软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(1):129−144 [doi: 10.13328/j.cnki.jos.004553]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种用于流交换的代理重签名方案*

孙 奕[1,2,3], 陈性元[2], 杜学绘[2,3], 陈 亮[2], 徐 建[2]

[1](北京交通大学 计算机与信息技术学院,北京 100044)

[2](解放军信息工程大学,河南 郑州 450000)

[3](数学工程与先进计算国家重点实验室,河南 郑州 450000)

通讯作者: 孙奕, E-mail: 11112072@bjtu.edu.cn

摘 要: 为了解决大规模复杂网络环境下流交换的安全问题,首次将代理重签名技术应用到安全流交换中,提出一种用于流交换的基于陷门 Hash 函数的代理重签名方案.首先,针对陷门 Hash 函数在流交换应用中存在的密钥泄露问题,提出一种新的基于椭圆曲线的无密钥泄露的陷门 Hash 函数(EDL-MTH),并对其安全性加以证明;然后,基于 EDL-MTH 构造了一个在随机预言模型下可证明适应性选择消息攻击安全的代理重签名方案;最后,通过一个示例分析了该方案在安全流交换中的应用和性能.

关键词: 流交换;陷门 Hash 函数;代理重签名;适应性选择消息攻击

中图法分类号: TP309 **文献标识码**: A

中文引用格式: 孙奕,陈性元,杜学绘,陈亮,徐建.一种用于流交换的代理重签名方案.软件学报,2015,26(1):129−144. http://www.jos.org.cn/1000-9825/4553.htm

英文引用格式: Sun Y, Chen XY, Du XH, Chen L, Xu J. Proxy re-signature scheme for stream exchange. Ruan Jian Xue Bao/ Journal of Software, 2015,26(1):129−144 (in Chinese). http://www.jos.org.cn/1000-9825/4553.htm

Proxy Re-Signature Scheme for Stream Exchange

SUN Yi[1,2,3], CHEN Xing-Yuan[2], DU Xue-Hui[2,3], CHEN Liang[2], XU Jian[2]

[1](School of Computer & Information Technology, Beijing Jiaotong University, Beijing 100044, China)

[2](PLA Information Engineering University, Zhengzhou 450000, China)

[3](State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450000, China)

Abstract: To tackle the problems of security stream exchange in the large-scale complicated network, this paper applies proxy re-signature technology for the first time to solve the flow exchange security issues, and proposes a proxy re-signature scheme based on trapdoor Hash function for stream exchange. Firstly, aiming at the key exposure problem of trapdoor Hash function for stream exchange, a new trapdoor Hash functions without key exposure (EDL-MTH) is put forward and its security is analyzed. Then, a new proxy re-signature scheme based on EDL-MTH is constructed and is proved against the chosen-message attack in the random oracle model. Furthermore, the performance of the scheme is analyzed contrast to the existing proven security proxy signature scheme, and the result shows the efficiency becomes more prominent while the scale of stream exchange is increased. Finally, a case study is provided to demonstrate its availability and performance in security stream exchange.

Key words: stream exchange; trapdoor Hash functions; proxy re-signature; chosen adaptively message attack

1 引 言

计算机网络的出现,使得一个个孤立的信息通过数据交换构成了一个广阔的信息共享与交换平台,因此在网络中处处渗透着信息交换的思想.然而人们在享受信息共享与交换带来巨大优越性的同时,随之而产生的病

* 基金项目: 国家高技术研究发展计划(863)(2012AA012704); 河南省科技创新人才计划(114200510001)
收稿时间: 2013-03-19; 修改时间: 2013-08-21; 定稿时间: 2013-12-09

毒传播、恶意攻击、非授权访问、信息泄露等问题也对网络的健康发展构成了重大的威胁.为此,人们采取了许多安全措施,如网络隔离、通过网闸摆渡要交换的数据、划分不同的安全域、在每个区域边界上部署防火墙对交换的数据进行安全检测等,这些技术解决了一部分数据安全交换的问题.但是,随着云计算、传感器网络、P2P 网络等大规模复杂网络的快速发展,对数据安全交换技术提出了新的挑战.交换的数据不仅仅是静态的、固定大小的已知数据,而是一种连续的、无限的、快速的、随时间变化的流(如电视会议、实时监控、在线视频、股票交易等数据).所谓安全流交换,就是指将这种流从一个网络域安全的流转到另一个安全级别更高的网络域.传统的方法是在区域边界上设置防火墙来保证流入数据的安全性,但是现在网络上很多攻击都能穿透防火墙,因此需要在应用层缓存整个流再对数据进行复杂的信息过滤和检测等.这种方式不能满足时延低、实时性高、计算复杂度小、所需存储空间少的要求,而且针对流的特点缓存整个流再进行验证和检测也是不现实的.

针对流的特点,实现流交换的安全性关键有 3 点:

(1) 流的可信性由流源头来负责,即,从流的发源地确定流的身份及其可信性;

(2) 由于流的实时性和连续性,要能够即来即验证,实现对流片段完整性的实时检测;

(3) 流经路径的可信性,即,存在一条可验证安全性的虚拟的流交换路径.

基于以上分析,代理重签名技术凭借其独特的签名转换功能[1,2],可以实现很多适合于跨域流交换所需要的功能,如跨域透明认证、提供遍历的路径证明、分散代理的签名权利、简化证书管理等.但是现有的可证明安全性的代理重签名算法通常需要进行幂运算,效率较低,延时较长,不能满足实时性较高的动态流交换要求.因此,本文将陷门 Hash 函数的概念引入到代理重签名算法中,构造了一个可证明安全的适用于流交换的代理重签名方案.

1.1 相关工作

1.1.1 陷门 Hash 函数

陷门 Hash 函数的概念最早来源于 Brassard 等人[3]提出的陷门承诺的思想,Krawczyk 和 Rabin[4]首次构造了一种陷门 Hash 函数,并基于该陷门 Hash 函数构造了一种变色龙签名方案.陷门 Hash 函数也称为变色龙 Hash 函数,是一种单向陷门函数,可以防止除陷门信息拥有者以外的任何人,能够对每次给定的输入计算出碰撞.陷门 Hash 函数最初被用来设计变色龙签名和不可否认签名[5],能够实现签名信息的不可抵赖性和不可传递性.1990 年,Even 等人[6]首次提出了在线/离线签名方案.后来,Shamir 和 Tauman[7]应用陷门 Hash 函数开发了一种新的机制,称为“Hash-sign-switch”,它能够将任何一种签名方案转化为一种在线/离线签名方案.基于陷门 Hash 函数构造的在线/离线签名方案与变色龙签名方案所不同的是:变色龙签名方案中是接收者拥有陷门信息,而在线/离线签名方案中是签名者拥有陷门信息.

Ateniese 和 deMedeiros[8]在 2004 金融密码年会上第一次提出了陷门 Hash 函数密钥泄露的问题,并首次提出基于身份的变色龙 Hash 函数来解决密钥泄露的问题.所谓陷门 Hash 函数密钥泄露问题,是指当有多个用户对不同消息使用相同的陷门 Hash 值时,将会导致陷门密钥的泄露,详细说明见第 2.3 节.文献[8]的主要思想是,基于身份和一次交易信息构成一次性密钥信息来解决密钥泄露的问题.本方案中,签名伪造只能导致签名者恢复与一个交易相关的陷门信息,因此签名者不能拒绝其他交易消息的签名,同时又没有泄露密钥.但是这种思想只能解决密钥泄露的部分问题,因为每次交易都需要改变接收者的公钥.同年,Chen 等人[9]第一次使用双线性对构造了基于 GDH 无密钥泄露的变色龙 Hash 函数.后来,Ateniese 和 deMedeiros 又在文献[10]中对文献[8]进行改进,提出了 3 种无密钥泄露的变色龙 Hash 函数:一种基于双线对,另两种基于 RSA 假设.2009 年,Gaoetal[11]提出一种基于 Schnorr 签名的无密钥泄露的变色龙 Hash 函数,然而该方案在签名者和接收者之间有一个交互式的协议,违背了最初定义变色龙 Hash 函数和签名方案的本意.

不同的无密钥泄露的陷门 Hash 函数方案构造的签名方案主要分为两大类:

- 一类用于构造变色龙签名方案,该类方案通常采用一种短签名来构造一次性密钥来解决密钥泄露问题,如文献[8−12];
- 另一类用于构造在线/离线签名方案,该方案通常是通过构造不同的陷门元素来解决密钥泄露问题.例

如:文献[13,14]提出一种特殊双陷门 Hash 函数族,将密钥分为两部分——长久密钥和一次性密钥;文献[15]分别基于 DL 和因式分解构造了多碰撞陷门 Hash 函数族;文献[16]在文献[14]的基础上加以改进,引入对称密钥的概念,构造了一种 3 陷门 Hash 函数.

但是,以上这些陷门 Hash 函数不适合实时的、存储空间有限的、连续的流交换方案.

