基于网格的 Ad Hoc 网络签密算法及应用

陈少华 樊晓光 丛 伟 黄金科 孙贤明

(空军工程大学航空航天工程学院 西安 710038)

摘 要 基于身份的签密算法具有计算开销小的优点,适用于 Ad Hoc 网络的密钥管理,能有效确保消息的机密性和认证性。针对现有基于身份的签密算法的不足,以低负载、可扩展性强、高连通性的网格为逻辑结构,提出了一种高效的基于网格的签密算法,并将其应用于 Ad Hoc 网络的密钥管理中,降低了系统密钥管理的通信代价和计算代价。经分析表明,在随机预言机模型下该签密算法是安全的,相应的密钥管理方案比传统方案更安全高效,网络的抗攻击能力更强。

关键词 Ad Hoc 网络,网格,签密,密钥管理,随机预言模型

中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.06.028

Grid-based Identity Signcryption Algorithm and Application in Ad Hoc Network

CHEN Shao-hua FAN Xiao-guang CONG Wei HUANG Jin-ke SUN Xian-ming

(College of Aeronautics and Astronautics Engineering, Air Force Engineering University, Xi'an 710038, China)

Abstract Identity-based signcryption has the advantage of low computation cost, which is suitable for the key management of Ad Hoc network and can guarantee the confidentiality and authentication of information. Aiming at the deficiencies of the existing identity-based signcryption algorithm, taking a grid as the logical structure, which has the advantages of low overhand, high expandability and high connectivity, this paper proposed an efficient grid-based identity signcryption algorithm. By using this algorithm in the key management scheme of Ad Hoc network, the scheme reduces the communication and computation cost of key management. The analysis show that the signcryption algorithm is secure in the random oracle model, the key management is more safe and efficient, and the Ad Hoc network has good ability of resistance to the attack.

Keywords Ad Hoc network, Grid, Signcryption, Key management, Random oracle model

Ad Hoc 网络是由一组带有无线收发装置的移动节点或自主终端组成的自治系统,这种网络不需事先布置基站,因此在军事通信、移动商务、抢险救灾等场合得到广泛应用。然而,由于 Ad Hoc 网络具有无线传播、无中心控制、通信能量受限和拓扑结构动态变化等固有特性,使其安全问题更加突出[1],传统网络的安全措施不能解决 Ad Hoc 网络的问题。

机密性和认证性是信息安全传输的两个基本要求,密码学的传统解决方法是"先签名后加密"的分步式方法,其计算和通信代价为签名和加密的总和,效率较低。郑玉良[2]提出的签密技术能够在一个逻辑步内完成数字签名和公钥加密。签密能同时高效地满足机密性和认证性的安全需求[3-4]。在 Ad Hoc 网络的密钥管理和安全路由中签密算法已经有很多应用,其中文献[5-8]研究了基于身份的签密算法,该算法有效地提高了 Ad Hoc 网络的安全性并且具有很高的效率。Zhang 等人[9]提出了目前计算效率最高的基于身份的签密算法 S-IDSC,但是其验证部分只能在特定条件下成立。密钥管理作为构建 Ad Hoc 安全网络系统的核心问题之一,对提高网络的安全性起着举足轻重的作用[10-11]。文献[12]提出的基于身份的分层分布式密钥管理方案更适用于 Ad Hoc 网络,但其在密钥更新时存在缺陷。基于上述分析,本文在张宇等人[13]提出的基于身份的签密算法的基础上进行分析和改进,提出了基于网格的签密算法,并将其运用于 Ad Hoc 网络的密钥管理中。通过分析验证可以得到,其计算和传输代价较小,更加适合 Ad Hoc 网络的特性,能实现高效的 Ad Hoc 网络密钥管理需求。

1 研究基础

1.1 双线性对及 Diffie-Hellman 问题

定义 1(双线性对) 设 G 是一个阶为 q 的循环群,P 是 G 的生成元。V 是另一个阶为 q 的循环乘法群。假设在 G 和 V 中,

到稿日期:2016-05-08 返修日期:2016-07-12 本文受国家自然科学基金(61402517),中航工业集团预研基金资助项目(619010601)资助。

陈少华(1992-),男,硕士,主要研究方向为移动自组织网络、信息安全,E-mail:876915107@qq.com;**樊晓光**(1965-),男,博士,教授,主要研究方向为移动自组织网络、机载计算机网络;**丛 伟** 女,博士,副教授,主要研究方向为操作系统、综合航电;**黄金科** 男,博士,主要研究方向为 Ad Hoc 网络;**孙贤明** 男,硕士,主要研究方向为综合航电故障诊断。

离散对数问题是难解的。G 和 V 之间的双线性映射对为 $e:G*G\rightarrow V$，其满足以下条件。

(1)双线性性：任意 $p1,p2\in Z_q^*$，$Q\in G$，$e(p1+p2,Q)=e(p1,Q)e(p2,Q)$，且 $e(Q,p1+p2)=e(p1,Q)e(p2,Q)$；

