标准模型下群签名的批验证协议 *

王少辉[1], 王高丽[2]
(1. 南京邮电大学 计算机学院, 南京 210046; 2. 东华大学 计算机学院, 上海 201620)

摘 要: 首次对标准模型下群签名的批验证协议进行了研究。利用小指数测试技术和双线性对映射的特殊性质,为目前效率较高的两个群签名方案按照多人签署相同消息和不同消息这两种情况,分别设计了相应的批验证协议。相较于单独验证,认证效率大大提高。

关键词: 数字签名; 群签名; 批验证; 小指数算法; 标准模型
中图分类号: TP309 **文献标志码**: A **文章编号**: 1001-3695(2010)04-1461-03
doi:10.3969/j.issn.1001-3695.2010.04.071

Batch verification of group signature without random oracles

WANG Shao-hui[1], WANG Gao-li[2]
(1. *College of Computer, Nanjing University of Post & Telecommunication, Nanjing* 210046, *China*; 2. *School of Computer Science & Technology, Donghua University, Shanghai* 201620, *China*)

Abstract: This paper conducted a research on the batch verification protocols of the group signature under the standard model at the first time. Using the small exponentiation technique and the special property of the bilinear maps, according to signing the same message or not, presented the batch verification algorithms for two efficient group signature scheme without random oracle. Compared with the original verification algorithm, improved the efficiency a lot.

Key words: digital signature; group signature; batch verification; small exponentiation algorithm; standard model

数字签名及其验证是网络安全中的重要问题,验证一个签名的有效性一般需要大的模指数运算,而大的模指数运算是非常耗时的。为了提高效率,1994 年,Naccache 等人[1]首先提出了批验证协议的概念。其基本思想是将多个签名(可能来自不同的签名者)放在一起,形成一个批,对该批进行验证,如果该批通过验证,则接受该批中的所有签名,否则,拒绝批中所有签名。对批验证的研究主要集中在两个方面,即为具体的签名方案设计安全的批验证协议[2~7]与一般性的批验证协议设计模型[8]。

群签名是由 Chaum 等人[9]在 1991 年提出的。与普通的签名体制相比较,群签名可提供签名者的匿名性,即一个验证者只能辨别出签名是某个群中的一名成员所签,同时在必要时又可以通过群管理员确定签名者的身份。群签名的这些突出特点,使得它适合许多特殊的应用环境,如投票系统、竞标系统等。2007 年,Jan Camenisch 等人[10]提出了短签名的批验证问题,并提出为群签名设计批验证协议这一公开问题;最近,Anna Lisa Ferrara 等人[11]对短签名批验证的实用性进行了较为详尽的研究,并且在随机预言(random oracle)模型下对短群签名的批验证问题进行了研究和设计。

本文利用文献[8]提出的小指数测试方法,对标准模型下的两个有代表性的短群签名方案[12](记为 BW 方案)和文献[13](记为 LCSL 方案)的批验证协议进行了研究,这两个方案都利用了双线性映射这个工具。按照不同签名者签署相同消息和不同消息这两种情形,分别给出了批验证协议。

1 背景知识

1.1 双线性配对的基本概念

定义 1 双线性对。设 G_1 和 G_2 是阶为 n 的有限循环乘法群,双线性对是指满足以下特性的一个映射 $e: G_1 \times G_1 \to G_2$。

1)双线性 $e(u^a, v^b) = e(u,v)^{ab}, u,v \in G_1, a,b \in Z$。

2)非退化性 存在 $g \in G_1$,使得 $e(g,g)$ 在 G_2 的阶为 n。

3)可计算性 对于所有的 $u,v \in G_1$,存在有效的算法来计算 $e(u,v)$。

这样的群能在有限域上的超奇异椭圆曲线上找到,而双线性对可由 Weil 对或者 Tate 对获得。通常基于双线性对的方案考虑的群的阶是素数,而在文献[12,13]中标准模型下的群签名方案中,群的阶不是素数,而是一个因数为两个素数的合数,这主要用于签名者的追踪。

