doi:10.11959/j.issn.1000-436x.2015185

# 选择密文安全的可验证 Mix-Net 协议

李龙海，黄诚强，许尚妹，付少锋
（西安电子科技大学 计算机学院，陕西 西安 710071）

**摘 要**：提出了一种在选择密文攻击下可证明安全的可验证 Mix-Net 协议。在 Wikström Mix-Net 方案基础上，引入了新的密钥生成算法和秘密混洗零知识证明构造方法，提高了安全性。在不暴露输入密文与输出明文匹配关系的条件下，任何人都可以根据 Mix 服务器公布的证据验证输出结果的正确性，即满足可公开验证性；任意发送者还可以追踪和检验自己输入的密文的处理过程，即满足发送者可验证性。基于随机预言机假设证明了该协议在适应性选择密文攻击模型下的安全性。与之前具有类似安全属性的方案相比，所提协议无需信任中心，无需用户与服务器之间的多轮交互，计算和通信复杂度更低，因此是构建安全电子选举协议的理想密码学工具。
**关键词**：混合网络；秘密混洗证明；选择密文安全；电子选举
**中图分类号**：TP393.08 **文献标识码**：A

# CCA-secure verifiable Mix-Net protocol

LI Long-hai, HUANG Cheng-qiang, XU Shang-mei, FU Shao-feng
(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

**Abstract:** A CCA-secure verifiable Mix-Net protocol with provable security was proposed. The protocol was based on Wikström's Mix-Net and improves its security by introducing an improved key generation algorithm and a new method for constructing proof of secret shuffling. Without revealing the correspondence between inputs and outputs, proposed protocol enables everyone to verify the correctness of output plaintexts through checking evidences broadcasted by each server. Thus, it satisfies public verifiability. Any sender can trace and examine the processing procedure of its ciphertext. Thus, proposed protocol satisfies sender verifiability. It is prored to be CCA-secure under the assumption of random oracle. Compared with previous mix-net schemes which are CCA-secure, proposed protocol does not require any trusted center, and incurs fewer interactions between servers which also resulting in a lower computation and communication complexity. Hence, the protocol is an ideal cryptographic tool for constructing secure electronic election protocol.
**Key words:** mix network; proof of secret shuffling; CCA-secure; electronic election

## 1 引言

Mix-Net是由Chaum最早提出的用于实现隐私保护的密码学工具[1]，被广泛应用在匿名电子邮件[1]、匿名通信[2]、电子选举[3,4]、匿名 Web 浏览[5]、电子支付[6]等网络应用系统中。Mix-Net 的核心功能由 Mix 服务器实现。将一组密文$(c_1,c_2,\cdots,c_n)$输入到 Mix 服务器之后，该服务器将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安全的，则攻击者无法确定输入与输出之间的匹配关系，也就无法确定任意消息 $m_i$ 的发送者。因此，Mix-Net 协议是实现发送者匿名性的最常用的技术手段。

为了避免完全信任单个服务器的问题，在复杂 Mix-Net 实现多个 Mix 服务器被级联在一起形成 Mix 网络(mix network)。这些服务器依次对多重加密的输入密文组进行部分解密和重排序，最后输出明文。将 Mix 服务器对密文组的处理过程称为混洗

**收稿日期**：2014-10-14；**修回日期**：2014-12-25
**基金项目**：国家自然科学基金资助项目(61101142)；中央高校基本科研基金资助项目(K50510030012)
**Foundation Items:** The National Natural Science Foundation of China (61101142); The Fundamental Research Funds for the Central Universities(K50510030012)

(shuffle)。在级联方案中，只要有一个服务器是诚实的，就能够保证输入和输出之间的不可关联性。如何防止恶意服务器在混洗过程中篡改消息、如何让用户或任意第三方验证可以输出结果的正确性，是设计 Mix-Net 的难点。因此设计安全、高效的 Mix-Net 协议在密码学研究领域是一项具有挑战性的任务。

在过去的二十几年中，研究人员提出了很多 Mix-Net 协议[7~10]，由于缺少严格的安全性定义和证明，相当数量的方案都被攻破[11~15]，或者只适用于简单的被动攻击模型。在近期的研究中逐渐引入了公钥加密中的可证明安全理论[11]，因此更多的 Mix-Net 协议在设计时考虑了选择密文攻击(CCA, chosen cipher attack)等主动攻击模型以获得更强的安全性。

本文提出了一种新的可证明安全的抗选择密文攻击 Mix-Net 协议。与之前的几种 CCA 安全的 Mix-Net 相比[10,16~19]，所提协议无需信任中心，无需用户与服务器之间的多轮交互，Mix 服务器之间也只有一轮交互，计算和通信复杂度也有所降低，因此是更加实用的方案。该 Mix-Net 还同时满足可公开验证性(public verifiability)和发送者可验证性(sender verifiability)，前者指任何人都可以根据 Mix 服务器公布的相关证据验证输出结果的正确性，后者是指任意发送者都可以利用简单的算法追踪和检验自己输入的密文的处理过程。在设计基于 Mix-Net 的电子选举系统时，可验证性是非常关键的安全特性，因为任何人都可以验证选举结果的正确性，投票者也可以追踪和检验自己的选票是否被计入选举结果。因此，本文设计的可验证 Mix-Net 是构建安全电子选举系统的更理想的密码学工具。

本文的另一贡献是基于 Wikström 提出的 ElGamal 型 Mix-Net 加解密方法[10]设计了一种新的秘密混洗零知识证明(zero knowledge proof of secret shuffle)协议。Wikström 原文中构造的零知识证明存在严重安全漏洞[15]，因此其 Mix-Net 也是不安全的。所提零知识证明可以视为对 Wikström Mix-Net 方案[10]的完善。

## 2 相关工作

1981 年，Chaum[1]首次提出了 Mix-Net 的概念，并给出了一种基于 RSA 的具体实现。在之后的二十几年中，研究人员提出了很多新的 Mix-Net 设计方案，这些方案大致分为两类：逐级解密型[1,8,10]和再加密型[7,9]。逐级解密型 Mix-Net 由多个 Mix 服务器依次对输入密文组进行部分解密和重排序，直至输出明文。该类型 Mix-Net 的优点是效率高、用户可自由选择参与混洗的服务器，缺点是密文长度与服务器数目相关，且很难实现可公开验证性。再加密型 Mix-Net 多是基于 ElGamal 等具有同态特性的加密算法构造的，输入消息只被加密一次，Mix 服务器组依次对密文进行再加密和重排序，最后再联合解密。再加密型的优点是密钥管理简单、密文长度与服务器数目无关、容易保证完整性和可验证性，缺点是必须依赖同态公钥算法，因此效率较低，而且消息的传输路径是固定的。再加密型 Mix-Net 多被用在安全性要求较高但传输延时要求较低的应用中，如电子选举等。本文设计的 Mix-Net 属于逐级解密型，但同样具有再加密型的优点。

在不影响匿名性的前提下，用较低的代价保证混洗消息的完整性和可验证性是设计 Mix-Net 的难点所在。针对该问题目前已经提出的解决方案主要包括以下 4 种：1) 设置多个参照组对比混洗结果[7]；2) 随机抽取部分样本检验[20]；3)在密文消息中加入冗余信息辅助验证[8,9]；4) 利用秘密混洗零知识证明技术[10,21~23]。基于秘密混洗零知识证明技术的 Mix-Net 具有交互次数少、模型简单、容易实现，可证明安全等优点，是该领域的主流研究方向。本文的 Mix-Net 也采用了该技术。

由于缺少严格的安全性定义和安全性证明，早期提出的大部分 Mix-Net 方案都被攻破[11~15]，或者只适用于简单的被动攻击模型。例如基于随机抽样检测的 Mix-Net[20]被广泛应用在大规模电子选举中[3]，但经过十多年后被 Khazaei 等发现仍存在安全漏洞[30]。Küsters 等[31]在严格概率模型下对 Khazaei 改进后的方案进行了验证，其安全性才算得到了可靠的证明。近期的研究开始借鉴公钥加密体制中的可证明安全理论。Abe 最早在选择密文攻击模型下定义了基于挑战游戏方式 Mix-Net 的安全性[11]，但并未给出符合其安全定义的具体构造方法。迄今能在文献中找到的符合 CCA 强安全性定义的 Mix-Net 方案只有文献[10,16~19]。其中，Wikström 的 3 种 Mix-Net[10,16,17]和 Khazaei 的 Mix-Net[19]都是在 UC (universal composability)模型[24]下证明了安全性。假定 UC 模型中的通信模块能够保证消息的可靠递交，则 UC 模型下的 Mix-Net 安全蕴含了 CCA

安全[18]。目前尚不清楚这 2 种安全定义是否等价。Camenisch 的 Mix-Net[18]在 2 种模型下都证明了安全性。

上述 5 种 CCA 安全的 Mix-Net 都存在缺点；其中，文献[16]无法公开验证，且用户和服务器之间需要进行复杂的交互，服务器的计算复杂度也很高；文献[19]也无法公开验证，且服务器之间需要多轮交互；文献[10,17]2 个 Mix-Net 都基于 Wikström 构造的秘密混洗零知识证明，该证明协议被发现存在严重安全漏洞[15]，因此这 2 类 Mix-Net 也是不安全的；Camenisch 的 Mix-Net[18]不满足发送者可验证性，且需要信任中心，虽然该中心可以用全体成员参与的多方安全掷硬币协议代替，但这样大大增加了交互次数。

本文提出了一种新的可证明安全的抗选择密文攻击 Mix-Net 协议，其中，采用了与 Wikström 类似的加解密方法，并引入了新设计的秘密混洗零知识证明。

## 3　基础工具

### 3.1　离散对数证明

设 $G$ 表示一个阶为 $q$ 的有限循环群，$q$ 为素数，$g$ 为 $G$ 的生成元。假定在 $G$ 上 DDH (decision Diffie-Hellman)成立。

