2023-04-21

Created: 2023-04-22 01:03:03

Chats

- SGX.0418.md
  - `[09:17]` What are types of Intel SGX Keys?
  - `[13:53]` What are default enclaves?
  - `[13:56]` What are Intel provided enclaves?
  - `[14:45]` Is a SIGSTRUCT a certificate?
  - `[15:10]` Can I say that SIGSTRUCT is enclave identity?
  - `[15:11]` Can I say that SIGSTRUCT contains enclave identity?
  - `[15:23]` Does MRSIGNER reside in SIGSTRUCT?
  - `[15:30]` Where MRSIGNER reside?
  - `[15:36]` What is hard-coded MRSIGNER?
  - `[15:40]` What is Intel's hard-coded MRSIGNER?
  - `[15:41]` What is the value of the hard-coded MRSIGNER provided by Intel?
  - `[15:58]` Can I create myself enclave with SIGSTRUCT signed by my RSA private key and run the enclave in Intel SGX environment?
  - `[16:24]` How do I allow the platform owner to trust my enclave?
  - `[16:26]` Where the trusted signer list reside in Intel SGX environment?
- Papers.SGX.0419.md
  - `[15:29]` Does MRSIGNER reside in SIGSTRUCT?

Q & A

SGX.0418.md

Question: `[09:17]` What are types of Intel SGX Keys?

Answer:

Intel SGX provides several types of keys to facilitate secure operations within and around enclaves. The main types of Intel SGX keys are:

1. Seal Key: A key used to encrypt and authenticate data that will be stored outside the enclave, ensuring the confidentiality and integrity of the data. Seal keys are derived from a hardware root key unique to the processor and are not accessible outside the enclave.
2. Provisioning Key: A key that is used during the provisioning process to establish a secure channel between the enclave and a provisioning service. This key is derived from a hardware root key and can only be generated by an enclave with the provisioning attribute set.
3. Provisioning Seal Key: A seal key used specifically for the provisioning process. It is derived from the same hardware root key as the provisioning key but is used to protect data used during provisioning.
4. Report Key: A key used to generate and verify enclave reports, which are data structures containing information about an enclave's identity and state. The report key is derived from a hardware root key and is accessible only within the enclave.
5. Attestation Key: A key used in the remote attestation process to sign enclave quotes, which are assertions about the enclave's identity and state. The attestation key is stored inside the enclave and is linked to an Intel Enhanced Privacy ID (EPID) or an ECDSA Attestation key.

These keys play different roles in ensuring the security and privacy of enclaves and their associated data. They are generated and managed by the SGX architecture in a way that provides strong isolation and protection against unauthorized access or tampering.

Question: `[13:53]` What are default enclaves?

Answer:

In the context of Intel SGX, there isn't a specific term called "default enclaves." However, if you are referring to pre-built or pre-defined enclaves, they can be understood as enclaves that are included with an SGX-enabled platform or provided by a third-party as part of a software package.

These pre-built or pre-defined enclaves can be used as starting points for developers or as building blocks to create more complex applications. They may offer basic functionality, like cryptographic operations, or be more specialized, providing a specific service for a given application.

Nevertheless, it is essential to understand that enclaves are typically developed and tailored to specific use cases, ensuring that they meet the security and performance requirements for the given scenario. The term "default enclave" might not accurately represent the customized nature of enclaves in Intel SGX applications.

Question: `[13:56]` What are Intel provided enclaves?

Answer:

Intel provides some enclaves, known as Intel-provided enclaves, that offer specific functionality to support the Intel SGX ecosystem. These enclaves are designed to help developers implement security features and other services in their SGX applications more efficiently. Some of the Intel-provided enclaves include:

1. Quoting Enclave (QE): The Quoting Enclave is responsible for generating quotes as part of the SGX remote attestation process. It takes a report from another enclave, verifies it, and creates a quote that can be sent to a remote party for validation.
2. Provisioning Enclave (PvE): The Provisioning Enclave is responsible for managing the EPID (Enhanced Privacy ID) provisioning process. It helps in obtaining the EPID key and certificate for the platform from the Intel Attestation Service (IAS).
3. Platform Services Enclave (PSE): The Platform Services Enclave provides additional services and functionalities that may be required by other enclaves, such as monotonic counters, trusted time, and sealed storage.
4. Architectural Enclave Service Manager (AESM): While not an enclave itself, the AESM is a system service that manages the interaction between Intel-provided enclaves and the platform, helping to facilitate the attestation and provisioning processes.

