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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 November 2014 (20.11.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/185845 Al

(51) International Patent Classification:
H04L 9/08 (2006.0 1) *H04L 9/32* (2006.0 1)

(21) International Application Number:
PCT/SE20 13/05 1299

(22) International Filing Date:
6 November 2013 (06.1 1.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/822,488 13 May 2013 (13.05.2013) US

(71) Applicant: **TELEFONAKTIEBOLAGET L M ERIC¬SSON (PUBL)** [SE/SE]; SE-164 83 Stockholm (SE).

(72) Inventors: **GEHRMANN, Christian;** Skordevagen 2C, SE-22738 Lund (SE). **MORENIUS, Fredric;** Rasundavagen 173, SE-16936 Solna (SE). **PALADI, Nicolae;** Kummelbyvagen 1J, SE-19140 Sollentuna (SE).

(74) Agent: **EGRELIUS, Fredrik;** Ericsson AB, Patent Unit Kista DSM, SE- 16480 Stockholm (SE).

(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).

Published:

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

WO 2014/185845 A1

(54) Title: PROCEDURE FOR PLATFORM ENFORCED SECURE STORAGE IN INFRASTRUCTURE CLOUDS

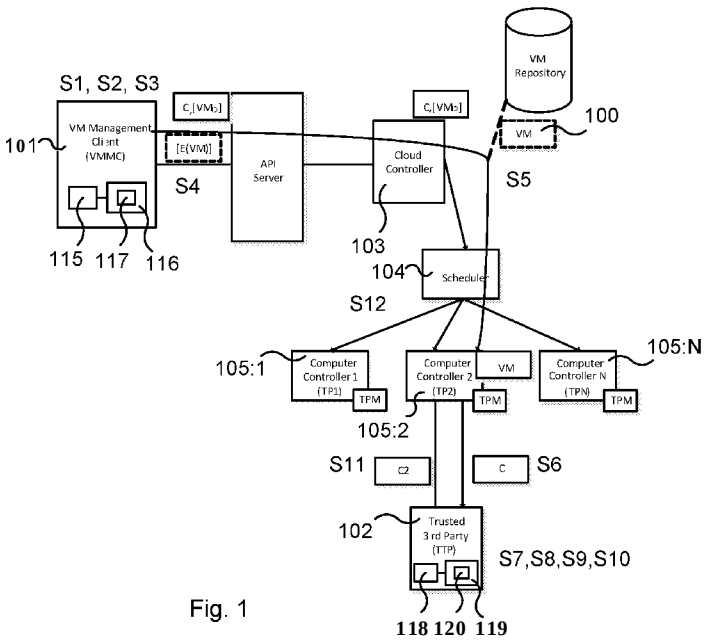

Fig. 1

(57) Abstract: The present invention relates to a secure component for protecting data in a storage entity and a method at the secure component of protecting data in the storage entity. Further, the present invention relates to a secure domain manager for securely associating a communicating party with a storage domain and a method at the secure domain manager of securely associating the communicating party with the storage domain. Moreover, the present invention relates to a trusted third party for verifying correctness of a launch package created by a secure domain manager to securely associate a communicating party with a storage domain and a method at the trusted third party to verify correctness of the launch package created by the secure domain manager to securely associate the communicating party with the storage domain.

PROCEDURE FOR PLATFORM ENFORCED SECURE STORAGE IN INFRASTRUCTURE CLOUDS

TECHNICAL FIELD

The present invention relates to a secure component for protecting data in a storage entity and a method at the secure component of protecting data in the storage entity. Further, the present invention relates to a secure domain manager for securely associating a communicating party with a storage domain and a method at the secure domain manager of securely associating the communicating party with the storage domain. Moreover, the present invention relates to a trusted third party for verifying correctness of a launch package created by a secure domain manager to securely associate a communicating party with a storage domain and a method at the trusted third party to verify correctness of the launch package created by the secure domain manager to securely associate the communicating party with the storage domain. The present invention relates further to corresponding computer programs and computer program products.

BACKGROUND

During recent years there has been a strong move in the marketplace towards the use of virtualization technologies. Among other capabilities, virtualization allows one to run unmodified legacy operating systems and applications on new hardware platforms using Virtual Machines (VMs). This is realized through on-the-fly translation from one hardware instruction set to another with the assistance of a so-called hypervisor or Virtual Machine Monitor (VMM). The VMM type considered here runs in the most privileged mode of a system and has full control over all vital system resources. A VMM-based system not only allows instruction translation but also, more importantly, increased system utilization as multiple VMs can run simultaneously on a single powerful hardware platform, opening for new business models and a new business landscape. This implies, for example, that existing services rather easily can be migrated into large and dynamic computing clusters, what often is referred to as "the cloud".

The cloud model where the customer is allowed to run a complete VM, including operating system, is often referred to as the Infrastructure as a Service (IaaS).

The new flexibility however has a price: increased security risks. Previously, physically isolated software systems might now run in VMs on the same physical node as other, completely unrelated, VMs. This allows for new types of attacks between VMs running simultaneously on the same hardware. Also, the VMM becomes a target for new types of attacks. Once the VMM is compromised the whole system is compromised. Furthermore, persistent data that was previously stored locally on a physical machine or within protected network boundaries in a central database, must now be available to a VM running in a potentially hostile network. Hence, there must be means to protect stored data and make sure that plaintext data only is exposed to VMs which are authorized to have access to it.

A large amount of academic and industry research has been carried out in the area of IaaS or "cloud" storage security.

For example, the prior art cryptographic storage system CloudProof allows read/write access that achieves integrity, confidentiality, fork-consistency and freshness, along with capabilities to provide proofs of data tampering. Kamara et al. introduced CS2 in "CS2: A searchable cryptographic cloud storage system", where in addition to the properties achieved in CloudProof, the cloud storage solution also achieves global integrity and scalable search. The latter is in particular achieved through the use of symmetric searchable encryption.

An IaaS cloud model may, e.g., use a so called OpenStack Nova architecture where a VM Management Client (VMMC) launches and controls VMs through well-defined API(s). From a security perspective, there are several problems. First, there must be means for the VMMC to have guarantees on that the VM instance not will be launched on an adverse host (wrongly configured or deliberately misconfigured host software/hardware platform).

Second, VM image integrity must be guaranteed prior to VM instance launch. These two conditions can be satisfied through alternative methods, as, e.g., implemented in CloudProof as well as in other prior art techniques using Trusted Computing Technologies as defined by the Trusted Computing Group (TCG) in combination with well-designed protocols for VM launch and migration.

However, these methods and protocols do not address the problem of how a VM instance (once it has been launched or migrated) obtains secure access to sensitive persistent data it is potentially dependent on. In particular, there should be means to protect both confidentiality and integrity of sensitive data during storage, and only allow authorized VM instances to have access to the plaintext representation of the data. While protection of data can be achieved through encryption and integrity protection, Message Authentication Codes (MAC) algorithms, the cryptographic keys used for encryption and MAC calculations must be available to the VM instance whenever it requires the data. Furthermore, cryptographic key handling schemes should minimize VMMC involvement in order to minimize the risks of mismanagement or client compromise.

SUMMARY

An object of the present invention is to solve, or at least mitigate, these problems in the art.

The present invention addresses these problems by providing trusted VM launch, storage confidentiality and integrity protection of data to be stored.

This problem is attained in a first aspect of the present invention by a method at a secure component of protecting data in a storage entity. The method comprises receiving, from a communicating party, a request to store data in one of a plurality of storage entities of a storage domain, encrypting a request for cryptographic keys associated with said one of a plurality of storage entities with a session key of the storage domain to protect confidentiality and authenticity of the request, and sending, to a trusted third party, the

encrypted request for cryptographic keys. Further, the method comprises receiving, from the trusted third party, cryptographic keys to protect confidentiality and integrity of the data to be stored, which cryptographic keys have been encrypted with the session key of the storage domain, decrypting the encrypted cryptographic keys, and associating the received cryptographic keys with the storage domain and the communicating party. Moreover, the method comprises encrypting the data with the cryptographic keys and storing the encrypted data in said one of a plurality of storage entities.

This problem is attained in a second aspect of the present invention by a method at a secure domain manager of securely associating a communicating party with a storage domain. The method comprises creating a launch package comprising an identifier for the storage domain, and an assertion that the secure domain manager is authorized to associate the communicating party with said storage domain, and digitally signing the launch package. Further, the method comprises acquiring a certificate from a trusted certificate authority configured to certify a verification key corresponding to that used for digitally signing the launch package, and submitting the digitally signed launch package and the certificate to a trusted third party for verification, such that the communicating party subsequently can be securely associated with the storage domain.

This problem is attained in a third aspect of the present invention by a method at a trusted third party to verify correctness of a launch package created by a secure domain manager to securely associate a communicating party with a storage domain. The method comprises receiving a digitally signed a launch package comprising an identifier for the storage domain, and an assertion that the secure domain manager is authorized to associate the communicating party with said storage domain, and receiving a certificate issued by a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch package. Further, the method comprises verifying the assertion to ensure that the secure domain manager is authorized to associate the communicating party with

said storage domain, such that the communicating party subsequently can be securely associated with the storage domain.

Further, a device according to the respective method is provided.

Thus, the problem is attained in a fourth aspect of the present invention by a secure component for protecting data in a storage entity. The secure component comprises a processor and a memory. The memory contains instructions executable by the processor, whereby the secure component is operative to receive, from a communicating party, a request to store data in one of a plurality of storage entities of a storage domain, and encrypt a request for cryptographic keys associated with said one of a plurality of storage entities with a session key of the storage domain to protect confidentiality and authenticity of the request. Further, the secure component is operative to send, to a trusted third party, the encrypted request for cryptographic keys, and receive, from the trusted third party, cryptographic keys to protect confidentiality and integrity of the data to be stored, which cryptographic keys have been encrypted with the session key of the storage domain. Moreover, the secure component is operative to decrypt the encrypted cryptographic keys, associate the received cryptographic keys with the storage domain and the communicating party, and encrypt the data with the cryptographic keys and store the encrypted data in said one of a plurality of storage entities.

