(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 109150789 A
(43)申请公布日 2019.01.04

(21)申请号 201710450451.9

(22)申请日 2017.06.15

(71)申请人 沈阳高精数控智能技术股份有限公司
地址 110168 辽宁省沈阳市东陵区南屏东路16-2号

(72)发明人 胡毅 李力 孙砚辉 毕筱雪 刘劲松 吴迪

(74)专利代理机构 沈阳科苑专利商标代理有限公司 21002
代理人 许宗富

(51)Int.Cl.
H04L 29/06(2006.01)

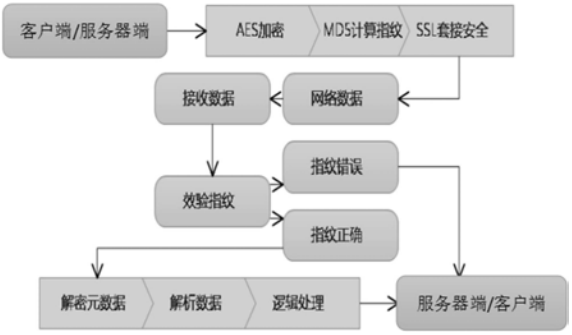
权利要求书1页 说明书3页 附图2页

(54)发明名称

一种用于数字化车间信息安全的混合加密通信方法

(57)摘要

本发明涉及一种用于数字化车间信息安全的混合加密通信方法,包括:在通讯过程中使用SSL对传输通道进行连接加密,使用自定义的AES算法对网络传输中的数据进行加密,并且在发送端加密数据后使用自定义的MD5对加密的数据生成发送端数字指纹,在接收端同时对发过来的加密数据使用MD5生成接收端数字指纹,通过指纹效验判断数据在传输过程中是否被篡改或丢失。本发明有效的消除了数字化车间管理系统实时采集的数据在端到端的通讯传输中被篡改的可能,极大可能的提高了数字化车间管理系统通讯和数据的安全性和可靠性。



CN 109150789 A

1.一种用于数字化车间信息安全的混合加密通信方法,其特征在于,包括以下步骤:

步骤1:在通讯过程中,发送端使用SSL技术对传输通道进行连接加密;

步骤2:发送端对网络传输中的数据进行加密生成加密数据并发送给接收端;

步骤3:发送端对加密数据进行处理生成发送端数字指纹;同时接收端对接收的加密数据进行处理生成接收端数字指纹;

步骤4:通过对发送端数字指纹和接收端数字指纹进行指纹效验,判断出数据在传输过程中是否被篡改或丢失。

2.按照权利要求1所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述步骤2中发送端是采用自定义的AES算法对网络传输中的数据进行加密生成加密数据,包括:通过修改AES算法的安全哈希算法字典长度、置换表、以及置换选择和位移规则对要传输的数据片段或文件进行加密。

3.按照权利要求2所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述修改安全哈希算法字典长度为将字典长度随机修改成一个属于2^{64}范围内的数值。

4.按照权利要求2所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述修改置换表包括修改初始置换表、逆向初始置换表、扩展置换表。

5.按照权利要求2所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述修改置换选择和位移规则为将出现置换的位置按照数组范围进行混淆。

6.按照权利要求1所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述步骤3中发送端是采用自定义的MD5算法对加密数据进行处理生成发送端数字指纹,包括:通过修改MD5算法的置换数组组合和输出字节序列对发送端生成的加密数据进行处理生成发送端数字指纹。

7.按照权利要求1所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述步骤3中接收端是采用自定义的MD5算法对接收的加密数据进行处理生成接收端数字指纹,包括:通过修改MD5算法的置换数组组合和输出字节序列对接收端接收的加密数据进行处理生成接收端数字指纹。

8.按照权利要求6或7所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述修改MD5算法的置换数组组合采用修改轮转和算法分离计算赋值的方法,所述修改输出字节序列采用修改区块转换过程变量的方法。

