计算机科学2002Vol. 29№. 4

电子选举:理论、实践与未来

Electronic Voting:Theory,Practice and Future

郑燕飞　陈克非

(上海交通大学计算机系　上海200030)

Abstract Electronic voting is a method using modern network technology to implement the election. Much cryptology knowledge applies to the protocol of electronic voting, but due to the security and anonymity of the voting procedure, the application of it is still in the research phase. This paper introduces different electronic voting protocols. On the basis of expatiating and trimming foreign and demotic research on electronic voting protocol, it does detailed research on the anonymity of the voting procedure . At the same time, this paper also points out the problem and difficulty in the research of electronic voting , and vistas the research foreground.

Keywords Electronic selection, Electronic voting, Anonymous channel, Multi-party compute

1. 引言

当今社会,几乎我们身边的每一件事都可以被计算机实现,尤其是随着近几年来 WWW 的几乎爆炸式的发展,通讯和网络技术的突飞猛进,我们的生活和工作都出现一种不可避免的趋势——无纸化、自动化。选举,作为社会公民所享有的一项基本权利,它的方式本身也不可能在这样一个日新月异的社会环境中停滞不前,因此,电子选举应运而生。电子投票的发展从它在1884年第一次被人提出到现在这近百年并不是一帆风顺的,其主要原因是一个电子投票系统的安全性和公平性是很难得到保证的。直到密码学突破军事应用的限制,在各领域得到广泛的应用以及密码学本身的飞速发展,电子选举才有了可喜的进展。

本文介绍了电子选举出现以来的几种较典型的具有代表性的协议,阐述作者对目前国内外的研究成果和已实现系统的看法和观点。电子选举协议是对密码学领域内多种密码学知识的一种综合应用,它涉及 RSA、数字签名、匿名信道、零知识证明、安全多方计算等方面。

一个电子选举协议至少应达到现实生活中一个普通投票系统的安全性质,因此一个电子选举协议应满足下列基本性质:

1)计票的完整性,即所有的有效投票都能被正确计入。

2)投票过程的稳固性,即不诚实的投票者不会破坏整个投票。

3)选票内容的匿名性,即每个投票者的投票内容不会被除本人以外的任何个人或组织(包括负责记票的记票中心)得知。

4)不可复用性,即每个投票者不能把自己的选票投寄两次。

5)可证实性,即任何人都不可能伪造选举结果,选举的结果可以被大家证实。

除此以外还有一些附加特性可以使得电子投票协议更加满足社会的需要,比如一个具有无收条性质的电子投票协议可以很好地防止投票者凭借投票后收到的收条来证明自己投的什么票以此达到卖选票的目的。在这种方案中先由一个可信中心为每一个投票者随机地产生一对包含"yes/no"和"no/yes"的选票,运用零知识证明的方法,可信中心向大家证明他产生的的确是有效选票,并通过可变色点的比特承诺告知投票者为他产生的选票对是按怎样的顺序发送给他的。而可变色点的可变色性使得投票者在拿到选票后既能伪造证据证明自己收到的是"yes/no",又能伪造证据证明自己收到的是"no/yes"对[10],因此投票者就不能取得买选票人的信任使他相信自己投的是什么票。

2. 电子选举协议的现状

2.1 电子选举协议的分类

电子选举协议的策略根据整个选举过程中是否有可信第三方的参与可分为两大类:自我判决的协议和有可信第三方参与的协议。有可信第三方参与的协议根据其匿名性和安全性的实现方法的不同又可分为使用匿名信道的投票协议和利用多方计算的投票协议。

2.2 自我判决型协议

这种协议不需要除投票者以外的参与者,投票的安全性由投票者的多次加密和签名来实现。匿名性由算法中对选票的多次重排序来实现,这种类型投票协议的步骤如下:

(每个投票者都有自己的公钥)

1. 每个投票者为自己的投票 V 附加一个随机数 R

2. 对(V,R)用每个投票者的公钥依次加密

3. 再对此加密结果用每个投票者的公钥依次加密,但在每次加密前都附加一个随机数

假设有 N 个投票者,那么此时加密结果如下:

$E_n(R_n,E_{n-1}(\cdots E_1(R_1,E_n(E_{n-1}(\cdots E_1(V,R)\cdots)))\cdots))$

4. 从投票者 N 到投票者1依次把所有的选票解密,抽取出自己的随机数后,把余下的选票部分搅乱次序,发给下一个投票者,当投票者1把所有的选票解密后,选票的形式如下:

$E_n(E_{n-1}(\cdots E_1(V,R)\cdots))$

5. 从投票者 N 到投票者1再依次把所有的选票解密,但这次投票者 I 把选票解密,再对解密结果签名后发送给投票者 I−1,投票者 I−1先检验投票者 I 的签名,如果是有效的,再对其进行解密,签名,发送给下一个投票者

此过程中的选票形式如下:

$S_i(E_{i-1}(\cdots E_1(V,R)\cdots))$

6. 投票者1的签名由所有的投票者来验证,并且检查选票列表中的随机数以确定自己的选票被正确计入。

在这种投票过程中，投票者的总数保持不变，投票者不会被恶意的攻击者替代，如果投票者被替代，那么在第一轮的解密过程中他将被发现，并且根据签名追踪找到恶意攻击者；投票者每次乱序所有的投票使得投票的保密性得以实现；随机数 R 的使用使得每个投票者能够检验自己的投票是否被正确计入。

2.3 有可信第三方参与的协议

这种类型的协议不仅仅有投票者参加，还有其他如认证中心、记票中心等第三方参加整个投票过程，这些中心的参与可以节约大量的计算成本和通信成本。例如认证中心可以对投票者的身份进行认证，确认其投票的权利，投票者只有拿着认证中心签名的选票才可以进行投票，这样也从一定程度上减小了恶意攻击的可能性。而这种有第三方加入的投票协议也更接近现实生活中的真正投票过程，因此有第三方的投票协议成为现在电子投票协议的主流，但是第三方的加入同时也带来了许多安全隐患，尤其是在保护投票者的隐私和其选票的保密性方面和防止投票组织者伪造选票方面，这就要运用各种密码学例如盲签名、匿名信道、安全多方计算来达到一个现实生活中普通的投票过程所具有的安全特性。

2.3.1 有可信第三方参与的选举协议的主体框架

在电子选举中最著名的投票方案是1994年由 Fujioka，Okamoto 和 Ohta 提出的[1]，这种投票方案至今仍是各种投票协议设计的主要框架。下面简单描述它的实现过程：

1. 每一个投票者向认证中心提供自己的身份证明，索取认证码

2. 认证中心随机地产生认证码并安全分发它们

3. 投票者在投票时任意选择一个随机数作 ID、和自己的选票、认证码一起加入盲签因子用认证中的公钥加密后发给认证中心

4. 认证中心用自己的私钥解密，验证投票者的身份，如果是合法投票者则为其选票签名，发送给投票者

5. 投票者去掉自己加入的盲签因子，验证认证中心的签名，用计票中心的公钥加密后通过匿名信道发送给计票中心

6. 计票中心用私钥解密，验证认证中心的签名，公布有效选票，计数并公布结果

在这个协议中不需要每个投票者都有自己的公钥，但是所有的通信需要会话密钥加密，此协议可以在认证中心验证选票所有者身份时有效地防止选举者多次投票，因为投票者的盲因子的存在使得即使认证中心和计票中心联合起来也不能获得选票的内容；而匿名信道防止了计票中心对投票者的追踪；在协议的每次通信过程中都应使用消息认证码防止恶意攻击者在窃听过程中修改选票内容。