(2)非退化性：存在 $P,Q\in G$，使得 $e(P,Q)\neq 1$；

(3)可计算性：存在有效算法，对于 $P,Q\in G$，可计算 $e(P,Q)$。

双线性映射 e 可以通过有限域超椭圆曲线上的 Weil 对进行构造[14]。

定义 2　设 G 是一个 q 阶加法群，P 是生成元，计算 Diffie-Hellman 问题(Computational Diffie-Hellman Problem，CDHP)是指，对于任意给定的 aP,bP，计算 abP，其中 a,b 是未知的。

定义 3　设 q 阶循环加法群为 $(G_1,+)$ 和乘法群为 $(V,\cdot)$，$e:G*G\rightarrow V$ 是一个双线性映射，P 是 G 的生成元，判定双线性 Diffie-Hellman 问题(Decisional Bilinear Diffie-Hellman Problem，DBDH)是指，对于任意给定 (aP,bP,cP,l)，判定等式 $l=e(P,P)^{abc}$ 是否成立，其中 a,b,c 是未知的，$l\in V$。

定义 4　在 $(G_1,+)$ 上，给出 $P,aP,bP,cP\in G$，计算 abP 是困难的，但是判定 $l=e(P,P)^{abc}$ 是容易的，称这种群为 GDH(Gap Diffie-Hellman)群。

本文的密钥管理方案是基于 GDH 群提出的。

1.2　基于身份的签密算法基础

Malone-Lee 在文献[15]中首次提出了基于 ID 的签密算法思想，在文献[16]中对算法进行了完善并给出了其形式化定义，包括以下几个部分。

(1)系统建立(setup)：输入安全参数 1^k，由私钥生成中心(Private Key Generator，PKG)生成数据对 $(params,s)$，其中 $params$ 为公开参数，s 为主密钥。

(2)密钥提取(extract)：输入身份 ID_U 和主密钥 s，由 PKG 生成公私钥对 (d_U,Q_U)。

(3)签密(signcryption)：签密包括签名和加密两个部分，用户 A 使用其私钥 d_A 对消息 m 进行签名得到 σ，并产生随机数 r，进一步利用 (d_A,ID_B,m,σ,r) 生成密文 c。

(4)解签密(unsigncrytion)：解签密包括解密和验证两部分。输入 (c,d_B)，解密得到 (m,ID_A,σ)，再进行验证，若输出 $\perp$(Invalid)表示 c 为无效密文，否则输出 m。

1.3　基于身份签密算法的安全性定义

定义 5(机密性)　如果不存在任何多项式有界的攻击者 Ω 以不可忽略的优势赢得如下游戏，则这种签密算法是适应性选择身份和密文攻击下密文不可区分的(IND-IBSC-CCA2)。

1)系统建立：挑战者 C 输入安全参数 1^k，运行建立程序(Setup)，给攻击者 Ω 返回系统公开参数 $params$，将 s 作为密钥保留。

2)第一阶段询问：攻击者对挑战者进行如下询问。

Extract 询问：攻击者 Ω 给挑战者 C 提交一个任意 ID，Ω 计算私钥 $d=Extract(ID)$，并将结果发送给 Ω。

Signcryption 询问：攻击者 Ω 给挑战者 C 提交一个发送者 ID、接收者 ID 和明文 m，C 使用发送者进行签名，然后用接收者的公钥进行加密，并将结果发送给 Ω。

Unsigncryption 询问：攻击者 Ω 给挑战者 C 提交一个接收者 ID 和密文 c，C 使用接收者的私钥进行解密，然后验证消息/签名对。若成立，则返回明文 m，否则返回 $\perp$。

3)攻击者 Ω 输出一个身份对 $\{ID_A,ID_B\}$ 和明文对 $\{m_0,m_1\}$，ID_A 和 ID_B 不能是 Signcryption 询问中出现的 ID。

4)挑战阶段：挑战者 C 随机地选择 $b\in\{0,1\}$，利用 ID_A 的私钥 $d_A=Extract(ID_A)$ 对消息 m_b 进行签名，用 ID_B 的公钥加密产生密文 c，C 发送 c 给 Ω。

5)第二阶段询问：攻击者 Ω 继续使用第一阶段的询问对挑战者进行询问。但是不能对 ID_B 进行 Extract 询问，不能对 c 进行 Signcryption 询问。

最后 Ω 返回 b'，如果 $b=b'$，则 Ω 赢得游戏，其优势为 $Adv[\Omega]=|\Pr[b'=b]-\frac{1}{2}|$。

定义 6(不可伪造性)　如果不存在多项式有界的攻击者 Ω 以不可忽略的优势赢得如下游戏，则这种签密算法是适应性选择身份和消息攻击下存在性不可伪造的(EUF-ID-CMA)。

