

Apparatus And Method For Preventing Theft of Algorithms From VPN Software During Use and/or Storage

Field

5

The present invention relates to an apparatus and a method for preventing algorithm theft from Virtual Private Network (VPN) Software during use and/or storage, in particular, using Trusted Execution Environments (TEEs) to prevent such theft. The apparatus and method may be applied in all computer systems that use VPN Software to encrypt/decrypt data payloads and provide integrity protection to data payloads.

10

Background

15

To ensure employees of an organization can communicate with the organization's networks securely, a common practice is to issue devices such as desktop and laptop computers and mobile phones to the employees and install Virtual Private Network (VPN) Software on these devices. The devices may also include VPN Gateways, which are standalone computers that host VPN Software and other networking functions.

20

The VPN Software may be configured to use the organization's proprietary algorithms, which perform functions such as encryption and decryption that are critical to the operation of the VPN Software. These algorithms may enhance the security and/or performance of the communication channels between the devices and the organization's networks.

25

The organization wants to keep these algorithms confidential from the employees and third parties to safeguard its intellectual property. In addition, if malicious actors can find weaknesses in the stolen proprietary algorithms via reverse engineering or analysis, the security of all communications in the organization may be compromised.

30

The organization may try to protect its algorithms with additional levels of security. For example, the algorithms can be encrypted in the devices at rest and decrypted for use in the VPN Software only when an employee supplies a valid password or inserts a valid security smart card.

35

These measures may prevent malicious actors from extracting algorithms from stolen devices, but they are ineffective against malicious employees (insiders or infiltrators) who are legitimate owners of the issued devices and have permission to access the algorithms. In addition, if employees use weak passwords or lose their smart cards, these measures would not be effective.

40

Summary

45

According to an example of the present disclosure, there are provided an apparatus and a method for preventing algorithm theft from Virtual Private Network (VPN) Software during use and/or storage, as claimed in the independent claims. Some optional features are defined in the dependent claims.

Brief Description of the Drawings

50

Fig. 1 illustrates 3 steps of an example of the present disclosure.

Fig. 2 illustrates in more detail step 1 of Fig. 1.

Fig. 3 illustrates in more detail step 2 of Fig. 1.

55

Fig. 4 illustrates in more detail step 3 of Fig. 1.

Fig. 5 illustrates in more detail step 4 of Fig. 1.

Detailed Description

Some terms used in the present disclosure are explained below.

Algorithm – An algorithm refers to a set of instructions designed to perform specific task. For example, cryptographic algorithms perform encryption, key derivation, authenticated encryption, message authentication, generate and validate digital signatures, and elliptic curve cryptography.

Issuer - An issuer refers to any organization, for instance, a company. The issuer is the owner of algorithms and wants the algorithms to be deployed in the devices used by users. The issuer does not trust the users or their devices with the algorithms.

User - A user refers to a member in an organization, for example, an employee in a company. The user has a device that is installed with a VPN Software. The VPN Software is configured with the issuer's algorithms. The user uses the VPN Software to communicate with the issuer or other users securely.

Device – A computing apparatus issued to the user by the issuer. Devices may include desktop and laptop computers and mobile phones to the employees and install Virtual Private Network (VPN) Software on these devices. Devices may also include VPN Gateways, which are standalone computers that host VPN Software and other networking functions.

Trusted Execution Environment (TEE) - A Trusted Execution Environment (TEE) is defined as a tamper-resistant processing environment that runs on a separation kernel. It guarantees the authenticity of the executed code, integrity of runtime states, and the confidentiality of its code, data and runtime states stored on a persistent memory. TEE offer protection against the operating system, hypervisor, privileged users, other applications, and other TEE instances. TEE also supports attestation that proves its trustworthiness to remote entities. Among the most used hardware implementations of TEEs in modern processors are Intel's Software Guard Extensions (SGX) and ARM's TrustZone. It should be appreciated that the present disclosure is not limited to TEE that are enforced in hardware. TEEs that are enforced in software only are also covered.

Intel Software Guard Extensions (SGX) – Introduced in 2015 with Intel's Skylake microarchitecture, Intel SGX is a CPU hardware extension and Intel's latest iteration of trusted computing design that aims to provide secure remote computation using trusted hardware. Intel SGX provides applications with TEEs known as enclaves that are isolated from other processes, guaranteeing the confidentiality and integrity of computations within them. Intel SGX may be used for examples of the present disclosure.

