# Research of Encryption Techniques on Anonymous Voting System

XU Jing, WANG Zhen-xing, WANG Qian

*National Digital Switching System Engineering and Technological Research Center, NDSC, Zhengzhou, China*

*xujingcwz@126.com*

**Abstract:** A strong anonymous voting system based on secret sharing is design in this paper. The project adopts both symmetric encryption and asymmetric encryption algorithms to ensure and increase security of communication content base on routing idea of Crowds system. When the project complete to anonymity communication select reasonable routing, it also complete to the receiver secret share distribution management; and the validity of share is affirmed by hash functions，and receiver server complete the identity comeback and count. Theoretic analysis and experiment shows this voting system has better performance and anti-attack capability; it's a provided with wide application value anonymous voting system. .

**Keywords:** anonymous vote; secret sharing; hash functions; anti-attack

# 匿名投票系统中加密技术研究

**徐静，王振兴，王倩**

国家数字交换系统工程技术研究中心，郑州，中国，450002

xujingcwz@126.com

**【摘要】**本文基于秘密共享技术提出了一种高匿名度的匿名投票方案。该方案基于 Crowds 系统的路由思想，采用对称加密和非对称加密两种算法来保证投票者的匿名身份，在完成匿名通信合理选择路由的同时，也完成了接收方服务器秘密份额的分发管理；在确认投票终端身份和数量时选取 HASH 算法验证秘密份额的正确性，并由接收方服务器完成对其身份的恢复和统计。理论分析和实验表明，该匿名投票系统具备较好的抗攻击能力和通信性能，是一种具有广泛应用价值的匿名投票系统。

**【关键词】**匿名投票；秘密共享；HASH 算法；抗攻击

## 1 引言

1981年Chaum首先发表了关于匿名通信的开创性文章[1]，同时也提出了基于Mix net的投票方案，它能够保证投票者的匿名性。但是Chaum所提出的Mix net方案存在安全上的漏洞，同时其匿名通信的效率不高。随后又出现了一系列基于匿名通信技术的投票系统[2]。文献[3]是一种利用门限思想实现的投票系统，并利用零知识来证明用户投票的有效性。文献[4]则将秘密共享方案与盲签名体制结合起来实现了电子投票方案。

国家 863 项目：信息安全目标导向类课题（Information Security goal-oriented type of issue）；项目名称：网络信息系统安全测试评估体系框架与支撑平台(Network Information Systems Security Assessment and Test System Framework and Support Platform, No.2006AA01Z449)。

1998年Bell实验室AT&T提出了Crowds系统，为用户提供匿名Web浏览能力。其基本思想是把Web用户组成一个称为crowd 的群体，用一系列互相合作的代理来维持用户组中的匿名性。需要匿名保护的主机必须加入系统成为成员，在被保护的同时也提供匿名服务。当成员需要发起一次匿名通信时，将请求转发给其他成员，其他成员将该请求转发或直接递交给Web服务器。当群体中的成员收到另一个成员发送的请求时，将基于概率选择算法随机地转发，随后的该请求信息及服务器端的应答都沿着这条路径传递。由于Crowds 采用随机转发，没有过多的消息开销，因而其效率高，可扩展性好。但Crowds 系统的主要设计目的是抵制Web



978-1-935068-01-3 © 2009 SciRes.

服务器获得访问者的真实地址，并不能抵御通信流分析。对于拥有监听整个网络流量能力的攻击者，它的匿名性无法得到较好保障，而且Crowds 系统只提供发送者匿名，并不提供接收者匿名和关系匿名。Crowds的设计者Reiter和Rubin还提出了匿名等级的概念，他们把匿名分成了六个等级，并在此基础上对Crowds系统的匿名性进行了分析。

本文借鉴了 Crowds 系统的路由思想[5]和秘密共享算法[6-8]，在保证效率和抵御通信流分析的同时，完成投票秘密份额的分发和分布存储。其主要流程如下：在投票发起时，由投票发起者负责生成秘密份额和挑选投票协作者；开始投票后，投票发起者向下一个投票协作者发送相关信息，同时以一定概率决定是否向投票服务器发送信息；每个收到信息的投票协作者基于概率决定将信息转给下一个协作者或相邻节点，同时以另一概率决定是否向投票服务器发送消息；非协作者收到信息后直接将其转交下一个协作者，但不向投票服务器发送消息；投票服务器收到足够多的信息后，可以恢复和统计出投票结果。