Thus, the problem is attained in a fifth aspect of the present invention by a secure domain manager for securely associating a communicating party with a storage domain. The secure domain manager comprises a processor and a memory. The memory contains instructions executable by the processor, whereby the secure domain manager is operative to create a launch package comprising an identifier for the storage domain and an assertion that the secure domain manager is authorized to associate the communicating party with said storage domain, and digitally sign the launch package. Further the secure domain manager is operative to acquire a certificate from a trusted certificate authority configured to certify a verification key corresponding to

that used for digitally signing the launch package, and submit the digitally signed launch package and the certificate to a trusted third party for verification, such that the communicating party subsequently can be securely associated with the storage domain.

Thus, the problem is attained in a sixth aspect of the present invention by a trusted third party for verifying correctness of a launch package created by a secure domain manager to securely associate a communicating party with a storage domain. The trusted third party comprises a processor and a memory. The memory contains instructions executable by the processor, whereby said trusted third party is operative to receive a digitally signed launch package comprising an identifier for the storage domain and an assertion that the secure domain manager is authorized to associate the communicating party with said storage domain. Further the trusted third party is operative to receive a certificate issued by a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch package; and verify the assertion to ensure that the secure domain manager is authorized to associate the communicating party with said storage domain, such that the communicating party subsequently can be securely associated with the storage domain.

Thus, the present invention advantageously addresses the outlined problems of the prior art by providing a persistent data protection. The present invention is advantageous for a number of reasons:

1. Data can be stored in the infrastructure cloud using any suitable storage units, such as block storage devices (e.g., Internet Small Computer System Interface (iSCSI) or similar). Confidentiality and integrity of the data is advantageously protected during storage.

2. Confidentiality and integrity protection of data on IaaS compute hosts can advantageously be ensured by a trusted VMM or by an entity trusted by the VMM (e.g., a privileged VM domain), further referred to as a "Secure Component" (SC). The SC has access to the storage keys needed to perform

the necessary cryptographic operations on the data during storage and information retrieval.

3. Data stored in the IaaS cloud using the scheme described in the present invention is associated with specific storage domains. In the present invention, a storage domain typically corresponds to a particular organization or administrative domain that utilizes the cloud services (including the storage service) offered by the IaaS provider, i.e., a single administrative domain that typically only handles data storage for its own domain and not for any other domains. All data in a single domain is advantageously protected with the same storage protection master key, the domain key.

4. During the entire lifetime of a communicating party, referred to throughout the description as a VM, the VM is advantageously securely associated with a particular storage domain.

5. All keys used to encrypt/decrypt and integrity check data in a single domain are advantageously handled by a special trusted entity in the network referred to as a Trusted Third Party (TTP). The TTP assigns and handles domain keys. The domain keys cannot ever leave the TTP.

As can be deducted from the above, the present invention advantageously introduces:

· Principles for securely assigning a communicating party in the form of a VM to a particular storage domain at VM launch. This is mainly undertaken by the secure domain manager according to the second aspect of the present invention, also referred to as the VMMC, and further by the TTP according to the third aspect of the present invention.

· Principles for the SC to securely retrieve encryption and integrity protection keys when writing data to a persistent data storage area in the IaaS cloud.

• Principles for the SC to securely retrieve decryption and integrity verification keys for data retrieval.

Preferred embodiments of the present invention will be set out in the following.

The present invention allows protection of persistent data at storage in an IaaS cloud almost transparently from the point of view of VMMC. To achieve this, the present invention only relies on a trusted third party and a trusted configuration of a communicating party, i.e. a target node. The trustworthiness of the target node is ensured through the TCG sealing mechanism which makes the trust explicit. Only trusted entities in the IaaS have access to the security critical storage parameters or plaintext data, allowing arbitrary storage medium to be used as long as reliability (not confidentiality and integrity) can be guaranteed for these storage mediums. This opens up for secure and cost efficient storage handling in IaaS clouds.

To this end, protection of data on storage units in a cloud environment is provided. The approach of protecting data comprises intercepting a secure storage read or write request from a trusted VM, assigning the requesting VM into a trusted storage domain determined by a storage domain identifier (ID), communicating with a trusted entity in the from of a TTP to acquire confidentiality and integrity protection keys for the trusted storage domain, determining an available storage resource, and confidentiality and/or integrity protecting the data requested to be stored by the VM. The storage domain ID is specified by the VMMC at VM launch. Preferably, the method is performed by a Secure Component (SC) running on a VMM, i.e., a node of a cloud environment.

In an embodiment of the present invention, the integrity and/ or confidentiality protection keys acquired from the trusted entity are calculated using a domain wide master key which is kept protected by the trusted entity.

In a further embodiment of the present invention, the integrity and confidentiality keys are calculated using a random nonce provided by the trusted entity.

In yet another embodiment of the present invention, upon requesting a new storage confidentiality and integrity key, the random nonce chosen by the trusted entity is sent back to the SC in the form of a token and stored as part of a metadata field of the protected stored data.

In still another embodiment of the present invention, a key internally kept by the trusted entity, in the form of an integrity key is used to integrity protect the token and thus the metadata.

In embodiments of the present invention,, when the VM requests read or write of protected data, the SC uses the metadata and sends it to the trusted entity in order to retrieve the confidentiality and/ or integrity protection keys used to decrypt and/ or verify the stored data, or encrypt and integrity protect data to be stored.

In an embodiment of the present invention, the storage domain ID assigned to the VM is provided by the VM owner to the VMMC at launch of the VM.

In further embodiments of the present invention, the storage confidentiality and/ or integrity protection keys provided by the trusted entity are sealed, i.e., encrypted using a trusted platform module in the form of the session key of the storage domain, to a trusted state of the target platform which is hosting the SC, i.e., the VMM, such that the keys will only be available to the SC if it is running on a trusted platform.

In still further embodiments of the present invention, after successfully acquiring confidentiality and/ or integrity protection keys from the trusted entity, the SC keeps these keys in a protected cache memory for future use.

In an embodiment of the present invention, the SC sends a reference to the chosen storage resource and domain to be used for future write read access to the storage resource back to the requesting VM.

In the third aspect of the present invention, upon request for storage integrity and/ or confidentiality keys for a particular storage domain, the trusted entity checks that the client that has launched the VM is authorized to request

based on any of the following parameters provided by the client at VM launch: the storage domain ID for the storage domain that the client intend to use for the VM, and an assertion authorizing the client to access and store protected data at the chosen storage domain.

In further embodiments of the present invention, the launch package comprises an ID of the VM to be launched and/or an encrypted nonce.

In the second aspect of the present invention, at launch via the trusted platform, the client (i.e. the VMMC) sends a launch message to the trusted entity, the launch message comprising a signature over parameters provided by the client at VM launch. Further, the client sends, via the trusted platform, a certificate certifying the key used to sign the launch message to the trusted entity.

Further embodiments of the present invention will be discussed in the detailed description.

Thus, embodiments of the present invention address enable trusted VM launch, key management, data access authorization, and persistent storage confidentiality and integrity protection, in order to ensure protected access to persistent VM instance sensitive data in IaaS models.

It is noted that the invention relates to all possible combinations of features recited in the claims. Further features of, and advantages with, the present invention will become apparent when studying the appended claims and the following description. Those skilled in the art realize that different features of the present invention can be combined to create embodiments other than those described in the following.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is now described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 illustrates a system implementing embodiments of the present invention;

Figure 2 illustrates a flowchart of an embodiment of a method of securely associating a communicating party with a storage domain according to the second aspect of the present invention;

Figure 3 illustrates a flowchart of an embodiment of a method of verifying correctness of a launch package in order to securely associate a communicating party with a storage domain according to the third aspect of the present invention;

Figure 4 shows a flowchart of a further embodiment of the method according to the third aspect of the present invention;

Figure 5 illustrates a system implementing embodiments of the present invention;

Figure 6 illustrates a flowchart of an embodiment of a method of protecting data in a storage entity according to the first aspect of the present invention;

Figure 7 illustrates the system of Figure 5 implementing further embodiments of the present invention;

Figure 8a shows a secure component according to an embodiment of the first aspect of the present invention;

Figure 8b shows a secure domain manager according to an embodiment of the second aspect of the present invention; and

Figure 8c shows a trusted third party according to an embodiment of the third aspect of the present invention.

DETAILED DESCRIPTION

The invention will now be described more fully hereinafter with reference to the accompanying drawings, in which certain embodiments of the invention are shown. This invention may, however, be embodied in many different

forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided by way of example so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. Like numbers refer to like elements throughout the description.

Figure 1 illustrates a system implementing embodiments of the second and third aspect of the present invention. The system set out in Figure 1 will be described in detail in the following. However, before the detailed description is given, a more general embodiment of the method according to the second aspect of the present invention, as well as of the third aspect, will be described. The method according to the second aspect of the present invention of securely associating a communicating party, such as a VM 100, with a storage domain (not shown in Figure 1; described in the following with reference to Figures 5 and 7) is performed at a secure domain manager, e.g. a VMMC 101. In practice, the method at the VMMC 101 is performed by a processing unit 115 embodied in the form of one or more microprocessors arranged to execute a computer program 117 downloaded to a suitable storage medium 116 associated with the microprocessor, such as a Random Access Memory (RAM), a Flash memory or a hard disk drive. The processing unit 115 is arranged to carry out the method according to embodiments of the first aspect of the present invention when the appropriate computer program 117 comprising computer-executable instructions is downloaded to the storage medium 116 and executed by the processing unit 115. The storage medium 116 may also be a computer program product comprising the computer program 117. Alternatively, the computer program 117 may be transferred to the storage medium 116 by means of a suitable computer program product, such as a floppy disk or a memory stick. As a further alternative, the computer program 117 may be downloaded to the storage medium 116 over a network. The processing unit 115 may alternatively be embodied in the form of a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a complex programmable logic device (CPLD), etc.

Now, in order to advantageously securely associate a communicating party, such as a VM 100, with a storage domain, an embodiment of a method according the second aspect of the present invention is proposed, a flowchart of which is shown in Figure 2. Reference is further made to Figure 1. In a first step, S101, the VMMC 101 creates a launch package comprising an identifier for the storage domain, and an assertion that the secure domain manager is authorized to associate the VM 100 with said storage domain. In step S102, the VMMC 101 digitally signs the launch package using an appropriate signing key, and in step S103, the VMMC 101 acquires a certificate from a trusted certificate authority (not shown in Figure 1) for certifying a verification key corresponding to the signing key previously used for digitally signing the launch package. Finally, in step S104, the VMMC 101 submits the digitally signed launch package and the certificate to a trusted third party 102 for verification, such that the VM 100 subsequently can be securely associated with the storage domain.