VPN Software - A VPN Software is a software based technology that establishes a secure connection between end-points. Some VPN Software work in the background automatically, while others have front-end interfaces that allow users to interact with and configure them.

Server – A computing apparatus that renders one or more functionalities for other programs or computing apparatuses. In an example of the present disclosure, an issuer's server is a computing apparatus that is dedicated to the fulfilment of the issuer's responsibilities in the work flows for VPN software usage (work flow details are provided below). The issuer's server may have a (physical or virtual) database for storing

information related to the work flows for VPN software usage, such as the TEE identities and public keys belonging to users' devices.

- 5 Intermediate Representation (IR) – An IR is data structure or code used internally by a compiler or a virtual machine to represent source code. An IR is designed to be conducive for further processing, such as conversion into bytecode.

In an example of the present disclosure, there is provided an apparatus using TEEs to provide one or more of the following features:

- 10 • Protect algorithms at rest in an apparatus such as a desktop computer, laptop computer, mobile device including tablet devices, smartphones and VPN Gateways;
 • Protect algorithms in use by a VPN Software;
 • Protect algorithms in transit (For example, during over-the-air update of algorithms and/or software). In another example of the present disclosure, there is provided a
 15 corresponding method performing the processes of the apparatus; and
 • Remote management (updating) of algorithms.
 Such algorithms may include proprietary encryption algorithms and/or proprietary hashing algorithms that improve the performance of the VPN Software.

- 20 Trusted Platform Module (TPM) – A TPM is a secure cryptoprocessor used to establish root of trust in security systems.

With such protections in place, a user is allowed to use the VPN Software with the issuer's algorithms, but the user is prevented from unauthorised extraction of the algorithms. Other
 25 state-of-the-art solutions cannot achieve this.

Conventionally, TEEs have been used in Digital Rights Management (DRM) technologies such as Google's Widevine to protect digitally encoded information (such as media or software) on computers. Generally, TEE is used in DRM technologies to allow users to
 30 view copyrighted contents but not allow them to be cloned. The goals, use cases, and implementation of such DRM technologies and examples of the present invention are different.

Another conventional use of TEEs is illustrated in "EndBox" (i.e. David Goltzsche et al.,
 35 "EndBox: Scalable Middlebox Functions Using Client-Side Trusted Execution", IEEE Explore, Electronic ISBN: 978-1-5386-5596-2, Electronic ISSN: 2158-3927, 23 July 2018). Generally, EndBox is a system that outsources and decentralizes middlebox functions, such as load balancing, intrusion detection, firewalling, and DDoS prevention, to network clients. EndBox employs TEEs to secure the operation of middlebox functions
 40 in untrusted client hardware. The goals, use cases, and implementation of EndBox and examples of the present invention are different.

A third conventional use of TEEs is to build Key Management Services (KMS). Fortanix's Self-Defending Key Management Service employs Intel SGX to build a secure, verifiable
 45 and scalable KMS for the cloud. The goals, use cases, and implementation of such KMS technologies and examples of the present invention are different.

In contrast, examples of the present disclosure employ TEEs to allow users to use VPN Software that are configured with the issuer's algorithms, and yet protect against theft of
 50 such algorithms. As the theft of such algorithms is prevented, the issuer's intellectual property (in respect of the algorithms) is safeguarded. Malicious actors are also prevented from discovering weaknesses in said algorithms via reverse engineering or analysis to compromise the security of communication channels.

In an example of the present disclosure, the example employs Trusted Execution Environment (TEE) technology to augment operations of VPN Software.

Specifically, the example employs TEE technology to protect algorithms on end devices against copying or extraction by the device owners, while allowing VPN Software to use these algorithms to protect device communications.

The example comprises two parts: 1) a novel VPN Software that employs TEE technology to ensure the confidentiality and integrity of algorithms, and 2) a series of work flows for an issuer and users to use the VPN Software and TEE technology to manage and use these algorithms while ensuring their confidentiality and integrity.

For this example, it is assumed that an issuer wants to issue a device such as a desktop or laptop computer, tablet, smartphone and/or VPN Gateway to a user with VPN Software installed. The issuer wants to configure the VPN Software to use proprietary algorithms but the issuer is concerned about revealing them to malicious parties. The issuer also needs to manage the algorithms remotely. It is also assumed that each of the devices is TEE-capable such as comprising a TEE-capable central processing unit (CPU). Each TEE-capable device also has a unique identity, for example, a unique identifier or code.

