

吴凌杰, 孟真, 严俊峰. 基于混合加密 API 算法对地震数据安全传输的研究[J]. 华南地震, 2022, 42(2): 46–52. [WU Lingjie, MENG Zhen, YAN Junfeng. A Study on Secure Transmission of Seismic Data Based on Hybrid Encryption API Algorithm [J]. South China journal of seismology, 2022, 42(2): 46–52]

基于混合加密 API 算法对地震数据安全传输的研究

吴凌杰, 孟 真, 严俊峰

(浙江省地震局, 杭州 310013)

摘要: API 给地震数据传输和共享提供了创新性的解决办法, 但其开放的属性给地震数据安全传输带来了隐患。为解决 API 传输地震数据的安全性问题, 提出了一种混合加密 API 算法。该算法基于 AES 和 RSA 加密算法, 并与 API 运行机制相结合, 实现了地震数据加密传输。研究表明: 混合加密 API 算法能在占用少量服务器资源且无须事先约定密钥的情况下, 快捷稳定地加解密地震数据并进行安全传输。

关键词: 混合加密; API; 地震数据; 数据安全

中图分类号: TP391

文献标识码: A

文章编号: 1001-8662(2022)02-0046-07

DOI: 10.13512/j.hndz.2022.02.07

A Study on Secure Transmission of Seismic Data Based on Hybrid Encryption API Algorithm

WU Lingjie, MENG Zhen, YAN Junfeng

(Zhejiang Earthquake Agency, Hangzhou 310013, China)

Abstract: API provides innovative solutions for seismic data transmission and sharing, but its open attributes bring hidden dangers to the safe transmission of seismic data. To solve the security problem of seismic data transmission by API transmitting, a hybrid encryption API algorithm is proposed, which is based on AES and RSA encryption algorithm and combined with the API operation mechanism to realize the encrypted transmission of seismic data. It is shown that the hybrid encryption API algorithm can quickly and stably encrypt and decrypt seismic data and transmit seismic data with occupying a small amount of server resources and without prior agreement of key

Keywords: Hybrid encryption; API; Seismic data; Data security

收稿日期: 2022-02-10

基金项目: 浙江省地震局科技项目(2020ZJJ01)

作者简介: 吴凌杰(1994-), 男, 硕士, 助理工程师, 主要从事地震信息化工作。

E-mail: 401018856@qq.com

0 引言

地震科学数据涵盖学科多、数据体量大、时间跨度长、动态性强和数据结构异构等特点,导致现存系统在体量、功能和性能方面难以满足数据安全共享需求^[1-2]。目前,应用程序接口(Application Programming Interface, API)做为移动设计、云计算、物联网、大数据及社交网络等应用数据交互的一种核心技术^[3],为地震数据传输和共享提供了创新性和颠覆性的解决办法。但是 API 的开放属性导致其很容易出现:用户账号密码被截获、表单数据被篡改和业务数据泄露等问题,极大地威胁到使用 API 传输地震数据的安全性^[4-6]。因此,如何使用 API 安全传输地震数据是当下亟需研究的一项课题。

目前保障 API 调用时数据安全的常用方案有通信使用 HTTPS 协议、请求签名、身份确认机制和对请求响应加解密等。HTTPS 协议^[7-8]的主要作用是确认双方的身份和建立安全通道,保障数据的安全传输。但是 HTTPS 不能防中间人攻击,使用成本高,占用服务器资源多而且响应速度慢。对于传输大体量数据和震情发生后的紧急情况,HTTPS 并不是最优选择。姜建武等^[9]使用双重签名验证、请求体加密等方式来加固用户数据传输安全性。请求签名在一定程度上可以保证数据传输安全性,但是在数据传输前需要提前约定密钥。由于地震灾害的突发性和不可预测性,应急情况下几乎没有条件在数据传输前就约定好密钥或者输入密钥。殷佳庭等^[10]使用请求和响应均加解密的办法,请求前数据先使用 AES 算法加密,接收数据后再用 AES 密钥解密,由于 AES 密钥的唯一性,密钥需要由服务端跟客户端事先约定,同样在地震应急场景中一定的局限性。

综上所述,针对地震数据体量大、对响应速度要求高以及无法事先交换密钥的应急需求,本文基于 RSA 和 AES 两种加密算法结合 API 运行机制提出了一种混合加密 API 算法,并将混合加密 API 算法与不加密 API 进行对比,探索两者在数据加密、API 性能和资源消耗等方面的差异并对差异产生的原因进行分析解释。

