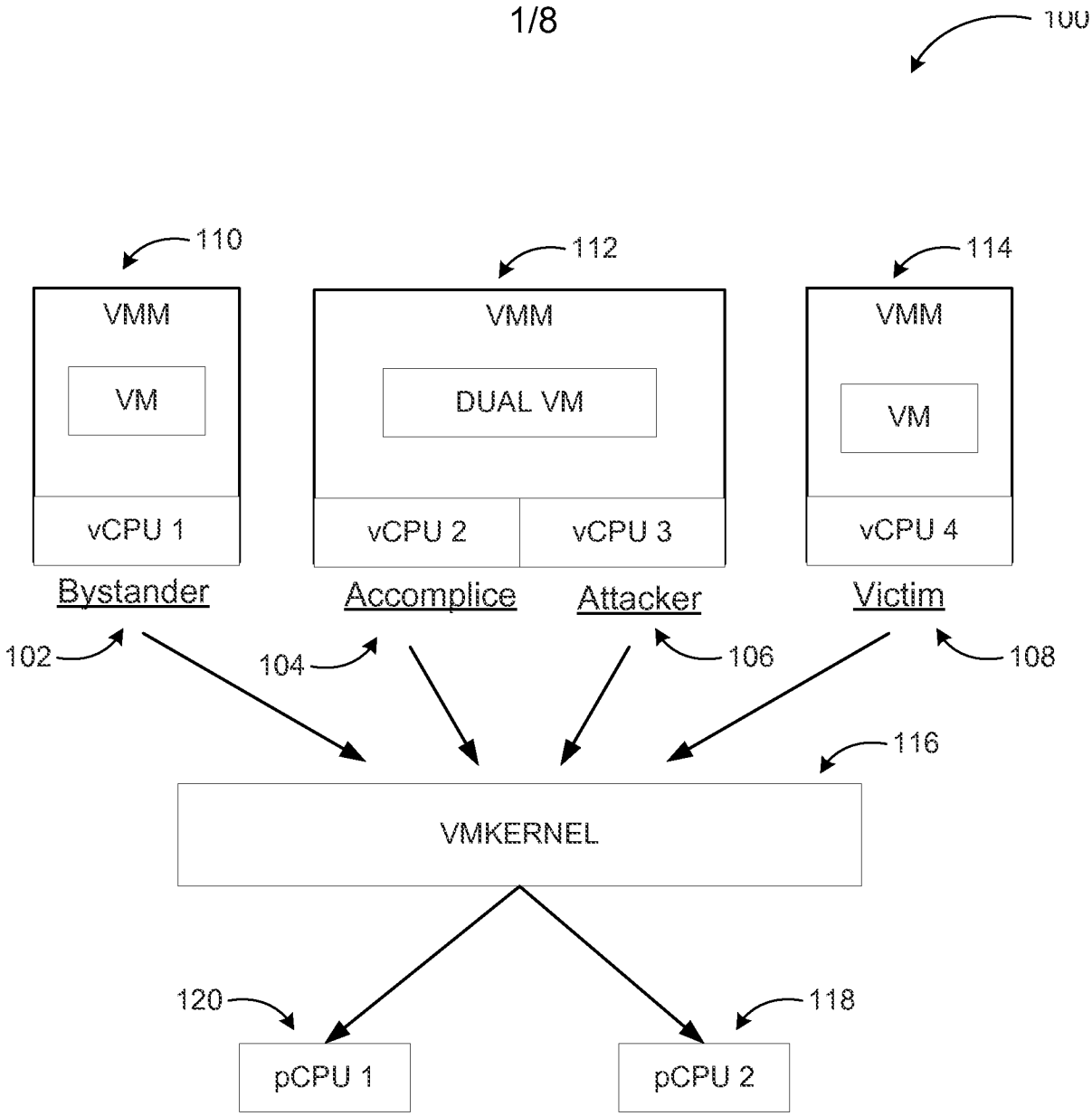

100
110
112
114
VMM
VM
VMM
DUAL VM
VMM
VM
vCPU 1
vCPU 2
vCPU 3
vCPU 4
Bystander
Accomplice
Attacker
Victim
102
104
106
108
116
VMKERNEL
120
118
pCPU 1
pCPU 2

