(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 103988466 A
(43)申请公布日 2014.08.13

(21)申请号 201280061831.9
(22)申请日 2012.12.11
(30)优先权数据
11306672.4 2011.12.15 EP
(85)PCT国际申请进入国家阶段日
2014.06.13
(86)PCT国际申请的申请数据
PCT/EP2012/075091 2012.12.11
(87)PCT国际申请的公布数据
WO2013/087629 EN 2013.06.20
(71)申请人 汤姆逊许可公司
地址 法国伊西莱穆利诺
(72)发明人 L. 埃尔艾马尼 M. 乔耶
(74)专利代理机构 北京市柳沈律师事务所 11105
代理人 吕晓章
(51)Int.Cl.
H04L 9/08(2006.01)
H04L 9/30(2006.01)
H04L 9/32(2006.01)

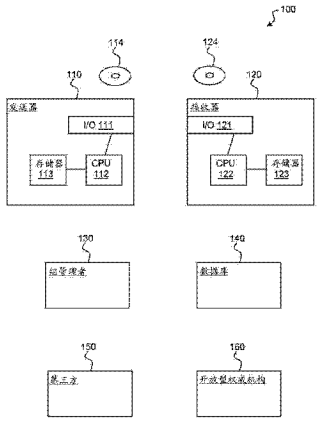
权利要求书2页 说明书9页 附图1页

(54)发明名称

群加密方法及设备

(57)摘要

本发明通过对接收者的公共密钥的别名而不是公共密钥本身进行加密来改进现有技术的群加密方案。组管理者公布在公共数据库DB上的别名的加密、对应的公共密钥以及对应的证书。别名是对公共密钥适当选取的函数f的得到的值，并且可以被视为是该公共密钥的散列。由于别名可以小于公共密钥，因此这可以使得得到的架构的大小和成本显著地降低。具体地，不需要将第二加密策略应用与在接收者的公共密钥中的群元素同样多的次数。



CN 103988466 A

1. 一种关于标签 t 为接收者使用公共密钥 pk 对明文 m 进行群加密以得到密文 c 的方法，该方法包含在设备 (110) 中的以下步骤：

得到签名密钥 OTS.sk 和验证密钥 OTS.vk；

通过计算 $c_1 = E_1.\mathrm{Encrypt}_{\{pk\}}(m, OTS.vk)$ 和 $c_2 = E_2.\mathrm{Encrypt}_{\{pk0A\}}(f(pk), OTS.vk)$ 来创建第一加密值 c_1 和第二加密值 c_2，其中，E_1 是第一加密算法，E_2 是第二加密算法并且 f 是映射函数；

使用所述签名密钥 OTS.sk 通过计算 $s = OTS.\mathrm{Sign}_{\{OTS.sk\}}(c_1, c_2, t)$ 来生成所述第一加密值 c_1、所述第二加密值 c_2 和所述标签 t 上的签名 s，其中，OTS.Sign 是签名算法；以及

输出所述密文 c，其中，所述密文 c 包含所述第一加密值 c_1、所述第二加密值 c_2、所述验证密钥 OTS.vk 和所述签名 s。

2. 如权利要求 1 所述方法，其中，消息 m 满足可公开验证的关系 R。

3. 一种对包含第一加密值 c_1、第二加密值 c_2、验证密钥 OTS.vk 和签名 s 的群加密 c 进行解密的方法，其中，所述签名 s 关于所述第一加密值 c_1、第二加密值 c_2 和标签 t，该方法包含在设备 (120) 中的以下步骤：

接收所述群加密 c；

关于验证密钥 OTS.vk 验证所述签名 s；以及

如果成功地验证了所述签名 s，则使用解密算法 E_1 和所述验证密钥 OTS.vk 对所述第一加密值 c_1 进行解密。

4. 如权利要求 3 所述的方法，其中，签名验证步骤还包含验证所述第一加密值 c_1 的解密满足公开关系 R。

5. 一种用于关于标签 t 为接收者使用公共密钥 pk 对明文 m 进行群加密以得到密文 c 的设备 (110)，该设备 (110) 包含处理器 (112)，该处理器被配置为：

得到签名密钥 OTS.sk 和验证密钥 OTS.vk；

通过计算 $c_1 = E_1.\mathrm{Encrypt}_{\{pk\}}(m, OTS.vk)$ 和 $c_2 = E_2.\mathrm{Encrypt}_{\{pk0A\}}(f(pk), OTS.vk)$ 来创建第一加密值 c_1 和第二加密值 c_2，其中，E_1 是第一加密算法，E_2 是第二加密算法并且 f 是映射函数；

关于所述第一加密值 c_1、所述第二加密值 c_2 和所述标签 t 使用所述签名密钥 OTS.sk 通过计算 $s = OTS.\mathrm{Sign}_{\{OTS.sk\}}(c_1, c_2, t)$ 来生成签名 s，其中，OTS.Sign 是签名算法；以及

输出所述密文 c，其中，所述密文 c 包含所述第一加密值 c_1、所述第二加密值 c_2、所述验证密钥 OTS.vk 和所述签名 s。

6. 如权利要求 5 所述的设备，其中，消息 m 满足可公开验证的关系 R。

7. 一种用于对包含第一加密值 c_1、第二加密值 c_2、验证密钥 OTS.vk 和签名 s 的群加密 c 进行解密的设备 (120)，其中，所述签名 s 关于所述第一加密值 c_1、第二加密值 c_2 和标签 t，该设备 (120) 包含处理器 (122)，该处理器被配置为：