Again with reference to Figure 1, the method according to the third aspect of the present invention of verifying correctness of a launch package created by a VMMC 101 to securely associate a VM 100 a with a storage domain is performed at the TTP 102. In practice, similar to the description of the VMMC 101 hereinabove, the method at the TTP 102 is performed by a processing unit 118 embodied in the form of one or more microprocessors arranged to execute a computer program 120 downloaded to a suitable storage medium 119 associated with the microprocessor, as previously has been discussed with reference to the VMMC 101.

In order to advantageously verify correctness of a launch package created by the VMMC 101 to securely associate the VM 100 a with a storage domain, an embodiment of a method according the third aspect of the present invention is proposed, a flowchart of which is shown in Figure 3. Reference is further made to Figure 1. In a first step, S201, the TTP 102 receives, from the VMMC 101, a digitally signed launch package comprising an identifier for the storage domain, and an assertion that the VMMC 101 is authorized to associate the VM 100 with the storage domain. In a second step S202, the TTP 102 receives

a certificate issued by a trusted certificate authority (not shown in Figure 1) for certifying a verification key corresponding to that used for digitally signing the launch package. Thereafter, the TTP 102 verifies in step S203 the assertion to ensure that the VMMC 101 is authorized to associate the VM 100 with the storage domain, such that the VM 100 subsequently can be securely associated with the storage domain.

Figure 4 shows a flowchart of a further embodiment of the method according to the third aspect of the present invention. After step S203 has been performed by the TTP 102, it proceeds to step S204 of generating a session domain key for the storage domain and a target platform. Further, in step S205, the TTP 102 cryptographically protects the storage domain identifier and the session domain key with a secret key of a trusted platform 105:1, 105:2, 105:N on which the VM 100 is to reside before sending, in step S206, the cryptographically protected storage domain identifier and the session domain key to the trusted platform 105, wherein the trusted platform 105 can make available to a secure component the session domain key and securely associate the VM 100 with the storage domain.

Again with reference to Figure 1, the VM 100 launch procedure illustrated therein according to embodiments of the second and third aspects of the present invention will be described in detail in the following. The VM launch procedure is undertaken for securely associating a VM 100 with a particular storage domain at VM launch.

Si. The VMMC 101 prepares a VM 100 launch package containing a launch message, M. M comprises the following:

a. A certain storage domain ID, DID. In the following, it is assumed DID = A.

b. An assertion, AS, proving to a TTP 102, that the VMMC 101 is authorized to issue VMs 100 for storage domain A, i.e. that the VMMC 101 is authorized to associate the VMs 100 with the storage domain A.

c. Optionally a nonce, NVMMC, encrypted with the public key of the TTP 102, K_PUTTP.

d. Optional additional parameters such as required target Platform Security Level (SP), a VM identity, VMID and a hash HVM of the target VM image (i.e., the actual binary of the VM to be launched) or an encrypted key, KVM that can be used to decrypt the encrypted target VM image E(VM).

52. In addition, the VMMC 101 produces a digital signature, SIG, over all content in M using the private key of the VMMC 101, K_PIVMMC.

53. The VMMC 101 retrieves a certificate, **CertvMMc,** from a trusted certificate authority (not shown in Figure 1) that certifies the public key used by the VMMC 101. The corresponding private key is K_PIVMMC.

54. The VMMC 101 prepares a data structure C containing M, **CertvMMc** and SIG. Thus, the data structure C is configured to comprise the signed launch package created in step Si and the certificate acquired in step S3. The data structure C is sent to the IaaS provider, optionally along with E(VM) or an indication of the VM image that should be chosen for launch from a publicly available VM store (IDVMIMG).

55. The IaaS provider, which in Figure 1 is embodied in the form of a cloud controller 103 and a scheduler 104, selects a suitable available trusted platform (TP) 105 in the provider network and transfers the data structure C optionally together with the VM image (in encrypted or plaintext form) or the IDVMIMG to the chosen TP 105. Each TP 105 may optionally comprise a trusted platform module (TPM) for encrypting the storage confidentiality and/or integrity protection keys.

S6. When the data structure C reaches the intended TP 105 , the TP 105 contacts the TTP 102 and sends the data structure C to the TTP 102 for verification.

57. The the TTP 102 verifies the certificate, **CertvMMc,** and the signature SIG. If both CertvMMc and SIG are valid, the TTP 102 proceeds with the next step; otherwise it aborts with an error message to the intended TP 105.

58. The TTP 102 checks the assertion, AS (using the VMMC identity and key information provided in **CertvMMc)** and checks if the VMMC 101 is authorized to use storage domain A. If so, the TTP 102 proceeds with the next step, otherwise it aborts with an error message to the intended TP 105.

59. Using its private key, the TTP 102 decrypts NVMMC optionally received in the data structure C.

Sio. The TTP 102 generates a session domain key (SDA) for domain A and the target platform.

511. Parameters A and SDA (optionally together with other parameters such as NVMMC and HVM) are sealed to a protected state of the intended TP 105, i.e. encrypted with a public key of the TP 105 which has a property of only being available to the TP if the TP is booted in a trusted state. The encrypted message, denoted C2, is sent back to the intended TP 105.

512. The intended TP 105 unseals (i.e. decrypts) C2 and makes sure SDA is available to the SC which is to run on the TP. The TP 105 optionally assigns the TP unique identity VMID to the VM and associates that with domain A. The VM 100 is then launched in a secure isolated execution compartment on the TP 105.

Figure 5 illustrates a system implementing the first aspect of the present invention. The system set out in Figure 5 will be described in detail in the following. However, before the detailed description is given, a more general embodiment of the method according to the first aspect of the present invention will be described. The method according to the first aspect of the present invention of protecting data in a storage entity is performed at a secure component (SC) 106, such as, e.g., a computer, a server, a supervising entity in a telecom system, etc., i.e., a device having computing capabilities.

In practice, the method at the SC 106 is performed by a processing unit 121 embodied in the form of one or more microprocessors arranged to execute a computer program 123 downloaded to a suitable storage medium 122 associated with the microprocessor, as previously has been discussed in connection to Figure 1 and the VMMC 101 and the TTP 102.

In order to advantageously protect data in a storage entity, an embodiment of a method according the first aspect of the present invention is proposed, a flowchart of which is shown in Figure 6. Reference is further made to Figure 5. In a first step, S301, the SC 106 receives, from a communicating party in the form of a VM 100, a request to store data in one of a plurality of storage entities 107 of a storage domain 109, which storage entity 107 may be embodied as a specific storage area on a storage resource such as a server 108a. In Figure 5, a number of servers 108a, 108b, 108c make up the storage domain 109. It should be noted that the storage resources io8a-c comprised in the storage domain 109 could be remotely arranged, and are thus not necessarily adjacently located each other. In fact, they could even be located in different parts of the world. In a second step S302, the processing unit 121 encrypts a request for cryptographic keys associated with said one 107 of a plurality of storage entities with a session key of the storage domain 109 to protect confidentiality and authenticity of the request. Thereafter, in step S303, the SC 106 sends the encrypted request for cryptographic keys to a TTP 102. In a fourth step S304, the SC 106 receives from the TTP 102 cryptographic keys to protect confidentiality and integrity of the data to be stored, which cryptographic keys have been encrypted with the session key of the storage domain 109. In subsequent steps S305 and S306, the SC 106 decrypts the encrypted cryptographic keys and associates the received cryptographic keys with the storage domain 109 and the VM 100. Finally, in step S307, the SC 106 encrypts the data with the cryptographic keys and stores the encrypted data in said one 107 of a plurality of storage entities.

Again with reference to Figure 5, the data set-up/first time write procedure illustrated therein according to embodiments of the first aspect of the present invention will be described in detail in the following.

521. The VM 100 with identity VMID running on the TP 105 requests access to a storage resource 108a, i.e., block device or database (denoted as SR), in the provider network being part of a storage domain 109 also comprising servers 108b and 108c. The storage resource reference is denoted SRID. Using SRID as reference, the VM 100 specifically requests a data write to a storage entity 107 denoted x in SR (this can be a block or other storage structure). This request is intercepted or received by the SC 106 and the following procedure applies.

a. The SC 106 sends, protected under key SDA, a request to the TTP 102 for new storage entity keys for entity x, domain A and SRID.

b. The TTP 102 checks, using the key SDA, that the request received above is correct. If so, the TTP 102 fetches the domain key for domain A and may generate a large enough nonce value, NTTP. Next, the TTP 102 optionally uses a suitable pseudorandom function (PRF), PRF (KMA, NTTP) to generate data encryption and integrity protection keys for storage entity x (denoted KE_X and KIx respectively). Next the TTP 102 generates a token, T, consisting of NTTP, the domain ID A and SRID. Furthermore, the TTP 102 uses an internal integrity key to calculate an integrity check value, over token T, MAC(T).

c. Next, the TTP 102 sends T, MAC(T) and the keys KEx and KIx to the SC 106. This message is confidentiality and integrity protected using SDA.

d. The SC 106 receives the message generated in step c and decrypts T, MAC(T) and the KE_X and KI_X keys and associates them with domain A and VMID.

522. The SC 106 stores T and MAC(T) received from the TTP 102 in step d above as part of storage metadata for the new storage entity x in SR 108a.

523. The SC 106 uses keys KE_X and KI_X to confidentiality and/or integrity protect the data stored in storage entity x in SR 108a.

In terms of performance and efficiency, it is advisable that the storage entity unit is selected so that the communication frequency between the SC 106 and

TTP 102 and SC activities does not incur a larger performance penalty than what is acceptable by the involved parties. Also, the storage entity unit should be selected so that integrity protection meta-data does not consume a larger portion of storage than what is acceptable by the involved parties.

Figure 7 illustrates the system of Figure 5, but where a consecutive data write and read procedure is undertaken, i.e. after the set-up procedure illustrated in the flowchart of Figure 6 with reference to steps S301-S307, and in greater detail in steps S2ia-d, S22 and S23 of Figure 5, has been undertaken. The consecutive data write/read procedure according to embodiments of the first aspect of the present invention is described in detail in the following.