FIG. 1

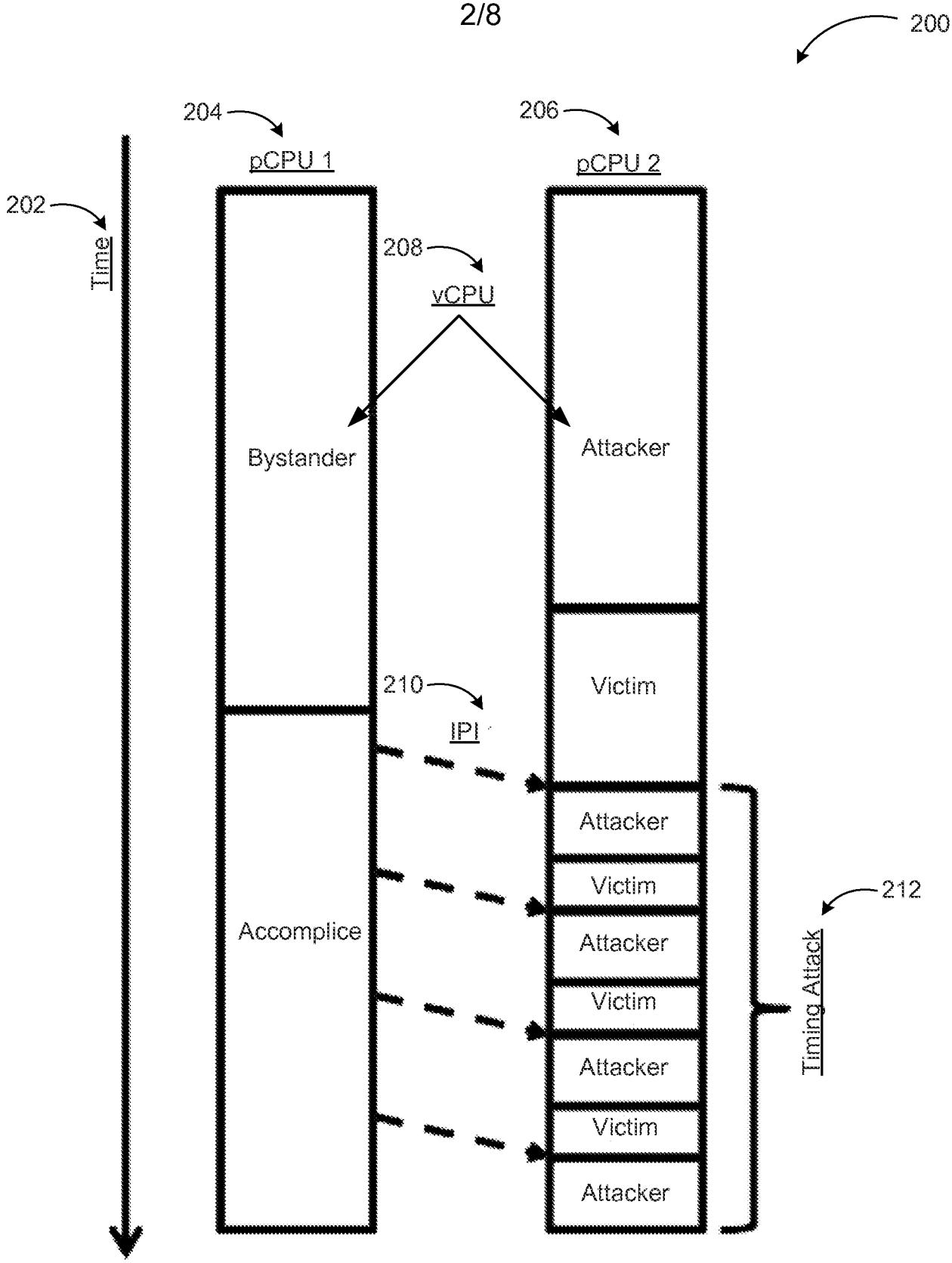

200
204
pCPU 1
206
pCPU 2
202
Time
208
vCPU
Bystander
Attacker
Victim
210
IPI
Attacker
Victim
Accomplice
Attacker
Victim
Attacker
Victim
Attacker
212
Timing Attack