设 $x\in\mathbb{Z}_q$，$y=g^x$，且 $g$ 和 $y$ 是公开的，但只有 $\mathcal{P}$ 知道 $x$。示证者 $\mathcal{P}$ 利用离散对数证明 $\mathbb{P}_{\mathrm{DL}}(x,g,y)$ 可以在不暴露关于 $x$ 任何信息的条件下向任意验证者 $\mathcal{V}$ 证明 $\mathcal{P}$ 知道离散对数 $x$ 的事实。实现 $\mathbb{P}_{\mathrm{DL}}(x,g,y)$ 的方法有很多，这里给出一种非交互式证明实现方法。

定义散列函数：$H_1:G^2\to G$，$H_2:G^2\to G$，$H_3:G^3\to\mathbb{Z}_q$。在安全性分析中它们都将被视为随机预言机(RO, random oralce)[25]。

非交互式 $\mathbb{P}_{\mathrm{DL}}(x,g,y)$ 的具体构造方法如下。

任取 $s\in\mathbb{Z}_q$，计算 $u=g^s$，$\overline{g}=H_1(y,u)$，$\overline{y}=\overline{g}^x$，$\overline{u}=\overline{g}^s$，$\hat{g}=H_2(\overline{y},\overline{u})$，$\hat{y}=\hat{g}^x$，$\hat{u}=\hat{g}^s$　$e=H_3(\hat{g},\ \hat{y},\ \hat{u})$，$f=s+ex$。最后 $\mathcal{P}$ 输出$(\overline{y},\hat{y},e,f)$。

$\mathcal{V}$ 接收到$(\overline{y},\hat{y},e,f)$之后，计算 $u=g^f y^{-e}$，$\overline{g}=H_1(y,u)$，$\overline{u}=\overline{g}^f\overline{y}^{-e}$，$\hat{g}=H_2(\overline{y},\overline{u})$，$\hat{u}=\hat{g}^f\ \hat{y}^{-e}$。最后 $\mathcal{V}$ 验证等式 $e=H_3(\hat{g},\ \hat{y},\ \hat{u})$ 是否成立。如果成立，则 $\mathcal{V}$ 输出 1；否则输出 0。

上述 $\mathbb{P}_{\mathrm{DL}}(x,g,y)$ 的交互式版满足诚实验证者零知识性(honest verifier zero-knowledge)，因此在随机预言机模型中可以通过控制 RO 的输出伪造不可区分的 $\mathbb{P}_{\mathrm{DL}}$。具体地讲，假定仿真器 $\mathcal{S}_{\mathrm{DL}}$ 能够控制 $H_1$, $H_2$, $H_3$ 的输出，那么 $\mathcal{S}_{\mathrm{DL}}$ 在不知道离散对数 $x$ 的条件下能够构造仿真 $\mathcal{S}_{\mathrm{DL}}(g,y)$，使 $\mathcal{S}_{\mathrm{DL}}(g,y)$ 和 $\mathbb{P}_{\mathrm{DL}}(x,g,y)$ 的概率分布是不可区分(indistinguish-able)的。构造 $\mathcal{S}_{\mathrm{DL}}(g,y)$ 的方法如下。

任取 $e,f,t_1,t_2\in\mathbb{Z}_q$，然后令 $u=g^f y^{-e}$，令 $H_1(y,u)$ 输出 $g^{t_1}$ ($\overline{g}=g^{t_1}$)，令 $\overline{y}=y^{t_1}$，$\overline{u}=\overline{g}^f\overline{y}^{-e}$，令 $H_2(\overline{y},\overline{u})$ 输出 $g^{t_2}$ ($\hat{g}=g^{t_2}$)，令 $\hat{y}=y^{t_2}$，$\hat{u}=\hat{g}^f\ \hat{y}^{-e}$，令 $H_3(\hat{g},\hat{y},\hat{u})$ 输出 $e$。最后令仿真输出 $\mathcal{S}_{\mathrm{DL}}(g,y)=(\overline{y},\hat{y},e,f)$。

### 3.2　离散对数相等证明

**定义 1**　离散对数相等关系 $R_{\mathrm{EDL}}$。$R_{\mathrm{EDL}}\subset G^4$ 为：$(g,h,y,z)\in R_{\mathrm{EDL}}$ 当且仅当 $\exists\, x\in\mathbb{Z}_q$，满足 $y=g^x$ 且 $z=h^x$。

利用证明 $\mathbb{P}_{\mathrm{EDL}}(x,g,h,y,z)$，$\mathcal{P}$ 可以在不暴露 $x$ 的条件下向验证者 $\mathcal{V}$ 证明 $(g,h,y,z)\in R_{\mathrm{EDL}}$。以下给出一种非交互式 $\mathbb{P}_{\mathrm{EDL}}$ 实现。

定义散列函数：$H_4:G^4\to G$，$H_5:G^2\to G$，$H_6:G^3\to\mathbb{Z}_q$。

构造非交互式 $\mathbb{P}_{\mathrm{EDL}}$ 如下：任取 $s\in\mathbb{Z}_q$，计算 $u=g^s$，$v=h^s$，$\overline{g}=H_4(y,z,u,v)$，$\overline{y}=\overline{g}^x$，$\overline{u}=\overline{g}^s$，$\hat{g}=H_5(\overline{y},\overline{u})$，$\hat{y}=\hat{g}^x$，$\hat{u}=\hat{g}^s$　$e=H_6(\hat{g},\ \hat{y},\ \hat{u})$，$f=s+ex$。$\mathcal{P}$ 输出$(\overline{y},\hat{y}\ ,e,f)$。

$\mathcal{V}$ 接收到$(\overline{y},\hat{y}\ ,e,\ f)$之后，计算 $u=g^f y^{-e}$，$v=h^f z^{-e}$，$\overline{g}=H_4(y,z,u,v)$，$\overline{u}=\overline{g}^f\overline{y}^{-e}$，$\hat{g}=H_5(\overline{y},\overline{u})$，$\hat{u}=\hat{g}^f\ \hat{y}^{-e}$。最后 $\mathcal{V}$ 验证等式 $e=H_6(\hat{g},\ \hat{y},\ \hat{u})$ 是否成立。如果成立，则 $\mathcal{V}$ 输出 1；否则输出 0。

类似于 $\mathbb{P}_{\mathrm{DL}}$，关于 $\mathbb{P}_{\mathrm{EDL}}$ 同样可以通过控制 RO 的输出构造不可区分的仿真 $\mathcal{S}_{\mathrm{EDL}}(g,h,y,z)$。

### 3.3　解密置换关系证明

为了保证混洗过程的正确性，Mix-Net 需要用到秘密混洗零知识证明(zero knowledge proof of secret shuffling)协议。这里先给出解密置换关系 $R_{\mathrm{DP}}$ 的定义。

**定义 2**　解密置换关系 $R_{\mathrm{DP}}$。对任意正整数 $n$，设 $\Sigma_n$ 表示{1, 2,…,$n$}上所有置换构成的集合，则定义关系 $R_{\mathrm{DP}}\subset G^3\times G^{2n}\times G^{2n}$ 为

$$\left\{g,z,y,\{(u_i,v_i)\}_{i=1}^n,\{(U_i,V_i)\}_{i=1}^n\right\}\in R_{\mathrm{DP}}$$

当且仅当存在 $\omega\in\mathbb{Z}_q$，$x\in\mathbb{Z}_q$ 和置换 $\pi\in\Sigma_n$ 满足 $z=g^{\omega}$，$y=g^x$，且对 $i$=1, 2,…,$n$，有

$$(U_i, V_i) = (u_{\pi(i)}^{\omega}, v_{\pi(i)} u_{\pi(i)}^{-\omega x})$$

利用证明协议 $\mathbb{P}_{\mathrm{DP}}$，示证者 $\mathcal{P}$ 可以在不暴露秘密 $(\omega, x)$ 和 $\pi$ 的条件下向验证者 $\mathcal{V}$ 证明 $(g, z, y, \{(u_i, v_i)\}_{i=1}^n, \{(U_i, V_i)\}_{i=1}^n)$ 满足关系 $R_{\mathrm{DP}}$。

这里先假设证明协议 $\mathbb{P}_{\mathrm{DP}}$ 是存在的，然后在第 5 节给出 $\mathbb{P}_{\mathrm{DP}}$ 的具体实现。

### 3.4 同态承诺混洗证明

设 $c$=com($m_1$, $m_2$,…,$m_n$; $r$)，表示关于消息 $m_1$, $m_2$,…,$m_n$的承诺(commitment)函数，$r$是随机化参数。并设 com 满足同态特性，即对任意的($m_0$, $r_0$)和($m_1$, $r_1$)有

$$\mathrm{com}(m_0; r_0)\mathrm{com}(m_1; r_1)=\mathrm{com}(m_0+m_1; r_0+r_1)$$

这里用到的同态承诺函数具体可以在群 $G$ 上利用 Pedersen 的方案[27]构造，此时 $m_1,\cdots,m_n,r\in \mathbb{Z}_q$，$c$=com($m_1,m_2,\cdots,m_n;r$)$\in G$。

**定义 3** 承诺置换关系。$R_{\mathrm{CP}}$ 对任意正整数 $n$，定义关系 $R_{\mathrm{CP}} \subset \mathbb{Z}_q^n \times G$ 为

$(\{m_i\}_{i=1}^n, c) \in R_{\mathrm{CP}}$ 当且仅当 $\exists\, r \in \mathbb{Z}_q, \pi \in \Sigma_n$，满足 $c$=com($m_{\pi(1)},\cdots,m_{\pi(n)}$; $r$)

利用证明协议 $\mathbb{P}_{\mathrm{CP}}(r, \pi, \{m_i\}_{i=1}^n, c)$，示证者 $\mathcal{P}$ 能够在不暴露秘密 $r$ 和 $\pi$ 的条件下证明 $(\{m_i\}_{i=1}^n, c)$ 满足关系 $R_{\mathrm{CP}}$。

Groth[23]给出了一种 $\mathbb{P}_{\mathrm{CP}}$ 的具体实现。如果承诺函数 com 采用 Pedersen 的方案，根据文献[23]的证明，$\mathbb{P}_{\mathrm{CP}}$ 是一个包括 4 轮交互的诚实验证者零知识证明协议，且破坏其健壮性(soundness)等价于解 CDH (computational Diffie-Hellman)难题。$\mathbb{P}_{\mathrm{CP}}$ 的具体构造方法这里不再给出，可以参考文献[23]。

