·安全技术· 文章编号：1000-3428(2016)12-0139-06 文献标志码：A 中图分类号：TP309

一种基于身份的服务器辅助验证签密方案

王彩芬，康步荣

（西北师范大学 计算机科学与工程学院，兰州 730070）

摘 要：为解决已有基于身份的签密算法效率不高的问题，考虑低端设备计算能力弱的特点，引入服务器辅助验证思想，提出一种服务器辅助验证签密方案，通过服务器完成验证过程中的一些复杂运算，减少算法验证阶段的计算量和运行时间，使得签密算法可应用在低端设备上。基于判定双线性 Diffie-Hellman 问题及计算的 Diffie-Hellman 困难问题假设，在随机预言模型中，证明该方案满足不可伪造性和机密性。

关键词：签密；基于身份；服务器辅助验证；双线性对；随机预言模型；不可伪造性

中文引用格式：王彩芬，康步荣. 一种基于身份的服务器辅助验证签密方案[J]. 计算机工程，2016，42(12)：139-144.

英文引用格式：Wang Caifen，Kang Burong. An ID-based Server-aided Verification Signcryption Scheme[J]. Computer Engineering，2016，42(12)：139-144.

An ID-based Server-aided Verification Signcryption Scheme

WANG Caifen，KANG Burong

（College of Computer Science and Engineering，Northwest Normal University，Lanzhou 730070，China）

【Abstract】 In order to improve the efficiency of the existing ID-based signcryption algorithm，taking into consideration the characteristic of low computing power of low-end devices，introducing the idea of server-aided verification，this paper presents a server-aided verification signcryption scheme. Complex operations in the verification process are carried out through the server，which effectively reduces the amount of computation and running time in the validation phase of the algorithm，and makes the application of the algorithm in the low-end device possible. Based on the difficult hypothesis of Decision Bilinear Diffie-Hellman（DBDH） problem and Calculation of Diffie-Hellman（CDH） problem，it proves the scheme meets the unforgeability and confidentiality in random oracle model.

【Key words】 signcryption；ID-based；server-aided verification；bilinear pairing；random oracle model；unforgeability

DOI：10.3969/j.issn.1000-3428.2016.12.025

0 概述

文献[1]提出基于身份密码体制的概念，其思想是直接选择表示用户身份的信息（如姓名、身份证号、邮箱号等）作为公钥，无需使用数字证书，有效地减少了证书管理的存储和计算开销。文献[2]提出了首个基于身份的签密方案，但是由文献[3]可知该方案不满足安全性。为了提高签密运算的效率，已经采用了例如预计算和脱机运算等技术，尽管这些技术能够减少计算开销，但是对低功率设备而言很多签密系统的计算量仍然很大，例如，双线性对的计算开销就很大[4-5]。但是，由于双线性对良好的特性，它又往往被认为是构造基于身份的加密和签名的基础部分。因此，对基于身份的签密体制而言，降低运算量仍是一项极具挑战性的任务。

近年来，网络技术的发展日新月异，各种各样的电子设备已经广泛普及，成为了人们日常生活中的必需品。随之而来地，也出现了许多低端设备。所谓低端设备是指外观小、计算能力非常有限的电子设备。例如，智能卡、移动终端、射频识别设备、嵌入式设备、无线传感器、电子钥匙等都属于低端设备。此外，低端计算设备有限的电池容量使得其能源供应也受到约束，像无线传感器等工作于偏僻地区的设备，更换电池极为不便。所以，即使它们具有相应的计算能力，但由于耗电量很大，必须节省使用这些计算功能。而且，要对低端计算设备进行安全和隐私保护，需要耗费大量的投资成本，这样的限制使得现有密码技术难以应用。因此，针对低端设备计算