接收所述群加密 c；

关于验证密钥 OTS.vk 验证所述签名 s；以及

如果成功地验证了所述签名 s，则使用解密算法 E_1 和所述验证密钥 OTS.vk 对所述第一加密值 c_1 进行解密。

8. 如权利要求 7 所述的设备，其中，所述处理器还被配置为验证所述第一加密值 c_1 的

解密满足公开关系 R。

9. 一种计算机程序产品 (114)，在其上存储在被处理器执行时实施如权利要求 1 或 2 所述的方法的指令。

10. 一种计算机程序产品 (124)，在其上存储在被处理器执行时实施如权利要求 3 或 4 所述的方法的指令。

群加密方法及设备

技术领域

[0001] 本发明一般涉及加密码术，更具体地，涉及群加密 (group encryption)。

背景技术

[0002] 这部分旨在向读者介绍与在下面说明和 / 或请求保护的本发明的各个方面相关的技术领域的各个方面。相信该论述有助于向读者提供背景信息以便更好地理解本发明的各个方面。相应地，应当理解的是，这些陈述应当就此而论地阅读，而不是作为对现有技术的承认。

[0003] 在这部分中，定义了群加密原语，呈现了必要的构件块（公共密钥加密、基于标签的加密以及一次性签名），并且说明了群加密的技术现状。

[0004] 作为群签名的加密模拟，Kiayias-Tsiounis-Yung 提出了群加密（参见 Aggelos Kiayias、Yiannis Tsiounis 和 Moti Yung 的《Group Encryption》，ASIACRYPT2007：181-199 页）。群加密在期望隐藏在一组合法用户中的一接收者（解密者）的情况下很有用。

[0005] 示意性示例是想要将某些广告发送给订阅用户的网络服务提供商 (NSP)，其中，该订阅用户的配置文件 (profile) 与要发送的广告相匹配。同时，NSP 想要在对接收者的确切身份保密的同时，向其客户（亦即，付款给该 NSP 来发送广告的公司）证明其确实在其用户组内发送了讨论的广告。广告的接收者的隐私也应当保留在该 NSP 的用户组之内。

[0006] 群加密形成解决该问题的看上去合理的方案，因为其允许发送者（在该示例中的 NSP) 对给目标用户的消息（广告）进行加密，另外，使验证者能够检查形成的密文是有效的（例如，检查对应的明文满足某种关系）并且用户组中的某匿名成员能够对其进行解密。群加密还支持如下功能：在发生纠纷的情况下，由指定的权威机构打开密文并且往下恢复 (recover down) 接收者的身份。

[0007] 更形式化地，群加密 (GE) 方案涉及注册组成员的组管理者以及能够从对应的密文恢复接收者的身份的开放型权威机构 (OA)。

[0008] 构成 GE 方案基础的主要过程有：

[0009] • Join。GM 与可能的组成员之间的交互式协议。GM 发布关于组成员的公共密钥 pk_i 的证书 $cert_i$。GM 还在公共数据库 DB 中存储 $(pk_i, cert_i)$ 对。

[0010] • Encrypt。在目标组成员的公共密钥 pk 下关于输入标签 t 生成输入消息 m 上的密文 c，其中，标签是指定加密上下文的二进制串。为了防止发送伪造的消息，要求要加密的消息 m 满足某一先验关系：m 是关于关系 R 的“实例”x（公共值）的“证据 (witness)”，亦即，$(m,x) \in R$。在这个意义上说，Encrypt 还输出与经加密的证据相对应的实例 x。

[0011] •Decrypt。关于输入标签 t 使用与公共密钥 pk 相对应的私有密钥 sk 恢复被加密成输入密文 c 的消息 m，其中，密文是用公共密钥 pk 创建的。该过程还检查被恢复的消息是否是输入实例 x 的证据，亦即，是否 $(m,x) \in R$。如果是这种情况，则算法输出 m，否则输出“失败 (Fail)”。

[0012] •Open。输入密文 c、标签 t 和 OA 的私有密钥，并且恢复公共密钥 pk，其中，密文是关于输入标签 t 在该公共密钥 pk 下创建的。

[0013] •Prove。从创建密文的实体向任何验证者提供交互式或非交互式的证明。应当使验证者确信讨论的密文是有效的（例如，底层消息满足关系 R）并且某匿名注册组成员可以对其进行解密。

[0014] 公共密钥加密 (PKE) 方案包含产生（公共密钥，私有密钥）形式的对的密钥产生算法、使用接收者的公共密钥生成输入消息的加密的加密算法以及使用适当的私有密钥恢复被加密成输入密文的消息的解密算法。

[0015] 无论对于加密还是对于解密，基于标签的加密方案 (TBE) 都进一步地需要另一参量 (argument)，即标签。非形式化地，标签是指定关于加密的信息（日期、上下文等）的适当长度的二进制串。

[0016] 一次性数字签名方案可以用于对至多一条消息进行签名；否则，签名就可以被伪造。经签名的每条消息均需要新的公共密钥。如同“正常的”数字签名一样，用密钥产生算法、签名算法以及验证算法来定义它们。指定公共密钥，一次性签名方案的安全性依赖于提供消息和对应签名的新的有效对的困难程度。

[0017] 上文提到的 Kiayias-Tsiounis-Yung 的论文提供了一种用于安全的群加密方案的通用架构，该方案使用用于认证用户公共密钥的数字签名方案 S、用于对消息进行加密的基于标签的加密方案 E_1、用于对接收者公共密钥进行加密的另一基于标签的加密方案 E_2 以及用于对所用密钥及其证书进行承诺的承诺方案 (commitment scheme)。更详细地，该方案工作如下：

[0018] ● Join。GM 使用其私有签名密钥 S.sk 生成关于用户的公共密钥 pk 的签名 s（换言之，证书）。GM 还将 (pk, s) 存储在公共数据库 DB 中。

