新的无证书混合签密 *

金春花，李学俊，魏鹏娟，王立川
（西安电子科技大学 计算机网络与信息安全教育部重点实验室，西安 710071）

摘 要：在已有无证书签密的基础上，将 tag-KEM（tag-key encapsulation mechanism）技术引入到无证书公钥密码体制中，实现了一种无证书的 tag-KEM 签密方案，与 DEM 相结合，可构成无证书的混合签密方案，并在随机预言模型下证明了该混合签密方案是安全的。该方案的对运算比 Li Fa-gen 的无证书混合签密方案的对运算少一次，效率更高。

关键词：无证书签密；混合签密；tag 密钥封装机制；数据封装机制
中图分类号：TP309 **文献标志码**：A **文章编号**：1001-3695(2011)09-3527-05
doi:10.3969/j.issn.1001-3695.2011.09.092

New certificateless hybrid signcryption

JIN Chun-hua, LI Xue-jun, WEI Peng-juan, WANG Li-chuan
(*Key Laboratory of Computer Network & Information Security for Ministry of Education, Xidian University, Xi'an* 710071, *China*)

Abstract: This paper investigated the possibility of constructing certificateless signcryption schemes using tag-KEM techniques in the certificateless signcryption setting. It could build certificateless hybrid signcryption scheme by combing with DEM. Under the random oracle model, the hybrid signcryption scheme was proved to be secure. The pairings are less a time than that of Li Fa-gen's certificateless hybrid signcryption scheme, and the efficiency is higher.

Key words: certificateless signcryption; hybrid signcryption; tag-KEM; DEM(data encapsulation mechanism)

1 概述

签密[1]是 Zheng 在 1997 年提出的，它能够在一个合理的逻辑步骤内同时完成数字签名和公钥加密两项功能，其计算量和通信成本都要低于传统的先签名后加密，因而它是实现保密且认证的传输消息较为理想的方法。此后许多签密方案被陆续提出[2-4]，这些签密方案从结构上来看都属于混合密码体制。作为公钥密码体制的一种类型，混合签密体制也不可避免地需要面临一般公钥密码体制中存在的公钥随机化问题。为简化密钥管理问题，在实际应用中把主体和其公钥结合起来是十分必要的。Shamir[5]最早提出了基于身份的密码体制来解决这一问题。在基于身份密码体制中，用户的公钥可以直接从其身份信息（如姓名、身份证号、e-mail 地址等）得到，而私钥则是由一个称为私钥生成中心（private key generation，PKG）的可信方生成。基于身份的公钥密码体制所固有的密钥托管问题，致使该系统下的签密方案有潜在的安全威胁。无证书公钥密码体制是介于传统公钥密码体制和基于身份公钥密码体制之间的一种特殊公钥密码体制，最初由文献[6]提出。它既克服了基于身份的公钥密码体制所固有的密钥托管问题，同时又克服了传统公钥密码体制下的证书管理问题。

在无证书公钥密码体制中，密钥生成中心（key generating center，KGC）生成用户的部分私钥，用户使用这个部分私钥和自己选取的秘密值独立生成自己的公钥和私钥。这样既消除了基于身份密码体制中的密钥托管问题，也无须使用传统公钥密码体制中的公钥认证中心发布和管理公钥证书。无证书公钥密码体制因其独特的优势而被广泛研究，出现了许多无证书公钥加密方案和签名方案[7,8]。

2003 年 Cramer 等人[9]对混合密码进行形式化定义和分析，提出了著名的 KEM-DEM 结构。它由两部分组成，即密钥封装技术（KEM）与数据封装技术（DEM）。KEM 部分利用公钥加密算法产生用于对称加密的密钥和对该密钥的加密密文；DEM 利用 KEM 中得到的加密密钥，用对称加密算法对真正需要加密的消息进行加密。其中 KEM 与公钥加密很类似，不同的是，通过 KEM 产生的是对称密钥和对该密钥的加密密文而不是消息的加密密文，而 DEM 与对称加密类似，只是它所用的加密密钥是由 KEM 随机产生的，是一次性对称加密。

KEM/DEM 混合结构的最大优点是将整个加密算法分为相互独立的两部分，各部分的安全性可以分别研究。为了使整个混合加密达到某种安全，只要各组成部分分别达到相应的安全水平即可，这为混合签密结构的安全性分析带来了极大的益处。2005 年 Bjørstad[10]和 Dent[11,12]对 KEM-DEM 结构进行了扩展[13]，借鉴混合加密的思想提出了混合签密结构。文献[2,10,12]对这种混合签密结构作进一步研究，给出了签密的 KEM/DEM 结构和相应的内部安全与外部安全的安全模型，使得签密研究取得了重大进展。

收稿日期：2010-09-22；**修回日期**：2010-11-01 **基金项目**：国家自然科学基金资助项目（60772136）；高等学校学科创新引智计划资助项目（B08038）；中央高校基本科研业务费专项资金资助项目（JY10000901010）

作者简介：金春花（1980-），女，山东郓城人，硕士，主要研究方向为计算机信息安全、密码学（xajch0206@163.com）；李学俊（1969-），女，副教授，主要研究方向为信息安全、椭圆曲线；魏鹏娟（1985-），女，硕士，主要研究方向为信息安全；王立川（1984-），男，硕士，主要研究方向为信息安全.