## 4 Mix-Net 详细设计

### 4.1 符号与假设

以下用符号 $S_1,\cdots,S_N$ 表示 $N$ 个发送者，$M_1,\cdots,M_k$ 表示 $k$ 个 Mix 服务器。

设 $G$ 表示阶为 $q$ 的有限循环群，$q$ 为素数，$g$ 为 $G$ 的生成元。假定在群 $G$ 上 DDH 成立。

假设存在具有认证功能的可靠广播系统(BBS 系统)。在协议运行的任意时刻，任意参与者 $P_i$ 都能可靠地向 BBS 上写入消息 $msg$，任何人都能够从 BBS 上读到消息($c$, $P_i$, $msg$)，其中，$c$ 是不断自增的计数值。消息一旦写入 BBS 就不能修改，用户只能向 BBS 追加消息。几乎所有的 Mix-Net 协议都需要建立在上述可靠 BBS 假设之上。这种安全 BBS 的具体构造方法可参考文献[28]。

一个 Mix-Net 协议 $\mathcal{M}$ 包括 3 个子协议：系统参数生成协议 $\mathcal{M}.Gen$、用户消息加密协议 $\mathcal{M}.Enc$ 和密文消息混洗协议 $\mathcal{M}.Shuffle$。

### 4.2 参数生成

Mix 服务器 $M_1,\cdots,M_k$ 共同运行 $\mathcal{M}.Gen(1^{\kappa})$ 协议生成系统公共参数和各个服务器的私钥与公钥，其中 $\kappa$ 为安全参数。具体生成过程如下。

**步骤 1** $k$ 个 Mix 服务器根据参数 $\kappa$ 共同选取群 $G$、$G$ 的生成元 $g$，以及 $H_1$、$H_2$、$H_3$、$H_4$、$H_5$、$H_6$ 等散列函数，并将 $G$ 和 $g$ 以及相关散列函数的描述信息写入 BBS。

**步骤 2** 从 $M_1$ 直至 $M_k$，依次生成密钥($\omega_i, x_i$)和对应的部分公钥。具体方法为：对任意服务器 $M_i$，任取 $\omega_i, x_i \in \mathbb{Z}_q$，然后将 $z_i = g^{\omega_i}$，$y_i = g^{x_i}$ 以及证明 $\mathbb{P}_{\mathrm{DL}}(\omega_i, g, z_i)$ 和证明 $\mathbb{P}_{\mathrm{DL}}(x_i, g, y_i)$ 写入 BBS，并利用可验证秘密共享协议[27]将私钥 $\omega_i, x_i$ 共享给其他 $k$–1 个服务器，密钥恢复门限设为 $\frac{k}{2}$+1。

**步骤 3** 设 $Z_0 = g$。从 $M_1$ 直至 $M_k$，依次计算 $Z_i = Z_{i-1}^{\omega_i}, Y_i = Y_{i-1} Z_i^{x_i}$，并将 $Z_i, Y_i$ 以及证明 $\mathbb{P}_{\mathrm{EDL}}(\omega_i, g, Z_{i-1}, z_i, Z_i)$ 和 $\mathbb{P}_{\mathrm{EDL}}(x_i, g, Z_i, y_i, Y_i Y_{i-1}^{-1})$ 写入 BBS。

以上 3 步中，如果某服务器公布的证明 $\mathbb{P}_{\mathrm{DL}}$ 或 $\mathbb{P}_{\mathrm{EDL}}$ 被验证无效，则删除该服务器，之后协议重新运行。上述协议执行成功之后，任意发送者 $S_j$ 都能从 BBS 上获得这些公开信息，$S_j$ 加密自己的消息时只需用到公钥 $Y = Y_k$。

### 4.3 用户消息加密

为发送消息 $m \in G$，发送者 $S_i$ 执行 $\mathcal{M}.Enc(m)$ 生成密文。具体加密过程为任取 $r \in \mathbb{Z}_q$，计算：$u = g^r, v = Y^r m$ 和证明 $\mathbb{P}_{\mathrm{DL}}(r, g, u)$。

最后用户 $S_i$ 将密文 $C$=($u$,$v$, $\mathbb{P}_{\mathrm{DL}}(r, g, u)$)发送到 BBS。为了简化描述，假定每个用户只发送 1 个消息。

根据文献[29]，$\mathcal{M}.Enc$ 为 CCA 安全的加密算法。密文 $C$ 中 $\mathbb{P}_{\mathrm{DL}}(r, g, u)$ 部分用于证明 $S_i$ 知道秘密 $r$，也就证明了 $S_i$ 知道明文 $m$。

### 4.4 详细混洗过程

混洗过程的启动可以由 2 类事件触发：一是 BBS 上用户发送的加密消息数量超过某个门限，二是到达某个约定的时间点。以后者作为触发条件必须采用同步通信模型，在本文以第一类事件为触发

条件，并假定门限为 $n$，即至少有 $n$ 个用户发送了密文。设 BBS 上用户的密文消息形成列表 $L_*=\{(u_i,v_i,\mathbb{P}_{\mathrm{DL}}(r_i,g,u_i))\}_{i=1}^n$。

针对 $L_*$，服务器 $M_1,\cdots,M_k$ 共同运行消息混洗协议 $\mathcal{M}.Shuffle$。

**步骤 1**　对任意服务器 $M_j$，从 BBS 上读取 $L_*$，然后针对每一个密文$(u_i,v_i,\mathbb{P}_{\mathrm{DL}}(r_i,g,u_i))$验证其证明 $\mathbb{P}_{\mathrm{DL}}$ 的有效性。如果 $\mathbb{P}_{\mathrm{DL}}$ 无效，则从 $L_*$中去掉该密文。为了简化描述，假定 $L_*$中所有的消息都是有效的，这样，最后任意服务器都可以由 $L_*$得到密文列表 $L_0=\{(u_i,v_i)\}_{i=1}^n$。

**步骤 2**　以 $L_0$ 为输入，从服务器 $M_1$ 开始直至 $M_k$ 依次对密文列表进行部分解密和随机重排序，并生成秘密混洗证明。具体过程如下。

1) 令 $j$ 初值为 1。

2) $M_j$ 以密文列表 $L_{j-1}=\{(u_i,v_i)\}_{i=1}^n$ 为输入，任取置换 $\pi_j\in\Sigma_n$ 并计算输出列表

$$L_j=\{(u_i',v_i')\}_{i=1}^n,\quad (u_i',v_i')=(u_{\pi_j(i)}^{\omega_j},v_{\pi_j(i)}u_{\pi_j(i)}^{-\omega_j x_j})$$

其中，$\omega_j,x_j$ 为 $M_j$ 的私钥。$M_j$ 将 $L_j$ 写入 BBS。

3) $M_j$ 以 $\omega_j,x_j,\pi_j$ 为秘密输入，以 $g$, $z_j$, $y_j$, $L_{j\text{-}1}$ 和 $L_j$ 为公共输入生成非交互式证明 $\mathbb{P}_{\mathrm{DP}}$，并将该证明写入 BBS。

4）如果 $j<k$，则令 $j=j+1$，并回到步骤 2)。

**步骤 3**　在步骤 2 中，在 $M_j$ 公布了输出 $L_j$ 和 $\mathbb{P}_{\mathrm{DP}}$ 之后，其他服务器都会根据 BBS 上的公开信息检验 $L_j$ 和 $\mathbb{P}_{\mathrm{DP}}$ 的正确性。特别对于 $M_j$ 的下一级服务器 $M_{j+1}$，在验证 $L_j$ 的正确性之后才会启动自己的部分解密和重排序过程。

**步骤 4**　如果超过 $\frac{k}{2}$ 的服务器发现 $M_j$ 的输出是无效的，则这些服务器将利用秘密共享技术共同恢复 $M_j$ 的私钥 $\omega_j,x_j$，并代替 $M_j$ 对 $L_{j\text{-}1}$ 进行部分解密得到 $L_j$。具体过程不再详细叙述。

很容易验证，如果所有服务器都是诚实的，则最后一个服务器的输出 $L_k$ 中包含明文列表 $\{m_i\}_{i=1}^n$。

## 5　秘密混洗证明协议

虽然 Wikström[10]已经给出了一种 $\mathbb{P}_{\mathrm{DP}}$ 证明协议的具体实现方法，但 Peng[15]发现其存在严重安全漏洞，并被证明其健壮性是不可修复的。为此，需要设计新的交互式证明协议用于证明关系 $R_{\mathrm{DP}}$。本协议采用了与 Groth 证明协议[23]类似的技术实现证明的零知识性和健壮性。新设计的秘密混洗证明协议还需要用到 Groth 的同态承诺混洗证明协议 $\mathbb{P}_{\mathrm{CP}}$ [23]。

下面构造一种包含 7 轮交互的证明协议 $\mathbb{P}_{\mathrm{DP}}$，用于证明 $\{g,z,y,\{(u_i,v_i)\}_{i=1}^n,\{(U_i,V_i)\}_{i=1}^n\}$ 满足关系 $R_{\mathrm{DP}}$。$\{g,z,y,\{(u_i,v_i)\}_{i=1}^n,\{(U_i,V_i)\}_{i=1}^n\}$ 是示证者 $\mathcal{P}$ 和验证者 $\mathcal{V}$ 的公共输入，$(\omega,x)$和置换 $\pi$ 是 $\mathcal{P}$ 的秘密输入(witness)。

$\mathcal{P}$ 和 $\mathcal{V}$ 之间具体的交互式证明过程如下，其中，集合 $\mathbb{Z}_q$ 中任意 2 个元素的加减法运算为模 $q$ 运算。

