An Efficient Distributed PKI for Structured P2P Networks

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong
`firstname.lastname@supelec.fr`

SUPÉLEC, SSIR Group (EA 4039), France

IEEE P2P, September 9 2009
Seattle, US

Security in P2P Networks

Traditional View

- Security is enforced by a central point
- *Capacities* may be proved by certificates (Certification Authorities)

Specificities of P2P Networks

Dynamic and *Collaborative* networks without *Central Authority*

Distributed Certification (Threshold Cryptography)

- Capacities are still proved by certificates
- These certificates are signed collaboratively by members

$\Rightarrow$ *Trust that t% of the nodes would not collude*

Applications

Admission Control [COPS '08]

Sybil protection, only genuine members are certified

Misbehaving Nodes Exclusion [I2CS '08]

Nodes are monitored, misbehaviors are detected and excluded

Secure Naming of Resources

- P2P SIP directory (unique and provable intelligible names)
- P2P DNS system

$\Rightarrow$ Intelligible names, not $h(PublicKey)$

Outline

Background

Related Work

Fixed Number [Kong *et al.*, 01]

– Certificate generated by a fixed number of peers (t, n)

– Mainly suits MANETs

Fixed Ratio with a Server [Saxena *et al.*, 03]

+ Certificate generated by a fixed ratio of the peers

– Uses a central counter of the network size

– $(t, n) \rightarrow (t, t) \rightarrow (t', n')$: Robustness problem

Fixed Ratio without any Center (our previous scheme [AIMS 08])

+ Certificate generated by a fixed ratio of the peers

+ Fully distributed scheme, no center

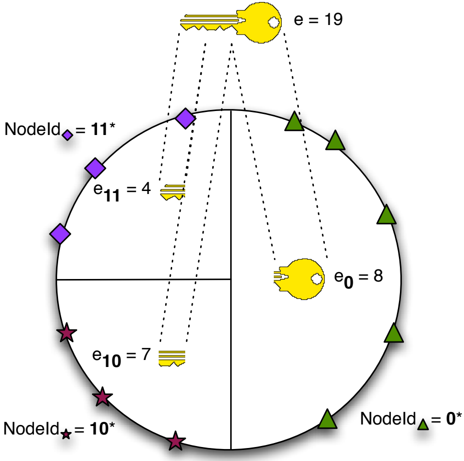
– Byzantine agreements in groups (20 to 40 peers)

Fixing the Threshold Ratio

- RSA, $S = (e, m)$
- s additive shares e_i
- Rep on g peers (*sharing group*)
- Ratio $t = \frac{s}{n} = \frac{1}{g}$
- $o^e[m] = (\prod o^{e_i}[m])[m]$

t enforced by groups size

- g_{min}: minimal size
- g_{max}: maximal size
- $\frac{1}{g_{max}} < t < \frac{1}{g_{min}}$



Maintenance

Three main operations

- Split: splits a group composed of more than g_{max} members
- Merge: merges two groups of less than g_{min} members
- Refresh: randomize shares after a split operation

Maintenance relies on byzantine agreements

- Costly when groups are composed of 20 to 40 members
- Peers join and leave : which peers participate ?
- Difficult to implement

$\Rightarrow$ **Novel maintenance operations without agreements**

Split Operation

Principle

- When a group is composed of more than g_{max} members
- Create two shares from one ($e_{i0} + e_{i1} = e_i$)

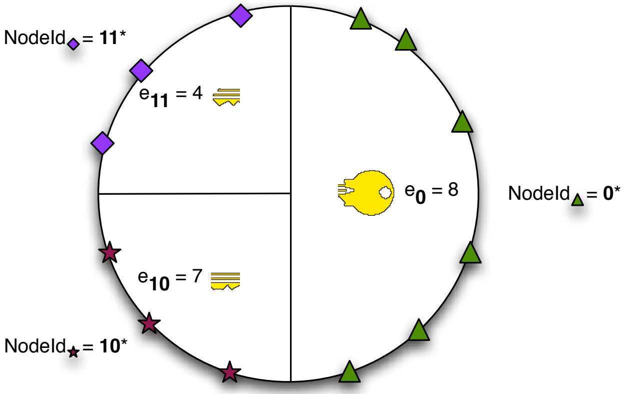
Split e_i

1. Decide a random value e_{i0}, $e_{i1} = e_i - e_{i0}$
2. Migrate to the new groups e_{i0} and e_{i1}
3. Refresh shares e_{i0} and e_{i1}