1)系统建立：挑战者 C 输入安全参数 1^k，运行建立程序(Setup)，给攻击者 Ω 返回系统公开参数 $params$，将 s 作为密钥保留。

2)询问阶段：攻击者按照机密性中的规则对挑战者进行多项式有界次询问。

3)最后，攻击者 Ω 返回 ID_B 以及密文 c，(m,ID_A,σ) 是用 B 的私钥解密出来的三元组，如果 $ID_A\neq ID_B$，Ω 赢得游戏，其优势为 $Adv[\Omega]=\Pr[\Omega\ wins]$。

2　基于网格的签密算法描述

基于网格的签密算法(Grid-based Identity Signcryption Algorithm，G-IDSC)以基于身份的签密算法为基础，引入网格布局，通过网格顶点来分配移动节点的 ID，算法的具体结构如下。

2.1　系统参数

输入安全参数 1^k，PKG 在椭圆曲线上定义两个阶为 q 的循环群 $(G,+)$ 和 $(V,*)$，P 为 G 的生成元。记 $e:G\times G\rightarrow V$ 为 Weil 对变形得到的双线性变换。PKG 选择 3 个杂凑函数 $H_1:\{0,1\}^{l_I}\rightarrow G$，$H_2:V\rightarrow\{0,1\}^n$，$H_3:G\times\{0,1\}^n\rightarrow Z_q^*$ 和加解密算法 (E,D)，其中 l_I,n 分别表示 ID 的比特长度和明文比特长度。PKG 随机选择一个主密钥 $s_0'\in Z_q^*$，$s_0=s_0'+d$，计算得到相应的公钥 $P_0=s_0P$。公开系统参数 $(G,V,e,P,P_0,E,D,H_1,H_2,H_3)$。

2.2　密钥提取

PKG 依据网格的逻辑结构给当前移动自组织网络的节点分配 ID，若网络中有 N 个节点，构造带有 $3l$ 个多项式集合 $\{f_{(L,i)}(x,y,z),f_{(W,i)}(x,y,z),f_{(H,i)}(x,y,z)\}(i=0,1,2,\cdots,$

l)的三维网格,其中 $l=\lceil\sqrt[3]{N}\rceil$。每个节点对应一个空闲的网格顶点作为其 $ID_i(i=0,1,2,\cdots,N)$。PKG 计算相应的公钥 $Q_i=H_1(ID_i)$和私钥 $d_i=s_0Q_i$。

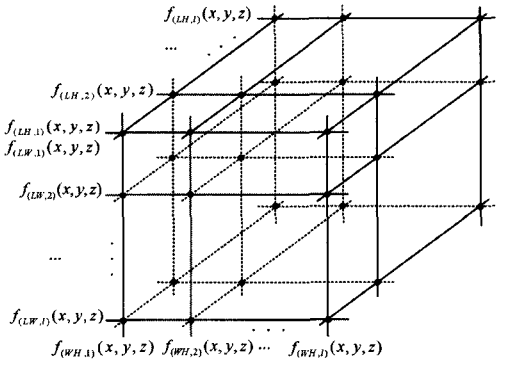

图 1 $l\times l\times l$ 三维网格

2.3 发送者签密

假设 A 要将消息 m 发送给 B,A 将执行如下签密操作:

(1)计算 $Q_B=H_1(ID_B)$;

(2)选择随机数 $r\in Z_q^*$,计算 $k_1=rQ_B$,$w=e(d_A,Q_B)$,$k_2=H_2(w)$;

(3)计算 $h=H_3(k_1,m)$,$S=\frac{d_A}{h+r}$,$c=E_{k_2}(S\parallel m)\in Z_q^*$;

(4)A 发送密文 $\sigma=(c,k_1)$给 B,B 收到密文后进行解签密操作。

2.4 接收者解签密

B 接收密文后执行如下操作:

(1)计算 $Q_A=H_1(ID_A)$,$w=e(d_B,Q_A)$;

(2)计算 $k_2=H_2(w)$;

(3)计算 $S\parallel m=D_{k_2}(c)$;

(4)计算 $h=H_3(k_1,m)$,验证 $e(S,k_1+hQ_B)=w$ 是否成立,成立则接受 m,否则输出⊥。

在上述算法中:

$H_2(e(d_A,Q_B))=H_2(e(s_0Q_A,Q_B))=H_2(e(s_0Q_B,Q_A))=H_2(e(d_B,Q_A))$

$e(S,k_1+hQ_B)=e(\frac{d_A}{r+h},rQ_B+hQ_B)=e(d_A,Q_B)=w$

所以其正确性得以保证。

3 应用 G-IDSC 的 Ad Hoc 密钥管理方案

在部署 Ad Hoc 网络初期需要使用 PKG 来组织网络,选择计算能力强、运行稳定、通行良好的节点作为服务节点,其他节点作为普通用户节点。在系统建立完成之后为了避免单点失效问题的出现,放弃 PKG,原 PKG 的功能由网络中的节点分布式完成。每个节点维护一张状态表,其结构如表 1 所列。

