基础设施与软件架构

本期议题：Java机密计算 – 为Java应用打造安全金钟罩

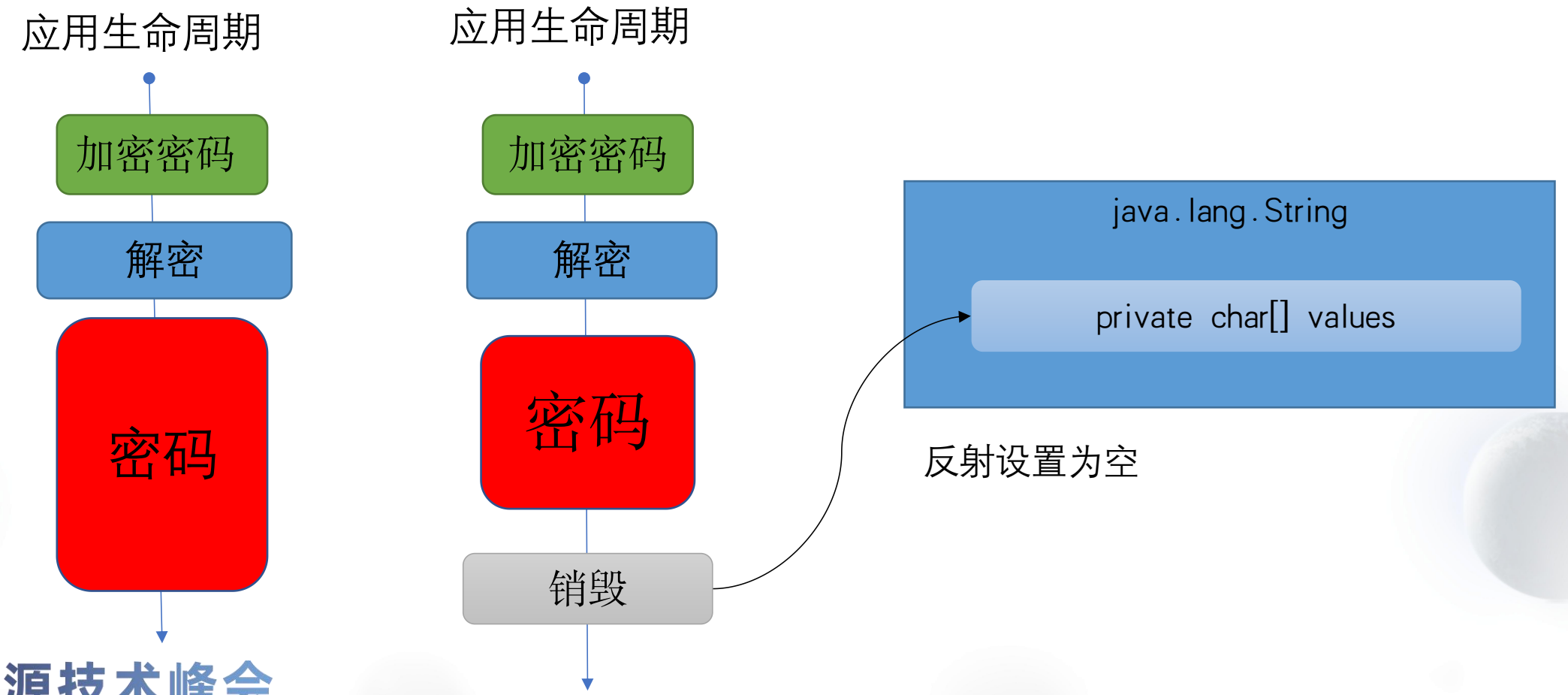
林子熠 Friday, June 02, 2023

关于我

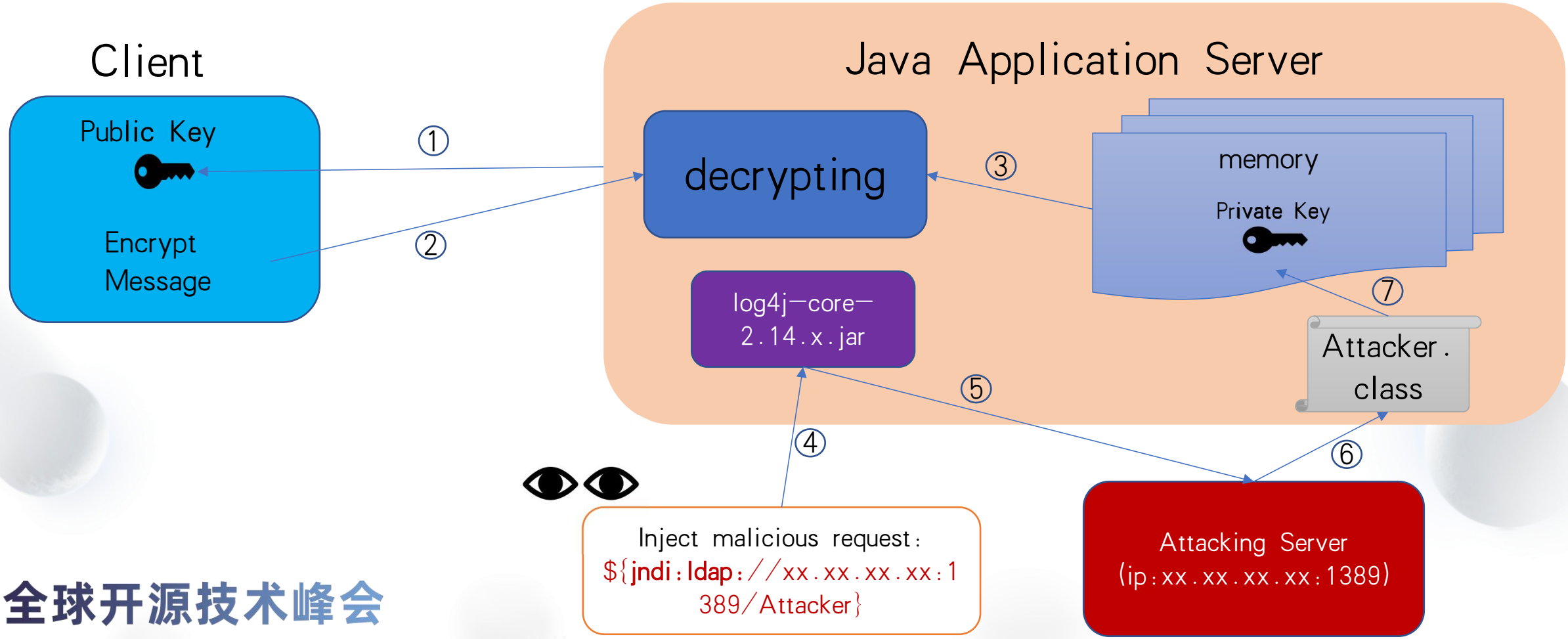
- 林子熠 博士 CCF系统专委会执行委员
- Apache Committer
- 龙蜥社区机密计算SIG maintainer
- 阿里巴巴JVM团队技术专家，负责GraalVM Java静态编译和静态分析的开发和应用
- 《GraalVM与Java静态编译原理与应用》作者
- 发表多篇CCF A类论文
- ACM SIGSOFT杰出论文奖获得者

动机 - 如何保障Java应用中的密码安全呢?