Part 1: Novel VPN Software architecture

In the first part, a novel architecture is proposed for the VPN Software.

It is proposed that the components of the VPN Software are classified as either “trusted” or “untrusted”.

Components of the VPN Software that must process the issuer’s algorithms unencrypted or perform attestation are classified as “trusted”.

For example, trusted components may include:

- one or more software components for execution of cryptography operations on data using one or more cryptographic algorithms;
- one or more software components for sealing or unsealing of one or more software components; and/or
- one or more software components for the attestation of one or more enclaves in the TEE

Optionally, the trusted components may also include one or more software components to enable updating of the trusted one or more software components.

All other components that do not handle the issuer’s algorithms unencrypted or perform attestation will be generically classified as “untrusted”. For example, they include:

- one or more software components that render the VPN Software’s graphical user interface; and/or
- one or more software components that handle user authentication.

The “trusted” components form the “Trusted Computing Base” of the VPN Software. During runtime, the TEE-capable CPU will provide enclaves that the “trusted” components can run within. The enclaves protect the integrity and confidentiality of the operations that take place within the enclaves.

For the present example, the size of the Trusted Computing Base is kept as small as possible to achieve the following.

- To make the Trusted Computing Base easy to audit and maintain. A large Trusted Computing Base is more susceptible to software flaws that may compromise the algorithms to be protected.
- To have the Trusted Computing Base run within enclaves and not incur significant overheads.

For example, the Trusted Computing Base may be restricted only to one or more of the following mentioned earlier as trusted components and has no other trusted components:

- one or more software components for execution of cryptography operations on data using one or more cryptographic algorithms;
- one or more software components for sealing or unsealing of one or more software components; and/or
- one or more software components for the server to attest the one or more enclaves in the TEE

Part 2: Work Flows for VPN Software Usage

In the second part, with reference to Fig. 1, there is provided a workflow in 4 steps as follows:

1. In a step 1, the issuer generates a unique public/private key pair for each device issued to the users. The private key is securely downloaded to the device, where it is sealed to the device's trusted platform module. The public key is saved in the issuer's database. The public/private key will be used to verify the identity of the device during attestation in a later step. Such attestation will be described in detail later.
2. In a step 2, the issuer converts an algorithm to be protected during use or storage at each issued device into a program and compiles the program into bytecode. Bytecode is a program code that has been compiled from source code into low-level code designed for a software interpreter.
3. In a step 3, each device (e.g. computer) of the devices issued to the users is configured to contact the issuer to download the algorithm bytecode for the VPN Software to be operated on the device.
4. In a step 4, each device is configured to use the algorithm bytecode within the enclaves in the device when the algorithm is to be used with the VPN Software in operation.

Step 1 is performed as an initialization phase. Steps 2, 3 and 4 are performed when a system comprising an issuer's server 203 and a device 201 (issued to the user by the issuer) is initialized and is in operation. In actual operation, there can be more than one of the device 201 configured in the same manner.

Fig. 2 illustrates in more detail step 1 of Fig. 1. Fig. 2 shows a device 201 with a TEE-capable processor 202. Server software 205 of the issuer's server 203 generates a unique public/private key pair 206 for the device 201. The private key is securely downloaded to the device 201, where it is sealed to the trusted platform module (not shown in Fig. 2) of the TEE-capable processor 202. The public/private key pair 206 will be used for attestation in a later step. Such attestation will be described in detail later. The device 201 is installed with a VPN Software that requires one or more of the algorithm bytecode from the issuer to operate.

Before the device 201 may be used, the issuer records the device's CPU identity and public key 206 into an issuer's database 204 that can be accessible to the issuer's server 203. The database 204 will allow the issuer to maintain the list of users of the issued devices (such as the device 201) and their CPU identities. The database 204 will be used

to authenticate requests from devices to download algorithm bytecode from the issuer's server 203 in step 2 of Fig. 1.

Fig. 3 illustrates in more detail step 2 of Fig. 1. The issuer converts algorithm 302 to be subject to data protection during use or storage at the device 201 of Fig. 2 into a source code 304 and compiles the source code 304 into bytecode 306. The issuer's server 203 transfers bytecode 306 to the device 201 in step 3 of Fig. 1.