1 加密算法介绍

当前密码学体系中的加密算法一般分为对称加密算法和非对称加密算法两种^[11]。对称加密算法就

是使用相同的密钥对数据进行加解密。非对称加密算法是指加解密使用不同的密钥,通常此类算法有两个密钥,为公钥和私钥。发送方使用接收方的公钥对明文进行加密,接收方使用自己的私钥进行解密获得明文信息。公钥是可以对外界开放的,私钥只能由持有人知道。

1.1 AES 算法

AES 算法是当前使用最多的分组对称加密算法,分组指在加密和解密时把明文分成长度相等的几组。在 AES 的标准规范中,分组长度固定为 128 位。密钥的长度可以使用 128 位、192 位或 256 位,不同的密钥长度对应的加密轮数也不同^[12]。差异如表 1 所示。本文使用 128 位密钥加密。

表 1 AES 密钥长度和加密轮数
Table 1 Key length and encryption round of AES

密钥长度/位	推荐加密轮数/次
128	10
192	12
256	14

AES 加密过程涉及到四个环节,分别为轮密钥加、字节代换、行位移和列混淆。轮密钥加是将 128 位的密钥同状态矩阵中的数据进行逐位异或操作。字节代换采用的 S 盒是 16×16 字节的二维表,所有元素在整个加密过程中保持定值。行位移是做一个简单的循环移位操作。列混淆是将移位后的状态矩阵与固定矩阵相乘。

1.2 RSA 算法

RSA 加密算法是当前较为流行的一种公钥密码算法。算法的核心是运用一种产生复杂的、伪随机数据序列的模运算^[13]。RSA 算法是建立在“大数分解和素数检测”的理论基础上,两个大素数相乘在计算机上很容易实现,但是将该乘积分解为两个大素数因子的计算量却非常大。RSA 算法需要欧拉函数、欧拉定理和模反元素等基础数学知识。算法生成密钥的过程主要分为三步:① 随意生成两个大素数 $p, q (p \neq q)$, 令 $n = p \times q$; ② 根据欧拉函数性质可得 $r = \varphi(n) = \varphi(p) \varphi(q) = (p - 1)(q - 1)$; ③ 随机选择一个小于 r 的整数 e , 且 e 与 r 互为素数。根据模反元素两个正整数 e 与 r 互素,那么一定可以找到整数 d , 使得 $ed - 1$ 被 r 除,可求得 e 关于 r 的模反

元素 d 。可得公钥为 (n, e) , (n, d) 是私钥。最后销毁 p, q 。

得到公钥后, 需要对明文信息 m 进行加密, 将公钥和明文信息带入式子 $m \wedge e = c \pmod{n}$, 计算得到的 c 即为密文。解密时将密钥跟密文代入式子 $c \wedge d = m \pmod{n}$, 即可得到明文 m 。

1.3 混合加密API算法

AES算法将数据和密钥按字节来处理, 加解密运算速度快, 适用于大批量数据的处理。但由于很难解决密钥分发的安全性和数字签名等问题, 用于API加密传输时, 数据的安全性较弱^[14]。而RSA算法使用公钥加密私钥解密, 公钥可以向外界公布, 只要私钥不泄露, 很大程度能保障数据的安全性。但是RSA算法的短板在于加解密都是进行大数计算, 处理大体量数据会增加运算时间和设备性能消耗, 用于API加密传输会严重影响数据传输效率^[15]。为了解决AES算法密钥的短板, 同时弥补RSA算法处理速度慢的缺点。本文利用AES算法运算速度快的特性来处理需要传输的数据, 使用RSA算法密钥配置特性高的特点配置AES密钥, 同时结合通信网络的运行机制^[16], 提出了一种混合加密API算法。

算法具体流程为: 当客户端启动时, 发送请求到服务端, 服务端用RSA算法生成公钥 $publickey_s$ 和私钥 $privatekey_s$, 并将 $publickey_s$ 返回给客户端。客户端拿到服务端返回的公钥后, 用RSA算法生成公钥 $publickey_b$ 和私钥 $privatekey_b$, 并把 $publickey_b$ 交给服务端, 此时前后端通信建立。当客户端需要给服务端发送数据时, 客户端先生成128位密钥 key , 然后把需要发送的数据和

key 用AES算法加密, key 再用 $publickey_b$ 加密成 $aeskey$ 后和密文一并传输给服务端, 服务端给客户端发送数据亦是如此。在服务端与客户端建立通讯后, 前后端生成的 $publickey$ 和 $privatekey$ 不再改变, key 则会在每次传输数据前重新生成。服务端与客户端通信建立流程图如下图1所示, 数据加密传输流程示意图如图2所示。