[0019] ● $\mathrm{Encrypt}_{\{pk\}}(m,t)$。为了关于标签 t 使用公共密钥 pk 为接收者对消息 m 进行加密（使得 $(m,x) \in R$，其中，x 是公共值），爱丽丝：

[0020] ○创建关于 pk 的承诺 c_3 以及关于证书的承诺 c_4；

[0021] ○关于标签 (t, c_3, c_4) 在开放型权威机构的公共密钥 pk_{OA} 下使用 E_2.Encrypt 对 pk 进行加密，得到结果 c_2；

[0022] ○关于标签 (t, c_2, c_3, c_4) 在公共密钥 pk 下使用 E_1.Encrypt 对 m 进行加密，得到结果 c_1；

[0023] ○返回元组 (c_1, c_2, c_3, c_4) 作为关于 t 的在 pk 下的 m 的群加密。

[0024] ● $\mathrm{Decrypt}_{\{sk\}}(c,t,x)$。首先，将 c 解析为 (c_1, c_2, c_3, c_4)；然后，针对 c_1 和 (t, c_2, c_3, c_4) 使用私有密钥 sk 调用 E_1.Decrypt；如果 $(m,x) \in R$，则返回结果，即 m，否则返回“失败”。

[0025] ● $\mathbf{Open}_{\{sk_{OA}\}}(c,t)$。首先，将 c 解析为 (c_1, c_2, c_3, c_4)；然后，对 c_2 和 (t, c_3, c_4) 使用 OA 的私有密钥 skOA 调用 E_2.Decrypt 并返回结果。

[0026] ● Prove(c, t, x)。爱丽丝，即创建了密文 $c = (c_1, c_2, c_3, c_4)$ 的实体，关于标签 t 提供如下证明：该密文是有效的并且可以用与公共密钥相对应的私有密钥进行解密，其中，公共密钥被加密成 c_2 并在 c_3 中承诺，公共密钥的证书在 c_4 中承诺。爱丽丝还证明隐藏在密文下面的消息是关于公开关系 (public relation)R 的 x 的证明。爱丽丝使用用于产生

c 的私有币 (private coins)(亦即,用于产生承诺 c_3 和 c_4 以及加密 c_1 和 c_2 的随机值)以提供以上证明。

[0027] Cathalo-Libert-Yung 提供了群加密方案的具体实现(参见 Julien Cathalo、Benoit Libert、Moti Yung 的《Group Encryption:Non-interactive Realization in the Standard Model》,ASIACRYPT2009 :179-196 页)。该方案使用 Shacham 的加密方案(参见 Hovav Shacham 的《A Cramer-Shoup Encryption Scheme from the Linear Assumption and from Progressively Weaker Linear Variants》,Cryptology ePrint Archive,报告 2007/074) 对消息进行加密,并使用 Kiltz 的加密(参见 Eike Kiltz 的《Chosen-Ciphertext Security from Tag-Based Encryption》,TCC2006 :581-600 页)对接收者的公共密钥进行加密。该解决方案通过将承诺 c_3 和 c_4 推迟 (waive) 至在 Prove 过程下面的证明中,从而脱离由 Kiayias-Tsiounis-Yung 提供的架构。

[0028] 更精确地,如果 S 指代在论文中给出的数字签名方案,OTS 指代任何安全的一次签名方案,[Kiltz] 指代 Kiltz 的加密方案,并且 [Shacham] 指代 Shacham 的加密方案,则将该方案定义为 :

[0029] ● Join 关于输入公共密钥 pk,GM 使用其私有签名密钥 S.sk 生成签名(或证书)并且将 (pk,cert) 存储在公共数据库 DB 中。

[0030] ● $\text{Encrypt}_{\{pk\}}(m,t)$ 为了关于标签 t 使用公共密钥 pk 为接收者对消息 m 进行加密(其中,m 是某 (x,y) 的 Diffie-Hellman 解 :e(m,g) = e(x,y),其中,e 是在消息空间 G 下的配对,g 是该群的产生器 (generator)),爱丽丝 :

[0031] ○调用 OTS.keygen 产生一对签名密钥和验证密钥 (OTS.sk,OTS.vk) ;

[0032] ○ 创建 c_1 = [Schacham].$\text{Encrypt}_{\{pk\}}$(m,(OTS.vk,t)) 以及 c_2 = [Kiltz].$\text{Encrypt}_{\{pkOA\}}$(pk,OTS.vk)。用于 c_1 的标签是 (OTS.vk,t),用于 c_2 的标签是验证密钥 OTS.sk ;

[0033] ○使用 OTS.sk 生成 (c_1,c_2,t) 上的一次性签名 ;s = $\text{OTS.Sign}_{\{OTS.sk\}}(c_1,c_2,t)$;

[0034] ○返回 c = $(c_1,c_2,$OTS.vk,s) 作为关于 t 的在 pk 下的 m 的群加密。

[0035] ● $\text{Decrypt}_{\{sk\}}(c,t,x,y)$

[0036] ○将 c 解析为 $(c_1,c_2,$OTS.vk,s) ;

[0037] ○ 关于 OTS.vk 检查在 (c_1,c_2,t) 上的签名 s ;如果 $\text{OTS.Verify}_{\{OTS.vk\}}$(s,(c1,c2,t)) = 0,则返回“失败”,否则计算 [Shacham].$\text{Decrypt}_{\{sk\}}(c_1,$(OTS.vk,t) :如果其是 (x,y) 的 Diffie-Hellman 解,则返回计算出的值,否则返回“失败”。