1.2 Bellare、Garay 和 Rabin 测试技术

批验证的概念提出之后,涌现了大量的论文对常用签名算法的批验证进行设计,但是这些方案基本上都是不安全的。在文献[8]中,Bellare、Garay 和 Rabin 对于幂指数运算的签名算法的批验证方式进行了详细研究,他们给出了批验证的具体概念和安全定义,当然他们的定义中要求签名是来自同一个签名者,并且提出了三种技术为已有的签名方案设计批验证协议,分别为随机子集合测试、小指数测试和水桶测试。文献[11]将批验证的概念扩展到来自不同签名者的一般情况,并将小指数测试应用到以双线性映射为设计基础的签名方案中。下面

收稿日期: 2009-08-27; **修回日期**: 2009-10-09 **基金项目**: 国家自然科学基金资助项目(60903181);南京邮电大学校基金资助项目(NY208072)

作者简介:王少辉(1977-),男,山东潍坊人,博士,主要研究方向为公钥密码学、密码分析(wangshaohui@njupt.edu.cn);王高丽(1981-),女,讲师,博士,主要研究方向为公钥密码学、分组密码的分析与设计.

直接给出文献[8]中关于小指数测试的定义。

定义 2　小指数测试。设 g 是一群的生成元。要对 n 个对 (x_i, y_i) 批验证其是否满足等式 $y_i = g^{x_i}$ $(i = 1, \cdots, n)$ 小指数测试采用如下方法：

a)随机选取长度为 l 比特的随机数 $s_1, \cdots, s_n$；

b)计算 $x = \sum_{i=1}^{n} x_i s_i$，$y = \prod_{i=1}^{n} y_i^{s_i}$；

c)验证如果 $g^x = y$，那么就接收整个等式；否则丢弃。

采用如上方法接收一个错误对的概率是 2^{-l}。

1.3　随机预言模型下群签名的批验证协议

一个群签名方案，通常包含了系统建立、用户加入、签名、认证和打开五个算法。简单地说，系统建立和用户加入算法主要生成系统的相关安全参数和用户的私钥等信息，签名算法是用户利用自身的私钥对消息签名；认证算法则是签名的验证过程，群签名的效果是要求能够验证消息来自某个群，但是并不能具体到某个签名者；打开算法是指当存在异议时，群管理者通过该算法追踪到签名者。

文献[11]给出了随机预言模型下群签名方案的批验证协议，他们为了能够对群签名实现批验证，给出了设计该类型方案的一般化方法。其中，他们的方法适用于基于双线性对映射的群签名方案。其方法主要分为四步，本文只对他们的想法作简要的说明，具体如何运用该方法，不再赘述：a)因为直接为已有的随机预言模型下的群签名方案实行批验证的处理是非常困难的，所以首先要对群签名方案的认证算法进行改进；b)运用 Bellare 等人提出的小指数测试方法对新的验证算法实行批验证处理；c)根据双线性对的性质，采用一些技巧对验证步骤进行优化；d)如果批验证失败，采用分而治之的方法去确认坏的签名。

2　BW 群签名方案的批验证协议

在本章中，采用小指数测试方法，结合双线性对的一些特殊性质，针对 BW 方案给出了第一个标准模型下的批验证群签名方案。BW 方案的安全性是基于计算 Diffie-Hellman 问题和子群判断问题。

在随机预言模型下，群签名的批验证协议通常需要对原协议的验证算法作修改，而在标准模型下可以看到并不需要对原群签名方案的签名和验证算法进行修改。下面首先给出 BW 群签名方案的描述，在此省略了打开算法，然后分别按照多个签名者签署同一消息和不同消息给出相应的批验证协议。

2.1　BW 群签名方案

1)系统建立　首先选择 $n = pq$。其中：p 和 q 是两个大的素数，建立一个阶为 n 的循环群 G，它的两个子群 G_p 和 G_q 的阶分别为 p 和 q。g 是群 G 的生成元，h 是群 G_q 的生成元，随机选择 $\alpha, w \in Z_n$，计算 $A = e(g, g)^{\alpha} \in G_T$，$\Omega = g^w \in G$。选择 $m + 2$ 个随机生成元 $u, v', v_1, \cdots, v_m$。那么系统的公共参数包括 $PP = (g, h, u, v', v_1, \cdots, v_m, \Omega, A)$，而主加入密钥为 $MK = (g^{\alpha}, w)$。