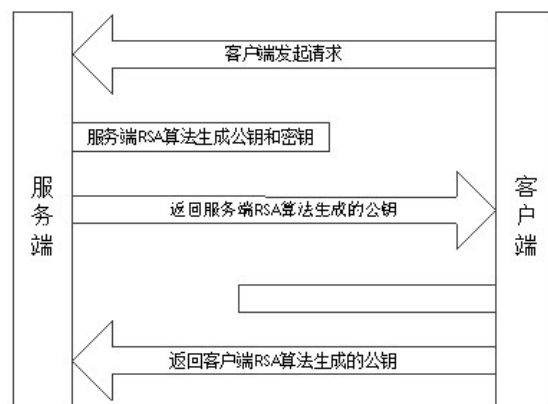


图1 服务端与客户端通信示意图

Fig.1 Schematic diagram of communication between server and client

2 数据来源

地震监测是地震学科的重要基础与核心业务。由于地球内部的不可入性, 目前只能在地表建设大量的观测台站来监测地震情况, 台站产生的数据再通过庞大的网络系统进行汇聚^[17]。因此地震科学数据主要来源就是台站的基础数据和观测数据。本文选用国家地震烈度速报与预警工程中台站监控系统的台站基础数据表来验证算法的加密效果。台站基础数据表结构见表2, 其中台站的位

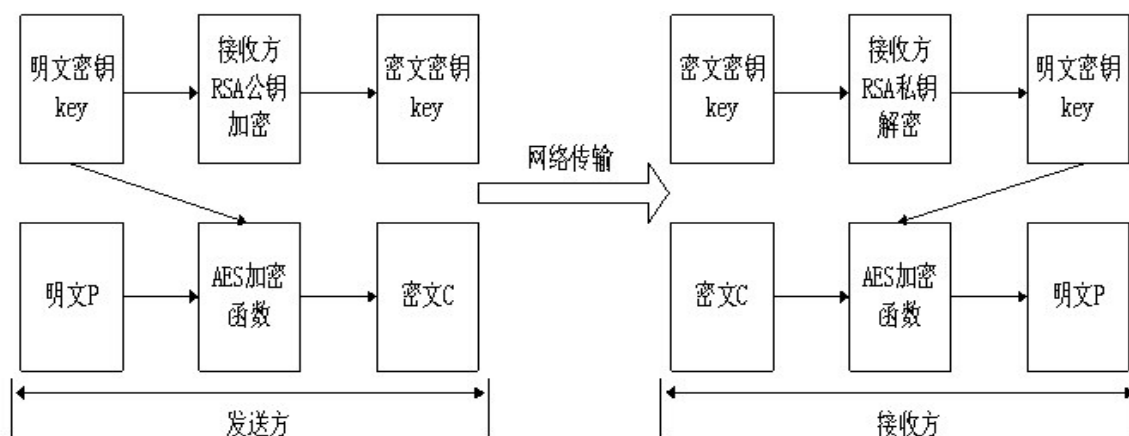


图2 数据加密传输流程图

Fig.2 Flow chart of data encryption transmission process

置和通信数据等都为敏感数据。同时选用浙江省地球物理台网海宁台垂直摆倾斜观测系统东西分量 24 h 的记录量作为测试 API 性能数据,该系统每秒钟记录一次数据,一天的数据量会打包在一起。

3 实验结果与分析

3.1 加密性验证

本文使用 JAVA 语言和 Springboot 框架实现混

合加密 API 和不加密 API 的功能。为了验证混合加密 API 能否在实际业务中有效的加密和传输数据。本文设置了如下交互方式,当服务端与客户端建立通信后,客户端以 HTTP 协议发起 POST 请求且请求携带参数为 data="stationimf ormation"时,服务端以 JSON 格式的台站敏感数据响应。实验过程中通过抓包软件抓取加密和不加密 API 请求和响应的数据。未加密 API 抓包数据如图 3 所示,加密 API 抓包数据如图 4 所示。