- 密码以明文保存在内存中，很容易泄漏



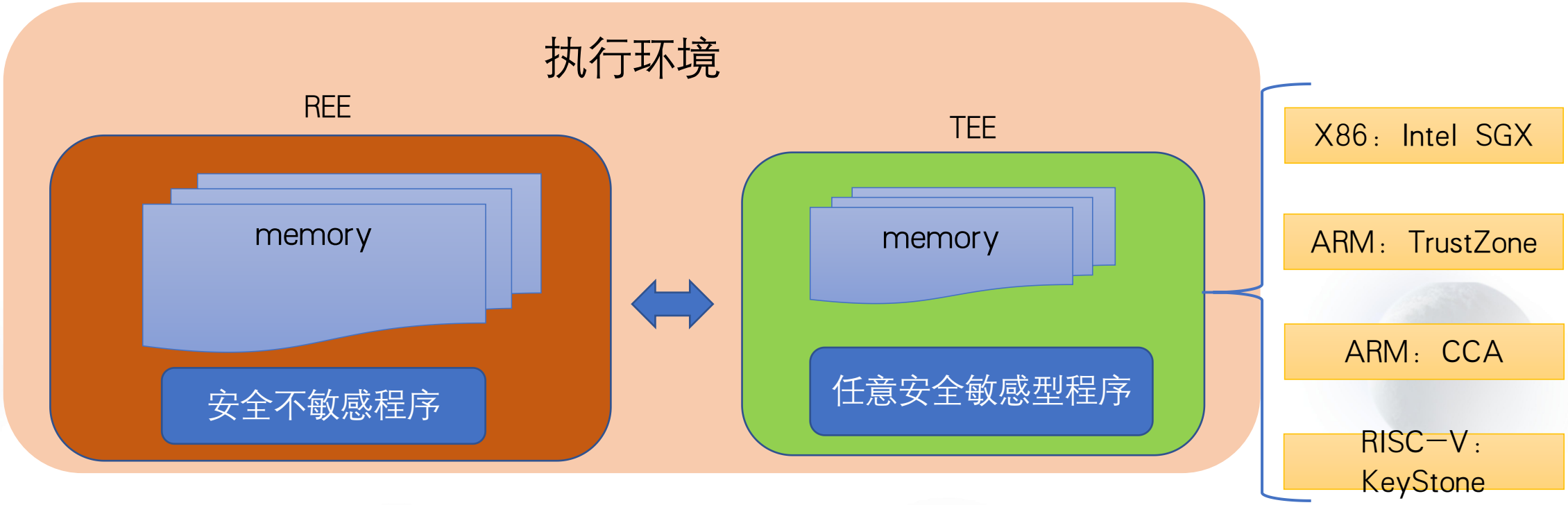
动机 - Log4j漏洞示意



- https://nvd.nist.gov/vuln/detail/CVE-2021-44228

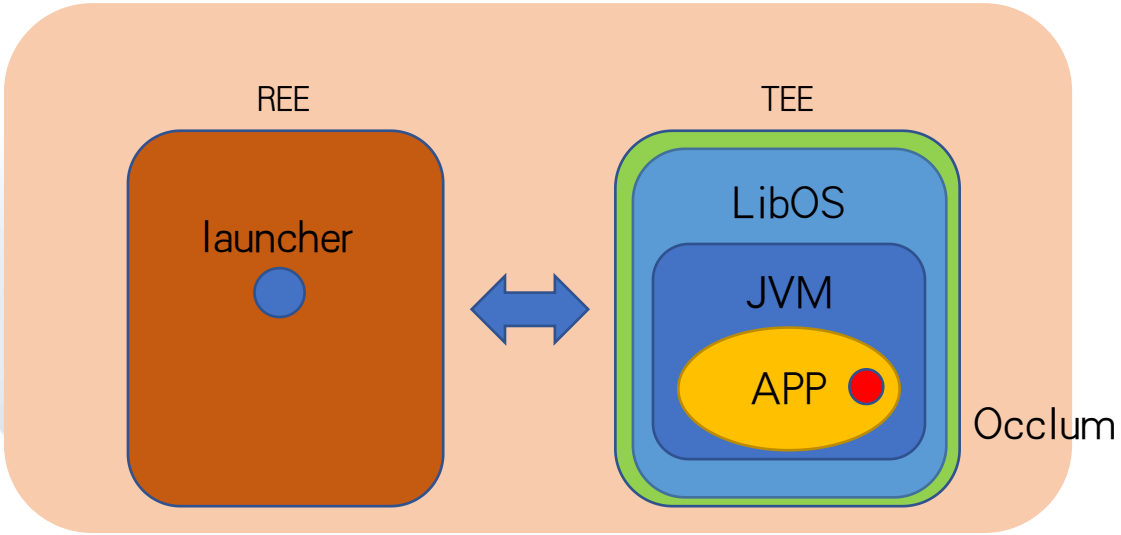