2)加入　对于需要加入的用户，群管理者会首先选择一个随机数 S_{ID}，满足 $w + S_{ID} \neq 0$，为其生成签名密钥 $K_{ID} = (K_1, K_2, K_3) = ((g^{\alpha})^{\frac{1}{\omega + S_{ID}}}, g^{S_{ID}}, u^{S_{ID}})$。

3)签名　当有消息 $M = (\mu_1, \cdots, \mu_m) \in \{0,1\}^m$ 进行签名时，用户首先使用加入群时获得的签名密钥 K_{ID} 来产生一个两级的签名，随机选择 s，将 $v' \prod_{i=1}^{m} v_i^{\mu_i}$ 记为 $F(m)$。计算：

$$\theta = (\theta_1, \theta_2, \theta_3, \theta_4) = (K_1, K_2, K_3 \times (F(m))^s, g^{-s})$$

接着 θ 必须进行盲化处理，以获得匿名性和不可联系性，签名者接着选择随机指数 $t_1, t_2, t_3, t_4 \in Z_n$，再计算：

$$\sigma_1 = \theta_1 \times h^{t_1}, \sigma_2 = \theta_2 \times h^{t_2}, \sigma_3 = \theta_3 \times h^{t_3}, \sigma_4 = \theta_4 \times h^{t_4}$$

然后产生两个元素：

$$\pi_1 = h^{t_1 t_2} \times (\theta_1)^{t_2} \times (\theta_2 \Omega)^{t_1}, \pi_2 = u^{t_2} \times g^{-t_3} \times (F(m))^{t_4}$$

最后消息 M 的签名是：

$$\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \pi_1, \pi_2) \in G^6$$

4)验证　验证者首先计算 $T_1 = A^{-1} \times e(\sigma_1, \sigma_2 \Omega)$ 和 $T_2 = e(\sigma_2, u) \times e(\sigma_3, g)^{-1} \times e(\sigma_4, F(m))^{-1}$，然后检验公式 $T_1 = e(h, \pi_1)$ 和 $T_2 = e(h, \pi_2)$ 是否成立，若成立，那么签名是合法的，否则无效。

2.2　*BW* 群签名方案的批验证协议

在基于双线性对的方案中，双线性对运算是最为耗时的，所以在下面的性能分析中，笔者只比较对运算的数目。上述的群签名认证算法，一共需要 6 个对运算，如果是验证 n 个签名，一共需要 $6n$ 个对的运算，计算还是比较复杂和费时的。对上述群签名的批验证处理分两种情况：一种是多个签名者对同一消息签名的情况；另一种是多个签名者对不同消息签名的情况，在实际的群签名中，后者同一个签名者签署多个签名是没有区别的。可以看出，群签名的验证算法实际上就是通过验证下边的两个等式是否成立达到的：

$$A^{-1} \times e(\sigma_1, \sigma_2 \Omega) = e(h, \pi_1)$$

$$e(\sigma_2, u) \times e(\sigma_3, g)^{-1} \times e(\sigma_4, F(m))^{-1} = e(h, \pi_2)$$

1)多个签名者签署同一消息的批认证处理

假设 n 个签名者对同一消息 m 进行签名为 $\sigma_i = (\sigma_{1i}, \sigma_{2i}, \sigma_{3i}, \sigma_{4i}, \pi_{1i}, \pi_{2i})$，$i = 1, \cdots, n$，应用小指数测试法，首先选择长度为 l 比特的随机数 $s_1, \cdots, s_n$，对签名进行如下验证：

$$\prod_{i=1}^{n} (A^{-1})^{s_i} \times e^{s_i}(\sigma_{1i}, \sigma_{2i} \Omega) = \prod_{i=1}^{n} e^{s_i}(h, \pi_{1i}) \tag{1}$$

$$\prod_{i=1}^{n} (e^{s_i}(\sigma_{2i}, u) \times e^{s_i}(\sigma_{3i}, g)^{-1} \times e^{s_i}(\sigma_{4i}, F(m))^{-1} = \prod_{i=1}^{n} e^{s_i}(h, \pi_{2i}) \tag{2}$$