S31. The VM 100, which maybe identified to the TP 105 by the VM identifier VMID, requests, e.g., by using the SRID reference, to write data to or read data from the entity x (denoted 107) in SR 108a being part of a storage domain 109 also comprising servers 108b and 108c. This request is intercepted or received by the SC 106 and the following procedure applies.

a. The SC 106 checks, using VMID, if the required integrity and confidentiality keys needed to integrity protect and/or encrypt (write) or verify and/or decrypt (read) the requested storage entity x are already cached in a memory 116 of the SC 106. If that is the case, it proceeds to step 32 below. Otherwise, it locates T and MAC(T) (used to protect x) in SR storage metadata and continues to step b.

b. The SC 106 sends T and MAC(T) to the TTP 102, along with a request for the data entity keys for x, domain A and SRID. All is protected under key SDA.

c. The TTP 102 checks, using key SDA, that the request received above is correct. If so, the TTP 102 verifies the received token T (using its own internal MAC key). If the token is valid, the TTP 102 checks that the DID and SRID contained in T match the DID and SRID indicated by the SC 106. If all the above verifications are successful, The TTP 102 uses the KMA domain key and optionally the NTTP nonce in T to derive the KE_X and KI_X keys.

d. Next, the TTP 102 sends the keys KE_X and KI_X to the SC 106; the keys being confidentiality and integrity protected using SDA.

e. The SC 106 receives and decrypts the encrypted keys.

S32. The SC 106 uses keys KE_X and KI_X to encrypt and/or integrity protect (write) or decrypt and/ or integrity check (read) data at storage entity x.

Thus, with the procedure illustrated in Figure 7, secure data writes/reads can advantageously be made to/from the storage entity x.

Figure 8a shows a secure component 106 according to an embodiment of the first aspect of the present invention. The secure component 106 comprises receiving means 130 adapted to receive, from a communicating party, a request to store data in one of a plurality of storage entities of a storage domain, and encrypting means 131 adapted to encrypt a request for cryptographic keys associated with said one of a plurality of storage entities with a session key of the storage domain to protect confidentiality and authenticity of the request. The secure component 106 further comprises sending means 132 adapted to send, to a trusted third party 102, the encrypted request for cryptographic keys, and the receiving means 130 further being adapted to receive, from the trusted third party, cryptographic keys to protect confidentiality and integrity of the data to be stored, which cryptographic keys have been encrypted with the session key of the storage domain. Moreover, the secure component 106 comprises decrypting means 133 adapted to decrypt the encrypted cryptographic keys, associating means 134 adapted to associate the received cryptographic keys with the storage domain and the communicating party. The encrypting means 131 is further adapted to encrypt the data with the cryptographic keys and store the encrypted data in said one of a plurality of storage entities. The receiving means 130 and sending means 132 may comprise a communications interface for receiving and providing information to other parties. The receiving means 130, encrypting means 131, sending means 132, decrypting means 133 and associating means 134 may (in analogy with the description given in

connection to Figure 5) be implemented by a processor embodied in the form of one or more microprocessors arranged to execute a computer program downloaded to a suitable storage medium associated with the microprocessor, such as a RAM, a Flash memory or a hard disk drive. The receiving means 130 and the sending means 132 may comprise one or more transmitters and/or receivers and/or transceivers, comprising analogue and digital components and a suitable number of antennae for radio communication.

Figure 8b shows a secure domain manager 101 according to an embodiment of the second aspect of the present invention. The secure domain manager 101 comprises creating means 140 adapted to create a launch package comprising an identifier for a storage domain, and an assertion that the secure domain manager 101 is authorized to associate a communicating party with the storage domain, and further comprises signing means 141 adapted to digitally sign the launch package. Moreover, the secure domain manager 101 comprises acquiring means 142 adapted to acquire a certificate from a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch package, and further comprises submitting means 143 adapted to submit the digitally signed launch package and the certificate to a trusted third party for verification, such that the communicating party subsequently can be securely associated with the storage domain. The submitting means 142 may comprise a communications interface providing information to (and possibly receiving information from) other parties. The creating means 140, signing means 141, acquiring means 142 and submitting means 143 may (in analogy with the description given in connection to Figure 1) be implemented by a processor embodied in the form of one or more microprocessors arranged to execute a computer program downloaded to a suitable storage medium associated with the microprocessor, such as a RAM, a Flash memory or a hard disk drive. The submitting means 140 may comprise one or more transmitters and/ or receivers and/or transceivers, comprising analogue and digital components and a suitable number of antennae for radio communication.

Figure 8c shows a trusted third party 102 according to an embodiment of the third aspect of the present invention. The trusted third party 102 comprises receiving means 150 adapted to receive a digitally signed launch package comprising an identifier for a storage domain, and an assertion that a secure domain manager is authorized to associate a communicating party with the storage domain. The receiving means 150 is further adapted to receive a certificate issued by a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch package. Further, the trusted third party 102 comprises verifying means 151 adapted to verify the assertion to ensure that the secure domain manager is authorized to associate the communicating party with the storage domain, such that the communicating party subsequently can be securely associated with the storage domain. The receiving means 150 may comprise a communications interface receiving information from (and possibly providing information to) other parties. The receiving means 150 and verifying means 151 may (in analogy with the description given in connection to Figure 1) be implemented by a processor embodied in the form of one or more microprocessors arranged to execute a computer program downloaded to a suitable storage medium associated with the microprocessor, such as a RAM, a Flash memory or a hard disk drive. The receiving means 150 may comprise one or more receivers and/ or transmitters and/ or transceivers, comprising analogue and digital components and a suitable number of antennae for radio communication.

The invention has mainly been described above with reference to a few embodiments. However, as is readily appreciated by a person skilled in the art, other embodiments than the ones disclosed above are equally possible within the scope of the invention, as defined by the appended patent claims.

CLAIMS

1. A method at a secure component of protecting data in a storage entity, comprising:

receiving (S301), from a communicating party, a request to store data in one of a plurality of storage entities of a storage domain;

encrypting (S302) a request for cryptographic keys associated with said one of a plurality of storage entities with a session key of the storage domain to protect confidentiality and authenticity of the request;

sending (S303), to a trusted third party, the encrypted request for cryptographic keys;

receiving (S304), from the trusted third party, cryptographic keys to protect confidentiality and integrity of the data to be stored, which cryptographic keys have been encrypted with the session key of the storage domain;

decrypting (S305) the encrypted cryptographic keys;

associating (S306) the received cryptographic keys with the storage domain and the communicating party;

encrypting (S307) the data with the cryptographic keys and storing the encrypted data in said one of a plurality of storage entities.

2. The method of claim 1, wherein the cryptographic keys for protecting confidentiality and integrity of the data to be stored are based on a master domain key.

3. The method of claim 2, wherein the cryptographic keys for protecting confidentiality and integrity of the data to be stored are further based on a nonce.

4. The method of claim 3, further comprising the step of:

receiving, from the trusted third party, a token based on said nonce, and an integrity check value for said token.

5. The method of claim 4, wherein the received token and integrity check value are encrypted with the session key of the storage domain.

6. The method of claim 5, further comprising the steps of:

decrypting the encrypted token and integrity check value; and

associating the token and integrity check value with the storage domain and the communicating party.

7. The method of any one of claims 1-6, further comprising:

receiving, from the communicating party, a request to read the encrypted data in said one of a plurality of storage entities of the storage domain;

fetching, from a local cache, the cryptographic keys associated with said one of a plurality of storage entities; and

decrypting the encrypted data with the cryptographic keys associated with said one of a plurality of storage entities.

8. The method of claim 7, further comprising:

sending, to the trusted third party, in case the cryptographic keys are not in the local cache, the token and integrity check value along with the request to read the encrypted data in said one of a plurality of storage entities of the storage domain, encrypted with the session key of the storage domain;

receiving, from the trusted third party, the cryptographic keys associated with said one of a plurality of storage entities if the trusted third party verifies correctness of the token, said cryptographic keys being encrypted with the session key of the storage domain; and

decrypting the encrypted cryptographic keys.

9. The method of any one of claims 1-6, further comprising:

receiving, from the communicating party, a request to store new data in said one of a plurality of storage entities of the storage domain;

fetching, from a local cache, the cryptographic keys associated with said one of a plurality of storage entities; and

encrypting the new data with the cryptographic keys associated with said one of a plurality of storage entities and storing the encrypted new data in said one of a plurality of storage entities.

10. The method of claim 9, further comprising:

sending, to the trusted third party, in case the cryptographic keys are not in the local cache, the token and integrity check value along with the a request to store new data in said one of a plurality of storage entities of the storage domain, encrypted with the session key of the storage domain;

receiving, from the trusted third party, the cryptographic keys associated with said one of a plurality of storage entities if the trusted third party verifies correctness of the token, said cryptographic keys being encrypted with the session key of the storage domain; and

decrypting the encrypted cryptographic keys.

11. The method of any one of the preceding claims, wherein the communicating party is a virtual machine running on a trusted platform.

12. The method of any one of the preceding claims, further comprising:

sending a storage entity identifier to the communicating party to be used for identifying a storage entity associated with said storage entity identifier.

13. A method at a secure domain manager of securely associating a communicating party with a storage domain, comprising:

creating (S101) a launch package comprising an identifier for the storage domain, and an assertion that the secure domain manager is authorized to associate the communicating party with said storage domain;

digitally signing (S102) the launch package;

acquiring (S103) a certificate from a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch package;

submitting (S104) the digitally signed launch package and the certificate to a trusted third party for verification, such that the communicating party subsequently can be securely associated with the storage domain.

14. The method of claim 13, wherein the launch package further is arranged to comprise:

a nonce cryptographically protected by a key of the trusted third party.

15. The method of claims 13 or 14, further comprising:

receiving the identifier for the storage domain from the communicating party.

16. A method at a trusted third party to verify correctness of a launch package created by a secure domain manager to securely associate a communicating party with a storage domain, comprising:

receiving (S201) a digitally signed launch package comprising an identifier for the storage domain, and an assertion that the secure domain manager is authorized to associate the communicating party with said storage domain;

receiving (S202) a certificate issued by a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch package; and

verifying (S203) the assertion to ensure that the secure domain manager is authorized to associate the communicating party with said storage domain, such that the communicating party subsequently can be securely associated with the storage domain.