**步骤 1**　$\mathcal{P}$ 任取 $r,r_d,d_1,\cdots,d_n,d_\omega,d_x,r_U,d_U,r_V,d_V,r_1,r_2,r_3,r_4\in\mathbb{Z}_q$，并计算

$D_\omega=g^{d_\omega}$，$D_x=g^{d_x}$

$c=\mathrm{com}(\pi(1),\cdots,\pi(n);r)$

$c_d=\mathrm{com}(-d_1,\cdots,-d_n;r_d)$

$U_d=(\prod_{i=1}^n U_i^{-d_i})g^{r_U}$

$V_d=(\prod_{i=1}^n v_{\pi(i)}^{-d_i})g^{r_V}$

$C_1=\mathrm{com}(r_U;r_1)$, $C_2=\mathrm{com}(d_U;r_2)$

$C_3=\mathrm{com}(r_V;r_3)$, $C_4=\mathrm{com}(d_V;r_4)$

然后 $\mathcal{P}$ 将 $D_\omega$, $D_x$, $c$, $c_d$,$U_d$, $V_d$,$C_1$,$C_2$, $C_3$, $C_4$ 发送给 $\mathcal{V}$。

**步骤 2**　$\mathcal{V}$ 任取 $t_1,\cdots,t_n\in\mathbb{Z}_q$，然后将 $t_1,\cdots,t_n$ 发送给 $\mathcal{P}$。

**步骤 3**　$\mathcal{P}$ 计算

$f_i=t_{\pi(i)}+d_i$　(for $i$=1,$\cdots$, $n$)

$u_\omega=(\prod_{i=1}^n u_i^{t_i})^{d_\omega}g^{d_U}$

$U_x=(\prod_{i=1}^n U_i^{f_i})^{-d_x}g^{d_V}$

然后 $\mathcal{P}$ 将 $f_1,\cdots,f_n,u_\omega$ 和 $U_x$ 发送给 $\mathcal{V}$。

**步骤 4**　$\mathcal{V}$ 任取 $\lambda,e\in\mathbb{Z}_q$，然后以 $\{\lambda i+t_i\}_{i=1}^n$ 和 $c'=c^\lambda c_d\,\mathrm{com}(f_1,\cdots,f_n;0)$ 为输入运行协议 $\mathbb{P}_{\mathrm{CP}}$，最后 $\mathcal{V}$ 将 $\lambda,e$ 和 $\mathbb{P}_{\mathrm{CP}}$ 的第 1 次交互输出发送给 $\mathcal{P}$。

**步骤 5**　$\mathcal{P}$ 计算

$\rho=\lambda r+r_d$

$c'=c^\lambda c_d\,\mathrm{com}(f_1,\cdots,f_n;0)$

$=\mathrm{com}(\lambda\pi(1)+t_{\pi(1)},\cdots,\lambda\pi(n)+t_{\pi(n)};\rho)$

然后 $\mathcal{P}$ 以 $\pi$，$\rho$，$\{\lambda i+t_i\}_{i=1}^n$ 和 $c'$ 为输入运行证明协议 $\mathbb{P}_{\mathrm{CP}}$，并将 $\mathbb{P}_{\mathrm{CP}}$ 的第 2 次交互输出发送给 $\mathcal{V}$。

**步骤 6**　$\mathcal{V}$ 将 $\mathbb{P}_{\mathrm{CP}}$ 的第 3 次交互内容发送给 $\mathcal{P}$。

**步骤 7**　$\mathcal{P}$ 计算

$f_\omega = e\omega + d_\omega$， $f_U = er_U + d_U$, $z_U$=$er_1$+$r_2$,

$f_x = ex + d_x$， $f_V = er_V + d_V$, $z_V$=$er_3$+$r_4$,

然后$\mathcal{P}$将$f_\omega$、$f_U$、$z_U$、$f_x$、$f_V$、$z_V$和$\mathbb{P}_{\mathrm{CP}}$的第4次交互输出发送给$\mathcal{V}$。

**步骤8** $\mathcal{V}$做如下3类验证。

1) 前面各轮交互中接收到的数值是否属于正确的集合$\mathbb{Z}_q$和$G$;

2) 验证$\mathbb{P}_{\mathrm{CP}}$中$\mathcal{P}$的输出是否符合$\mathbb{P}_{\mathrm{CP}}$协议要求;

3) 验证以下等式是否成立

$$(\prod_{i=1}^{n} u_i^{-t_i f_\omega})(\prod_{i=1}^{n} U_i^{f_i})^e U_d^e u_\omega = g^{f_U} \tag{1}$$

$$[(\prod_{i=1}^{n} v_i^{-t_i})(\prod_{i=1}^{n} V_i^{f_i})V_d]^e (\prod_{i=1}^{n} U_i^{f_i})^{f_x} U_x = g^{f_V} \tag{2}$$

$z^e D_\omega = g^{f_\omega}$

$y^e D_x = g^{f_x}$

$C_1^e C_2 = \mathrm{com}(f_U; z_U)$

$C_3^e C_4 = \mathrm{com}(f_V; z_V)$

以上3类验证全部通过，$\mathcal{V}$输出1；否则，$\mathcal{V}$输出0。

上面给出了$\mathbb{P}_{\mathrm{DP}}$的交互式实现。由于$\mathbb{P}_{\mathrm{DP}}$满足诚实验证者零知识特性，通过选取恰当的散列函数并利用Fiat和Shamir技巧[26]，很容易将$\mathbb{P}_{\mathrm{DP}}$改造为非交互式证明。

## 6 分析

### 6.1 混洗证明安全性分析

**定理1** 第5节的$\mathbb{P}_{\mathrm{DP}}$协议是关于解密置换关系$R_{\mathrm{DP}}$的诚实验证者零知识证明协议，破坏$\mathbb{P}_{\mathrm{ZK}}^{R_{\mathrm{DP}}}$的健壮性等价于解CDH(computational Diffie-Hellman)问题。

**证明**

完整性：将$\mathcal{P}$的秘密输入($\omega, x$)和置换$\pi$，及公共输入$(g, z, y, \{(u_i, v_i)\}_{i=1}^n, \{(U_i, V_i)\}_{i=1}^n)$，直接代入$\mathbb{P}_{\mathrm{DP}}$很容易证明其完整性。

诚实验证者零知识：按照如下方式构造$\mathbb{P}_{\mathrm{DP}}$的2类仿真器$\mathcal{S}_{\mathrm{DP}}^{(1)}$和$\mathcal{S}_{\mathrm{DP}}^{(2)}$，其中，$\mathcal{S}_{\mathrm{DP}}^{(1)}$的输入只包含公共参数$(g, z, y, \{(u_i, v_i)\}_{i=1}^n, \{(U_i, V_i)\}_{i=1}^n)$，$\mathcal{S}_{\mathrm{DP}}^{(2)}$的输入不仅包含公共参数，还包含秘密输入$\pi$。另外，由于$\mathbb{P}_{\mathrm{CP}}$满足诚实验证者零知识特性，因此存在关于$\mathbb{P}_{\mathrm{CP}}$的仿真器$\mathcal{S}_{\mathrm{CP}}$能够在不知道秘密$\pi$的条件下生成与真实协议输出不可区分的交互式证明脚本。$\mathcal{S}_{\mathrm{CP}}$将被用作$\mathcal{S}_{\mathrm{DP}}^{(1)}$和$\mathcal{S}_{\mathrm{DP}}^{(2)}$的子程序。

$\mathcal{S}_{\mathrm{DP}}^{(1)}$生成仿真脚本的具体方法为：从$\mathbb{Z}_q$中任取质询$t_1, \cdots, t_n, \lambda, e$，从$\mathbb{Z}_q$中任取$f_1, \cdots, f_n, r, r_d$，$r_1$，$r_3$，$f_\omega, f_U, z_U, f_x, f_V, z_V$，从$G$中任取$U_d$，$V_d$，令$c$=com(0,⋯,0;$r$)，$c_d$=com(0,⋯,0;$r_d$)，$C_1$=com(0;$r_1$)，$C_3$=com(0;$r_3$)，令$D_\omega = g^{f_\omega} z^{-e}$, $D_x = g^{f_x} y^{-e}$，$C_2 = \mathrm{com}(f_U; z_U) C_1^{-e}$，$C_4 = \mathrm{com}(f_V; z_V) C_3^{-e}$。

令

$$u_\omega = g^{f_U} (\prod_{i=1}^{n} u_i^{t_i f_\omega})(\prod_{i=1}^{n} U_i^{f_i})^{-e} U_d^{-e}$$

$$U_x = g^{f_V} [(\prod_{i=1}^{n} v_i^{-t_i})(\prod_{i=1}^{n} V_i^{f_i})V_d]^{-e} (\prod_{i=1}^{n} U_i^{f_i})^{-f_x}$$

最后以$\{\lambda i + t_i\}_{i=1}^n$和$c' = c^\lambda c_d$ com($f_1$,⋯,$f_n$;0)为输入，运行$\mathbb{P}_{\mathrm{CP}}$的仿真器$\mathcal{S}_{\mathrm{CP}}$并将其输出作为$\mathcal{S}_{\mathrm{DP}}^{(1)}$输出的一部分。

$\mathcal{S}_{\mathrm{DP}}^{(2)}$生成仿真脚本的具体方法为：从$\mathbb{Z}_q$中任取$t_1, \cdots, t_n, \lambda, e, f_1, \cdots, f_n, r, r_d$，$r_1$，$r_2$，$r_3$，$r_4$，$f_\omega, r_U, r_V$，$d_U, d_V, f_x$。令$d_i = f_i - t_{\pi(i)}$，$c$= com($\pi$(1),$\pi$(2),⋯,$\pi$($n$);$r$)，$c_d$=com($d_1$, $d_2$,⋯, $d_n$;$r_d$)。令$C_1$=com($r_U$;$r_1$)，$C_2$=com($d_U$;$r_2$)，$C_3$=com($r_V$;$r_3$)，$C_4$=com($d_V$;$r_4$)，$f_U = er_U + d_U$，$z_U$=$er_1$+$r_2$，$f_V = er_V + d_V$，$z_V$=$er_3$+$r_4$，$D_\omega = g^{f_\omega} z^{-e}$, $D_x = g^{f_x} y^{-e}$。令

$$U_d = (\prod\nolimits_{i=1}^{n} U_i^{-d_i}) g^{r_U}, V_d = (\prod\nolimits_{i=1}^{n} v_{\pi(i)}^{-d_i}) g^{r_V}$$