As an alternative to step 3 of Fig. 1, the issuer's server 203 may instead transfer the source code 304 to the device 201 directly, or compile the source code 304 into an intermediate representation and transfer the intermediate representation to the device 201. The advantage of this alternative approach is that devices of different architectures may share the same source code or intermediate representation. However, under this alternative approach, the devices have to take on the task of compiling the source code or intermediate representation into algorithm bytecode in step 3 of Fig. 1 after they are downloaded from the server 203, or in step 4 of Fig. 1 before the devices use the algorithm bytecode.

In the present example, the process of demonstrating that an enclave has been established in a secure environment and initialized with the VPN Software is referred to as attestation.

Fig. 4 illustrates in more detail step 3 of Fig. 1. Fig. 4 shows the device 201 of Fig. 2 and the issuer's server 203 of Fig. 2. A VPN Software 402 residing at the device 201 is configured to contact the issuer's server 203 in a step A to query whether new bytecodes of an algorithm are available to be downloaded. The VPN Software 402 can be configured to conduct step A periodically.

When contacted and there is a new algorithm bytecode (or bytecodes) to be downloaded to the device 201, the server 203 will send an attestation challenge in a step B to the VPN Software 402. The VPN Software 402 will provision a new enclave 404 in a step C. The enclave 404 will prepare a response to the attestation challenge in a step D, which the VPN Software 402 will forward to the server 203 in a step E. The VPN Software 402 will also unseal a private key from the trusted platform module (not shown in Fig. 4) of the TEE-capable processor 202. As mentioned above, the private key was generated and sealed to the trusted platform module in step 1 of Fig. 1. The private key will be used to digitally sign the response to the attestation challenge in step D. When the server 203 receives the digitally signed response to the attestation challenge in step E, the server 203 will use the corresponding public key stored in the database 204 in step 1 of Fig. 1 to validate the digital signature to verify the identity of the device 201.

Other than to verify the identity of the device 201, another purpose of the attestation is to allow the server 203 to attest or verify that the enclave 404 has been established in a secure environment. The response prepared by the enclave 404 will provide data for the server 203 to make this attestation or verification. Specifically, the server 203 will verify the following through the attestation challenge:

- The enclave 404 is loaded with the VPN Software;
- The state of the enclave 404 has been initialized with the correct parameters of the VPN Software, such as version and operational modes; and
- The authenticity of the enclave 404, for example, checking that the enclave 404 is a real enclave running on the corresponding TEE-capable processor 202 in Fig. 2 and that it is not an imitation, simulation, or replay.

Hence, to enable the attestation or verification, the response prepared by the enclave 404 would contain data relating to the VPN Software such as software name and version, data

indicating that the enclave 404 has been initialized with the correct parameters, and data indicating authenticity of the enclave 404, in addition to the digital signature for the verification of the identity of the device 201.

- 5 If the attestation procedure in steps B to E is successful, the enclave 404 will establish a secure communication channel with the issuer's server 203 at a step F. It is emphasised that any components external of the enclave 404, including the operating system and other unauthorized enclaves on the same device 201, will not be able to eavesdrop on this secure communication channel. Through this secure communication channel, the
10 bytecode (or bytecodes) that the issuer will share with the enclave 404 in a step G is safeguard from any party external to the respective attested enclave 404.

If the attestation procedure in steps B to E is unsuccessful, the updating process is terminated.

- 15 After the secure communication channel is established at step F, the attested and authorized TEE enclave 404 can, at step G, download algorithm bytecode (or bytecodes) from the issuer's server 203 via the secure communication channel. The enclave 404 will seal the downloaded algorithm bytecode (or bytecodes) into a data storage 406, for
20 example, onto a disk of device 201. During sealing, the data of the downloaded algorithm bytecode (or bytecodes) is encrypted with a key known only to the authorised enclave 404. Any party external to the respective attested and authorized enclave 404 cannot read the sealed data. The VPN Software 402 is configured to provision the enclave 404 to unseal the sealed data when the algorithm bytecode (or bytecodes) is to be used.

- 25 Fig. 5 illustrates in more detail step 4 of Fig. 1. Fig. 5 shows the device 201 of Fig. 4, the TEE-capable processor 202 of Fig. 4, the VPN Software 402 of Fig. 4, the enclave 404 of Fig. 4 and the data storage (e.g. a hard disk) 406 of Fig. 4.