## 2 方案描述

AT&T的 Crowds协议是用一系列互相合作的代理来维持用户组中的匿名性。需要匿名保护的主机必须加入系统成为成员，在被保护的同时也提供匿名服务。当成员需要发起一次匿名通信时，将请求转发给其他成员，其他成员将该请求转发或直接递交给Web服务器。

传统的Crowds系统中每个成员上运行一个名为jondo的代理程序，用于转发自本地浏览器或者其他成员上 jondo的 HTTP请求。初始时，jondo向系统中行使管理功能的成员（称为blender）注册，并获得系统中的活动jondo列表及相应的共享密钥。当收到来自本地浏览器的HTTP请求时，jondo从jondo列表中随机选取一个作为后继（可能是它自己），并将请求转发给该后继jondo。当后继jondo接收到请求后，以概率pf（1/2<pf<1）将该请求继续转发，否则以概率1-pf将请求直接提交给接收者。也就是说，发送给系统中任何一个jondo的可能性是pf，而发送给服务器的概率是1-pf。每一步跳都决定是否将HTTP请求直接交给目的服务器或者根据转发可能性将它转发给下一个随机选择的成员，重路由路径中允许存在循环。一旦选择了一条重路由路径，在24小时的时段内，所有从该发送者到该接收者的匿名通信都将使用这条路径。新的成员可以在一些特定的时段加入Crowds并形成新的路径。

设定初始阈值 $k$，协商一个足够安全的 *hash* 函数。任何一个拟参与投票的用户（作为投票发起者或者投票协作者）在进入网络时，向网络中广播所持有的公钥。为了方便讨论，本文约定：$E_{key}^{(1)}$ 和 $E_{key}^{(2)}$ 分别表示加密密钥为 $key$ 的对称加密算法和非对称加密算法，$D_{key}^{(1)}$ 和 $D_{key}^{(2)}$ 分别表示解密密钥为 $key$ 的对称解密算法和非对称解密算法。方案的参与者包括投票服务器 SE、投票发起者 SP 和投票协作者集合 U。

### 2.1 SP 生成报文阶段

SP 发起投票前，从网络中任意选取若干个用户，不妨设该用户集合为 $\{U_1,\cdots,U_n\}$，约定用户集内用户的公钥依次为 $e_1,\cdots,e_n$，私钥依次为 $d_1,\cdots,d_n$，SE 的公钥为 $es$，SE 的私钥为 $ds$。SP 为完成匿名投票，执行以下操作：

①根据初始阈值 $k$，SP 随机生成一个 $k$-1 次多项式（$b$ 即是投票结果）：$f(x)=\left(a_1x^{k-1}+a_2x^{k-2}+\cdots+a_{k-1}x+b\right)$以及随机数 $r_0,r_1,r_2,\cdots,r_n$。

SP 计算 $f(r_0),f(r_1),\cdots,f(r_n)$。

②SP 生成 $2(n+1)$个位于[0-1]区间内的随机数，分为两组：转发下一跳概率集 $P=\{p_0,p_1,p_2,\cdots,p_n\}$ 和递交投票概率集 $Q=\{q_0,q_1,q_2,\cdots,q_n\}$，要求

$$n\sum_{i=0}^{n}q_i \succ k。$$

③SP 将随机数集合：$R=\{r_0,r_1,r_2,\cdots,r_n\}$中各数次序随机扰乱，得到新的随机数集合 $R'=\{x_0,x_1,x_2,\cdots,x_n\}$

③SP 生成发送报文。设 SP 所选定的路径为：$SP\rightarrow U_1\rightarrow\cdots\rightarrow U_n$。

SP 计算子报文：$flow_1=E_{en}^{(2)}(r_n),E_{rn}^{(1)}(null,p_n,q_n,f(r_n),x_n,hash(x_n),hash(f(x_n)))$，*null* 表示下一跳为空，即不需要再向下传递；同时要求加密前填充待加密流，以保证密文长度固定。在 $flow_1$ 的基础上，按以下方式计算子报文 $flow_2$：首先填充待加密流，以保证经过每一轮计算，报文长度增加幅度不变；然后计算 $E_{e(n-1)}^{(2)}(r_{n-1})$；

令 $\nabla^1=E_{r(n-i)}^{(1)}(U_{n-i+1},p_{n-i},q_{n-i},f(r_{n-i}),x_{n-i},hash(x_{n-i}),hash(f(x_{n-i})),flow_i)$；