最近,Chandrasekhar[17]提出了一种新颖的对陷门 Hash 函数的应用,将陷门 Hash 函数应用到签名分期偿还机制中,构造了一种基于陷门 Hash 函数的流认证方案.该方案大大提高了对流认证的效率,本文正是由此得到启发,将其扩展到代理重签名方案中,首次提出一种新颖的、高效的代理重签名方案.与文献[17]中的方案不同:文献[17]中所提方案的陷门密钥由发送方掌握;而本文的方案陷门密钥由代理掌握,并且将一个由两方参与的流认证方案扩展为可由第三方控制的安全流交换方案.此外,基于本文所提出的陷门 Hash 函数构造的流交换方案消除了文献[17]所提方案中的幂运算和逆运算,极大地提高了计算效率,缩短了交换时延.

1.1.2 代理重签名

代理重签名是现代密码学的一个新兴研究领域.在 1998 年的 EUROCRYPT 上,Blaze,Bleumer 和 Strauss (BBS)[18]首次提出了代理重签名(proxy re-signature)的概念.代理重签名体制中存在一个半可信的代理,可以把受托者的签名转换为委托者关于同一个消息的签名,而这个代理人不能得到受托者或委托者的签名密钥,也不能任意生成他们的签名.

代理重签名的概念虽然很早就被提出,但是由于 Blaze 等人未给出代理重签名的形式化定义,人们很容易把它与其他签名类型(如代理签名、传递签名、群签名、多签名、聚合签名等)相混淆,而且 Blaze 等人提出的第一个代理重签名方案在受理人签名的产生过程和签名转换的过程中,用户需要进行 k 个幂指数计算,计算效率低,不适合在实际应用中使用.为了改变这种情况,Ateniese 等人[19]给出了代理重签名形式化的定义,特别是与上述几种签名类型加以区别,并提出两个方案 S_{bi} 和 S_{uni},其中,S_{bi} 是双向的、多用的,S_{uni} 是单向的、单用的.但是,他们对安全属性的描述不够简练,所给出的用公式表示的安全模型存在很多冗余.第一个可证明安全的代理重签名方案 S_{mb} 是由 Shao 等人[20]提出来的,同时还提出了第一个基于身份的方案 $S_{id\text{-}mb}$.这两个方案最主要的问题是需要大量的公开参数和耗时的计算,而且他们定义的安全模型有很多限制,影响了方案在实际应用中的推广.

近年来,代理重签名成为密码学研究的一个热点,除了这些基本的代理重签名方案[19−22]以外,还有一些特殊用途的代理重签名方案.如:研究从一种签名机制转换为另一种签名机制的代理重签名方案[23,24];研究解决代理重签名密钥泄漏问题的门限代理重签名方案[25−28];研究解决隐私保护问题的盲代理重签名[29];无证书代理重签名[30];简单、通用、可组合代理重签名方案[31]等.但这些方案都没有考虑对实时性、海量的动态流数据进行认证和签名转换的效率问题,成为代理重签名方案在大规模海量流数据中应用的瓶颈.本文将无密钥泄露的陷门 Hash 函数引入到代理重签名算法中,不仅可以大大提高代理重签名的实时性和安全性,还可以使代理重签名方案以一种高效的方式处理流数据.

1.2 论文的组织

本文第 1 节介绍一些预备知识.第 2 节描述本文提出的一种新的无密钥泄露的陷门 Hash 函数,并对其安全性进行分析.第 3 节给出基于该陷门 Hash 函数的代理重签名方案的形式化定义.第 4 节基于椭圆曲线离散对数给出具体的用于流交换的基于陷门 Hash 函数的代理重签名方案.第 5 节对方案的安全性及效率进行分析,并在随机预言模型下证明该方案在适应性选择消息攻击下是不可伪造的.同时,将方案的效率与现有的几种典型的可证明安全性的代理重签名方案进行对比分析.第 6 节举例说明本文提出的代理重签名方案在流交换中的应用,并分析本方案在安全性方面和性能方面与传统的流交换方案相比的优势.第 7 节给出论文的结论.

2 预备知识

2.1 椭圆曲线离散对数假设

令 l 为一个素数的幂,$E(F_l)$是有限域 F_l 上的椭圆曲线.令#$E(F_l)$为 $E(F_l)$的阶,$E(F_l)$中元素 P 的阶为素数 q 且

$q|\#E(F_l)$.记 G 是 P 生成的一个 q 阶循环群.椭圆曲线离散对数问题(ECDLP):给定$(P,Q)\in E(F_l)$,寻找一个整数$a\in Z_q$,使得在 G 中有 $Q=aP$.

定义 1(椭圆曲线离散对数假设). 如果没有一种概率多项式时间(PPT)算法在时间 T 内以至少ε的概率解决群 G 上的椭圆曲线离散对数问题,则称群 G 上的(T,ε)-ECDLP 假设成立.

2.2 陷门Hash函数

定义 2(陷门 Hash 族). 一个陷门 Hash 族由一对二元组(I,H)所组成:

- I 是一种概率多项式时间的密钥生成算法,输入 1^k,输出 Hash/陷门密钥对(HK,TK),使得 HK,TK 的长度是 k 的多项式;
- H 是一个随机的 Hash 函数族,H 中的每一个函数关联一个 Hash 密钥 HK,其作用于消息空间 M 中的一个消息和有限空间 R 中一个随机数.Hash 函数 H_{HK} 的输出与 TK 无关.

一个陷门 Hash 函数族(I,H)满足如下性质:

- 有效计算:给定 Hash 密钥 HK 和一对$(m,r)\in M\times R$,$H_{HK}(m,r)$在多项式时间内可计算;
- 抗碰撞:不存在多项式时间的算法 A,在只输入 HK 的情况下,以不可忽略的概率得到两对(m_1,r_1),$(m_2,r_2)\in M\times R$ 满足 $m_1\neq m_2$ 且 $H_{HK}(m_1,r_1)=H_{HK}(m_2,r_2)$(其概率与 HK 有关,在这里$(HK,TK)\leftarrow I(1^k)$,且与算法 A 投掷的随机硬币有关);
- 陷门碰撞:存在一种概率多项式时间的算法.输入$(HK,TK)\leftarrow I(1^k)$,一对$(m_1,r_1)\in M\times R$ 和消息 $m_2\in M$,输出 $r_2\in R$ 满足:$H_{HK}(m_1,r_1)=H_{HK}(m_2,r_2)$.如果 r_1 在 R 上服从均匀分布,则 r_2 在 R 上服从的分布与均匀分布在计算上是不可区分的.

2.3 代理重签名的标准模型

定义 3[19]. 代理重签名方案是一个多项式时间内的元组 $PRS=(KeyGen,ReKey,Sign,ReSign,Verify)$,满足:

(1) $(KeyGen,Sign,Verify)$是标准的密钥生成、签名、验证算法;

(2) 输入(pk_A,sk_A,pk_B,sk_B)后,重签名密钥生成算法 Rekey 为代理产生一个密钥 $rk_{A\to B},rk_{A\to B}$,能将 Alice 的签名转换为 Bob 的签名(Bob 是委托者,Alice 是受托者),pk_A 和 sk_A 是 Alice 的公钥和私钥,pk_B 和 sk_B 是 Bob 的公钥和私钥,sk_A 在这个式子中不是必需的;

(3) 输入 Alice 的公钥 pk_A、消息 m、Alice 在 m 上的签名 $\sigma_A(m)$,重签名函数 $ReSign$ 后,若 $Verify(pk_A,m,\sigma_A(m))=1$,则输出 $\sigma_B(m)$;否则,输出⊥.

正确性:正确性要满足两个条件.对于消息空间中的任何消息 m 和公钥密钥对(pk,sk),$(pk',sk'\leftarrow KeyGen(1^k))$,假定$\sigma=Sign(sk,m)$和 $rk\leftarrow ReKeyGen(pk,sk,pk',sk')$,则必须满足以下两个条件:

- $Verify(pk,m,\sigma)=1$;且
- $Verify(pk',m,ReSign(rk,\sigma))=1$.

即,所有由签名算法和重签名算法合法产生的签名均能通过签名验证.

3 一种新的无密钥泄露的陷门 Hash 函数

本节在文献[13,14]定义的无密钥泄露陷门 Hash 函数的基础上进行改进,构造了一个新的基于椭圆曲线离散对数假设的无密钥泄露陷门 Hash 函数方案,并对其安全性进行了分析.

3.1 形式化定义

定义 4(双陷门 Hash 函数族). 一个双陷门 Hash 函数族由一个三元组(I,I',H)所组成:

- I是一种概率多项式时间的密钥生成算法,输入 1^k,输出一对长久 Hash/陷门密钥对(HK,TK),使得 HK,TK 的长度是 k 的多项式有关;
- I'是一种概率多项式时间的密钥生成算法,输入 1^k,输出一对一次性 Hash/陷门密钥对(HK',TK'),使得 HK,TK 的长度是 k 的多项式有关;

- H 是一个随机的 Hash 函数族.输入 1^k,一对 Hash/陷门密钥对(HK,TK),一对$(m,r)\in M\times R$ 和一个消息 $m'\neq m$,输出一个碰撞参数 r'和 HK',使得 $H_{HK}(m,r)=H_{HK'}(m',r')$.当 $HK\neq HK'$时,HK'与其关联的陷门密钥 TK'称为一次性密钥对.