- 30 With reference to Fig. 5, when the VPN Software 402 installed on the device 201 is in use or in operation and the VPN Software 402 is configured to use an algorithm under data protection, the VPN Software 402 will provision the enclave 404 in a step A'. The enclave 404 is the enclave that sealed one or more algorithm bytecode 504 (e.g. bytecode 306 in Fig. 3) of the algorithm in the data storage 406 and holds the key to unseal the one or
35 more algorithm bytecode 504. The VPN Software 402 will submit an input 502 for the one or more algorithm bytecode 504 of the algorithm to be used in a step B' to the enclave 404. The enclave 404 will unseal the one or more algorithm bytecode 504 from the data storage 406 in a step C'. The enclave 404 will execute the one or more algorithm bytecode 504 in a step D'. If required, the enclave 404 will execute the one or more algorithm
40 bytecode 504 based on the supplied input 502. Results 506 of the operation will be returned to the VPN Software 402 in a step E'. Memory dedicated to the enclave 404 will be encrypted to prevent external parties from inspecting the operations of the enclave 404.

- 45 Examples of the present disclosure can solve the following problems. VPN is useful and issuers (organizations) want to install VPN Software in the devices issued to users (staff/members). Such VPN Software may use proprietary algorithms. There is currently no solution to prevent theft of such algorithms by staff/members. Such theft can also happen if the issued devices of staff/members are stolen. Secrecy of customized algorithms provide an added layer of defence, increasing the difficulty of an adversary
50 attack against software systems especially when reverse engineering is not viable. An issuer's (organization's) computer network can be compromised if weaknesses in the stolen algorithms can be discovered.

- 55 It is common practice to prevent the theft of such algorithms by encrypting them and only allowing staff/members to decrypt them with passwords/smart cards. However, these measures only prevent theft for stolen devices. If the staff/members of the organization

are malicious actors, they can decrypt and steal the algorithms. The algorithms can also be extracted from memory during usage, even if the algorithms are encrypted at rest.

Examples of the present disclosure may have the following advantages:

- 5 • Compatibility with any algorithms to be subject to data protection;
- Does not just rely on application security (e.g. data encryption, password requirement);
- Resistant against software and hardware (non-TEE) flaws; and
- Resistant against insider (e.g. malicious employee) threats.
- 10 • TEEs protect code and data from disclosure or modification.
- Through an attestation process, relevant parties can verify or authorise an enclave's identity and securely provide algorithms and update them through the verified or authorised enclave.
- Sealing/unsealing functions let TEEs save (and seal) and unseal algorithms to a data storage (e.g. on hard disk) securely.
- 15 • Trusted computing base and attack surface of a device are reduced. Before using TEE, the trusted computing base is the entire application or applications (e.g. the VPN Software) running on the device, operating system of the device, and the hardware of the device. After using TEE, the trusted computing base is an enclave application (small component of the original application) and the TEE hardware implementation, if implemented.
- 20

In examples of the present disclosure, the external device hosting the VPN Software has to be TEE-capable, for example, comprising a TEE-capable CPU.

25 An example of the device 201 in Fig. 2 may have the following components in electronic communication via a bus:

1. a display;
2. non-volatile memory and/or non-transitory computer readable medium;
- 30 3. random access memory ("RAM");
4. N number of processing components (i.e. "one or more processors" or "one or more central processing units");
5. a transceiver component that includes N number of transceivers for Internet/Intranet use, and/or Wireless Network communications;
- 35 6. user controls i.e. user input devices;
7. optionally, image capturing components;
8. optionally, audio signal capturing components;
9. optionally, audio speakers; and
- 40 10. Input/Output interfaces for connecting to the user input devices (such as mouse, joystick, keyboard, sensors for detecting user gestures, and the like), the audio speakers, display, image capturing components and/or audio signal capturing components.

The server 203 in Fig. 2 may have the same hardware components and essential software components as the device 201 but may exclude the TEE features.

45 The display generally operates to provide a presentation of graphical content (e.g. graphical user interface) to a user, and may be realized by any of a variety of displays (e.g., CRT, LCD, HDMI, micro-projector and OLED displays). In general, the non-volatile memory functions to store (e.g., persistently store) data and executable code including

50 code that is associated with the functional components of VPN Software, TEE related software (where applicable), etc. In some cases, for example, the non-volatile memory includes bootloader code, modem software, operating system code, file system code, and code to facilitate the running of VPN Software and TEE related software (where applicable) as well as other components well known to those of ordinary skill in the art that

55 are not depicted for simplicity.