9.按照权利要求1所述一种用于数字化车间信息安全的混合加密通信方法,其特征在于,所述步骤4中是通过比对发送端数字指纹和接收端数字指纹是否相同来进行指纹效验。

一种用于数字化车间信息安全的混合加密通信方法

技术领域

[0001] 本发明涉及数字化车间信息安全领域,具体涉及一种用于数字化车间信息安全的混合加密通信方法。

背景技术

[0002] 在数字化车间管理(如图1)中,数据由服务器向终端或终端传输到服务器传输过程中,数据信息容易被监听或者篡改,会造成信息泄露或数据显示虚假数据如果在工控领域可能接收错误的指令信息,从而给工业生产带来巨大危害和损失,所以为了保证终端与服务器以及工业设备和管理系统之间网络通讯的可靠与数据安全,使用安全的通讯连接和可靠的数据加密算法来保证通讯和数据安全非常必要。

[0003] AES是美国联邦政府采用的区块加密标准,用于取代之前的标准DES。全世界各个行业使用AES作为加密基础算法的领域非常广泛。截至2006年AES已经成为全球对称密钥加密中最流行的算法之一。AES具备三种加密特征:1.能最大程度抵抗已知攻击;2.与平台无关,加密解密效率快,编解码紧凑;3.设计简单。因此选择AES在不会损耗太多终端和服务器性能的情况下同时具备了较为可靠的数据加密方案。AES中混乱的密钥分散是分组密码算法设计的基本依据,抵御已知明文差分和线性攻击,变长密钥是设计重点。作为标准的加密算法,各大平台或软件语言内部都集成了标准算法。但集成算法也存在一些安全隐患。在逆向工程中通过分析原始程序内存得到密钥数据并不困难,困难的是逆向出原始的算法。虽然AES作为某些场景的标准,但语言继承算法的通用性太强。因此通过源程序得到密钥然后在用语言继承算法尝试解密的数据破解方式仍然可行,并且太过容易。

[0004] MD5数字指纹或签名算法是加密算法的一个衍生,用于对数据生成唯一的数字指纹或签名数据,不对数据本身进行加密处理,只生成唯一的指纹密钥,指纹算法具备长度固定,容易计算,抗修改以及碰撞概率小的特征。与AES算法类似,作为通用和开放的标准算法,在开发时选择开源或语言自带算法会让系统安全性大打折扣。开源的算法和语言自带的算法由于其开放性,因此被最多的开发人员使用,也产生了最多的指纹字典。理论上MD5不可逆,但由于通用特性已经存在很多的MD5指纹映射数据库,因此短数据通过数据字典映射比对的方式很容易得到元数据。

发明内容

[0005] 针对标准的AES数据加密算法和标准的MD5指纹摘要算法在数据加解密和密文验证过程中存在的缺点与不足,提出一种改进的AES数据加密算法和改进的MD5指纹摘要算法并实现了数字化车间管理系统的通信安全。

[0006] 本发明为实现上述目的所采用的技术方案是:一种用于数字化车间信息安全的混合加密通信方法,包括以下步骤:

[0007] 步骤1:在通讯过程中,发送端使用SSL技术对传输通道进行连接加密;

[0008] 步骤2:发送端对网络传输中的数据进行加密生成加密数据并发送给接收端;

[0009] 步骤3:发送端对加密数据进行处理生成发送端数字指纹;同时接收端对接收的加密数据进行处理生成接收端数字指纹;

[0010] 步骤4:通过对发送端数字指纹和接收端数字指纹进行指纹效验,判断出数据在传输过程中是否被篡改或丢失。

[0011] 所述步骤2中发送端是采用自定义的AES算法对网络传输中的数据进行加密生成加密数据,包括:通过修改AES算法的安全哈希算法字典长度、置换表、以及置换选择和位移规则对要传输的数据片段或文件进行加密。