高等级的安全保障——机密计算

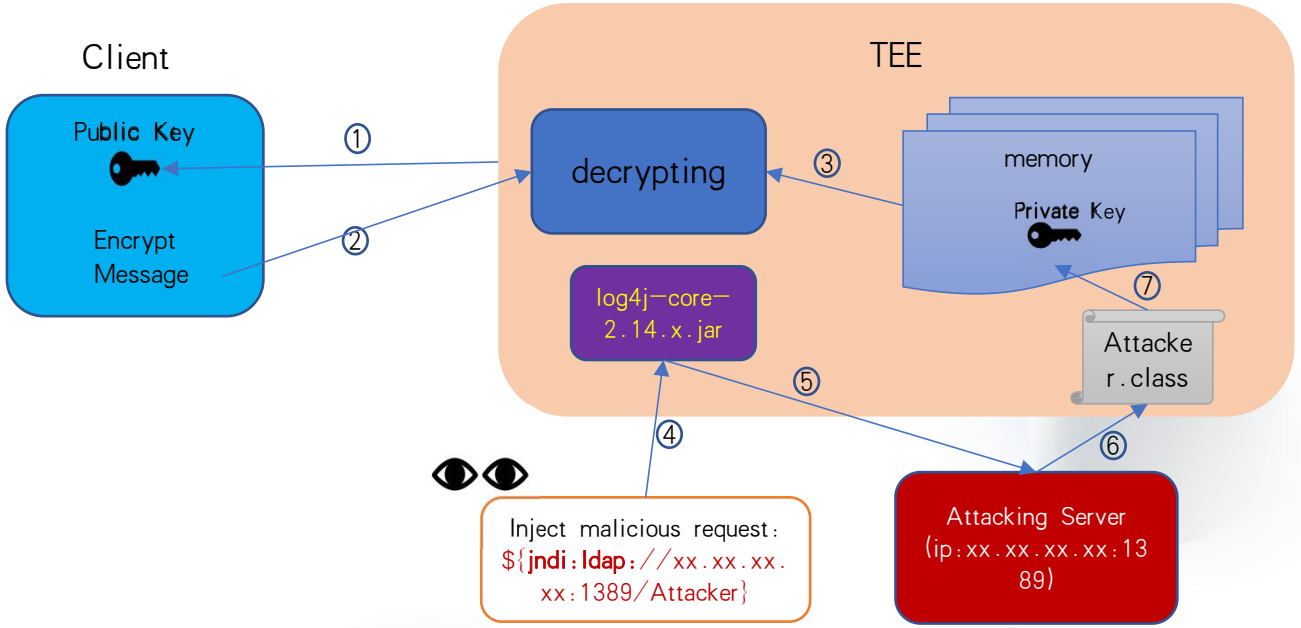
- 三大安全支柱：存储时加密、传输时加密、运行时加密
- 硬件隔离出安全与非安全环境，仅信任CPU，实现最高安全等级
- 用于多方安全计算、同态加密、联邦计算、区块链等诸多场景