基金项目：国家自然科学基金(61163038，61262057，61562077)；甘肃省高等学校科研项目(2015B-220)。

作者简介：王彩芬(1963—)，女，教授、博士生导师，主研方向为云计算、无线传感器网络；康步荣，硕士。

收稿日期：2015-12-28 **修回日期：**2016-02-18 **E-mail：**1306414991@qq.com

能力弱、能源供应有限、安全保护成本高等弊端,提出适应于低端设备的加密方案成为必然[4]。

为了降低基于身份的签密体制中的运算开销,并提高加密方案在低端设备上的可行性效率,引入一个服务器来完成验证过程中的复杂运算。为了提高 RSA 算法的认证效率,文献[6]提出了服务器辅助验证签名的概念,其主要思想是用一个辅助服务器和验证者交互完成签名验证算法,以此减少验证者在验证过程中的计算量,从而提高算法的验证效率。文献[7]把服务器辅助验证的思想嵌入到离散对数签名方案中,用以提高离散对数密码算法的运算效率。文献[8]提出了更广义上的无假设服务器辅助验证模型和一个双线性映射基于属性的服务器辅助验证方案。文献[4]提出了一个基于身份服务器辅助验证的短消息签名算法,该算法有以下3 个优点:1)签名长度只有 160 bit;2)在验证阶段不需要双线性对运算,具有轻量级的计算量,3)有效地避免了基于身份签名方案中存在的密钥托管问题。近年来服务器辅助验证在密码学领域中仍是个广受关注的话题。文献[5]提出了基于属性的服务器辅助验证签名方案。文献[9-11]中又分别把服务器辅助验证方法与云计算、轻量级设备以及移动计算结合起来提出了一些新的方案。现有的服务器辅助验证签密方案还很少,为此,本文提出一种新签密方案,并证明该方案的安全性。

1 基础知识

本节介绍与方案有关的数学基础知识,包含基于身份的签密算法和双线性对相关性质,以及以双线性对为基础的密码学困难问题假设。

1.1 基于身份的签密算法

根据已有文献可知,基于身份的签密方案包括以下 4 个算法:setup 算法,extract 算法,signcrypt 算法和 unsigncrypt 算法。

1)setup 算法:输入参数 k,生成系统私钥 s 和系统公开参数 $Params$,其中,s 保密;$Params$ 对外公开。

2)extract 算法:输入$(ID_u,s,Params)$,执行操作 $d_u=extract(ID_u,s,Params)$,输出 ID_u 的私钥 d_u,并将 d_u 秘密传输给 ID_u。

3)signcrypt 算法:输入(d_s,ID_r,m),发送者计算 $\sigma=signcrypt(Params,d_s,ID_r,m)$,输出密文 σ。

4)unsigncrypt 算法:接收者以(d_r,ID_s,σ)作为算法的输入,执行算法恢复出明文 m 或者输出符号 $\perp$,其中输出符号 $\perp$ 表示 σ 是无效密文。

1.2 双线性对

给定 G_1 和 G_2 是 2 个群,阶都为 q,其中,G_1 是加法群;G_2 是乘法群;映射 $e:G_1\times G_1\rightarrow G_2$ 称为双线性映射。如果 e 满足如下性质:

1)双线性:$\forall P,Q\in G_1$,$\forall a,b\in Z_q^*$,$e(aP,bQ)=e(P,Q)ab$,$e(P+Q,R)=e(P,R)e(Q,R)$,$e(P,R+Q)=e(P,R)e(P,Q)$。

2)非退化性:对 $P,Q\in G_1$,存在有效的算法计算 $e(P,Q)\neq 1$。

3)可计算性:对所有的 $P,Q\in G_1$,存在有效的算法计算 $e(P,Q)$。

1.3 数学难题

计算的 Diffie-Hellman(Calculation of Diffie-Hellman,CDH)问题:对 $\forall a,b\in Z_q^*$,已知(P,aP,bP),计算 abP 目前为止还是困难的。

判定双线性 Diffie-Hellman(Decision Bilinear Diffie-Hellman,DBDH)问题:对 $\forall a,b,c\in Z_q^*$,给定五元组(P,aP,bP,cP,h)。其中,$h\in G_2$,判断 $h=e(P,P)abc$是否成立。

2 形式化定义和安全模型

本节给出新方案的形式化定义和安全模型,方案中涉及签密者、辅助服务器和验证者三方。

2.1 形式化定义

基于身份的服务器辅助验证签密方案主要由系统设置算法、用户私钥提取算法、签密算法、解签密算法、服务器设置算法和服务器验证算法(SAV-Verify)构成[4],前 4 个算法同 1.1 节所述,剩余 2 个算法做如下定义:

1)服务器设置算法:以系统参数 $Params$ 作为输入,生成比特串 $VString$,该串中包含了验证者要进行预计算的信息,若没有预计算则 $VString=(P)$。

2)SAV-Verify 算法:由于签密验证者的计算能力有限,不能独立完成验证和解签密操作,验证者输入$(m,\sigma,ID,VString)$,在服务器的协助下运行该算法完成对消息的验证。

2.2 安全模型定义

参考文献[3-4]对基于身份的服务器辅助验证签密方案的安全模型做了如下定义:

定义 1(保密性)令 A 表示一个攻击者(Attacker),用符号 Π 表示基于身份的服务器辅助验证签密算法,攻击者 A 与挑战者 C 完成游戏 1 定义的交互过程。

游戏 1

1)C 输入参数 k,执行 setup 算法,获得 s 和 $Params$,秘密保存 s,并将 $Params$ 输出给 A。

2)在这一阶段,A 可做多次预言操作,具体过程如下:

(1)extract 询问:攻击者 A 随机选取一个身份信息 ID_u 发送给挑战者,C 以 ID_u 作为询问的输入,输出其对应的私钥 d_u,即 $d_u=extract(ID_u)$,然后将 d_u

返回给 A。

(2)signcrypt 询问:攻击者 A 任意生成 2 个身份信息 ID_A,ID_B 和一条明文消息 m,发送给 C,由 C 计算 ID_A 发送给 ID_B 的消息 m 对应的密文 σ,即 $\sigma = signcrypt(m, d_A, ID_B)$,并将 σ 作为输出返还给 A。

(3)SAV-Verify 询问:在用户公钥 Q_{ID} 下自适应地询问(m,σ),挑战者 C 和攻击者 A 交互完成服务器辅助验证过程,询问执行结束,C 将输出发送给攻击者 A。

3)A 选择 2 个等长的消息 m_0,m_1,以及 2 个挑战身份 ID_i,ID_j,要求 ID_i,ID_j 没有做过 extract 操作,然后 C 公平地掷一枚硬币 $\theta \in \{0,1\}$,计算 $\sigma^* = signcrypt(m_\theta, d_A, ID_B)$,并将 σ^* 作为输出返还给 A。

4)猜测过程中,A 可以做阶段 1)所述的操作多项式有界次,但要求他不对身份 ID_i,ID_j 做 extract 操作,并且不允许他对 σ^* 进行 SAV-Verify 操作。

5)A 输出一个比特 θ'作为对 θ 的猜测。如果 $\theta' = \theta$,则攻击者 A 获胜。

以上讨论的 A 被称为 IND-IBSS-CCA2-Attacker,用 $Adv_{\Pi}^{\text{IND-IBSS-CCA2}}(A) = |\Pr[\theta' = \theta] - 1/2|$ 来表示 A 在游戏 1 中取胜的概率。假如对任意一个 IND-IBSS-CCA2 攻击者 A,它的猜测优势在概率时间 t 内都小于 ε,则说方案 Π 是关于(t,ε) - IND-IBSS-CCA2 安全的。

定义 2(不可伪造性)　令 F 表示一个伪造者(Forger),符号 Π 表示一个基于身份的服务器辅助验证签密方案,伪造者 F 和挑战者 C 完成游戏 2 定义的交互过程。

游戏 2

1)C 输入系数 k,执行 setup 算法,获得 s 和 $Params$,秘密保存 s,并将 $Params$ 输出给 F。

2)伪造者 F 可做多项式次数的游戏 1 中的询问,最后,F 输出一个新的项(σ', ID_A, ID_B),即这个项既不能是通过运行 signcrypt 询问所获得的,也不能是通过运行 extract 询问所获得的。如果 SAV-Verify 算法输出一个有效的消息 m,则 F 取胜。

以上讨论的 F 被称为 EUF-IBSC-CMF-Forger,用来表示 F 在游戏 2 中取胜的概率 $Adv_{\Pi}^{\text{EUF-BISC-CMF}}(F) = |\Pr[\theta' = \theta] - 1/2|$。假如对任意一个 EUF-IBSC-CMF 伪造者 F,它的猜测优势在概率时间 t 内都小于 ε,则方案 Π 是关于(t,ε) - EUF-IBSC-CMF 安全的。

3　具体实现方案

根据 2.1 节中形式化定义的基于身份的服务器辅助验证签密算法,给出了该方案的具体实现过程和计算步骤,以下是对每个算法的详细说明。

1)系统建立算法:给定参数 k,定义 G_1 是由 P 生成阶为 q 的循环加群,G_2 是循环乘法群。KGC 选择双线性映射 $e: G_1 \times G_1 \to G_2$。KGC 设置 $s \in Z_q^*$ 做主密钥,$P_{\text{pub}} = sP \in G_1$ 为系统公钥。(E,D) 是安全的对称加密算法,选取 3 个单向哈希函数 $H_1: \{0,1\}^* \to G_1$, $H_2: G_2 \to \{0,1\}^n$, $H_3: \{0,1\}^n \times G_2 \to Z_q^*$,其中,$n$ 是消息 m 的长度。公开系统参数为 $Params = (G_1, G_2, e, P, P_{\text{pub}}, E, D, H_1, H_2, H_3)$。

2)密钥提取算法:用户向 KGC 提交身份 $ID \in \{0,1\}^*$,KGC 计算 $Q_{ID} = H_1(ID) \in G_1$,$S_{ID} = s^{-1} Q_{ID}$,$D_{ID} = sQ_{ID}$,将(S_{ID}, D_{ID})秘密传送给 ID。

3)签密方案:Alice 要向 Bob 发送消息 $m \in \{0,1\}^n$ 时,按照以下过程输出密文 σ:

(1)计算:$Q_{ID_B} = H_1(ID_B)$。

(2)随机选取 $x \in_R Z_q^*$,计算 $X_A = xQ_{ID_A}$,$\omega = e(X_A, Q_{ID_B})$,$k_1 = e(X_A, P)$,$k_2 = H_2(\omega)$。

(3)密文 $c = E_{k_2}(m)$。

(4)签名 $r = H_3(c, k_1)$,$S = (x - r) S_{ID_A}$。

(5)Alice 要签发的密文为 $\sigma = (c, r, S)$。

4)解密及签名验证算法:Bob 输入密文 σ,算法进行以下步骤:

(1)计算 $R'_A = rQ_{ID_A}$,$k'_1 = e(S, P_{\text{pub}})\, e(R'_A, P)$,$k'_2 = H_2[e(S, D_{ID_B})\, e(R'_A, D_{ID_B})]$。

(2)恢复消息 $m = D_{k'_2}(c)$。

(3)计算 $r' = H_3(c, k'_1)$,如果 $r' = r$,就认为 m 有效,反之,就输出 $\perp$。

5)服务器设置算法:已知系统参数 $Params$,则 $VString = (P)$。

6)服务器辅助验证算法:Bob 和 Aided-server 执行如下操作:

(1)Bob 计算 $R'_A = rQ_{ID_A}$,$R'_B = rQ_{ID_B}$,将初始化参数 $VString = (\sigma, R'_A, R'_B)$ 发送给 Aided-server。

(2)Aided-server 计算 $k'_{11} = e(S, P_{\text{pub}})$,$k'_{12} = e(R'_A, P)$,$k'_{21} = e(Q_{ID_A}, R'_B)$,$k'_1 = k'_{11} k'_{12}$,将$(k'_1, k'_{21})$发送给 Bob。

(3)Bob 计算 $k'_{22} = e(S, D_{ID_B})$,$\omega' = k'_{21} k'_{22}$,$k'_2 = H_2(\omega')$。

(4)恢复消息 $m = D_{k'_2}(c)$。

(5)Bob 计算 $r' = H_3(c, k'_1)$,判断 $r' = r$ 是否成立。若成立,就认为明文 m 有效并接收;反之,输出 $\perp$。

4　安全性和效率分析

本节给出新方案的正确性证明,并对其机密性和不可伪造性进行分析。

4.1 正确性分析

定理 1 SAV-Verify 算法的验证过程是正确的。

证明：

$$\begin{aligned}k_1' &= k_{11}'k_{12}' = e(S,P_{\text{pub}})e(R_A',P)\\ &= e(sS,P)e(R_A',P)\\ &= e(sS+R_A',P)\\ &= e(s(x-r)s^{-1}Q_{ID_A}+rQ_{ID_A},P)\\ &= e(xQ_{ID_A},P)=e(X_A,P)=k_1\end{aligned}$$

$$\begin{aligned}k_2' &= H_2(\omega')=H_2[e(Q_{ID_A},R_B')e(S,D_{ID_B})]\\ &= H_2[e(Q_{ID_A},rQ_{ID_B})e(S,sQ_{ID_B})]\\ &= H_2[e(rQ_{ID_A},Q_{ID_B})e(sS,Q_{ID_B})]\\ &= H_2[e(rQ_{ID_A}+s(x-r)s^{-1}Q_{ID_A},Q_{ID_B})]\\ &= H_2[e(xQ_{ID_A},Q_{ID_B})]=H_2[e(X_A,Q_{ID_B})]\\ &= H_2(\omega)=k_2\end{aligned}$$

即 $k_1'=k_1,k_2'=k_2$。

4.2 不可伪造性

用 $q_i(i=1,2,3)$ 表示预言机 H_i 的最大询问次数，用 q_s 表示 signcrypt 的最大询问次数，用 q_{SA} 表示 SAV-Verify 的最大询问次数。

定理 2 设有一 EUF-IBSC-CMF 伪造者 F 能在时间 t 内，以概率 ε 在游戏 2 中取胜，那么就有挑战者 C 能在时间 $t'<(q_1+(q_2+q_3))t$ 内，以概率 $\varepsilon'>\left(\varepsilon\dfrac{1}{q_2C_{q_1}^1}\right)$ 解决 CDH 难题。

证明：随机选择一个 CDH 困难问题实例 (P,aP,bP) 来挑战 C，假设存在 F 可以在不可忽略的优势 ε 下成功攻破模型 EUF-IBSC-CMF。C 利用 F 来解决给出的 CDH 困难问题实例，即计算出 abP 的值。C 和 F 进行预言机 H_1,H_2,H_3 来模拟系统，具体步骤如下：

