对一种 $(t, N-2)$ 弹性 Mix Net 的密码学分析

李龙海[1,2]，付少锋[1]，肖国镇[2]

(1. 西安电子科技大学 计算机学院，陕西 西安 710071；
2. 西安电子科技大学 综合业务网理论及关键技术国家重点实验室，陕西 西安 710071)

摘要：分析了 Gao 等人提出的 $(t, N-2)$ 弹性 Mix Net 方案，发现存在严重安全漏洞. 主动攻击者利用 Elgamal 算法的可展性构造具有相关性的密文组，然后通过观察对应明文组的相关性获得输入与输出的对应关系，最终破坏 Mix Net 的秘密性. 两个不同服务器组中的恶意服务器可以相互勾结利用共谋攻击使 Mix Net 输出错误结果，并以不可忽略的概率逃过验证协议的检验. 分析结果说明 Gao 的方案不满足 $(t, N-2)$ 弹性，且基于该 Mix Net 的电子投票应用也是不安全的.

关键词：匿名通信；Mix Net；共谋攻击

中图分类号：TN918 **文献标识码**：A **文章编号**：1001-2400(2007)06-0926-04

Cryptanalysis of a $(t, N-2)$-resilient Mix Net

LI Long-hai[1,2], *FU Shao-feng*[1], *XIAO Guo-zhen*[2]

(1. School of Computer Science and Technology, Xidian Univ., Xi'an 710071, China; 2. State Key Lab. of Integrated Service Networks, Xidian Univ., Xi'an 710071, China)

Abstract: We analysed Gao et al.'s $(t, N-2)$-resilient Mix Net scheme and found some serious security flaws in their design. In order to break Mix Net's privacy, an active attacker can construct a list of ciphertexts with some relativity by utilizing the malleability of the ElGamal encryption scheme, and then observe the corresponding relativity of plaintexts to get the relationship between input and output elements. The malicious servers from two different groups can initiate collusion attacks proposed by this paper to make the Mix Net system output wrong and cheat the verifying protocol with non-negligible probability of success. The result of analysis shows that Gao et al.'s scheme does not satisfy $(t, N-2)$-resilience and that the electronic voting application based on their Mix Net is also insecure.

Key Words: anonymous communication; Mix Net; collusion attacks

为了在网络通信中实现发送者匿名性，Chaum 在 1981 年[1]提出了一种称为 Mix Net 的密码学工具. Mix Net 的功能可以概括为：输入一组密文 $(c_1, c_2, \cdots, c_N)$，输出解密后所得明文组 $(m_1, m_2, \cdots, m_N)$ 的一个随机置换. 如果该置换保密并且加密算法是语义安全的，则任何第三方都无法确定输入与输出之间的对应关系. Mix Net 已经被广泛应用于匿名电子邮件[1]、匿名通信[2]、电子投票[3,4]、电子支付[5]、匿名 Web 浏览等需要保护用户隐私的互联网应用系统中.

Mix Net 必须满足的安全特性包括：(1)秘密性，即第三方发现输入与输出之间对应关系的概率不优于随机猜测；(2)正确性，即输入密文所对应的明文必须全部出现在输出中；(3)可验证性，即输出错误结果而不被发现的概率是可以忽略的. 具体实现时一般将多个 Mix 服务器串联在一起并依次对输入密文组进行盲化和置换，最后所有服务器共同对盲化后的密文进行门限解密并输出明文. 在这种结构中，假定最多有 $t-1$ 个恶意服务器和 $N-2$ 个恶意用户，如果能够同时满足秘密性、正确性和可验证性，则称系统是 $(t, N-2)$ 弹性的[6].

收稿日期：2007-06-16

基金项目：国家自然科学基金面上项目资助(60473028)

作者简介：李龙海(1976-)，男，讲师，西安电子科技大学博士研究生.