令

$$u_\omega = g^{f_U} (\prod_{i=1}^{n} u_i^{t_i f_\omega})(\prod_{i=1}^{n} U_i^{f_i})^{-e} U_d^{-e}$$

$$U_x = g^{f_V} [(\prod_{i=1}^{n} v_i^{-t_i})(\prod_{i=1}^{n} V_i^{f_i})V_d]^{-e} (\prod_{i=1}^{n} U_i^{f_i})^{-f_x}$$

最后以$\{\lambda i + t_i\}_{i=1}^n$和$c' = c^\lambda c_d$ com($f_1$,⋯,$f_n$;0)为输入，运行$\mathcal{S}_{\mathrm{CP}}$并将其输出作为$\mathcal{S}_{\mathrm{DP}}^{(2)}$输出的一部分。

下面比较$\mathcal{S}_{\mathrm{DP}}^{(1)}$、$\mathcal{S}_{\mathrm{DP}}^{(2)}$以及真实协议$\mathbb{P}_{\mathrm{DP}}$输出之间的差别。首先，很容易验证$\mathcal{S}_{\mathrm{DP}}^{(1)}$和$\mathcal{S}_{\mathrm{DP}}^{(2)}$的输出都能通过$\mathcal{V}$的验证。其次，$\mathcal{S}_{\mathrm{DP}}^{(2)}$与真实$\mathbb{P}_{\mathrm{DP}}$协议输出相比，除了$\mathcal{S}_{\mathrm{CP}}$输出部分，其他部分具有完全相同的概率分布。对于具有多项式计算能力的攻击者$\mathbb{P}_{\mathrm{CP}}$的诚实验证者零知识特性保证了$\mathcal{S}_{\mathrm{CP}}$输出与$\mathbb{P}_{\mathrm{DP}}$输出对应部分是不可区分的。因此，$\mathcal{S}_{\mathrm{DP}}^{(2)}$与$\mathbb{P}_{\mathrm{DP}}$的输出是不可区分。再次，$\mathcal{S}_{\mathrm{DP}}^{(1)}$与$\mathcal{S}_{\mathrm{DP}}^{(2)}$的输出相比，除了$c, c_d, C_1, C_3, U_d$和$V_d$，其他部分都具有完全相同的概率分布。由于Pedersen承诺函数com具有无条件的隐藏性[27]，因此$\mathcal{S}_{\mathrm{DP}}^{(1)}$和$\mathcal{S}_{\mathrm{DP}}^{(2)}$的$c, c_d, C_1, C_3$也是不可区分的。在$\mathcal{S}_{\mathrm{DP}}^{(1)}$中，

$U_d$,$V_d$是从群 $G$ 中任取的元素，$U_d$,$V_d$符合均匀分布；在 $\mathcal{S}_{\mathrm{DP}}^{(2)}$ 中，$r_U$ ,$r_V$ 是从 $\mathbb{Z}_q$ 中任取的元素，$U_d$,$V_d$ 同样在 $G$ 上均匀分布。因此，$\mathcal{S}_{\mathrm{DP}}^{(1)}$ 和 $\mathcal{S}_{\mathrm{DP}}^{(2)}$ 的的输出具有不可区分性。最后根据混合证明定理(hybrid argument)，$\mathcal{S}_{\mathrm{DP}}^{(1)}$ 与 $\mathbb{P}_{\mathrm{DP}}$ 的输出是不可区分，$\mathcal{S}_{\mathrm{DP}}^{(1)}$ 就是关于 $\mathbb{P}_{\mathrm{DP}}$ 的满足诚实验证者零知识特性的仿真器。

健壮性：假定只具有多项式计算能力的证明者 $\mathcal{P}^*$ 能够在不知道秘密输入 $\omega,x$ 和 $\pi$ 的条件下针对任意的质询 $t_1,\cdots,t_n$ , $\lambda,e$ 和 $\mathbb{P}_{\mathrm{CP}}$ 的质询生成有效的交互式证明脚本。下面利用 $\mathcal{P}^*$ 推导出矛盾。

$\mathcal{P}^*$ 输出的 $f_1,\cdots,f_n$ 必然满足 $f_i=t_{\pi(i)}+d_i$ ($i$=1, …, $n$)，且 $\pi$ 与事先封装在 $c$ 中的置换是一致的，否则，与 $\mathbb{P}_{\mathrm{CP}}$ 的健壮性矛盾。

用不同的 $e$ 值输入 $\mathcal{P}^*$ 直到获得 2 个合法的输出，假定符合条件的 2 个值分别为 $e$ 和 $e'$，则有 $C_1^eC_2=\mathrm{com}(f_U;z_U)$，$C_1^{e'}C_2=\mathrm{com}(f_U';z_U')$，$z^eD_\omega=g^{f_\omega}$，$z^{e'}D_\omega=g^{f_\omega'}$。因此，

$$C_1^{e-e'}=\mathrm{com}(f_U-f_U';z_U-z_U')$$
$$z^{e-e'}=g^{f_\omega-f_\omega'}$$

进一步可得

$$r_U=\frac{f_U-f_U'}{e-e'}$$
$$\omega=\frac{f_\omega-f_\omega'}{e-e'}$$

以及 $d_U$ 的值。

用同样的办法可得 $r_V$、$d_V$ 和 $x$ 的值。基于以上获得值将式(1)改写为

$$(\prod_{i=1}^n u_i^{-t_i\omega}\prod_{i=1}^n U_i^{f_i}U_d)^e(\prod_{i=1}^n u_i^{-t_id_\omega})u_\omega=g^{er_U+d_U}$$

上式针对不同的 $e$ 值都成立，则说明

$$\prod_{i=1}^n u_i^{-t_i\omega}\prod_{i=1}^n U_i^{f_i}U_d=g^{r_U} \tag{3}$$

将 $f_i=t_{\pi(i)}+d_i$ 代入式(3)可得

$$(\prod_{i=1}^n u_i^{-t_i\omega}\prod_{i=1}^n U_i^{t_{\pi(i)}})(\prod_{i=1}^n U_i^{d_i})U_d=g^{r_U}$$

上式针对不同的质询 $t_1,\cdots,t_n$ 都能保持成立说明

$$\prod_{i=1}^n u_i^{-t_i\omega}\prod_{i=1}^n U_i^{t_{\pi(i)}}=1 \tag{4}$$

式(4)针对超过 $n$+1 种不同的 $t_1,\cdots,t_n$ 都成立的事实说明，对任意的 $i$，都满足 $U_i=u_i^\omega$。有 $\frac{1}{q}$ 的概率不满足，与输入 $U_i$ ,$u_i$ 的取值有关，但该概率是关于安全参数 $\kappa$ 的可忽略函数。

同理，将式(2)改写成

$$[(\prod_{i=1}^n v_i^{-t_i}V_i^{f_i})(\prod_{i=1}^n U_i^{f_i})^xV_d]^e(\prod_{i=1}^n U_i^{f_i})^{d_x}U_x=g^{er_V+d_V}$$

上式针对不同的 $e$ 值都成立说明

$$(\prod_{i=1}^n v_i^{-t_i}V_i^{f_i})(\prod_{i=1}^n U_i^{f_i})^xV_d=g^{r_V} \tag{5}$$

将式(5)再展开获得

$$(\prod_{i=1}^n v_i^{-t_i})(\prod_{i=1}^n V_iU_i^x)^{t_{\pi(i)}}(\prod_{i=1}^n V_iU_i^x)^{d_i}V_d=g^{r_V}$$

上式针对不同的质询 $t_1,\cdots,t_n$ 都能保持成立说明

$$(\prod_{i=1}^n v_i^{-t_i})(\prod_{i=1}^n V_iU_i^x)^{t_{\pi(i)}}=1$$

即对任意的 $i$，都满足 $V_i=v_iU_i^{-x}$。

因此，如果 $\mathcal{P}^*$ 不违背 $\mathbb{P}_{\mathrm{CP}}$ 的健壮性，则有对任意的 $i$，都满足 $U_i=u_i^\omega$，$V_i=v_iU_i^{-x}$，即 $\{(u_i,v_i)\}_{i=1}^n$, $\{(U_i,V_i)\}_{i=1}^n$ 满足关系 $R_{\mathrm{DP}}$。

如果不满足 $R_{\mathrm{DP}}$，$\mathcal{P}^*$ 欺骗 $\mathcal{V}$ 等价于破坏 $\mathbb{P}_{\mathrm{CP}}$ 的健壮性，破坏 $\mathbb{P}_{\mathrm{CP}}$ 的健壮性等价于破坏承诺函数 com 的绑定性[23]。由于采用了 Pedersen 承诺方案，因此破坏 com 的绑定性等价于解 CDH 问题。(证毕)

### 6.2　Mix-Net 安全性分析

设$\{S_1,\cdots,S_N\}$为用户集合，$\{M_1,\cdots,M_k\}$为 Mix 服务器集合。$\mathcal{A}$ 表示任意具有多项式计算能力的攻击者，并且 $\mathcal{A}$ 能够控制除了$\{S_1,\cdots,S_H\}$之外的用户和除了$\{M_1,\cdots,M_h\}$之外的 Mix 服务器。另外要求 $\mathcal{A}$ 能够控制的恶意用户集合和恶意服务器集合在协议运行期间不能改变。$\mathcal{M}$ 表示一种具体的 Mix-Net 实现协议。

**定义 4**　正确性。设诚实用户的消息输入为 $K$=$\{m_1,\cdots,m_h\}$，$\mathcal{M}$ 运行结束后的输出为 $K'$=$\{m_1',\cdots,m_n'\}$。如果满足 $K\subseteq K'$，则称 $\mathcal{M}$ 的该次执行是正确的，或称输出 $K'$ 是正确的。

对于本文设计的 Mix-Net 协议，很容易验证：如果所有参与者都是诚实的，则一定能产生正确输出。

**定义 5**　健壮性。设 $K$= $\{m_1,\cdots,m_h\}$是 $\mathcal{A}$ 确定的诚实用户的输入，在 $\mathcal{M}$ 运行期间 $\mathcal{A}$ 尽力通过所控制的用户和服务器破坏协议的正常工作。如果 $\mathcal{M}$ 不能产生正确输出的概率是 $\kappa$ 的可忽略函数，则称 $\mathcal{M}$ 满足健壮性。

