(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115996116 A
(43)申请公布日 2023.04.21

(21)申请号 202211481481.3

(22)申请日 2022.11.24

(71)申请人 道和邦(广州)电子信息科技有限公司

地址 510440 广东省广州市白云区嘉禾街鹤龙二路96号粤旺大厦C栋305室

(72)发明人 陈书增

(51)Int.Cl.

H04L 9/08(2006.01)

H04L 9/32(2006.01)

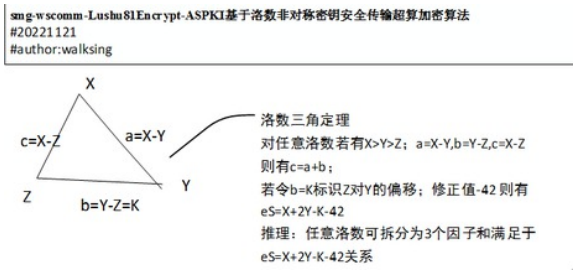
权利要求书4页 说明书7页 附图7页

(54)发明名称

smg-wscomm-Lushu81Encrypt-ASPKI基于洛数非对称密钥安全传输超算加密算法

(57)摘要

Lushu81Encrypt-ASPKI本发明可实现对任意对任意节点通过部署内置相同的wscomm.jar模组配置SPK(keyX,keyX)密钥对,可实现跨节点非对称密钥安全超速通信,实现超高安全性,高可靠性信息保密压缩传输;而在通讯加密、解密分别采用不同的密钥进行操作,本发明采用非对称算法基于洛数三角定理可实现超高速密钥的安全交换,缩短通讯握手流程,加快信息高保密信息互联互通,可广泛运用于军民俩用,高效率加密解密技术为超算力输出,超分布式存储输出必放一异彩。



CN 115996116 A

1.一种基于基于洛数81九宫方阵推衍的高安全高效高压缩数据的超算加密算法(参专利公布申请号:CN202210772306.3)可跨节点预置的jar库装备,其特征如下:

1.1其内部主要特征方法如下封装包Luoshu81Encrypt

eS=d3encrypt(K,Y,X),

X=d3decrypt(K,Y,eS)

Y=d3ydecrypt(K,X,eS)

K=d3kdecrypt(X,Y,eS)

S=hashLuoshu(S0) 自定hashLuoshu算法

X=d3decryptHash(keyhashK,keyhashY,eS) 解密基础函数

keyhashY=d3hashY(keyK,KeyX,eS) 逆向通过K,X,eS反推求的keyhashY即Y的hashLuoshu密钥

keyhashK=d3hashY(keyX,KeyY,eS) 逆向通过X,Y,eS反推求的keyhashK即K的hashLuoshu密钥

int s=d3e(int x,int y,int k) 逻辑单元对x,y,k加密算法封装为s

int x=d3x(int y,int s,int k) 逻辑单元解密x

int y=d3y(int x,int s,int k) 逻辑单元解密y

int k=d3k(int x,int y,int s) 逻辑单元解密 k

注:这里的d3代表direct 3 三个方向之意。

2.洛数三角定理及运用特征如下(参专利公布申请号:CN202210772306.3):

任意三个九宫洛数用线连接起来组成三角形,即X,Y,Z (满足于X>Y>Z)

若a=X-Y,b=Y-Z,c=X-Z;则有c=a+b.

证明:依命题,a+b=>(X-Y)+(Y-Z)=X-Z,而c=X-Z故a+b=c成立,由交换律c=a+b亦成立,证毕;本发明依据洛数三角定理,已知X,Y;设Z=Y-K;令eS=X+Y+Z,则有

eS=X+Y+Z=X+Y+(Y-K)=X+2Y-K

eS=X+2Y-K 实际运算中采用修正值去掉ascii码低32码(控制符),并在实际编码中为保持Y>Z,Y采用+20,Z采用+10补位修正运算,故修正差为-42

这里的K标识Y与Z的偏移

实际公式为eS=X+2Y-K-42

d3e={eS=X+2Y-K-42;return eS;}

d3x={X=eS-2Y+K+42;return X;}

d3y={Y=(eS-X+K+42)/2;return Y;}

依据洛书三角定理,X,Y,K共同组合成eS;知道任意3者可确定另一者;对任意数字eS可拆分为3个数X,Y,K,满足于eS=X+2Y-K-42;

为防止暴力破解实际运算引入了加盐K修正算法,并采用了流密钥hashLuoshu分组算法,即单元流eS=d3e(K,Y,X) 每个单元的dk值偏移不同,导致破解困难;dk单元值变化依据密钥K=hashLuoshu(K),不知道K是无法破解方程组;且对任意四元一次方

程,知其2者,则有无数方程解,知其3者,则有唯一方程解,这是本发明算法安全基础;

非对称共享密钥基础:

对任意K,X可事先预内置在通信服务器节点服务器配置文件内,并加以签名保护,ASPK

安全通讯分俩阶段协议进行:

sessionOnASPK 函数被代理封装实现了aspk1,aspk2 二阶段握手协议封装于ioserver-server.jar,ioserver-core.jar,ioserver-client.jar 客户端、服务端均集成sessionOnASPK函数在通讯连接建立时智能代理并建立非对称密钥安全握手连接;在通讯消息发送时自动检测并采用对称加密、解密,ioserver-core底层封装了消息过滤器智能过滤aspk协议指令;

协议过滤器被内置于VMESever 中

public static final String aspkFilter="\\&(act=aspk|msgtype=echo|ver=)";

标识该匹配的字段不加密采用私有处理规则;

2.1第一阶段STEP 1 :

协议头字段 &act=aspk&step=1

C客户机,向S服务器发起连接请求,并随机生成OTP一次会话密钥Y;以配置X,K

调本算法eS=d3encrypt(K,Y,X), 发送密文eS给服务器S;

S:收到ASPK协议请求并收到密文eS;调用keyhashY=d3hashY(K,X,eS)进行解密得到Y的hashLuoshu 基密钥(注:不可能完全得到Y密钥,因hashLuoshu不可逆);

拿到基密钥即可调基算法实现解密, X0=d3decryptHash(keyhashK,keyhashY,eS) 然后S端比较X==X0如果为真则证得C客户端与S端拥有相同的K,X,证明客户端可信;

注1:以上X,K不变求得Y hash密钥;

ASPK协议阶段1:客户端C生成新的Y,用K,Y加密X得eS, 只是泄露了eS发送到服务端S;用K,X,解密Y基密钥再解密X以校验客户端安全性;

2.2第二阶段STEP2:

协议头字段 &act=aspk&step=2;

S服务端随机生成OTP临时密钥X=$skey, 调本算法eS=d3encrypt(K,Y,$skey), 发送密文eS给服务器S;注这里的Y即可以是Y密钥亦可传递Y的hashLuoshu基密钥在底层统一转为Y基密钥进行处理;C端客户机收到密文eS 且step=2,用Y,K,eS 调X=d3decrypt(K,Y,eS)

解出密钥X并为16位数字,可验证服务器可信并用此X作为通信双方的安全临时OTP密钥,仅限于本次通信会话;

注2:以上Y,K不变,传递变化的X ;

2.3基于2.1,2.2 以skey作为X,在K,Y不变的前提下加密X得eS并发送(泄露eS)eS给客户端C,C端解密完成服务端校验,并成功传递了OTP密钥。

3.内置Luoshu81超算加密算法wscomm.jar特征如下:

服务器S,与客户机C 必须配置相同的密钥对X,K用以验证双方通信第一次口令Y,

如下:X,K

aspk.keyX=2022112213412197

aspk.keyK=2022091203059093

(实例可随机定义,实际发布不公开)

aspk代表非对称共享密钥协议

keyX其本质相当于2 中的X;

keyK 其本质相当于2中的K;

只不过这里对X,K进行了流密钥分组由整数转为流密钥字串序列

但整数易破解,长流密钥序列难以破解,K,X相当于集成算法不公开,基层API wscomm.jar 可通过自实现(c++)(ByteCoderEncryptor.so/dll)防编译库,并对产生jar采用virbox Protector对虚拟机java/jar进行2次保护,对jar加密并签名运行保护以实现核心api其安全性加载。

4.一种超高速非对称密钥安全传输算法,其特征如下:

4.1 本发明利用了九宫洛数三角定理规律约定了通信双方预置共享密钥K,X对以共享密钥对为基础进行校验发送者密钥,并校验发送者与接收者是否具有共享密钥对;一方以密钥Y,K对X加密eS;另一方以K,X对eS进行解密产生Y基密钥(hashLuoshu),而同时不泄露密钥Y,此为一种典型的非对称密钥加密解密方案;本算法中X,Y,K共同组合成为eS实际传输(仅暴露)eS

四元一次方程知3个未知量,有方程唯一解;知小于3 未知量有无数方程解,导致求解困难,间接证明算法的安全性;

4.2 基于4.1 本发明仅暴露1个未知量eS;解方程需要3个未知量;故导致求解困难;反证算法安全性;

4.3 基于4.1 为防止暴力破解,任何人知道算法,但不知道K基密钥序列导致破解困难;破解困难的原因是随着破解增加,解密的明文字元与密文字元混杂一起,而攻击者已经不能正确区分哪些是明文,哪些是密文字元;而只有正确的密文K序列才能破解出正确的明文序列;K为共享密钥对不传递;无从获取;

4.4 基于2.1 在ASPK协议第一阶段 发送者C利用的内置SPK(keyX,keyK) 自己随机生成keyY ,用eS=d3e(K,Y,X) 对X加密得到密文eS;K,X不变;变化了Y目的传递Y的密钥给接受者S;

但S通过共享密钥对SPK(keyK,keyX),Y=d3hashY(K,X,eS)可解密得到Y的基密钥;而2.1加密的本质是对Y,K基密钥(hashLuoshu)进行,拿到基密钥变相于拿到Y;然后X1=d3x(K,Y,eS)比较X1,X;如果发送者C,与接受者S内置相同的K,则X一定是相同即反证C,S是持有相同密钥的可信服务器 ;

4.5 基于2.2 接收者S在验证发送者C;利用4.4得到的Y基密钥(hashLuoshu),用eS=d3e(K,Y,X) 这里X=$skey=$X2即新产生的OTP密钥;此时公式算法相同,但保持K,Y不变,变化了自变量X,发给发送者C,此时C得到eS,step=2阶段协议,因为自身具有K,Y可利用函数X2=d3decrypt(K,Y,eS),解密后X2即本次通讯的对称加密OTP口令skey;

4.6 基于4.4,4.5 在双方通讯中关键口令Y,与X2->$skey均不涉及明码传递,期间以密文被嵌入eS传递,eS有无数方程解也即等于无解;攻击者必须同时具备eS,K,(X/Y) 掌握3个未知量方能破解;而这是不可能;从而反证算法安全;

4.7 基于4.4,4.5 可知只需要经过两步通讯,实际发送->接收(校验)->发送 3步完成服务器相互验证,密钥传递;简化复杂流程;同时基于2 洛数三角定理不涉及复杂乘法仅用到加、减法运算,从算法上提高几个数量级,是轻量级运算,提高加密、解密计算速度;

4.8 基于 1.1 本发明实现基于洛数超算加密、解密,自定hashLuoshu杂凑算法,并支撑数据高倍压缩安全传输;

实际运算中,如果原文X小于32则会自动补位运算,补位目的是尽可能遍历hashkeyY,

hashkeyK,在逆运算中能够完整的算出对应的hashkey基密钥;故密文比原文长;实际测算正常密文压缩率约为123%,洛数对称加密压缩率109%;洛数非对称加密效率较洛数对称加密快1.5倍;

4.9 基于3,4本发明可用于并行、分布式系统下多节点间,核心密钥超算加密安全传输,以及可成为军用、民用领域超高、超快,超安全通讯核心集成方案。

smg-wscomm-Lushu81Encrypt-ASPKI基于洛数非对称密钥安全传输超算加密算法

技术领域

[0001] 本发明为一种新型颠覆性超速非对称密钥安全传输超算加密算法Lushu81Encrypt-ASPKI该发明可解决并行,分布式计算机节点之间需要进行安全通讯问题,通过集成签名wscomm API内置共享密钥SPK(KeyX,keyK)可配置,并对api进行读写签名保护,防反编译保护,完成共享密钥部署;通过建立通讯链路对双方进行签名验证双方节点信任问题,进而进行交互临时OTP通讯密钥完成完全通讯,通过推理洛数超算加密Luoshu81Encrypt(现今世界第一对称超算加密),并基于Luoshu共享内置wscomm.jar api,只要双方节点都内置wscomm.jar模组,双方即可按本发明集成方案完成安全通讯,并对双方进行可信验证;而无需泄露SPK,该发明为基础api应用服务,可应用在所有领域,无论是信息安全,还是数据压缩,快算,超算等基础应用,极大的加快信息的安全共享服务,与基础通信保密服务,经过测算Luoshu81Encrypt d3encrypt非对称密钥算法较Luoshu81Encrypt encrypt对称密钥算法快1.5倍,数据压缩率123%,而Luoshu81Encrypt对称算法数据压缩率109%(中英文混合线上实测数据);本发明Luoshu81Encrypt非对称密钥算法主要用来进行OTP密钥安全传输与交换;中途只泄露密文eS,需要双密钥(keyY,keyK)才能解密,对于任意洛数根据洛数三角定理可知道,可拆分为eS=X+2Y-K-42(X>Y>K);据此公式为四元一次方程,任意3个未知量确定,可确定一个方程唯一解;任意小于2个未知量确定,方程有无数解;这是本发明的基础,也是本发明算法安全保障;采用本发明可取代非对称加密算法中大数分解PKI加密基础设施方案,现有非对称PKI技术算法复杂,加密速度慢;本发明颠覆性解决了两大难题;本发明可防统计频率攻击,同一字符在不同顺序标识不同含义,可防密钥数字差攻击,密钥字元在不同位置含义不同,一个数差,即解密失败。

[0002] 本项发明,为SMG-VME可迭代分布式操作系统,SMG-VME-AFS可迭代分布式存储系统延伸的价值,iSBS/mbs可迭代超级商城动态令牌技术的应用基础,主要应用于信息的高倍编码,高效率的加密基础超算应用,本发明重点用于更为安全的并行节点间信息安全传输,以及超算加密。

[0003] 本项发明,为SMG-VME系列下软件工程实现的社会使命,其目标为解决超算,超存储输出,以及信息的安全传输,安全获取。

[0004] 本项发明 依赖于前专利:Luoshu81Encrypt,幻数MagicNumberEncrypt,Base84进制高效率编码技术

专利审查:

截至2022-11-21网络搜索暂无同案例

背景技术

[0005] 本发明主要应用在工业互联网,万物智能互联,分布式计算,万物互联数据信息的加密交换,信息的高速运算。本发明最初用于iSBS/mbs可迭代超级商城用户动态令牌ECToken的加密领域,ECToken动态令牌技术(详见前专利);

本发明已经部署于SMG-AFS可迭代分布式存储跨节点的数据的安全传输;具体表现在AFS并行数据节点在Socket层封装了非对称密钥安全交换协议ASPK,基于TCP/IP本发明自实现了安全协议层TLS,节点建立连接客户端内置sessionOnASPK (IoSession,String s) 代理连接发起安全通信ASPK协议,C&S双方服务器完成2阶段协议握手,完成双方节点可信认证,并协商OTP $skey口令安全传输,对底层数据采用Luoshu81Encrypt对称算法进行加密,进行密文传输。

发明内容

[0006] 见图1-9

附图说明

[0007] 图1是洛数三角定理示意图;

洛数三角定理对任意洛数若有X>Y>Z;a=X-Y,b=Y-Z,c=X-Z 则有c=a+b;若令b=K标识Z对Y的偏移;修正值-42 则有eS=X+2Y-K-42 推理:任意洛数可拆分为3个因子和满足于eS=X+2Y-K-42关系

图2 wscomm.jar Luoshu81Encrypt 洛数加密函数功能图;

300 wscomm.jar 加密算法api核心包

100 Luoshu81Encrypt 洛数超算加密类(含对称/非对称)

101 encrypt/decrypt 对称加密、解密

102 eS=d3encrypt(K,Y,X) 非对称加密

103 X=d3decrypt(K,Y,eS) 非对称解密

104 SPK aspk.keyX,aspk.keyK 共享密钥对

105 hashLuoshu杂凑算法

110 MagicNumberEncrypt 幻数超算加密

120 Base84编码

图3 Luoshu81Encrypt 非对称加密、解密功能图;

102 eS=d3encrypt(K,Y,X) 非对称加密

K,Y可以是基密钥(HashLuoshu值),亦可是原文密钥底层会统一检测转为基密钥进行运算;