在 Mix Net 中保证正确性和可验证性的方法基本可以分为两类. 一类是采用零知识证明技术,如文献[3,7～9]提出的一些可公开验证的方案;另一类是采用分布式冗余计算的方法,并依靠内部诚实服务器对结果进行检验,如文献[6,10,11]的方案. 最近,Gao 等在文献[12]中提出了一种综合利用两种方法的 Mix Net,并声称该方案是 $(t,N-2)$ 弹性的且在效率上优于先前的多种方案. 以该研究为基础,Gao 等随后在文献[4]中设计了一种基于该 Mix Net 和盲签名的电子投票方案. 笔者对 Gao 等提出的 Mix Net 进行了安全性分析,指出了方案中的几处严重漏洞,并构造了多种攻击方法破坏其秘密性、正确性和可验证性以证明该系统无法满足 $(t,N-2)$ 弹性. 分析结果同时表明 Gao 的电子投票方案[4]也是不安全的.

1　文献[12]的 $(t, N-2)$ 弹性 Mix Net

1.1　系统的建立

设 p,q 是两个大素数,且满足 $p=2q+1$、在群 Z_p^* 上离散对数问题难解. 在 Z_p^* 中选取 q 阶元素 g,并设由 g 生成的子群为 G_q. 参与协议的包括 N 个用户和 $2t$ 个 Mix 服务器,其中最多有 $N-2$ 个恶意用户和 $t-1$ 个恶意服务器. 另外系统中还有若干个备选的 Mix 服务器. $2t$ 个服务器利用 Shamir$(t, 2t)$ 门限方案共享秘密钥 $x\in Z_q$,并以$(p, q, y=g^x)$ 作为系统的公开钥. 每个服务器 i 公布 $y_i=g^{x_i}$,x_i 表示 i 所掌握的关于 x 的秘密份额. Mix Net 的不同参与者之间通过公布栏交换信息. 这里将公布栏视为具有认证功能的可靠的广播信道.

1.2　主协议描述

假定在公布栏中公布有一组需要解密的密文 $c=(c_1, c_2, \cdots, c_N)$,其中 $c_i=(a_i, b_i)=(g^{r_i}, m_i y^{r_i})$,$r_i\in_U Z_q$,$m_i$ 是 c_i 所对应的明文,$i=1, 2, \cdots, N$. 系统的 Mix 服务器组共同执行以下协议以获得明文组 $(m_1, m_2, \cdots, m_N)$ 的一个随机置换:

(1) 把 $2t$ 个服务器随机分为 group_1 和 group_2 两组,每组包含 t 个服务器.

(2) group_1 对密文组 c 执行盲化子协议 GroupBlind(group_1, c) 获得输出 C_1;group_2 对 c 执行盲化子协议 GroupBlind(group_2, c) 获得输出 C_2.

(3) group_1 对 C_1 进行门限解密获得输出 M_1;group_2 对 C_2 进行门限解密获得输出 M_2.

(4) 执行验证子协议 Verify(group_1)和 Verify(group_2),如果发现有作弊者则将其去除并补充新的 Mix 服务器进入系统,然后返回步骤(1).

(5) 所有 Mix 服务器比较 M_1 是否等于 M_2. 如果相等则 Mix Net 输出 M_1,协议结束;反之则返回步骤(1).

1.3　子协议描述

1.3.1　盲化子协议 GroupBlind(group, c)

令 $L_0=c=((a_{0,1}, b_{0,1}), (a_{0,2}, b_{0,2}), \cdots, (a_{0,N}, b_{0,N}))$. 服务器 $S_i\in$group $(i=1, 2, \cdots, t)$ 依次顺序执行如下操作:任取 $r_{i,j}\in Z_q$ $(j=1, 2, \cdots, N)$ 和置换 π_i,并计算

$$L_i=\pi_i((a_{i-1,1}g^{r_{i,1}}, b_{i-1,1}y^{r_{i,1}}), (a_{i-1,2}g^{r_{i,2}}, b_{i-1,2}y^{r_{i,2}}), \cdots, (a_{i-1,N}g^{r_{i,N}}, b_{i-1,N}y^{r_{i,N}})) \quad .$$

最后将 L_i 在公布栏中公布并作为服务器 S_{i+1} 的输入. 整个子协议的最终输出为 L_t.

1.3.2　验证子协议 Verify(group)

该协议主要用于验证 group 中的服务器是否在盲化子协议中执行了正确的操作.