定义 5. 一个双陷门 Hash 函数族(I,I',H)应满足如下性质:

性质 1(有效计算). 给定 Hash 密钥 HK 和一对$(m,r)\in M\times R$,$H_{HK}(m,r)$在多项式时间内可计算.

性质 2(陷门碰撞). 存在一种概率多项式时间的算法.输入$(HK,TK)\leftarrow I(1^k)$,一对$(m,r)\in M\times R$ 和消息 $m'\in M$,输出 $r'\in R$ 满足:$H_{HK}(m,r)=H_{HK'}(m',r')$.如果 r 在 R 上服从均匀分布,则 r'在 R 上服从的分布与均匀分布在计算上是不可区分的.

性质 3(抗碰撞). 不存在多项式时间算法 A,在只输入 HK 的情况下,以不可忽略的概率得到两对(m,r),$(m',r')\in M\times R$ 满足:$[m\neq m']\wedge[TH_{HK}(m,r)=H_{HK'}(m',r')]$.当 $HK=HK'$时称为简单碰撞,当 $HK\neq HK'$时称为一次性碰撞.

性质 4(抗密钥泄露). 不存在多项式时间的算法 A,在输入一个长久 Hash 密钥 HK,两个一次性 Hash 密钥 HK'和 HK'',两对(m,r),$(m',r')\in M\times R$ 并且满足$[m\neq m']\wedge[TH_{HK}(m,r)=H_{HK'}(m',r')]$情况下,能够以不可忽略的概率得到长久陷门密钥 TK.

性质 5(语义安全). 在给定消息 m 的陷门 Hash 值 C 的情况下,m 的条件熵 $H[m|C]$等于 m 的熵 $H[m]$.换句话说,消息 m 的陷门 Hash 值 C 没有泄露任何关于 m 的信息.

3.2 一种基于椭圆曲线的无密钥泄露的陷门Hash函数方案

本节基于椭圆曲线离散对数构造一种新的无密钥泄露的双陷门 Hash 函数方案.

定义 6(一种基于椭圆曲线离散对数的无密钥泄露的陷门 Hash 方案 EDL-MTH). EDL-MTH 是一个四元组:

$$TH=(SysParGen,KeyGen,THGen,TrapColGen).$$

- $SysParGen$:令 l 为一个素数的幂,$E(F_l)$是有限域 F_l 上的椭圆曲线.令$\#E(F_l)$为 $E(F_l)$的阶,$E(F_l)$中元素的 P 阶为素数 q 且 $q|\#E(F_l)$.记 G 为由元素 P 生成的子群.定义一个安全 Hash 函数 $f:Z_q\times G\rightarrow Z_q$,则系统参数为 $params=\{G,q,P,f\}$;
- $KeyGen$:使用系统参数 $params$ 生成陷门/Hash 密钥对,$(TK,HK)=(\alpha,Y)$,这里随机选择 $\alpha\in_R Z_q^*$,计算 $Y=\alpha P$.
- $THGen$:实体选择元素 $r\in Z_q$,使用 Hash 密钥 Y 生成消息 $m\in Z_q$ 的陷门 Hash 函数,陷门 Hash 函数定义为
$$TH_Y(m,r)=f(m,Y)Y+rP.$$
- $TrapColGen$:给定陷门/Hash 密钥对,$(TK,HK)=(\alpha,Y)$, $(m\times r)\in Z_q^*\times Z_q^*$ 及 $m'\in Z_q^*$,计算碰撞 $r'\in_R Z_q^*$.步骤如下:
  (1) 选择一次性陷门 $\beta\in_R Z_q^*$,计算 $K=\beta P$;
  (2) 使用陷门密钥α和β计算 r',使得 $TH_Y(m,r)=TH_K(m',r')$,计算 $r'=\alpha f(m,Y)-\beta f(m',K)+r$.

3.3 EDL-MTH方案安全性分析

证明 EDL-MTH 方案的安全性,即证明在群 G 上的离散对数问题假设下,EDL-MTH 满足定义 5 的 5 种性质.

定理 1. EDL-MTH 方案在(T,ε)-ECDLP 假设下,群 G 上的椭圆曲线离散对数问题是难解的.

证明:

(1) 有效性

给定 Hash 密钥 HK 和一对$(m,r)\in M\times R$,可以在多项式时间内计算出 $H_{HK}(m,r)$.

(2) 碰撞性

假设给定 Hash 密钥(HK,HK')、陷门密钥(TK,TK')、一对$(m,r)\in M\times R$ 和消息 $m'\in M$,找到 r',使得:

$$f(m,Y)Y+rP=f(m',K)K+r'P.$$

r'值可通过下式计算得出:

$$r'=\alpha f(m,Y)-\beta f(m',K)+r.$$

如果 r 在 R 上服从均匀分布,则 r'在 R 上服从的分布与均匀分布在计算上是不可区分的.

(3) 抗碰撞性

基于定义 5,我们需要考虑两种情况:

(a) $HK=HK'$(针对抵抗简单碰撞伪造);

(b) $HK\neq HK'$(针对抵抗一次性碰撞伪造).

- 情况(a):当 $HK=HK'$时.

抗碰撞伪造意味着假设输入系统参数 *params* 和 Hash 密钥 HK,不存在一个 PPT 碰撞伪造者 F,能够以不可忽略的概率ε成功输出,满足$[m\neq m']\wedge[TH_{HK}(m,r)=H_{HK}(m',r')]$.

反证法.假设存在一个 PPT 碰撞伪造者 F,能够以不可忽略的概率ε成功输出,满足$[m\neq m']\wedge[TH_{HK}(m,r)=H_{HK}(m',r')]$.给定一个离散对数难题的实例$\langle G,q,P,Y\rangle$,其中,陷门/Hash 密钥为$(y,Y=yP)$.由于 $TH_{HK}(m,r)=H_{HK}(m',r')$,可以得到等式:$yf(m,Y)+r=yf(m',Y)+r' \bmod q$,从而推出 $y=((f(m,Y)-f(m',Y))^{-1}+r'-r) \bmod q$.

因此可以求出 y,违背(T,ε)-ECDLP.得证.

事实上,情况(a)中描述的简单碰撞问题,就是引言中提到的著名的密钥泄露问题.通过以上分析可知:如果给定两对消息满足$(m,r),(m',r')\in M\times R$,则第三方就可以通过 $TK=((f(m,HK)-f(m',HK))^{-1}+r'-r) \bmod q$ 成功地计算出陷门密钥 TK,从而产生密钥泄露的问题.

- 情况(b):当 $HK\neq HK'$时.

假设这里存在一个 PPT 碰撞伪造者 F,能够以不可忽略的概率ε抵抗定义 6 中提出的陷门 Hash 方案.给定 Hash 密钥 $HK,HK'\neq HK$ 和系统参数 *params*,在多项式时间运行 F 并输出$\langle m,r,m',r'\rangle$,这里,$m\neq m',r\neq r',TH_{HK}(m,r)=TH_{HK'}(m',r')$具有不可忽略的概率.假设 F 可以构造一个 PPT 算法 h,能够破解离散对数难题.给定一个离散对数难题的实例$\langle G,q,P,X\rangle$,h 需要找到 $x\in Z_q^*$ 满足 $X=xP$,h 选择 $y\in_R Z_q^*$ 计算 $Y=yP$,独立平行运行伪造者 F 的两个实例,F 的每个实例输入$\langle G,q,P,X\rangle$是随机选择的.直到 F 的每个实例分别产生一个碰撞伪造 $\langle m_1,r_1,m_1',r_1'\rangle$ 和 $\langle m_2,r_2,m_2',r_2'\rangle$,如果 $m_1=m_2$ 或者 $r_1=r_2$ 或者 $m_1'=m_2'$ 或者 $r_1'=r_2'$,F 重复执行.

给定 $TH_{HK}(m_1,r_1),TH_{HK'}(m_1',r_1')$ 并且 $TH_{HK}(m_2,r_2)=TH_{HK'}(m_2',r_2')$,我们获得下面两个线性等式:

$$xf(m_1,X)+r_1=yf(m_1',Y)+r_1' \bmod q,$$

$$xf(m_2,X)+r_2=yf(m_2',Y)+r_2' \bmod q.$$

显然,以上两个线性等式是可解的,因此可以求出 x 和 y,违背(T,ε)-ECDLP 假设.得证.

(4) 抗密钥泄露性

根据 EDL-MTH 方案,所谓抗密钥泄露意味着给定两个元组$\langle m,r,HK\rangle,\langle m',r',HK'\rangle$,这里,$m\neq m',r\neq r'$并且 $TH_{HK}(m,r)=TH_{HK'}(m',r')$,能够输出 TK 的一种 PPT 算法的概率是可以忽略的.

反证法.假设这里存在一种 PPT 算法,能够在多项式时间以不可忽略的概率输出 TK,则可以通过下式计算出 Hash 密钥 HK'的离散对数 TK':

$$TK'=(f(m',HK')^{-1}(f(m,HK)TK+r-r') \bmod q.$$

显然违背(T,ε)-ECDLP 假设.得证.因此,提出的陷门 Hash 方案是抗密钥泄露的.