2.3.2 利用匿名信道实现投票的匿名性

从上面描述的投票过程中可以看出匿名信道对于保证选票的匿名性的重要性，匿名信道是通过多个被称作 MIX 的转发中心遵从一定的协议来实现，在文[8]中定义了一种 A Mix-Type Anonymous Channel，这种匿名信道在实现匿名转发的同时可实现个人验证，即每个投票者在通过此匿名信道转发后仍然可验证自己的选票被正确计入，建立在这种 Mix-Type 的匿名信道基础上的可实现带有普遍认证性质的匿名信道，具体方法见文[9]。所谓普遍认证是指任何个人或中心包括没有参加投票的人在内都可以对选票者的投票进行验证，确认所有投票的投票者被匿名信道正确转发。

假设匿名信道是由 n 个 center 来实现的，定义参数如下：

Public Information: $p=kq+1$ (p,q prime)
: $g=(g')^k \bmod p$ (g' is a generator mod p)
public key of center I : $y=g^{x_i} \bmod p$
secret key of center I : $x_i \in Z_q^*$
Message from the sender : m
Define $w_i = y_{i+1}\, y_{i+2} \cdots y_n$ and $w_n=1$
Center I compute G_{i+1}, M_{i+1} on input G_i, M_i，如下：
$G_{i+1}=G_i \cdot g^{r_i} \bmod p$
$M_{i+1}=M_i \cdot w^{r_i}/G_i^{x_i}$

当 center I+1收到 center I 发给它的 G_i，M_i，它第一步先计算并公布 G_{i+1}，并利用零知识证明 G_{i+1}的正确性；第二步计算并以乱序公布 G_{i+1}和 M_{i+1}，这时它要在乱序的条件下向所有的人证明它的确是正确计算 M_{i+1}，并没有篡改投票者的选票内容，这是通过利用置换群的零知识证明来实现的。其过程简单描述如下。设：

$$A=\begin{pmatrix} a_i^{(1)} \\ a_i^{(2)} \end{pmatrix} \quad B=\begin{pmatrix} a_{\pi(i)}^{(1)} g^{r_{\pi(i)}} \bmod p \\ a_{\pi(i)}^{(2)} w^{r_{\pi(i)}} \bmod p \end{pmatrix}$$

其中 $a_i^{(1)}$代指前面的 G，$a_i^{(2)}$代指前面的 M/H，$r_1, r_2, \cdots$是由 center 产生的随机数，π 是一个置换。

下面证明 B 的确是由 A 以前面定义的方式产生的，但不暴露 π 的置换信息，这是一个交互式的零知识证明过程。

1. 证明者随机选取 $t_i \in Z_{p-1}$和一个置换 λ，计算并公布：

$$C=\begin{pmatrix} a_{\lambda(i)}^{(1)} g^{t_{\lambda(i)}} \bmod p \\ a_{\lambda(i)}^{(2)} w^{t_{\lambda(i)}} \bmod p \end{pmatrix}$$

2. 有1/2的可能性，验证者要求证明者出示 λ 和 t_i，验证者检查 C 是否和 A，λ，t_i，一致

3. 有1/2的可能性，验证者要求证明者出示 $\lambda' = \lambda o \pi$ 和 $t_i' = t_i - r_i$. 验证者通过计算

$$C=\begin{pmatrix} b_{\lambda(i)}^{(1)} g^{t'_{\lambda(i)}} \bmod p \\ b_{\lambda(i)}^{(2)} w^{t'_{\lambda(i)}} \bmod p \end{pmatrix}$$

检查是否能由 B 产生 C。

2.3.3 利用多方计算实现投票的匿名性

利用多方的投票协议一样需要一个认证中心对选举者的身份进行验证，并对选票进行签名使其成为有效选票，在投票时的通信过程也同样需要加密并使用认证码，这些与利用匿名信道的投票协议中的处理过程类似，本处不再重复描述。这种协议的主要思想是把选票内容分割为 N 个部分，分发给 N 个投票中心，利用数论知识共同计算出最终的选举结果。

这种协议使用 Benaloh 和 Yung 提出的利用部分兼容的同态加密函数组来实现高效的零知识证明过程，关于部分兼容的同态加密函数的定义可见文[3]，这种加密函数(设为 E)具有性质 $E(x+y)=E(x)E(y)$，利用一个定义在 Z_p^* 上(p 为一个大质数)的部分兼容的同态加密函数可以构造一个有效的交互式的零知识证明过程，用来证明：

