



**Universidade de Aveiro**

**Dep. Electrónica, Telecomunicações e Informática**

**Outubro 2006**

# **Anonimato em e-Voting**

**Orientador Prof. André Zúquete**

Mestrado em Eng. Elect. e Telecom.

Carlos Filipe Marques Almeida

# Índice

Robust Electronic Voting System (REVS) .....	1
Anonimato nas comunicações .....	2
Mix-Nets .....	2
Mix Rings .....	3
REVS com Mix Rings .....	4
Modelo do sistema.....	5
Mensagens no Mix Ring .....	7
Tolerância a falhas .....	8
Falha nos <i>Counters</i> de recepção de voto e de envio de recibo .....	8
Referências .....	9

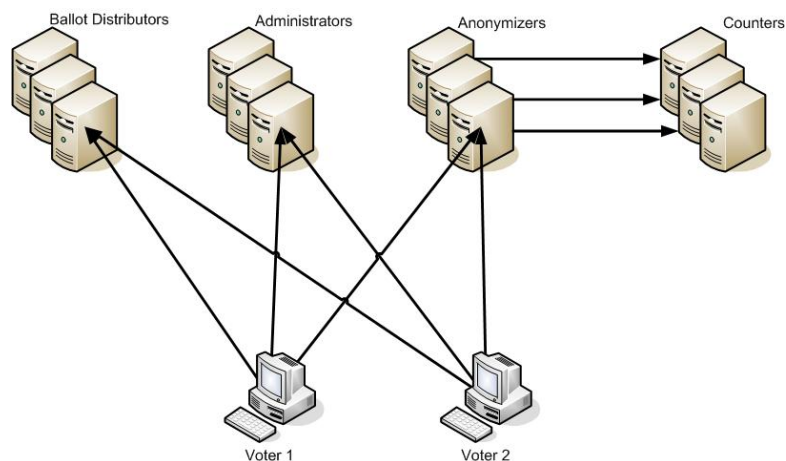
## Robust Electronic Voting System (REVS)

O REVS é um sistema de votação electrónica desenvolvido por Rui Joaquim [1]. No projecto deste sistema prestou-se particular atenção ao desenvolvimento de uma solução que apresentasse: (1) disponibilidade, eliminando os “single points of failure”, conseguida com a inclusão de replicação dos servidores envolvidos; (2) reinício da votação, tanto por vontade do votante como devido a falhas de rede; (3) resistência a conspirações, não permitir que um ou mais servidores, até um número  $j$ , interfiram numa eleição.

Este sistema é composto por quatro tipos de servidores, os quais desempenham os seguintes papéis:

- *Ballot Distributors*. São responsáveis pela distribuição das informações relativas a uma dada eleição;
- *Administrators*. São os servidores responsáveis pela validação dos votos, neste caso existe um número mínimo de validações,  $\frac{n}{2} + 1$  em que  $n$  corresponde ao número total de *Administrators*, para que o voto seja aceite;
- *Anonymizers*. A função destes servidores é a de proteger a identidade dos votantes, impossibilitando os *Counters* de associarem o voto a uma determinada máquina;
- *Counters*. o papel deste servidor é verificar se os votos possuem o número de assinaturas necessárias para ser considerado um voto válido e é o responsável pela realização da contagem no fim da eleição, eliminando os votos repetidos.

Na figura 1, pode ser observado esquematicamente a arquitectura do REVS.



**Figura 1 - Arquitectura REVS**

## **Anonimato nas comunicações**

Existem, hoje em dia, variadíssimas razões para que haja a necessidade de estabelecer comunicações anónimas, mas nem todas elas poderão ser consideradas legítimas. Contudo, no caso dos sistemas de votação electrónica este é um dos aspectos mais relevantes, uma vez que na maior parte das eleições o voto secreto é um direito dos eleitores.

Deste modo, surge a necessidade de criar um mecanismo que permita ocultar a identidade dos intervenientes numa comunicação. Com base neste princípio surgiram as Mix-Nets [2] por David Chaum e mais recentemente os Mix Rings [3] por Matthew Burnside e Angelos Keromytis, os quais serão apresentado a seguir.