(5) 语义安全性

由于 m,r 是独立的变量.因此条件概率$\mu(m|C)=\mu(m|r)$成立,那么我们就能证明条件熵 $H[m|C]=H[m]$:

$$\begin{aligned}H[m\mid C]&=-\sum_m\sum_c\mu(m,c)\log(\mu(m\mid c))\\&=-\sum_m\sum_c\mu(m,c)\log(\mu(m))\\&=-\sum_m\mu(m)\log(\mu(m))\\&=H[m].\end{aligned}$$

综上所述,基于(T,ε)-ECDLP 假设,EDL-MTH 方案满足可计算性、碰撞性、抗碰撞性、抗密钥泄露性和语

义安全性. □

4 基于陷门 Hash 函数的代理重签名方案的形式化定义

定义 7(基于陷门 Hash 函数的代理重签名). 一个基于陷门 Hash 函数的代理重签名方案是一个六元组:

$$MTH\text{-}PRS=(ParGen,KeyGen,ReKeyGen,ReSign_{ini},ReSign_{sub},Verify).$$

- *ParGen*:输入一个安全参数,输出系统参数 *par*;
- *KeyGen*:输入系统参数 *par* 生成系统私/公钥对(sk,pk),陷门/Hash 密钥对(TK,HK)=(x,Y);
- *ReKeyGen*:输入一个受托者的公私钥对(pk_A,sk_A)和一个委托者的公私钥对(pk_B,sk_B),输出一个重签名密钥 $rk_{A\to B}$.代理者使用 $rk_{A\to B}$ 可将受托者的签名转换为委托者的签名;
- $ReSign_{ini}$:初始化签名算法(与在线/离线算法不同的是,陷门密钥由代理掌握):
  (1) 设初始化消息为 m',选择一个随机数 r',计算并存储 $h=TH_{Y_0}(m',r')$, 这里,Y_0 为长久 Hash 密钥,则其相应的长久陷门密钥为 x_0;
  (2) 受托者运行 PRS 的签名算法 *Sign* 生成对 h 的原始签名 $\sigma_A(h)$,但代理者不知道受托者的私钥 sk_A 的任何信息;
  (3) 代理使用 pk_A,m',r'验证 $\sigma_A(h)$的有效性,若无效,则终止;
  (4) 否则,运行 PRS 的重签名算法 *ReSign*,输入 $\sigma_A(h)$,$rk_{A\to B}$,m',r',生成重签名 $\sigma_B(h)$;
- $ReSign_{sub}$:后续信息重签名算法:
  (1) 输入待签名消息 m_i,一个私钥 sk_A,运行 PRS 的签名算法 *Sign* 生成消息 m_i 的签名 $\theta_A(m_i)$;
  (2) 使用公钥 pk_A 验证 $\theta_A(m_i)$的有效性,若 $Verify(pk_A,m_i,\theta_A)=1$,随机选择 Y_i 并计算 r_i 满足 $TH_{Y_0}(m',r')=TH_{Y_i}(m_i,r_i)$,输出 $\theta_B(m_i)=(r_i,Y_i)$;否则,输出⊥;
- *Ver*:验证算法:
  (1) 初始信息验证.输入消息 m',r',计算 $h=TH_{Y_0}(m',r')$, 并保存在缓存中.若验证 $Verify(pk_B,TH_{Y_0}(m',r'),\sigma_B)=1$,则 $\sigma_B(h)$是对应于公钥 pk_B 的消息 m'的合法重签名;否则,输出⊥;
  (2) 输入消息 m_i,若验证 $Verify(pk_B,TH_{Y_i}(m_i,r_i),\theta_B)=1$,则 θ_B 是对应于公钥 pk_B 的消息 m_i 的合法重签名;否则,输出⊥.

5 基于 EDL-MTH 的代理重签名方案

本节基于第 3 节提出的 EDL-MTH 方案构造一个新的代理重签名方案 MTH-PRS,该方案可以将已有的任意一个代理重签名方案转换为一个用于流交换的代理重签名方案.本方案主要由五方实体参与:委托者、受托者(发送方)、代理、接收方和密钥管理中心.密钥管理中心负责密钥及重签名密钥的管理,由于篇幅所限,具体细节不在本文中讨论.这里,受托者相当于流的发送方,代理相当于一个具有认证和密钥转换功能的安全流交换服务器,委托者相当于流的被授权访问者,接收方通过验证委托者的签名判断交换的流的安全性.接收方不能直接验证发送方的签名,只能验证经过安全流交换服务器认证后转换的委托者的签名,从而实现由安全流交换服务器控制的跨域流交换.

定义 8(流). 用 $S=\{b_0,b_1,\ldots,b_n\}$表示一个逻辑上有序的流,其中,$b_i=(m_i,\sigma)(0\leqslant i\leqslant n)$表示流中的一个片段,$m_i$ 表示片段 b_i 的信息,σ表示片段 b_i 的签名.根据实现签名的层次不同,片段代表的含义有所不同.如:在网络层实现签名,则一个片段可以表示为一个数据包.

一个基于陷门 Hash 函数的代理重签名方案 $MTH\text{-}PRS=(ParGen,KeyGen,ReKeyGen,ReSign_{ini},ReSign_{sub},Ver)$ 由以下有效算法组成:

1. 系统参数生成 *ParGen*:

令 l 为一个素数的幂,$E(F_l)$是有限域 F_l 上的椭圆曲线.令$\#E(F_l)$为 $E(F_l)$的阶,$E(F_l)$中元素的 aP 阶为素数 q,且 $q|\#E(F_l)$.记 G 为由元素 P 生成的子群.定义一个安全 Hash 函数 $f:Z_q\times G\to Z_q$.给定陷门密钥/Hash 密钥对:(TK,

HK)=(α,Y),这里随机选择 $\alpha \in_R Z_q^*$,计算 $Y=\alpha P$.陷门 Hash 函数定义为

$$TH_{HK}(m,r)=f(m,Y)Y+rP.$$

令 PRS=($KeyGen$,$ReKey$,$Sign$,$ReSign$,$Verify$)为任何一个可证明安全的代理重签名方,则系统参数为

$$par=\{E(F_l),G,q,P,f,TH_{HK},PRS\}.$$

2. 密钥生成 *KeyGen*:

- 输入 1^k,运行原始签名方案的密钥生成算法 *KeyGen*,输出一个受托者的公私钥对(pk_A,sk_A)和一个委托者的公私钥对(pk_B,sk_B);
- 输入 1^k,运行陷门 Hash 函数的密钥生成算法,输出长期 Hash/陷门密钥对($HK=Y=xP$,$TK=x$).

3. 重密钥生成 *ReKeyGen*:

运行 PRS 的重密钥生成算法 *ReKey*,输入一个受托者的公私钥对(pk_A,sk_A)和一个委托者的公私钥对(pk_B,sk_B),输出一个重签名密钥 $rk_{A\to B}$.代理使用 $rk_{A\to B}$ 可将受托者的签名转换为委托者的签名.

4. 初始块重签名 $ReSign_{ini}$:

- 受托者生成初始块的原始签名,具体操作如下:
  (1) 受托者 A 输入流片段的初始块消息 m_0、Hash 密钥 Y、一个随机数 r_0,计算 $TH_{Y_0}=f(m_0,Y_0)Y_0+r_0P$,这里,$Y_0$ 为长久 Hash 密钥,则其相应的长久陷门密钥为 x_0;
  (2) 受托者 A 计算 $h=TH_{Y_0}(m_0,r_0)$,并运行 PRS 的签名算法 *Sign*,在公钥 pk_A 下生成对 h 的原始签名 $\sigma_A=Sign(h)$,但交换代理不知道受托者的私钥 sk_A 的任何信息;
  (3) 发送(m_0,r_0,σ_A)给代理;
- 代理接收到信息后进行认证,认证通过后生成重签名,具体操作如下:
  (1) 输入(m_0,r_0)和长期密钥 Y_0,计算 $TH_{Y_0}(m_0,r_0)=f(m_0,Y_0)Y_0+r_0P$ 并保存在缓存中;
  (2) 运行验证算法,若 $Verify(pk_A,TH_{Y_0}(m_0,r_0),\sigma_A)=1$,则签名有效;否则,输出$\perp$;
  (3) 运行 PRS 的重签名算法 *ReSign*,生成重签名 σ_B:$ReSign(\sigma_A,rk_{A\to B},h)=\sigma_B$;
  (4) 将$\langle m_0,r_0,\sigma_B\rangle$转发给接收者.

5. 后续块重签名 $ReSign_{sub}$:

- 受托者输入后续待签名消息 m_i,运行 PRS 的签名算法 *Sign*,生成消息 m_i 的签名 $\theta_{A_i}(m_i)$;
- 代理验证签名的有效性,若 $Verify(pk_A,m_i,\theta_{A_i})=1$,则消息合法;否则,输出$\perp$;
- 代理将合法消息的签名转换为委托者的签名.对于合法的消息 m_i,代理随机选择 $x_i\in_R Z_q$,计算 $Y_i=x_iP$,则(x_i,Y_i)为一次性陷门/Hash 密钥对;计算 $r_i=f(m_0,Y_0)x_0-f(m_i,Y_i)x_i+r_0$,则对消息 m_i 的重签名为 $\theta_B(m_i)=(r_i,Y_i)$,并将$\langle m_i,\theta_B(m_i)\rangle$转发给接收者.