对X先进行Base84编码(X含中英文混合);依次遍历取ASCII码

dx(ascii X)>32 ;30>Y>20;20>K>10 (上述规则保障X>Y>K,满足于洛数定理);对K,Y进行hashLuoshu运算并分组对dx进行eS=d3e(K,Y,dx)运算,后按Luoshu81进制编码表进行编码,-标识+81 -后如果是数字自动按溢出位计算否则按1位数字溢出,如-2,溢出81x2否则81x1 进行编码

1021 eS=d3e(K,Y,X)非对称逻辑单元X加密

eS=X+2Y-K-42;

103 X=d3decrypt(K,Y,eS) 非对称解密

1031 Y=d3hashY(K,X,eS) 非对称对Y基密钥解密(hash值)

S2服务器收到S1用Y,K加密发的eS;可以用自身的X,K对eS解密

得到Y的基密钥(Y的hash值)

10311 Y=d3y(K,X,eS) 非对称逻辑单元Y解密

Y=(eS-X+K+42)/2

1032 K=d3hashK=(X,Y,eS) 非对称K解密

10321 K=d3k(X,Y,eS)非对称逻辑单元K解密

K=X+2Y-eS-42

1033 X=d3decryptHash(K,Y,eS) 非对称X基密钥解密

K,Y是基密钥即原K,Y的hashLuoshu值

注:这里通过Y的基密钥是可以解密X;这里的解的X是明文不是基密钥,也只有X可解出明文;K,Y均只能解出基密钥

本算法本质为102的逆运算分组解码,分组调用K,Y基密钥,再调逻辑单元10331 dx=d3x(K,Y,eS) 然后逆调用(char) int81(dx) 解码为ascii缓存输出明文

10331 X=d3x(K,Y,eS) 非对称逻辑单元X解密

X=eS-2Y+K+42;

图4是节点S1与节点S2 基础wscomm.jar部署逻辑图;

1 S1节点 S1可信服务器(可为物理、虚拟机)

11 CT1{vm179,192.168.1.179} 为VM179节点终端实例

2 S2节点 S2可信服务器

21 CT2{vm180,192.168.1.180} 为VM180节点终端实例(实际可为外网ip)

104 SPK共享密钥对(keyX,keyK)内置于wscomm.jar 可配置

300 wscomm.jar 共享模组,内置104 SPK(KeyX,KeyK共享密钥对),并读写签名,加密编译防反编译,启动签名认证

图5是S1节点向S2发起非对称密钥安全通信Socket请求时序流程图;

调用流程:

如图5

ASPK TLS协议调用

STEP1:

1 S1->S2发起连接请求

S1 ->11 S1随机生成OTP16位数字密钥作为keyY ->12 调自配置共享密钥SPK(keyX,KeyK)->13 eS=d3encrypt(K,Y,X) 调X加密器进行加密得到eS->发送eS密文到S2

2 S2->21 读取eS->22 读取系统配置共享密钥K,X->23 Y=d3hashY(K,X,eS) 解密得到Y的基密钥hash值->24 X1=decryptHash(K,Y,eS) ->25 校验X1==X如果不为真exit ,不然S1可信->

STEP2:

26 S2知道S1可信,随机产生OTP16位数字密钥作为$X2->$skey 缓存->27 eS=d3encrypt(K,Y,$skey) 调用K,Y(hashkeyY),$skey 加密得到eS发送到S1 ->14 S1读取eS,step=2(第2阶段ASPK协议)->15 $X2=d3decrypt(K,Y,eS)=>$skey 以K,Y为密钥解密eS得到$X2->$skey 验证$skey为16位数字,即证明S2可信;$skey即为即将通讯的OTP密钥本次会话有效,缓存$skey

->到此ASPK2阶段通讯握手协议结束 S1,S2成功验证双方,并完成OTP密钥交换

->16 S1以$skey为密钥采用Luoshu81Encrypt 对称超算加密算法对数据加密发送密文$e->S2 28采用对称算法解密得到明文$d ->17,29 S1,S2 通讯结束清理$skey,$keyY

小结：

STEP1 : S2利用K,X,eS算出S1 Y的hash值(不能算得原始密钥,因hashLuoshu算法不可逆)

利用 K,Y,eS解密得到X1比较X验证了S1可信；

利用共享密钥K,X变化Y

STEP2:S2 产生OTP $X2=$skey 重新调eS=d3encrypt(K,Y,$skey) S1解密eS得到skey

利用了共享密钥K,Y 变化X

2阶段都是用一组公式变化俩个自变量算得需要的自变量

图6是对图2,图3,图4标号注明图；

图7 是对图5调用逻辑文字图；

图8是S1节点发起调用控制台日志图；

图9是S2节点接收调用控制台日志图；

发明要义：

1. 一种基于基于洛数81九宫方阵推衍的高安全高效高压缩数据的超算加密算法(参专利公布申请号:CN202210772306.3)可跨节点预置的jar库装备,其特征如下:

1.1其内部主要特征方法如下封装包Luoshu81Encrypt

eS=d3encrypt(K,Y,X),

X=d3decrypt(K,Y,eS)

Y=d3ydecrypt(K,X,eS)

K=d3kdecrypt(X,Y,eS)

S=hashLuoshu(S0) 自定hashLuoshu算法

X=d3decryptHash(keyhashK,keyhashY,eS) 解密基础函数

keyhashY=d3hashY(keyK,KeyX,eS) 逆向通过K,X,eS反推求的keyhashY即Y的hashLuoshu密钥

keyhashK=d3hashY(keyX,KeyY,eS) 逆向通过X,Y,eS反推求的keyhashK即K的hashLuoshu密钥

int s=d3e(int x,int y,int k) 逻辑单元对x,y,k加密算法封装为s

int x=d3x(int y,int s,int k) 逻辑单元解密x

int y=d3y(int x,int s,int k) 逻辑单元解密y

int k=d3k(int x,int y,int s) 逻辑单元解密 k

注:这里的d3代表direct 3 三个方向之意。

[0008] 洛数三角定理及运用特征如下(参专利公布申请号:CN202210772306.3)：

任意三个九宫洛数用线连接起来组成三角形,即X,Y,Z (满足于X>Y>Z)

若a=X-Y,b=Y-Z,c=X-Z;则有c=a+b.

证明:依命题,a+b=>(X-Y)+(Y-Z)=X-Z,而c=X-Z故a+b=c成立,由交换律c=a+b亦成立,证毕;本发明依据洛数三角定理,已知X,Y;设Z=Y-K;令eS=X+Y+Z,则有

eS=X+Y+Z=X+Y+(Y-K)=X+2Y-K

eS=X+2Y-K 实际运算中采用修正值去掉ascii码低32码(控制符),并在实际编码中为保持Y>Z,Y采用+20,Z采用+10补位修正运算,故修正差为-42

这里的K标识Y与Z的偏移

实际公式为eS=X+2Y-K-42

d3e={eS=X+2Y-K-42;return eS;}

d3x={X=eS-2Y+K+42;return X;}

d3y={Y=(eS-X+K+42)/2;return Y;}

依据洛书三角定理,X,Y,K共同组合成eS;知道任意3者可确定另一者;对任意数字eS可拆分为3个数X,Y,K,满足于eS=X+2Y-K-42;

为防止暴力破解实际运算引入了加盐K修正算法,并采用了流密钥hashLuoshu分组算法,即单元流eS=d3e(K,Y,X) 每个单元的dk值偏移不同,导致破解困难;dk单元值变化依据密钥K=hashLuoshu(K),不知道K是无法破解方程组;且对任意四元一次方

程,知其2者,则有无数方程解,知其3者,则有唯一方程解,这是本发明算法安全基础;

非对称共享密钥基础:

对任意K,X可事先预内置在通信服务器节点服务器配置文件内,并加以签名保护,ASPK安全通讯分俩阶段协议进行:

sessionOnASPK 函数被代理封装实现了aspk1,aspk2 二阶段握手协议封装于ioserver-server.jar,ioserver-core.jar,ioserver-client.jar 客户端、服务端均集成sessionOnASPK函数在通讯连接建立时智能代理并建立非对称密钥安全握手连接;在通讯消息发送时自动检测并采用对称加密、解密,ioserver-core底层封装了消息过滤器智能过滤aspk协议指令;

协议过滤器被内置于VMESever 中

public static final String aspkFilter="\\&(act=aspk|msgtype=echo|ver=)";

标识该匹配的字段不加密采用私有处理规则;

2.1第一阶段STEP 1 :