In many implementations, the non-volatile memory is realized by flash memory (e.g., NAND or NOR memory), but it is certainly contemplated that other memory types may be utilized as well. Although it may be possible to execute the code from the non-volatile memory, the executable code in the non-volatile memory is typically loaded into RAM and executed by one or more of the N processing components.

One or more computer programs (e.g. the VPN Software) may be stored on any machine or computer readable medium that may be non-transitory in nature. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with the device 201. The machine or computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the Wireless LAN (WLAN) system.

The N processing components (or "one or more processors") in connection with RAM generally operate to execute the instructions stored in non-volatile memory to effectuate the functional components. As one skilled in the art (including ordinarily skilled) will appreciate, the N processing components may include a video processor, modem processor, DSP, graphics processing unit (GPU), and other processing components. As TEE capability is crucial to the device 201, in the case that TEE hardware is implemented, the N processing components will include the necessary TEE enabled hardware. In some implementations, the N processing components are configured to determine a type of software activated on the device 201.

The transceiver component may include N transceiver chains, which may be used for communicating with external devices via wireless networks. Each of the N transceiver chains may represent a transceiver associated with a particular communication scheme. For example, each transceiver may correspond to protocols that are specific to local area networks, cellular networks (e.g., a WIFI network, a CDMA network, a GPRS network, a UMTS networks), and other types of communication networks. In some implementations, the communication of the transceiver component with communication networks enables a location of connected devices to be determined.

Examples of the present disclosure may have the following features:

An apparatus (e.g. 201 in Fig. 2) for preventing algorithm theft from Virtual Private Network (VPN) Software during algorithm use and/or storage, wherein the apparatus comprises: a processor (e.g. the N number of processing components) configured to execute instructions in a memory to operate the apparatus to: during the operation of a VPN Software (e.g. 402 in Fig. 4) at the apparatus, process one or more software components of the VPN Software classified as trusted such that the one or more trusted software components are hosted and run by one or more enclaves (e.g. 404 in Fig. 4) in a Trusted Execution Environment (TEE), and process one or more software components of the VPN Software classified as untrusted such that the one or more untrusted software components run outside of the one or more enclaves in the TEE, wherein the one or more trusted software components includes one or more software components of an algorithm to be run by the VPN Software for data protection.

The apparatus may be operable to communicate with a server (e.g. 203 in Fig. 2), wherein the server is configured to attest the one or more enclaves in the TEE prior to allowing communication between the attested one or more enclaves in the TEE and the server.

The attested one or more enclaves in the TEE and the server may be configured to conduct communication via a secure communication channel established between the server and each of the attested one or more enclaves in the TEE.

5 The apparatus may be configured to contact the server to update the one or more trusted software components to be hosted and run by the one or more enclaves in the TEE, and the update takes place after communication between the attested one or more enclaves in the TEE and the server is allowed.

The attestation conducted by the server may verify one or more of the following:

- the one or more enclaves in the TEE is loaded with the VPN Software;
- the one or more enclaves in the TEE has been initialized with correct parameters;
- 10 and
- the authenticity of the one or more enclaves in the TEE.

15 Data downloaded during the update may be sealed by the one or more enclaves in the TEE downloading the data to protect the downloaded data, and, to access the sealed data, the sealed data has to be unsealed by the one or more enclaves in the TEE that sealed the data.

20 The one or more software components of the algorithm to be run by the VPN Software for data protection may comprise one or more software components for execution of cryptography operations on data using one or more cryptographic algorithms.

25 The one or more software components of the algorithm to be run by the VPN Software for data protection may comprise one or more software components for sealing or unsealing of one or more software components.

The one or more software components of the algorithm to be run by the VPN Software for data protection may comprise one or more software components for the server to authenticate the one or more enclaves in the TEE.

30 Optionally, the one or more software components of the algorithm to be run by the VPN Software for data protection may comprise one or more software components to enable updating of the trusted one or more software components.

35 The algorithm may be converted into a program and the program may be compiled into one or more algorithm bytecode for downloading to the apparatus and for execution by the one or more enclaves in the TEE.