在TEE中运行Java程序现状

- TEE中只能运行native程序
- 现有方案：Occlum，在TEE中支持JVM，再运行Java程序
- 存在的问题
  - 可信计算基（Trusted Computing Base，TCB）太大，安全性降低
  - 性能下降

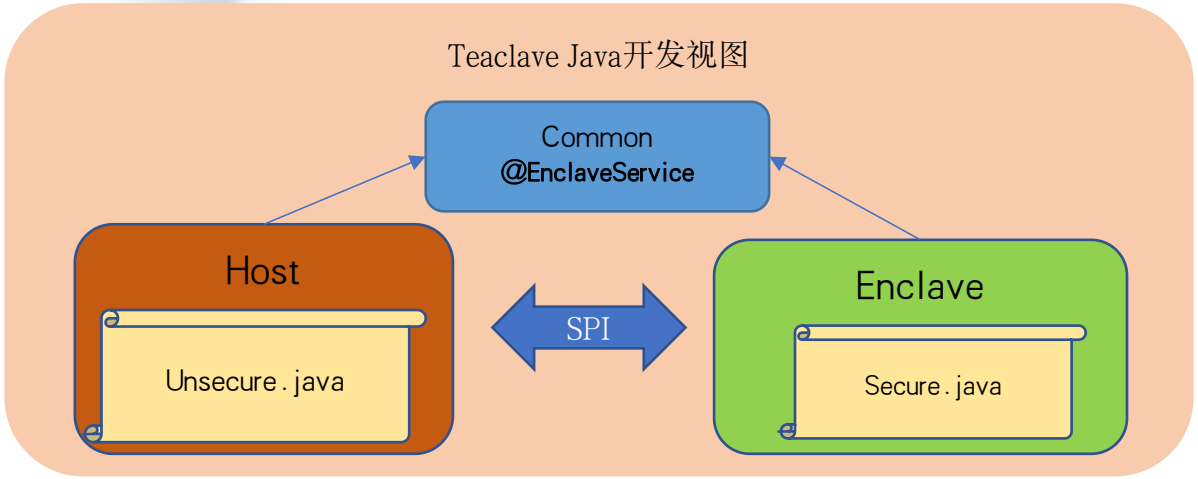




Teaclave Java TEE SDK

- Teaclave Java TEE SDK定义了Java机密计算的编程模型，实现了Java机密计算的开发框架和构建工具链
  - 运行时：从非机密Java程序通过JNI管理和调用TEE中的native机密程序
  - 开发时：使用Java编写非机密与机密程序。机密程序服务化，通过SPI机制调用
  - 构建时：Javac编译Java非机密程序，GraalVM编译Java机密程序为native动态库
- 获得成果
  - *Xinyuan Miao, Ziyi Lin, Shaojun Wang, Lei Yu, Sanhong Li, Zihan Wang, Pengbo Nie, Yuting Chen, Beijun Shen, He Jiang*. **Lejacon: A Lightweight and Efficient Approach to Java Confidential Computing on SGX.** ICSE 2023. Distinguished paper.
  - 正在Apache Teaclave社区开源孵化：https://teaclave.apache.org
    - 贡献单位： 百度、阿里云、蚂蚁、Intel
    - Teaclave Faas Platform：通用隐私计算平台
    - Teaclave SGX SDK：Intel SGX平台Rust语言SDK
    - Teaclave TrustZone SDK：ARM TrustZone平台Rust语言SDK
    - Teaclave Java TEE SDK：Intel SGX平台Java语言SDK https://github.com/apache/incubator-teaclave-java-tee-sdk

Teaclave Java编程模型 – 机密任务服务化



机密服务声明

```java
@EnclaveService
public interface AuthenticationService {
    /**
     * Given an encrypted input password, check if it is the correct password.
     * @param inputPwd the encrypted password to be authenticated
     * @return true if the given password is correct.
     */
    boolean authenticate(String inputPwd);
}
```

```java
public class Main {
    public static void main(String[] args) throws Exception {
        Enclave enclave = EnclaveFactory.create();
        Iterator<AuthenticationService> services = enclave.load(AuthenticationService.class);
        String pwd = "encryptedPwd"; // assume this is an encrypted password
        while (services.hasNext()) {
            AuthenticationService authenticationService = services.next();
            if (authenticationService.authenticate(pwd)) {
                System.out.println("Passed");
            } else {
                System.out.println("Rejected");
            }
        }
        enclave.destroy();
    }
}
```