协议头字段 &act=aspk&step=1

C客户机,向S服务器发起连接请求,并随机生成OTP一次会话密钥Y;以配置X,K

调本算法eS=d3encrypt(K,Y,X), 发送密文eS给服务器S;

S:收到ASPK协议请求并收到密文eS;调用keyhashY=d3hashY(K,X,eS)进行解密得到Y的hashLuoshu 基密钥(注:不可能完全得到Y密钥,因hashLuoshu不可逆);

拿到基密钥即可调基算法实现解密, X0=d3decryptHash(keyhashK,keyhashY,eS) 然后S端比较X==X0如果为真则证得C客户端与S端拥有相同的K,X,证明客户端可信;

注1:以上X,K不变求得Y hash密钥;

ASPK协议阶段1:客户端C生成新的Y,用K,Y加密X得eS, 只是泄露了eS发送到服务端S;用K,X,解密Y基密钥再解密X以校验客户端安全性;

2.2第二阶段STEP2:

协议头字段 &act=aspk&step=2;

S服务端随机生成OTP临时密钥X=$skey, 调本算法eS=d3encrypt(K,Y,$skey),发送密文eS给服务器S;注这里的Y即可以是Y密钥亦可传递Y的hashLuoshu基密钥在底层统一转为Y基密钥进行处理;C端客户机收到密文eS 且step=2,用Y,K,eS 调X=d3decrypt(K,Y,eS)

解出密钥X并为16位数字,可验证服务器可信并用此X作为通信双方的安全临时OTP密钥,仅限于本次通信会话;

注2:以上Y,K不变,传递变化的X ;

2.3基于2.1,2.2 以skey作为X,在K,Y不变的前提下加密X得eS并发送(泄露eS)eS给客户端C,C端解密完成服务端校验,并成功传递了OTP密钥。

[0009] 内置Luoshu81超算加密算法wscomm.jar特征如下:

服务器S,与客户机C 必须配置相同的密钥对X,K用以验证双方通信第一次口令Y,如下:X,K

aspk.keyX=2022112213412197

aspk.keyK=2022091203059093

(实例可随机定义,实际发布不公开)

aspk代表非对称共享密钥协议

keyX其本质相当于2 中的X;

keyK 其本质相当于2中的K;

只不过这里对X,K进行了流密钥分组由整数转为流密钥字串序列

但整数易破解,长流密钥序列难以破解,K,X相当于集成算法不公开,基层API wscomm.jar 可通过自实现(c++)(ByteCoderEncryptor.so/dll)防编译库,并对产生jar采用virbox Protector对虚拟机java/jar进行2次保护,对jar加密并签名运行保护以实现核心api其安全性加载。

[0010] 一种超高速非对称密钥安全传输算法,其特征如下:

4.1 本发明利用了九宫洛数三角定理规律约定了通信双方预置共享密钥K,X对以共享密钥对为基础进行校验发送者密钥,并校验发送者与接收者是否具有共享密钥对;一方以密钥Y,K对X加密eS;另一方以K,X对eS进行解密产生Y基密钥(hashLuoshu),而同时不泄露密钥Y,此为一种典型的非对称密钥加密解密方案;本算法中X,Y,K共同组合成为eS实际传输(仅暴露)eS

四元一次方程知3个未知量,有方程唯一解;知小于3 未知量有无数方程解,导致求解困难,间接证明算法的安全性;

4.2 基于4.1 本发明仅暴露1个未知量eS;解方程需要3个未知量;故导致求解困难;反证算法安全性;

4.3 基于4.1 为防止暴力破解,任何人知道算法,但不知道K基密钥序列导致破解困难;破解困难的原因是随着破解增加,解密的明文字元与密文字元混杂一起,而攻击者已

经不能正确区分哪些是明文,哪些是密文字元;而只有正确的密文K序列才能破解出正确的明文序列;K为共享密钥对不传递;无从获取;

4.4 基于2.1 在ASPK协议第一阶段 发送者C利用的内置SPK(keyX,keyK) 自己随机生成keyY ,用eS=d3e(K,Y,X) 对X加密得到密文eS;K,X不变;变化了Y目的传递Y的密钥给接受者S;

但S通过共享密钥对SPK(keyK,keyX),Y=d3hashY(K,X,eS)可解密得到Y的基密钥;而2.1加密的本质是对Y,K基密钥(hashLuoshu)进行,拿到基密钥变相于拿到Y;然后X1=d3x(K,Y,eS)比较X1,X;如果发送者C,与接受者S内置相同的K,则X一定是相同即反证C,S是持有相同密钥的可信服务器 ;

4.5 基于2.2 接收者S在验证发送者C;利用4.4得到的Y基密钥(hashLuoshu),用eS=d3e(K,Y,X) 这里X=$skey=$X2即新产生的OTP密钥;此时公式算法相同,但保持K,Y不变,变化了自变量X,发给发送者C,此时C得到eS,step=2阶段协议,因为自身具有K,Y可利用函数X2=d3decrypt(K,Y,eS),解密后X2即本次通讯的对称加密OTP口令skey;

4.6 基于4.4,4.5 在双方通讯中关键口令Y,与X2->$skey均不涉及明码传递,期间以密文被嵌入eS传递,eS有无数方程解也即等于无解;攻击者必须同时具备eS,K,(X/Y)掌握3个未知量方能破解;而这是不可能;从而反证算法安全;

4.7 基于4.4,4.5 可知只需要经过两步通讯,实际发送->接收(校验)->发送 3步完成服务器相互验证,密钥传递;简化复杂流程;同时基于2 洛数三角定理不涉及复杂乘法仅用到加、减法运算,从算法上提高几个数量级,是轻量级运算,提高加密、解密计算速度;

4.8 基于 1.1 本发明实现基于洛数超算加密、解密,自定hashLuoshu杂凑算法,并支撑数据高倍压缩安全传输;

实际运算中,如果原文X小于32则会自动补位运算,补位目的是尽可能遍历hashkeyY,hashkeyK,在逆运算中能够完整的算出对应的hashkey基密钥;故密文比原文长;实际测算正常密文压缩率约为123%,洛数对称加密压缩率109%;洛数非对称加密效率较洛数对称加密快1.5倍;

4.9 基于3,4本发明可用于并行、分布式系统下多节点间,核心密钥超算加密安全传输,以及可成为军用、民用领域超高、超快,超安全通讯核心集成方案。

[0011] 现实意义:

smg-wscomm-Lushu81Encrypt-ASPKI基于洛数非对称密钥超算加密安全传输算法;本发明可实现对任意对任意节点通过部署内置相同的wscomm.jar模组配置SPK(keyX,keyX)密钥对,并实施防反编译、启动前进行读写保护可建立可信服务,通过节点间试探性发送签名;可实现对节点完全可信认证,发起方发送试探密钥,接受方解密完成校验,并响应一个预期密钥,发起方解密预期密钥成功,则完成双方可信认证后续以此密钥进行加密会话,可实现超高安全性,高可靠性信息保密压缩传输;而在通讯加密、解密分别采用不同的密钥进行操作,本发明采用非对称算法基于洛数三角定理可实现超高速密钥的安全交换,缩短通讯握手流程,加快信息高保密信息互联互通,可广泛运用于军民俩用,高效率加密解密技术为超算力输出,超分布式存储输出必放一异彩。