17. The method of claim 16, further comprising:

generating (S204) a session domain key for the storage domain and a trusted platform on which the communicating party is to reside;

cryptographically protecting (S205) the storage domain identifier and the session domain key with a secret key of the trusted platform;

sending (S206) the cryptographically protected storage domain identifier and the session domain key to the trusted platform, wherein the trusted platform can make available to a secure component the session domain key and securely associate the communicating party with the storage domain.

18. The method of claims 16 or 17, wherein the received digitally signed launch package further comprises a nonce cryptographically protected by a key of the trusted third party, and the method further comprising:

decrypting the cryptographically protected nonce.

19. The method of any one of claims 16-18, the method further comprising:

acquiring an identifier of the communicating party and including the identifier of the communicating party in the launch package.

20. A computer program (123) comprising computer-executable instructions for causing a device (106) to perform at least parts of the steps recited in any one of claims 1-12 when the computer-executable instructions are executed on a processor (121) included in the device.

21. A computer program product comprising a computer readable medium (122), the computer readable medium having the computer program (123) according to claim 20 embodied therein.

22. A computer program (117) comprising computer-executable instructions for causing a device (101) to perform at least parts of the steps recited in any one of claims 13-15 when the computer-executable instructions are executed on a processor (115) included in the device.

23. A computer program product comprising a computer readable medium (116), the computer readable medium having the computer program (117) according to claim 22 embodied therein.

24. A computer program (120) comprising computer-executable instructions for causing a device (102) to perform at least parts of the steps recited in any one of claims 16-19 when the computer-executable instructions are executed on a processor (118) included in the device.

25. A computer program product comprising a computer readable medium (119), the computer readable medium having the computer program (120) according to claim 24 embodied therein.

26. A secure component (106) for protecting data in a storage entity comprising: a processor (121) and a memory (122), said memory containing instructions executable by said processor, whereby said secure component is operative to:

receive, from a communicating party (100), a request to store data in one (107) of a plurality of storage entities of a storage domain (109);

encrypt a request for cryptographic keys associated with said one of a plurality of storage entities with a session key of the storage domain to protect confidentiality and authenticity of the request;

send, to a trusted third party (102), the encrypted request for cryptographic keys;

receive, from the trusted third party (102), cryptographic keys to protect confidentiality and integrity of the data to be stored, which cryptographic keys have been encrypted with the session key of the storage domain;

decrypt the encrypted cryptographic keys;

associate the received cryptographic keys with the storage domain and the communicating party (100);

encrypt the data with the cryptographic keys and storing the encrypted data in said one (107) of a plurality of storage entities.

27. The secure component (106) of claim 26, wherein the cryptographic keys for protecting confidentiality and integrity of the data to be stored are based on a master domain key.

28. The secure component (106) of claim 27, wherein the cryptographic keys for protecting confidentiality and integrity of the data to be stored are further based on a nonce.

29. The secure component (106) of claim 28, further being operative to:

receive, from the trusted third party (102), a token based on said nonce, and an integrity check value for said token.

30. The secure component (106) of claim 29, wherein the received token and integrity check value is encrypted with the session key of the storage domain (109).

31. The secure component (106) of claim 30, further being operative to:

decrypt the encrypted token and integrity check value;

associate the token and integrity check value with the storage domain (109) and the communicating party (100).

32. The secure component (106) of any one of claims 26-31, further being operative to:

receive, from the communicating party (100), a request to read the encrypted data in said one (107) of a plurality of storage entities of the storage domain (109);

fetch, from a local cache (122), the cryptographic keys associated with said one (107) of a plurality of storage entities; and

decrypt the encrypted data with the cryptographic keys associated with said one (107) of a plurality of storage entities.

33. The secure component (106) of claim 32, further being operative to:

send, to the trusted third party (102), in case the cryptographic keys are not in the local cache (122), the token and integrity check value along with the request to read the encrypted data in said one (107) of a plurality of storage entities of the storage domain (109), encrypted with the session key of the storage domain;

receive, from the trusted third party (102), the cryptographic keys associated with said one (107) of a plurality of storage entities if the trusted third party (102) verifies correctness of the token, said cryptographic keys being encrypted with the session key of the storage domain; and

decrypt the encrypted cryptographic keys.

34. The secure component (106) of any one of claims 26-32, further being operative to:

receive, from the communicating party (100), a request to store new

data in said one (107) of a plurality of storage entities of the storage domain (109);

fetch, from a local cache (122), the cryptographic keys associated with said one (107) of a plurality of storage entities; and

encrypt the new data with the cryptographic keys associated with said one (107) of a plurality of storage entities and storing the encrypted new data in said one (107) of a plurality of storage entities.

35. The secure component (106) of claim 34, further being operative to:

send, to the trusted third party (102), in case the cryptographic keys are not in the local cache (122), the token and integrity check value along with the a request to store new data in said one (107) of a plurality of storage entities of the storage domain (109), encrypted with the session key of the storage domain;

receive, from the trusted third party (102), the cryptographic keys associated with said one (107) of a plurality of storage entities if the trusted third party (102) verifies correctness of the token, said cryptographic keys being encrypted with the session key of the storage domain; and

decrypt the encrypted cryptographic keys.

36. The secure component (106) of any one of claims 26-35, wherein the communicating party (100) is a virtual machine running on a trusted platform.

37. The secure component (106) of any one of claims 26-36, further being operative to:

send a storage entity identifier to the communicating party (100) to be used for identifying a storage entity associated with said storage entity identifier.

38. A secure domain manager (101) for securely associating a communicating party (100) with a storage domain (109) comprising: a processor (115) and a memory (116), said memory containing instructions executable by said processor, whereby said secure domain manager is

operative to:

create a launch package comprising an identifier for the storage domain, and an assertion that the secure domain manager (101) is authorized to associate the communicating party (100) with said storage domain;

digitally sign the launch package;

acquire a certificate from a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch package;

submitting the digitally signed launch package and the certificate to a trusted third party (102) for verification, such that the communicating party (100) subsequently can be securely associated with the storage domain.

39. The secure domain manager (101) of claim 38, wherein the launch package further is arranged to comprise:

a nonce cryptographically protected by a key of the trusted third party (102).

40. The secure domain manager (101) of claims 38 or 39, further being operative to:

receive the identifier for the storage domain (109) from the communicating party (100).

41. A trusted third party (102) for verifying correctness of a launch package created by a secure domain manager (101) to securely associate a communicating party (100) with a storage domain (109), the trusted third party comprising: a processor (118) and a memory (119), said memory containing instructions executable by said processor, whereby said trusted third party is operative to:

receive a digitally signed launch package comprising an identifier for the storage domain, and an assertion that the secure domain manager (101) is authorized to associate the communicating party (100) with said storage domain;

receive a certificate issued by a trusted certificate authority for certifying a verification key corresponding to that used for digitally signing the launch

package; and

verify the assertion to ensure that the secure domain manager (101) is authorized to associate the communicating party (100) with said storage domain, such that the communicating party (100) subsequently can be securely associated with the storage domain.

42. The trusted third party (102) of claim 41, further being operative to:

generate a session domain key for the storage domain (109) and a trusted platform (105) on which the communicating party (100) is to reside;

cryptographically protect the storage domain identifier and the session domain key with a secret key of the trusted platform (105);

send the cryptographically protected storage domain identifier and the session domain key to the trusted platform (105), wherein the trusted platform can make available to a secure component (106) the session domain key and securely associate the communicating party (100) with the storage domain.

43. The trusted third party (102) of claims 41 or 42, wherein the received digitally signed launch package further comprises a nonce cryptographically protected by a key of the trusted third party (102), the trusted third party further being operative to:

decrypt the cryptographically protected nonce.

44. The trusted third party (102) of any one of claims 41-43, further being operative to:

acquire an identifier of the communicating party (100) and include the identifier of the communicating party (100) in the launch package.

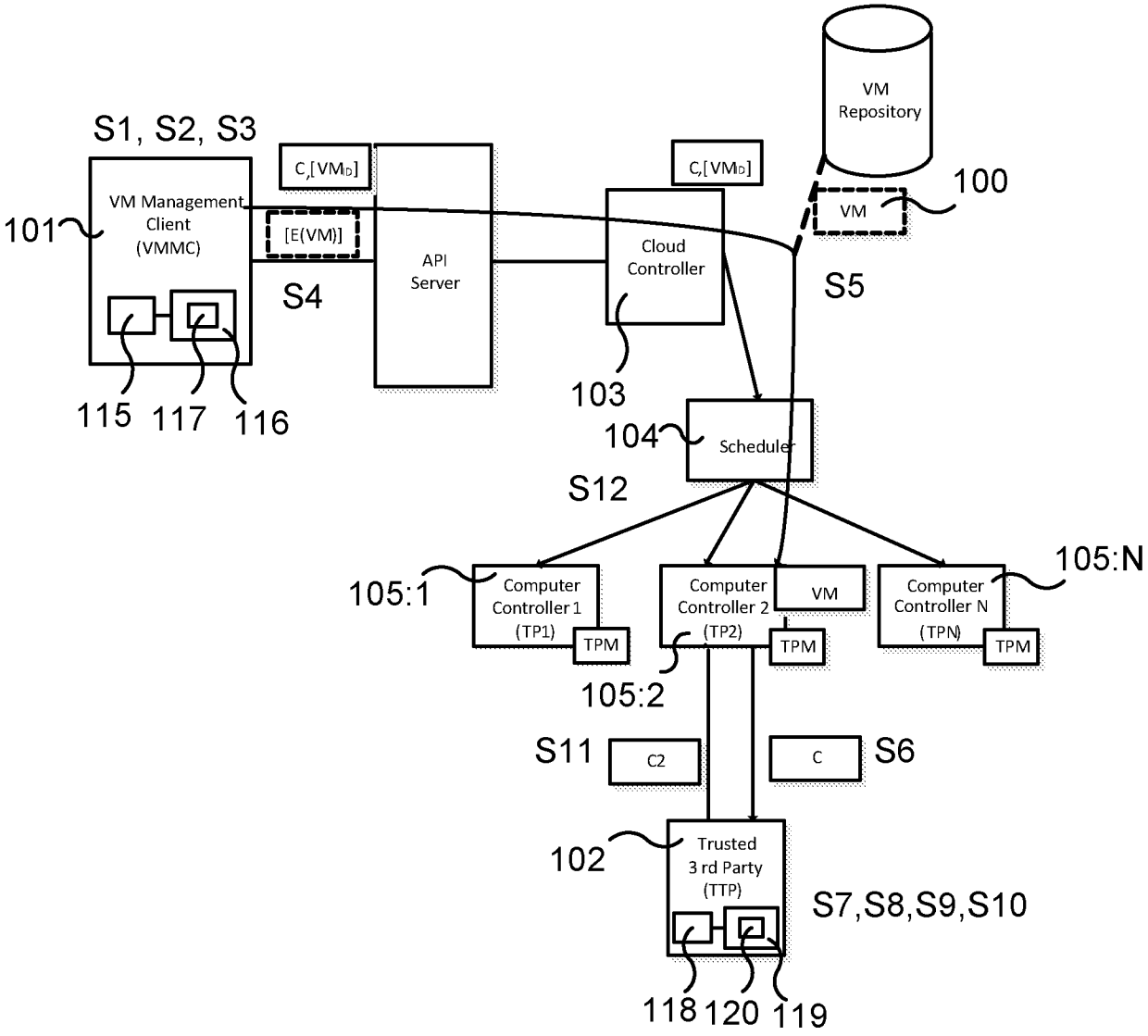
S1, S2, S3
VM Management Client (VMMC)
101
C,[VMID]
[E(VM)]
S4
API Server
C,[VMID]
Cloud Controller
103
VM Repository
100
VM
S5
115
117
116
104
Scheduler
S12
105:1
Computer Controller 1 (TP1)
TPM
Computer Controller 2 (TP2)
VM
TPM
105:2
Computer Controller N (TPN)
TPM
105:N
S11
C2
C
S6
102
Trusted 3 rd Party (TTP)
S7,S8,S9,S10
118
120
119