These Intel-provided enclaves serve as a foundation for the Intel SGX ecosystem, assisting developers in creating secure applications and services on SGX-enabled platforms.

Question: `[14:45]` Is a SIGSTRUCT a certificate?

Answer:

A SIGSTRUCT can be thought of as a certificate-like structure in the Intel SGX ecosystem, but it is not a traditional X.509 certificate. The SIGSTRUCT is a data structure that contains information about an enclave, such as the MRENCLAVE (measurement of the enclave), MRSIGNER (measurement of the enclave signer), and other attributes. It also contains the enclave's signature, generated using the enclave author's private key.

The purpose of the SIGSTRUCT is to provide a signed description of an enclave, which can be used during the enclave's loading process. The Intel SGX-enabled platform uses the SIGSTRUCT to verify the integrity and authenticity of the enclave before allowing it to be executed. In this sense, it serves a similar purpose to a certificate, but its format and usage are specific to the Intel SGX environment.

Question: `[15:10]` Can I say that SIGSTRUCT is enclave identity?

Answer:

It's not entirely accurate to say that the SIGSTRUCT is the enclave identity. The SIGSTRUCT is a signed data structure containing information about an enclave, including the MRENCLAVE (measurement of the enclave) and MRSIGNER (measurement of the enclave signer). It plays a role in verifying the integrity and authenticity of an enclave during the loading process.

The enclave identity is typically derived from attributes such as MRENCLAVE and MRSIGNER, which are part of the SIGSTRUCT. In this sense, the SIGSTRUCT contributes to the enclave identity, but it is not the identity itself. The enclave identity is more about the unique combination of characteristics (MRENCLAVE, MRSIGNER, and other attributes) that identify a specific enclave.

Question: `[15:11]` Can I say that SIGSTRUCT contains enclave identity?

Answer:

Yes, you can say that the SIGSTRUCT contains enclave identity, as it includes important attributes such as MRENCLAVE and MRSIGNER, which together contribute to the unique identity of an enclave. These attributes, along with other information in the SIGSTRUCT, are used to verify the integrity and authenticity of an enclave during the loading process.

Question: `[15:23]` Does MRSIGNER reside in SIGSTRUCT?

Answer:

Yes, MRSIGNER resides in the SIGSTRUCT. MRSIGNER is the hash of the public key used to sign the enclave. The SIGSTRUCT is a data structure that contains the enclave's identity and integrity information, including the MRSIGNER value, MRENCLAVE (hash of the enclave's initial state), and the enclave's version, among other things. The SIGSTRUCT is used during the enclave's loading process to verify its authenticity and integrity.

Question: `[15:30]` Where MRSIGNER reside?

Answer:

MRSIGNER resides in the SIGSTRUCT, which is a data structure that contains the enclave's identity and integrity information. The MRSIGNER value is the hash of the public key used to sign the enclave. When the enclave is loaded, the SIGSTRUCT is used to verify the authenticity and integrity of the enclave, ensuring that it has not been tampered with and that it originates from a trusted source.

Question: `[15:36]` What is hard-coded MRSIGNER?

Answer:

A hard-coded MRSIGNER refers to a predefined MRSIGNER value that is used in a specific context or for a particular purpose. This MRSIGNER value is typically associated with a known and trusted enclave developer or a specific enclave application. By checking for a hard-coded MRSIGNER value during the enclave attestation process, it is possible to ensure that the enclave is signed by the expected entity and has not been tampered with or modified by an unauthorized party.

Question: `[15:40]` What is Intel's hard-coded MRSIGNER?

Answer:

Intel does not have a universal hard-coded MRSIGNER value applicable to all enclaves. However, Intel provides MRSIGNER values for their specific Intel-provided enclaves or other enclaves they develop. These MRSIGNER values are associated with Intel's signing key and are used to verify the authenticity of the enclaves. In the context of Intel SGX, MRSIGNER values are specific to the enclaves they represent and are unique to the entity or organization that signed the enclave.

If you are working with an Intel-provided enclave, you can find the associated MRSIGNER value in the enclave's documentation or other relevant materials. Keep in mind that MRSIGNER values are specific to individual enclaves, and the value you need to use will depend on the particular enclave you are working with.

Question: `[15:41]` What is the value of the hard-coded MRSIGNER provided by Intel?

Answer:

As an AI language model, I do not have real-time access to the specific MRSIGNER values provided by Intel for their enclaves. However, I can provide you with some guidance on how to find the MRSIGNER value associated with a particular Intel-provided enclave.