令 choice 为 Hash(M_1, M_2, C_1, C_2). 服务器 $S_i\in$ group $(i=1, 2, \cdots, t)$ 在公布栏中公布集合 $G\subset\{1, 2, \cdots, N\}$,使得 $|G|=|\{j \mid \text{choice}[j]=0, 1\leqslant j\leqslant N\}|$. 其中,choice$[j]$ 是指把 choice 看成二进制字符串的第 j 位. 然后 S_i 公布 $G_i, Y_i, G_i', Y_i'\in G_q$,并证明:

(1) $G_i\cdot\prod_{1\leqslant j\leqslant N,\text{choice}[j]=0}a_{i-1,j}=\prod_{j\in G}a_{i,j}$,　$Y_i\cdot\prod_{1\leqslant j\leqslant N,\text{choice}[j]=0}b_{i-1,j}=\prod_{j\in G}b_{i,j}$.

(2) $G_i'\cdot\prod_{1\leqslant j\leqslant N,\text{choice}[j]=1}a_{i-1,j}=\prod_{j\in\{1,\cdots,N\}-G}a_{i,j}$,　$Y_i'\cdot\prod_{1\leqslant j\leqslant N,\text{choice}[j]=1}b_{i-1,j}=\prod_{j\in\{1,\cdots,N\}-G}b_{i,j}$.

(3) $\log_g G_i=\log_y Y_i$,　$\log_g G_i'=\log_y Y_i'$.

证明(3)时要用到证明两个离散对数相等的零知识证明技术. 如果 S_i 无法提供有效证明,则说明 S_i 为作弊者.

2 对秘密性的攻击

2.1 攻击方法一

在文献[12]的方案中,明文是用具有可展性的 ElGamal 加密算法加密的. 如果恶意用户以其他用户的密文输入为基础构造具有某种相关性的密文作为自己的输入,那么两个输入所对应的明文也是相关的. 攻击者通过观察 Mix Net 输出中不同明文之间的相关性可以很容易地猜出输入与输出的对应关系. 利用上述漏洞的具体的攻击实例如下:

假设用户 i 先于恶意用户 j 在公布栏上公布了输入 $c_i=(a_i, b_i)=(g^{r_i}, m_i y^{r_i})$,则用户 j 公布自己的输入为 $c_j=(a_i, kb_i)$,$k\in_U G_q$. 这样 c_j 所对应的明文为 km_i. 等待 Mix Net 最终输出明文组 $(m_1, m_2, \cdots, m_N)$ 之后,用户 j 对其进行扫描. 如果发现 m_r 和 m_s 满足 $m_r=km_s$,则显然 m_s 即为用户 i 的输入,输出 m_s 对应于输入 c_i,因此方案的秘密性遭到破坏.

文献[10]中给出了一种抵御此类攻击的方法,即要求用户 i 提交输入 $c_i=(a_i, b_i)=(g^{r_i}, m_i y^{r_i})$ 时必须额外提供一个以为 a_i 公开钥 r_i 为秘密钥的对 (a_i, b_i) 的 ElGamal 签名[13]. 文献[10]已经证明上述加密方法是不可展的. 只有提供了有效签名的用户密文输入才会被 Mix Net 接受.

2.2 攻击方法二

为保证正确性,原方案的关键技术是把同一密文组让两个不同的秘密共享组分别进行盲化、解密然后再对比结果是否一致. 该方法直接拿解密后的明文进行对比使得在协议最终完成之前过早地暴露了明文. 这是该方案的一个严重漏洞. 首先,在基于 Mix Net 的电子投票[4]中这将意味着在正式的结果公布之前部分投票信息已经被泄漏. 其次,该漏洞会被恶意服务器利用以破坏系统的秘密性. 具体的攻击实例如下:

设 $S_1\in \text{group}_1$ 为恶意服务器,并且在主协议第(2)步执行盲化子协议 GroupBlind(group_1, c) 时 S_1 最先获得用户输入的密文组 $c=(c_1, c_2, \cdots, c_N)$. 设 $c_1=(a_1, b_1)$. S_1 的攻击方法为:

(1) S_1 任取 $k\in G_q$ 并令 $c'=((a_1, kb_1), (a_2, b_2/k), c_3, \cdots, c_N)$.