[0012] 所述修改安全哈希算法字典长度为将字典长度随机修改成一个属于2^{64}范围内的数值。

[0013] 所述修改置换表包括修改初始置换表、逆向初始置换表、扩展置换表。

[0014] 所述修改置换选择和位移规则为将出现置换的位置按照数组范围进行混淆。

[0015] 所述步骤3中发送端是采用自定义的MD5算法对加密数据进行处理生成发送端数字指纹,包括:通过修改MD5算法的置换数组组合和输出字节序列对发送端生成的加密数据进行处理生成发送端数字指纹。

[0016] 所述步骤3中接收端是采用自定义的MD5算法对接收的加密数据进行处理生成接收端数字指纹,包括:通过修改MD5算法的置换数组组合和输出字节序列对接收端接收的加密数据进行处理生成接收端数字指纹。

[0017] 所述修改MD5算法的置换数组组合采用修改轮转和算法分离计算赋值的方法,所述修改输出字节序列采用修改区块转换过程变量的方法。

[0018] 所述步骤4中是通过比对发送端数字指纹和接收端数字指纹是否相同来进行指纹效验。

[0019] 本发明具有以下有益效果及优点:

[0020] 1.安全性高,减少了因标准AES和MD5算法通用性太强容易被攻击破解的可能,改进后的方法在保证数据安全的同时又没有对数字化车间管理系统的性能和网络过多的损耗。

[0021] 2.通用性高,本发明以数字化车间管理系统安全拓展模式来提供一个可靠的信息安全方案,可适用于其他需要的场景。

附图说明

[0022] 图1数字化车间管理架构图;

[0023] 图2为本发明整体流程图;

[0024] 图3SSL技术方案流程图。

具体实施方式

[0025] 下面结合附图及实施例对本发明做进一步的详细说明。

[0026] 如图1所示,数字化车间管理架构图。采集服务器通过串口转网口将工业设备上的数据采集,并以一定的格式存储到database,管理服务器将database中的数据以图表的形式通过管理客户端(Android端和IOS端)供用户查看。

[0027] 本发明的总体流程如图2所示,一种用于数字化车间信息安全的混合加密通信方

法,主要包括以下步骤:

[0028] 步骤1:在通讯过程中使用SSL对传输通道进行连接加密(SSL技术方案流程图如图3所示):(1)利用RSA安全传输AES生成密钥所需的Seed(32字节)。(2)利用AES_encrypt/AES_decrypt对Socket上面的业务数据进行AES加密/解密。理论上只需要AES就能保证全部流程,但由于AES加密所需要的AES-KEY是一个结构。这样一个结构,如果通过网络进行传输,就需要对它进行网络编码,OpenSSL里面没有现成的API所以就引入RSA来完成首次安全的传输,保证Seed不会被窃听。同样,只使用RSA也能完成全部流程,但由于RSA的处理效率比AES低,所以在业务数据传输加密上还是使用AES。在实际的Socket应用开发时,需要将这些步骤插入到Client/Server网络通信的特定阶段。

[0029] 步骤2:使用自定义的AES算法对网络传输中的数据进行加密:(1)修改标准算法的安全哈希算法字典长度,随机修改一个基于2^{64}范围内的数值。只要不与开源的库中的数值相同即可,修改此值将导致加密结果与标准算法或多数开源库算法产生的结果出现最大可能的差异化,并且无法使用标准算法和开源库算法解密。但从内存中取得字典长度的并不困难,因此为了进一步加强安全,还需要进一步对算法修正。(2)修改算法置换表,AES包含多种置换,包括但不限于初始置换,逆向初始置换,扩展置换。方案示例:例如标准算法置换数组如下{14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7},我们将其修改为{16,6,12,1,5,11,15,9,4,13,3,14,4,7,9,0}。上述示例描述一个置换参数赋值来产生不同的结果。假设1+1=2但是我们现在将加数和被加数值修改,那么结果等于2的可能就不会太大,即便相同的结果解密还原那么也会还原出不同的数据。以此进行数组置换实现与标准算法和开源算法不同的数组顺序从而产生截然不同的加密结果。如果算法独立,那么表示无法与标准算法兼容,从而最大可能的提升算法安全。