### ***Mix-Nets***

Segundo Chaum, este tipo de redes consiste numa rede composta por vários *mixes* (máquinas que desempenham o papel de intermediários na comunicação), cada um deles possuindo uma chave pública conhecida por todos os intervenientes.

Neste tipo de redes, quando um determinado utilizador pretende enviar uma mensagem anónima para outro utilizador, o primeiro passo é definir um percurso através dos *mixes* e construir a mensagem composta por várias camadas cifradas, correspondendo uma camada a cada um dos *mixes* que compõem a rota escolhida. Cada uma das camadas possui a informação relativa ao *next-hop*, bem como a camada da mensagem cifrada com a chave

do próximo destinatário. À medida que a mensagem vai percorrendo a rota traçada as várias camadas que a compõem vão sendo retiradas, até que ao chegar ao último salto a mensagem apenas é composta pelo endereço do destinatário e pela mensagem cifrada com a chave pública do mesmo.

O processo de cifragem da mensagem por camadas pode ser descrito pela expressão seguinte:

$$K_n(R_n, K_{n-1}(R_{n-1}, \dots, K_2(R_2, K_1(R_1, K_d(R_d, M), E)) \dots))$$

na qual M é a mensagem a enviar ao destinatário, E é o endereço do destinatário, os K's são as chaves públicas utilizadas para a construção da mensagem por camadas,  $K_d$  é a chave do destinatário e  $K_1$  a  $K_n$  são as chaves dos *mixes*, sendo  $K_1$  o mais próximo do receptor e  $K_n$  o mais próximo do emissor. Os R's representam *strings* aleatórias utilizadas apenas para garantir a robustez da cifragem.

### **Mix Rings**

Os Mix Rings, projectados por Matthew Burnside e Angelos Keromytis [3], representam uma evolução relativamente às Mix-Nets. Neste caso, em vez de a comunicação apenas ser estabelecida no momento em que um dado utilizador pretende comunicar com outro, esta ocorre sempre desde o momento de criação do anel.

Para um dado anel existe sempre tráfego em circulação, a uma taxa de transmissão constante. Este tráfego não possui informação válida e é apelidado por “cover trafic”, uma vez que a sua função é apenas a de camuflar o tráfego intencional. Contudo, para a maior parte dos *mixes* que compõem o anel, é impossível determinar se uma dada mensagem corresponde a tráfego real. Os únicos que conseguem distinguir estas mensagens são o *mixer* responsável pela iniciação do anel (e corresponde ao emissor) e o *mixer* que recebe a mensagem do tipo *fan-out*.

As mensagens a circular no anel são similares às apresentadas no caso das Mix-Nets, só que neste caso o endereço do destinatário é o endereço do próprio emissor, de forma a completar o anel. Tal não significa que o emissor esteja apenas a comunicar consigo próprio, uma vez que existe uma opção (*fan-out*) nas mensagens que permite, a um *mixer*, a divisão da mensagem recebida, enviando uma para o destinatário especificado e mantendo outra,

composta apenas por informação irrelevante, a circular no anel de maneira a manter inalterado o número de pacotes na rede.

Sempre que o *mixer* emissor, responsável pela iniciação do anel, pretender enviar uma mensagem para um dado receptor, deve escolher aleatoriamente um *mixer* pertencente ao anel para servir como ponto de saída da mensagem, após esta operação procede à construção da mensagem por camadas, tendo em consideração o seguinte formato:

$$\{T, H_1, B_1, [H_2, B_2], P\}$$

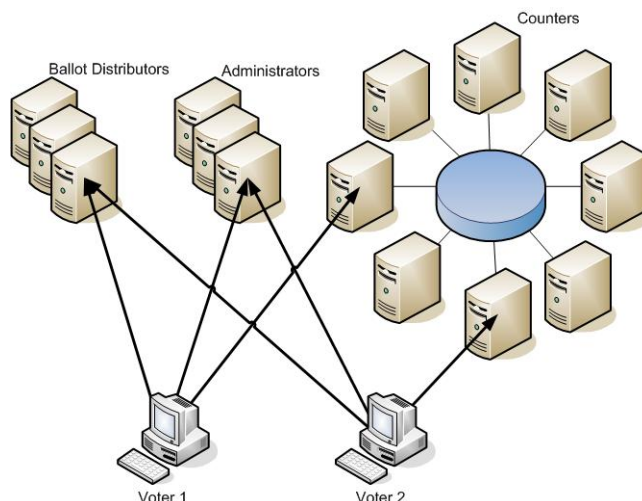
Na mensagem, T corresponde às opções de envio que poderão ser de *set-rate*, *delay* ou *fan-out*. O endereço IP e porto do *next-hop* são indicados no cabeçalho H<sub>1</sub> e B<sub>1</sub> corresponde ao corpo da mensagem cifrado recursivamente com a chave de sessão deste. A mensagem poderá possuir ainda um cabeçalho H<sub>2</sub> e um corpo de mensagem B<sub>2</sub> caso se trate de uma mensagem de *fan-out*. Por fim, à medida que a mensagem vai percorrendo o anel cada *mixer* deverá acrescentar P à mensagem, devido à remoção dos cabeçalhos, para garantir que o tamanho da mensagem a circular na rede permanece constante.

Deste modo, sempre que se deseje enviar uma nova mensagem basta substituir uma das que estão a circular no anel pela mensagem pretendida.

## REVS com Mix Rings

Neste projecto o que se pretende é aproveitar as características disponibilizadas pelos Mix Rings e adaptar o REVS para utilizar esta técnica de anonimato das comunicações. Porquê utilizar mais servidores (*Anonymizers*) para garantir comunicações anónimas, se pudemos utilizar os *Counters*, indispensáveis ao funcionamento do sistema, para garantir esse anonimato?

Na figura seguinte é apresentada a arquitectura proposta com o objectivo de eliminar os *Anonymizers*. Note-se que apesar de os *Counters* estarem representados por uma ligação em anel, este anel é apenas lógico.



**Figura 2 - Arquitectura REVS proposta**

### ***Modelo do sistema***

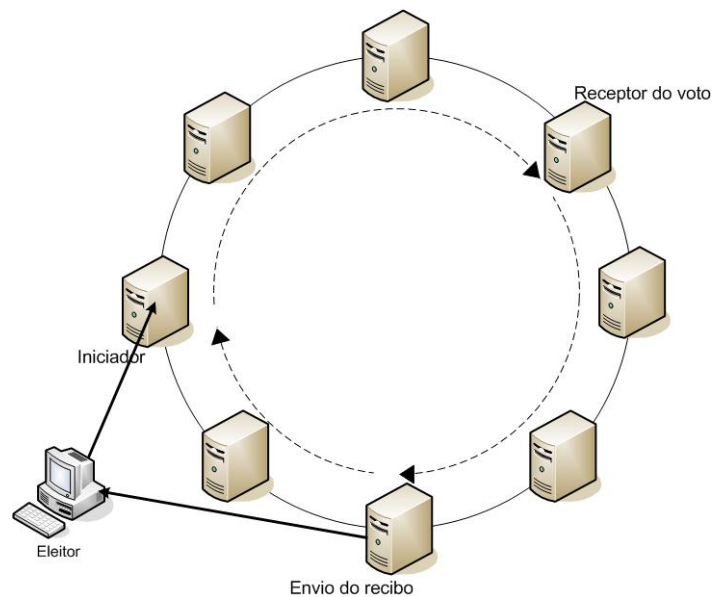
A infra-estrutura do anel corresponde ao conjunto ou subconjunto dos servidores *Counter* configurados no sistema, em que cada um possui um chave pública conhecida por todos os intervenientes na comunicação.

Neste nosso caso, quem inicia a comunicação do tráfego de camuflagem é sempre um servidor *Counter*. Contudo, as mensagens com conteúdo importante são provenientes de fora do anel: são construídas pela aplicação do votante e enviadas para um dos *Counters*, não sendo necessariamente o mesmo para todos os votantes. O *Counter* receptor substitui uma das suas mensagens de *cover traffic* pela mensagem recebida do votante, colocando-a a circular no anel.