(2) S_1 对 c' 进行盲化和置换并将结果在公布栏上公布.

(3) 待执行完主协议第(3)步之后,S_1 对比 group_1 的解密输出 M_1 和 group_2 的解密输出 M_2. 如果发现存在 $m_{r1}, m_{r2}\in M_1, m_{s1}, m_{s2}\in M_2$ 满足 $m_{r1}=km_{s1}, m_{r2}=m_{s2}/k$,则判断 m_{s1} 为 c_1 所对应的明文,m_{s2} 为 c_2 所对应的明文.

(4) 在主协议第(4)步中,如果 choice[1]和 choice[2]取相同的值,则 S_1 计算:

$$R_1=\sum_{1\leqslant j\leqslant N,\text{choice}[j]=0} r_{1,j}, R_1'=\sum_{1\leqslant j\leqslant N,\text{choice}[j]=1} r_{1,j}\quad .$$

然后公布

$$G_1=g^{R_1},\quad Y_1=y^{R_1},\quad G_1'=g^{R_1'},\quad Y_1'=y^{R_1'}\quad .$$

并以 R_1 和 R_1' 作为论据构造关于 $\log_g G_1=\log_y Y_1$ 和 $\log_g G_1'=\log_y Y_1'$ 的零知识证明.

如果 choice[1]和 choice[2]取不同的值,则 S_1 无法公布合法的 G_1, Y_1, G_1', Y_1'. 因此只有 50% 的概率 S_1 会被确定为作弊者. 在这种情况下,虽然 S_1 会被从系统中去除,但系统的秘密性也遭到了破坏. 因此可以称这种攻击方法为"自杀式攻击".

2.3 攻击方法三

验证子协议 Verify 的核心思想是:随机选取输入密文组的一个子集 H,然后令每个 Mix 服务器公布一个输出密文组的子集 G,并证明 G 所对应的明文之积等于 H 所对应的明文之积. 该方法仅能以 1/2 的概率验证服务器是否正确执行了盲化操作,并且还会暴露 Mix 服务器的部分输入、输出对应关系(因为 G 中的输出必然与 H 中的输入是对应的).

利用上述漏洞的攻击实例如下:

假设 $S_1, S_2, \cdots, S_{t-1}\in \text{group}_1$ 全部为恶意服务器,它们可以相互勾结实现对秘密性的共谋攻击. 设 $H=$

$\{j \mid \text{choice}[j]=0, 1 \leqslant j \leqslant N\}$,$G$ 为 S_t 在验证子协议中公布的集合. 攻击者任取 $i \in G$ 和 $j \in H$,并猜测 $m_i \in M_1$ 所对应的输入密文为 $c_{\pi_1^{-1}\pi_2^{-1}\cdots\pi_{t-1}^{-1}(j)} \in c$,其中 $\pi_1, \pi_2, \cdots, \pi_{t-1}$ 分别为 $S_1, S_2, \cdots, S_{t-1}$ 在执行 GroupBlind 协议时所使用的置换. 显然这种方法猜中的概率为 $2/N$,它优于完全随机猜测的概率 $1/N$,因此系统的秘密性遭到破坏.

3　对正确性和可验证性的攻击

为使 Mix Net 输出错误结果(破坏正确性)并且不被发现(破坏可验证性),group_1 和 group_2 中的恶意服务器可以相互勾结,在执行盲化子协议时对各自输入密文组中相互对应的密文作相同的修改. 由于 ElGamal 加密的可展性,在主协议第 3 步中获得的 M_1 和 M_2 必然是一致的. 又因为 Verify 子协议每次发现欺骗行为的概率只有 $1/2$,所以上述攻击行为能够通过验证的概率是不可忽略的. 基于该思想的攻击实例如下:

设 $S_1, S_2, \cdots, S_i \in \text{group}_1$ 和 $S_1', S_2', \cdots, S_j' \in \text{group}_2$ $(i+j \leqslant t-1)$ 全部为恶意服务器,他们相互勾结,完成以下攻击:

(1) 在执行 GroupBlind(group_1, c) 时,设 S_i 的输入 $L_{i-1} = (c_1, c_2, \cdots, c_N)$. S_i 任取 $r \in G_q$ 并计算 $\hat{L}_{i-1} = ((a_1, rb_1), (a_2, b_2/r), c_3, \cdots, c_N)$,然后 S_i 对 $\hat{L}_{i-1}$ 进行盲化和置换并将结果在公布栏上公布.