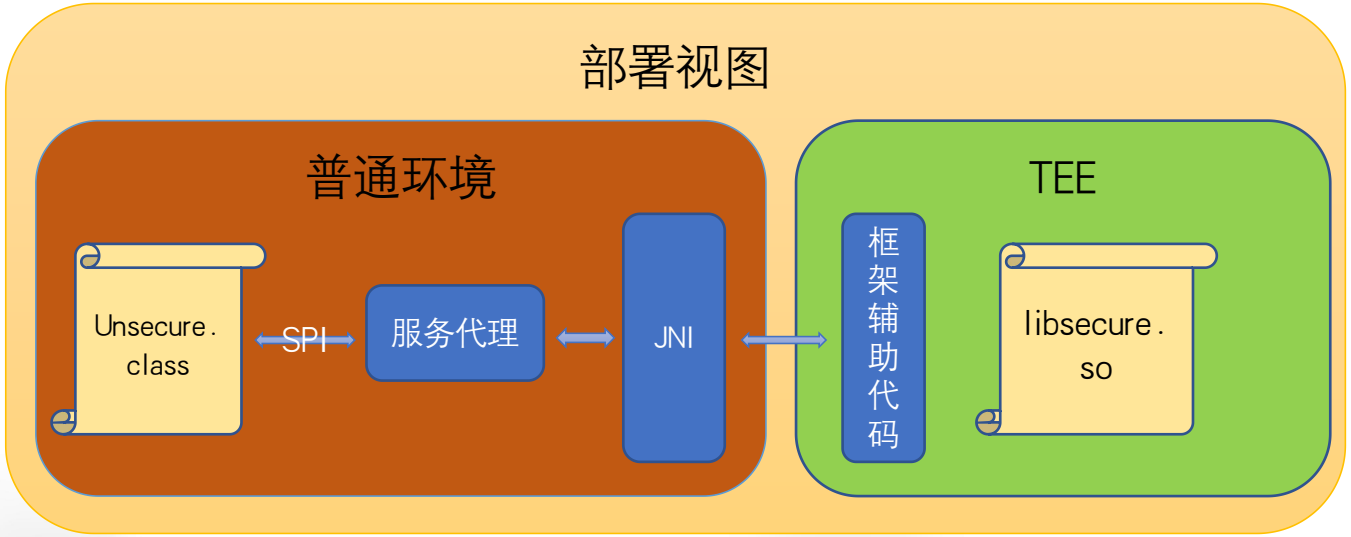
普通程序中调用服务

```java
@AutoService(AuthenticationService.class)
public class AuthenticationServiceImpl implements AuthenticationService {

    private String pwd = "somePwd"; // assume it's got at runtime.

    @Override
    public boolean authenticate(String inputPwd) {
        String decryptedInputPwd = decrypt(inputPwd);
        return pwd.equals(decryptedInputPwd);
    }

    private static String decrypt(String inputPwd) {
        return inputPwd; // assume it's decrypted with private key
    }
}
```

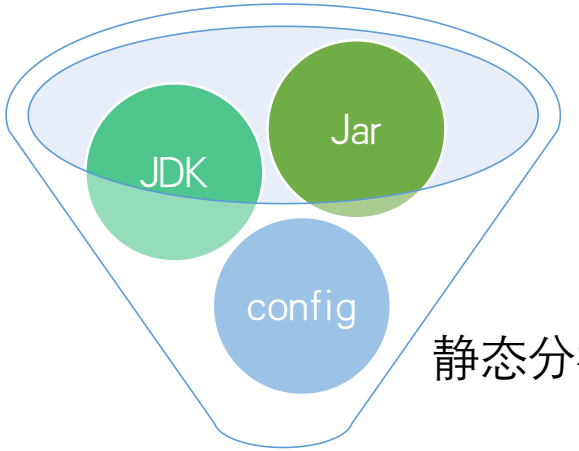
机密服务实现

Teaclave Java构建模型

- Maven一键构建
- 分别编译
  - 普通程序 -> javac编译部署到普通环境
  - 机密服务 -> GraalVM静态编译为动态库文件部署到TEE中
- GraalVM是由Oracle主导的开源纯Java实现的多语言高性能运行时平台
  - GraalVM编译器
  - SubstrateVM Java静态编译框架和运行时支持