FIG. 2

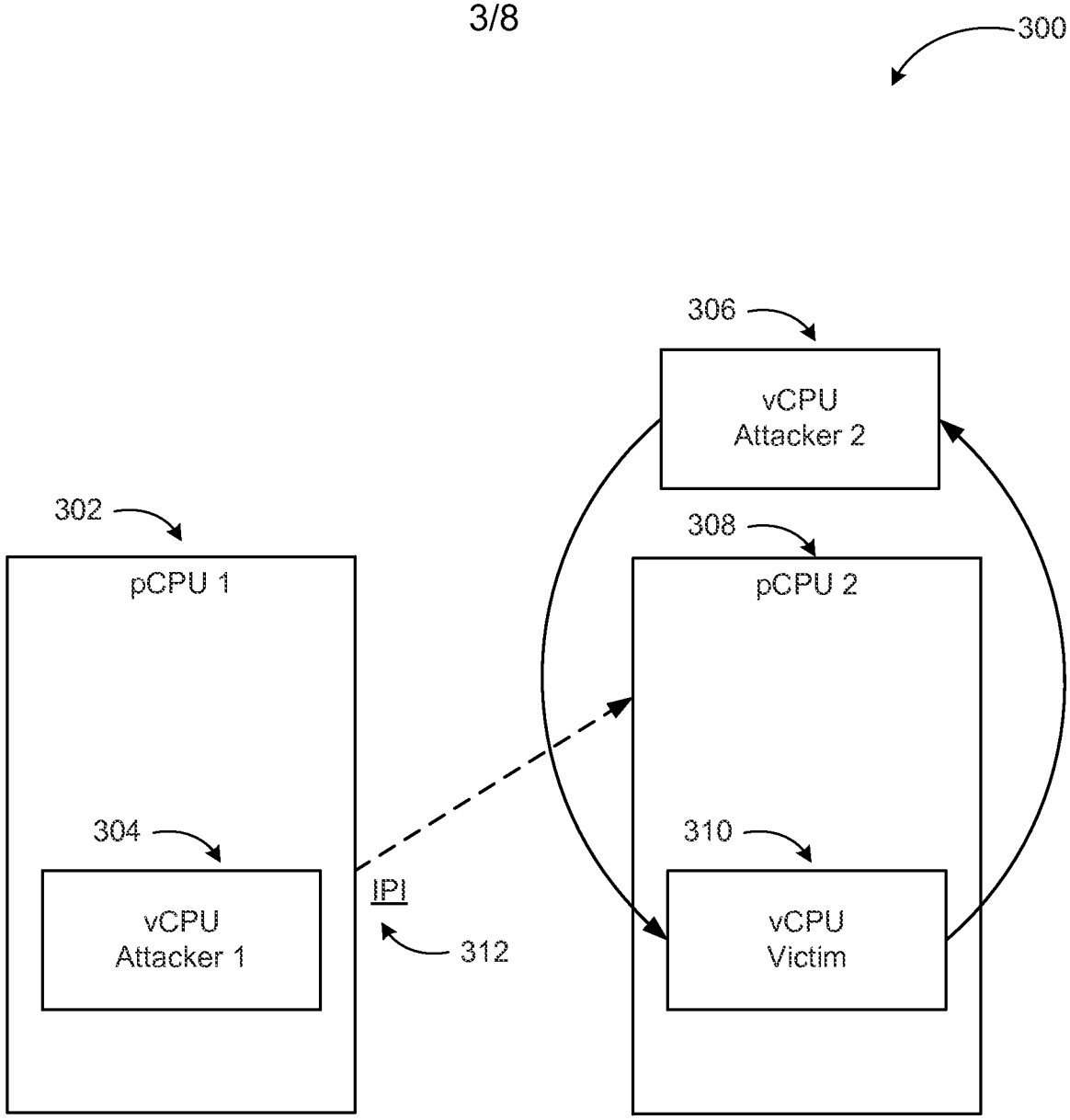

300
306
vCPU
Attacker 2
302
308
pCPU 1
pCPU 2
304
310
IPI
312
vCPU
Attacker 1
vCPU
Victim