$X_1+X_2+\cdots+X_n=a+b$，

$X+Y \in \{1,-1\}$ 其中 $X_1, X_2, \cdots X_n, X, Y$ 是对选票的随机分割

当然这种交互式的零知识证明为了达到一定的安全参数需要进行许多次交互，使用 Fiat-Shamir 在文[4]中的方案可以使这个证明过程简化为非交互式的，以减少证明过程中的通信量和计算量。

下面简单描述一个有两个投票中心的投票协议：

1. 每个投票者 I 选择自己的选票 $V_i \in \{0,1\}$，1代表赞成，0代表反对，然后随机选取 $X_i^{(1)}$和 $X_i^{(2)}$使得：

$X_i^{(1)}+X_i^{(2)}=V_i \bmod q$

2. 投票者 I 公布 $E_1(X_i^{(1)})=a_1^{X_i^{(1)}}$和 $E_2(X_i^{(2)})=a_2^{X_i^{(2)}}$，并在不暴露关于 $X_i^{(1)}$，$X_i^{(2)}$，V_i 的条件下证明 $X_i^{(1)}+X_i^{(2)} \in \{1,$

$-1\}$.其中 E_i 为同态加密函数。

3. 每个投票者分别用投票中心 C 和投票中心 C 的公钥加密 $X_i^{(1)}$ 和 $X_i^{(2)}$。

4. 投票中心收到 $X_i^{(1)}$ 和 $X_i^{(2)}$ 后分别用 E_1 和 E_2 对其加密，与前面投票者公布出来的数字比较，验证其真实性。

5. 每个投票中心 j 把所有的 $X_i^{(j)}$ 加起来模 q 得到 t_j，公布结果，而最终的投票结果是 $T=t_1+t_2$ 在这个过程中每个投票者都可通过计算：

$$E_j(t_j)=\prod_j E_j(X_i^{(j)})$$

来验证投票中心是否正确计票。

这种协议可以很容易扩展到 m 个投票中心，具体的扩展方法见文[2]。在此协议中只要有一个投票中心是诚实的，选票的安全性就可以得到保证。对此协议的攻击需要计算群上的离散对数问题，为了提高安全性，我们还可以使用椭圆曲线上的离散对数。

存在的问题及展望 现在已经有许多关于电子投票的各种特色方案的提出，但是没有一个方案能够满足电子投票的所有安全要求，尤其是在防止投票中心伪造选票这一方面，因此利用安全多方计算的投票协议越来越受到人们的关注，成为目前研究的主流。我们还可以看到无论是利用匿名信道或是安全多方计算的投票协议都要用到零知识证明，而即使把交互式的零知识证明转化为非交互式的，在投票的过程中我们依然面对着较大的通信量和计算量，而对于投票者来说除保密性、匿名性等要求之外，简单方便是用户最希望得到的。因此如何简化用户的投票过程，降低投票协议进行的通讯和计算成本，也是电子选举协议研究中不可忽视的问题。

随着对电子投票方案研究的深入，在国外已经出现了一些实验用的电子投票系统，比如 California 的 Voting By Mail[5]，由 MIT 研究实现的 EVOX[6]，Princeton 大学的校园选举系统，Washington 大学的 Sensus 系统等。其中 EVOX 和 Sensus 系统都是基于前文提到的 FOO 投票协议设计并实现的，不过这些系统的实现都是基于匿名信道的，同样具有对选举的组织机构依赖性过大的缺点，因此基于安全多方计算的电子选举系统有待于我们的开发和实现。可以预见，在不久的将来，随着密码学和网络技术的发展，我们可以研究并实现出真正实用、方便的电子选举系统。