GraalVM静态编译Java程序



静态分析（指向分析）

可达代码

优点

- 程序自动划分
- 有效降低TCB

编译

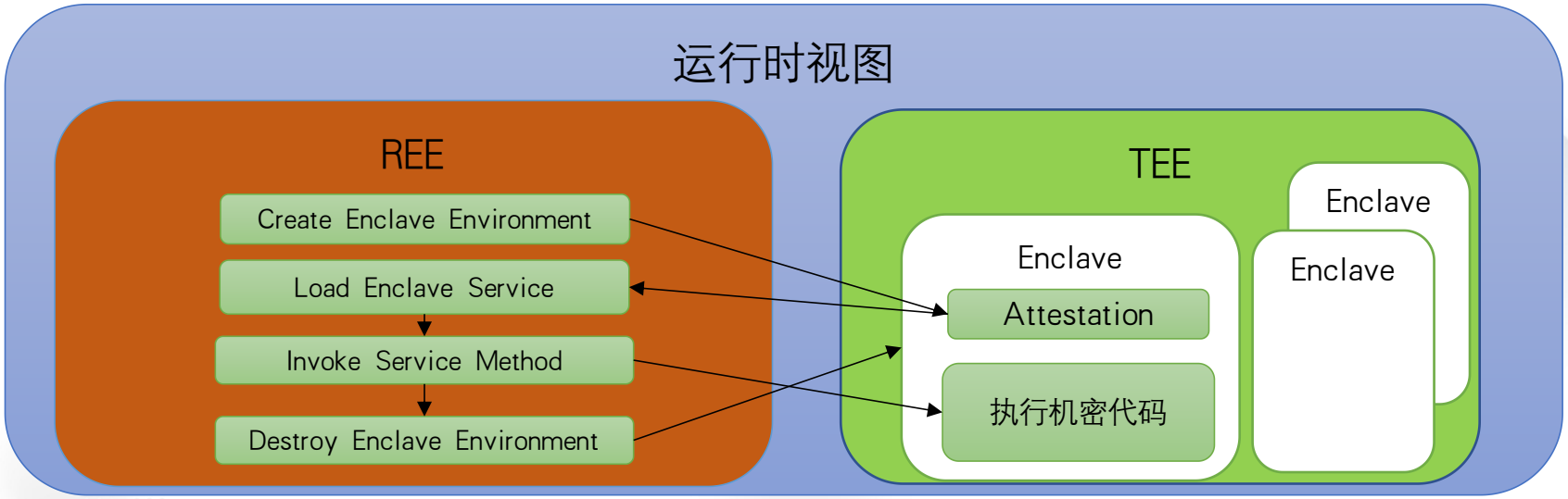
- 可执行文件
- Musl静态可执行文件
- 动态库文件

优点

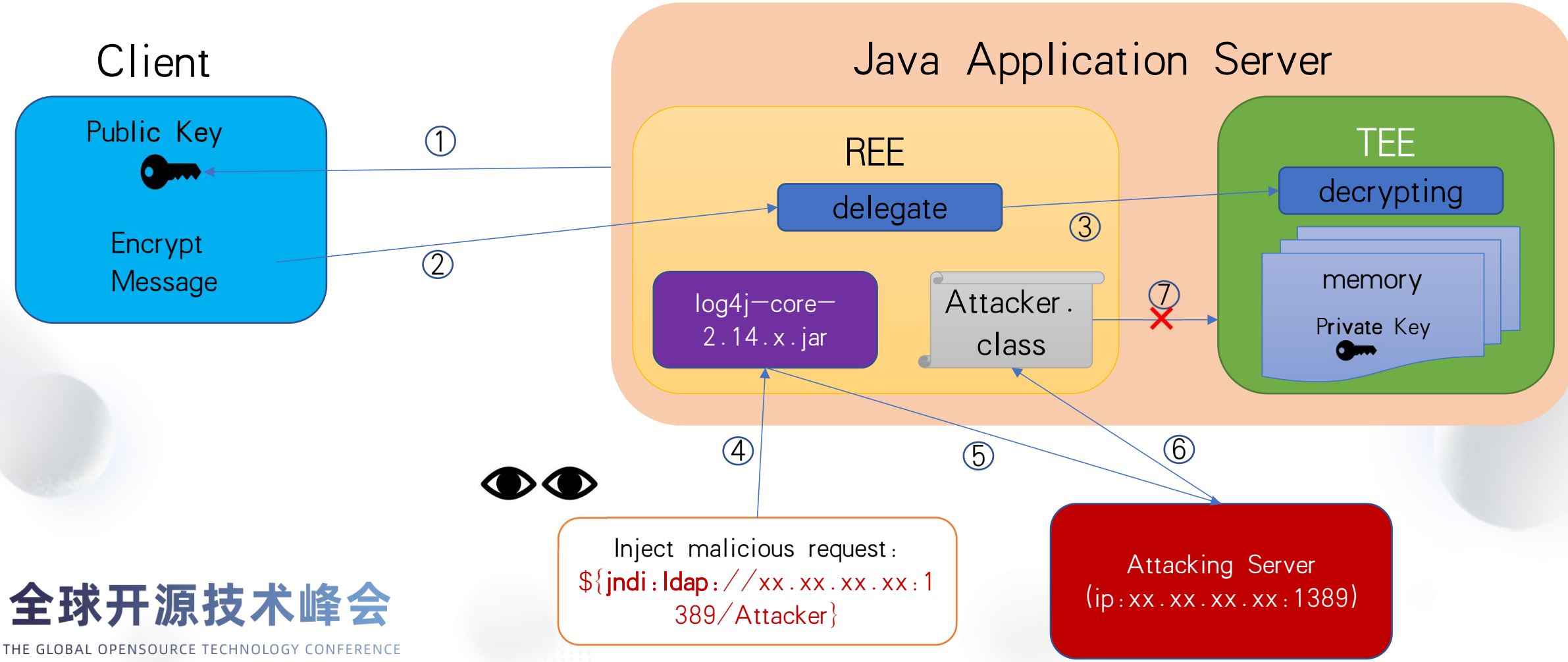
- 安全性提升：减少动态性
- 性能提升：无需解释执行和JIT编译

Teaclave Java机密计算生命周期管理

- Teaclave Java提供了机密计算任务的生命周期管理API
  - 创建Enclave环境：机密计算的运行时环境
  - 远程证明机密环境安全性：证明当前的SGX环境是真实可靠的
  - 加载Enclave服务：在Enclave环境中绑定机密计算服务
  - 调用服务函数：调用机密计算服务提供的函数，执行机密计算
  - 销毁Enclave环境：使用完毕，销毁环境，释放TEE资源
- 同一个Java应用可以在TEE中管理多个相互隔离的Enclave运行时环境