FIG. 3

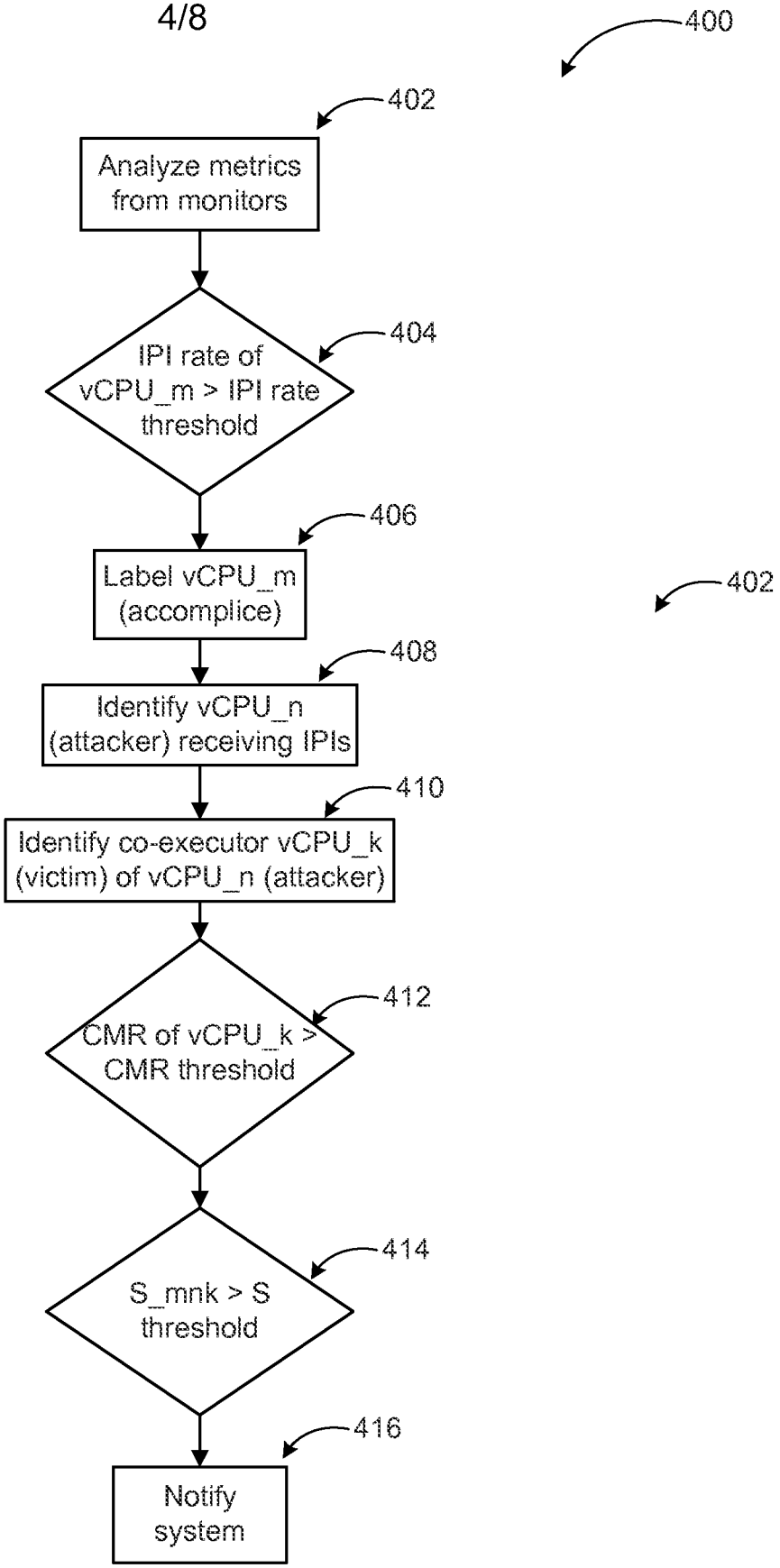

400
402
Analyze metrics from monitors
404
IPI rate of vCPU_m > IPI rate threshold
406
Label vCPU_m (accomplice)
402
408
Identify vCPU_n (attacker) receiving IPIs
410
Identify co-executor vCPU_k (victim) of vCPU_n (attacker)
412
CMR of vCPU_k > CMR threshold
414
S_mnk > S threshold
416
Notify system