2005 年,文献[14]引入了一种新的混合加密结构 tag-KEM/DEM。所谓的 tag-KEM 是在密钥封装 KEM 中连同另外一个信息 tag 一起封装。tag-KEM 实际上是一种可认证的 KEM,可如同一般 KEM 那样与 DEM 结合形成混合密码体制,但用 tag-KEM 代替 KEM 可以得到性能更好的混合密码。已有研究表明在这种新的混合结构下,如果 tag-KEM 是 CCA 安全的,DEM 只要是被动攻击安全的,就可使整个混合加密达到 CCA 安全水平。2009 年,Li 等人[15]提出了无证书的混合签密方案,在此基础上,根据文献[16]的方案构造了一个新的无证书混合签密方案,并在随机预言模型下证明了该方案满足内部安全性要求。

2　预备知识

2.1　双线性对

令 G_1 为 q 阶加法循环群,G_2 为 q 阶乘法循环群,其中 q 是素数。双线性对 $\hat{e}: G_1 \times G_1 \to G_2$ 是满足以下性质的映射:

假设 G_1 和 G_2 这两个群中的离散对数问题都是困难问题。双线性对是指满足下列性质的一个映射:

a)双线性。对于 $P, Q \in G_1$,以及 $a, b \in Z_q$,有 $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$。

b)非退化性。$\hat{e}(P, Q) \neq 1$。

c)可计算性。对于 $P, Q \in G_1$,存在一个有效的算法来计算 $\hat{e}(P, Q)$。

2.2　复杂性假设

a)离散对数问题。设 g 为循环群 G 的生成元,群 G 的阶为 q,对于 $y \in G$,给定 $g^x = y$,求 $x \in Z_q$ 的值。

b)计算 Diffie-Hellman(CDHP)问题。对于 $a, b \in Z_q^*$,给定 P、aP、bP,计算 abP。

c)离散对数的数字签名。设 p、q 为两个大素数,且 $q \mid p-1$,$g, h \in Z_q^*$ 的阶为 q,对 $x \in Z_q^*$,$y = g^x$,如果 $c = H(g \| y \| g^s \| y^c \| m)$,则 (c, s) 是 y 相应于底数 g 的对消息 m 的签名。

2.3　无证书签密(CLSC)模型

无证书签密方案的合法参与者包含:私钥生成中心 KGC,签密者 S,解签密者 R。此签密模型由如下七个多项式时间算法组成:

a)系统参数建立算法。输入参数 k,输出主密钥 s。计算 $P_{pub} = sP$ 并输出系统参数 params。

b)部分私钥提取算法。输入 params、主密钥 s 和用户的身份 $ID_i \in \{0,1\}^*$。输出用户的部分私钥 D_i。

c)设置秘密值算法。输入 params 和用户的身份 ID_i,输出用户的秘密值 x_i。

d)私钥提取算法。输入 params、用户的部分私钥 D_i 和用户的秘密值 x_i,输出用户的私钥 S_i。

e)公钥提取算法。输入 params、用户的身份 ID_i 以及用户的秘密值 x_i,输出用户的公钥 PK_i。

f)签密算法。输入 params、消息 m、签密者的私钥 S_s、解签密者的身份 ID_r,PK_r,输出签密 SC。

g)解签密算法。输入 params、签密者的公钥 PK_s、解签密者的私钥 S_r,输出明文 m。如果签密验证通过,输出“接受”;否则输出“拒绝”。

2.4　数据封装机制(DEM)

DEM 是一次对称加密方案,它由两个算法组成:

a)Encrypt。DPT 的加密算法。输入安全参数 k,对称密钥 K 和消息 m,输出密文 c。本文用 $c \leftarrow \mathrm{Enc}(K, m)$ 来表示。

b)Decrypt。DPT 的解密算法。输入安全参数 k,对称密钥 K 和密文 c,输出消息 m 或特殊符号 $\perp$。

注意:DEM 没有密钥生成算法。因为它的密钥是由 KEM 随机生成的。

2.5　无证书的签密 TKEM(CLSC-TKEM)模型

CLSC-TKEM 由下列八个算法组成:

a)系统参数建立算法。同 2.3 节中的 CLSC。

b)部分私钥提取算法。同 2.3 节中的 CLSC。

c)设置秘密值算法。同 2.3 节中的 CLSC。

d)私钥提取算法。同 2.3 节中的 CLSC。

e)公钥提取算法。同 2.3 节中的 CLSC。

f)对称密钥产生算法。输入系统参数 params,发送者的全私钥 S_s,身份 ID_s 和公钥 PK_S,接收者的身份 ID_r 和公钥 PK_r,输出对称密钥 K 和内部状态信息 ω。本文表示为 $(K\omega) = \mathrm{sym}(\mathrm{params}, S_s, ID_s, PK_s, ID_r, PK_r)$。

g)封装算法。输入内部状态信息 ω 和任意比特串 τ,输出封装 φ,表示为 $\varphi = \mathrm{Encap}(\omega, \tau)$。

h)解封装算法。输入系统参数 params,封装 φ,标志 τ,发送者的身份 ID_s 和公钥 PK_S,接收者的身份 ID_r 和公钥 PK_r,接收者的全私钥 S_r,输出对称密钥 K 或 $\perp$,表示为 $K = \mathrm{Decap}(\mathrm{params}, \varphi, \tau, ID_s, PK_s, S_r, ID_r, PK_r)$。

2.6　安全模型

无证书混合签密方案需要同时考虑签密的机密性和不可伪造性。为了考虑 CLSC-TKEM 的安全性,只需要把 CLSC 的安全模型适应性添加到 TKEM 框架中。文献[6]定义了两种类型的攻击:a)攻击者 A_I 能替换用户的公钥,但不能获得主密钥 s,同时 A_I 能够多项式次适应性进行 hash 询问,如果最后 A_I 输出一个在自己选取的新公钥下有效的消息签密对,且该消息没有进行过签密询问,则敌手 A_I 获得成功;b)攻击者 A_II 可以获取主密钥,但不能替换公钥,与 A_I 一样也能多项式次适应性进行 hash 询问,最后 A_II 输出一个有效的消息签密对,且该消息没有进行过签密询问,则敌手 A_II 获得成功。

3　无证书的混合签密

可结合 CLSC-TKEM 和 DEM 构成 CLSC 方案,其中 tag 是 DEM 输出的密文,这样的构造能够得到更简单的方案和更好的安全性规约。本文得到一个与文献[16]相似的方案,如下所示:

CLSC.Setup　输入参数 1^k:

1.(params, s) ←CLSC-TKEM.Setup(1^k)

2. 输出系统参数 params 和主密钥 s

CLSC.Partial-Private-Key-Extract　输入 params、主密钥 S 和用户的身份 $ID_i \in \{0,1\}^*$:

1. $D_i \leftarrow$CLSC-TKEM,Partial-Private-Key-Extract(params, s, ID_i)

2. 输出用户的部分私钥 D_i

CLSC.Set-Secret-Value　输入 params 和用户的身份 D_i:

1. $x_i \leftarrow$CLSC-TKEM.Set-Secret-Value(params, ID_I)

2. 输出用户的秘密值 x_i

CLSC. Set-Private-Key　输入 params、用户的部分私钥 D_i 和用户的秘密值 x_i:

1. S_i←CLSC-TKEM. Set-Private-Key(params, D_i, X_i)

2. 输出用户的私钥 S_i

CLSC. Generate-User-Keys　输入 params、用户的身份 ID_i 以及用户的秘密值 x_i:

1. PK_i←CLSC-TKEM. Generate-User-Key(params, ID_1, X_i)

2. 输出用户的公钥 PK_i

CLSC. Signcrypt　输入 params,消息 $m \in \{0,1\}^*$,发送者的全私钥 S_s,身份 ID_s 和公钥 PK_s,接收者的身份 ID_r 和公钥 PK_r

1. (K,ω)←CLSC-TKEM. Sym(params, S_s, ID, PK_s, PK_r)

2. c←DEM. Enc(K, m)

3. φ←CLSC-TKEM. Encap(ω, c)

4. 输出密文 σ←(ω, c)

CLSC. Unsigncrypt　输入 params,密文 σ,发送者的身份 ID_s 和公钥 PK_s,接收者的全私钥 S_r,身份 ID_r 和公钥 PK_r:

1. K←CLSC-TKEM. Decap(params, φ, c, ID_s, PK_s, S_r, ID_r, PK_r)

2. 如果 K = ⊥,输出⊥并停止。

3. m←DEM. Dec(K, c)

4. 输出消息 m。

定理 1　设 CLSC 是由 CLSC-TKEM 和 DEM 组成的混合签密方案,如果 CLSC-TKEM 是 IND-CCA2 安全的,DEM 是 IND-PA 安全的,则 CLSC 是 IND-CCA2 安全的。特别地,有

$$\mathrm{adv}_{\mathrm{CLSC}}^{\text{IND-CCA2-}i}(A) \leqslant 2\mathrm{adv}_{\text{CLSC-TKEM}}^{\text{IND-CCA2-}i}(B_1) + \mathrm{adv}_{\mathrm{DEM}}^{\text{IND-}PA\text{-}i}(B_2)$$