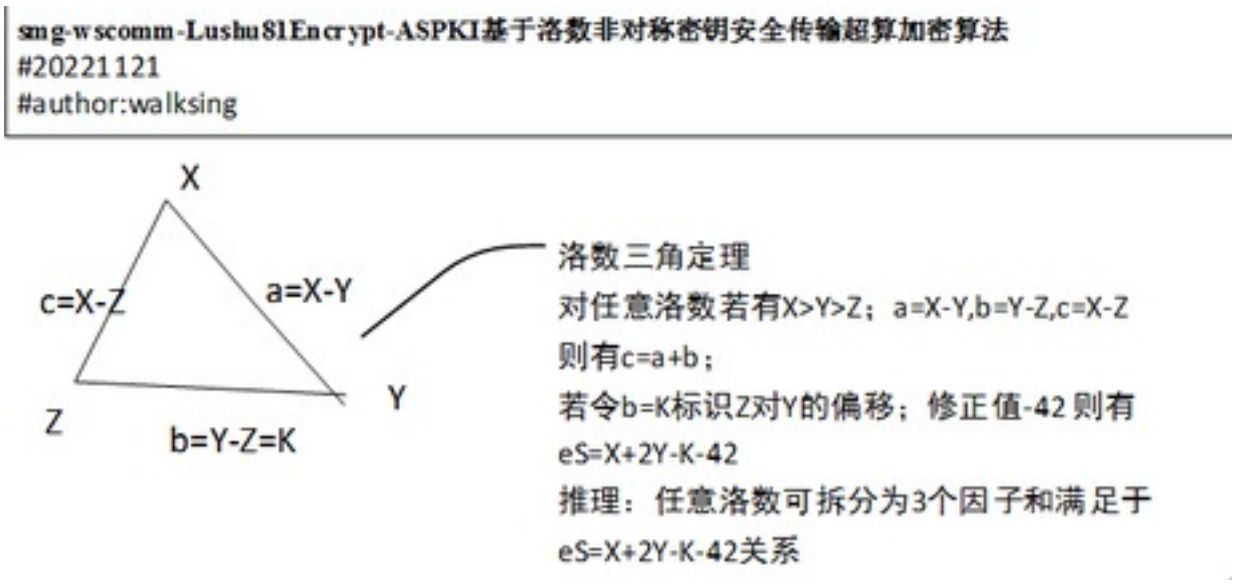

smg-wscomm-Lushu81Encrypt-ASPKI基于洛数非对称密钥安全传输超算加密算法
#20221121
#author:walksing
X
c=X-Z
a=X-Y
Y
Z
b=Y-Z=K
洛数三角定理
对任意洛数若有X>Y>Z；a=X-Y,b=Y-Z,c=X-Z
则有c=a+b；
若令b=K标识Z对Y的偏移；修正值-42 则有
eS=X+2Y-K-42
推理：任意洛数可拆分为3个因子和满足于
eS=X+2Y-K-42关系

图1

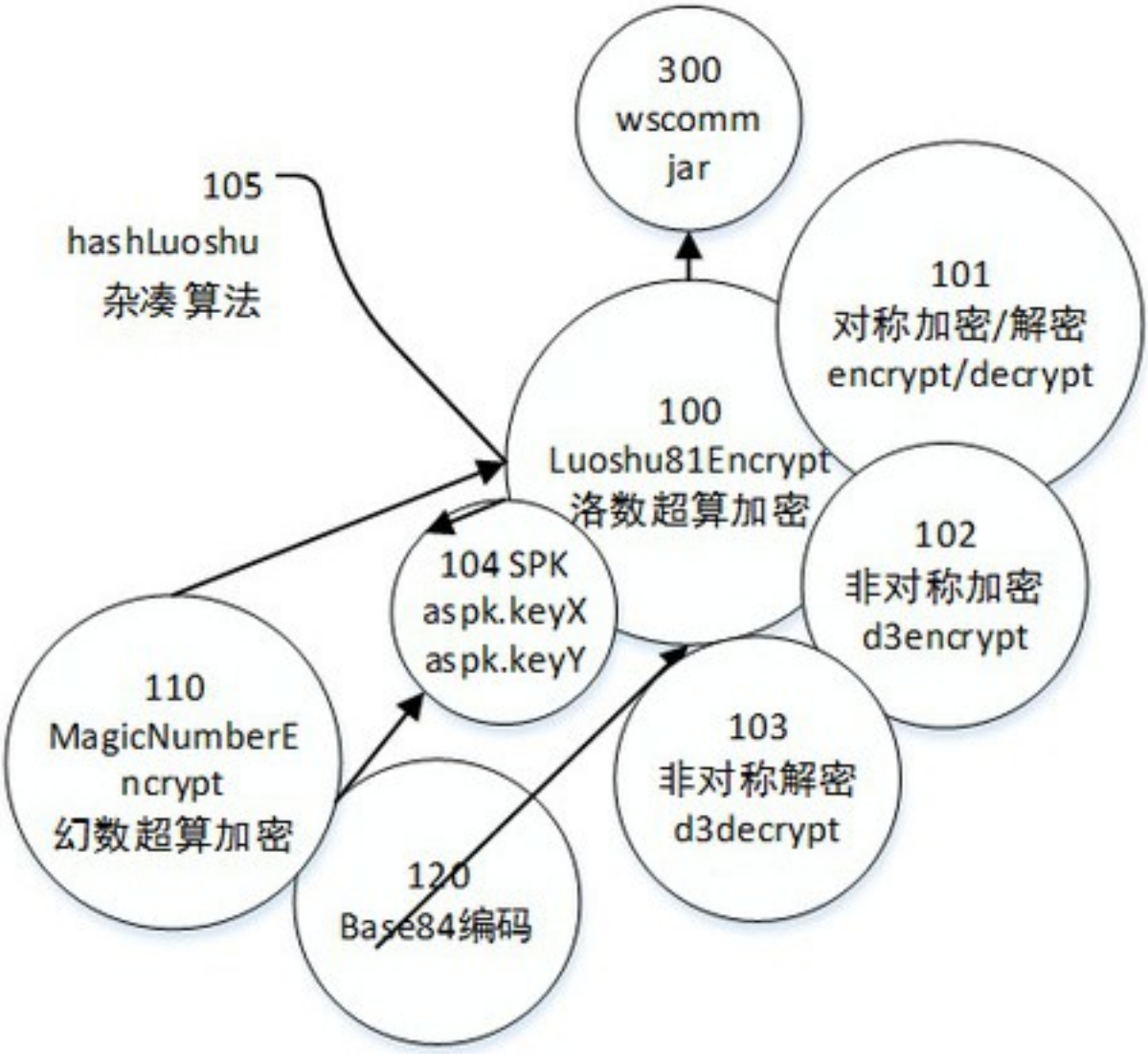

300
wscomm
jar
105
hashLuoshu
杂凑算法
101
对称加密/解密
encrypt/decrypt
100
Luoshu81Encrypt
洛数超算加密
102
非对称加密
d3encrypt
104 SPK
aspk.keyX
aspk.keyY
110
MagicNumberE
ncrypt
幻数超算加密
103
非对称解密
d3decrypt
120
Base84编码