FIG. 4

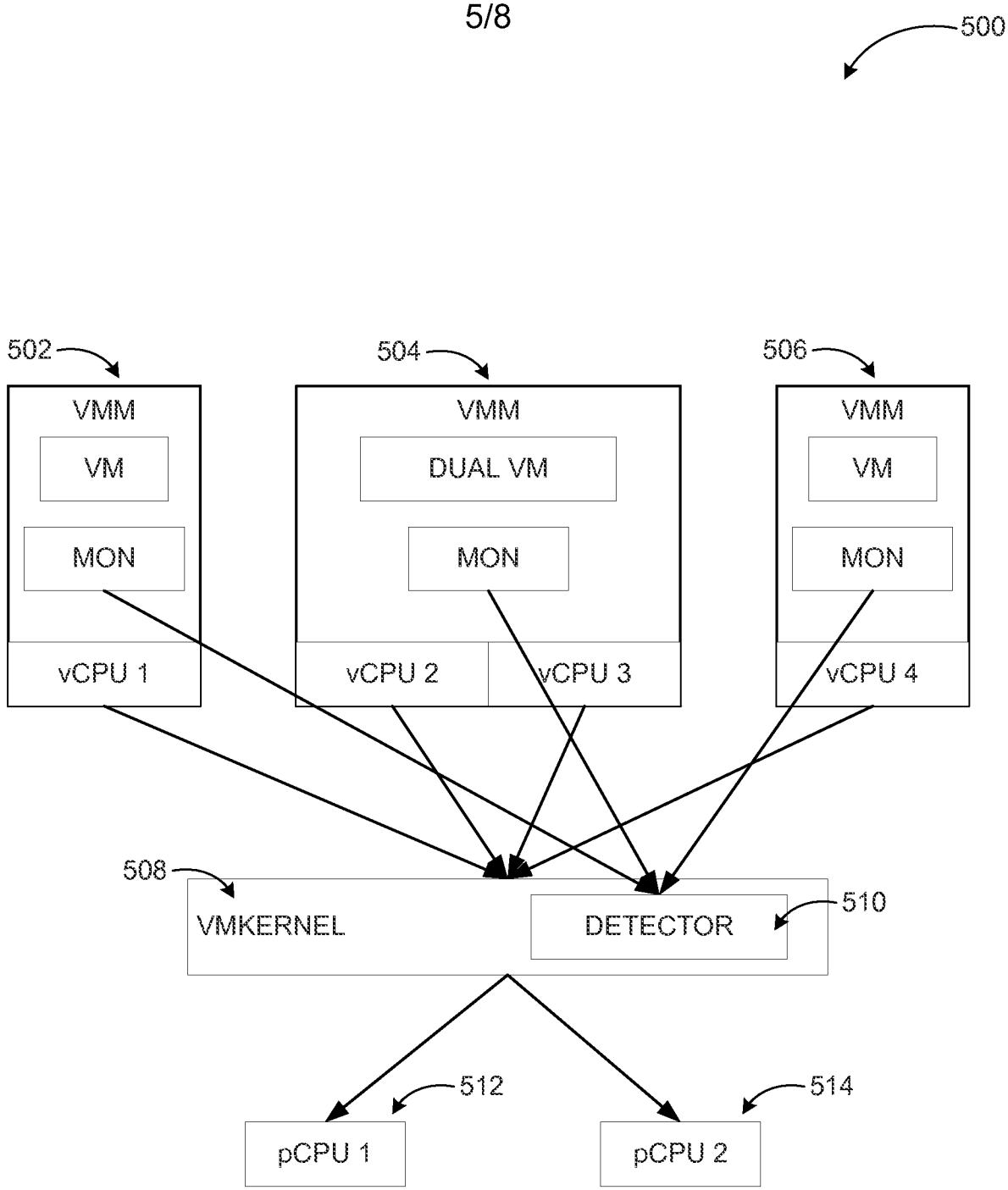

500
502
VMM
VM
MON
vCPU 1
504
VMM
DUAL VM
MON
vCPU 2
vCPU 3
506
VMM
VM
MON
vCPU 4
508
VMKERNEL
DETECTOR
510
512
pCPU 1
514
pCPU 2