表 2 台站基础数据表
Table 2 Basic data of station

字段名称	实体	字段类型	字段长度
记录号	id	int	11
台网代码	net_code	varchar	5
台站代码	sta_code	varchar	5
中文台名	sta_cname	varchar	20
纬度	sta_lat	decimal	10
经度	sta_lon	decimal	10
高程	sta_elev	decimal	10
距本地地面深度	local_depth	decimal	7
传输接口方式	transmit_inter	varchar	32
IP 地址	host	varchar	32
端口号	port_number	int	11
掩码	instr_mask	varchar	32
网关	instr_gateway	varchar	32

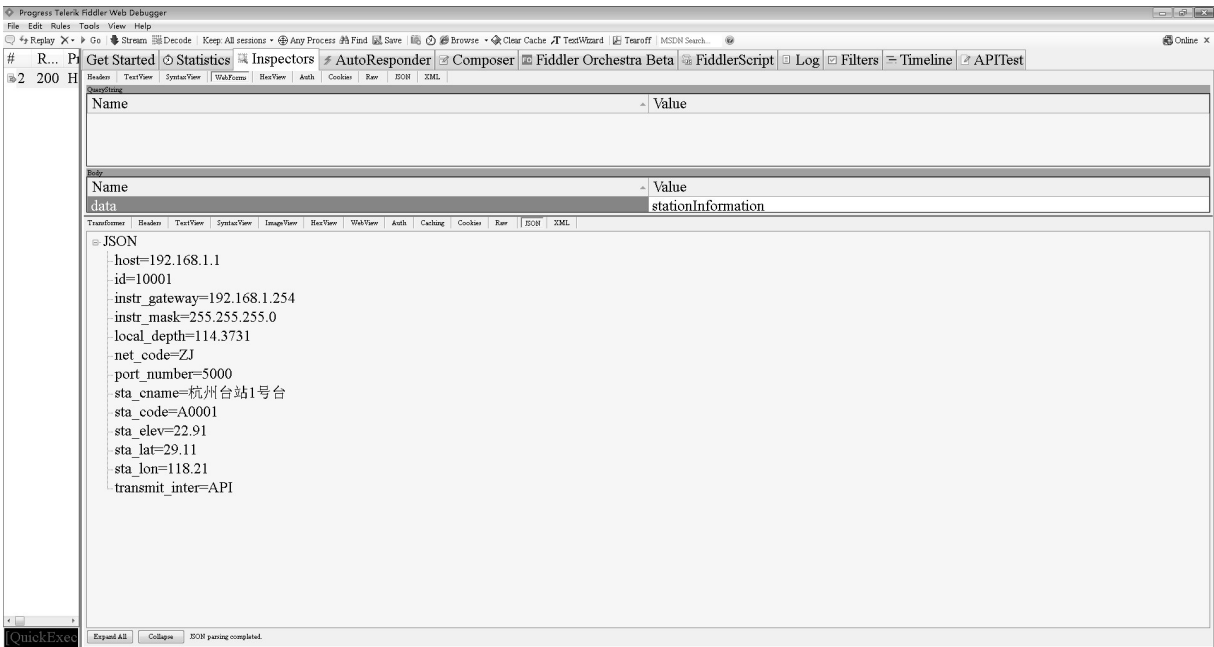


图 3 未加密 API 数据抓包图

Fig.3 Unencrypted API data packet capture diagram

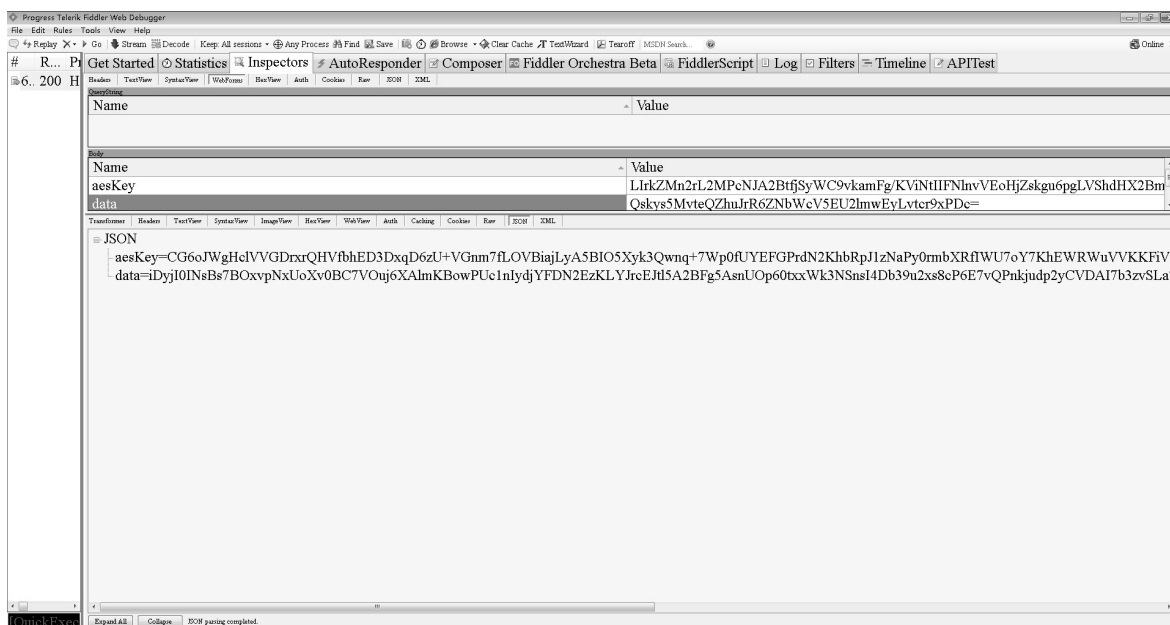


图4 加密API数据抓包图

Fig.4 Encrypted API data packet capture diagram

图中请求体的参数 Name 和 Value 分别表示客户端 POST 请求携带的参数名称和值, JSON 为客户端返回的数据。从图 3 可以看出, 未加密 API 客户端请求参数 data 和服务端返回的 JSON 数据都被抓到, 台站敏感数据一览无余。