**定理 2**　如果攻击者至多能控制 $\frac{k}{2}-1$ 个 Mix 服

务器，则本文提出的 Mix-Net 协议满足健壮性。

**证明** BBS 系统保证了各个服务器的输出能被所有参与者看到。

如果恶意服务器在 $\mathcal{M}.Gen$ 阶段存在恶意行为，则它们公布的证明 $\mathbb{P}_{\mathrm{DL}}$ 和 $\mathbb{P}_{\mathrm{EDL}}$ 能够被诚实服务器验证通过的概率是可忽略的。这些服务器被发现作弊之后将从系统中被删除，剩余的服务器将重新生成公共参数和密钥。

如果恶意服务器在 $\mathcal{M}.Shuffle$ 阶段存在恶意行为，则它们公布的证明 $\mathbb{P}_{\mathrm{DP}}$ 能够被诚实服务器验证通过的概率是可忽略的。一旦某服务器 $M_j$ 被发现是恶意的，该服务器将被删除。只要剩余的服务器中超过 $\frac{k}{2}$ 是诚实的，那么这些服务器可以利用门限秘密共享技术共同恢复 $M_j$ 的私钥 $\omega_j, x_j$，并代替 $M_j$ 实施混洗操作，Mix-Net 协议可以继续运行直至输出正确结果。

因此，只要恶意服务器数目小于等于 $\frac{k}{2}-1$，所提 Mix-Net 协议就能以压倒性概率产生正确输出。(证毕)

**定义 6** 可公开验证性。设 $\mathcal{V}$ 表示只能读取 BBS 信息的任意 Mix-Net 正确性验证者。如果 Mix-Net 输出错误结果且 $\mathcal{V}$ 输出 1 的概率是 $\kappa$ 的可忽略函数，则称该 Mix-Net 满足可公开验证性。

**定理 3** 本文提出的 Mix-Net 协议满足可公开验证性。

**证明** 任意验证者都可以通过 $\mathcal{M}.Gen$ 阶段的证明 $\mathbb{P}_{\mathrm{DL}}$ 和 $\mathbb{P}_{\mathrm{EDL}}$ 验证协议公共参数的正确性，通过 $\mathcal{M}.Enc$ 阶段的证明 $\mathbb{P}_{\mathrm{DL}}$ 验证用户输入消息的合法性，通过 $\mathcal{M}.Shuffle$ 阶段的证明 $\mathbb{P}_{\mathrm{DP}}$ 验证混洗过程的正确性。这 3 个阶段的证明都能验证通过，则保证了输出结果的正确性。要想输出错误结果而逃过验证者的检验，必将面临破坏 $\mathbb{P}_{\mathrm{DL}}$、$\mathbb{P}_{\mathrm{EDL}}$ 或 $\mathbb{P}_{\mathrm{DP}}$ 的健壮性，该事件发生的概率是 $\kappa$ 的可忽略函数。(证毕)

另外，本文的 Mix-Net 还满足发送者可验证性。具体地讲，对任意服务器 $M_j$ 的输入 $L_{j-1}=\{(u_i,v_i)\}_{i=1}^{n}$ 和输出 $L_j=\{(U_i,V_i)\}_{i=1}^{n}$，任意发送者都可以利用自己调用 $\mathcal{M}.Enc$ 时生成的秘密值 $r$ 追踪自己的消息在 $L_{j-1}$ 和 $L_j$ 中的精确位置。具体方法如下。

对 $L_{j-1}$ 中的任意元素 $(u_i,v_i)$，计算 $U_i'=Z_j^r$，$V_i'=v_iY_j^{-r}Y_{j-1}^r$，然后将 $(U_i',V_i')$ 与 $L_j$ 中的所有元素逐一匹配。如果发现 $(U_l,V_l)=(U_i',V_i')$，则可以确定 $(u_i,v_i)$ 和 $(U_l,V_l)$ 即为该发送者的消息。利用上述方法，发送者可以从第一级服务器追踪自己的消息是否被正确处理，直至最后一级。

**定义 7** CCA 匿名性。定义攻击者 $\mathcal{A}$ 和 $\mathcal{M}$ 之间的一个游戏 Game ANON-CCA($\mathcal{M},\mathcal{A}$)如下。

1) 运行 $\mathcal{M}.Gen$ 生成公共参数和每个 Mix 服务器的公私钥对，并将公共参数和所有公钥公布到BBS上。

2) $\mathcal{A}$ 被允许以任意输入调用 $\mathcal{M}.Enc$ 和 $\mathcal{M}.Shuffle$ 任意多项式次数，即 $\mathcal{A}$ 可以将 $\mathcal{M}$ 用作解密预言机(decryption oracle)。

3) $\mathcal{A}$ 生成 2 个明文列表 $\mathrm{MSG}_0=\{m_1,\cdots,m_H\}$ 和 $\mathrm{MSG}_1=\{m_{\pi(1)},\cdots,m_{\pi(H)}\}$，其中 $\pi\in\Sigma_H$。

4) 任取 $b\in\{0,1\}$，然后以 $\mathrm{MSG}_b$ 作为 $H$ 个诚实用户的输入(恶意用户输入由 $\mathcal{A}$ 任选)，运行 $\mathcal{M}.Enc$ 和 $\mathcal{M}.Shuffle$，最后将 $\mathcal{M}$ 运行期间所有公开的中间结果和最后输出发送给 $\mathcal{A}$。

5) $\mathcal{A}$ 被再次允许将 $\mathcal{M}$ 用作解密预言机多项式。

6) 最后 $\mathcal{A}$ 输出 $b'\in\{0,1\}$。

$\mathcal{A}$ 赢得上述游戏的优势被定义为

$$Adv_{\mathcal{M}}^{\text{ANON-CCA}}(\mathcal{A})=\left|\Pr(b'=b)-\frac{1}{2}\right|$$

如果对于任意的攻击者 $\mathcal{A}$，$Adv_{\mathcal{M}}^{\text{ANON-CCA}}(\mathcal{A})$ 都是关于 $\kappa$ 的可忽略函数，则称 $\mathcal{M}$ 满足 CCA 匿名性。

**定理 4** 如果攻击者至多能控制 $k-1$ 个 Mix 服务器和 $N-2$ 个用户，基于随机语言机模型和 DDH 问题困难性假设，本文提出的 Mix-Net 协议满足 CCA 匿名性。

**证明** 首先将定义 7 中的游戏转化为另一个等价游戏。对于 $\mathcal{A}$ 选定的 $\mathrm{MSG}_0$、$\mathrm{MSG}_1$ 和 $\pi$，总能找到 $H$ 个置换 $\sigma_1,\cdots,\sigma_H\in\Sigma_H$，满足 $\pi=\sigma_1,\sigma_2,\cdots,\sigma_H$，且 $\sigma_j$ 和 $\sigma_{j+1}$ 之间只有 2 个元素发生了位置互换。定义 $H$ 个新游戏 $\mathrm{Game}_1,\cdots,\mathrm{Game}_H$。设 $\mathrm{MSG}_0^{(0)}=\mathrm{MSG}_0=\{m_1,\cdots,m_H\}$，则 $\mathrm{Game}_j$ ($1\leqslant j\leqslant H$)和定义 7 的游戏是相同的，除了在 $\mathrm{Game}_j$ 中 $\mathcal{A}$ 选定的 2 个明文列表为 $\mathrm{MSG}_0^{(j)}=\{m_{\sigma_1\cdots\sigma_{j-1}(1)},\cdots,m_{\sigma_1\cdots\sigma_{j-1}(H)}\}$ 和 $\mathrm{MSG}_1^{(j)}=\{m_{\sigma_1\cdots\sigma_j(1)},\cdots,m_{\sigma_1\cdots\sigma_j(H)}\}$。显然 $\mathrm{MSG}_1^{(H)}=\mathrm{MSG}_1=\{m_{\pi(1)},\cdots,m_{\pi(H)}\}$。设 $\mathcal{A}$ 在 $\mathrm{Game}_j$ 中的优势为 $Adv_j$，则有如下不等式成立

$$Adv_{\mathcal{M}}^{\text{ANON-CCA}}\leqslant\sum_{j=1}^{H}Adv_j \tag{6}$$

现假定存在攻击者 $\mathcal{A}$ 能够在定义 7 的游戏中获得不可忽略的优势，则根据不等式(6)，必存在 $j(1\leqslant j\leqslant H)$使 $\mathcal{A}$ 在 $\text{Game}_j$ 中获得不可忽略的优势。

下面利用 $\text{Game}_j$ 中的 $\mathcal{A}$ 来解任意的 DDH 问题，即构造区分器 $\mathcal{D}$，$\mathcal{D}$ 根据输入$(g^{\alpha},g^{\beta},g^{\gamma})$判断 $\gamma$ 是否等于 $\alpha\beta$。

$\mathcal{D}$ 模拟 Mix-Net 协议 $\mathcal{M}$ 和 $\mathcal{A}$ 之间进行 $\text{Game}_j$ 游戏。恶意服务器由 $\mathcal{A}$ 控制，而诚实服务器由 $\mathcal{D}$ 控制。$\mathcal{D}$ 的具体模拟方法如下。

1) $\mathcal{D}$ 选定任意诚实服务器 $M_l$(根据假设，至少存在 1 个诚实的服务器)，不妨设 $l$>1。在 $\mathcal{M}.Gen$ 阶段的第 2 步，$\mathcal{D}$ 令 $y_l=g^{\alpha}$，并通过控制散列函数 $H_1$、$H_2$、$H_3$(均被视为 RO)的输出伪造证明 $\mathbb{P}_{\text{DL}}=\mathcal{S}_{\text{DL}}(g,y_l)$，$\omega_l$ 仍然按照协议规定从 $\mathbb{Z}_q$ 中任取。除了 $M_l$ 之外的诚实服务器严格按照协议规定生成密钥$(\omega_i,x_i)$和对应的公钥 $z_i=g^{\omega_i}$，$y_i=g^{x_i}$。