[0038] ● $\text{Open}_{\{skOA\}}(c,t)$

[0039] ○将 c 解析为 $(c_1,c_2,$OTS.vk,s) ;

[0040] ○关于 OTS.vk 检查 (c_1,c_2,t) 上的签名 s ;如果 $\text{OTS.Verify}_{\{OTS.vk\}}(s,(c_1,c_2,t))$ = 0,则返回“失败”,否则调用 [Kiltz].$\text{Decrypt}_{\{skOA\}}(c_2,$OTS.vk) 并返回其结果。

[0041] ● Prove(c,t,x,y)。爱丽丝,即关于标签 t 创建了密文 c 的实体,提供如下非交互式的证明 :c 是良构的,并且具有经认证的公共密钥的某匿名成员可以对其进行解密。

[0042] 由 Kiayias-Tsiounis-Yung 和 Cathalo-Libert-Yung 提供的方案通过使用满足强安全性概念的加密方案(亦即,针对强大敌手比较安全的加密方案)对该架构进行实例化来得到安全的群加密。根据使用的构件块,得到的实现比较如下 :

[0043] 1.Kiayias-Tsiounis-Yung：他们的通用架构的实例化得到使用1024比特模量的大小为2.5kB的密文以及大小为70kB的证明。而且，该证明需要与验证者之间的交互，因此，如果证明者想要将相同的证明运行若干次，则她需要记住用于产生密文的所有随机性。

[0044] 2.Cathalo-Libert-Yung改进上述方案；其得到更小的密文（使用256比特模量的1.25kB）和更小的证明（16.125kB）。而且，该证明具有非交互式的优点，因此不需要有状态的证明者（stateful prover）。然而，该证明使用开销很大的Groth-Sahai证明系统，该系统需要数百或数千个配对等式验证，这使其变得相当不实用。

[0045] 因此，技术人员将认识到，由于密文和证明的大小或代价，无论是Kiayias-Tsiounis-Yung还是Cathalo-Libert-Yung的开销都仍然相当大。

[0046] 例如，无论是Kiayias-Tsiounis-Yung还是Cathalo-Libert-Yung均采取对公共密钥的每个成分（公共密钥总是包括群元素的矢量）进行加密，因此将开销同样大的（就资源使用而言）加密（“E_2”或[Kiltz]）应用n次，其中，n表示在接收者的公共密钥中的元素的数量。

[0047] 技术人员将意识到，需要一种提供改进的GE方案的解决方案。本发明就提供这样的解决方案。

发明内容

[0048] 第一方面，本发明涉及一种关于标签t为接收者使用公共密钥pk对明文m进行群加密以得到密文c的方法。一种设备，得到签名密钥OTS.sk和验证密钥OTS.vk；通过计算$c_1 = E_1.\mathrm{Encrypt}_{\{pk\}}(m, OTS.vk)$和$c_2 = E_2.\mathrm{Encrypt}_{\{pkOA\}}(f(pk), OTS.vk)$来创建第一加密值$c_1$和第二加密值$c_2$，其中，$E_1$是第一加密算法，$E_2$是第二加密算法，并且f是映射函数；使用所述签名密钥OTS.sk通过计算$s = OTS.\mathrm{Sign}_{\{OTS.sk\}}(c_1, c_2, t)$来生成所述第一加密值$c_1$、所述第二加密值$c_2$和所述标签t上的签名s，其中，OTS.Sign是签名算法；并且，输出所述密文c，其中，所述密文c包含所述第一加密值c_1、所述第二加密值c_2、所述验证密钥OTS.vk以及所述签名s。

[0049] 在第一优选实施例中，消息m满足可公开验证的关系R。

[0050] 第二方面，本发明涉及一种对包含第一加密值c_1、第二加密值c_2、验证密钥OTS.vk和签名s的群加密c进行解密的方法，其中，所述签名s关于所述第一加密值c_1、所述第二加密值c_2和标签t。一种设备，接收所述群加密c；关于验证密钥OTS.vk验证所述签名s；并且，如果成功地验证了所述签名s，则使用解密算法E_1和所述验证密钥OTS.vk对所述第一加密值c_1进行解密。

[0051] 在第一优选实施例中，验证签名还包含验证所述第一加密值c_1的解密满足公开关系R。

[0052] 第三方面，本发明涉及一种用于关于标签t为接收者使用公共密钥pk对明文m进行群加密以得到密文c的设备。该设备包含处理器，该处理器被配置为：得到签名密钥OTS.sk和验证密钥OTS.vk；通过计算$c_1 = E_1.\mathrm{Encrypt}_{\{pk\}}(m, OTS.vk)$和$c_2 = E_2.\mathrm{Encrypt}_{\{pkOA\}}(f(pk), OTS.vk)$来创建第一加密值$c_1$和第二加密值$c_2$，其中，$E_1$是第一加密算法，$E_2$是第二加密算法，并且f是映射函数；使用所述签名密钥OTS.sk通过计算$s = OTS.\mathrm{Sign}_{\{OTS.sk\}}(c_1, c_2, t)$来生成所述第一加密值$c_1$、所述第二加密值$c_2$和所述标签t上的签名s，其中，

OTS.Sign 是签名算法；并且，输出所述密文 c，其中，所述密文 c 包含所述第一加密值 c_1、所述第二加密值 c_2、所述验证密钥 OTS.vk 以及所述签名 s。

[0053] 在第一优选实施例中，消息 m 满足可公开验证的关系 R。

[0054] 第四方面，本发明涉及一种用于对包含第一加密值 c_1、第二加密值 c_2、验证密钥 OTS.vk 和签名 s 的群加密 c 进行解密的设备，其中，所述签名 s 关于所述第一加密值 c_1、所述第二加密值 c_2 和标签 t。该设备包含处理器，该处理器被配置为：接收所述群加密 c；关于验证密钥 OTS.vk 验证所述签名 s；并且，如果成功地验证了所述签名 s，则使用解密算法 E_1 和所述验证密钥 OTS.vk 对所述第一加密值 c_1 进行解密。