而图 4 中 aesKey 为使用后端公钥加密的 AES 密钥, data 为请求数据, 均为密文。返回的 JSON 信息也为密文。在无法获得密钥的情况下, 想要破解密文的难度非常大, 很好的保护台站敏感数据的安全性, 为地震数据安全传输提供了可靠的保证。

3.2 接口性能测试

为了测试混合加密 API 在实际业务中的接口性能表现, 本文使用由 Apache 公司基于 Java 开发的 API 性能检测工具 Jmeter 来做测试。该软件可以模拟用户对 API 发起请求和接受响应, 且可以编写 Java 脚本对数据进行加解密操作。由于 Jmeter 在做批量测试时, 无法先从服务端获取公钥, 因此在编写 Jmeter 测试脚本时, 需先将后端公钥写进脚本, 保证首次发起请求时客户端与服务端已建立通信。

同样以未加密 API 作为对照组进行测试, 测试使用 Jmeter 软件模拟用户以 HTTP 协议对 API 发起 1 进程 500 次的 POST 请求。发起请求时, 未加密 API 直接将台站观测数据作为请求参数发送, 服务

端接收后原样返回测试数据, Jmeter 接收数据后完成一次请求与响应。加密 API 则在发起请求前先将台站观测数据进行加密, 以密文作为请求参数, 服务端在获取密文进行解密后, 再将测试数据进行加密返回给 Jmeter, 等到 Jmeter 解密获得明文, 完成一次请求与响应过程。为减小网络因素给测试结果带来误差, 客户端与服务端运行在同一子网内。服务器为 32 GB 内存, 2.4 GHz CPU 的深信服云服务器。未加密与混合加密 API 请求响应时间图如 5 和图 6 所示。请求响应时间是指从客户端发送一个请求, 到客户端接收到服务器返回响应结果时长。