Nas mensagens provenientes de um eleitor, existem dois destinatários configurados, aleatoriamente, numa das camadas que as compõem. O primeiro destinatário é um *Counter* escolhido aleatoriamente e corresponde ao servidor responsável pelo armazenamento do voto. O segundo destinatário, também escolhido aleatoriamente, é outro *Counter* que tem como função efectuar o *fan-out* da mensagem e devolver a porção da mensagem que pertence ao recibo para o eleitor que é externo ao anel.

Cada *Counter* efectua previamente uma escolha aleatória dos vários *Counters* para participar na criação do anel, distribuindo-os sequencialmente num anel. Após a criação do anel é iniciado o encaminhamento anónimo do “*cover traffic*” através do Mix Ring.

Na figura 3, podemos observar a sequência executada aquando da recepção de um voto por parte de um eleitor. Após a recepção do voto, por parte de um dos *Counters*, este utiliza o anel criado anteriormente para a transmissão do mesmo. Note-se que o *Counter* responsável pelo envio do recibo ao eleitor encontra-se sequencialmente após o receptor do voto (um *Counter* pertencente ao anel). Com isto pretende-se que no caso de existir uma falha de comunicação não seja enviado o recibo sem o correspondente armazenamento do voto.



**Figura 3 – Possíveis rotas utilizadas na recepção de um voto**

Quando a mensagem relativa ao voto alcança o *Counter* responsável pelo envio do recibo ao eleitor, este divide a mesma, à semelhança da mensagem de *fan-out* existente nos Mix Rings, enviando uma para o eleitor e outra para o *next-hop* continuando a ser transmitida indistintamente do restante tráfego de camuflagem, garantindo-se assim que a quantidade de mensagens em circulação no anel mantém-se inalterada, deste modo do ponto de vista de um intruso cada participante no anel pode ser, num dado instante, o portador da mensagem com significado a percorrer a rede.

Sempre que um *Counter* recebe uma mensagem, proveniente de um dos outros *Counters* intervenientes no anel, este necessita de decifrar a mesma para saber as informações relativas ao *next-hop* e às opções de envio. Antes de enviar a mensagem para o próximo destinatário, o *Counter* deverá acrescentar informação ao corpo da mensagem para que esta permaneça com

o tamanho original, após a remoção dos cabeçalhos da camada correspondente.

### Mensagens no Mix Ring

As mensagens a circular no Mix Ring serão semelhantes às apresentadas por Burnside e Keromytis. Assim, cada mensagem possui o seguinte formato:

$$\{T, E_1, M_1, [V], [E_3, R], P\}$$

Onde cada mensagem é composta por um conjunto de tipos T possíveis que poderão ser um dos seguintes: voto, recibo ou não especificado. E<sub>1</sub> representa o endereço IP e porto do *next-hop* e M<sub>1</sub> corresponde ao corpo da mensagem encriptado com a chave de sessão do *next-hop*. Uma mensagem do tipo voto é também composta pelo campo V que corresponde ao boletim de voto a ser entregue a esse *Counter*. Para uma mensagem do tipo recibo E<sub>3</sub> possui o endereço IP e porto do Eleitor para que lhe possa ser entregue o recibo R.

Por fim, à medida que se vão removendo as camadas da mensagem é necessário realizar um preenchimento da mensagem de forma a garantir que esta permanece com o mesmo tamanho, é este o objectivo do campo P.

A figura seguinte exemplifica uma possível sequência de troca de mensagens entre os *Counters* constituintes de um ciclo de comunicação.