[0055] 在第一优选实施例中，所述处理器还验证所述第一加密值 c_1 的解密满足公开关系 R。

[0056] 第五方面，本发明涉及一种计算机程序产品，在其上存储在被处理器执行时实施第一方面的方法的指令。

[0057] 第六方面，本发明涉及一种计算机程序产品，在其上存储在被处理器执行时实施第二方面的方法的指令。

附图说明

[0058] 现在将参考附图通过非限制性的示例来说明本发明的优选特征，附图中：

[0059] 图 1 示出了根据本发明的优选实施例的群加密系统。

具体实施方式

[0060] 本发明的主要的发明思想是对接收者的公共密钥的别名 (alias) 而不是公共密钥本身进行加密。组管理者 (GM) 在公共数据库 DB 中公布公共密钥、对应的别名的加密和证书。别名是应用在公共密钥上的适当选取的映射函数 f 得到的值。

[0061] 优选地，使用函数 f 的计算是易于实施的，优选地，该函数减小输入的大小，并且两个不同的输入值不应当得到在数据库 DB 中的同一条目。可以将映射函数 f 说成是一种无冲突的 (collision resistant) 散列函数，通过使得组管理者通过例如对新消息进行随机化直至其在数据库中的条目是独一无二的为止来确保该特性。

[0062] 因为别名可以小于公共密钥，所以这可以使得到的架构的大小和成本显著地降低。具体地，不需要将第二加密方案应用与在接收者的公共密钥中的群元素同样多的次数。

[0063] 然而，缺点是：在 Open 过程中，需要开放型权威机构 OA 在数据库 DB 中查找别名的原象（公共密钥）。幸运的是，求助于 Open 过程的情况出现得非常少；亦即，仅在发生纠纷的情况下。

[0064] 本发明的群加密方案使用多个构件块（将稍后在说明书中给出示例）：

[0065] ●加密方案：使用两个加密方案 E_1 和 E_2。必须注意的是，为了本发明的目的，E_1 和 E_2 是弱安全的就足够了，定义如下：

[0066] 弱安全的加密方案是一种未达到“最高的”安全性概念的方案。E_1 的正确的安全性概念是指在选择性标签弱选择密文攻击 (selective tag weak chosen ciphertext attack) 下是不可区分的且匿名的 (IND-st-wCCA 以及 ANO-st-wCCA)。对于 E_2，仅要求 IND-st-wCCA 安全性。

[0067] 这两种安全性概念都使安全性目标 (IND 或 ANO) 与攻击模型 (st-wCCA) 相结合。

[0068] 非形式化地,不可区分性 (IND) 目标表示难以从密文得到关于消息的信息。匿名性 (ANO) 是指难以从密文推断出关于公共密钥的信息。

[0069] 关于攻击模型 st-wCCA,其指代如下场景 :攻击者提前 (在接收质询公共密钥之前) 承诺她希望被质询的标签,而她不被允许发出涉及该质询标签的解密查询。

[0070] ●签名或认证方案 :使用对群元素进行签名的签名方案。一种适合的候选是保留结构的签名方案 S,亦即,一种验证密钥、消息和签名是群元素,并且验证算法包括配对等式验证的谓词 (predicate) 的方案。

[0071] ●一次性签名方案 :使用安全的一次性签名 OTS。

[0072] ●关系 R :使用可公开验证的关系。

[0073] 函数 f :使用可高效计算 (能够在输入的大小的多项式时间内进行评估) 的函数 f。

[0074] 使用这些构件块,可以将该方案构造如下 :

[0075] ● Join。关于输入的公共密钥 pk,GM 除了使用 S.sk(S 是所使用的认证方案) 计算关于 pk 的签名 (或证书)cert 之外还计算 f(pk)。GM 还将 (pk,f(pk),cert) 存储在公共数据库 DB 中。注意,GM 可以进行简单的测量以避免冲突,亦即,避免使用 f 将两个不同的公共密钥 pk 和 pk' 映射成相同的值。将在下文详细说明一种在使用具体函数 f 的情况下的可能的测量。

[0076] ● $\mathrm{Encrypt}_{\{pk\}}(m,t)$。为了关于标签 t 使用公共密钥 pk 为接收者对消息 m(该消息是关于已知的关系 R 的某 x 的证据) 进行加密,实体 :

[0077] ●调用 OTS.keygen 产生一对签名密钥和验证密钥 (OTS.sk,OTS.vk)。

[0078] ●创建 $(c_1,c_2)=(E_1.\mathrm{Encrypt}_{\{pk\}}(m,OTS.vk),E_2.\mathrm{Encrypt}_{\{pkOA\}}(f(pk,OTS.vk))$。应当注意的是,将 OTS.vk 视为标签。

[0079] ●使用 OTS.sk 生成 (c_1,c_2,t) 上的签名 ;$s=OTS.\mathrm{Sign}_{\{OTS.sk\}}(c_1,c_2,t)$。

[0080] ●返回 $c=(c_1,c_2,OTS.vk,s)$ 作为关于 t 的在 pk 下的 m 的群加密。

[0081] ● $\mathrm{Decrypt}_{\{sk\}}(c,t,x)$。

[0082] ●将 c 解析为 $(c_1,c_2,OTS.vk,s)$。

[0083] ●关于 OTS.vk 验证在 (c_1,c_2,t) 上的签名 s。如果 $OTS.\mathrm{Verify}_{\{OTS.vk\}}(s,(c_1,c_2,t))=0$,则返回“失败”,否则计算 $E_1.\mathrm{Decrypt}_{\{sk\}}(c_1,OTS.vk)$:如果它是关于关系 R 的 x 的证据,则返回计算出的值,否则返回“失败”。