(2) 在执行 GroupBlind(group_2, c) 时,设 S_j' 的输入为 L_{j-1}'. S_j' 计算

$$k_1 = \pi_{j-1}'\pi_{j-2}'\cdots\pi_1'\pi_1^{-1}\ \pi_2^{-1}\ \cdots\pi_{i-1}^{-1}(1)\quad,$$

$$k_2 = \pi_{j-1}'\pi_{j-2}'\cdots\pi_1'\pi_1^{-1}\ \pi_2^{-1}\ \cdots\pi_{i-1}^{-1}(2)\quad,$$

其中 $\pi_1, \pi_2, \cdots, \pi_{i-1}$ 分别为 $S_1, S_2, \cdots, S_{i-1}$ 在执行盲化协议时所使用的置换;$\pi_1', \pi_2', \cdots, \pi_{j-1}'$ 分别为 $S_1', S_2', \cdots, S_{j-1}'$ 在执行盲化协议时所使用的置换.

然后 S_j' 将 L_{j-1}' 中的密文 (a_{k_1}', b_{k_1}'),(a_{k_2}', b_{k_2}') 分别替换为 (a_{k_1}', rb_{k_1}') 和 $(a_{k_2}', b_{k_2}'/r)$ 得到 $\hat{L}_{j-1}'$. 最后 S_j' 对 $\hat{L}_{j-1}'$ 进行盲化和置换并将结果在公布栏上公布.

如果用户输入所对应的明文组为 $(m_1, m_2, \cdots, m_N)$,那么执行完主协议第(3)步之后获得的 M_1 和 M_2 经过排序都等于 $(rm_1, m_2/r, \cdots, m_N)$. 在执行验证协议时,如果 $\text{choice}[1] = \text{choice}[2]$ 并且 $\text{choice}[k_1] = \text{choice}[k_2]$,则 S_i 和 S_j' 的欺骗行为都不会被发现. 因此,上述攻击可以成功破坏协议正确性和可验证性的概率为 $1/4$,该概率是不可忽略的.

在上述攻击中,即便 S_i 和 S_j' 无法与其他服务器勾结(例如只有 S_i 和 S_j' 两个恶意服务器),他们也可以用随机猜测法获得相互匹配的密文. 例如 S_j' 可以从 L_{j-1}' 中任选 c_{k_1}', c_{k_2}' 与 S_i 选定的 $c_1, c_2 \in L_{i-1}$ 进行匹配. 这样做能够匹配成功的概率为 $1/(N(N-1))$,而 S_i 和 S_j' 攻击成功的概率为 $1/(4N(N-1))$. 当 N 较小时,该概率是不可忽略的. 实际上,为了使 Mix Net 输出错误结果而不被发现的概率小于 ε,参照组数目 λ 必须满足

$$\left(1/(2N(N-1))\right)^{\lambda-1}\Big/2 \leqslant \varepsilon\quad.$$

4　结 束 语

由以上分析可以看出,文献[12]提出的 Mix Net 方案是不安全的. 恶意用户或恶意服务器可以利用 ElGamal 加密算法的可展性进行主动攻击,以破坏系统的秘密性. 该方案保证正确性的核心技术也存在致命缺陷,两个参照组中的恶意服务器可以相互勾结利用笔者提出的共谋攻击任意篡改系统的输出,并且能以不可忽略的概率通过检验. 用增加冗余 Mix 服务器和参照组数目的方法虽然可以提高检出错误的概率,但系统开销也会大幅度增加,导致其效率将大大低于最近提出的一些完全基于零知识证明的方案[3,7],所以在其基础上做进一步改进的意义不大. 从研究趋势上看,用零知识证明协议保证 Mix Net 健壮性的方法[3,7~9]具有可证明安全性、可公开验证、实现效率高等优点,且完全可以避免笔者提出的攻击,因此已经成为 Mix Net 通信方面的研究热点.

(下转第 934 页)