2) 如何获得 $g^{\omega_1\cdots\omega_l\alpha}$。在后面的模拟中，$\mathcal{D}$ 需要生成 $g^{\omega_1\cdots\omega_l\alpha}$，但 $\mathcal{D}$ 不知道 $\alpha$。$\omega_1,\cdots,\omega_{l-1}$ 中的某些密钥可能属于某些恶意 Mix 服务器(由 $\mathcal{A}$ 控制)，也不为 $\mathcal{D}$ 所知。在这些限制下，$\mathcal{D}$ 生成 $g^{\omega_1\cdots\omega_l\alpha}$ 的具体方法为在服务器 $M_1$ 构造证明 $\mathbb{P}_{\text{DL}}(\omega_1,g,z_1)$ 时，$\mathcal{D}$ 令随机预言机 $H_1$ 在输入 $(z_1,u)$ 时输出 $g^{\alpha}$，由此可以获得 $g^{\omega_1\alpha}$。同理，在 $M_2$ 构造证明 $\mathbb{P}_{\text{DL}}$ 时，$\mathcal{D}$ 令 $H_1$ 在适当的询问点上输出 $g^{\omega_1\alpha}$，这样就可以获得 $g^{\omega_1\omega_2\alpha}$。依此类推，直至获得 $g^{\omega_1\cdots\omega_l\alpha}$。

用类似的技巧，$\mathcal{D}$ 可以获得 $\eta_1=g^{\omega_1\beta}$，$\eta_2=g^{\omega_1\omega_2\beta},\cdots,\eta_k=g^{\omega_1\cdots\omega_k\beta}$，和 $\theta_1=g^{\omega_1x_1\beta}$，$\theta_2=g^{\omega_1\omega_2x_2\beta}$，$\cdots$，$\theta_k=g^{\omega_1\cdots\omega_kx_k\beta}$，以及 $\theta_{\gamma}=g^{\omega_1\cdots\omega_l\gamma}$。

3) 在 $\mathcal{M}.Gen$ 第 3 步，构造公钥 $Z$ 的方法保持不变。在构造公钥 $Y_l$ 时，$\mathcal{D}$ 令 $Y_l=Y_{l-1}g^{\omega_1\cdots\omega_l\alpha}$，$M_l$ 关于 $Y$ 的有效性证明 $\mathbb{P}_{\text{EDL}}$ 用 $\mathcal{S}_{\text{EDL}}(g,Z_l,y_l,Y_lY_{l-1}^{-1})$ 代替。

4) 解密预言机的实现。$\mathcal{D}$ 按照如下方式实现 Mix-Net 解密预言机(ANON-CCA 游戏的第 2 步和第 5 步)：在任意恶意用户 $S_i$ 构造证明 $\mathbb{P}_{\text{DL}}(r_i,g,u_i)$ 时，$\mathcal{D}$ 令 $H_1$ 在恰当询问点上的输出 $Z_kZ_l^{-1}=g^{\omega_{l+1}\cdots\omega_k}$，这样就可以获得 $g^{\omega_{l+1}\cdots\omega_kr_i}$，令 $H_2$ 在恰当询问点上的输出为 $Y_l^{-1}$，这样可以获得 $Y_l^{-r_i}$。

设解密预言机获得的初始输入密文列表为 $L_0=\{(u_i,v_i)\}_{i=1}^n$。当混洗过程进行到服务器 $M_l$ 时，$\mathcal{D}$ 首先检查前面 $l$−1 级服务器公布的 $\mathbb{P}_{\text{DP}}$ 证明的合法性。如果发现存在无效的 $\mathbb{P}_{\text{DP}}$，则解密预言机拒绝提供服务，输出无效消息 $\perp$。如果 $l$−1 个 $\mathbb{P}_{\text{DP}}$ 证明都是有效的，则 $M_l$ 根据 $L_0$ 计算 $L'=\{(g^{\omega_{l+1}\cdots\omega_kr_i},v_iY_l^{-r_i})\}_{i=1}^n$，然后任取 $\pi\in\Sigma_n$，并根据 $\pi$ 对 $L'$ 进行重排序获得最终输出 $L_l$，$M_l$ 还须通过控制相应 RO 的输出伪造非交互式证明 $\mathbb{P}_{\text{DP}}$。

其他的诚实服务器 $M_i$ 仍然按照协议规定对输入密文列表进行部分解密和重排序，因为$(\omega_i,x_i)$是 $M_i$ 已知的密钥。

5) DDH 问题的嵌入。在 $\text{Game}_j$ 游戏的第 3 步，$\mathcal{A}$ 确定了 $\text{MSG}_0^{(j)}$ 和 $\text{MSG}_1^{(j)}$ 之后，对比 $\text{MSG}_0^{(j)}$ 和 $\text{MSG}_1^{(j)}$，只存在 2 个元素位置发生了互换，不妨设为 $m_1$ 和 $m_2$，对应的 2 个诚实发送者为 $S_1$ 和 $S_2$。$\mathcal{D}$ 任取 $t_1,t_2\in\mathbb{Z}_q$，将 $m_1$ 和 $m_2$ 对应的密文分别设置为 $c_1$=( $u_1=g^{\beta t_1}$，$v_1=m_1\prod_{i=1}^k\theta_i^{t_1}$ ) 和 $c_2$= ( $u_2=g^{\beta t_2}$，$v_2=m_2\prod_{i=1}^k\theta_i^{t_2}$ )。实际上，$g^{\gamma}$ 被嵌入到了 $\theta_{\gamma}=g^{\omega_1\cdots\omega_l\gamma}$ 中。$\mathcal{D}$ 任取 $b\in\{0,1\}$，然后以 $\text{MSG}_b^{(j)}$ 作为诚实用户的输入运行协议 $\mathcal{M}$。除了 $S_1$ 和 $S_2$，其他诚实用户严格按照协议规定运行。如果 $b$=0，则令 $S_1$ 输出 $c_1$，$S_2$ 输出 $c_2$；否则，令 $S_1$ 输出 $c_2$，$S_2$ 输出 $c_1$。$S_1$ 和 $S_2$ 对应的证明 $\mathbb{P}_{\text{DL}}$ 也需要用 $\mathcal{S}_{\text{DL}}$ 伪造。

设初始输入密文列表为 $L_0=\{(u_i,v_i)\}_{i=1}^n$。当混洗进行到服务器 $M_l$ 时，$M_l$ 根据 $L_0$ 计算 $L'=\{(g^{\omega_{l+1}\cdots\omega_kr_i},v_iY_l^{-r_i})\}_{i=3}^n$。$M_l$ 令 $u_1'=\eta_{l+1}^{t_1}$，令 $v_1'=m_1\prod_{i=l+1}^k\theta_i^{t_1}$；令 $u_2'=\eta_{l+1}^{t_2}$，$v_2'=m_2\prod_{i=l+1}^k\theta_i^{t_2}$。$\mathcal{D}$ 将$(u_1',v_1')$和$(u_2',v_2')$添加到 $L'$ 中形成列表 $L''$。$M_l$ 任取 $\pi\in\Sigma_n$，并根据 $\pi$ 对 $L''$ 进行重排序获得最终输出 $L_l$，$M_l$ 还须通过控制相应 RO 的输出伪造证明 $\mathbb{P}_{\text{DP}}$。

6) 如果 $\mathcal{A}$ 的输出 $b'=b$，则 $\mathcal{D}$ 认为 $\gamma=\alpha\beta$，输出 1；否则认为 $\gamma\neq\alpha\beta$，输出 0。

在上述模拟游戏中，如果 $\gamma=\alpha\beta$，则 $\mathcal{D}$ 的模拟输出和真实 Mix-Net 协议的运行脚本具有相同的概率分布。如果 $\gamma\neq\alpha\beta$，则 $v_1$、$v_2$ 对 $\mathcal{A}$ 而言完全均匀分布，对 $\mathcal{A}$ 赢得 ANON-CCA 游戏没有任何帮助。如果定义区分器 $\mathcal{D}$ 的优势为

$$Adv(\mathcal{D})=|\Pr(\mathcal{D}(g^{\alpha},g^{\beta},g^{\gamma})=1\,|\,\gamma=\alpha\beta)-\Pr(\mathcal{D}(g^{\alpha},g^{\beta},g^{\gamma})=1\,|\,\gamma\neq\alpha\beta)|$$

则 $Adv(\mathcal{D})=|\Pr(\mathcal{A}_{\mathcal{D}}=b\,|\,\gamma=\alpha\beta)-\Pr(\mathcal{A}_{\mathcal{D}}=b\,|\,\gamma\neq\alpha\beta)|=|\Pr(\mathcal{A}_{\mathcal{M}}=b)-1/2|=Adv_{\mathcal{M}_j}^{\text{ANON-CCA}}(\mathcal{A})$。根据 6)，$Adv_{\mathcal{M}_j}^{\text{ANON-CCA}}(\mathcal{A})$ 是不可忽略的。因此这里构造的 $\mathcal{D}$ 也

能够以不可忽略的优势解决 DDH 问题。这与 DDH 假设矛盾。(证毕)

### 6.3 效率分析

迄今在文献中找到的具有 CCA 强安全性，并且能公开验证的 Mix-Net 方案只有文献[10,17,18]。其中 WikStrom 的 2 种 Mix-Net[10,17]都存在严重安全漏洞[15]，Camenisch 的 Mix-Net[18]和本文方案最为相似，但该方案需要信任中心的参与。在下面的计算和通信复杂度分析中，主要与 Camenisch 方案[18]进行对比。为了便于比较，并未考虑预计算和批量验证等效率优化技巧，也未考虑用户或服务器作弊的情况。

设表示群 $G$ 中的任一元素需要用 $l_G$ bit，表示 $\mathbb{Z}_q$ 中的元素需要 $l_q$ bit。用 exp 表示群 $G$ 上的 1 个指数运算，并作为估算计算复杂度的基本函数。