阶段 1 C 将系统参数发送给 F，其中，$P_{\text{pub}}=b^{-1}P$。对 H_1,H_2,H_3，*signcrypt*，SVA-Verify 的预言结果分别存储在表 L_1,L_2,L_3L_S,L_{SV} 中。以下是具体询问过程：

1）H_1 询问。C 从 $\{1,2,\cdots,q_l\}$ 中选取一个随机数 i。当 F 做第 $u=i$ 次 H_1 操作时，C 输出 $Q_{ID_i}=H(ID_i)=aP$ 给 F，并在表 L_1 中添加记录 $(ID_i,Q_{ID_i},\perp)$。这样，与身份 ID_i 相应的私钥对分别为 $(S_{ID_i},D_{ID_i})=(bQ_{ID_i},b^{-1}Q_{ID_i})=(baP,b^{-1}aP)$。当 F 对 H_1 做第 $u\neq i$ 次操作时，C 从 Z_q^* 中选取一个随机数 bu，算出值 $Q_u=buP$，输出 $Q_u=H(ID_u)=buP$ 给 F，并在表 L_1 中添加元组 (ID_u,Q_u,bu)。

2）F 像定义 1 中描述的游戏 1 那样询问 H_2,H_3，extract 算法。

3）signcrypt 询问：F 任意输出 m 及 ID_A,ID_B 进行 signcrypt 询问。假设身份 ID_A,ID_B 执行此询问之前已对其做了 H_1 询问，则对于 ID_A,ID_B 和 ID_i,ID_j 的关系可做如下 2 种讨论：

（1）如果 $ID_A\neq ID_i$ 且 $ID_A\neq ID_j$

C 查询表 L_1，找出 ID_B 对应元组 (ID_B,Q_B,b_B) 中的 Q_B，并对 ID_A 执行 extract 询问产生对应的 S_{ID_A}，执行操作 $\sigma=signcrypt(m,S_{ID_A},Q_B)$，最后将 σ 返回给 F。

（2）如果 $ID_A=ID_i(ID_B=ID_i)$

C 查询 L_1 表，找出与身份，ID_B 相应的公钥 Q_A，Q_B，并用身份 ID_B 去做 extract 询问，获得其私钥对 (S_B,D_B)，再从 Z_q^* 中选择 2 个随机数 x,S，计算 $X_A=xQ_A,k_1=e(X_A,P),\omega=e(X_A,Q_B),k_2=H_2(\omega)$。挑战者 C 先去 L_3 列表检索是否已存在元组 (k_1,m,r) 的记录，如果有，C 重新选取随机数 x，直到在表 L_3 的记录中未出现过 (k_1,m)，随后算出 $r=H_3(m,k_1)$ 和 $\sigma=E_{k_2}(m\|r\|S)$ 的值，并将 σ 返还给伪造 F 者，F 无法判断 σ 是否有效。

4）unsigncrypt 询问：F 收到 σ 时，可能对 σ 进行 unsigncrypt 操作，C 就回答 σ 无效。

当 C 收到对于身份 ID_A 但不是关于 ID_i 的密文 σ 时，C 运行服务器辅助验证算法，计算出 ω,k_2，然后做运算 $m\|r\|S=D_{k_2}(\sigma)$，取出其中的前 n 个字节作为恢复出的明文消息。算出 $k_1=e(S,P_{\text{pub}})e(R_A,P)$ 的值，检查表 L_3 中是否存在元组 (k_1,m,r) 的记录，若有，就输出明文 m。反之，就输出符号 $\perp$。

如上操作可做多项式有界次，之后 F 要从做过预言询问的身份信息中选出挑战身份，那么能选中 ID_i 的概率大于 $\dfrac{1}{C_q^1}$。如果 F 不选取 ID_i，那么意味着 C 将失败。

阶段 2 伪造

F 以一个不可忽略的优势 ε 输出身份 ID^* 对消息 m 的有效签密 $\sigma^*=(c^*,r^*,S^*)$。如果满足以下限制条件，那么伪造者 F 获胜，并退出游戏。

1）(m^*,ID^*) 未被进行签密询问；

2）$ID^*=ID_i$；

3）伪造者 F 确信 (m^*,σ^*) 是有效的。

因为 (m^*,σ^*) 对 SAV-Verify 是有效的，有如下关系：

$$\begin{aligned}S^* &= (x^*-r^*)S_{ID^*}=(x^*-r^*)bQ_{ID^*}\\ &= (x^*-r^*)baP\end{aligned}$$

所以可获得如下关系：$abP=\dfrac{S^*}{x^*-r^*}$。这意味着 F 能以概率 ε 解决 CDH 问题。

4.3 保密性

用 $q_i(i=1,2,3)$ 表示预言机 H_i 的最大询问次数，用 q_s 表示签密算法 signcrypt 的最大询问次数，用 q_{SA} 表示服务器辅助验证算法 SAV-Verify 的最大