其中:$i \in \{\mathrm{Ⅰ}, \mathrm{Ⅱ}\}$, $\mathrm{adv}_{\mathrm{CLSC}}^{\text{IND-CCA2-}i}(A)$, $\mathrm{adv}_{\text{CLSC-TKEM}}^{\text{IND-CCA2-}i}(B_1)$, $\mathrm{adv}_{\mathrm{DEM}}^{\text{IND-}PA\text{-}i}(B_2)$ 分别是 CLSC 的 IND-CCA2、CLSC-TKEM 的 IND-CCA2 和 IND-PA 的优势。

证明　定义序列 game_0、game_1、game_2 为修改的攻击游戏。游戏之间唯一的区别是以怎样的环境响应 A 的 oracle 询问。

$\sigma' = (\varphi', c')$ 为提交给 A 的挑战密文,根据比特 b,由它的挑战 oracle 来加密消息 m_0 或者 m_1。用 K'表示对称密钥,通过挑战 oracle 来生成挑战的密文,或者,用敌手选择的身份 ID'_s、ID'_r 解封装 φ'。对任何 $i=0,1,2$,在游戏 game_i 中,用 S_i 表示事件 $\delta' = \delta$,δ 表示由 A 的挑战 oracle 选择的比特,δ'表示由 A 输出的比特。而它们之间的概率就是 A 的任意选择和 A 的 oracle 的任意选择。

引理 1　设 E、E'和 F 是定义在概率空间中的事件,表示为 $P_r[E \wedge \neg F] = P_r[E' \wedge \neg F]$,可得到 $|P_r[E] - P_r[E']| \leqslant P_r[F]$。

a) game_0。在实际攻击中,通过运行对称密钥产生算法和由此产生的密钥响应 A 的询问,来模拟敌手的观点。所以,敌手 A 的观点和在实际攻击中的观点是一样的。因此,有

$$\left| P_r[S_0] - \frac{1}{2} \right| = \frac{1}{2}\mathrm{adv}_{\mathrm{IDSC}}^{\text{IND-CCA2-}i}(A)$$

其中:$i \in \{\mathrm{Ⅰ}, \mathrm{Ⅱ}\}$。

b) game_1。在这个游戏中,本文稍微地修改了解签密 oracle 如何响应 A 的询问。调用挑战的签密 oracle 之后把发送者的身份 ID_s、接收者的身份 ID_r 和 (φ, c) 提交给解签密 oracle。如果 $ID_s = ID'_s$,$ID_r = ID'_r$,$\varphi = \varphi'$,对第一种类型的敌手来说,ID'_s 和 ID'_r 的公钥都没有被替换,解签密 oracle 不能用真实的解签密程序得到对称密钥。它用 K'解密密文 c 并把结果发送给 A。

很明显,这样的改变并没有影响到敌手,所以 $P_r[S_2] = P_r[S_1]$。

c) game_2。在这个游戏中,用一个随机的对称密钥 K^* 代替 K'来修改 game_1。其结果遵循下面的两个引理。

引理 2　存在一个 ppt 的算法 B_1,其运行时间与 A 的一样,有 $|P_r[S_2] - P_r[S_1]| = \mathrm{adv}_{\text{CLSC-TKEM}}^{\text{IND-CCA2-}i}(B_1)$。其中 $i \in \{\mathrm{Ⅰ}, \mathrm{Ⅱ}\}$。

证明　为了证明上式成立,必须说明如何构造一个无证书的签密 TKEM 的敌手 B_1,违背 IND-CCA2-Ⅰ(resp. IND-CCA2-Ⅱ)的安全性。

当 A 调用包括他的签密、解签密和挑战的签密 oracle 在内的任何 oracle 时,B_1 用他自己对应的 oracle 回答这些询问。

当 A 对发送者的身份 ID_s,接收者的身份 ID_r 和明文 m 执行签密询问时,B_1 按以下步骤执行:

a)用他自己的对称密钥生成的 oracle 对身份 (ID_s, ID_r) 执行对称密钥产生询问获得对称密钥 K。

b)计算 c←DEM. Enc(K, m)。

c)用他自己的密钥封装 oracle 对 c 执行密钥封装询问获得 φ。

d)把密文 $\sigma = (\varphi, c)$ 发送给 A。

当 A 对发送者的身份 ID_s、接收者的身份 ID_r 和密文 $\sigma = (\varphi, c)$ 执行解封装询问时,B_1 按以下步骤执行:

a)用他自己的密钥解封装 oracle 对 (φ, c, ID_s, ID_r) 执行解封装询问获得对称密钥 K。

b)如果 $K = \perp$,返回⊥并终止。

c)计算 m←DEM. Dec(K, c) 并返回 m。

为了响应 A 对发送者的 ID_s、接收者的身份 ID_r 和密文 $\sigma = (\varphi, c)$ 的解封装询问,B_1 按以下步骤执行:

a)如果 $(ID_s, ID_r, \varphi) \neq (ID'_s, ID'_r, \varphi')$,他的执行过程和 A 调用他自己的挑战签密 oracle 之前的过程是一样的。

b)在 CLSC 第一种类型的攻击中,如果 $(ID_s, ID_r, \varphi) = (ID'_s, ID'_r, \varphi')$,公钥已经被替换,然后 B_1 通过 A 提供的解封装 oracle 来响应 A,并输入 $(ID'_s, ID'_r, \varphi', c')$ 获得对称密钥 K,用 K 解签密 c 并发送给 A。

c)否则,B_1 用 K_b 解签密 c 并发送给 A。

模拟结束时,A 输出 δ'。如果 $\delta' = \delta$,B_1 输出 $b' = 1$,意味着 K_b 是真正的对称密钥 K;否则,A 输出 $b' = 0$,意味着 K_b 是一个随机密钥。

当 K_b 是真正的密钥时,A 就像是在 game_1 中运行,这意味着 $P_r[S_1] = P_r[\delta' = \delta | b = 1] = P_r[b' = 1 | b = 1]$;当 K_b 是任意的密钥时,A 就像是在 game_2 中运行,这意味着 $P_r[S_2] = P_r[\delta' = \delta | b = 0] = P_r[b' = 1 | b = 0]$。

从 CLSC-TKEM 的安全性定义,有 $\mathrm{adv}_{\text{CSLC-TKEM}}^{\text{IND-CCA2-}i}(B_1) = |2P_r[b' = b] - 1| = |P_r[b' = 1 | b = 1] - P_r[b' = 1 | b = 0]|$,所以该结果成立。

引理 3　存在一个 ppt 的算法 B_2,其运行时间和 A 的一样,有 $\left| P_r[S_2] - \frac{1}{2} \right| = \frac{1}{2}\mathrm{adv}_{\mathrm{DEM}}^{\text{IND-PA}}(B_2)$。

证明　为了构造这样一个 B_2,只需要像在 game_2 中运行 A 一样。运行合适的 CLSC-TKEM 算法,以便在调用 A 的挑战签密 oracle 之前响应 A 的询问。当 A 对发送者的身份 ID'_s、接收者的身份 ID'_r 和消息 (m_0, m_1) 调用他的挑战签密 oracle 时,仅依赖 B_2 的挑战加密 oracle 加密 (m_0, m_1) 得到密文 c'。对 (ID'_s, ID'_r) 执行对称密钥产生询问和密钥封装询问,得到 K'和 φ',并把 (φ', c') 发送给 A。B_2 像以前一样继续回答 A 的询

问,但是不能对密文(ID'_s,ID'_r,φ',c)执行解封装询问,对某些 c。

在这个例子中有两种情况:

a)如果处理 CLSC 方案中类型 I 的敌手,公钥已经被替换。B_2 用已经提供的私钥对(ID'_s,ID'_r,φ',c')执行解封装获得 K,并用 K 解密 c 来响应 A。

b)否则,用 B_2 的解密 oracle 询问 c 来响应 A。

在这个模拟中,由 B_2 运行的 A 和在 game_2 中运行的 A 方式是一样的。

定理 2　假设 CLSC 是由 CLSC-TKEM 和 DEM 构成的混合签密方案,如果 CLSC-TKEM 是 sUF-CMA 安全的,那么,CLSC 是 sUF-CMA 的。特别地:

$$\mathrm{adv}_{\mathrm{CLSC}}^{\text{sUF-CMA-}i}(A) \leqslant \mathrm{adv}_{\text{CLSC-TKEM}}^{\text{sUF-CMA-}i}(B)$$

其中:$i \in \{\mathrm{I},\mathrm{II}\}$。

证明　假设敌手 A 以概率 $\mathrm{adv}_{\mathrm{CLSC}}^{\text{sUF-CMA-}i}(A)$(其中 $i \in \{\mathrm{I},\mathrm{II}\}$)破坏 CLSC 的签密方案,根据这个构造一个算法 B,它也至少以概率 $\mathrm{adv}_{\mathrm{CLSC}}^{\text{sUF-CMA-}i}(A)$ 破坏 CLSC-TKEM 的 sUF-CMA 游戏。

敌手 B 是由运行的敌手 A 构造成的,响应敌手 A 的询问如下:

当 A 调用包括他的签密、解签密和挑战的签密 oracle 在内的任何 oracle 时,B 用他自己对应的 oracle 回答这些询问。

当 A 对发送者的身份 ID_s、接收者的身份 ID_r 和明文 m 执行签密询问时,B 按以下步骤执行:

a)用他自己的对称密钥生成的 oracle 对身份(ID_s,ID_r)执行对称密钥产生询问获得对称密钥 K。

b)计算 $c \leftarrow \mathrm{DEM.Enc}(K,m)$。

c)用他自己的密钥封装 oracle 对 c 执行密钥封装询问,获得 φ。

d)把密文 $\sigma=(\varphi,c)$ 发送给 A。

当 A 对发送者的身份 ID_s、接收者的身份 ID_r 和密文 $\sigma=(\varphi,c)$ 执行解封装询问时,B 按以下步骤执行:

a)用他自己的密钥解封装 oracle 对(φ,c,ID_s,ID_r)执行解封装询问获得对称密钥 K。

b)如果 $K=\perp$,返回 $\perp$ 并终止。

c)计算 $m \leftarrow \mathrm{DEM.Dec}(K,c)$ 并返回 m。

最后,A 输出一个伪造(m',σ',ID'_s,ID'_r),其中(φ',c')←σ',B 输出(τ',φ',ID'_s,ID'_r),其中 $\tau'=c'$。

很明显,这个算法很好地模拟了 A 运行的环境,如果 A 赢得了 CLSC 的 sUF-CMA-i 的游戏,则 B 以同样的概率赢得了 CLSC-TKEM 的 sUF-CMA-i 游戏。

4　本文提出的改进方案

a)系统参数建立。设 G_1、G_2 分别是阶为 q 的加法循环群和乘法循环群,P 为 G_1 的生成元,ID_s 和 ID_r 分别为发送者和接收者的身份。三个 hash 函数 $H_1:\{0,1\}^* \rightarrow G_1$,$H_2:\{0,1\}^* \rightarrow Z_q^*$,$H_3:\{0,1\}^* \rightarrow \{0,1\}^n$,其中 n 为 DEM 的长度。KGC 随机选取主密钥 $s \in Z_q^*$,并计算系统公钥 $P_{\mathrm{pub}}=sP$。令 $T=\hat{e}(P,P)$,则 KGC 公开系统参数 $\{G_1,G_2,P,P_{\mathrm{pub}},T,q,e,n,H_1,H_2,H_3\}$,并保密主私钥 s。

b)部分私钥提取。输入用户的身份 $\mathrm{ID}_i(i=s,r)$,KGC 计算 $Q_{\mathrm{ID}_i}=H_1(\mathrm{ID}_i)$,输出部分私钥 $D_i=sQ_i$。

c)设置秘密值。用户随机选取 $x_i \in Z_q^*(i=s,r)$ 作为自己的秘密值。

d)设置私钥。用户运行此算法,计算自己的私钥 $S_i=(x_i,D_i)(i=s,r)$。

e)设置公钥。用户运行此算法,计算自己的公钥 $PK_i=T^{x_i}(i=s,r)$。

f)对称密钥产生。输入发送者的全私钥 S_s,身份 ID_r 和公钥 PK_s,接收者的身份 ID_r 和公钥 PK_r,算法如下:

(a)随机选取 $b,b_1,b_2 \in Z_q^*$,计算 $R_1=T^{b_1}$,$R_2=T^{b_2}$。

(b)计算 $Z=bQ$。

(c)计算 $t=\hat{e}(S_s,Q_r)PK_r^{x_s}$。

(d)$K=H_3(t)$。

(e)$\omega=(b,R_1,R_2,Z,S_s,\mathrm{ID}_s,PK_s,\mathrm{ID}_r,PK_r)$。

(f)输出 K 和 ω。

(g)封装:输入任意的比特串标志 τ 和内部状态信息 ω,算法如下:

①计算 $h=H_2(\tau \| R_1 \| R_2 \| PK_s \| PK_r)$;

②计算 $U=b_1P-hD_s$,$u=b_2-x_sh$;

③输出 $\varphi=(Z,U,u,h)$。

(h)解封装:输入发送者的身份 ID_s 和公钥 PK_S,接收者的全私钥 S_r 和身份 ID_r,公钥 PK_r,封装 φ 和标志 τ,算法如下:

①计算 $t=\hat{e}(S_r,Z)PK_s^{x_r}$;

②计算 $K=H_3(t)$;

③验证

$$h=H_2(\tau \| \hat{e}(U,P)\hat{e}(Q_s,P_{\mathrm{pub}}) \| T^u PK_s^{\,h} \| PK_s \| PK_r)$$