[0030] 步骤3:在发送端和接收端加密数据后使用自定义的MD5算法生成数字指纹:

[0031] (1)修改置换数组组合,例如:

[0032] 假设组FF中的填充顺序为(a,b,c,d,x,s,ac),而内部的计算方式为(a)+=F((b),(c),(d))+(x)+ac;那么将填充和计算顺序自定义。

[0033] 假设组FF中填充的数值定义如(a,b,c,d,x[0],S11,0xd76aa478);根据自己的规定填充为(c,a,b,d,x[0],S11,0xd76aa478)。

[0034] (2)修改输出字节序列。在outpout中假设标准输出为:output[j+1]=(byte)((input[i]>>8)&0xff);自定义输出如下:output[j+3]=(byte)((input[i]>>8)&0xff);

[0035] 上述内容为,如果标准算法定义了一个数组填充123456789那么我们自己定义一个填充数组为987654321,参考标准并不遵循标准,而填充过程与算法是内部的,无法被看到,因此将填充规则改变之后数据结果将不再与标准算法相同,而标准算法也无法解密其数据。

[0036] 经过上述技术手段,MD5算法的加密结果将完全不同于标准和开源库所产生的结果,以此防止通过字典碰撞来破解数据的可能。

[0037] 步骤4:通过指纹效验判断数据在传输过程中是否被篡改或丢失。

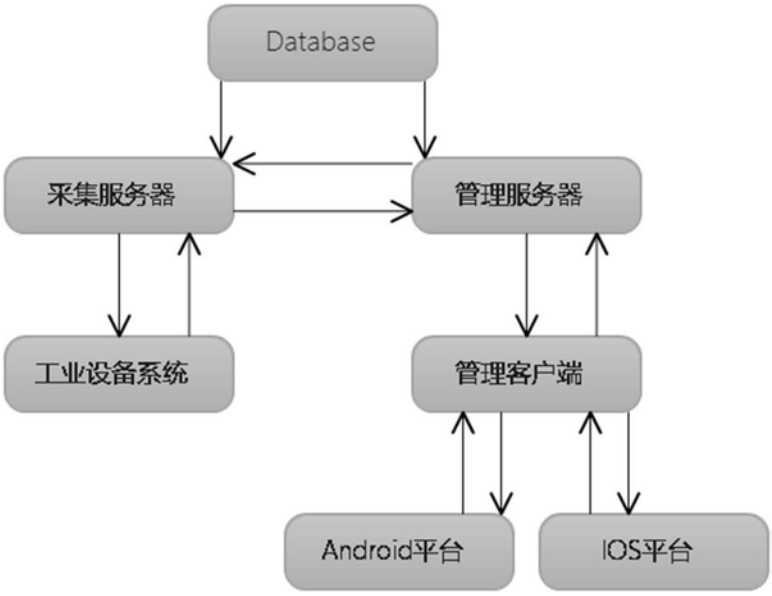

Database
采集服务器
管理服务器
工业设备系统
管理客户端
Android平台
IOS平台

图1

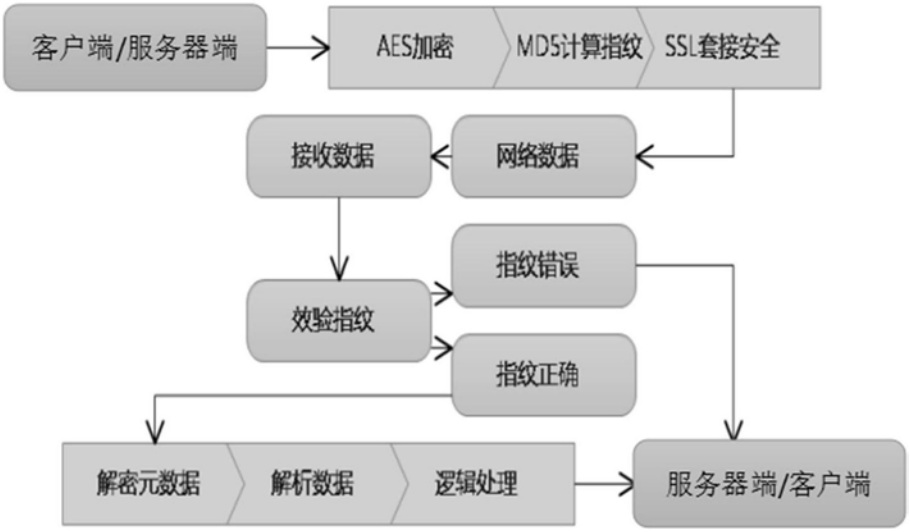

客户端/服务器端
AES加密
MD5计算指纹
SSL套接安全
接收数据
网络数据
指纹错误
效验指纹
指纹正确
解密元数据
解析数据
逻辑处理
服务器端/客户端

图2

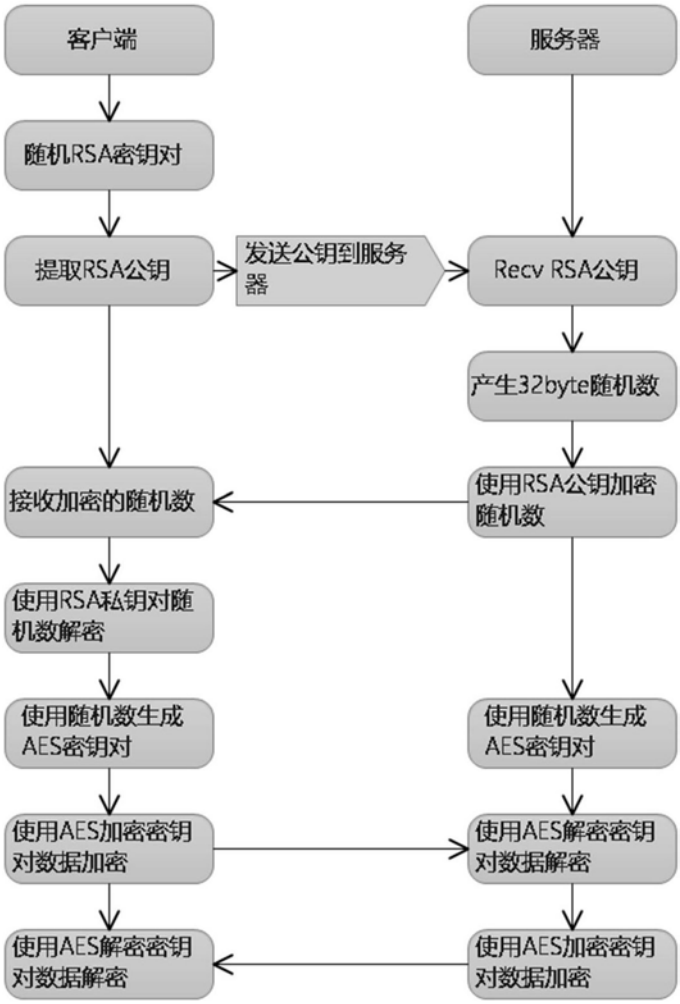

客户端
服务器
随机RSA密钥对
提取RSA公钥
发送公钥到服务器
Recv RSA公钥
产生32byte随机数
接收加密的随机数
使用RSA公钥加密随机数
使用RSA私钥对随机数解密
使用随机数生成AES密钥对
使用随机数生成AES密钥对
使用AES加密密钥对数据加密
使用AES解密密钥对数据解密
使用AES解密密钥对数据解密
使用AES加密密钥对数据加密

图3