询问次数。

定理 3　假设存在一个 IND-IBSS-CCA2 攻击者 A 能在时间 t 内,以概率 ε 在游戏 1 中取胜,则就有一挑战者 C 能在时间 $t' < (q_1 + (q_2 + q_3)t_e)$ 内,以 $\varepsilon' > \left(\varepsilon \frac{1}{q_2 C_{q_1}^1}\right)$ 的概率解决 DBDH 问题。其中,t_e 是一次双线性对运算所需的时间。

证明:选择一个 DBDH 困难问题实例 $(P, P_1, P_2, P_3, P_4) = (P, aP, bP, cP, Z)$ 来挑战 C,设有一攻击者 A 可以以概率 ε 攻破 IND-IBSS-CCA2 模型。C 利用 A 来解决 DBDH 困难问题实例,即要判断等式 $Z = e(P,P)^{abc}$ 是否成立。C 攻击者 A 通过交互询问预言机 H_1, H_2, H_3 来模拟系统,具体步骤如下:

C 将 $Params = (G_1, G_2, e, P, P_{pub}, E, D, H_1, H_2, H_3)$ 发送给 A。H_1, H_2, H_3, *signcrypt*, SAV-Verify 的预言结果分别存储在表 $L_1, L_2, L_3 L_S, L_{SV}$ 中。以下是具体询问过程:

1) H_1 询问:C 从 $\{1,2,\cdots,q_1\}$ 中选取 2 个随机数 i, j。当 A 做第 $u = i$ 次 H_1 操作时,C 输出 $Q_{ID_i} = H(ID_i) = aP$ 给 A,并在表 L_1 中添加元组 $(ID_i, Q_{ID_i}, \perp)$。当 A 做第 $u = j$ 次 H_1 操作时,C 输出 $Q_{ID_j} = H(ID_j) = bP$ 给 A,并在表 L_1 中添加元组 $(ID_j, Q_{ID_j}, \perp)$。此时,与身份 ID_i, ID_j 相应的私钥对分别为 $(S_{ID_i}, D_{ID_i}) = (s^{-1}aP, saP)$, $(S_{ID_j}, D_{ID_j}) = (s^{-1}bP, sbP)$。当 A 做第 $u \neq i, j$ 次 H_1 操作时,C 从 Z_q^* 中任意选取 bu,计算 $Q_u = buP$,输出 $Q_u = H(ID_u) = buP$,并在表 L_1 中添加元组 (ID_u, Q_u, bu)。

2) H_2 询问:C 首先检查表 L_2 中是否存在元组 (ω, k_2, X_{ID_u}) 的记录,如果有,C 将 k_2 返回给 A,对于 $ID_u = ID_i$ 或 $ID_u = ID_j$ 的询问,输出 $k_2 = H_2(\omega)$ 作为回答,并在表 L_2 中添加元组 (ω, k_2, cQ_{ID_i}) 或 (ω, k_2, cQ_{ID_j})。若没有,那么 C 从 Z_q^* 中选取一个随机数 k_2 返回给 A,并在表 L_2 中添加元组 (ω, k_2, X_{ID_u})。

3) H_3 询问:C 首先检索表 L_3 中是否已存在元组 (k_1, m, r) 的记录,若已记录,则 C 将 r 作为返回值发送给 A。若没有,C 从有限域 Z_q^* 中选取一个随机数,将其赋值给 r,然后把 r 返回给攻击者 A,并在 L_3 表中添加元组 (k_1, m, r)。

4) extract 询问:假设 ID_u 执行 *extract*(ID_u) 操作前已作了 H_1 操作,如果 $ID_u = ID_i$ 或 $ID_u = ID_j$,C 将失败。否则,C 在表 L_1 中检索出 ID_u 的对应元组 (ID_u, Q_u, b_u),计算 ID_u 的私钥 $du = buP$ 作为对 A 的回答。

5) signcrypt 询问:A 选择任意的明文 m 和用户身份 ID_A, ID_B,进行 signcrypt 询问。假设在对 ID_A, ID_B 执行操作前已经对其执行过 H_1 操作,则身份 ID_A, ID_B 和 ID_i, ID_j 的关系可做如下 3 种讨论:

(1)如果 $ID_A \neq ID_i$ 且 $ID_A \neq ID_j$

C 查询表 L_1,找出 ID_B 对应记录 (ID_B, Q_B, b_B) 中的 Q_B,并对 ID_A 执行 extract 操作得到 d_A,计算 $\sigma = signcrypt(m, d_A, Q_B)$,最后将 σ 输出给 A。