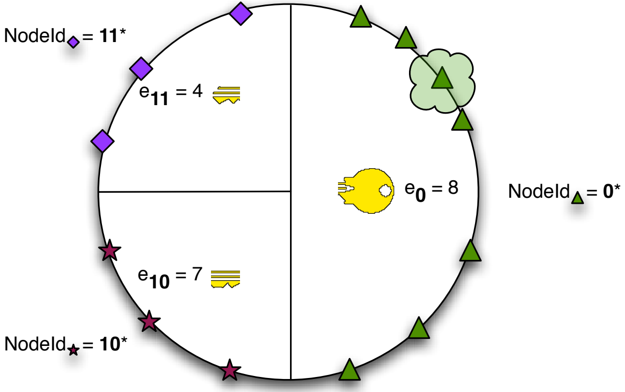
Byzantine agreements

- Decide to split
- Decide e_i

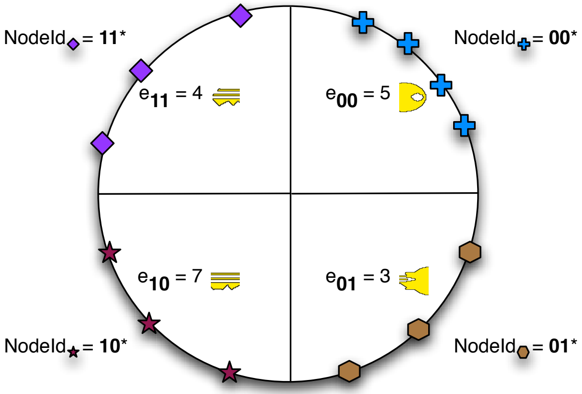
Splitting a share, $g_{max} = 6$



Splitting a share, $g_{max} = 6$



Splitting a share, $g_{max} = 6$



Precompute all possible shares

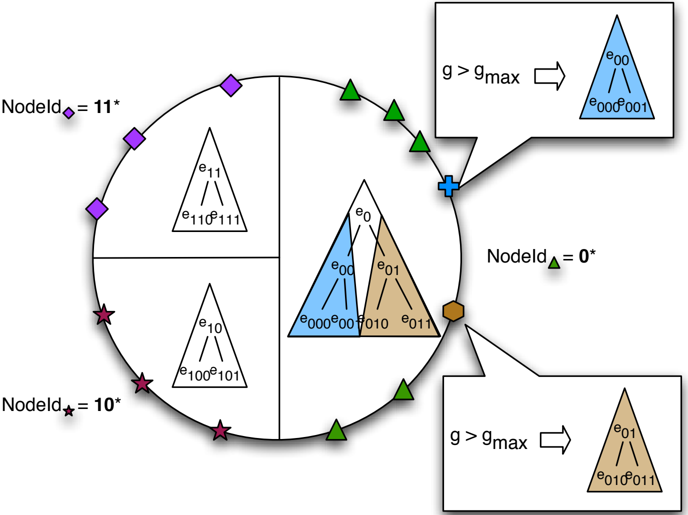
Sharing trees

- Every peer of e_i know the sharing tree of e_i
- The sharing tree of e_i contains all the possible subshares of e_i
- This tree is implicit and can be calculated from e_i :

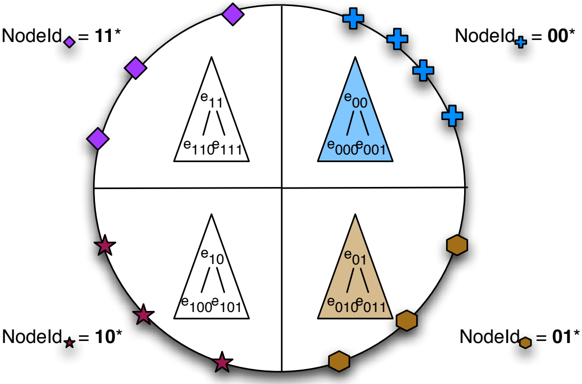
$$e_{x0} = RNG_{h(e_x)},\ e_{x1} = e_x - e_{x0}$$

- No need to store the whole tree, only the root
- Every peer take the same decision without any agreement, at slightly different moments

Splitting a share without agreements



Splitting a share without agreements



Confidentiality of the shares

Each share must be known in only one sharing group

- $\frac{1}{g_{max}} < t < \frac{1}{g_{min}}$ iff peers know only one share
- After a split, every peer of e_i know both created shares ($e_i = e_{i0} + e_{i1}$)

$\Rightarrow$ Refresh operation randomizes shares and sharing trees

Refresh operation

Principle

- After a split, to enforce confidentiality of shares
- Exchange some random value between two shares