FIG. 5

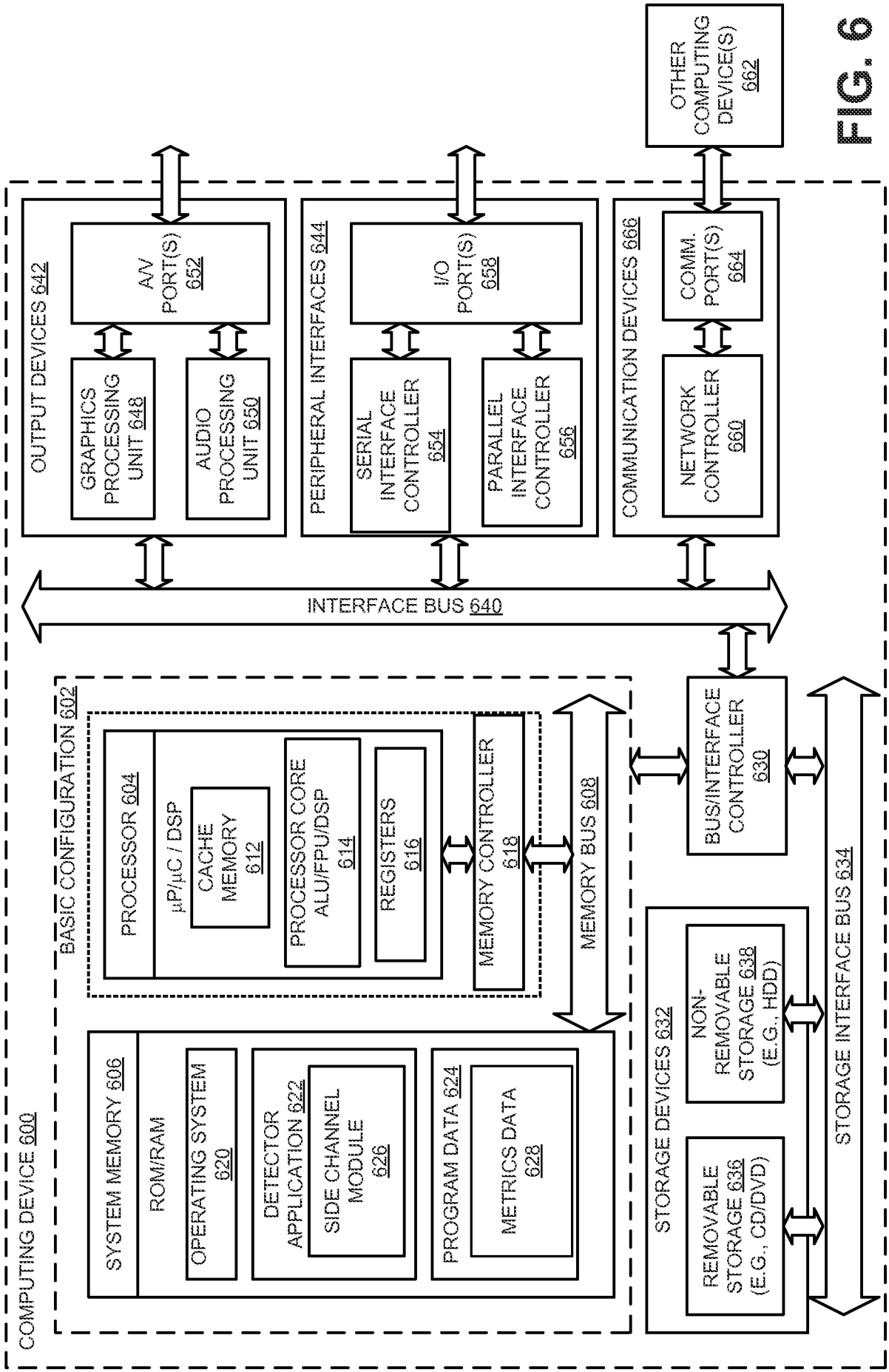

COMPUTING DEVICE 600
BASIC CONFIGURATION 602
PROCESSOR 604
μP/μC / DSP
CACHE MEMORY 612
PROCESSOR CORE ALU/FPU/DSP 614
REGISTERS 616
MEMORY CONTROLLER 618
MEMORY BUS 608
SYSTEM MEMORY 606
ROM/RAM
OPERATING SYSTEM 620
DETECTOR APPLICATION 622
SIDE CHANNEL MODULE 626
PROGRAM DATA 624
METRICS DATA 628
BUS/INTERFACE CONTROLLER 630
STORAGE DEVICES 632
NON-REMOVABLE STORAGE 638 (E.G., HDD)
REMOVABLE STORAGE 636 (E.G., CD/DVD)
STORAGE INTERFACE BUS 634
INTERFACE BUS 640
OUTPUT DEVICES 642
A/V PORT(S) 652
GRAPHICS PROCESSING UNIT 648
AUDIO PROCESSING UNIT 650
PERIPHERAL INTERFACES 644
I/O PORT(S) 658
SERIAL INTERFACE CONTROLLER 654
PARALLEL INTERFACE CONTROLLER 656
COMMUNICATION DEVICES 666
COMM. PORT(S) 664
NETWORK CONTROLLER 660
OTHER COMPUTING DEVICE(S) 662