Fig. 1

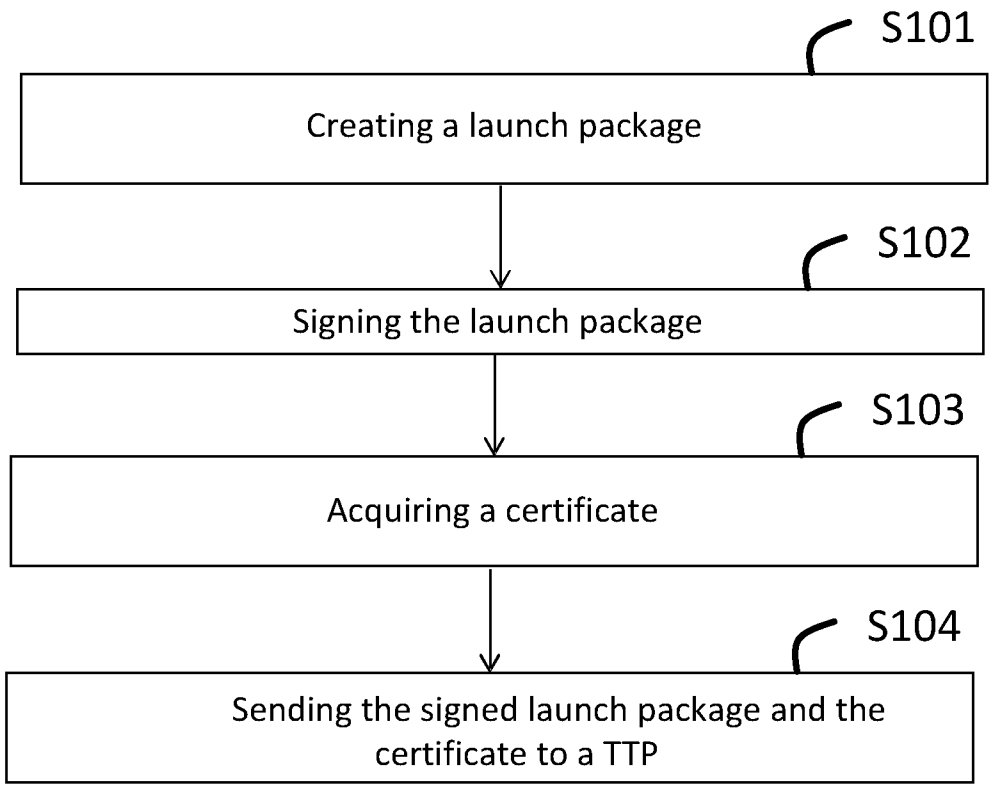
S101
Creating a launch package
S102
Signing the launch package
S103
Acquiring a certificate
S104
Sending the signed launch package and the certificate to a TTP

Fig. 2

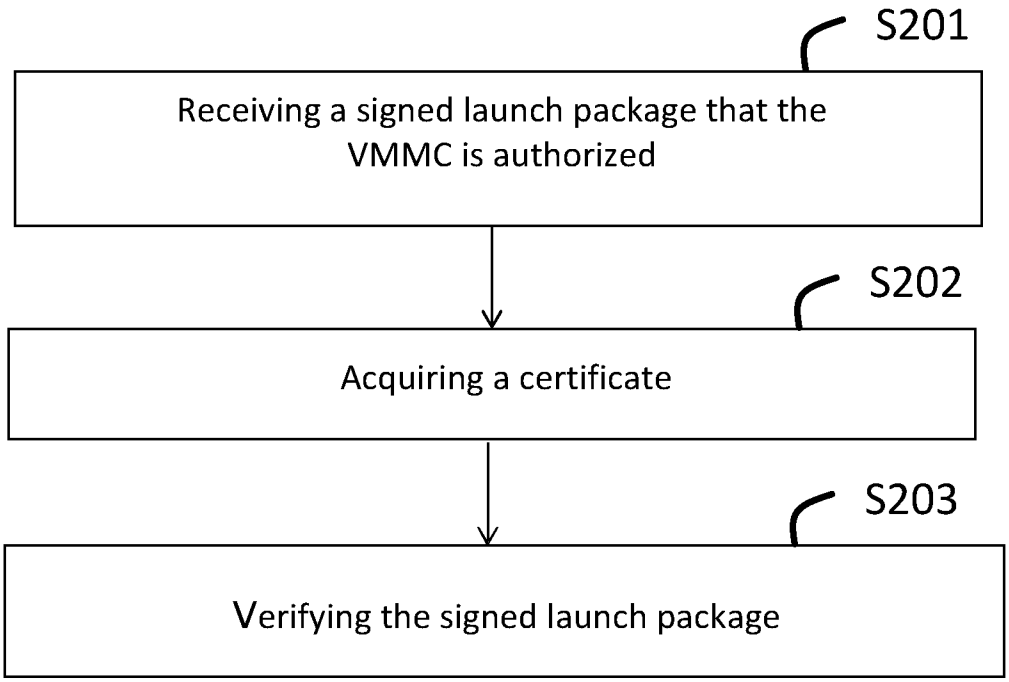
S201
Receiving a signed launch package that the VMMC is authorized
S202
Acquiring a certificate
S203
Verifying the signed launch package

Fig. 3

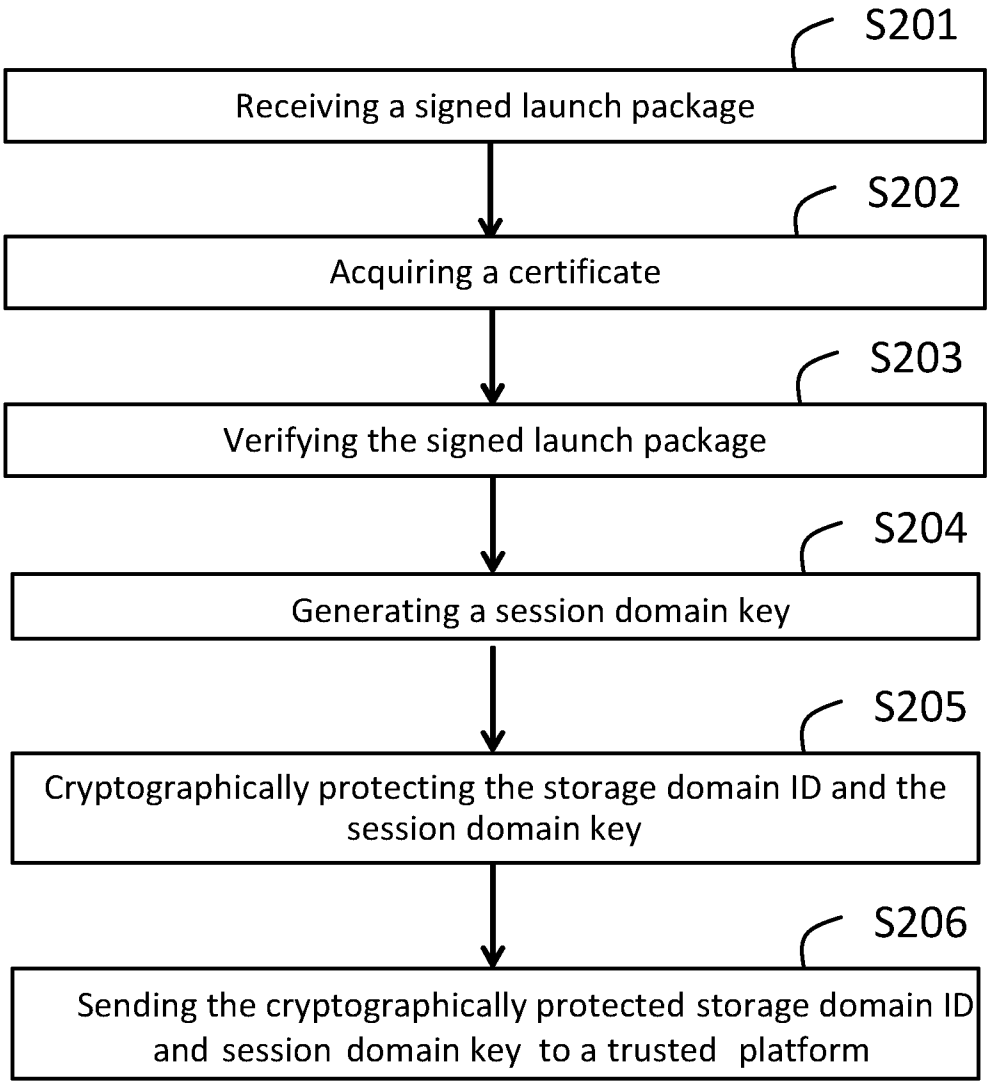
S201
Receiving a signed launch package
S202
Acquiring a certificate
S203
Verifying the signed launch package
S204
Generating a session domain key
S205
Cryptographically protecting the storage domain ID and the session domain key
S206
Sending the cryptographically protected storage domain ID and session domain key to a trusted platform