图2

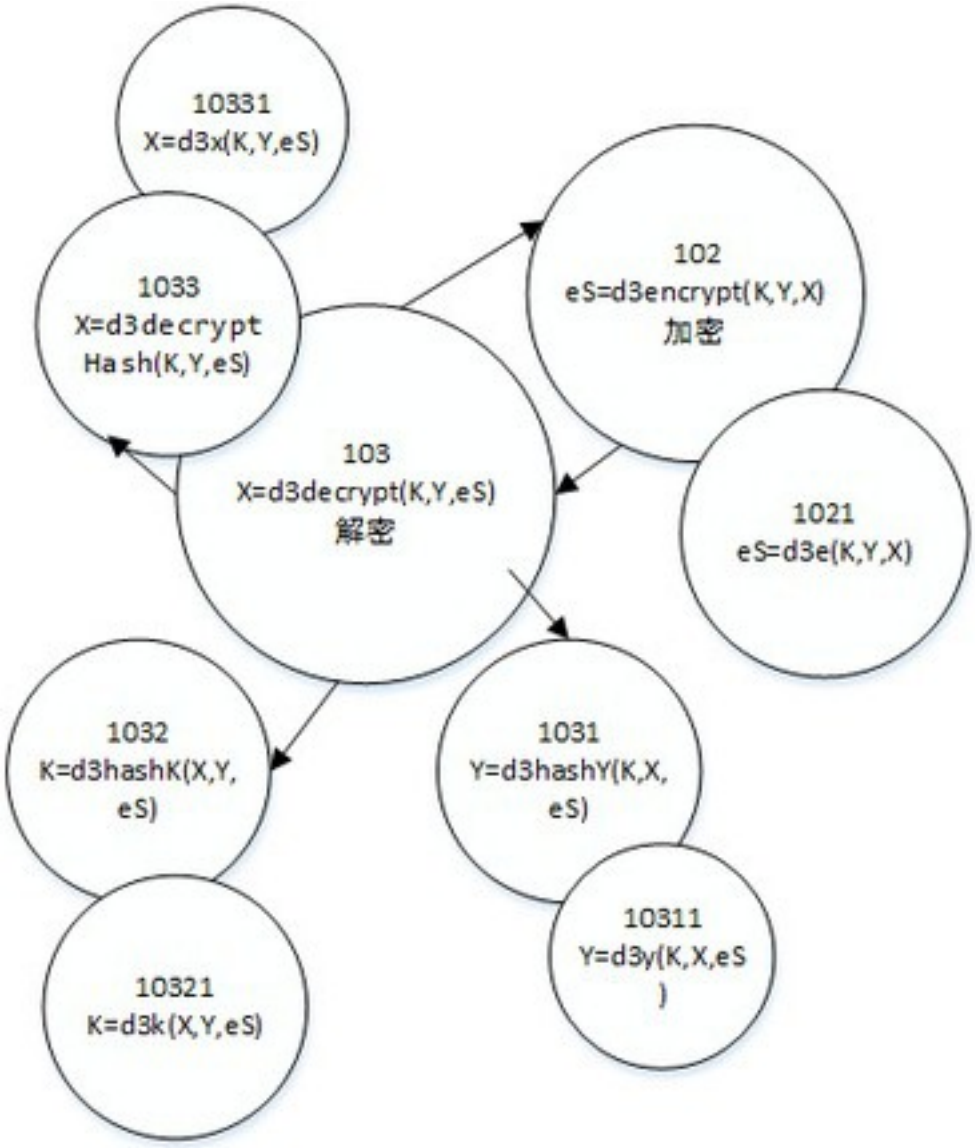

10331
X=d3x(K,Y,eS)
1033
X=d3decrypt
Hash(K,Y,eS)
102
eS=d3encrypt(K,Y,X)
加密
103
X=d3decrypt(K,Y,eS)
解密
1021
eS=d3e(K,Y,X)
1032
K=d3hashK(X,Y,
eS)
1031
Y=d3hashY(K,X,
eS)
10311
Y=d3y(K,X,eS
)
10321
K=d3k(X,Y,eS)

图3

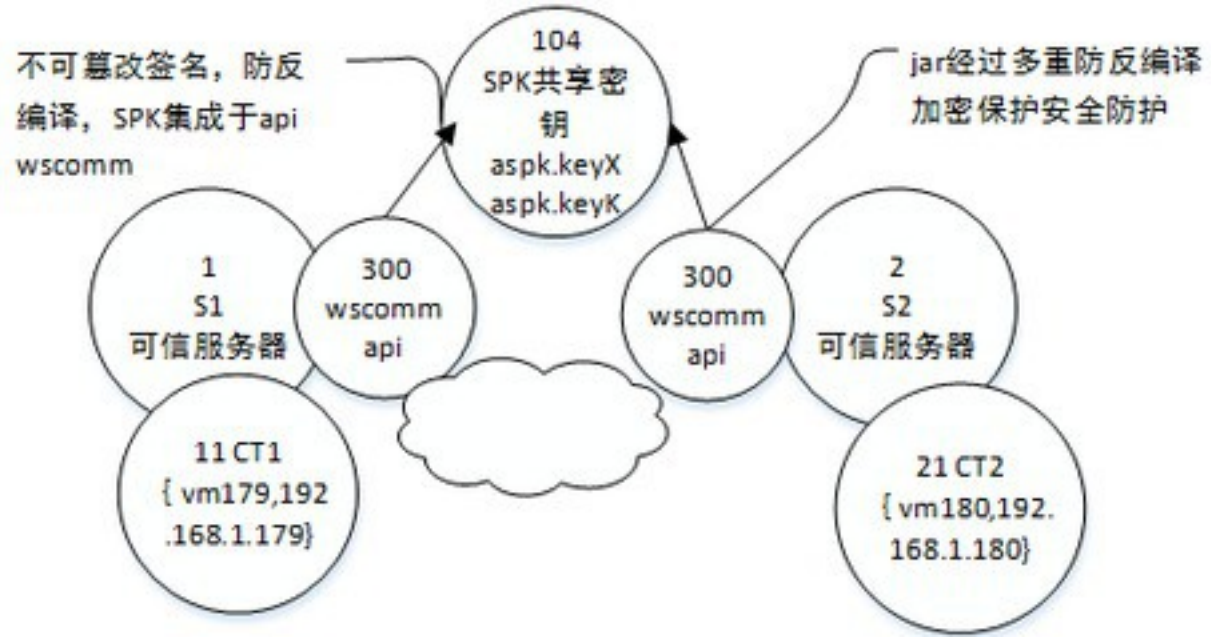

不可篡改签名，防反编译，SPK集成于api wscomm
104
SPK共享密钥
aspk.keyX
aspk.keyK
jar经过多重防反编译加密保护安全防护
1
S1
可信服务器
300
wscomm
api
300
wscomm
api
2
S2
可信服务器
11 CT1
{ vm179,192.168.1.179}
21 CT2
{ vm180,192.168.1.180}

图4

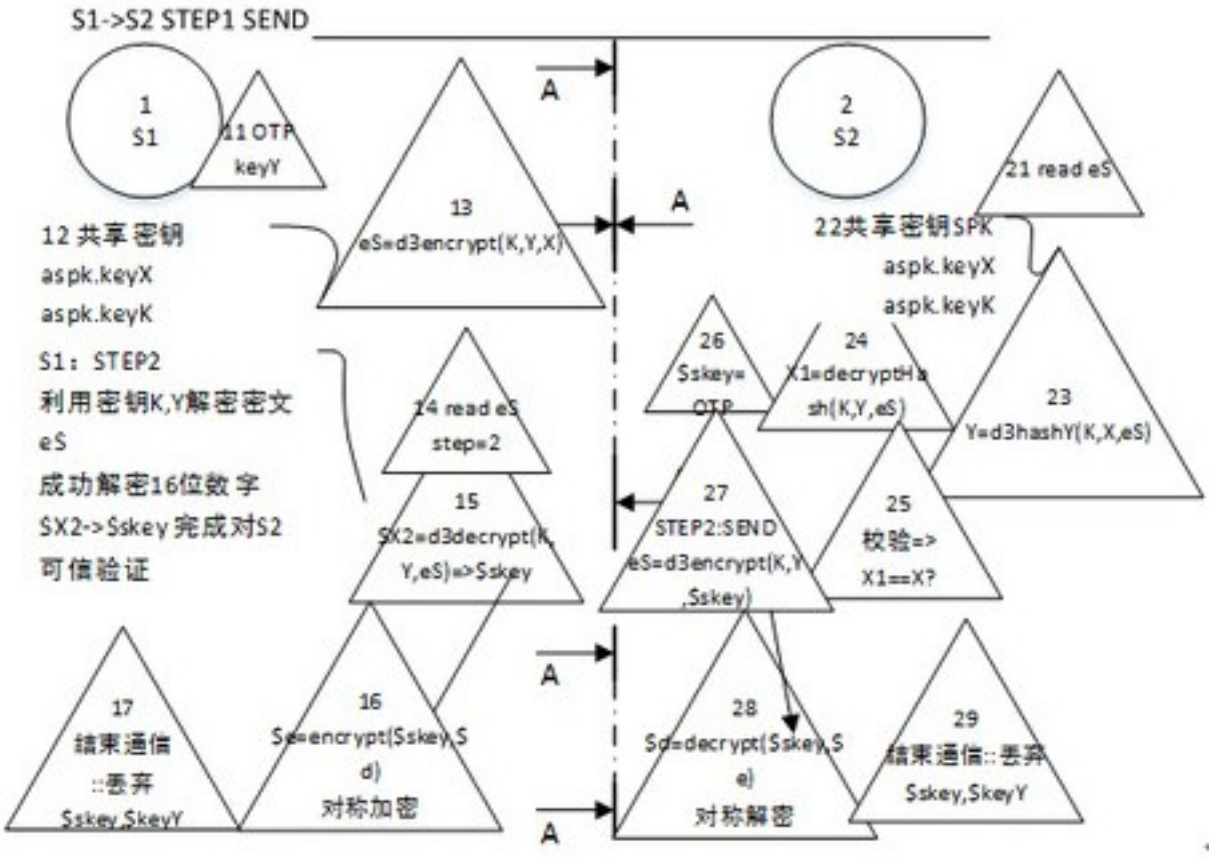

S1->S2 STEP1 SEND
1
S1
11 OTP
keyY
A
2
S2
21 read eS
13
eS=d3encrypt(K,Y,X)
A
12 共享密钥
aspk.keyX
aspk.keyK
22共享密钥SPK
aspk.keyX
aspk.keyK
S1：STEP2
利用密钥K,Y解密密文eS
成功解密16位数字
SX2->Sskey完成对S2可信验证
26
Sskey=OTP
24
X1=decryptHash(K,Y,eS)
23
Y=d3hashY(K,X,eS)
14 read eS
step=2
15
SX2=d3decrypt(K,Y,eS)=>Sskey
27
STEP2:SEND
eS=d3encrypt(K,Y,Sskey)
25
校验=>
X1==X?
A
17
结束通信
::丢弃
Sskey,SkeyY
16
Se=encrypt(Sskey,Sd)
对称加密
28
Sd=decrypt(Skey,Se)
对称解密
29
结束通信::丢弃
Sskey,SkeyY
A