表 1 节点状态图

身份	是否为服务节点	邻居服务节点数	邻居用户节点数	计时器
ID	*N/Y*	*count*1	*count*2	*time*

3.1 初始化设置

PKG 按 G-IDSC 中的方法为每个节点分配 ID,记为 ID_i,并得到公私钥对(Q_i,d_i),在 Z_q^* 中选取阶为 q 的秘密值 r 以及主密钥 s_0',计算 $s=s_0r=(s_0'+d)r=s'+d'$,按照 shamir(n,t)门限密码共享方案,PKG 在 Z_q^* 中随机选取 $t-1$ 次多项式:$f(x)=(s'+a_1x+a_2x+\cdots+a_{k-1}x^{t-1})\bmod p$,$s'$为节点的共享密钥;$v(x)=(d'+b_1x+b_2x^2+\cdots+b_{t-1}x^{t-1})\bmod p$,$d'$为服务节点的共享密钥。

PKG 取得各个节点的公钥 Q_i,计算 $w_i=e(Q_i,rP_0)$,$k_i=H_2(w_i)$,若节点为用户节点则计算 $y_i=f(k_i)\bmod p$;若节点为服务节点则计算 $y_i'=y_i+(v(k_i)\bmod p)=y_i+v_i$,然后得到 $P_i=s_0y_i$,$P_i'=s_0y'$。PKG 任意选取密钥种子函数 $g(x)$,将 d_i,w_i,$g(x)$使用 G-IDSC 进行签密后为各个节点分配密钥份额,用户节点获得密钥份额 $f(k_i)$,服务节点获得密钥份额 $f(k_i)+v(k_i)$。

3.2 新节点的加入

当外部节点 N_{new} 申请加入自组织网络系统时,首先向邻居节点广播加入申请消息,邻居节点收到申请消息后由其中的 t 个服务节点共同为新节点生成密钥份额。

若与 N_{new} 相邻的节点有 m 个,这些节点收到加入申请后为新节点分配一个未被使用的网格顶点作为其 ID。当 $m\geqslant t$ 时,直接使用 shamir 秘密共享机制生成并共享密钥份额,当 $m<t$ 时,邻居服务节点向相邻的服务节点转发 N_{new} 的加入申请,直到有 t 个服务节点收到申请消息。

t 个服务节点为 N_{new} 分配新的共享密钥份额,每个节点计算:

$$d_{inew}=y_ir^{-1}Q_{new},1\leqslant i\leqslant t$$

$$y_{inew}=y_ic_i(k_{new}),1\leqslant i\leqslant t$$

其中 $c_i(x)=\prod_{j=1,j\neq i}^{t}\frac{x-x_j}{x_i-x_j}$。

N_{new} 根据 Lagrange 差值计算:$d_{new}=\sum_{i=1}^{t}y_ic_i(0)r^{-1}Q_{new}$,$y_{new}=\sum_{i=1}^{t}y_ic_i(k_{new})$,分别验证 $e(d_{inew},P)=e(r^{-1}Q_{new},P_i)(1\leqslant i\leqslant t)$和 $e(d_{new},P)=e(Q_i,P_0)$来确保 N_{new} 收到的私钥份额及其私钥是否有效。

3.3 节点的退出

当自组织网络中的节点将要退出时,向所有成员广播退出申请消息。邻居节点更新其状态表,如果退出的是服务节点,则 *count*1←*count*1−1;如果退出的是用户节点,则*count*2←*count*2−1。节点退出的原因较多,如所处的环境过于复杂、自身资源耗尽、节点移动到通信范围之外或者受到外部的恶意攻击等,如果节点因受到攻击而退出当前 Ad Hoc 网络系统,那么退出节点 N_{quit} 所携带的当前系统的密钥份额可能会对网络造成两种威胁:1)恶意节点获取的当前密钥份额一旦超过门限值,将会求出系统密钥,对网络中传输的信息进行窃听或者恶意篡改,使得网络处于不安全状态;2)退出节点和外部节点进行联合攻击,在新认证方面造成威胁。因此,在节点退出之后,由其邻居节点发起系统密钥更新请求。

3.4 系统密钥更新

若当前网络系统有节点退出或者计时器 *time* 到达更新

时间，为了防止退出节点的联合攻击以及确保服务节点的安全，需要对服务节点系统密钥份额进行更新。基于 G-IDSC 的 Ad Hoc 网络服务节点系统密钥份额更新步骤如下。

(1)系统参数使用公开的$(G,V,e,P,P_0,E,D,H1,H2,H3)$。

(2)更新发起：退出节点 N_{quit} 的邻居节点广播更新请求消息，或者 *time* 计时结束时由网络中具有最大通信量的节点来广播请求消息。收到该请求消息的邻居服务节点需要对请求节点 N_{req} 的身份进行验证，并且随机生成一个一元多项式 $f_i(x)$和 $v_i(x)$，满足 $f_i(0)=0,v_i(0)=0$。然后邻居节点为其生成对应的系统密钥份额影子，并对该影子进行 signcryption 运算，将得到的密钥份额的密文 σ 通过公开信道发送到 N_{req}。