Fig. 4

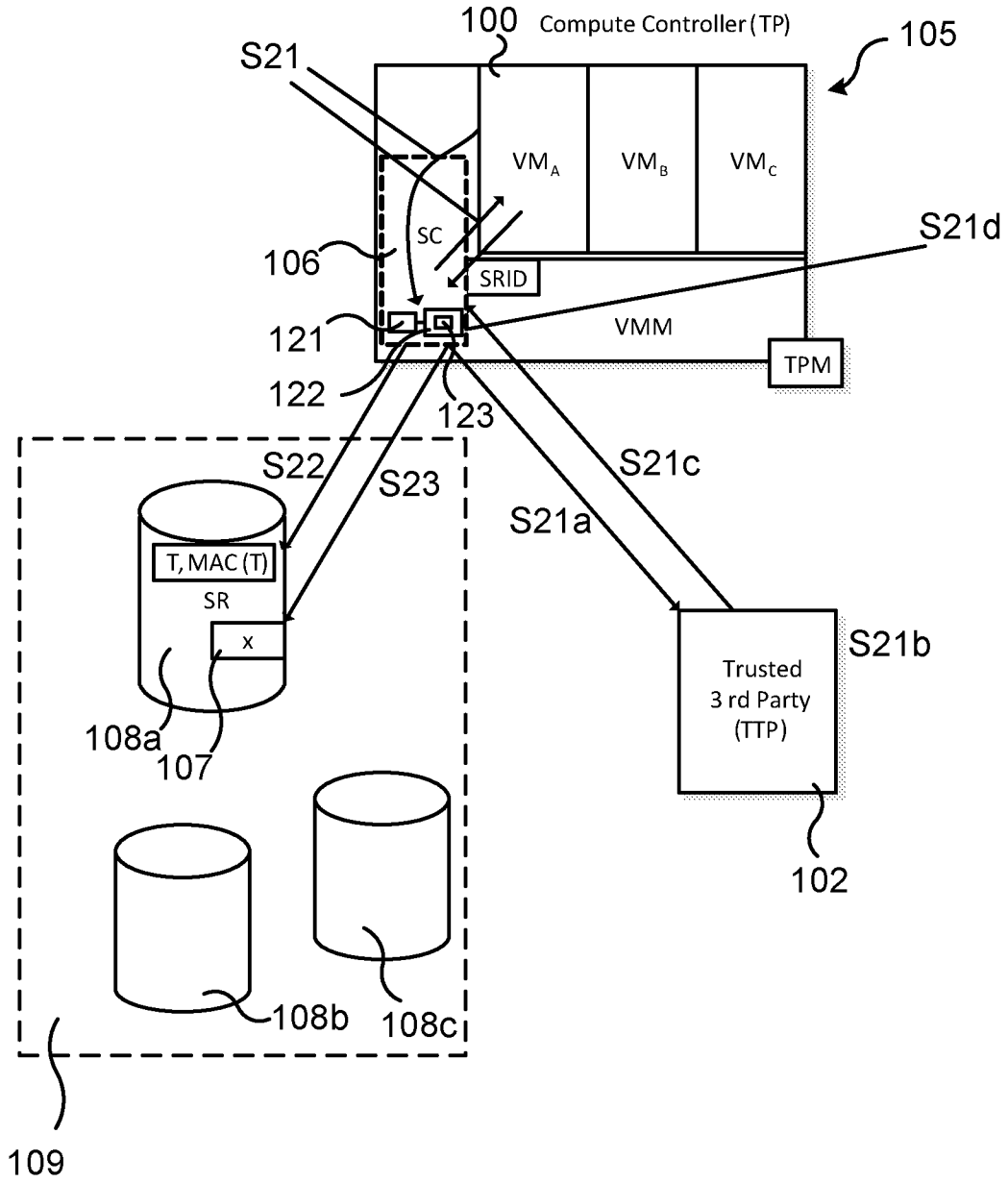
100
Compute Controller (TP)
105
S21
VM A
VM B
VM C
SC
106
SRID
S21d
VMM
121
TPM
122
123
S22
S23
S21c
S21a
T, MAC (T)
SR
x
S21b
Trusted
3 rd Party
(TTP)
108a
107
102
108b
108c
109

Fig. 5

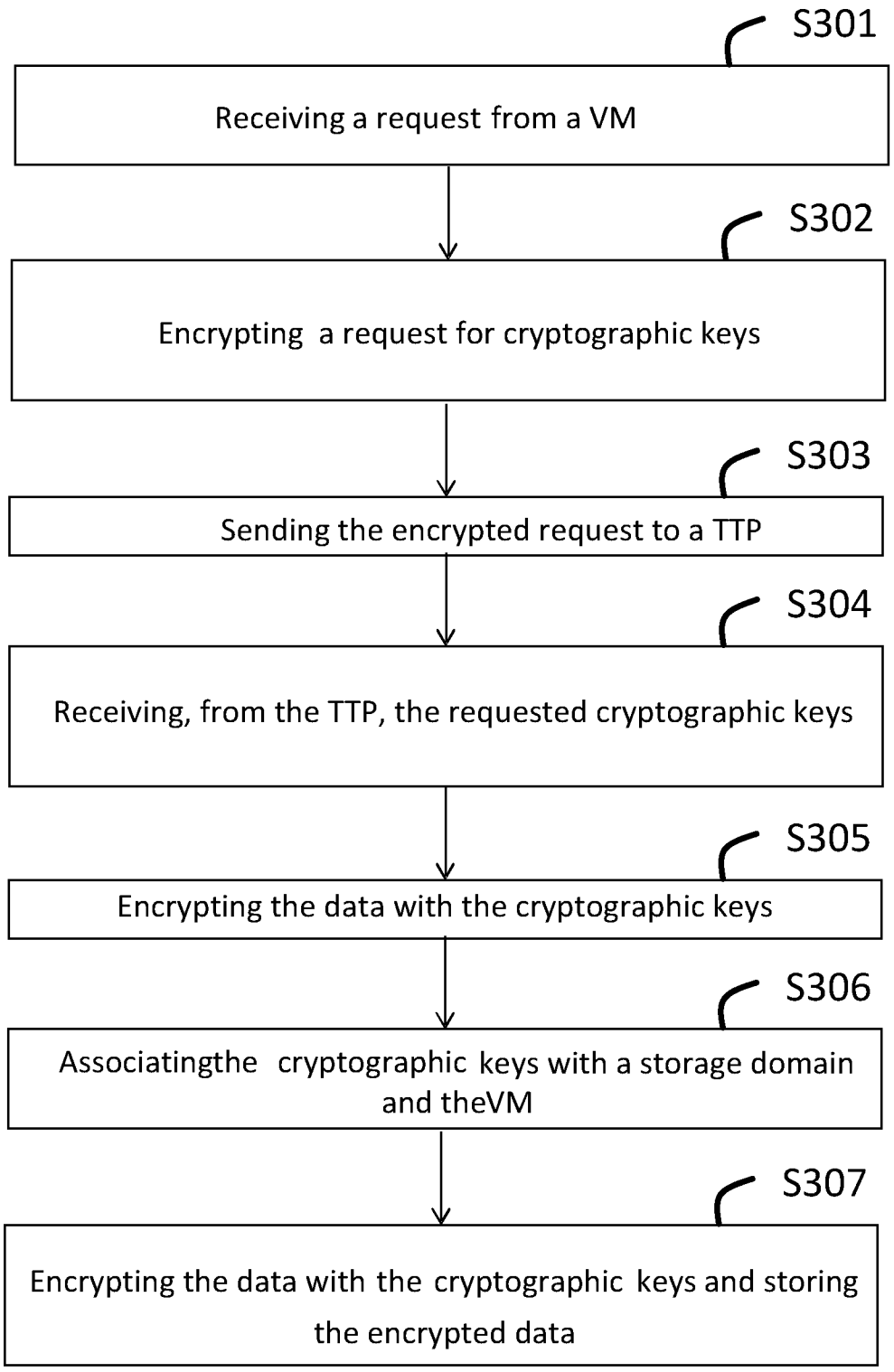
S301
Receiving a request from a VM
S302
Encrypting a request for cryptographic keys
S303
Sending the encrypted request to a TTP
S304
Receiving, from the TTP, the requested cryptographic keys
S305
Encrypting the data with the cryptographic keys
S306
Associatingthe cryptographic keys with a storage domain and theVM
S307
Encrypting the data with the cryptographic keys and storing the encrypted data

