Workgroup: TEEP
Internet-Draft: draft-yang-teep-tee-dp-00
Published: 3 April 2023
Intended Status: Informational
Expires: 5 October 2023
Authors: P. Yang China Mobile; T. Pang Huawei Technology Co.,Ltd.; X. Zhang AntGroup

Trusted Execution Environment Distributed Provisioning Protocol

Abstract

In big data area, computing resource manager like MESOS[MESOS], YARM[YARN], kubernets[Kubernetes] or computing framework like Spark[Spark], use Master-Worker structure to split computing task. In the master component, the computing task will be splited into different child tasks. Each of thess child tasks will be loaded to a executor which is managed by Worker. The Master and Worker are usually exist as cluster, cloud or other distributed framework. When the big data tasks needs to be processed in TEE in lifecycle, this document could be used for Master to provision child tasks in distributed TEEs.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction
2. Terminology
3. Use cases
4. Architecture
5. TEE Distributed Provisioning Protocol
  5.1. TEE Resource Request/Response
  5.2. TEE Secure Channel Request/Response
6. Security Considerations
7. IANA Considerations
8. References
  8.1. Normative References
  8.2. Informative References
Appendix A. Appendix 1 Full CDDL of TEE DP protocol
Authors' Addresses

1. Introduction

In big data area, different from stand-alone applications, big data application always need to be splitted into different child tasks by Master in big data computing framework. Then these tasks will be deployed to executors in local or remote by Workers. TEE could be used to protect the application and its secret data during the process, if only the whole process lifecycle is covered by TEE. Before running big data application, it is hard to predict how many computing resources are needed. Similarly, TEE resource also needs to be provisioned during the application process lifecycle. This document specifies the architecture and protocol of how big data computing framework provision and use TEE during application process lifecycle. The Trusted Execution Environment Provisioning (TEEP) architecture document [I-D.ietf-teep-architecture] provides design guidance and introduces the necessary terminology.

2. Terminology

The following terms are used:

*Big Data Computing Framework: An framework that is responsible for managing and spliting big data computing task, like Spark, MapReduce[MapReduce], etc. Usually, Big Data Computing Framework has its own Computing Resource Manage Framework. Like Spark supports standalone deploy mode.

*Computing Resource Manage Framework: An framework that is responsible for managing and scheduling computing resource in cluster, like YARN, MESOS and Kubernetes.

*Master: The entity in Computing Resource Manage Framework that is responsible for splitting computing task into different child tasks, and allocating computing resource to those child tasks.

*Executor: The executing entity that is responsible for executing child tasks in Worker. The current executor includes process, container and VM.

*Worker: The entity that is responsible for undertaking child tasks and manage Executors. Other terms like TAM, TEE, REE, TA will reuse the term definition defined in [I-D.ietf-teep-architecture].

3. Use cases

In federated machine learning, participants want to create a unified machine learning model without leaking private data owned by each other. TEE as a hardware based technology could make sure data inside this environment is integrated and confidential. If the federated machine learning participants trust this TEE and its TA, they could gather their data in that TEE and generate the final machine learning model. The architecture and protocol described in this document could be used in the federated machine learning scenario, and make sure the lifecycle of machine learning process is protected by TEE.

4. Architecture

The following figure shows the architecture of TEE distributed provisioning. In this architecture, Master is the management center of big data Computing Resource Management Framework. it also plays the role of TAM in TEEP architecture. When Master starts running big data applications, it forwards TEE computing resource request to Worker by TEE-DP protocol. Worker then occupies TEE computing resource and generate Executor which is running inside TEE. Meanwhile, TEE-DP protocol also includes secure channel negotiation message. Based on the secure channel between TAM and Executor, TEEP protocol could provision child tasks generated by Master securely.

```
                         +--------+
      +----------------->| Master |<------------------+
      |                  |  /TAM  |                   |
      |                  +--------+                   |
      |                       ^                       |
      |                       |                       |
      |TEE-DP          +-------+               TEE-DP|
      |protocol        |TEEP protocol         protocol|
      |                |                              |
      v                v                              v
  +---------+    +----------+     +----------+    +--------+
  |  Worker |<-->| Executor |     | Executor |<-->| Worker |
  +---------+    +----------+     +----------+    +--------+
                 |   TEE    |     |   TEE    |
                 +----------+     +----------+
```

Figure 1: TEEP Broker Models

5. TEE Distributed Provisioning Protocol

As described in architecture section, TEE distributed provisioning protocol has two message packages: TEE resource request/responses, TEE secure channel request/response. The message framework is shown below in CDDL format.

```
tee-dp-message = $tee-dp-message-type .within tee-dp-message-framework
tee-dp-message-framework = [
        type: $tee-dp-type,
        optionis: { * tee-dp-option}
]
tee-dp-option = (uint =>any)
$tee-dp-message-type /= tee-resource-request
$tee-dp-message-type /= tee-resource-response
$tee-dp-message-type /= tee-secure-channel-resquest
$tee-dp-message-type /= tee-secure-channel-response-direct
$tee-dp-message-type /= tee-secure-channel-response-indirect

$tee-dp-type = uint .size 1
TEE-resource-request = 1
TEE-resource-response = 2
TEE-secure-channel-request = 3
TEE-secure-channel-response-direct = 4
TEE-secure-channel-response-indirect = 5
```