(3)更新生成：请求节点对接收到的 σ 进行 unsigncryption 操作，获得系统密钥份额影子，并对其进行正确性验证。N_{req} 收到至少 t 个系统密钥的份额影子后，将新的份额添加到旧的份额上，使用 Lagrange 差值法生成系统密钥的份额，完成服务节点系统密钥的更新，即：$y'_{req}=y_{req}+\sum_{i=0}^{t}y_{(i,req)}$，$v'_{req}=v_{req}+\sum_{i=0}^{t}v_{(i,req)}$。

4　方案分析

4.1　签密算法安全性分析

定理 1　对于 G-IDSC，如果存在一个敌手 Ω 能够以不可忽略的概率 ξ 赢得 IND-IBSC-CCA2 游戏[17]（至多进行 q_i 次 $H_i(i=1,2,3)$询问，q_e 次 extract 询问，q_s 次 signcryption 询问和 q_u 次 unsigncryption 询问），则存在一个算法 C 能够在$t'<t+(q_s+2q_u)t_e$ 时，以 $\xi'>\frac{4\xi(q_1-q_e)(q_1-q_e-1)}{q_1^2(q_1-1)^2q_2}$解决 BDH 问题。

证明：假设$(P,P_1,P_2,P_3)=(P,aP,bP,cP)$是 C 随机接收到的 BDH 问题的实例，其目标是计算出 $e(P,P)^{abc}$。Ω 为 C 的子程序，为了避免碰撞，C 使用 3 个列表 L_1,L_2,L_3 来存储 $H_i(i=1,2,3)$的询问值。

参数生成：C 选择 $P_{pub}=cP$，其中 $c\in Z_q^*$ 相当于主密钥。C 将参数发送给 Ω。

H_1 询问：C 首先从$\{1,2,\cdots,q_1\}$中随机选择两个数 α 和 β，对 Ω 的第 α 次询问，回答 $H_1(ID_\alpha)=aP$，对 Ω 的第 β 次询问，回答 $H_1(ID_\beta)=bP$，将(ID_α,a)和(ID_β,b)存储在列表 L_1 中。对于第 $l\neq\alpha,\beta$ 次询问，计算 $Q_l=b_lP$，$d_l=b_lP_{pub}$，将(ID_l,b_l)存储在 L_1 中，回答 Q_l。

H_2 询问：输入参数 w，检查 L_2 中是否已经存在(w,k)，存在则回答 k，否则 C 从 Z_q^* 中选取随机数 k，将(w,k)存入 L_2 中并回答 k。

H_3 询问：输入参数(k_1,m)，检查 L_3 中是否存在(k_1,m,h_3)，若存在则回答 h_3，否则 C 从 Z_q^* 中选择新的 h_3，将(k_1,m,h_3)存入 L_3 中并回答 h_3。

extract 询问：输入参数 ID_i，若 $ID_i=ID_\alpha$ 或 $ID_i=ID_\beta$，C 将失败，否则 C 检查 L_1，回答 $d_i=dP_{pub}$。

signcryption 询问：Ω 提交(ID_i,ID_j,m)，若 $i\neq\alpha,i\neq\beta$，C 使用 extract 询问计算 i 的私钥并用签密算法生成签密值 σ。否则，C 任选 $r,h_3\in Z_q^*$，计算 $S=aP/h_3+r$，$k_1=rb_lcP-h_3b_lP$，将从未出现过的(k_1,m,h_3)存到 L_3 中，进一步计算 $w=e(Q_i,d_j)$，$k_2=H_2(w)$，$c=E_{k_2}(S\|m)$，发送密文 σ 给 Ω。

unsigncryption 询问：输入参数(σ,ID_i,ID_j)，若 σ 关于 ID_i 和 ID_j，但不是 ID_α,ID_β 时，Ω 收到 unsigncryption 询问，计算 $w=e(d_i,Q_j)$，若 $w\notin L_2$，则回答 $\perp$，否则检查 L_2，获得 $k_2=H_2(w)$，计算 $S\|m=D_{k_2}(c)$，若$(k_1,m)\notin L_3$，则回答 $\perp$，否则检查 L_3，获得 $h_3=H_3(k_1,m)$并返回 m。

下面计算 C 成功的概率。

如果 Ω 在询问阶段对 ID_α 和 ID_β 进行了 extract 询问，C 将失败。在 q_1 次密钥 extract 询问中，选择(ID_i,ID_j)作为挑战身份的概率大于$\frac{1}{C_{q_1}^2}=\frac{2}{q_1(q_1-1)}$，$C$ 在询问阶段退出游戏的概率为$\frac{C_{q_1-2}^{q_e}}{C_{q_1}^{q_e}}=\frac{(q_1-q_e)(q_1-q_e-1)}{q_1(q_1-1)}$。

设 A 为 Ω 在模拟过程中对 BDH 正确回答的询问次数，由文献[18]可知 $\Pr[A]\geqslant 2\xi$。Ω 随机从 L_2 中选中 A 所在条目的概率为$\frac{1}{q_2}$。