参 考 文 献

1 Fujioka, Okamot, Ohta. A Practical Secret Voting Scheme for Large Scale Elections

2 Kazue Sako, Joe Kilian. Secure Voting Using Partially Compatible Homomorphisms

3 Benaloh J C, Tuinstra D. Distributing the Power of a Government to Enhance the Pirvacy of Voters

4 Fiat A, Shamir A. How to Prove Yourself: Practical Solutions to Identification And Signature Problems

5 Herschberg M. Secure Electornic Voting Using The World Wide Web

6 DuRette B W. Multiple Administrators For Electronic Voting

7 Benalh J C, Turnstra D. Receipt-Free Secret-Ballot Elections

8 Park C, Itoh K, Kurosawa K. All/Noting Election Scheme and Anonymous Channels

9 Sako K, Kilian J. Receipt-Free Mix-Type Voting Scheme - A Practical Solution to The Implementation of a Voting Booth

10 Brassard G, Chaum D, Crepeau C. Minimum Disclosure Proofs of Knowledge

（上接第8页）

（1）从理论方面完善上述的需求描述及检测方法和已研制出的需求工程环境，并且也打算将此环境应用于更多、更复杂的实时嵌入式软件实例中，以使我们的需求描述及检测方法和环境更具有实用性和发挥较大的作用。

（2）将部件化的方法和技术运用到模拟执行中去，严格定义出规则的 DO 部分的语法，解释调用嵌入的组件，探索从需求到代码的可行途径。

参 考 文 献

1 Jahanian F, Mok A K. Safety Analysis of Timing Properties in Real-Time Systems. IEEE Trans. On Software Eng., 1986, 12(9): 890～904

2 Porisini A C, Ghezzi C, Kemmerer R A. Specification of Real-time System Using ASTRAL. IEEE Trans. on Software Eng., 1997, 23(9): 572～598

3 Harel D. Statecharts: A Visual Approach to Complex Systems. Science of Computer Programming, 1987, 8(3): 231～274

4 Harel D, et al. Statemate: A Working Environment for the Development of Complex Reactive Systems. IEEE Transactions on Software Engineering, 1990, 16(4): 403～413

5 Douglass B P. Designing Real_Time Systems with UML. http://www.embedded.com/98/9803fe2.htm.

6 Leveson N G, et al. Requirements Specification for Process Control Systems. IEEE Trans. on Software Eng., 1994, 20(9): 689～707

7 Wu Guoqing, Xiao Haifeng, Zheng Peng, et al. Specifying Requirements of Real-Time System with Rules and Templates. Wuhan University J. of Natural Sciences, 2000, 5(5)3: 278～284

8 Thompson J M, et al. Specification-Based Prototyping for Embedded System. Software Engineering Notes, 1999, 24(6): 163～179

9 Berzing V, Luqi, Yehudai A. Using Transitions in Specification-Based Prototying. IEEE Transaction on Software Engineering, 1993, 19(5): 436～452

10 Kramer B, Luqi, Berzins V. Compositional Semantic of a Real-Time Prototyping Language. IEEE Transaction on Software Engineering, 1993, 19(5): 453～477

11 Stocks P, Carrington D. A framework for specification-based testing. IEEE Trans. on Software Eng., 1996, 22(11): 777～793

12 Atlee J M, Gannon J. State-Based Model Checking of Event-Driven System Requirements. IEEE Trans. on Software Eng., 1993, 19(1): 25～39

13 Heimdahl M P E, Levison N G. Completeness and Consistency in Hierarchical State-Based Requirements. IEEE Transactions on Software Engineering, 1996, 22(6)

14 刘超，张莉编著. 可视化面向对象的建模技术——标准建模语言 UML 教程. 北京航空航天大学出版社, 1999

15 卢梅，李明树. 软件需求工程——方法及工具评述. 计算机研究与发展, 1999, 11(11)

16 毋国庆，胡春丽，蔡持峰，何峰. 面向嵌入式实时系统的需求模型. 计算机工程与科学, 1999(A1)

17 胡春丽，毋国庆，何峰，等. 从状态图到规则的转换. 小型微型计算机系统, 2001(6)

18 毋国庆，朱立松，王敏，蔡持峰. 嵌入式实时系统的软件需求测试. 软件学报(待发表)