Figure 2: TEE DP Message Framework

5.1. TEE Resource Request/Response

TEE resource request/response message is used by Master/TAM and Worker to negotiate TEE computing resource. The resource request message is sent by Master/TAM, then the Worker response this message. The only mandantory option in request message is MEMORY-size. Other items like CPU core number, CPU frequency, CPU architecture, memory encryption method and memory isolation method are optional. Another message that is mandantory in response message is TOKEN-tee-instance, which represents the identity of selected TEE instance. This token could be public key of AIK(attestation identity key) or otehr identity information. The relevant CDDL fragment is shown below. The complete CDDL structure is shown in Appendix.

```
memory-encryption-method = &(
        hardware-based: 0,
        software-based: 1,
        none: 2
)
memory-isolation-method = &(
        hardware-based: 0,
        software-based: 1,
        none: 2,
)
cpu-architecture = &(
        sgx-based: 0,
        sev-based: 1,
        trustzone-based: 2,
        other: 3,
)

tee-resource-request = [
        type: TEE-resource-request
        options:{
                ? CPU-core-number: uint .size 1
                ? CPU-frequency: unit .size 2
                ? CPU-arch: uint .bits cpu-architecture
                ;the cpu frequency unit is MHZ
                MEMORY-size: uint .size 4
                ;the memory size unit is MB
                ? Requested-memory-encryption-method: uint .bits memory-
                ? Requested-memory-isolation-method: uint .bits memory-i
        }
]

tee-resource-response = [
        type: TEE-resource-response
        options:{
                ? CPU-core-number: uint .size 1
                ? CPU-frequency: unit .size 2
                ? MEMORY-size: uint .size 4
                ? TOKEN-tee-instance: bstr .size 4
                ;this token represents the identity of TEE, it could be
                ? Requested-memory-encryption-method: uint .bits memory-
                ? Requested-memory-isolation-method: uint .bits memory-i
        }
]
```

{ #res-req title="TEE DP Resource Request Response " }

5.2. TEE Secure Channel Request/Response

Before executing TEEP protocol between Master/TAM and Executor in TEE DP architecutre, secure channel needs to be established first. There are two kinds of secure channel in TEE DP architecture, direct and indirect. Direct means Master/TAM direct connect with Executor by network, which also means the TEE of Executor support network communication, like TLS. Indirect means the Master/TAM and Executor connect with each other by other components like TEEP broker, or Worker. Master/TAM and Executor could transfer encrypted packages like COSE, JWE, by that component. The CDDL fragment is shown below.

```
secure-channel-negotiation-type= &(
        direct: 0;
        indirect: 1;
)
tee-secure-channel-resquest = [
        type: TEE-secure-channel-request
        options:{
                TOKEN-tee-instance:
                Secure-channel-negotiation-type: $$negotiation-type
        }
]

$$negotiation-type //= {
        direct: bool
        ip-type: bool ;true is ipv4, false is ipv6
}

$$negotiation-type //= {
        indirect: bool
        Protocol-name => uint .bits protocol-name
}


$$ipaddr //= (
        ipv4: bstr .size 4
)
$$ipaddr //= (
        ipv6: bstr .size 16
)

direct-extensions = (uint => any)
tee-secure-channel-response-direct = [
        type: TEE-secure-channel-response-direct
        options:{
                $$ipaddr
                port => uint .size 1
                *direct-extensions
        }
]

&protocol-name &= (
        cose: 0,
        jwe: 1,
        others: 2,
)

indirect-extentions = (uint => any)
tee-secure-channel-response-indirect = [
        type: TEE-channel-response-indirect
```

```
options:{
        Protocol-name => uint .bits protocol-name
        *indirect-extensions
}
```

Figure 3: TEE DP Secure Channel Negotiation

6. Security Considerations

The trust domain of TEE DP architecture is Master/TAM and Executor running in TEE. The Worker component do not have to be trusted by Master/TAM since it is only used for creating Executor and monitoring runtime status. The security of secure channel based on the secure channel negotiation mechanism which is out of the scope of this document.

7. IANA Considerations

TBD.

8. References

8.1. Normative References

[I-D.ietf-teep-architecture] Pei, M., Tschofenig, H., Thaler, D., and D. M. Wheeler, "Trusted Execution Environment Provisioning (TEEP) Architecture", Work in Progress, Internet-Draft, draft-ietf-teep-architecture-19, 24 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-teep-architecture-19>.

[I-D.ietf-teep-protocol] Tschofenig, H., Pei, M., Wheeler, D. M., Thaler, D., and A. Tsukamoto, "Trusted Execution Environment Provisioning (TEEP) Protocol", Work in Progress, Internet-Draft, draft-ietf-teep-protocol-12, 13 March 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-teep-protocol-12>.