本方案中用户加密消息需用 2 个 exp，构造 $\mathbb{P}_{\mathrm{DL}}$ 用 5 个 exp，最后形成的密文长度为 $4l_G+2l_q$ bit。而 Camenisch 方案中用户构造密文需用计算 6 个 exp，密文长度为$(6l_G+2l_q)$ bit。因此本方案用户端的计算复杂度略高于 Camenisch 方案，但通信复杂度略低。而 Mix-Net 系统性能关键取决于服务器端的复杂度。

假定在 1 次 Mix-Net 的运行中需要处理的消息数目为 $n$。在服务器端最耗时的操作是构造和验证 $\mathbb{P}_{\mathrm{DP}}$。Camenisch 的 Mix-Net 方案采用了 Groth 的混洗证明协议，而本文的 $\mathbb{P}_{\mathrm{DP}}$ 是对 Groth 协议[23]的改进。主要增加了 $u_\omega$ 和 $U_x$ 2 个元素，使示证者 $\mathcal{P}$ 的计算复杂度由 $6n+O(1)$[23]增加到 $8n+O(1)$。其中，$O(1)$表示常数个 exp 运算，在 $n$ 较大时可以忽略不计。验证 $\mathbb{P}_{\mathrm{DP}}$ 的计算量仍然为 $6n$，与原协议相同。虽然此处的复杂度高于文献[23]，但本方案将解密和混洗过程结合在一起，而 Camenisch 的 Mix-Net 仅包含再加密过程，再加密之后还有独立的联合解密过程(需要额外一轮交互)，在解密时每个服务器还要构造解密有效性证明，还要验证其他服务器的解密有效性。因此，整体上，本方案在计算和通信复杂度上低于 Camenisch 的方案，并且少一轮交互过程。表 1 详细对比了本方案和 Camenisch 方案的一个 Mix 服务器在协议运行期间各个主要操作的复杂度。

通过表 1 可以看出，当 Mix-Net 包含 2 个以上 Mix 服务器时，本文方案在性能上优于 Camenisch 的方案，而且其服务器数目 $k$ 越大，优势越明显。

**表 1　2 种方案复杂度比较**

| 性能 | 本方案 | 文献[23]的方案 |
|---|---|---|
| 验证消息合法性 | $3n$ | $2n$ |
| 再加密 | 0 | $2n$ |
| 构造混洗证明 | $8n$ | $6n$ |
| 验证混洗证明 | $6(k-1)n$ | $6(k-1)n$ |
| 解密 | $2n$ | $n$ |
| 构造解密证明 | 0 | $2n$ |
| 验证解密证明 | 0 | $2(k-1)n$ |
| 计算量合计 | $(6k+7)n$ | $(8k+5)n$ |
| 发送比特数 | $2nl_G+3nl_q$ | $3nl_G+5nl_q$ |
| 接收比特数 | $2knl_G+3(k-1)nl_q$ | $3knl_G+5(k-1)nl_q$ |

## 7 结束语

Mix-Net 协议作为一种实现匿名性、基本的密码学工具，获得了广泛应用，因此在严格定义的安全模型下进行协议设计并给出形式化安全性证明是十分必要的。本文提出了一种可证明安全的抗选择密文攻击的 Mix-Net 协议。强安全性保证使该协议可以更方便、更安全地集成到其他应用中，作为保证匿名性的基础设施。另外，方案还具有可公开验证、发送者可验证、无信任中心、交互次数少、复杂度低等优点。该方案的主要缺点是必须引入随机预言机假设，该假设过于依赖所选散列函数的随机特性。

## 参考文献：


[1] CHAUM D. Untraceable electronic mail, return addresses, and digital pseudonyms[J]. Communications of the ACM, 1981, 24(2): 84-88.

[2] DINGLEDINE R, MATHEWSON N, SYVERSON P. Tor: the second-generation onion router[A]. Proceedings of the 13th USENIX Security Symposium[C]. San Antonio, USA, 2004. 303-320.

[3] DEMIREL D, HENNING M, VANDEGRAAF J, *et al*. Prêt à voter providing everlasting privacy[A]. Proceedings of the 4th International Conference on E-Voting and Identity[C]. Guildford, UK, 2013. 156-175.

[4] YI X, OKAMOTO E. Practical Internet voting system[J]. Journal of Network and Computer Applications, 2013, 36(1): 378-387.

[5] GABBER E, BIBBONS P, MATIAS Y. How to make personalized Web browsing simple, secure, and anonymous[A]. Financial Cryptography ’97[C]. Anguilla, UK, 1997. 17-31.

[6] JAKOBSSON M, RAIHI D. Mix-based electronic payments[A]. Proceedings of SAC ’98[C]. London, UK, 1998. 157-173.

[7] JAKOBSSON M. Flash mixing[A]. Proceedings of PODC ’99[C]. Atlanta, Georgia, USA, 1999. 83-89.

[8] JAKOBSSON M, JUELS A. An optimally robust hybrid mix network[A]. Twentieth ACM Symposium on Principles of Distributed Computing[C]. New York, NY, USA, 2001. 284-292.

[9] GOLLE P, ZHONG S, BONEH D, *et al*. Optimistic mixing for exit-polls[A]. Advances in Cryptology-Asiacrypt ’02[C]. Queenstown, New Zealand, 2002. 451-465.

[10] WIKSTRÖM D. A sender verifiable mix-net and a new proof of a shuffle[A]. Advances in Cryptology-Asiacrypt ’05[C]. Chennai, India, 2005. 273-292.

[11] ABE M. Flaws in robust optimistic mix-nets and stronger security notions[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2006, 89(1): 99-105.

[12] WIKSTRÖM D. Five practical attacks for “optimistic mixing for exit-polls”[A]. Proceedings of Selected Areas of Cryptography (SAC)[C]. Ottawa Canada, 2003. 160-174.

[13] MITOMO M, KUROSAWA K. Attack for flash mix[A]. Proceedings of ASIACRYPT 2000[C]. Kyoto, Japan, 2000. 192-204.

[14] LI L H, FU S F, CHE X Q. A new relation attack on the optimistic mix-net[A]. International Symposium on Computer Network and Multimedia Technology(CNMT 2009)[C]. Wuhan, China, 2009. 1-4.

[15] PENG K. Failure of a mix network[J]. International Journal of Network Security & Its Applications, 2011, 3(1): 81-97.

[16] WIKSTRÖM D. A universally composable mix-net[A]. 1st Theory of Cryptography Conference[C]. 2004. 315-335.

[17] WIKSTRÖM D, GROTH J. An adaptively secure mix-net without erasures[A]. 33rd International Colloquium on Automata, Languages and Programming[C]. 2006. 276-287.

[18] CAMENISCH J, MITYAGIN A. Mix-network with stronger security[A]. 5th International Workshop on Privacy Enhancing Technologies[C]. Cavtat, Croatia, 2005. 128-146.

[19] KHAZAEI S, MORAN T, WIKSTRÖM D. A mix-net from any CCA2 secure cryptosystem[A]. Advances in Cryptology – ASIACRYPT 2012[C]. Beijing, China, 2012. 607-625.

[20] JAKOBSSON M, JUELS A, RIVEST R. Making mix nets robust for electronic voting by randomized partial checking[A]. Proceedings of USENIX'02[C]. San Francisco, USA, 2002. 339-353.

[21] FURUKAWA J, SAKO K. An efficient scheme for proving a shuffle[A]. Advances in Cryptology- Crypto’01[C]. Santa Barbara, California, USA, 2001. 368-387.

[22] NEFF A. A verifiable secret shuffle and its application to E-Voting[A]. Proceedings of ACM CCS ’01[C]. Philadelphia, Pennsylvania, USA, 2001. 116-125.

[23] GROTH J. A verifiable secret shuffle of homomorphic encryptions[J]. Journal of Cryptology, 2010, 23(4): 546-579.

[24] CANETTI R. Universally composable security: a new paradigm for cryptographic protocols[A]. 42nd IEEE Symposium on Foundations of Computer Science[C]. NY, USA, 2001. 136-145.

[25] BELLARE M, ROGAWAY P. Random oracles are practical: a paradigm for designing efficient protocols[A]. Proceedings of ACM CCS’ 93[C]. Fairfax, Virginia, USA, 1993. 62-73.

[26] FIAT A, SHAMIR A. How to prove yourself: practical solutions to identification and signature problems[A]. CRYPTO 1986[C]. 1986. 186-194.

[27] PEDERSEN P. Non-interactive and information theoretic secure verifiable secret sharing[A]. Advances in Cryptology-Crypto'91[C]. Santa Barbara, California, USA, 1991. 129-140.

[28] DOLEV D, STRONG H. Authenticated algorithms for byzantine agreement[J]. SIAM Journal on Computing, 1983, 12(4): 656-666.

[29] SHOUP V, GENNARO R. Securing threshold cryptosystems against chosen ciphertext attack[J]. Journal of Cryptology 2002, 15(2) : 75-96.

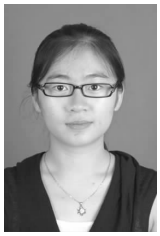
[30] KHAZAEI S, WIKSTRÖM D. Randomized partial checking Revisited[A]. Proceedings of the 13th International Conference on Topics in Cryptology[C]. San Francisco, CA, February, 2013. 115-128.

[31] KÜSTERS R, TRUDERUNG T, VOGT A. Formal analysis of chaumian mix nets with randomized partial checking[A]. Proceedings of the 2014 IEEE Symposium on Security and Privacy[C]. Washington, DC, USA, 2014. 343-358.

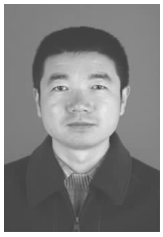
**作者简介：**

**李龙海**（1976-），男，河北冀州人，博士，西安电子科技大学副教授、硕士生导师，主要研究方向为匿名通信、隐私保护技术和计算机网络安全。

**黄诚强**（1989-），男，福建福州人，西安电子科技大学硕士生，主要研究方向为计算机与网络安全。



**许尚妹**（1990-），女，浙江杭州人，西安电子科技大学硕士生，主要研究方向为计算机与网络安全。



**付少锋**（1975-），男，陕西户县人，西安电子科技大学副教授，主要研究方向为计算机网络安全和嵌入式系统。