计算出 $\nabla^1$，最后得到：

$$flow_2=\left(E_{e(n-1)}^{(2)}(r_{n-1}),\nabla^1\right);$$

依此类推，经多轮计算得到：

$flow_n = E_{e1}^{(2)}(r_1), \nabla^{n-1}$，此即为最终待发送的报文。

④SP 基于概率 $p_0$ 决定是否将报文 $flow_n$ 直接向 $U_1$ 转发，或者向任一相邻节点转发。

⑤SP 基于概率 $q_0$ 决定是否向投票服务器 SE 发送报文。如果需要的话，则随机延迟若干时间后发送报文：

$flow = E_{es}^{(2)}(r_0, f(r_0), x_0, hash(f(x_0)))$。

### 2.2 投票协作者转发报文阶段

投票协作者 $U_i$ 收到报文：

$flow_{n-i} = (E_{e(n-i)}^{(2)}(r_{n-i}), \nabla^i)$ 后进行如下操作：

①记录报文长度为 *length*;

②利用已方私钥解密 $E_{e(n-i)}^{(2)}(r_{n-i})$，恢复 $r_{n-i}$；

③利用 $r_{n-i}$ 解密，计算 $D_{r(n-i)}^{(1)}(\nabla^i)$，恢复出 $U_{n-i+1}$，$p_{n-i}, q_{n-i}, f(r_{n-i}), x_{n-i}, hash(x_{n-i}), hash(f(x_{n-i}))$，$flow_i$；

④$U_i$ 根据 $x_{n-i}$ 计算 $hash'(x_{n-i})$，检查与解密所得的 $hash(x_{n-i})$ 是否相等，如果对称加密算法扩散性能良好，可以发现该报文是否被篡改。如果被篡改，$U_i$ 发送报警报文(长度与正常报文一致)给 SE，并终止下传。SE 在其他用户只读公告牌内公开出错节点信息，供 SP 重启投票过程时参考。

⑤$U_i$ 基于概率 $q_{n-i}$ 决定是否向投票服务器 SE 发送报文。如果需要的话，则经过随机时间延迟后，发送报文：

$flow = E_{es}^{(2)}(r_{n-i}, f(r_{n-i}), x_{n-i}, hash(f(x_{n-i})))$。

⑥$U_i$ 填充报文 $flow_{n-i+1}$，将报文长度恢复至 *length*，基于概率 $p_i$ 决定是否将 $flow_{n-i+1}$ 直接转发给下一跳，或者转发给相邻任意节点(这些节点接到该报文后直接转发，不另作处理)。

### 2.3 SE 恢复投票结果阶段

$SE$ 接收到足够多的报文后，基于拉格朗日插值公式对其进行恢复，获取投票结果。由于投票协作者中可能存在着恶意攻击者或者在传送过程发生篡改，需要对投票结果进行检测，在检测的基础上进行恢复和验证。其过程如下：

①SE 对投票协作者或投票发起者发来的报文计算 $D_{ds}^{(2)}(flow)$，获取报文内容。

② 令 $\partial_i = t_i(x_i, hash(f(x_i)))$，其中 $t_i$ 取值为 0 或 1，当其为 0 时说明 SE 未收到秘密份额 $(x_i, f(x_i))$ 的校验值；SE 根据所接收的报文，建立校验表 $\{\partial_0, \partial_1, \cdots, \partial_n\}$。

③SE 根据校验表检查 $(r_i, f(r_i))$ 是否正确。校验前提是收到 $(r_i, f(r_i))$ 的校验值 $(x_m, hash(f(x_m)))$，因为每个投票协作者基于概率提交报文，所以存在着校验值未发送的可能性。校验流程如下：在校验表中对 $r_i$ 执行匹配操作。若 $r_i = x_m$，则比较校验表中相应的 $hash(f(x_m))$ 与 $hash(f(r_i))$，如果两者相等，，则说明该秘密份额基本正确（这种秘密份额的假冒概率极小，例如在 10000 个用户的 P2P 网中存在 10 个攻击者，SP 挑选 10 个用户作为协作者，并且 SP 所挑选的发送校验值协作者和发送秘密份额的协作者均为攻击者时才能导致攻击成功，其概率约为 $10^{-7}$，可以忽略)。

④若通过校验的秘密份额数大于或等于 $k$，则按拉格朗日插值办法多次恢复投票结果；若小于 $k$，则从无法校验的秘密份额集合中多次随机挑选若干个进行尝试恢复(第一种情况仅恢复两次确认即可，后一种情况需要所有的秘密份额至少参与恢复操作一次)。多次恢复的结果一致时，则恢复成功；否则，采用大数表决方式来决定恢复结果。如果仍然无法确定恢复结果(例如在极端情况下，绝大多数秘密份额均是伪造的)，则恢复失败， SP 需要重新发起投票过程。