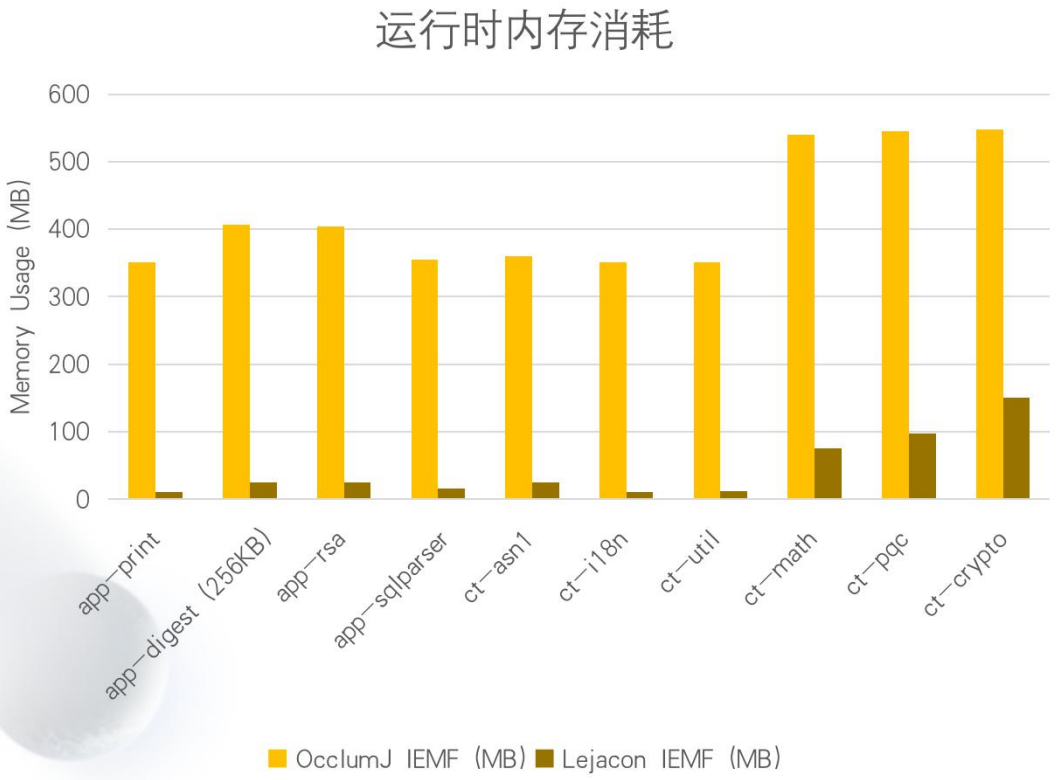
效果评估 - 抵御Log4J漏洞

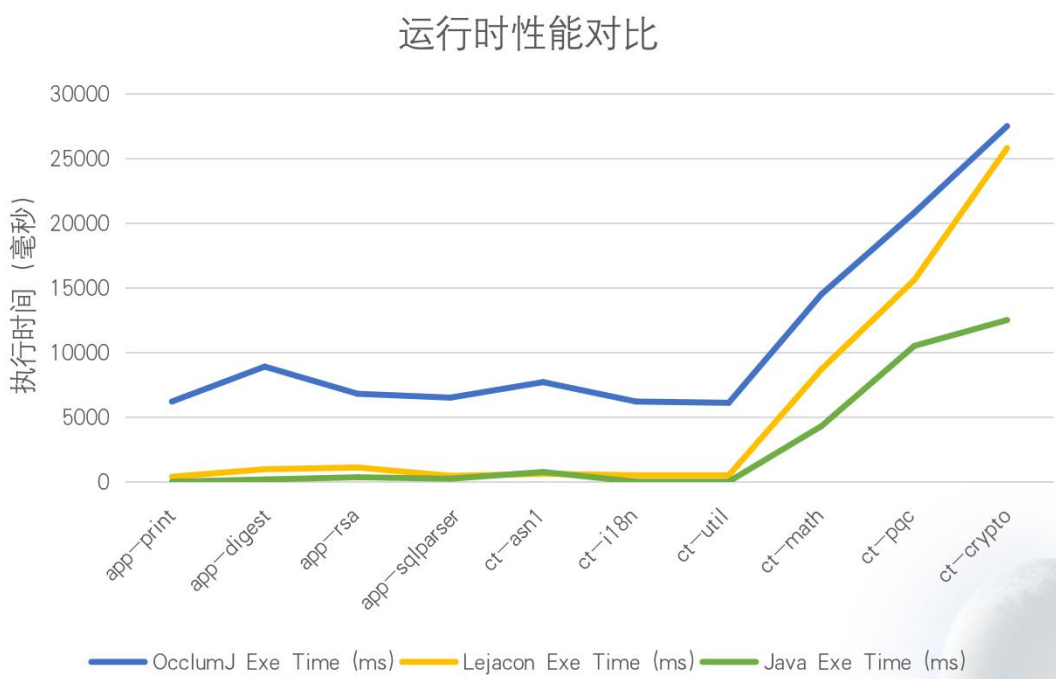


效果评估 - TCB

测试用例	依赖主要三方库	描述
app-print	\	打印一条消息字符串
app-digest	BouncyCastle-full	调用BouncyCastle计算hash值
app-rsa	BouncyCastle-full	调用BouncyCastle进行RSA加密
app-sqlparser	Druid	使用Druid进行SQL解析
ct-asn1	BouncyCastle-core	BouncyCastle-core的asn1子模块测试
ct-i18n	BouncyCastle-core	BouncyCastle-core的i18n子模块测试
ct-util	BouncyCastle-core	BouncyCastle-core的util子模块测试
ct-math	BouncyCastle-core	BouncyCastle-core的math子模块测试
ct-pqc	BouncyCastle-core	BouncyCastle-core的pqc子模块测试
ct-crypto	BouncyCastle-core	BouncyCastle-core的crypto子模块测试

效果评估 – 性能


运行时内存消耗
Memory Usage (MB)
600
500
400
300
200
100
0
app-print
app-digest (256KB)
app-rsa
app-sqlparser
ct-asn1
ct-i18n
ct-util
ct-math
ct-pqc
ct-crypto
OcclumJ IEMF (MB)
Lejacon IEMF (MB)


运行时性能对比
执行时间（毫秒）
30000
25000
20000
15000
10000
5000
0
app-print