利用双线性对对外的指数可以移到内部的性质，对式(1)和(2)的一些项进行处理，可以得到：

a) $\prod_{i=1}^{n} e(h, \pi_{1i}^{s_i}) \times \prod_{i=1}^{n} A^{s_i} = e(h, \prod_{i=1}^{n} \pi_{1i}^{s_i}) \times \prod_{i=1}^{n} A^{s_i}$

b) $\prod_{i=1}^{n} e(h, \pi_{2i})^{s_i} \times \prod_{i=1}^{n} e(\sigma_{4i}, F(m))^{s_i} \times \prod_{i=1}^{n} e(\sigma_{3i}, g)^{s_i} = e(h, \prod_{i=1}^{n} \pi_{2i}^{s_i}) \times e(\prod_{i=1}^{n} \sigma_{4i}^{s_i}, F(m)) \times e(\prod_{i=1}^{n} \sigma_{3i}^{s_i}, g)$

c) $\prod_{i=1}^{n} e(\sigma_{2i}, u)^{s_i} = e(\prod_{i=1}^{n} \sigma_{2i}^{s_i}, u)$

这样式(1)和(2)的验证可以转换为

$$e(h, \prod_{i=1}^{n} \pi_{1i}^{s_i}) \times \prod_{i=1}^{n} A^{s_i} = \prod_{i=1}^{n} e(\sigma_{1i}, \sigma_{2i} \Omega)^{s_i} \tag{3}$$

$$e(h, \prod_{i=1}^{n} \pi_{2i}^{s_i}) \times e(\prod_{i=1}^{n} \sigma_{4i}^{s_i}, F(m)) e(\prod_{i=1}^{n} \sigma_{3i}^{s_i}, g) = e(\prod_{i=1}^{n} \sigma_{2i}^{s_i}, u) \tag{4}$$

通过如上的处理，原来式(1)需要 $2n$ 个双线性对运算，式(2)需要 $4n$ 个对运算，而式(3)和(4)分别需要 $n + 1$ 个对运算和 4 个对运算。

2)多个签名者签署不同消息的批认证处理

此时 $F(m)$ 为不同的消息 $F(m_i)$，可以看出，验证式(1)不变，而式(2)改变如下，需要 $n + 3$ 个对运算：

$$e(h, \prod_{i=1}^{n} \pi_{2i}^{s_i}) \times \prod_{i=1}^{n} e(\sigma_{4i}, F(m_i))^{s_i} e(\prod_{i=1}^{n} \sigma_{3i}^{s_i}, g) = e(\prod_{i=1}^{n} \sigma_{2i}^{s_i}, u) \tag{5}$$

3　*LCSL* 群签名方案的批验证协议

LCSL 群签名方案的安全性基于 l-强 Diffie-Hellman 假设和

子群判断假设,相较于 BW 方案,LCSL 在签名长度和运算量上更优。

3.1 *LCSL* 群签名方案

1)系统建立　在此阶段主要是群和群上的双线性映射的建立过程。选择 $n=pq$,其中,p 和 q 是两个大的素数,建立一个阶为 n 的循环群 G,它的两个子群 G_p 和 G_q 的阶分别为 p 和 q。g 是群 G 的生成元,h 是群 G_q 的生成元,随机选择 $a\in Z_n^*$,并定义 $A=g^{\alpha}\in G$,另外选择一映射到 Z_n 的安全 hash 函数 H。那么系统的公共参数包括 $PP=(g,h,A)$,而主加入密钥为 $MK=\alpha$。

2)加入　对于需要加入的用户,群管理者会为他选择一个随机数 S_{ID},满足 $\alpha+S_{ID}\neq 0$,为其生成签名密钥 $K_{ID}=(K_1,K_2)=(S_{ID},g^{\frac{1}{\alpha+S_{ID}}})$。

3)签名　当有消息 $M\in\{0,1\}^m$ 进行签名时,用户首先使用加入群时获得的签名密钥 K_{ID} 来产生一个两级的签名:

$$\theta=(\theta_1,\theta_2,\theta_3)=(g^{K_1},K_2,g^{\frac{1}{K_1+H(M)}})$$

接着 θ 必须进行盲化处理,以获得匿名性和不可联系性,签名者接着选择随机指数 $t_1,t_2,t_3\in Z_n$,再计算:

$$\sigma_1=\theta_1\times h^{t_1},\sigma_2=\theta_2\times h^{t_2},\sigma_3=\theta_3\times h^{t_3}$$