6. 验证 *Ver*:接收方对接收到的信息进行验证.

- 初始块验证:
  (1) 计算 $TH_{Y_0}=f(m_0,Y_0)Y_0+r_0P$,并保存在缓存中;
  (2) 运行 PRS 的验证算法 *Verify*,若 $Verify(pk_B,TH_{Y_0}(m_0,r_0),\sigma_B)=1$, 签名有效;否则,输出$\perp$;
- 后续块验证:
  (1) 从缓存中恢复 $TH_{Y_0}=f(m_0,Y_0)Y_0+r_0P$;
  (2) 计算 $TH_{Y_i}=f(m_i,Y_i)Y_i+r_iP$;
  (3) 检测等式 $TH_{Y_i}(m_i,r_i)=TH_{Y_0}(m_0,r_0)$ 是否成立:如果相等则重签名 $\theta_B(m_i)=(r_i,Y_i)$有效,否则无效.

验证的有效性分析:由于 $r_i=f(m_0,Y_0)x_0-f(m_i,Y_i)x_i+r_0$,于是可以得到下式:

$$\begin{aligned} TH_{Y_i}(m_i,Y_i) &= f(m_i,Y_i)Y_i + r_iP \\ &= f(m_i,Y_i)Y_i + (f(m_0,Y_0)x_0 - f(m_i,Y_i)x_i + r_0)P \\ &= f(m_0,Y_0)x_0P + r_0P \\ &= TH_{Y_0}(m_0,Y_0). \end{aligned}$$

因为σ_B是 PRS 中委托者对 h 的有效重签名,所以σ_B也是 PRS 中委托者对 $TH_{Y_i}(m_i,r_i)$ 的合法重签名.

6 方案的安全性证明及效率分析

基于陷门 Hash 函数构造的代理重签名方案(MTH-PRS)的安全性可以归约为两个方面:(1) 所采用的代理重签名的安全性;(2) 所采用的陷门 Hash 函数的安全性,即,EDL-MTH 方案的安全性.如果攻击者可以输出 MTH-PRS 的一个伪造,则意味着攻击者可以伪造一个代理重签名或找到陷门 Hash 函数的一个碰撞.因此,只要所采用的代理重签名方案是安全的且陷门 Hash 函数是安全的,则相应的基于陷门 Hash 函数的代理重签名方案也是安全的.

6.1 攻击模型

本节攻击模型建立在文献[19,20,32]定义的代理重签名安全模型的基础之上.下面考虑一个游戏:

$$G_A^{\text{MTH-PRS}}(1^k).$$

这里,1^k是一个安全参数. $G_A^{\text{MTH-PRS}}(1^k)$ 是挑战者 C 和攻击者 A 之间的游戏.模型的具体描述如下:

1. 游戏建立

挑战者 C 以安全参数 1^k 为输入,运行 *ParGen* 算法生成公开参数 *par*,并将 *par* 发送给攻击者 A.

2. 问询阶段

攻击者 A 在多项式时间内可以适应性地向挑战者 C 相应的预言机进行以下问询:

- 密钥预言机 D_k:当攻击者 A 输入公钥 pk 向密钥预言机 D_k 问询时,密钥预言机 D_k 返回公钥 pk 相应的私钥 sk 给攻击者 A;当攻击者 A 输入 Hash 密钥 Y 向密钥预言机 D_k 问询时,密钥预言机 D_k 返回 Hash 密钥 Y 相应的陷门密钥 x 给攻击者 A;
- 重签名密钥预言机 D_{pk}:攻击者 A 输入一个受托者的公钥 pk_A 和一个委托者的公钥 pk_B 向重签名密钥预言机 D_{pk} 问询,重签名预言机 D_{pk} 返回一个重签名密钥 $rk_{A\to B}$ 给攻击者 A;
- Hash 函数预言机 O_f:攻击者 A 输入(m,Y)向 Hash 函数预言机 O_f 问询,Hash 函数预言机 O_f 返回 $f=f(m,Y)$ 给攻击者 A;
- 陷门 Hash 预言机 O_H:攻击者 A 输入(m,Y)向陷门 Hash 函数预言机 O_H 问询,Hash 函数预言机 O_H 选择 $m\in_R Z_q$,计算 $TH_Y(m,r)=f(m,Y)Y+rP$,并将 $TH_Y(m,r)$给攻击者 A;
- 原始签名预言机 O_s:攻击者 A 输入(pk,Y,m)向原始签名预言机 O_s 问询,原始签名预言机 O_s 返回一个关于 $h=TH_Y(m,r)$在公钥 pk_A 下的原始签名σ_A给攻击者 A;
- 重签名预言机 O_{rs}:当攻击者 A 输入(pk_A,pk_B,m,σ_A)向重签名预言机 O_{rs} 问询,重签名预言机 O_{rs} 返回消息 m 的一个重签名σ_B 给攻击者 A.当攻击者 A 输入$\{b_0,b_1,b_2,\ldots,b_n,m_{n+1}\}$向重签名 O_{rs} 问询(其中,$b_0=(m_0,\sigma_B)$,$b_i=(m_i,r_i,Y_i)$),重签名预言机 O_{rs} 返回消息 m_{n+1} 的一个重签名(r_{n+1},Y_{n+1})给攻击者 A.这里,$b_{n+1}=(m_{n+1},r_{n+1},Y_{n+1})$,并且 $h=TH_{Y_{n+1}}(m_{n+1},r_{n+1})$.

3. 伪造阶段

攻击者 A 的目标是输出一个或多个如下伪造:

- 类型 1

攻击者 A 能生成一个新初始消息 m_0^* 的合法签名 $\sigma_A^F(m_0^*)$ 和合法重签名 $\sigma_B^F(m_0^*)$.这里, m_0^* 从来没有被提交给 O_s 问询,并且$\langle\sigma_A^F,\sigma_B^F\rangle$能通过 MTH-PRS 对初始块的验证,即,伪造了 MTH-PRS 方案所关联的代理重签名方案的一个原始签名和重签名.

- 类型 2

攻击者 A 输出一个伪造的流 $S^F=\{b_0^F,b_1^F,b_2^F,...,b_n^F\}$ 满足如下的条件:

1) m_n^F 未向预言机 O_s 进行过问询;
2) $\{b_0^F,b_1^F,b_2^F,...,b_{n-2}^F,m_{n-1}^F\}$ 提交给 O_s 进行问询,并返回 $b_{n-1}^F=(m_{n-1}^F,r_{n-1}^F,Y_{n-1}^F)$;
3) b_0^F 在公钥 PK_A,PK_B 下分别通过了 MTH-PRS 的初始块的验证,$\{b_1^F,b_2^F,...,b_n^F\}$ 在公钥 PK_B 和 Hash 密钥 Y 下通过了 MTH-PRS 对后续块的验证.

攻击者在游戏后得到伪造 MTH-PRS 签名的优势定义为 $Adv_A^{\text{MTH-PRS}}(1^k)$, 这里,概率的大小完全取决于挑战者和攻击者之间的抛币概率:

$$\begin{aligned}Adv_A^{\text{MTH-PRS}}(1^k)=\Pr[&verify(pk_A,TH_{Y_0}(m_0,r_0),\sigma_A)=1\wedge\\&Verify(pk_B,TH_{Y_0}(m_0,r_0),\sigma_B)=1\wedge\\&verify(pk_A,m_i,\theta_A)=1\wedge\\&h=TH_{Y_0}(m_0,r_0)=YH_{Y_i}(m_i,r_i)]\geq\varepsilon.\end{aligned}$$

定义 9(不可伪造性). MTH-PRS 方案在适应性选择消息攻击下是不可伪造的,若对任意的攻击者 A 在多项式时间内赢得上述游戏的优势 $Adv_A^{\text{MTH-PRS}}(1^k)$ 是可忽略的.

6.2 安全性证明

本节给出关于基于 EDL-MTH 的代理重签名方案的详细安全分析.在第 2.3 节中我们已经详细分析了 MTH-PRS 方案所关联的 EDL-MTH 方案的安全性,定理 1 证明了 EDL-MTH 满足可计算性、碰撞性、抗碰撞性、抗密钥泄露性和语义安全性.下面在定理 1 的基础上,证明 MTH-PRS 方案在适应性选择消息攻击下是不可伪造的.

定理 2. 在随机预言模型下,假设采用的代理重签名方案 PRS 在适应性选择消息攻击下是不可伪造的,那么在群 G 上的(T,ε)-ECDL 问题假设成立的情况下,基于陷门 Hash 函数的代理重签名方案 MTH-PRS 在适应性选择消息攻击下也是不可伪造的.

证明:如果存在攻击者 A 可以在多项式时间内以不可忽略的优势 $Adv_A^{\text{MTH-PRS}}(1^k)$ 攻破上述方案,则要么存在一个攻击者可以在多项式时间内以不可忽略的优势攻破 PRS 方案;要么存在一种算法能够以不可忽略的优势 $Adv_B^{\text{ECDL}}(1^k)$ 破解(T,ε)-ECDL 问题.

设(T,ε)-ECDL 问题的一个实例$(P,aP)\in G$,B 的目标是确定 aP 的离散对数 $a\in Z_q$.