是否成立,若成立,输出 K;否则,输出 $\perp$。

5　方案分析

5.1　正确性验证

$$\hat{e}(U,P)\hat{e}(Q_{\mathrm{ID}_s},P_{\mathrm{pub}})^h=\hat{e}(b_1P-hD_s,P)\hat{e}(Q_{\mathrm{ID}_s},sP)^h=$$
$$\hat{e}(b_1P,P)\hat{e}(-hsQ_{\mathrm{ID}_s},P)\hat{e}(Q_{\mathrm{ID}_s},P)^{hs}=$$
$$\hat{e}(P,P)^{b_1}\hat{e}(Q_{\mathrm{ID}_s},P)^{-hs}\hat{e}(Q_{\mathrm{ID}_s},P)^{hs}=\hat{e}(P,P)^{b_1}=T^{b_1}=R_1$$
$$T^u PK_s^{\,h}=\hat{e}(P,P)^{b_2-x_sh}\hat{e}(P,P)^{x_sh}=\hat{e}(P,P)^{b_2-x_sh+x_sh}=$$
$$\hat{e}(P,P)^{b_2}=R_2$$

5.2　保密性

要验证对称密钥 K 的保密性,在解封装阶段,$K=H_3(\hat{e}(S_r,Z)PK_s^{x_r})$,在给定($Q_r$,$Z$,$P_{\mathrm{pub}}$)和($P$,$PK_s$,$PK_r$)的前提下,计算出 S_r 和 x_r 是困难问题。显然,本文方案满足保密性。

5.3　不可伪造性

在无证书密码系统中,存在两种类型的攻击:a)攻击者 A_{I} 能替换用户的公钥,但不能获得主密钥;b)攻击者 A_{II} 可以获得主密钥,但不能替换公钥。

下面证明攻击者在满足等式 $h=H_2(\tau \| \hat{e}(U,P)\hat{e}(Q_s,P_{\mathrm{pub}}) \| T^u PK_s^{\,h} \| PK_s \| PK_r)$ 的条件下构造(Z,U,u,h,τ),来证明本文方案具有不可否认性。

定理 3　在 CDH 困难问题假设下的随机预言机模型中,若攻击者为 A_{I},则本文方案具有不可伪造性。

证明(大意)　A_{I} 能替换发送者 s 的公钥,也就是说,A_{I} 知

道秘密值 x_s,但不知道 D_s。A_{I} 想要构造满足等式(Z,U,u,h,τ),也就意味着伪造一个 Hess 签名。在文献[17]中,Hess 签名的安全性归约为 CDH 困难问题,也就是离散对数签名问题。

定理 4　在 DL 困难问题假设下的随机预言机模型中,若攻击者为 A_{II},则本文方案具有不可伪造性。

证明(大意)　攻击者 A_{II} 知道主密钥,也就是说 A_{II} 知道接收者的部分私钥。在随机语言模型中(hash 函数变为随机预言机),区分者 C 接收一个随机的 DL 问题实例(T,T^a),他的目标是计算出 a。C 把 A_{II} 作为子程序并扮演 sUF-CMA 游戏中的挑战者,C 维护 L_1、L_2、L_3 三张列表,这些列表开始都为空,它们分别用于跟踪 A_{II} 对预言机 H_1、H_2、H_3 的询问。假设 A_{II} 用身份 ID_* 构造一个封装,则 ID_* 的公钥设为 T^a。利用分叉引理技术[18],得到两个对标志 τ_* 的伪造签名$(\tau_*,\mathrm{ID}_*,P_*,U,u,h)$和$(\tau_*,\mathrm{ID}_*,P_*,U',u',h')$,其中,$U\neq U'$,$h\neq h'$。$C$ 得到等式 $T^uP_*^h=T^{u'}P_*^{h'}$,通过计算 $u+ah=u'+ah'$,C 得到 $a=(u-u')(h'-h)^{-1}$,因此,C 解决了 DL 困难问题。

5.4　不可否认性

在解封装阶段,恢复用户的对称密钥 $t=\hat{e}(S_r,Z)PK_s^{x_r}$ 要用到签密的公钥 PK_s,在签名验证阶段 $h=H_2(\tau\|\hat{e}(U,P)\hat{e}(Q_s,P_{\mathrm{pub}})^h\|T^uPK_s^h\|PK_s\|PK_r)$ 中也要用到签密者的公钥,所以签密者无法否认自己的签密。因此,该方案满足不可否认性。

5.5　前向安全性

假设攻击者已获得了签密者的全私钥 S_s,但是如果不知道 r_s,攻击者不能得到 $\hat{e}(S_{\mathrm{ID}_r},Z)$ 的值,求不出对称密钥 K,所以本文方案是前向安全的。

5.6　可公开验证性

在解封装阶段,等式 $h=H_2(\tau\|\hat{e}(U,P)\hat{e}(Q_s,P_{\mathrm{pub}})^h\|T^uPK_s^h\|PK_s\|PK_r)$ 中的参数都已经公开,所以可供任意第三方验证。