(2)如果 $ID_A = ID_i$(或 ID_j),但 $ID_B \neq ID_i$ 且 $ID_B \neq ID_j$

C 查询表 L_1,找出 ID_A, ID_B 对应的公钥 Q_A, Q_B,并对 ID_B 执行 extract 询问,得其私钥对 (S_B, D_B),然后从 Z_q^* 中选择 2 个随机数 x', S',算出 $X_A' = x'Q_A$, $k_1' = e(X_A', P)$, $\omega = e(X_A', Q_B)$, $k_2 = H_2(\omega)$ 的值。挑战者 C 检查 L_3 表中是否已有元组 (k_1', m, r) 的记录,如果已有记录,那么 C 选取新的随机数 x',直到在 L_3 表中未出现过记录 (k_1', m),然后计算 $r' = H_3(m, k_1')$ 和 $\sigma' = E_{k_2}(m \| r' \| S')$ 的值,并将 σ' 发送给 A,A 认为 σ' 是有效的。

(3)如果 $ID_A = ID_i$(或 ID_j)且 $ID_B = ID_j$(或 ID_i)

C 查询表 L_1,检索出与身份 ID_A, ID_B 对应的公钥 Q_A, Q_B,从 Z_q^* 中选择 2 个随机数 x^*, S^*,计算 $X_A^* = x^* Q_A$, $k_1^* = e(X_A^*, P)$, $\omega = e(X_A^*, Q_B)$, $k_2^* = H_2(\omega)$。检查列表 L_3 中是否已有元组 (k_1^*, m, r) 的记录,若已有记录,C 选取新的 x',直到表 L_3 中未出现过 (k_1^*, m),然后计算 $r^* = H_3(m, k_1^*)$,最后做运算 $\sigma^* = E_{k_2^*}(m \| r^* \| S^*)$,获得相应密文 σ^* 作为对 A 的回答,因为 A 不能对 σ^* 执行 unsigncrypt 询问,所以 A 无法判断 σ^* 的有效性。

6) unsigncrypt 询问:攻击者 A 获得关于 ID_i 和 ID_j 的密文 σ^* 时可能要求 C 对其做解密运算,这时 C 就告诉 A 该密文无效。

当 C 收到关于 ID_A 和 ID_B 但不是关于 ID_i 和 ID_j 的签密 σ 时,C 运行服务器辅助验证算法,计算出 ω', k_2',然后做 $m \| r \| S = D_{k_2'}(\sigma)$ 运算,取出其中的前 n 个字节作为明文输出。算出 $k_1' = e(S, P_{pub}) e(R_A', P)$ 的值,检查表 L_3 中是否存在元组 (k_1, m, r) 的记录,若已记录,则输出消息 m。反之,就输出符号 $\perp$。

做如上预言过程多次之后,A 输出一对挑战身份,假设 A 要从做过预言询问的身份中选择出挑战身份,则选中身份 ID_i 和 ID_j 的优势大于 $\frac{1}{C_{q_1}^2}$。如果 A 不选取 ID_i 和 ID_j,那意味着 C 失败。

A 任意选取 2 个等长的消息 m_0, m_1 以及 2 个身份 ID_i 和 ID_j(ID_i, ID_j 不能是在 2)中执行过 extract 询问的身份)。C 公平地掷一枚硬币 $\theta \in \{0,1\}$,C 查询表 L_1,得到与身份 ID_i, ID_j 对应的公钥 Q_i, Q_j,再从 Z_q^* 中选择 2 个随机数 x^*, S^*,计算 $X^* = x^* Q_i, k_1^* = e(X_i^*, P)$。$C$ 检查 L_3 表中是否存在元组

(k_1^*, m_θ, r)的记录,若已记录,则重新选取 x^*,直到 L_3 列表没有元组(k_1^*, m_θ)的记录为止,然后计算 $r^* = H_3(m_\theta, k_1^*)$,选取 $\omega^* = Z$,算出 $k_2^* = H_2(\omega^*)$,最后,C 计算密文 $\sigma^* = E_{k_2^*}(m_\theta \| r^* \| S^*)$发送给 A。

在猜测过程中,A 可多次执行如上预言,但不能对 σ^* 做 unsigncrypt 操作。

最后,A 输出一个比特 θ',如果 $\theta' = \theta$,则表示 C 成功回答上述 DBDH 问题实例,即有 $e(X_{ID_j}, Q_{ID_i}) = e(cbP, aP) = e(X_{ID_i}, Q_{ID_j}) = e(caP, bP) = e(P, P)^{abc}$;否则,回答 0。

4.4 效率分析

表 1 是文献[4,12-15]和本文方案的验证效率对比。其中,用 Pa 表示对运算;用 Xor 代表异或运算;Mul 表示标量乘运算;Hash 表示验证算法中的哈希运算。N 为 G_1 中元素的长度;M 为明文消息 m 的长度;ID 为身份信息 ID 的长度;n,l 为固定长度;Z 为有限域 Z_q^* 中元素的个数。