1. 系统建立 Setup

B 以安全参数 1^k 为输入,运行 *ParGen* 算法生成公开参数 *par*,并将 *par* 发送给攻击者 A.

其中,B 分别模拟 Hash 函数随机预言机 O_f 和 O_H 以及签名预言机 O_s 和 O_{rs} 来回答 A 的问询.此外,B 还需维护 3 个初始化为空的 Hash 列表 f^{List},H^{List} 和 M^{List},用于保存 A 对随机预言机和签名预言机的问询以及相应 B 的模拟响应.

2. 问询阶段

- 当 A 以(m_i,Y_i)问询 O_f 时,B 进行如下的操作:
  (1) 检查$(m_i,Y_i)\in f^{List}$,如果存在,返回 f_i 给 A,也就是 $f_i=f(m_i,Y_i)$;
  (2) 否则,选择 $f_i\in_R Z_q$,使得 $f(m_i,Y_i)=f_i$,并将 f_i 保存在 f^{List};
- 当 A 以消息 m 问询 O_s 和 O_{rs} 时,B 进行如下操作:
  (1) 检查$\langle m,\cdot\rangle\in M^{List}$,如果存在,返回$\langle\sigma_A,\sigma_B\rangle$给 A;
  (2) 否则,选择 $r\in_R Z_q$,计算 $TH_{Y_0}(m,r)=f(m,Y_0)Y_0+rP$ 和 $h=TH_{Y_0}(m,r)$, 并将 $TH_{Y_0}(m,r)$ 保存在 H^{List};
  (3) 运行 PRS 的签名算法 *Sign* 生成对 h 的原始签名$\sigma_A=Sign(h)$,运行 PRS 的重签名算法 *ReSign*,生成对 h 的重签名σ_B;

(4) 将$\langle\sigma_A,\sigma_B\rangle$保存在$M^{List}$,并将$\langle\sigma_A,\sigma_B\rangle$返回给$A$;

- 当A以$\{b_0,b_1,\ldots,b_n,m_{n+1}\}$问询$O_s$和$O_{rs}$时,$B$进行如下操作:

(1) 检查$\{b_0,\ldots\}\in M^{List}$(这里,$b_0=(m_0,r_0,\sigma_A,\sigma_B)$),若不存在$B$,则返回⊥并中止;否则,$B$恢复$M^{List}$列表的信息为$\{b_0,b_1',\ldots,b_i'\}$.如果$i\prec n$,则问询无效,$B$返回⊥并中止;否则,重排$\{b_1,\ldots,b_n\}$使得$\forall s\prec t,\overline{r}_s\leqslant\overline{r}_t$.相似地,重排$\{b_1',\ldots,b_n'\}$得到$\{\overline{b}_1',\ldots,\overline{b}_n'\}$.对于$j$=1,…,$n$,检查$\overline{b}_j'=\overline{b}_j$,如果不等,问询无效,$B$返回⊥;

(2) 如果$i\succ n$,则返回p_{n+1}'(这里,$b_{n+1}'=(m_{n+1}',r_{n+1}',Y_{n+1}',\theta_{A_{n+1}}')$);否则,$B$进行如下操作:

(a) 随机选择$f_{n+1},r_{n+1}\in_R Z_q^*$,恢复$f_0=f(m_0,Y_0)$,如果不存在,则中止;

(b) 计算$Y_{n+1}=(f_0Y_0+r_0P-r_{n+1}P)/f_{n+1}$,这里,$f_{n+1}=f(m_{n+1},Y_{n+1})$,将$\langle m_{n+1},Y_{n+1},f_{n+1}\rangle$保存在表$f^{List}$;

(c) 问询签名预言机O_s获得受托者对消息m_{n+1}的原始签名$\theta_{A_{n+1}}$;

(3) 将$b_{n+1}=(m_{n+1},r_{n+1},Y_{n+1},\theta_{A_{n+1}})$保存到表$M^{List}$与序列$\{b_0,\ldots\}$相关链的位置;

(4) B返回$b_{n+1}=(m_{n+1},r_{n+1},Y_{n+1},\theta_{A_{n+1}})$给$A$,其中,$\theta_{B_{n+1}}=\langle r_{n+1},Y_{n+1}\rangle$.

3. 伪造阶段

最终,A输出类型1或类型2的伪造.若A输出类型1的伪造$\langle\sigma_A^F,\sigma_B^F\rangle$,则意味着存在关于消息$m_0^F$的签名$\langle\sigma_A^F,\sigma_B^F\rangle$,$\langle\sigma_A^F,\sigma_B^F\rangle$是代理重签名方案PRS的伪造签名,与定理2的假设前提相矛盾.若A输出类型2的伪造$\langle p_0^F,p_1^F,\ldots,p_n^F\rangle$,则意味着下面两个条件必然有一个是成立的:

1) 假设A以m问询O_s时得到响应$\langle\sigma_A,\sigma_B\rangle$,并且假设$A$以$(b,m')$($b=(m,r,Y,\sigma_A)$)问询$O_s$得到响应$\langle m',r',Y'\rangle$,$A$可以利用$\langle m,m',r,r',Y_0,Y'\rangle$计算出$Y$的离散对数$y_0$(即实现了密钥泄露),并用$y_0$生成一个伪造的块$p_n^F$;

2) A生成$\langle r_n^F,Y_n^F\rangle$,使得$TH_{Y_0}(m_0^F,r_0^F)=TH_{Y_n^F}(m_n^F,r_n^F)$,即,伪造了一个一次性碰撞.

以上任何一种情况,根据第3节中的定理1,B都可以计算出同一个消息m的两个合法签名(m,r,Y,θ_A)和(m,r',Y',θ_A),则可以通过$a=(y_0f(m_0,Y_0)+r_0-r)/f(m,Y)\bmod q$求得$aP$关于$P$的离散对数. □

6.3 算法的效率分析

流交换过程中,如果交换效率不高,会出现流断续现象,严重时会造成整个流的传输失败.而影响流交换效率的因素主要有两方面:一是交换过程所采用算法的计算复杂度带来的延时;二是由于高实时性通常采用的是不可靠的传输,会出现包丢失现象.因此,在交换过程中需要考虑包丢失所带来的延时.

本文提出的流交换方案主要思想是对初始块信息的处理采用原始的代理重签名方案,为的是充分发挥原始代理重签名方案在流交换中的优势(如流身份可验证性、流交换的可控性、简化流交换密钥管理等).但是针对流交换的实时性高、连续性强等特点,对后续流交换中的重签名生成算法和验证算法进行改进,引入了陷门Hash函数的思想,因此,本文在进行性能分析时只对后续流信息处理时的重签名算法和验证算法的效率进行比较.在本方案中,后续流信息交换时重签名的生成只需寻找陷门Hash函数的一次碰撞,对重签名验证也只需检测两个陷门Hash函数的值是否相等即可.

下面以一个具有n个数据块的流S为例,将已有的可证安全性的代理重签名方案和本方案的重签名生成算法和验证算法的效率进行比较,其结果见表1.

Table 1 Performance comparison of re-signature algorithm and verification algorithm

表1 重签名算法和验证算法的性能比较

签名方案	重签名效率	验证效率
S_{bi}[19]	$3(n-1)T_{\exp}$	$2(n-1)T_{pair}$
S_{uni}[19]	$(n-1)(3T_{\exp}+T_m)$	$2(n-1)T_{pair}$
S_{mb}[20]	$(n-1)(2T_{\exp}+3T_m+T_a)$	$3(n-1)T_{pair}$
Kim等人的方案[21]	$(n-1)(4T_{\exp}+T_m)$	$3(n-1)T_{pair}$
Sunitha等人的方案[22]	$(n-1)(4T_{\exp}+2T_m)$	$(n-1)(2T_{\exp}+T_m)$
本方案	$(n-1)(2T_m+T_a+T_s)$	$(n-1)(T_m+T_a)$

注:$T_a,T_s,T_m,T_{\exp},T_{pair}$分别代表一次模乘、一次加法、一次减法、一次指数运算、一次配对运算所需要的时间

7 MTH-PRS 在安全流交换中的应用示例及性能分析

在本节中,我们通过一个典型的应用示例来说明 MTH-PRS 在流交换中的应用,并说明本方案在安全性方面和性能方面比传统的流交换方案有明显的提高.

7.1 应用示例

本文关注的是使用改进的代理重签名方案实现实时的、高效的、安全流交换问题,目的是为组成数字流的每一块数据提供完整性,起源认证和不可抵赖性.

这里,我们以股票交易系统为例进行说明.在股票市场中有 3 个参与方:股票所有者、券商和股票交易所.其中,一个股票交易所为多个券商提供服务,一个券商为多个股票交易所提供服务.股票交易过程中,最重要的就是实时性和完整性,因此,股票所有者发出的交易请求必须经过签名,券商收到后要求能够做出快速响应,同时把验证后的请求发送给股票交易所.股票价格信息随时都在变化,股票所有者需要不断地与股票交易所进行交互才能实时地掌握股票信息,从而做出正确的决策.因此,整个股票交易过程中通常是一个频繁交互的流.这里,我们将 MTH-PRS 方案应用到股票交易过程中,图 1 描述了基于 MTH-PRS 的安全流交换的工作原理图.股票所有者(stock owner,简称 So)代表流的发送者,也就是代理重签名中的委托者;券商(brokerage,简称 Br)代表安全流交换的控制者,负责对 So 提交的流进行验证并对通过验证的流进行重签名;股票交易所(stock exchange,简称 Se)代表流的接收者.