[I-D.ietf-teep-usecase-for-cc-in-network] Yang, P., chenmeiling, Su, L., and T. Pang, "TEEP Usecase for Confidential Computing in Network", Work in Progress, Internet-Draft, draft-ietf-teep-usecase-for-cc-in-network-02, 20 October 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-teep-usecase-for-cc-in-network-02>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

8.2. Informative References

[Kubernetes] Community, K., "Kubernetes Cloud Controller Manager", 17 December 2022, <https://kubernetes.io/docs/concepts/architecture/cloud-controller/>.

[MapReduce]
Community, A. M., "MapReduce Overview", 27 July 2022, <https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Overview>.

[MESOS] Community, A. M., "MESOS Architecture", 2 March 2023, <https://mesos.apache.org/documentation/latest/architecture/>.

[Spark] Community, S., "Spark Overview", 2 March 2023, <https://spark.apache.org/docs/3.3.2/cluster-overview.html>.

[YARN] Community, A. H., "Apache Hadoop YARN", 29 July 2022, <https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html>.

Appendix A. Appendix 1 Full CDDL of TEE DP protocol

The full CDDL of TEE distributed provisioning protocol is shown below.

```
tee-dp-message = $tee-dp-message-type .within tee-dp-message-framework

tee-dp-message-framework = [
        type: $tee-dp-type,
        optionis: { * tee-dp-option}
]

tee-dp-option = (uint =>any)

$tee-dp-message-type /= tee-resource-request
$tee-dp-message-type /= tee-resource-response
$tee-dp-message-type /= tee-secure-channel-resquest
$tee-dp-message-type /= tee-secure-channel-response-direct
$tee-dp-message-type /= tee-secure-channel-response-indirect

$tee-dp-type = uint .size 1
TEE-resource-request = 1
TEE-resource-response = 2
TEE-secure-channel-request = 3
TEE-secure-channel-response-direct = 4
TEE-secure-channel-response-indirect = 5

memory-encryption-method = &(
        hardware-based: 0,
        software-based: 1,
        none: 2
)
memory-isolation-method = &(
        hardware-based: 0,
        software-based: 1,
        none: 2,
)

tee-resource-request = [
        type: TEE-resource-request
        options:{
                CPU-core-number: uint .size 1
                CPU-frequency: unit .size 2
                ;the cpu frequency unit is MHZ
                MEMORY-size: uint .size 4
                ;the memory size unit is MB
                ? Requested-memory-encryption-method: uint .bits memory-
                ? Requested-memory-isolation-method: uint .bits memory-i
        }
]

tee-resource-response = [
        type: TEE-resource-response
        options:{
                CPU-core-number: uint .size 1
                CPU-frequency: unit .size 2
                MEMORY-size: uint .size 4
                TOKEN-tee-instance: bstr .size 4
                ;this token represents the identity of TEE, it could be
                ? Requested-memory-encryption-method: uint .bits memory-
                ? Requested-memory-isolation-method: uint .bits memory-i
        }
]

secure-channel-negotiation-type= &(
        direct: 0;
        indirect: 1;
)
tee-secure-channel-resquest = [
        type: TEE-secure-channel-request
        options:{
                TOKEN-tee-instance:
                Secure-channel-negotiation-type: $$negotiation-type
        }
]

$$negotiation-type //= {
        direct: bool
        ip-type: bool ;true is ipv4, false is ipv6
}

$$negotiation-type //= {
        indirect: bool
        Protocol-name => uint .bits protocol-name
}

$$ipaddr //= (
        ipv4: bstr .size 4
)
$$ipaddr //= (
        ipv6: bstr .size 16
)

direct-extensions = (uint => any)
tee-secure-channel-response-direct = [
        type: TEE-secure-channel-response-direct
        options:{
                $$ipaddr
                port => uint .size 1
                *direct-extensions
        }
]
```

```
&protocol-name &= (
        cose: 0,
        jwe: 1,
        others: 2,
)

indirect-extentions = (uint => any)
tee-secure-channel-response-indirect = [
        type: TEE-channel-response-indirect
        options:{
                Protocol-name => uint .bits protocol-name
                *indirect-extensions
        }
]

; labels of mapkey for tee dp message parameters, uint (0..15)
CPU-core-number = 0
CPU-frequency = 1
MEMORY-size = 2
Requested-memory-encryption-method = 3
Requested-memory-isolation-method = 4
TOKEN-tee-instance = 5
Secure-channel-negotiation-type = 6
direct = 7
indirect = 8
ip-type = 9
Protocol-name = 10
ipaddr = 11
port = 12
direct-extensions = 13
indirect-extensions = 14
```

Figure 4: Full CDDL of TEE Distributed Provisioning Protocol

Authors' Addresses

Penglin Yang
China Mobile
No.32 Xuanwumen West Street
Beijing
China

Email: yangpenglin@chinamobile.com

Ting Pang
Huawei Technology Co.,Ltd.
127 Jinye Road, Yanta District
Xi'an
China

Email: pangting@huawei.com

Xiaomeng Zhang
AntGroup
World Financial Center, No.1 East 3rd Ring Middle Road, Chaoyang
District
Beijing
China

Email: zhangxiaomeng.zxm@antgroup.com