表 1 验证效率分析

方案	Pa	Xor	Mul	Hash	密文长度
文献[4]方案	1	0	2	3	$3N+Z+M$
文献[12]方案	1	0	1	3	$N+n+l$
文献[13]方案	4	0	2	2	$N+Z+M$
文献[14]方案	4	1	1	3	$2N+M+ID+Z$
文献[15]方案	1	1	$N+3$	3	$(n+2)N+ID+M$
本文方案	1	0	1	2	$N+Z+M$

从表 1 可知,在本文提出的方案中,签密验证者只做 1 次对运算,1 个乘法运算和 2 个散列函数,相较于文献[13]方案,本文方案的验证过程减少了 1 次乘法运算,3 个对运算;相较于同样使用了服务器辅助验证的文献[6]方案,本文方案减少了 1 次乘法运算和 1 次Hash 运算,并大大缩短了密文长度。所以,新的基于身份的服务器辅助验证方案减少了计算开销和算法的运行时间,并且节省了密文传输过程中的带宽,在计算成本和通信量成本方面综合起来具有一定的优势,有效并显著地提高了签密算法的验证效率。

5 结束语

本文构造一个新的高效的基于身份服务器的辅助验证签密方案,基于服务器辅助验证算法的理论,用一个高效的服务器对签名验证过程中的复杂运算进行预处理,从而有效地提高签名验证算法的效率,还分析了方案的安全性和运行效率。分析结果表明,该方案基于判定性线性 Diffie-Hellman 问题和计算性 Diffie-Hellman 问题满足不可伪造性和消息保密性。在接下来的研究工作中,考虑将基于属性的密码体制和匿名认证思想引入到服务器辅助验证算法中。

参考文献

[1] Shamir A. Identity-based Cryptosystems and Signature Schemes[C]//Proceedings of CRYPTO'84. Berlin, Germany:Springer,1984:47-53.

[2] Malone L J. Identity Based Signcryption[EB/OL]. (2015-10-21). http://www.signcryption.org/publications/pdffiles/MaloneLee-eprint2002-098.pdf.

[3] Ferraiolo D F, Sandhu R, Gavrila S, et al. Proposed NIST Standard for Role-based Access Control[J]. ACM Transactions on Information and System Security,2001,4(3):224-274.

[4] Zhang Jianhong, Sun Zhibin. An ID-based Server-aided Verification Short Signature Scheme Avoid Key Escrow[J]. Journal of Information Science and Engineering,2013,29(3):549-473.

[5] Wang Zhiwei, Xie Ruirui, Wang Shaohui. Attribute-based Server-aided Verification Signature[J]. Applied Mathematics & Information Sciences,2014,8(6):3183-3190.

[6] Quisquater J J, de Soete M. Speeding up Smart Card RSA Computations with Insecure Coprocessors[C]//Proceedings of the 2nd International Smart Card Conference. Amsterdam, the Netherlands:North-holland, 2000:191-197.

[7] Lim C H, Lee P J. Security and Performance of Server-aided RSA Computation Protocols[C]//Proceedings of Crypto'95. New York, USA:ACM Press,1995:70-83.

[8] Girault M, Lefranc D. Server-aided Verification:Theory and Practice[C]//Proceedings of Crypto'05. Berlin, Germany:Springer,2005:605-623.

[9] Liu Zhusong, Yan Hongyang, Li Zhike. Server-aided Anonymous Attribute-based Authentication in Cloud Computing[J]. Future Generation Computer Systems, 2015,52:61-66.

[10] Xu Lingling, Li Jin, Tang Shaohua, et al. Server-aided Verification Signature with Privacy for Mobile Computing[C]//Proceedings of the 4th International Conference on Emerging Intelligent Data and Web Technologies. Washington D. C., USA: IEEE Press, 2015:1-11.

[11] Guo Fuchun, Mu Yi, Susilo W, et al. Server-aided Signature Verification for Lightweight Devices[J]. Computer Journal,2014,57(4):481-493.

[12] 王彩芬,王筱娟,郝占军. 基于身份的新签密方案[J]. 计算机应用研究,2010,27(12):4630-4637.

[13] 王大星,滕济凯. 一种高效的基于身份的签密方案[J]. 微电子学与计算机,2010,27(7):57-59.

[14] 李发根,胡予濮,李 刚. 一个高效的基于身份的签密方案[J]. 计算机学报,2006,29(9):1641-1647.

[15] 庞辽军,高 璐,裴庆祺,等. 基于身份公平的匿名多接收者签密方案[J]. 通信学报,2013,34(8):161-168.

编辑 顾逸斐