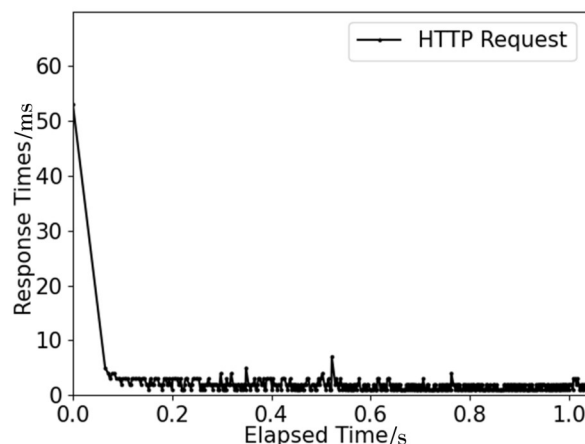


图5 未加密API请求响应时间图

Fig.5 Request response time graph of unencrypted API

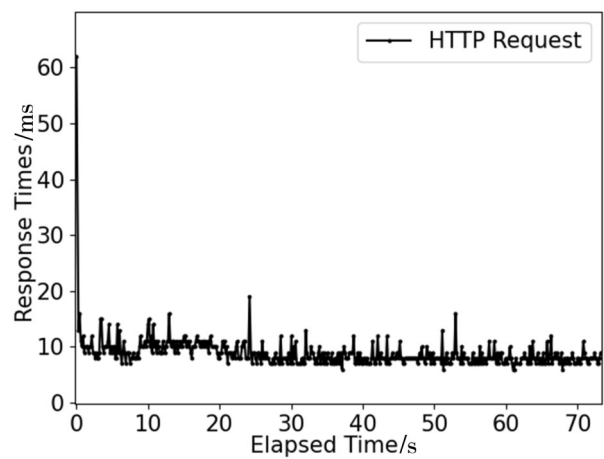


图6 加密API请求响应时间图
Fig.6 Request response time graph of encrypted API

从图中可以看到，首次发起请求的响应时间都是最长的，这是因为HTTP是建立在TCP之上的，

TCP握手会占用一定时间。从第二次开始，响应时间基本趋于稳定。但是混合加密API与未加密API响应时间存在差值。于是本文统计了所有请求的响应时间，如表3所示。

从表中可以看出，混合加密API运行稳定无失败。加密API平均响应时间比未加密API多6.97ms，在忽略网络等其他影响因素的情况下，这个时间即为服务器处理加解密算法的平均时长。统计发现，90%的响应时间差均在8ms以内，时间差几乎可以忽略不计。因此，混合加密API对地震数据传输速度产生的影响可以忽略不计。

由于地震系统有很多观测系统，在汇聚传输不加密数据时会对服务器的性能有较高的要求，因此接口的性能消耗也是一项非常重要的指标。于是本文还记录了响应过程中服务器的CPU和内存使用率分别，如图7和图8所示。

表3 接口响应时间表

Table 3 Response timetable of interface

接口类型	请求次数	平均响应时间/ms	最小响应时间/ms	最大响应时间/ms	90% 响应时间≤	成功率/(%)
未加密	500	1.86	1	53	3	100
混合加密	500	8.83	6	62	11	100

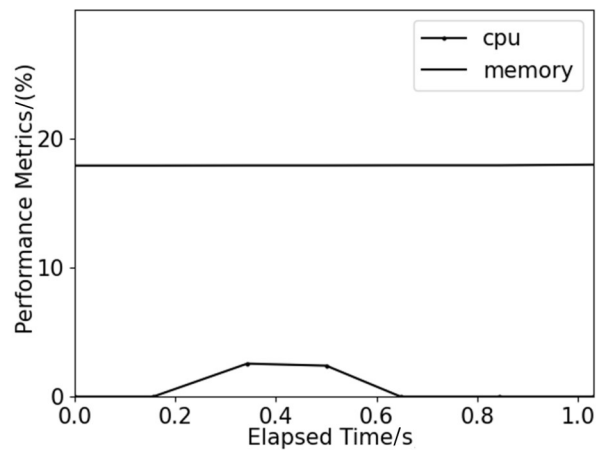


图7 未加密API运行性能图
Fig.7 Running performance graph of unencrypted API

对比内存使用量，两种接口对内存的消耗几乎相等，因此混合加密API对内存的影响和消耗不大。对于CPU性能来说，未加密API运行时CPU平均使用率约为0.71%，混合加密API为1.92%，相差1.21%，CPU资源占用量不多。

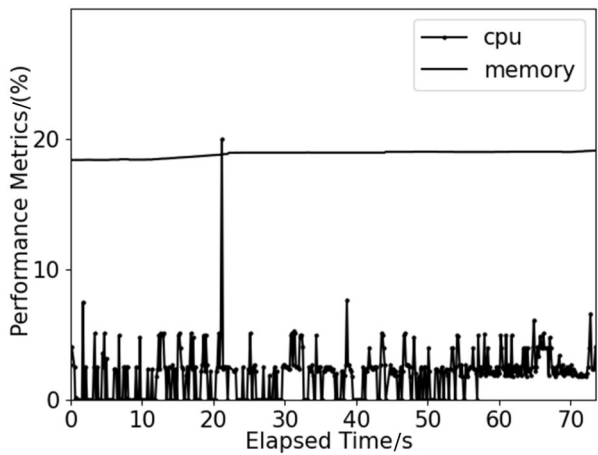


图8 加密API运行性能图
Fig.8 Running performance graph of encrypted API

3.3 局限性分析

网络攻防就像矛与盾，没有绝对锋利的矛也没有绝对坚硬的盾。本文介绍的混合加密API算法在一定程度上保障了地震数据传输的安全性和稳定性，但并不能完全阻止中间人的攻击，只能增加中