[0084] ● $\mathrm{Open}_{\{skOA\}}(c,t)$。

[0085] ●将 c 解析为 $(c_1,c_2,OTS.vk,s)$。

[0086] ●关于 OTS.vk 验证 (c_1,c_2,t) 上的签名 s。如果 $OTS.\mathrm{Verify}_{\{OTS.vk\}}(s,(c_1,c_2,t))=0$,则返回“失败”;否则调用 $E_2.\mathrm{Decrypt}_{\{skOA\}}(c_2,OTS.vk)$,其返回值 F。

[0087] ●关于函数 f 在 DB 中查找值 F 的原象,并返回该搜索的结果。

[0088] ● Prove(c,t,x)。关于标签 t 创建了密文 c 的实体使用用于产生 $c=(c_1,c_2,OTS.vk,s)$ 的随机币提供下面的证明 :

[0089] ●关于标签 OTS.vk 的在某公共密钥 pk 下的隐藏在 c_1 下面的消息的知识的证明,并且证明该消息是关于关系 R 的 x 的证据。

[0090] ●关于标签 OTS.vk 的在密钥 pk_{OA} 下的 c_2 的解密的知识的证明，并且证明该解密是函数 f 关于 pk 的值。

[0091] ●关于 pk 的证书 cert 的知识的证明。

[0092] 某些类的签名和加密方案使这些证明的高效的性能成为可能。

[0093] 对于非交互式的证明，优选地使用接受讨论的证据的知识的高效的非交互式证明的成分（例如，在签名 / 加密方案的情况下的消息或密钥，在函数 f 的情况下的原象，或者在关系R的情况下的证据），诸如兼容Groth-Sahai(Jens Groth、Amit Sahai的《Efficient Non-Interactive Proof Systems for Bilinear Groups》。EUROCRYPT2008 :415-432) 的密码系统。在这个意义上说，可以使用所谓的自同构签名（亦即，验证密钥消息和得到的签名是群元素并且验证算法包括配对乘积等式的逻辑乘积 (conjunction) 的签名方案）以及加密算法对输入实施群运算或配对运算的加密方案（这需要消息、公共密钥和密文是群元素）。函数 f 还对输入实施群运算（在双线性群的情况下实施配对运算）。同样的情形适用于关系 R。

[0094] 同样地，优选使用接受证据的高效的交互式的证明的成分。在这个意义上说，优选使用如下签名方案 :给出消息 M 上的签名 σ，使得能够定义同态函数 φ，使得 $\varphi(S,M)$ 的计算结果为 g(R,vk)，其中，vk 是验证密钥，g 是公共密钥，并且 (S,R) 是从 σ 转换的对，其中 R 不泄露关于 σ 或 M 的信息，S 是签名的“极为重要”的部分 ;隐含的作为基础的转换算法被称为 CONVERT 算法。优选地，还使用接受关于指定的密钥和指定的标签的解密的正确性的高效的证明的加密方案。而且，用于对公共密钥进行加密的方案 (E_2) 应该关于消息是同态的，方案 E_1 应该关于公共密钥和消息这两者是同态的。另外，加密方案 E_1 配有被称为 COMPUTE 算法的算法，该算法对输入关于指定标签 t 的在公共密钥 pk 下的对消息 m 的加密 c_1 生成关于相同的标签 t 的在另一公共密钥 pk' 下的对另一消息 m' 的另一加密 c'_1，使得 c_1 和 c'_1 的组合 (composition) 等于关于标签 t 的在 pk 和 pk' 的组合下的对 m 和 m' 的组合的加密，其中，需将组合理解为施加准备所涉及的元素所属的集合的代数群运算。而且，优选地，函数 f 是同态函数（适用于两个输入的组合的 f 是这两个输入的 f 的值的组合）。同样地，指定实例 x，关系 R 应当能够定义同态函数 F_R 和像 I，使得 F_R(w) = I，其中，w 是与实例 x 相对应的证据。

[0095] 与本发明一起使用的优选签名方案是由 Masayuki Abe、Georg Fuchsbauer、Jens Groth、Kristiyan Haralambiev 和 Miyako Ohkubo 在《Structure-Preserving Signatures and Commitments to Group Elements》(CRYPTO2010 :209-236 页）中提出的方案。

[0096] 与本发明一起使用的优选加密方案是由 David Cash、Eike Kiltz 和 Victor Shoup 在《The Twin Diffie-Hellman Problem and Applications》(Journal of Cryptology22(4) :470-504 页，2009 年）提供的弱安全的基于标签的变型。

[0097] 如果公共密钥是 n 维矢量的群元素，则优选函数 f 如下 :

[0098] ● $f : G^n \rightarrow G$

[0099] $(X_1,\ldots,X_n) \rightarrow X_1^{a_1} \ldots X_n^{a_n}$

[0100] 其中，(G, •) 是具有阶某 d 的群，n 是某整数，并且 a_1，……，a_n 是 Z_d 中的公共元素，亦即，以 d 为模的整数的集合。然后，函数 f 将群 G 中的 n 个元素的元组映射成在群 G