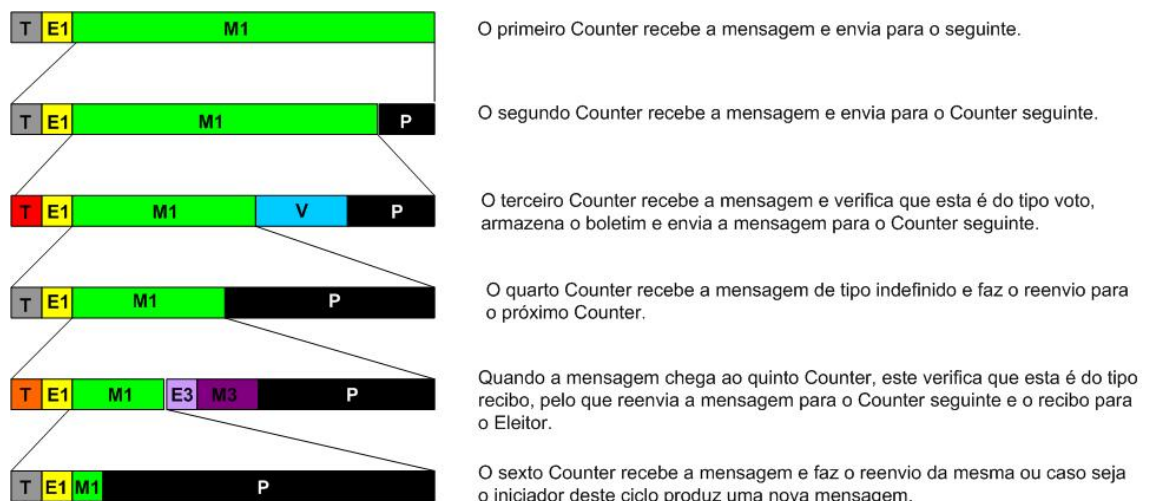
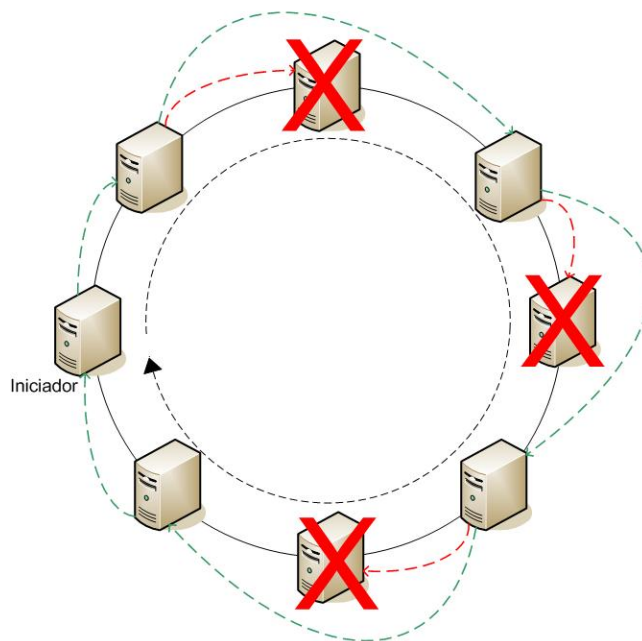


Figura 4 - Exemplo de uma sequência de troca de mensagens entre os elementos constituintes do anel

## **Tolerância a falhas**

Ambos os sistemas de anonimato, Mix-Nets e Mix Rings, não possuem tolerância a falhas. Tal que representa um problema à sua utilização num sistema de votação electrónica, uma vez que este tipo de sistemas necessitam de uma grande disponibilidade durante o período em que se encontra a decorrer a eleição. Deste modo, o passo seguinte será o desenho desta arquitectura com tolerância a falhas, o que implicará alterações no formato das mensagens que percorrem os anéis.

A figura 6, esquematiza uma possível situação de falha nos *Counters* que constituem o anel. Pretende-se que o sistema seja autónomo no que respeita à detecção de uma falha num *Counter* e o retire do anel, bem como que o volte a inserir assim que este volte a ficar disponível, não necessariamente na mesma posição sequencial no anel.



**Figura 5 - Ilustração de uma situação de falha no anel com recuperação do tráfego**

### **Falha nos *Counters* de recepção de voto e de envio de recibo**

Se a falha ocorrer no *Counter* escolhido, aleatoriamente, para armazenar o boletim de voto, devemos tomar medidas durante a construção da mensagem de forma a garantir que o voto é armazenado e que nenhum recibo é enviado sem que tal aconteça. Da mesma forma, se a falha ocorrer no *Counter*

escolhido para enviar o recibo ao votante, é preciso garantir que um outro *Counter* o possa fazer em alternativa.

## **Referências**

[1] Rui Joaquim. A fault tolerant voting system for the Internet. Tese de Mestrado, IST/UTL, Fevereiro 2005.

[2] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), Fevereiro 1981.

[3] Matthew Burnside and Angelos D. Keromytis. Low Latency Anonymity with Mix Rings. Department of Computer Science, Columbia University.