然后产生两个元素:

$$\pi_1=h^{t_1t_2}\times(A\theta_1)^{t_2}\times\theta_2^{t_1},\pi_2=h^{t_1t_3}\times\theta_3^{t_1}\times(g^{H(M)}\theta_1)^{t_3}$$

最后消息 M 的签名是:

$$\sigma=(\sigma_1,\sigma_2,\sigma_3,\pi_1,\pi_2)\in G^5$$

4)验证　验证者首先计算 $T_1=e(g,g)^{-1}\times e(\sigma_1A,\sigma_2)$ 和 $T_2=e(\sigma_1g^{H(M)},\sigma_3)\times e(g,g)^{-1}$,然后检验式 $T_1=e(h,\pi_1)$ 和 $T_2=e(h,\pi_2)$ 是否成立,如果成立,那么签名是合法的,否则无效。

3.2 *LCSL* 群签名方案的批验证协议

类似 3.1 节的讨论,群签名的验证算法实际上就是通过验证下边的两个等式成立而获得的:

$$e(g,g)^{-1}\times e(\sigma_1A,\sigma_2)=e(h,\pi_1)$$

$$e(\sigma_1g^{H(M)},\sigma_3)\times e(g,g)^{-1}=e(h,\pi_2)$$

1)多个签名者签署同一消息的批认证处理

假设 n 个签名者对同一消息 m 进行签名为 $\sigma_i=(\sigma_{1i},\sigma_{2i},\sigma_{3i},\pi_{1i},\pi_{2i})$,$i=1,\cdots,n$,应用小指数测试法,首先选择长度为 l 比特的随机数 $s_1,\cdots,s_n$,对签名进行如下验证:

$$\prod_{i=1}^{n}e^{s_i}(g,g)^{-1}\times e^{s_i}(\sigma_{1i}A,\sigma_{2i})=\prod_{i=1}^{n}e^{s_i}(h,\pi_{1i})\tag{6}$$

$$\prod_{i=1}^{n}e^{s_i}(\sigma_{1i}g^{H(M)},\sigma_{3i})\times e^{s_i}(g,g)^{-1}=\prod_{i=1}^{n}e^{s_i}(h,\pi_{2i})\tag{7}$$

利用双线性对的性质,对式(6)和(7)进行如下处理:

a)　$\prod_{i=1}^{n}e^{s_i}(h,\pi_{1i})\times\prod_{i=1}^{n}e^{s_i}(g,g)=\prod_{i=1}^{n}e(h,\pi_{1i}^{s_i})\times\prod_{i=1}^{n}e(g,g^{s_i})=e(h,\prod_{i=1}^{n}\pi_{1i}^{s_i})\times e(g,\prod_{i=1}^{n}g^{s_i})$

b)　$\prod_{i=1}^{n}e(h,\pi_{2i})^{s_i}\times\prod_{i=1}^{n}e^{s_i}(g,g)=\prod_{i=1}^{n}e(h,\pi_{2i}^{s_i})\times\prod_{i=1}^{n}e(g,g^{s_i})=e(h,\prod_{i=1}^{n}\pi_{2i}^{s_i})\times e(g,\prod_{i=1}^{n}g^{s_i})$

这样式(6)和(7)的验证可以转换为

$$e(h,\prod_{i=1}^{n}\pi_{1i}^{s_i})\times e(g,\prod_{i=1}^{n}g^{s_i})=\prod_{i=1}^{n}e^{s_i}(\sigma_{1i}A,\sigma_{2i})\tag{8}$$

$$e(h,\prod_{i=1}^{n}\pi_{2i}^{s_i})\times e(g,\prod_{i=1}^{n}g^{s_i})=\prod_{i=1}^{n}e(\sigma_{1i}g^{H(M)},\sigma_{3i})^{s_i}\tag{9}$$