⑤若 SE 恢复投票结果失败，在其它用户的只读公告牌内公布校验正确的随机数序列 $\{r_{t1}, r_{t2}, \cdots\}$。SP 据此可了解哪些节点传输的报文内容正确，进而结合所选路由来判断哪些节点是正常节点，而非攻击节点。秘密份额无法通过校验的原因有三种：秘密份额提交者是攻击者，秘密份额校验值提交者是攻击者，以及上游节点是攻击者。但是在网络环境中 SP 有众多可挑选的节点，不需耗用过多的计算能力进一步判别其中的攻击者和正常用户。

⑥ $SE_1$ 对恢复的结果查看投票票据 $(ID, R_i)$ 是否无误，正确即统计投票结果。当所有该批次的投票票据均已收集，$SE_1$ 公开投票结果。

## 3 方案分析

### 3.1 匿名度分析

匿名度 (Degree of Anonymity)是衡量系统提供的匿名保护有效性的尺度。目前关于匿名度的计算尚未形成统一的标准，总体上可以从定性和定量两个角度分析。从定量的角度来说，设 $P_1(x)$ 表示 x 为发送者的概率，其中实体 x 为一个由匿名系统内各实体组成的集合 S 中的一个元素，于是：

$$\sum_{y \in s} P_1(y) = 1$$

成员x匿名度为$d_x$，那么：



978-1-935068-01-3 © 2009 SciRes.

$$d_x = \sum_{y \in S \neq x} P_1(y), \quad d_x = 1 - P_1(x)$$

若S中各实体为发送者的概率相等的话，则：

$$d_x = 1 - \frac{1}{|s|}$$

从定性的角度来说，匿名度可以看成是一段连续的区域。整个连续区域根据提供匿名程度由强至弱的顺序可以分成以下六个等级：

- absolute privacy：表示攻击者无法推断出真实的发送者，即|S|=∞，$d_x$=1。
- beyond suspicion：表示实体x不比系统内其他实体更像发送者，即$d_x$>$d_y$。
- probable innocence：表示实体x比系统的其他实体更像发送者，但实体x作为发送者的概率不高于作为非发送者的概率，即1/2<$d_x$<$d_y$。
- possible innocence：同probable innocence。
- exposed：表示实体x不为发送者的概率小于1/2，即$d_x$<1/2。
- provably exposed：实体x比系统的其他实体更像发送者，且实体x作为发送者的概率高于作为非发送者的概率，即$d_x$=0。

本方案是基于 Crowds 系统的路由机制加以改进。对主要数据采用点对点临时会话密钥和对称加密算法进行加解密运算，减少了各节点计算量；同时各节点的临时会话密钥与秘密共享的秘密份额结合使用，临时会话密钥既是密钥也是秘密份额。各节点均按随机概率 $q_i$，且经随机延迟后向 SE 发送报文，攻击者难以从发送给 SE 中的报文流量中分析出 SP。

针对匿名通信的主要攻击方式之一是前趋攻击，即攻击者通过频繁致使通信失效，造成通信者多次重新发起通信，由此猜测出通信者。减少前趋攻击的一种方法是降低攻击者冒充正常用户的可能性。本方案在通信失败后，由 SE 公布某些正确节点信息，为 SP 重新选择节点发起投票提供了参考，从而有效地回避攻击者，减少攻击者的冒充行为。

在进行安全分析之前，假设本方案的外围协议已经保证了基本的防篡改、防重放等攻击。攻击者无法通过嗅探等被动攻击方法获得秘密信息，因为每个报文均采用对称和非对称两种加密算法进行加密。只要加密算法是安全的，攻击者就无法获得秘密信息。攻击者被选作投票协作者时只能获知以下参数：$p_{n-i}$，$q_{n-i}$，$f(r_{n-i})$，$x_{n-i}$，$hash(x_{n-i})$，$hash(f(x_{n-i}))$，$flow_i$。