FIG. 6

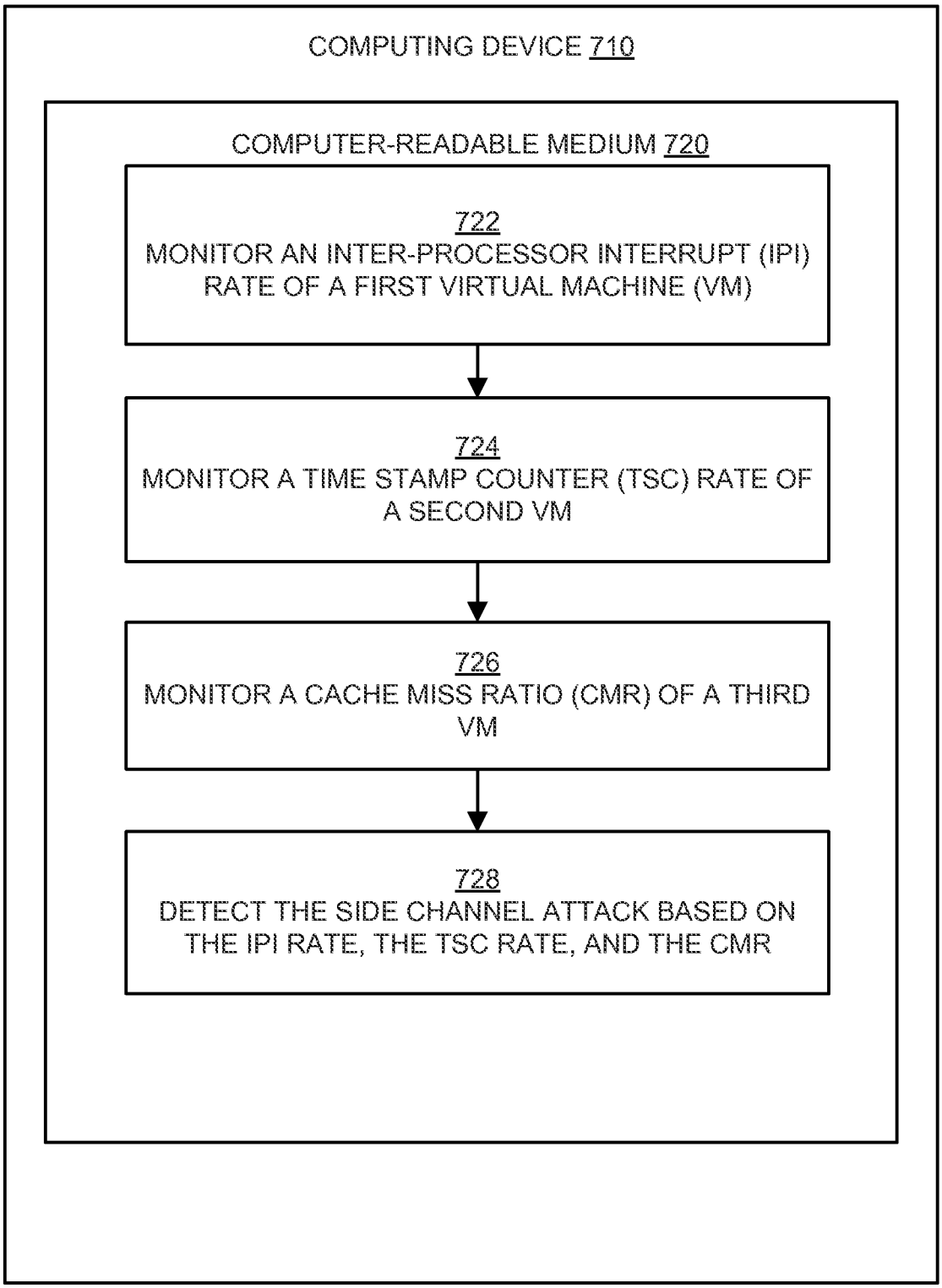

COMPUTING DEVICE 710
COMPUTER-READABLE MEDIUM 720
722
MONITOR AN INTER-PROCESSOR INTERRUPT (IPI) RATE OF A FIRST VIRTUAL MACHINE (VM)
724
MONITOR A TIME STAMP COUNTER (TSC) RATE OF A SECOND VM
726
MONITOR A CACHE MISS RATIO (CMR) OF A THIRD VM
728
DETECT THE SIDE CHANNEL ATTACK BASED ON THE IPI RATE, THE TSC RATE, AND THE CMR