图5


300 wscomm.jar 加密算法api核心包
100 Luoshu81Encrypt 洛数超算加密类（含对称/非对称）
101 encrypt/decrypt 对称加密、解密
102 eS=d3encrypt(K,Y,X) 非对称加密
1021 eS=d3e(K,Y,X)非对称逻辑单元X加密
103 X=d3decrypt(K,Y,eS) 非对称解密
1031 Y=d3hashY(K,X,eS) 非对称对Y基密钥解密（hash值）
10311 Y=d3y(K,X,eS) 非对称逻辑单元Y解密
1032 K=d3hashK=(X,Y,eS) 非 对称K解密
10321 K=d3k(X,Y,eS)非对称逻辑单元K解密
1033 X=d3decryptHash(K,Y,eS) 非对称X基密钥解密
10331 X=d3x(K,Y,eS) 非对称逻辑单元X解密
104 SPK aspk.keyX,aspk.keyK 共享密钥对
105 hashLuoshu杂凑算法
110 MagicNumberEncrypt 幻数超算加密
120 Base84编码
1 S1可信服务器
2 S2可信服务器
11 CT1 vm179:192.168.1.179
12 CT2 vm180:192.168.1.180

图6

调用流程：
如图
ASPK TLS协议调用
STEP1:
1 S1->S2发起连接请求
S1 ->11 S1随机生成OTP16位数字密钥 作为keyY ->12 调自配置共享密钥SPK(keyX,KeyK)->13
eS=d3encrypt(K,Y,X) 调X加密器进行加密得到eS->发送eS密文到S2
2 S2->21 读取eS->22 读取系统配置共享密钥K,X->23 Y=d3hashY(K,X,eS) 解密得到Y的基密钥hash
值->24 X1=decryptHash(K,Y,eS) ->25 校验X1==X?如果不为真exit，不然S1可信->
STEP2:
26 S2知道S1可信，随机产生OTP16位数字密钥作为$X2->$skey缓存->27 eS=d3encrypt(K,Y,$skey)
调用K,Y(hashkeyY),$skey 加密得到eS 发送到S1 ->14 S1读取eS ,step=2（第2阶段ASPK协议）->15
$X2=d3decrypt(K,Y,eS)=>$skey 以K,Y为密钥解密eS得到$X2->$skey验证$skey为16位数字，即证明
S2可信;$skey即为即将通讯的OTP密钥本次会话有效,缓存$skey
->到此ASPK2阶段通讯握手协议结束 S1,S2成功验证双方，并完成OTP密钥交换

->16 S1以$skey为密钥采用Luoshu81Encrypt 对称超算加密算法对数据加密发送密文$e->S2 28采用对称算法解密得到明文$d ->17,29 S1,S2 通讯结束清理$skey,$keyY