其中，前两个参数是节点固有属性，仅在该节点范围内有效。基于其他参数，攻击者只能获得一个秘密份额和一个秘密份额校验值，以及向下传递的数据。秘密共享算法保证了攻击者只有一份秘密份额时无法恢复投票结果。攻击者虽然可以篡改数据，但破坏秘密份额值$(r_{n-i}, f(r_{n-i}))$或校验值$hash(x_{i+1}, f(x_{i+1}))$只能增加恢复操作的计算量，而不会导致投票结果泄露；如果破坏下传数据 $flow_i$，将导致后继的秘密份额均失效，使 SE 无法获得足够的秘密份额来恢复投票结果，从而导致重启投票，但是 SP 可根据 SE 公布的信息，在重启投票选择路由时避开攻击者，从而使攻击失效。

### 3.2 正确性分析

此处正确性分析的含义是证明 SE 在本方案的框架内能正确收到 SP 的投票结果 $b$。本方案基于秘密共享方案来完成投票，分发者是 SP，参与者是 SP 和 U，秘密恢复者是 SE。每个参与者以概率 $p_i$ 向 SE 发送秘密份额，在正常情况下，SE 接收的秘密份额份数的期望值大于阈值 $k$，能保证 SE 有足够的秘密份额来恢复 $b$。SE 对接收的秘密份额进行正确性校验，如果通过校验，那么这些秘密份额可以认为是真实正确；无法校验的秘密份额也存在着是正确份额的可能性；而未通过校验的秘密份额仍是正确份额的可能性最小。SE 基于这三种秘密份额，可以通过较少的计算开销进行启发式的多次恢复，采用大数表决方式可以得到正确的秘密份额。

### 3.3 抗攻击性能分析

本方案的抗攻击特性主要表现在两个方面：投票信息上报子方案和投票信息结果统计子方案。

在投票信息上报子方案中，攻击者可能会产生如下几种攻击：篡改 $upflow$ 或 $flow$、重放 $upflow$ 或 $flow$、截断 $upflow$ 或 $flow$。重放不会对方案的运行造成过多影响，只会消耗计算能力。篡改 $upflow$ 或 $flow$ 以及截断 $upflow$ 或 $flow$ 会造成部分秘密份额失效。相对而言，篡改或截断 $upflow$ 的影响较小，一份 $upflow$ 只会影响到两份份额；而篡改或截断 $flow$ 的影响较大，有可能导致多份份额被破坏或者无法传递。但是考虑到攻击者数量与 P2P 网络规模而言，以及本方案对攻击者可以有效发现的特性而言，这样的情况出现的概率较小。在投票信息上报子方案中，只要有足够多的份额能正确到达 $SE$，就不会影响方案的正确运行。

为了防止投票服务器成为攻击者的焦点，也是为了让系统不存在单点失效的威胁，在投票信息结果统

计子方案中，份额存放于 $SE$ 中多个不同服务器中，这样可以有效降低攻击者通过控制投票服务器，达到从内部攻击方案的可能，只要有足够多的投票服务器正常工作，攻击者就无法篡改投票结果。

## 4 结束语

本文借鉴了 Crowds 系统的路由思想，设计了一种高效、正确、抗攻击的匿名投票算法。在保证发送者匿名的前提下简化了密钥管理过程，使得密钥管理更加安全方便，同时该方案与秘密共享算法的思想有机结合，将投票结果隐藏在多个秘密份额中，增强了安全性和匿名性，非常适合用于网上调查、投票等网络应用系统。

## References（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] K Ohta. An electrical voting scheme using a single administrator [A]. 1988 Spring National Convention Record. Berlin:IEICE, 1988.

[3] Gramer R. Gennaro R, Schoenmakers B. A Secure and Optimally Eifficient Multi-authority Election Scheme. In European Transactions on Tele- communications, 1997, 8 (5):481−490.

[4] Wen-Shenq Juang ,Chin-Laung Lei and Pei-Ling Yu, A Verifiable Multi-Authorities Secret Election Allowing Abstaining from Voting, International Computer Symposium (ICS’98), Tainan, Taiwan, December 1998.

[5] M.K.Reiter and A.D.Rubin. Anonymous Web Transactions with Crowds. Communications of the ACM, 42 (2):32−38, February 1999.

[6] K.J.Tan, H.W.Zhu, S.J.Gu. Cheater identification in (t,n) threshold scheme [J]. Computer Communication, 1999,22(8)：762−765.

[7] Okada K, Kurosawa K. MDS secret sharing scheme secure against cheaters [J]. IEEE Transactions on Information Theory, 2000, 46(3)：1078−1081.

[8] C.C Yang, T.Y.Chang, Hwang M.S. A (t,n) multi-secret sharing scheme [J]. Applied Mathematics and Comutation, 2004,151(2)：483−490.



978-1-935068-01-3 © 2009 SciRes.