FIG. 7

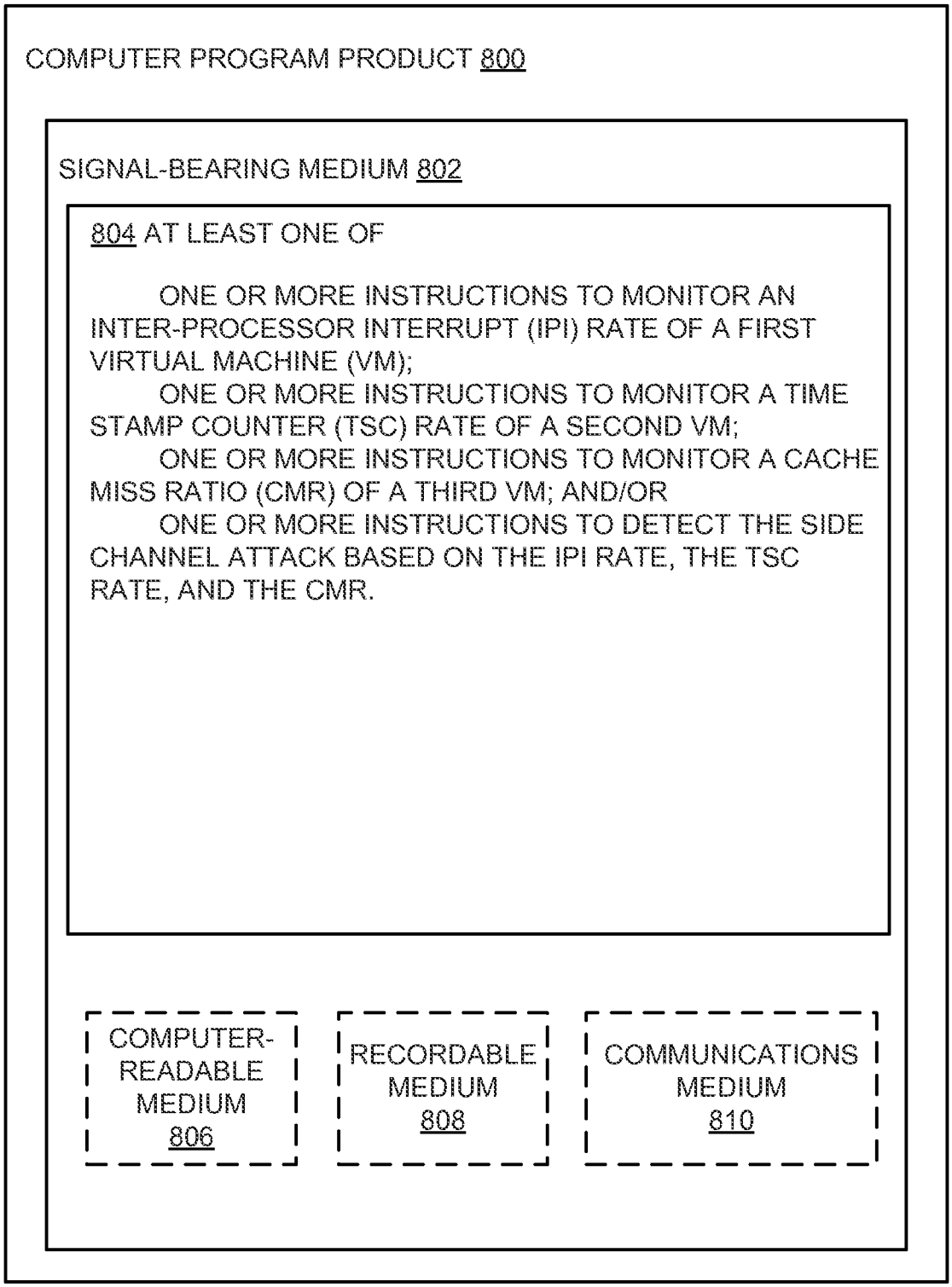

COMPUTER PROGRAM PRODUCT 800
SIGNAL-BEARING MEDIUM 802
804 AT LEAST ONE OF
ONE OR MORE INSTRUCTIONS TO MONITOR AN INTER-PROCESSOR INTERRUPT (IPI) RATE OF A FIRST VIRTUAL MACHINE (VM);
ONE OR MORE INSTRUCTIONS TO MONITOR A TIME STAMP COUNTER (TSC) RATE OF A SECOND VM;
ONE OR MORE INSTRUCTIONS TO MONITOR A CACHE MISS RATIO (CMR) OF A THIRD VM; AND/OR
ONE OR MORE INSTRUCTIONS TO DETECT THE SIDE CHANNEL ATTACK BASED ON THE IPI RATE, THE TSC RATE, AND THE CMR.
COMPUTER-READABLE MEDIUM 806
RECORDABLE MEDIUM 808
COMMUNICATIONS MEDIUM 810

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US13/73140

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 21/50 (2014.01)
USPC - 726/25
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) Classification(s): 726/22, 23, 24, 25
USPC Classification(s): G06F 21/50, 21/56, 21/67 (2014.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
MicroPatent (US-G, US-A, EP-A, EP-B, WO, JP-bib, DE-C,B, DE-A, DE-T, DE-U, GB-A, FR-A); ProQuest; IEEE; Google/Google Scholar

Keywords: side-channel attack, cache miss, interrupt, threat, detect, score, virtual machine, threshold, rate

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2013/172913 A2 (SETHUMADHAVAN, L. et al.) 21 November 2013; paragraph [0004], [0009], [0034], [0042], [0043], [0044], [0079].	1-15, 17-31
Y	US 2013/0297832 A1 (AHMAD, I. et al.) 07 November 2013; claim 8, figures 3, 4, paragraphs [0050], [0051], [0056].	1-15, 17-31
Y	US 5,761,715 A (TAKAHASHI, M.) 02 June 1998; column 7, lines 32-38.	6, 17, 26
Y	US 2011/0214187 A1 (WITTENSTEIN, A. et al.) 01 September 2011; paragraphs [0205], [0230], [0260].	7-10, 18-21, 27-29
Y	US 2013/0232577 A1 (WATTERS, J. et al.) 05 September 2013; paragraph [0019].	8, 19, 28
Y	US 7,418,733 B2 (CONNARY, I. et al.) 26 August 2008; column 13, line 67 - column 14, line 3.	10, 20, 29
Y	US 2012/0266221 A1 (CASTELLUCCIA, C. et al.) 18 October 2012; paragraphs [0025], [0028], [0029].	11, 22, 30

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
04 May 2014 (04.05.2014)	23 MAY 2014
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774