间人的攻击成本。此外,数据安全传输的方案有很多种,安全性的提高,必然也会带来成本、资源和时间消耗的增加。

4 结论

本文以 AES 和 RSA 加密算法为基础,结合 API 运行机制,将两种算法优势互补取长补短,构建了一种混合加密 API 算法。通过与不加密 API 进行比较,分析了两种不同地震系统数据,研究了混合加密 API 在地震数据传输的安全性、稳定性、快速性、适应性和资源消耗方面的特点。研究结论主要有以下几点:

(1)混合加密 API 算法能有效地对 API 传输的数据进行加密,在一定程度上保障了地震数据的安全性。

(2)在实际业务中,无需提前交换密钥,混合加密 API 算法就能高效稳定地运行,保障了应急情况下地震数据的安全传输。

(3)混合加密 API 算法对地震数据的传输速度不会产生影响。

(4)混合加密 API 算法要对数据进行加解密计算,对计算机的 CPU 资源有一定的消耗,对内存几乎无消耗。总体来说,只占用了少量的服务器资源。

参考文献

- [1] 谢有顺,李盛乐,刘小利,等.面向地震行业的地震数据共享服务平台的设计与实现[J].大地测量与地球动力学,2017,37(05):546-550.
- [2] 吴微微,陈学芬.大数据下地震科学资源服务平台建设的探索[J].国际地震动态,2018(08):156-157.
- [3] 石宏.一个正在被 API 驱动的互联网时代[J].计算机与网络,2018,44(04):44-45.

- [4] 崔喜萌,陈明.软件开发开放 API 接口的安全处理[J].网络安全技术与应用,2020(04):80-81.
- [5] 张玉清,王凯,杨欢,等.Android 安全综述[J].计算机研究与发展,2014,51(07):1385-1396.
- [6] 张聪,房鼎益,王怀军,等.一种基于 API 安全属性隐藏的软件保护方法[J].山东大学学报(理学版),2015,50(01):12-19.
- [7] Yadav RK. Man in middle attack in SSL and HTTPS[J]. International Journal of Computer Science and Mobile Computing,2015,4(5):566-573.
- [8] 赵国锋,陈勇,王新恒.针对 HTTPS 的 Web 前端劫持及防御研究[J].信息网络安全,2016(03):15-20.
- [9] 姜建武,胡焱,李景文.基于 Request Body 的 Open API 安全认证机制[J].科学技术与工程,2019,19(19):196-200.
- [10] 殷佳庭,陆婷婷.基于混合加密算法的数据接口安全研究与实现[J].集宁师范学院学报,2019,41(06):41-44.
- [11] 雷敏.实用信息安全技术:信息安全技术[M].北京:国防工业出版社,2014.
- [12] Pradhan C, Bisoi A K. Chaotic variations of AES algorithm[J]. International Journal of Chaos, Control, Modelling and Simulation, 2017,2(2):19-25.
- [13] 陈传波,祝中涛.RSA 算法应用及实现细节[J].计算机工程与科学,2006,28(9):13-14.
- [14] 蔺小梅,李国刚,张泽普.采用 OHNN 和 M-LFSR 的字序列密码加密方案[J].华侨大学学报(自然科学版),2014,35(05):519-522.
- [15] 许柯,刘绪崇,符振艾,等.网络信息加密 RSA 算法的运算速度和保密性优化[J].科技通报,2015,31(07):144-147.
- [16] 冷飞,徐进华,栾仕喜.RSA 融合 AES 算法的网络信息安全方法[J].华侨大学学报(自然科学版),2017,38(01):117-120.
- [17] 朱宏,王晓磊,刘磊,等.天津市地震局数据监控与展示系统的设计与实现[J].地震工程学报,2016,38(S2):231-235.