中的元素。

[0101] 关于 f 的这种选择，GM 可以通过对公共密钥进行系统的随机化来避免冲突。更精确地，GM 考虑在指数群 Z_d 中的随机 $r = (r_1, \cdots, r_n)$ 来对 pk 进行随机化以避免函数 f 的冲突；$pk = (X_1, ..., X_n) \leftarrow pk^r = (X_1^{r_1}, ..., X_n^{r_n})$。GM 还公布 r 以使接收者能够相应地更新其私有密钥，这只在 $X_i = g_i^{x_i}$ 时才可行，其中，g_i 是 G 的已知的产生器，x_i 是与 X_i 相对应的私有密钥。对新计算出的公共密钥计算证书，并将证书与密钥及其别名一起存储在 DB 中。

[0102] 最后，优选的关系 R 是 $(m, x, y) \in R \leftrightarrow e(m, P) = e(x, y)$，其中，e 是在域 G×H(G 和 H 是密码双线性群）内的高效配对，P 是 H 中的固定元素。

[0103] 使用公共密钥 pk 为接收者产生了密文 c 的证明者与任何验证者之间的交互式的 Prove 协议以三个阶段进行：承诺、质询和响应。在承诺阶段，证明者在输入组管理者的公共密钥 S.pk、公共 pk 和对应的证书时运行 CONVERT 算法以得到对 (S,R)。证明者还在输入 c_1 时运行 COMPUTE 算法并得到元组 (pk',m',c'_1)。接下来，证明者计算 F' = f(pk')、I'_R = F_R(m') 以及 $I' = \varphi(S', pk')$。最后，证明者计算作为在公共密钥 pk_{OA} 下的 F' 的加密的 c'_2。证明者将元组 (R,I',I'_R,c'_2) 发送给验证者。在质询阶段，在接收该元组时，验证者随机地选择整数 b 并且计算 I = g(R,S.pk) 和 I_R，使得 F_R(m) = I_R。验证者将质询 b 发送给证明者。在接收该质询时，证明者计算并发送值 z_S、z_{pk}、z_m 以及 z_F，其中，z_{pk} 是 pk' 和 pk^b 的组合，z_S 是 S' 和 S^b 的组合，z_m 是 m' 和 m^b 的组合，并且 z_F 是 F' 和 F^b 的组合。最后，证明者证明如下知识：(PoK1)c'_1 和 c_1^b 的组合是关于标签 t 的在公共密钥 z_{pk} 下的 z_m 的加密；以及，(PoK2)c'_2 和 c_2^b 的组合是关于标签 t 的在公共密钥 pk_{OA} 下的 z_F 的加密。在协议的结束，如果 (1) $\varphi(z_S, z_{pk})$ 等于 I' 和 I^b 的组合，(2)F_R(z_m) 是 I'_R 和 I^b_R 的组合，(3)f(z_{pk}) 等于 z_F，并且 (4)PoK1 和 PoK2 是有效的，则验证者接受。技术人员将观察到，当使用优选的加密方案进行实例化时，知识证明 PoK1 和 PoK2 归结为示出离散对数的相等性；其高效的方法可以根据 Claus P.Schnorr 的《Efficient signature generationby smart cards》(Journal of Cryptolopy,4(3)：161-179 页，1991 年）的开创性工作得到。另外参见 Jan Camenisch 的《Group signature schemes and payment systems based on the discrete logarithm problem》(博士论文，信息安全和密码学 ETH 系列第 2 卷，Hartung-Gorre Verlag，1998 年 (ISBN3-89649-286-1))。

[0104] 图 1 示出用于根据本发明的优选实施例的群加密的系统 100。为了便于说明和理解，省略了在该系统中的设备之间的连接。

[0105] 系统 100 包含发送器 110 和接收器 120，其中每一个均包含：被配置为与其他设备通信的至少一个接口单元 111、121、至少一个处理器（“处理器”）112、122 以及被配置为存储数据（如累加器和中间计算结果）的至少一个存储器 113、123。系统 100 还包含组管理者 130、数据库 140、第三方 150 以及开放型权威机构 160；虽然为了清楚而未示出，但是这些设备中的每一个均包含诸如处理器和存储器这样的必要的硬件。

[0106] 发送器 110 的处理器 112 被配置为实施本发明的群加密方案的 Encrypt 和 Prove 部分，接收器 120 的处理器 122 适用于对接收的群加密进行解密，亦即，实施 Decrypt。组管理者 130 被配置为实施 Join 部分，从而将数据存储在数据库 140 中。第三方 150 被配

置为验证由发送器提供的证明，并且开放型权威机构 160 被配置为实施群加密方案的 Open 部分。诸如 CD-ROM 或 DVD 这样的第一计算机程序产品 114 包含存储的指令，该指令在由发送器 110 的处理器 112 执行时实施根据本发明的 Encryption 和 Prove。第二计算机程序产品 124 包含存储的指令，该指令在由接收器 120 的处理器 122 执行时实施根据本发明的 Decrypt。

[0107] 技术人员将意识到，与现有技术方案相比，本发明的群加密方案可以使大小和成本显著地降低。例如，如果本发明的 GE 方案使用以下进行实例化，则得到 0.4kB 的密文（而不是现有技术中的 1.25kB 或 2.5kB）：

[0108] ●由 MasaYuki Abe、Georg Fuchsbauer、Jens Groth、Kristiyan Haralambiev 和 Miyako Ohkubo 在《Structure-Preserving Signatures and Commitments to Group Elements》(CRYPTO2010 :209-236 页）中提出的签名方案；

[0109] ●由 Jens Groth、Rafail Ostrovsky 和 Amit Sahai 在《Non-interactive Zaps and New Techniques for NIZK》(CRYPTO2006 :97-111 页）中提供的一次性签名；

[0110] ●由 David Cash、Eike Kiltz 和 Victor Shoup 在《The Twin Diffie-Hellman Problem and Applications》(Journal of Cryptology22(4) :470-504 页，2009 年）中提供的弱安全的基于标签的变型以对 E_1 和 E_2 进行实例化。