If you are working with an Intel-provided enclave, you can find the associated MRSIGNER value in the enclave's documentation, SDK materials, or other relevant resources provided by Intel. The MRSIGNER value will be unique to the specific enclave you are working with, so it is important to consult the appropriate materials related to that enclave.

Additionally, you can find the MRSIGNER value programmatically by extracting it from the enclave's SIGSTRUCT structure when the enclave is loaded. Keep in mind that doing so requires a proper understanding of the enclave's layout and structure.

Question: `[15:58]` Can I create myself enclave with SIGSTRUCT signed by my RSA private key and run the enclave in Intel SGX environment?

Answer:

Yes, you can create an enclave with a SIGSTRUCT signed by your own RSA private key and run the enclave in the Intel SGX environment. When you create an enclave, you'll need to generate a key pair (private and public keys) and use your private key to sign the enclave's SIGSTRUCT. The MRSIGNER value will then represent the hash of your public key.

However, running the enclave on a platform will require the platform owner or administrator to trust your enclave. Trust is established by either including your public key in the platform's trusted signer list or by having a trusted third party (such as Intel) vouch for your enclave through a certificate chain.

In summary, while it's technically possible to create an enclave signed by your own RSA private key, the enclave's trustworthiness and acceptance in the Intel SGX environment will depend on the platform owner or administrator's trust configuration.

Question: `[16:24]` How do I allow the platform owner to trust my enclave?

Answer:

To allow the platform owner to trust your enclave, you can follow these steps:

1. Share your public key: After signing your enclave with your private key, share the corresponding public key with the platform owner.
2. Establish trustworthiness: The platform owner needs to verify that your enclave is trustworthy. This can be done through a code review, an audit from a trusted third party, or other means of establishing trust in your enclave's code and functionality.
3. Add your public key to the trusted signer list: The platform owner can then add your public key to the trusted signer list on their platform. This list is used by the Intel SGX environment to determine which enclaves are trusted to run on the platform. The platform owner may maintain this list locally on the platform or use a remote attestation service to manage trusted keys.
4. Configure the platform: The platform owner may need to configure the Intel SGX environment to accept enclaves signed by non-Intel entities or update the trusted signer list to include your public key.
5. Remote attestation (optional): In some cases, you might want to use remote attestation to prove to a relying party that your enclave is running securely on the platform. To do this, you'll need to provide the remote attestation service with your enclave's measurement, along with any other required information, such as your enclave's MRSIGNER value. The attestation service can then verify the enclave's identity and provide an attestation report to the relying party.

By following these steps, you can establish trust with the platform owner and enable them to trust and run your enclave in their Intel SGX environment.

Question: `[16:26]` Where the trusted signer list reside in Intel SGX environment?

Answer:

In the Intel SGX environment, the trusted signer list is maintained by the platform owner, and it is not explicitly part of the SGX hardware or SDK. The list can be managed in various ways, depending on the implementation chosen by the platform owner.

Some possible ways to manage the trusted signer list are:

1. Locally on the platform: The platform owner can maintain a list of trusted public keys within the local configuration of the Intel SGX environment. This list can be updated as needed, and the platform owner can decide which enclaves are allowed to run based on the MRSIGNER values of those enclaves.
2. Remote attestation service: The platform owner can use a remote attestation service, such as Intel's Attestation Service, to manage the list of trusted public keys. In this case, the attestation service verifies the identity of the enclave, and the platform owner can decide whether to trust the enclave based on the attestation report provided by the service.
3. Certificate-based trust: The platform owner may choose to trust enclaves signed by specific Certificate Authorities (CAs). In this case, the platform owner maintains a list of trusted CAs, and enclaves signed by those CAs are considered trustworthy.

Ultimately, the management of the trusted signer list is determined by the platform owner and their specific implementation of the Intel SGX environment.

Papers.SGX.0419.md

Question: `[15:29]` Does MRSIGNER reside in SIGSTRUCT?

Answer:

Yes, MRSIGNER is included in the SIGSTRUCT data structure. The MRSIGNER value is a hash of the public key used to sign the enclave during the enclave build process. SIGSTRUCT is a data structure that contains the enclave's identity and other metadata necessary for loading the enclave into the Intel SGX architecture.

When an enclave is created, the SIGSTRUCT is presented to the hardware by the enclave builder. The hardware then checks the integrity and authenticity of the enclave by verifying the signature in the SIGSTRUCT. The MRSIGNER value in the SIGSTRUCT is used to identify the enclave signer (the enclave's author or authority) and helps to ensure the enclave's integrity and trustworthiness.