综上，C 成功的概率为：

$$\Pr[C]>2\xi\frac{2}{q_1(q_1-1)}\frac{(q_1-q_e)(q_1-q_e-1)}{q_1(q_1-1)}\frac{1}{q_2}=\frac{4\xi(q_1-q_e)(q_1-q_e-1)}{q_1^2(q_1-1)^2q_2}$$

在每次 signcryption 询问中，至多需要 1 次双线性对运算，在每次 unsigncryption 询问中，至多需要 2 次双线性对运算，所以算法 C 的计算时间为 $t'<t+(q_s+2q_u)t_e$。

定理 2　对于 G-IDSC，如果存在一个攻击者 Ω 能够以不可忽略的概率 ξ 赢得 EUF-IBSC-CMA 游戏[17]（至多进行 q_i 次 $H_i(i=1,2,3)$询问，q_e 次 extract 询问，q_s 次 signcryption 询问和 q_u 次 unsigncryption 询问），则存在一个算法 C 能够在 $t'<t+(q_s+2q_u)t_e$ 时，以 $\xi'>\frac{2\xi(q_1-q_e)(q_1-q_e-1)}{q_1^2(q_1-1)^2q_2}$解决 BDH 问题。

证明：假设$(P,P_1,P_2,P_3)=(P,aP,bP,cP)$是 C 随机接收到的 BDH 问题的实例，其目标是计算出 $e(P,P)^{abc}$。Ω 为攻击者，C 为挑战者。

参数生成：与定理 1 中方法类似；

询问阶段：与定理 1 中方法类似；

伪造：攻击者 Ω 输出(σ,ID_i,ID_j)，如果 ID_α,ID_β 与 ID_i，ID_j 中的任何一个都不相同，挑战者 C 将退出游戏，否则，检查 L_2 并从中取出(w,k)，返回 w。

C 在询问阶段退出游戏的概率为 $\frac{C_{q_1-2}^{q_e}}{C_{q_1}^{q_e}}=\frac{(q_1-q_e)(q_1-q_e-1)}{q_1(q_1-1)}$，在伪造阶段，选择$(ID_i,ID_j)$作为挑战身份的概率大于$\frac{1}{C_{q_1}^2}=\frac{2}{q_1(q_1-1)}$。

假设 A 为 BDH 问题的正确回答，若 ID_α,ID_β 与 ID_i，

ID_j 是相同的，并且 unsigncryption 询问的输出有效，则生成时所使用的密钥就是 $k_2=H_2(A)$，在 L_2 中随机选出 A 的概率为 $\frac{1}{q_2}$。

综上，C 成功的概率为：$\xi' > \frac{2\xi(q_1-q_e)(q_1-q_e-1)}{q_1^2(q_1-1)^2q_2}$，由定理 1 可知，算法 C 的计算时间为 $t'<t+(q_s+2q_u)t_e$。

4.2 密钥管理方案分析

4.2.1 方案安全性分析

(1)避免了节点的联合攻击

网络中的每个节点都维护一张状态表，当有节点退出时，邻居节点及时更新状态表中的邻居节点数目，并发起系统密钥的更新，或者当计时器 *time* 计时到达一定的周期值时，有服务节点发起密钥更新，这样有效地避免了退出节点的联合攻击，敌手完成攻击必须满足的条件也更加苛刻：1)敌手的计算时间必须保证在 *time* 周期内；2)敌手至少要搜集 t 个退出的节点，并且这些节点是在一个周期内退出的。

(2)保证了系统密钥和节点私钥的安全性

本方案将节点分为服务节点和普通用户节点，安全性基于门限秘密共享方案，分别对待服务节点密钥份额 $f(k_i)+v(k_i)$ 和用户节点密钥份额 $f(k_i)$，敌手无法从捕获的服务节点获知用户节点的信息，反之亦然。共享密钥 $s=(s_0'+d)r$ 的计算方法确保即使共享密钥被破解，主密钥依然安全，各个节点私钥不会受到威胁。