app-digest
app-rsa
app-sqlparser
ct-asn1
ct-i18n
ct-util
ct-math
ct-pqc
ct-crypto
OcclumJ Exe Time (ms)
Lejacon Exe Time (ms)
Java Exe Time (ms)

缺少SGX硬件怎么办？

- 当没有SGX硬件时，Teaclave Java从机密计算退化为安全沙箱计算
- GraalVM编译的native image本身具有隔离性和自举性，是一个安全容器
- Native image内部不支持动态特性，无法通过反射和动态类加载对其攻击
- Java程序与native image之间内存隔离，运行时获取native image的内存状态具有一定难度

总结

- Teaclave Java是一站式的Java机密计算解决方案
  - 编码时：提供编程模型
  - 构建时：提供构建工具链
  - 运行时：机密服务生命周期管理
- Teaclave Java安全性更高、性能更高
- Teaclave Java在缺少SGX硬件时依然可以提供安全沙箱级别的保护
- 参考
  - *Xinyuan Miao, Ziyi Lin, Shaojun Wang, Lei Yu, Sanhong Li, Zihan Wang, Pengbo Nie, Yuting Chen, Beijun Shen, He Jiang*. **Lejacon: A Lightweight and Efficient Approach to Java Confidential Computing on SGX.** ICSE 2023. Distinguished paper.
  - Apache开源孵化链接：https://github.com/apache/incubator-teaclave-java-tee-sdk
  - GraalVM与Java静态编译原理与应用

我的微信

Teaclave Java @Github

中文版万字长文详解