小结：

STEP1 : S2利用K,X,eS算出S1 Y的hash值（不能算得原始密钥，因hashLuoshu算法不可逆）

利用 K,Y,eS解密得到X1比较X验证了S1可信；

利用共享密钥K,X变化Y

STEP2:S2 产生OTP $X2=$skey 重新调eS=d3encrypt(K,Y,$skey) S1解密eS得到skey

利用了共享密钥K,Y 变化X

2阶段都是用一组公式变化俩个自变量算得需要的自变量

图7

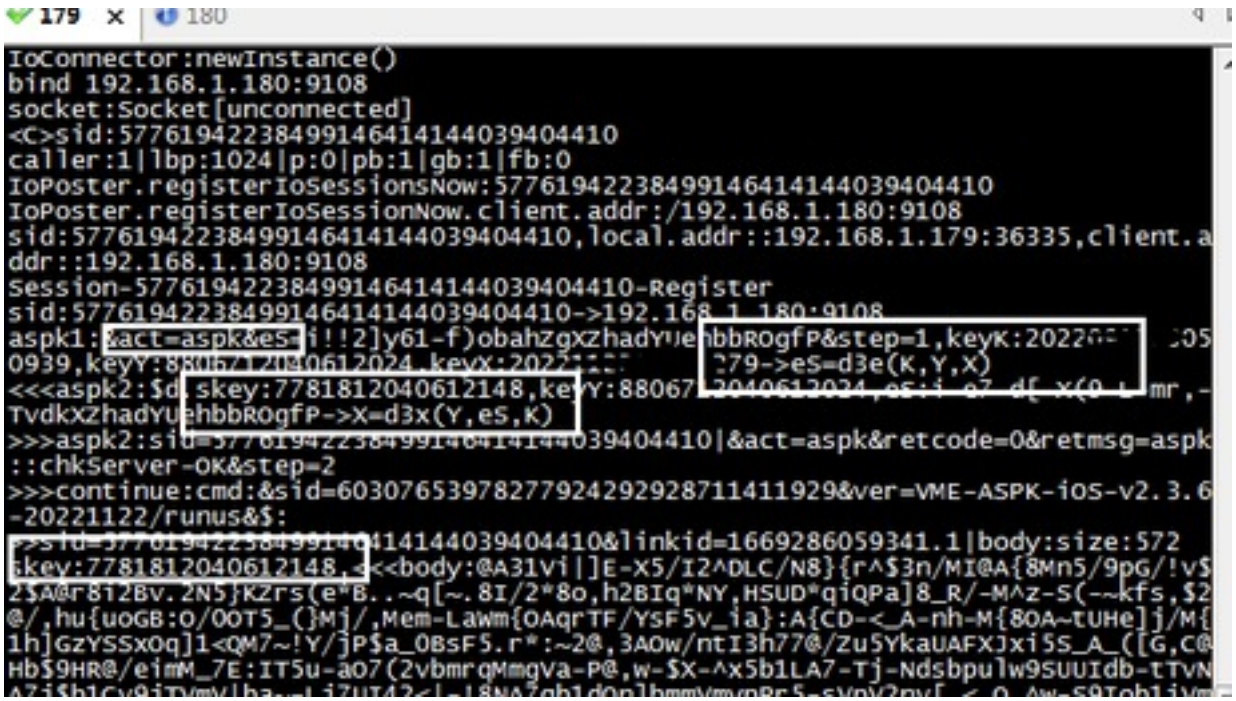

179
180
IoConnector:newInstance()
bind 192.168.1.180:9108
socket:Socket[unconnected]
<C>sid:577619422384991464141440394044l0
caller:1|lbp:1024|p:0|pb:1|gb:1|fb:0
IoPoster.registerIoSessionsNow:577619422384991464141440394044l0
IoPoster.registerIoSessionNow.client.addr:/192.168.1.180:9108
sid:577619422384991464141440394044l0,local.addr::192.168.1.179:36335,client.addr::192.168.1.180:9108
Session-577619422384991464141440394044l0-Register
sid:577619422384991464141440394044l0->192.168.1.180:9108
aspk1:&act=aspk&eS=i!!2]y61-f)obahZgXZhadYUehbbROgfP&step=1,keyK:2022
0939,keyY:88067
<<<aspk2:$d skey:7781812040612148,keyY:88067
TvdkXZhadYUehbbROgfP->X=d3x(Y,eS,K)
>>>aspk2:sid=577619422384991464141440394044l0|&act=aspk&retcode=0&retmsg=aspk::chkServer-OK&step=2
>>>continue:cmd:&sid=6030765397827792429292871141l929&ver=VME-ASPK-iOS-v2.3.6-20221122/runus&$:
skey:7781812040612148,<<<body:@A31Vi||]E-X5/I2^DLC/N8}{r^$3n/MI@A{8Mn5/9pG/!v$
2$A@r8i2Bv.2N5}KZrs(e*B..~q[~.8I/2*8o,h2BIq*NY,HSUD*qiQPa]8_R/-M^z-S(-~kfs,$2
@/,hu{uoGB:O/OOT5_(}Mj/,Mem-Lawm{OAqrTF/YsF5v_ia}:A{CD-<_A-nh-M{8OA~tUHe]j/M{
1h]GzYSSxOq]1<QM7~!Y/jP$a_OBsF5.r*:~2@,3AOw/ntI3h77@/Zu5YkaUAFXJxi5S_A_([G,C@
Hb$9HR@/eimM_7E:IT5u-aO7(2vbmrqMmgVa-P@,w-$X-^x5b1LA7-Tj-NdsbpuIw9SUUIdb-tTvN
aspk1:keyK:2022
,keyY:52
2572334573384837858437875308 71,keyX:2022
,tkeyX:2022
Y=d3y(X,eS,K),X=d3x(Y,eS,K))
8711411929|&act=aspk&retcode=0&retmsg=aspk::chkClient-OK&step=2
INFO | jvm 1 | 2022/11/24 18:34:19 | skey:7781812040612148->i-e7-d[-X(9-L-mr,-TvdkXZhadYUehbbROgfP&step=2->eS=d3e(K,Y,X)
INFO | jvm 1 | 2022/11/24 18:34:19 | vmeserver:|chkToken: OK
INFO | jvm 1 | 2022/11/24 18:34:19 | PPE::addr:
INFO | jvm 1 | 2022/11/24 18:34:19 | msgtype:|method:|&act=aspk
INFO | jvm 1 | 2022/11/24 18:34:20 | skey:7781812040612148,<<<body:@Ac^LLFSx7h5xR3A<j8quI$q6vn/MPUBni[B@O/$Y}}p<zpeT}}xAY(7I2Oaa-4{<_FLvGOqYGGMz2uI<q!jAqv3S9AQUjwOg{R(_v,,z8H,CSg-S7HNr^}(y~BTb~Vt[E@</t[_9tu33B~9.q2Z36~3eJGP5dTZ-kEyN6[(J|*@6E_gOq|SR_:AYXUa@A31iiD]Pe]O/_$<A{X@}t<i^@HAnHiy@AG^YhGNX_BP5R*($SN4I,AM~,RIbTQq(Q$eXA-SCe-SL-NkE)Nt/.2SH^D:25SoyU~/GTNXOqKGNRD:A{vUVt<i^@HAnHiy@AH_FKESN^_/.3$@9B<q/q$E7.2Mm]SDg*Ol6A1zOq.!9c928_@pyp6E6a63K_(N@A6@i{

图8

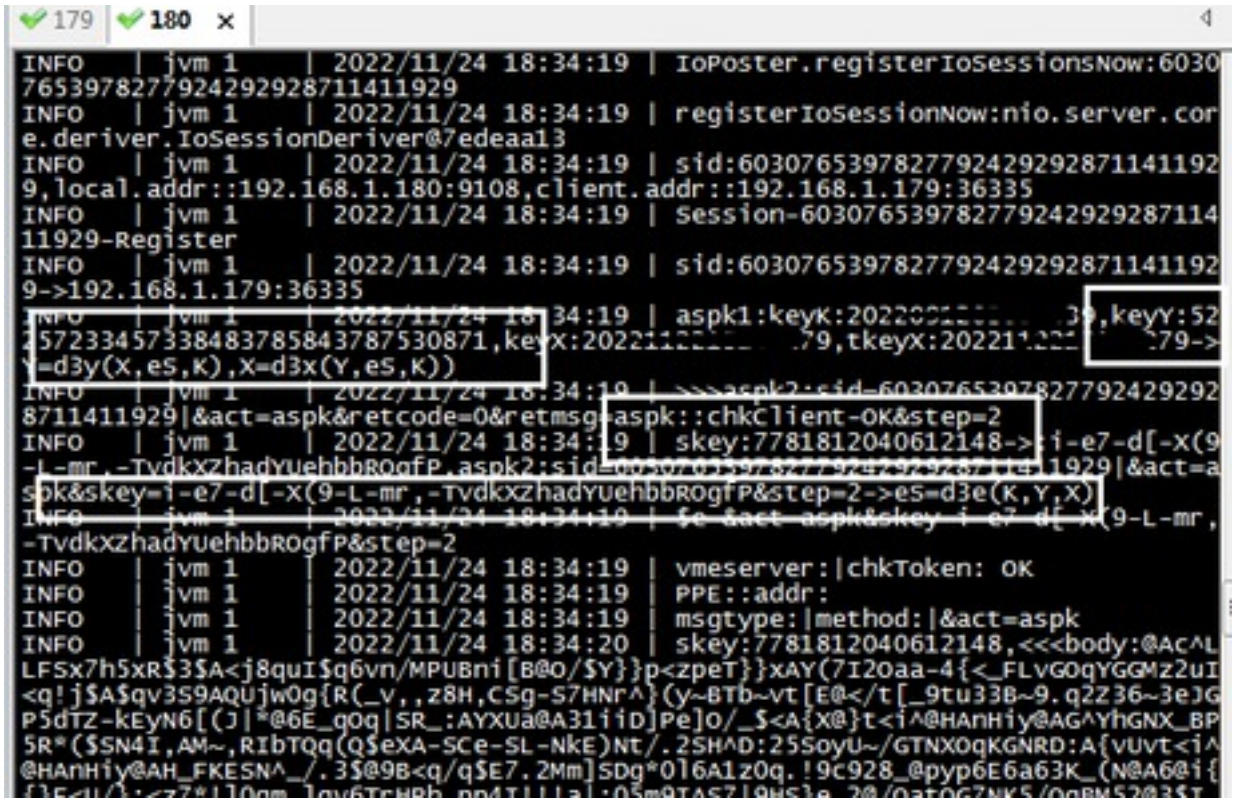

179
180
INFO | jvm 1 | 2022/11/24 18:34:19 | IoPoster.registerIoSessionsNow:6030765397827792429292871141l929
INFO | jvm 1 | 2022/11/24 18:34:19 | registerIoSessionNow:nio.server.core.deriver.IoSessionDeriver@7edeaa13
INFO | jvm 1 | 2022/11/24 18:34:19 | sid:6030765397827792429292871141l929,local.addr::192.168.1.180:9108,client.addr::192.168.1.179:36335
INFO | jvm 1 | 2022/11/24 18:34:19 | Session-6030765397827792429292871141l929-Register
INFO | jvm 1 | 2022/11/24 18:34:19 | sid:6030765397827792429292871141l929->192.168.1.179:36335
INFO | jvm 1 | 2022/11/24 18:34:19 | aspk1:keyK:2022
,keyY:52
2572334573384837858437875308 71,keyX:2022
,tkeyX:2022
Y=d3y(X,eS,K),X=d3x(Y,eS,K))
INFO | jvm 1 | 2022/11/24 18:34:19 | >>>aspk2:sid=6030765397827792429292871141l929|&act=aspk&retcode=0&retmsg=aspk::chkClient-OK&step=2
INFO | jvm 1 | 2022/11/24 18:34:19 | skey:7781812040612148->i-e7-d[-X(9-L-mr,-TvdkXZhadYUehbbROgfP,aspk2:sid=6030765397827792429292871141l929|&act=aspk&skey=i-e7-d[-X(9-L-mr,-TvdkXZhadYUehbbROgfP&step=2->eS=d3e(K,Y,X)
INFO | jvm 1 | 2022/11/24 18:34:19 | $e &act=aspk&skey=i-e7-d[-X(9-L-mr,-TvdkXZhadYUehbbROgfP&step=2
INFO | jvm 1 | 2022/11/24 18:34:19 | vmeserver:|chkToken: OK
INFO | jvm 1 | 2022/11/24 18:34:19 | PPE::addr:
INFO | jvm 1 | 2022/11/24 18:34:19 | msgtype:|method:|&act=aspk
INFO | jvm 1 | 2022/11/24 18:34:20 | skey:7781812040612148,<<<body:@Ac^LLFSx7h5xR3A<j8quI$q6vn/MPUBni[B@O/$Y}}p<zpeT}}xAY(7I2Oaa-4{<_FLvGOqYGGMz2uI<q!jAqv3S9AQUjwOg{R(_v,,z8H,CSg-S7HNr^}(y~BTb~Vt[E@</t[_9tu33B~9.q2Z36~3eJGP5dTZ-kEyN6[(J|*@6E_gOq|SR_:AYXUa@A31iiD]Pe]O/_$<A{X@}t<i^@HAnHiy@AG^YhGNX_BP5R*($SN4I,AM~,RIbTQq(Q$eXA-SCe-SL-NkE)Nt/.2SH^D:25SoyU~/GTNXOqKGNRD:A{vUVt<i^@HAnHiy@AH_FKESN^_/.3$@9B<q/q$E7.2Mm]SDg*Ol6A1zOq.!9c928_@pyp6E6a63K_(N@A6@i{

图9