4.2.2 方案正确性分析

系统进行密钥更新时，需要 t 个服务节点参与生成新的系统密钥 s，用 y_r 代替上文中的 y_{req}，有：

$$y_r'=y_r+\sum_{i=0}^{t}y_{ir},\ v_r'=v_r+\sum_{i=0}^{t}v_{ir}$$

因为服务节点获得的密钥份额为 $f(k_i)+v(k_i)$，所以 $y_r''=y_r'+v_r'$。

$$\begin{aligned}
f'(x)&=\sum_{r=1}^{t}y_r''c_r(x)\\
&=\sum_{r=1}^{t}(y_r'+v_r')c_r(x)\\
&=\sum_{r=1}^{t}(y_r+\sum_{i=0}^{t}y_{ir}+v_r+\sum_{i=0}^{t}v_{ir})c_r(x)\\
&=\sum_{r=1}^{t}(y_r+\sum_{i=0}^{t}y_{ir}+v_r+\sum_{i=0}^{t}v_{ir})\prod_{\substack{m=1\\m\neq r}}^{t}\frac{x-x_m}{x_i-x_m}\\
f'(0)&=\sum_{r=1}^{t}(y_r+\sum_{i=0}^{t}y_{ir}+v_r+\sum_{i=0}^{t}v_{ir})\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}\\
&=\sum_{r=1}^{t}(y_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}+\sum_{i=0}^{t}y_{ir}\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m})+\\
&\quad\sum_{r=1}^{t}(v_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}+\sum_{i=0}^{t}v_{ir}\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m})\\
&=\sum_{r=1}^{t}(y_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}+f_i(0))+\\
&\quad\sum_{r=1}^{t}(v_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}+v_i(0))\\
&=\sum_{r=1}^{t}(y_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}+0+v_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}+0)\\
&=\sum_{r=1}^{t}y_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}+\sum_{r=1}^{t}v_r\prod_{\substack{m=1\\m\neq r}}^{t}\frac{-x_m}{x_i-x_m}\\
&=f(0)+v(0)\\
&=s'+d'\\
&=s
\end{aligned}$$

4.2.3 方案性能分析

主要从运算复杂度和通信复杂度两个方面进行分析。本文提出的方案的运算主要体现在签密和解签密过程中，签密过程中需要 1 次双线性对预运算和 2 次标量乘运算；解签密过程中两种运算都各进行 1 次。使用 E,M,P 分别表示指数运算、标量乘运算和双线性对运算，$x(+y)$ 表示 x 次双线性对运算、y 次双线性对预运算，本文方案与已有签密算法的运算复杂度的对比结果如表 2 所列，可以看出本文方案在运算复杂度方面具有明显优势。

表 2 本文方案与现有方案的运算复杂度对比

方案	签密			解签密		
	P	E	M	P	E	M
文献[15]	0(+1)	0	3	3(+1)	1	0
文献[16]	0(+1)	0	4	2(+2)	2	0
文献[17]	0(+2)	2	2	2(+2)	2	0
本文方案	0(+1)	0	2	1(+1)	0	1

在通信复杂度方面，本文方案进行系统密钥更新时只需要 1 次广播和 $2(t+1)$ 次单播。在文献[11]中当邻居节点的数量不少于 t 时，网络系统处于最优状态，此时系统密钥更新 1 次广播和 $4t$ 次单播。当邻居节点只有 1 个时，网络系统处于最差状态，此时系统密钥更新 1 次广播和 $2t(t+1)$ 次单播，所以本文方案具有更低的通信代价。

结束语 签密算法作为保证消息机密性和验证性的重要机制，在 Ad Hoc 网络密钥管理中发挥着重要作用。本文在基于身份的签密算法的基础上，引入网格的逻辑布局，为每个节点按照网格顶点分配 *ID*，然后再进行签密和解签密。通过随机预言机模型证明了该算法的安全性，使用该算法的密钥管理方案也具有较低的运算代价和通信代价，更加适合 Ad Hoc 网络的安全需求。

参 考 文 献

[1] RAMANATHAN R, REDI J. A brief overview of mobile Ad Hoc networks: Challenges and Directions[J]. IEEE Communications Magazine 50th Anniversary Commemoratives Issue, 2002, 40(5): 20-26.

[2] ZHENG Y. Digital signcryption or how to achieve cost (signature & encryption)≪cost (signature)+cost (encryption)[C]// Advances in Cryptology, LNCS. Berlin: Springer, 1997: 165-179.

[3] LI J, ZHAO J, ZHANG Y. Certificateless online/offline signcryption scheme[J]. Security and Communication Networks, 2014, 8(11): 1979-1990.

[4] LI Z H, FAN K, LI H. Efficient multiple signcryption scheme based on two hard problems[J]. Journal of Beijing University of Posts and Telecommunications, 2013, 36(6): 23-26.

[5] LI F G, HU Y P, ZHANG C R. An identity-based signcryption scheme for multi-domain ad hoc networks[C]// ACNS 2007, LNCS 4521. 2007: 373-384.

[6] KIM H, SONG J, YOON H. A practical approach of ID-based

crypto system in ad hoc networks[C]// Wireless Communications and Mobile Computing. 2007:909-917.

[7] DENG H, AGRAWAL D P. TIDS: threshold and identity-based security scheme for wireless ad hoc networks[J]. Ad Hoc Networks, 2004, 2(3): 291-307.

[8] LI J F, WEI D W, KOU H Z. Identity-based and threshold key management in mobile ad hoc networks[C]// International Conference on Wireless Communications, Networking and Mobile Computing 2006WiCOM 2006). 2006: 1-4.

[9] ZHANG C R, ZHANG Y Q, LI F G, et al. New signcryption algorithm for secure communication of ad hoc networks[J]. Journal on Communications, 2010, 31(3): 19-24. (in Chinese)
张申绒，张玉清，李发根，等. 适于 ad hoc 网络安全通信的新签密算法[J]. 通信学报，2010，31(3)：19-24.