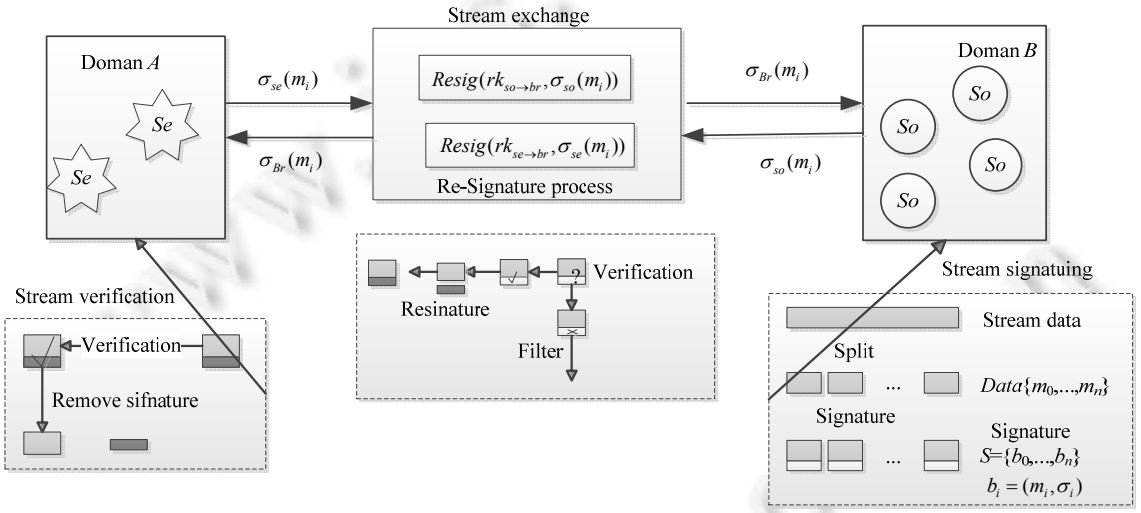

Fig.1 The working process of stream exchange

图 1 流交换工作过程图

安全流交换的具体过程如下:

(1) 我们假设这里存在一个密钥管理中心,负责产生、分发和管理密钥和重密钥;

(2) 当 *So* 要进行股票交易时,首先进行流初始化操作,生成初始块的签名$\sigma_{So}=Sign(rk_{So},m_0)$;券商收到请求后对$\sigma_{So}$ 进行验证,验证通过后生成重签名$\sigma_{Br}=ReSign(rk_{So\to Br},m_0)$,并将初始化信息和$\sigma_{Br}$ 转发给股票交换所;*Se* 对σ_{Br} 进行验证,验证通过后形成一条从股票所有者到券商、券商到股票交易所的安全流交换链;

(3) 在后续的交易过程中,*So* 生成流中每个块的签名放入块负载信息中;

(4) *Br* 收到块信息后,经验证后无需再计算重签名,只需按照 MTH-PRS 方案中的步骤 5 计算碰撞信息,将碰撞值加入信息块中转发给 *Se*.对于没有通过验证的块不再进行转发,这样不仅可以过滤非法用

户的流,又可以节约传输带宽和时间;

(5) *Se* 收到信息后,无需再对签名进行验证,只需计算陷门 Hash 值并从缓存中恢复出相应的陷门 Hash 值进行比对即可.

7.2 安全性分析

从安全的角度来看,在实际网络中,股票交易所通常位于安全性要求较高的内网,而广大股民通常都是来自互联网的用户,券商(相当于安全流交换服务器)将起到全面的控制与管理的作用:一方面,券商有效地隔离了非法股票所有者与股票交易所之间的直接交互,减少了互联网用户对股票交易所构成的安全威胁;另一方面,为了防止券商的权限过大(如,近期发生的光大证券乌龙事件),应用该方案可以有效降低券商权限,以保证股票所有者的合法权益.

在股票交易过程中,股票所有者将请求先发送给券商,得到放行确认,并同时由券商在请求中加入该券商的签名信息,再转发给相应的股票交易所.在整个过程中,与传统的签名方案相比,不仅可以验证股票所有者的签名,还具有中间券商的重签名控制,保证了交换源头身份的可信性、交换内容的完整性和交换方的不可抵赖性.而且对于券商的重签名还需要股票所有者的参与,限制了券商的权限,防止券商权限的滥用.另外,由于股票交易市场有着明确的编制与隶属关系,因此在密钥的分发和证书的撤消方面有着独特的优越性,可以最大限度地保证方案中私钥分发的安全性.此外,在第 6.2 节中已经证明:该应用采用的 MTH-PRS 方案在随机模型下是适应性选择消息攻击安全的,保证了签名的不可伪装性.以上分析表明,该流交换方案是安全的.

7.3 性能分析

从性能的角度来看,将 MTH-PRS 应用于流交换方案与传统的流交换方案相比具有以下优势:

(1) 实时性高、鲁棒性强

由于对数据流是分块签名,且每块的签名和验证(当第 1 块验证通过后)不受其他块的影响,可以独立进行,因此可以实现对流中数据块的实时签名和验证,无需等待接收整个流后再做处理,减少了延时.此外,数据块的独立性使得当出现块丢失时不会影响对收到块的验证,能够容忍包丢失,具有较好的鲁棒性.

(2) 计算资源少

对流的验证主要依赖于对初始块的处理,因此,缓存中只需保留初始块的陷门 Hash 值,而且无需提供接收整个流的空间,所需存储空间小,可以应用于计算资源有限的设备中(如手机、智能卡等).

(3) 计算开销小,交换效率高

在不考虑网络拥塞的情况下,流交换中的时延主要产生在对流的重签名和验证.从第 6.3 节表 1 的分析中可以看出:本文提出的 MTH-PRS 方案的效率比传统的代理重签名方案计算开销小、延时低、交换效率高.此外,券商还可以将来自不同股票所有者的流汇聚成一个流进行批验证,进一步提高重签名和验证的效率.

(4) 降低密钥管理复杂度

假设一个股票交易所可以为 m 个券商服务,一个券商可以为 n 个股票所有者服务,则传统的方案中股票交易所需要管理 mn 个公钥才能实现对所有股票所有者的验证,密钥管理复杂度为 $O(mn)$.采用本文方案对于股票交易所来说,只需管理 m 个公钥,复杂度为 $O(m)$;对于券商来说,只需管理 n 个公钥,复杂度为 $O(n)$.显然,$O(m)\prec O(mn)$,$O(n)\prec O(mn)$,因此,本方案可以降低密钥管理的复杂度.

(5) 流经路径可信,流交换可控

基于 MTH-PRS 实现的股票交易过程,从本质上是一种受控的流交换过程.券商可以对非法股票所有者的请求进行过滤、控制,或者说股票所有者和股票交易所之间必须经过第三方券商这条路径才能进行交换.对于流转过程复杂、流转路径要求严格的流交换应用,本方案的优势会更加突出.例如基于工作流的公文流转系统,如图 2 所示,假设公文必须从 A 经过 B,C,D,E 逐级流转到 Q,采用本文的方案可以在流转的节点上设置代理进行验证,生成下一步可验证的重签名,通过对签名的逐级验证和转换实现跟踪流交换路径和控制流交换范围的功能.

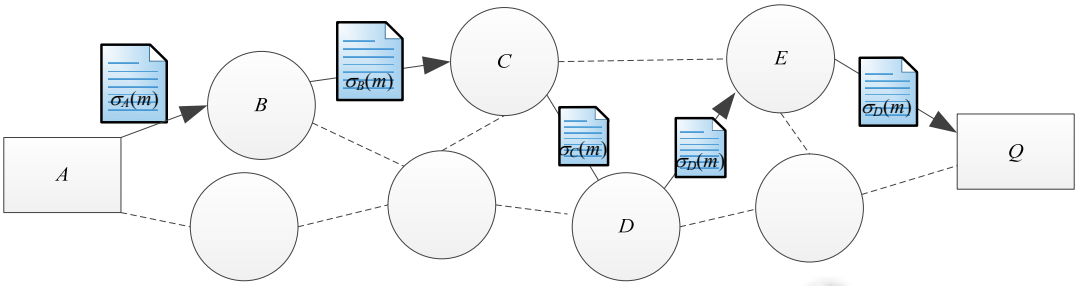

Fig.2 Document exchange path diagram based on the workflow

图 2 基于工作流的文档交换路径图

综上所述,本文提出的基于 MTH-PRS 的流交换方案极大地改善了流交换的安全性和整体性能.本方案不仅可以广泛地应用于内、外网之间或不同安全域之间受控的流交换,也可以适用于云计算、物联网、P2P 网络等大规模复杂网络环境下的应用.