Fig. 6

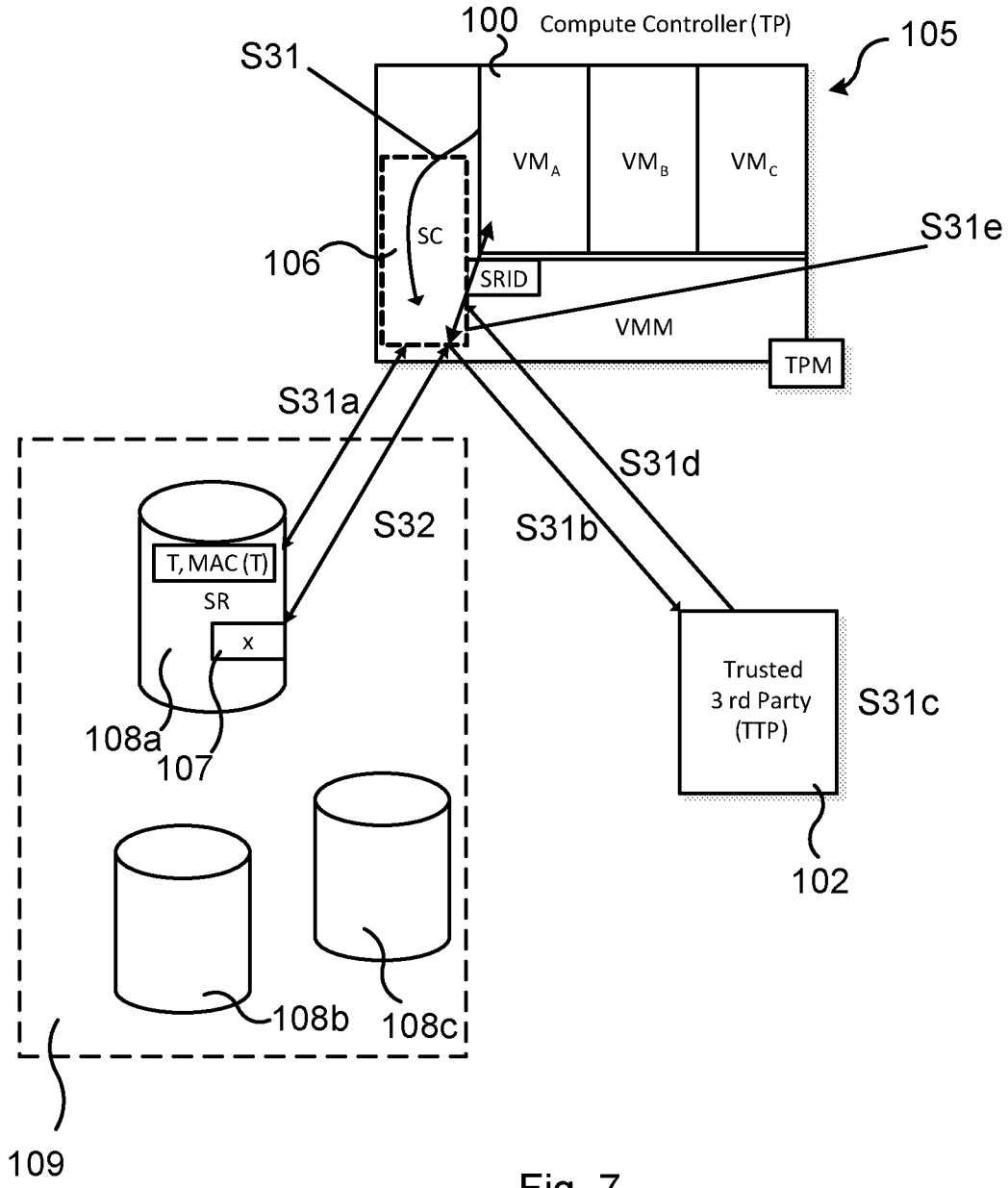
100
Compute Controller (TP)
105
S31
VM A
VM B
VM C
SC
106
S31e
SRID
VMM
TPM
S31a
S31d
S32
S31b
T, MAC (T)
SR
x
Trusted
3 rd Party
(TTP)
S31c
108a
107
102
108b
108c
109

Fig. 7

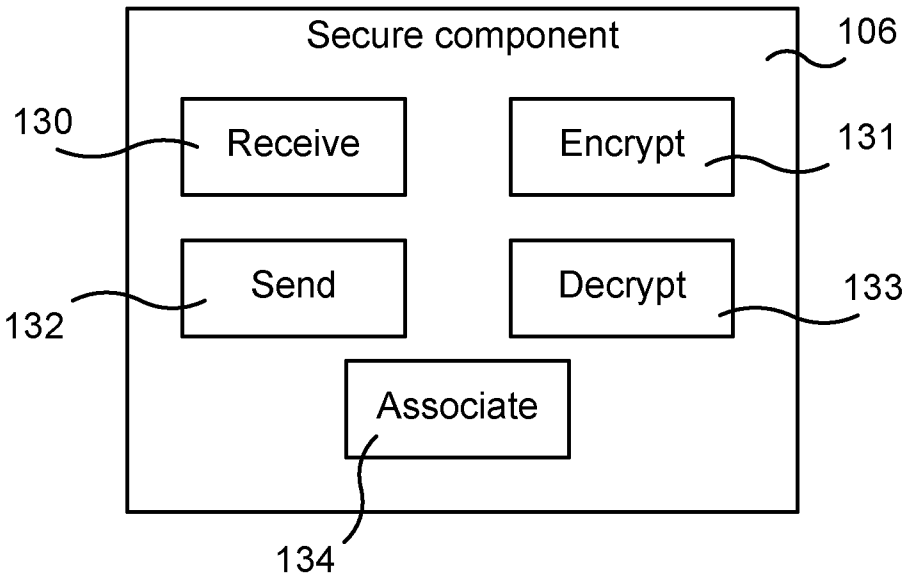

Fig. 8a

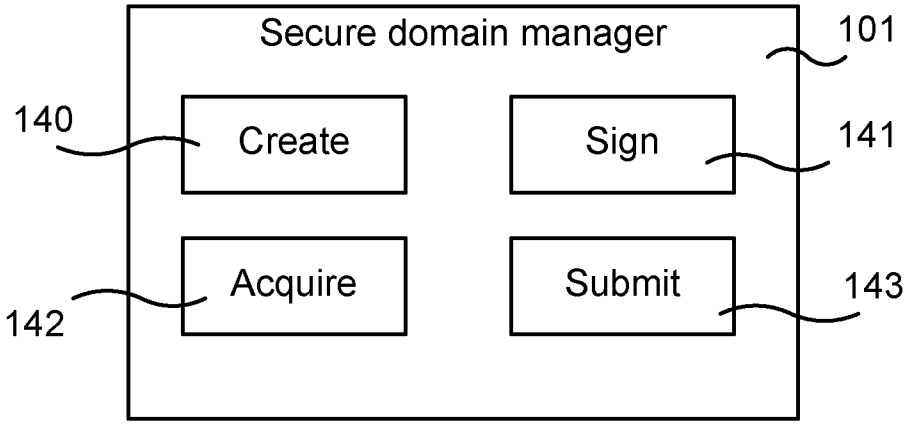

Fig. 8b

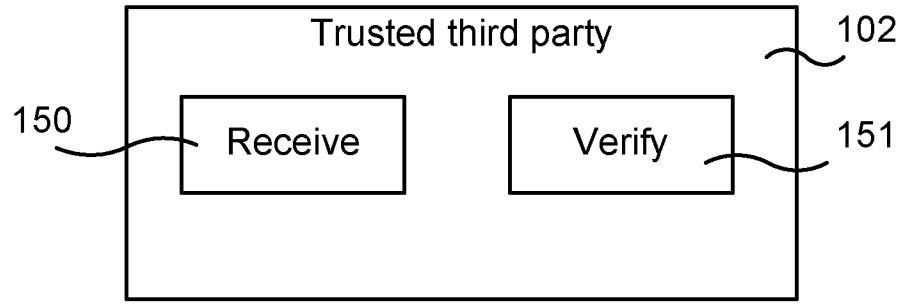

Fig. 8c

INTERNATIONAL SEARCH REPORT

International application No
PCT/SE2013/051299

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L9/08 H04L9/32
ADD.

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)
H04L G06F

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)
EPO-Internal , WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2013/042106 Al (PERSAUD ANDREW [US] ET AL) 14 February 2013 (2013-02-14) paragraph [0012] - paragraph [0021] paragraph [0029] paragraph [0043] - paragraph [0072] paragraph [0077] - paragraph [0079] paragraph [0082] - paragraph [0083] -----	1-12 ,20, 21,26-37
A	US 8 170 213 Bl (HARW00D JACK [US] ET AL) 1 May 2012 (2012-05-01) col umn 1, lines 15-20 col umn 2, line 11 - col umn 5, line 21 -----	1-12 ,20, 21,26-37

-/- .

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* 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" documentwhich may throw doubts on priority claim(s) orwhich 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
10 Apri l 2014	27/05/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Apostol escu, Radu

4

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE2013/051299

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No
PCT/SE2013/051299

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2012/148324 A1 (ERICSSON TELEFON AB L M [SE]; GEHRMANN CHRISTIAN [SE] ; MEHES ANDRAS [S) 1 November 2012 (2012-11-01)	13-19, 22-25, 38-44
A	paragraph [0041] - paragraph [0062] paragraph [0068] - paragraph [0084] paragraph [0065] - paragraph [0084]	1-12,20, 21,26-37
A	US 2011/302415 A1 (AHMAD IRFAN [US] ET AL) 8 December 2011 (2011-12-08) paragraph [0015] - paragraph [0043]	1-12,20, 21,26-37
Y	US 2011/302400 A1 (MAIN0 FAB10 R [US] ET AL) 8 December 2011 (2011-12-08) paragraph [0019] - paragraph [0035] paragraph [0042] - paragraph [0070]	13-19, 22-25, 38-44
A	US 2013/097296 A1 (GEHRMANN CHRISTIAN [SE] ET AL) 18 April 2013 (2013-04-18) paragraph [0021] - paragraph [0040] paragraph [0055] - paragraph [0058]	13-19, 22-25, 38-44
A	US 2004/064729 A1 (YELLEPEDDY KRISHNA KISH0RE [US]) 1 April 2004 (2004-04-01) abstract paragraph [0052] - paragraph [0068]	13-19, 22-25, 38-44

4

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/SE2013/051299

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013042106 A1	14-02-2013	NONE	
US 8170213 B1	01-05-2012	NONE	
WO 2012148324 A1	01-11-2012	EP 2702724 A1	05-03-2014
		US 2014032920 A1	30-01-2014
		WO 2012148324 A1	01-11-2012
US 2011302415 A1	08-12-2011	AU 2011261831 A1	25-10-2012
		EP 2577539 A1	10-04-2013
		JP 2013528872 A	11-07-2013
		US 2011302415 A1	08-12-2011
		WO 2011152910 A1	08-12-2011
US 2011302400 A1	08-12-2011	CN 103069428 A	24-04-2013
		EP 2577543 A1	10-04-2013
		US 2011302400 A1	08-12-2011
		WO 2011156261 A1	15-12-2011
US 2013097296 A1	18-04-2013	US 2013097296 A1	18-04-2013
		WO 2013057682 A1	25-04-2013
US 2004064729 A1	01-04-2004	CN 1487423 A	07-04-2004
		TW 1225353 B	11-12-2004
		US 2004064729 A1	01-04-2004
		US 2007174456 A1	26-07-2007

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-12, 20, 21, 26-37

   Method for protecting data in a storage entity

   ---

2. claims: 13-19, 22-25, 38-44

   Method for securely associating a communicating party with a storage domain

   ---