5.7　效率

对的运算是最耗时间的,在本文方案中,用到了四个对运算,而在 Li 的方案中[15],用到了六个对运算,相对来说,本文方案比文献[15]效率高。

5.8　与文献[16]方案的比较

表 1 给出了本文方案与文献[16]方案的比较结果,如表 1 所示。

表 1　本文方案与文献[16]方案的比较结果

scheme	security requirement				efficiency		
	Repu	FwSec	PubVer	ProSec	Pa	Mu	Ex
本文方案	Y	Y	Y	Y	4(+6)	4	3(+3)
文献[16]方案	N	Y	Y	Y	4(+7)	4	4(+4)

6　结束语

本文提出的无证书混合签密方案,保持了无证书加密方案在公钥分发和管理上的优势,在随机预言模型和 CDH、DL 困难问题的假设下,无证书的 tag-KEM 签密方案被证明是安全的,并且采用这种结构可以用更少的公钥加密开销,加密任意长度消息。

参考文献:

[1] ZHENG Y. Digital signcryption or how to achieve cost (signature &encryption << cost (signature) + cost (encryption) [C]//Advances in Cryptology-CRYPTO'97, Lecture Notes in Computer Science, vol 1294. Berlin: Springer-Verlag, 1997: 165-179.

[2] ZHENG Y. Signcryption and it's applications in efficient public key solutions[C]//Proc of ISW'97. Berlin: Springer, 1998: 291-312.

[3] YUM B H, LEE P J. New signcryption schemes based on KCDSA [C]//Proc of ICISC'01. Berlin: Springer, 2001: 305-317.

[4] ZHAO Fu-xiang, ZHAO Hong-yun, WNAG Yu-min. A novel scheme of the mobile agent with the cipher text only signature verification [J]. Journal of Computer Research and Development, 2001, 38(7): 811-814.

[5] SHAMIR A. Identity-based cryptosystems and signature schemes [C]//Advances in Cryptology, Proc of CRYPTO'84. Berlin: Springer, 1985: 48-53.

[6] AL-RIYAMI S S, PATERSON K. Certificateless public key cryptography[C]//Proc of Asiacrypt'03. [S. l.]: Springer-Verlag, 2003.

[7] CHENG Z H, COMLEY R. Efficient certificate-less public key encryption, Report 2005/012[R]. [S. l.]: Cryptology ePrint Archive, 2005.

[8] YUM D H, LEE P J. Generic construction of certificateless signature [C]//Proc of ACISP 2004, LNCS 3108. [S. l.]: Springer-Verlag, 2004: 200-211.

[9] CRAMER R, SHOUP V. Design and analysis of practical publickey encryption schemes secure against adaptive chosen ciphertext attack [J]. SIAM Journal on Computing, 2003, 33 (1): 167-226.

[10] BJϕRSTAD T E. Provable security of signcryption[D]. [S. l.]: Norwegian University of Technology and Science, 2005: 491-507.

[11] DENT A W. Hybrid cryptography, Report 2004/210[R]. [S. l.]: Cryptology ePrint Archive, 2004: 553-571.

[12] DENT A W. Hybrid signcryption schemes with insider security[C]// Proc of ACISP. [S. l.]: Springer-Verlag, 2005: 253-266.

[13] DENT A W. Hybrid signcryption schemes with outsider security [C]//Proc of ISC. Berlin: Springer, 2005: 203-217.

[14] ABE A, GENNARO R, KUROSAWA K, *et al*. Tag- KEM/DEM: a new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM [C]//Advance in Cryptology, LNCS 3494. Berlin: Springer-Verlag, 2005: 12-46.

[15] LI Fa-gen, SHIRASE M, TAKAGI T. Certificateless hybrid signcryption[C]//Proc of the 5th Information Security Practice and Experience Conference. [S. l.]: Springer-Verlag, 2009: 112-123.

[16] WU Chen-huang, CHEN Zhi-xiong. A new efficient certificateless signcryption scheme [C]// Proc of the 2008 International Symposium on Information Science and Engineering. Berlin: Springer-Verlag, 2008: 661-664.

[17] CAMENISCH J. Efficient and generalized group signatures[C]//Advances in EUROCRYPT, Lecture Notes in Computer Science, vol 1233. Berlin: Springer-Verlag, 1997: 465-479.

[18] HESS F. Efficient identity based signature schemes based on pairings [C]// Advances in SAC, Lecture Notes in Computer Science, vol 2595. Berlin: Springer-Verlag, 2002: 310-324.

[19] POINTCHEVAL D, STERN J. Security proofs for signature schemes [C]// Advances in EUROCRYPT, Lecture Notes in Computer Science, vol 1070. Berlin: Springer-Verlag, 1996: 387-398.

[20] 王会歌,王彩芬,易韩,等. 高效的无证书可公开验证签密方案[J]. 计算机工程,2009, 35(5): 147-150.