8 结 论

为了解决流跨域交换的安全问题,本文提出应用代理重签名技术实现对流源头认证、流数据完整性验证和交换的不可抵赖性.针对流交换的特点,本文引入陷门 Hash 函数的概念对代理重签名方案进行改进,提出一种新的无密钥泄漏的陷门 Hash 函数方案(EDL-MTH),并基于 EDL-MTH 构造一个适用于安全流交换的代理重签名方案(MTH-PRS).在随机预言模型下,证明了 MTH-PRS 在适应性选择消息攻击下是不可伪造的,并与已有的代理重签名方案进行了性能比较.其结果表明:在流交换中,本方案重签名效率更高,具有一定的实用性.将来的工作是在本文方案的基础上设计多流交换的动态授权方案.

References:

[1] Ateniese G, Fu K, Green M, Hohenberger S. Improved proxy re-encryption schemes with applications to secure distributed storage. In: Proc. of the 12th Annual Network and Distributed System Security Symp (NDSS 2005). Berlin, Heidelberg: Springer-Verlag, 2005. 29−43. [doi: 10.1145/1127345.1127346]

[2] Taban G, Crrdenas AA, Gligor VD. Towards a secure and interoperable DRM architecture. In: Proc. of the ACM DRM 2006. Berlin, Heidelberg: Springer-Verlag, 2006. 69−78. [doi: 10.1145/1179509.1179524]

[3] Brassard G, Chaum D, Crépeau C. Minimum disclosure proofs of knowledge. Journal of Computer and System Sciences, 1988, 37(2):156−189. [doi: 10.1016/0022-0000(88)90005-0]

[4] Krawczyk H, Rabin T. Chameleon Hashing and signatures. In: Proc. of the 7th Annual Network and Distributed System Security Symp. 2000. 143−154. http://www.freepatentsonline.com/6108783.html

[5] Chaum D, van Antwerpen H. Undeniable signatures. In: Proc. of the Advances in Cryptology-Crypto'89. LNCS 435, Springer-Verlag, 1989. 212−216. [doi: 10.1007/0-387-34805-0_20]

[6] Even S, Goldreich O, Micali S. On-Line/Off-Line digital signatures. In: Proc. of the Crypto'89. LNCS 435, Santa Barbara, Berlin: Springer-Verlag, 1990. 263−277. [doi: 10.1007/BF02254791]

[7] Shamir A, Tauman Y. Improved on-line/off-line signature schemes. In: Proc. of the Crypto 2001. LNCS 2139, Santa Barbara, Berlin: Springer-Verlag, 2001. 355−367. [doi: 10.1007/3-540-44647-8_21]

[8] Ateniese G, de Medeiros B. Identity-Based chameleon Hash and applications. LNCS 3110, Springer-Verlag, 2004. 164−180. [doi: 10.1007/978-3-540-27809-2_19]

[9] Chen XF, Zhang FG, Kim K. Chameleon Hashing without key exposure. In: Zhang K, Zheng Y, eds. Proc. of the 7th Int'l Conf. on Information Security (ISC). 2004. 87−98. [doi: 10.1007/978-3-540-30144-8_8]

[10] Ateniese G, de Medeiros B. On the key exposure problem in chameleon Hashes. In: Proc. of the SCN 2004. LNCS 3352, Springer-Verlag, 2005. 165−179. [doi: 10.1007/978-3-540-30598-9_12]

[11] Gao W, Li F, Wang X. Chameleon Hash without key exposure based on Schnorr signature. In: Proc. of the Computer Standards and Interfaces. 2009. 282−285. [doi: 10.1016/j.csi.2007.12.001]

[12] Chen XF, Zhang FG, Tian H, Wei B, Kim K. Discrete logarithm based chameleon Hashing and signatures without key exposure. Computers & Electrical Engineering, 2011,37(4):614−623. [doi: 10.1016/j.compeleceng.2011.03.011]

[13] Chen XF, Zhang FG, Susilo W, Mu Y. Efficient generic on-line/off-line signatures without key exposure. In: Proc. of the ACNS 2007. LNCS 4521, Berlin: Springer-Verlag, 2007. 18−30. [doi: 10.1007/978-3-540-72738-5_2]

[14] Chen XF, Zhang FG, Tian H, Wei B, Susilo W, Mu Y, Lee H, Kim K. Efficient generic on-line/off-line(threshold) signatures without key exposure. Information Sciences, 2008,178:4192−4203. [doi: 10.1016/j.ins.2008.06.022]

[15] Harn L, Hsin WJ, Lin C. Efficient on-line/off-line signature schemes based on multiple-collision trapdoor Hash families. The Computer Journal, 2010,53(9):1478−1484. [doi: 10.1093/comjnl/bxp044]

[16] Lin DR, Wang CI, Guan DJ. Efficient vehicle ownership identification scheme based on triple-trapdoor chameleon Hash function. Journal of Network and Computer Applications, 2011,34(1):12−19. [doi: 10.1016/j.jnca.2010.07.001]

[17] Chandrasekhar S, Chakrabarti S, Singhal M. A trapdoor Hash-based mechanism for stream authentication. IEEE Trans. on Dependable and Secure Computing, 2012,9(5):699−713. [doi: 10.1109/TDSC.2012.48]

[18] Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography. In: Proc. of the Advances in Cryptology: EUROCRYPT'98, Vol.1403. Helsinki, 1998. 127−144. [doi: 10.1007/BFb0054122]

[19] Ateniese G, Hohenberger S. Proxy re-signatures: New definitions, algorithms, and applications. In: Proc. of the ACM CCS 2005. Berlin, Heidelberg: Springer-Verlag, 2005. 3l0−319. [doi: 10.1145/1102120.1102161]

[20] Shao J, Cao Z, Wang L, Liang X. Proxy re-signature schemes without random oracles. In: Srinathan K, Rangan CP, Yung M, eds. Proc. of the Progress in Cryptology—INDOCRYPT 2007. Berlin, Heidelberg: Springer-Verlag, 2007. 197−209. [doi: 10.1007/978-3-540-77026-8_15]

[21] Kim K, Yie I, Lim S. Bidirectional proxy re-signature scheme in indocrypt 2007. Int'l Journal of Network Security, 2009,9(1): 8−11.

[22] Sunitha NR, Amberker BB. Proxy re-signature schemes: Multi-Use, unidirectional translations. Journal of Advances in Information Technology, 2011,2(3):165−176.

[23] Sunitha NR, Amberker BB. Proxy re-signature scheme that translates one type of signature scheme to another type of signature scheme. In: Recent Trends in Network Security and Applications, Vol.89. Berlin: Springer-Verlag, 2010. 270−279. [doi: 10.1007/978-3-642-14478-3_28]

[24] Yang HB, Sun JL, Wang XA, Cui J. Proxy re-signature scheme from CBS to IBS. Advanced Materials Research, 2011(304): 355−358. [doi: 10.4028/www.scientific.net/AMR.304.355]

[25] Yang XD, Wang CF. Threshold proxy re-signature schemes in the standard model. Chinese Journal of Electronics, 2010,19(2): 345−350.

[26] Yang XD, Wang CF, Lan CH, Wang B. Flexible threshold proxy re-signature schemes. Chinese Journal of Electronics, 2011,20(4): 691−696.

[27] Yang PY, Cao ZF, Dong XL. Threshold proxy re-signature. Journal of Systems Science and Complexity, 2011,24(4):816−824. [doi: 10.1007/s11424-011-8370-3]

[28] Yang XD, Wang CF, Zhang YL, Wei WY. A new forward-secure threshold proxy re-signature scheme. In: Proc. of the IEEE Int'l Conf. on Network Infrastructure and Digital Content (IC-NIDC 2009). 2009. 566−569. [doi: 10.1109/ICNIDC.2009.5360842]

[29] Deng YQ. A blind proxy re-signatures scheme based on random oracle. Advanced Materials Research, 2011,(204):1062−1065. [doi: 10.4028/www.scientific.net/AMR.204-210.1062]

[30] David P, Jacques S. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 2000,13(3):361−369. [doi: 10.1007/s001450010003]

[31] Hong X, Chen KF, Wan ZM. Simplified universally composable proxy re-signature. Ruan Jian Xue Bao/Journal of Software, 2010,21(8):2079−2088 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3634.htm [doi: 10.3724/SP.J.1001.2010.03634]

[32] Shao J, Feng M, Zhu B, Cao ZF, Liu P. The security model of unidirectional proxy re-signature with private re-signature key. In: Steinfeld R, Hawkes P, eds. Proc. of the Information Security and Privacy. Berlin, Heidelberg: Springer-Verlag, 2010. 216−232. [doi: 10.1007/978-3-642-14081-5_14]

附中文参考文献:

[31] 洪璇,陈克非,万中美.简单的通用可组合代理重签名方案.软件学报,2010,21(8):2079−2088. http://www.jos.org.cn/1000-9825/3634.htm [doi: 10.3724/SP.J.1001.2010.03634]

孙奕(1979－),女,河南郑州人,博士生,讲师,主要研究领域为网络与信息安全,数据安全交换.

陈亮(1991－),男,硕士生,主要研究领域为网络安全,信息内容安全.

陈性元(1963－),男,博士,教授,博士生导师,主要研究领域为网络与信息安全.

徐建(1980－),男,硕士,主要研究领域为应用密码学.

杜学绘(1968－),女,博士,教授,博士生导师,主要研究领域为多级安全,算法分析.