[10] ZHOU L D, HASS Z J. Securing ad hoc networks[J]. IEEE Network, Special Issue on Network Security, 1999, 13(6): 24-30.

[11] LIU Z Y, MAO S L. A new secure group key management scheme for ad hoc networks[J]. Control & Automation, 2006, 22(12): 3-4. (in Chinese)
刘知远，毛胜利. 一个新的 ad hoc 安全组密钥管理方案[J]. 微计算机信息，2006，22(12)：3-4.

[12] ZHANG Q Y, MIAO F M, YUAN Z T, et al. Identity-base group key management scheme in ad-hoc[J]. Journal on Communication, 2009, 30(10A): 85-92. (in Chinese)
张秋余，苗丰满，袁占亭，等. 基于身份的 Ad Hoc 组密钥管理方案[J]. 通信学报，2009，30(10A)：85-92.

[13] ZHANG Y, DU R Y, CHEN J, et al. Analysis and improvement of an identity-based signcryption[J]. Journal on Communications, 2015, 36(11): 174-179. (in Chinese)
张宇，杜瑞颖，陈晶，等. 对一个基于身份签密方案的分析与改进[J]. 通信学报，2015，36(11)：174-179.

[14] BONEH D, FRANKIN M. Identity based encryption from Weil pairing [C] // Kilian JCRYPTO2001. Berin: SpringerVerlag, 2001: 213-229.

[15] MALONE-LEE J. Identity based signcryption[EB/OL]. http://eprint. iacr. org/2002/098.

[16] CHEN L, MALONE-LEE J. Improved identity-based signcryption[M]// Public Key Cryptography-PKC 2005. Springer Berlin Heidelberg, 2005: 362-379.

[17] LIBERT B, QUISQUATER J. A new identity based signcryption scheme from pairings [C] // IEEE Information Theory Workshop. 2003: 155-158.

[18] BONEH D, FRANKLIN M. Identity-based encryption from the Weil pairing[J]. SIAM Journal on Computing, 2003, 32(3): 586-615.

(上接第 167 页)

[5] AAFER Y, DU W, YIN H. DroidAPIMiner Mining API-Level Features for Robust Malware Detection in Android[M]// Security and Privacy in Communication Networks. Springer International Publishing, 2013: 86-103.

[6] ARP D M, SPREITZENBARTH M, HUBNER M. Drebin: Effective and explainable detection of android malware in your pocket [C]// Network and Distributed System Security Symposium, NDSS 2014. San Diego, USA.

[7] KWONGYAN L, YIN H. DroidScope: Seamlessly Reconstructing the OS and Dalvik Semantic Views for Dynamic Android Malware Analysis [C]// Proceedings of the 21st USENIX Conference on Security Symposium. 2012: 29.

[8] RASTOGI V, CHEN Y, ENCK W. AppsPlayground: Automatic Security Analysis of Smartphone Applications [C]// Conference on Data and Application Security and Privacy. ACM, 2013: 209-220.

[9] BLASING T, BATYUK L, SCHMIDT A. An Android Application Sandbox System for Suspicious software Detection [C]// 5th International Conference on Malicious and Unwanted Software. 2010.

[10] Android NDK [EB/OL]. https://developer. android. com/tools/sdk/ndk/index. html.

[11] 盘点 2015 年度 10 大安卓手机系统级病毒[EB/OL]. (2016-2-19). http://bobao. 360. cn/learning/detail/2750. html.

[12] Androguard [EB/OL]. https://github. com/androguard/androguard.

[13] ZHANG M, DUAN Y, YIN H, et al. Semantics-aware android malware classification using weighted contextual api dependency graphs[C]// Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014: 1105-1116.

[14] AU K W Y, ZHOU Y F, HUANG Z, et al. Pscout: analyzing the android permission specification[C]// Proceedings of the 2012 ACM Conference on Computer and Communications Security. ACM, 2012: 217-228.

[15] Weka [EB/OL]. http://www. cs. waikato. ac. nz/ml/weka.

[16] Appchina [EB/OL]. http://www. appchina. com.

[17] Anzhi [EB/OL]. http://www. anzhi. com.

[18] Virus share [EB/OL]. http://www. virusshare. com.

[19] SIEFERS J, TAN G, MORRISETT G. Robusta: Taming the native beast of the JVM[C]// Proceedings of the 17th ACM Conference on Computer and Communications Security. ACM, 2010: 201-211.

[20] SUN M, TAN G. Nativeguard: Protecting android applications from third-party native libraries[C]// Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks. ACM, 2014: 165-176.

[21] VITOR A, ANOTONIO B, YANICK F, et al. Going Native: Using a Large-Scale Analysis of Android Apps to Create a Practical Native-Code Sandboxing Policy[C]// Symposium on Network and Distributed System Security(NDSS 2016). Diego CA, USA.