40 The algorithm may be converted into source code or an intermediate representation for downloading to the apparatus, and the apparatus is configured to compile the downloaded source code or intermediate representation into algorithm bytecode for execution by the one or more enclaves in the TEE.

A system comprising the apparatus and the server.

45 A method for preventing algorithm theft from Virtual Private Network (VPN) Software during algorithm use and/or storage, wherein the method comprising: during the operation of a VPN Software (e.g. 402 in Fig. 4) at an apparatus (e.g. 201 in Fig. 2), processing one or more software components of the VPN Software classified as trusted such that the one or more trusted software components are hosted and run by one or more enclaves (e.g. 50 404 in Fig. 4) in a Trusted Execution Environment (TEE), and processing one or more software components of the VPN Software classified as untrusted such that the one or more untrusted software components run outside of the one or more enclaves in the TEE, wherein the one or more trusted software components of an algorithm include one or more software components to be run by the VPN Software for data protection.

55

The method may further comprise: communicating with a server (e.g. 203 in Fig. 2), wherein the server is configured to attest the one or more enclaves in the TEE prior to allowing communication between the attested one or more enclaves in the TEE and the server.

5

The method may further comprise: conducting communication between the attested one or more enclaves in the TEE and the server via a secure communication channel established between the server and each of the attested one or more enclaves in the TEE.

10

The method may further comprise: contacting the server to update the one or more trusted software components to be hosted and run by the one or more enclaves in the TEE; and performing the update after communication between the attested one or more enclaves in the TEE and the server is allowed.

15

The attestation conducted by the server may verify one or more of the following:

- the one or more enclaves in the TEE is loaded with the VPN software;
- the one or more enclaves in the TEE has been initialized with correct parameters; and
- the authenticity of the one or more enclaves in the TEE.

20

The method may further comprise: sealing data downloaded during the update to protect the downloaded data such that, to access the sealed data, the sealed data has to be unsealed by the one or more enclaves in the TEE that sealed the data.

25

The method may further comprise: converting the algorithm into a program; compiling the program into one or more algorithm bytecode; downloading to the apparatus the one or more algorithm bytecode; and executing the one or more algorithm bytecode using the one or more enclaves in the TEE.

30

The method may comprise:

- converting the algorithm into a source code or an intermediate representation;
- downloading the source code or intermediate representation to the apparatus;
- compiling the source code or intermediate representation into algorithm bytecode; and

35

- executing the algorithm bytecode using the one or more enclaves in the TEE.

In the specification and claims, unless the context clearly indicates otherwise, the term “comprising” has the non-exclusive meaning of the word, in the sense of “including at least” rather than the exclusive meaning in the sense of “consisting only of”. The same applies with corresponding grammatical changes to other forms of the word such as “comprise”, “comprises” and so on.

40

While the invention has been described in the present disclosure in connection with a number of embodiments and implementations, the invention is not so limited but covers various obvious modifications and equivalent arrangements, which fall within the purview of the appended claims. Although features of the invention are expressed in certain combinations among the claims, it is contemplated that these features can be arranged in any combination and order.

45

50

Claims

1. An apparatus for preventing algorithm theft from Virtual Private Network (VPN) Software during algorithm use and/or storage, wherein the apparatus comprises:
 - a processor configured to execute instructions in a memory to operate the apparatus to:
 - during the operation of a VPN Software at the apparatus,
 - process one or more software components of the VPN Software classified as trusted such that the one or more trusted software components are hosted and run by one or more enclaves in a Trusted Execution Environment (TEE), and
 - process one or more software components of the VPN Software classified as untrusted such that the one or more untrusted software components run outside of the one or more enclaves in the TEE,
 - wherein the one or more trusted software components include one or more software components of an algorithm to be run by the VPN Software for data protection.
2. The apparatus as claimed in claim 1, wherein the apparatus is operable to communicate with a server, wherein the server is configured to attest the one or more enclaves in the TEE prior to allowing communication between the attested one or more enclaves in the TEE and the server.
3. The apparatus as claimed in claim 2, wherein the attested one or more enclaves in the TEE and the server are configured to conduct communication via a secure communication channel established between the server and each of the attested one or more enclaves in the TEE.
4. The apparatus as claimed in claim 2 or 3, wherein the apparatus is configured to contact the server to update the one or more trusted software components to be hosted and run by the one or more enclaves in the TEE, and the update takes place after communication between the attested one or more enclaves in the TEE and the server is allowed.
5. The apparatus as claimed in claim 2, 3 or 4, wherein the attestation conducted by the server verifies one or more of the following:
 - the one or more enclaves in the TEE is loaded with the VPN Software;
 - the one or more enclaves in the TEE has been initialized with correct parameters;
 - and
 - the authenticity of the one or more enclaves in the TEE.
6. The apparatus as claimed in claim 4, wherein data downloaded during the update is sealed by the one or more enclaves in the TEE downloading the data to protect the downloaded data, and, to access the sealed data, the sealed data has to be unsealed by the one or more enclaves in the TEE that sealed the data.
7. The apparatus as claimed in any one of claims 1 to 6, wherein the one or more software components of the algorithm to be run by the VPN Software for data protection comprises one or more software components for execution of cryptography operations on data using one or more cryptographic algorithms.
8. The apparatus as claimed in any one of claims 1 to 7, wherein the one or more software components of the algorithm to be run by the VPN Software for data protection comprises one or more software components for sealing or unsealing of one or more software components.