[0111] 另外，证明更短并且可以与验证者交互或不交互地执行（对于交互式的证明为 1kB，对于非交互式的证明为 2kB)，允许后者选择实施成本低的、交互式的证明或者成本高的、非交互式的证明。另外，该证明的验证需要 325 个配对计算（与现有技术中的 3895 个配对计算相比）。

[0112] 如先前所述，本发明的 GE 方案具有如下缺点 :在每个 Open 过程中访问数据库 DB 以便找到公共密钥的别名的原像。幸运的是，求助于 Open 的情况仅在出现冲突时才出现，因此非常少见。

[0113] 虽然在 GE 的背景下说明了本说明，但是其范围并不限制于这种密码方案。涉及存在于（在线）公共 DB 的（长）消息的加密的任何密码方案均可以同样地从本发明中受益。应用本发明，（短）别名将与 DB 的每条消息相关联并且被添加到 DB 中。然后，将用对别名的加密替换对消息的加密，因此减小了密文的大小。在解密时，将恢复别名并且还将使用向在线 DB 的请求来恢复相关联的消息。

[0114] 可以独立地或者以任何适当的组合提供在说明书、（当合适时）权利要求书和附图中公开的每个特征。被说明为以硬件实现的特征还可以实现为软件，反之亦然。在权利要求书中出现的标号仅用于示例，并且不应当对权利要求的范围起到限制效果。

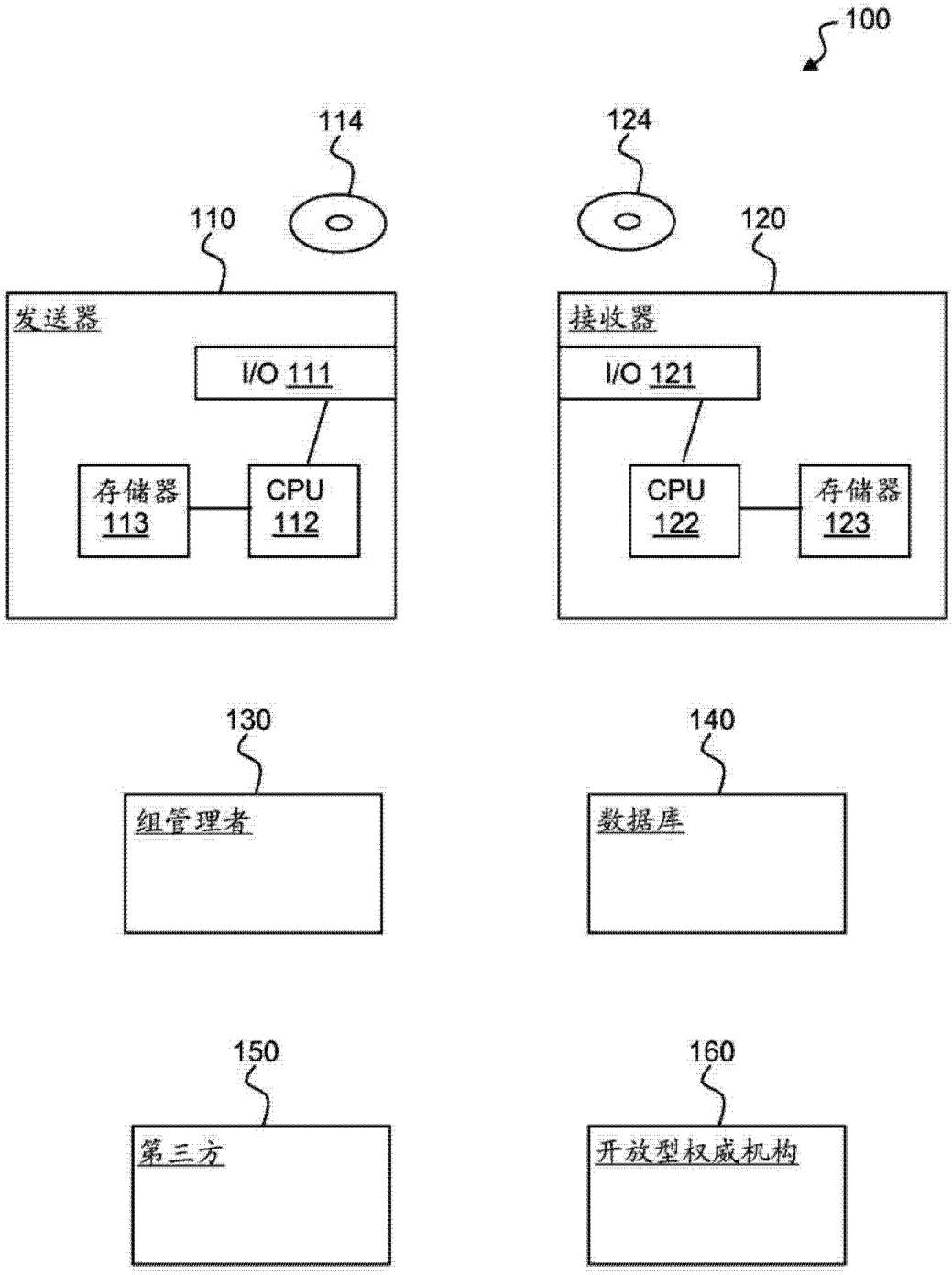

100
114
124
110
120
发送器
I/O 111
存储器
113
CPU
112
接收器
I/O 121
CPU
122
存储器
123
130
组管理者
140
数据库
150
第三方
160
开放型权威机构

图 1