在实际应用中,$e(g,g)^{-1}$ 可以作为预计算的结果加以公布,所以验证式(6)需要 $2n$ 个双线性对运算,而式(7)同样需要 $2n$ 个双线性对运算。批处理之后,式(8)和(9)均需要 $n+2$ 个对运算。

2)多个签名者签署不同消息的批认证处理

此时 $F(m)$ 为不同的消息 $F(m_i)$,可以看出,验证式(8)不变,而式(9)改变如下,仍需要 $n+2$ 个双线性对运算:

$$e(h,\prod_{i=1}^{n}\pi_{2i}^{s_i})\times e(g,\prod_{i=1}^{n}g^{s_i})=\prod_{i=1}^{n}e(\sigma_{1i}g^{H(M_i)},\sigma_{3i})^{s_i}$$

4　结束语

本文基于小指数测试方法,利用双线性对的特殊性质,首次研究了标准模型下群签名的批验证算法,并对目前标准模型下较高效的两个群签名方案的批验证协议进行了研究:

a)对于 BW 群签名方案,如果验证 n 个群签名,逐个验证的情况需要 $6n$ 个双线性对运算,而通过批验证方案,当对同一消息签名时,需要 $n+5$ 个对运算,当对不同的消息签名时,需要 $2n+4$ 个对运算;b)对于 LCSL 群签名方法,如果验证 n 个群签名,逐个验证的情况需要 $4n$ 个双线性对运算,而通过批验证方案,当对同一消息签名时,需要 $2n+4$ 个对运算,当对不同的消息签名时,需要 $2n+4$ 个对运算。

虽然采用批验证协议可以大大提高验证的效率,但是所需要的双线性对运算还是与签名的个数相关,设计安全的群签名方案,使得相应的批验证协议运算个数不依赖于签名个数是一个值得思考的问题。

参考文献:

[1] NACCACHE D, MRAIHI D, VAUDENAY S, *et al*. Can DSA be improved: complexity trade-offs with the digital signature standard [C]//Proc of Advances in Cryptology-EuroCrypt'94, LNCS 950. 1994:77-85.

[2] FIAT A. Batch RSA[C]//Proc of Advances in Cryptology-Crypto'89, LNCS 435. 1989:175-185.

[3] RAIHI D M, NACCACHE D. Batch exponentiation:a fast DLP based signature generation strategy[C]//Proc of the 3rd ACM Conference on Computer and Communications Security. 1996:58-61.

[4] LIM C H, LEE P J. Security of interactive DSA batch verification [J]. Electronics Letters, 1994, 30(19):1592-1593.

[5] HARN L. Batch verifying multiple DSA-type digital signatures [J]. Electronics Letters, 1998, 34(9):870-871.

[6] 吴秋新, 杨义先, 胡正名. 数字签名批验证的新方法[J]. 计算机工程与应用, 2000, 36(12):28-30.

[7] 张青坡,王立鹏, 陈鲁生. 一种全新的批验证协议[J]. 计算机工程与应用, 2005,41(4):127-130.

[8] BELLARE M, GARAY J A, RABIN T. Fast batch verification for modular exponentiation and digital signatures[C]//Proc of Advances in Cryptology-EuroCrypt'98, LNCS 1403. 1998:236-250.

[9] CHAUM D, HEYST E van. Group signatures[C]//Proc of Advances in Cryptology-EuroCrypt'91, LNCS 547. [S.l.]:Springer-Verlag, 1991:257-265.

[10] CAMENISCH J, HOHENBERGER S, PEDERSEN M. Batch verification of short signatures[C]//Proc of Advances in Cryptology-EuroCrypt'07, LNCS 4515. 2007:246-263.

[11] FERRARA A L,GREEN M, HOHENBERGE S, *et al*. On the practicality of short signatures batch verification, Report 2008/015[R]. [S.l.]:Cryptology ePrint Archive, 2008.

[12] XAXIER B, BRENT W. Full-domain subgroup hiding and constant-size group signatures[C]//Proc of PKC 2007, LNCS 4450. 2007:1-15.

[13] LIANG Xiao-hui, GAO Zhen-fu, SHAO Jun, *et al*. Short group signature without random oracles[C]//Proc of ICICS 2007. 2007:69-82.