9. The apparatus as claimed in any one of claims 2 to 8, wherein the one or more software components of the algorithm to be run by the VPN Software for data protection comprises one or more software components for the server to attest the one or more enclaves in the TEE.
10. The apparatus as claimed in any one of claims 1 to 9, wherein the algorithm is converted into a program and the program is compiled into one or more algorithm bytecode for downloading to the apparatus and for execution by the one or more enclaves in the TEE.
11. The apparatus as claimed in any one of claims 1 to 10, wherein the algorithm is converted into source code or an intermediate representation for downloading to the apparatus, and the apparatus is configured to compile the downloaded source code or intermediate representation into algorithm bytecode for execution by the one or more enclaves in the TEE.
12. A system comprising the apparatus as claimed in any one of claims 2 to 11 and the server.
13. A method for preventing algorithm theft from Virtual Private Network (VPN) Software during algorithm use and/or storage, wherein the method comprises:
 - during the operation of a VPN Software at an apparatus,
 - processing one or more software components of the VPN Software classified as trusted such that the one or more trusted software components are hosted and run by one or more enclaves in a Trusted Execution Environment (TEE), and
 - processing one or more software components of the VPN Software classified as untrusted such that the one or more untrusted software components run outside of the one or more enclaves in the TEE,
 - wherein the one or more trusted software components include one or more software components of an algorithm to be run by the VPN Software for data protection.
14. The method as claimed in claim 13, wherein the method further comprises: communicating with a server, wherein the server is configured to attest the one or more enclaves in the TEE prior to allowing communication between the attested one or more enclaves in the TEE and the server.
15. The method as claimed in claim 14, wherein the method further comprises: conducting communication between the attested one or more enclaves in the TEE and the server via a secure communication channel established between the server and each of the attested one or more enclaves in the TEE.
16. The method as claimed in claim 14 or 15, wherein the method further comprises: contacting the server to update the one or more trusted software components to be hosted and run by the one or more enclaves in the TEE; and performing the update after communication between the attested one or more enclaves in the TEE and the server is allowed.
17. The method as claimed in claim 14, 15 or 16, wherein the attestation conducted by the server verifies one or more of the following:
 - the one or more enclaves in the TEE is loaded with the VPN Software;
 - the one or more enclaves in the TEE has been initialized with correct parameters; and
 - the authenticity of the one or more enclaves in the TEE.

18. The method as claimed in claim 16, wherein the method further comprises:
sealing data downloaded during the update to protect the downloaded data such
that, to access the sealed data, the sealed data has to be unsealed by the one or
more enclaves in the TEE that sealed the data.

5

19. The method as claimed in any one of claims 13 to 18, wherein the method
comprises:

converting the algorithm into a program;

compiling the program into one or more algorithm bytecode;

10

downloading to the apparatus the one or more algorithm bytecode; and

executing the one or more algorithm bytecode using the one or more enclaves in
the TEE.

20. The method as claimed in any one of claims 13 to 19, wherein the method
comprises:

15

converting the algorithm into a source code or an intermediate representation;

downloading the source code or intermediate representation to the apparatus;

compiling the source code or intermediate representation into algorithm bytecode;

and

20

executing the algorithm bytecode using the one or more enclaves in the TEE.