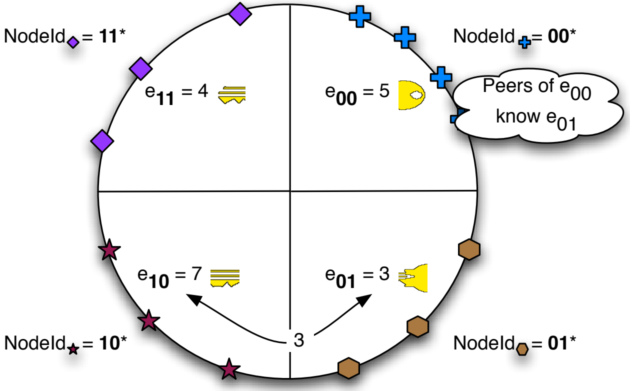
Refresh e_x with e_y

1. Decide a random value Δ
2. $e_x \rightarrow e_x + \Delta$
3. $e_y \rightarrow e_y - \Delta$

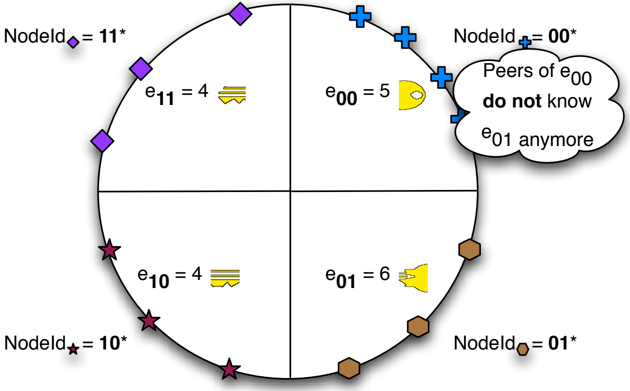
Byzantine agreements

- Decide/Accept to refresh
- Decide Δ

Refreshing e_{00} and e_{11}



Refreshing e_{00} and e_{11}

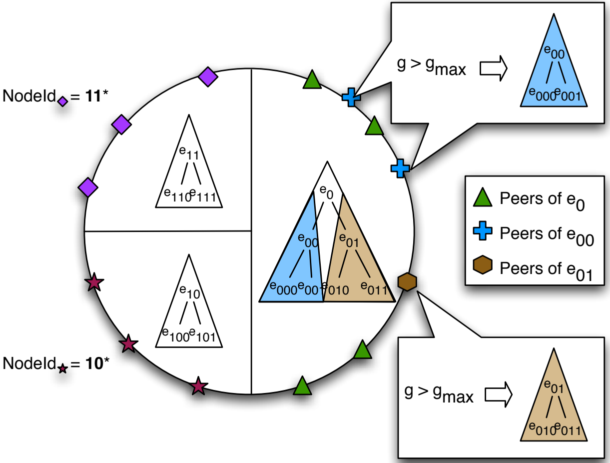


Needs

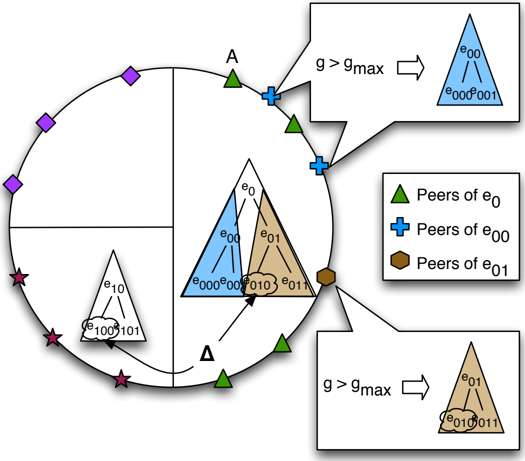
No Sync

Refresh // Split
Which group ?

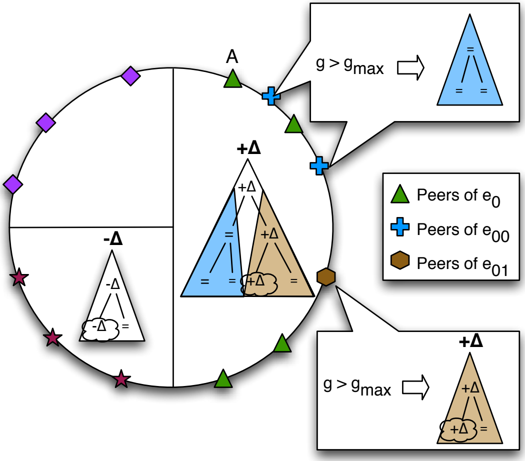
$\Rightarrow$ Refresh must handle inconsistent groups



Values are added to the leafs of sharing trees



Values are added to the leafs of sharing trees



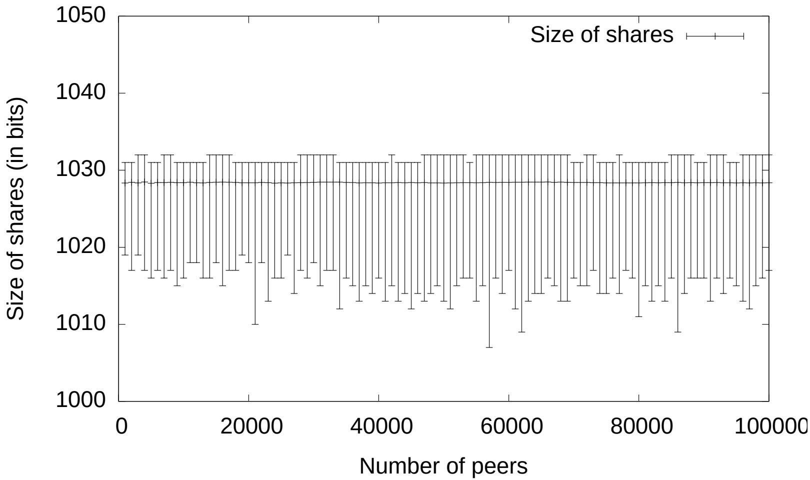
Analysis and Results

Setup

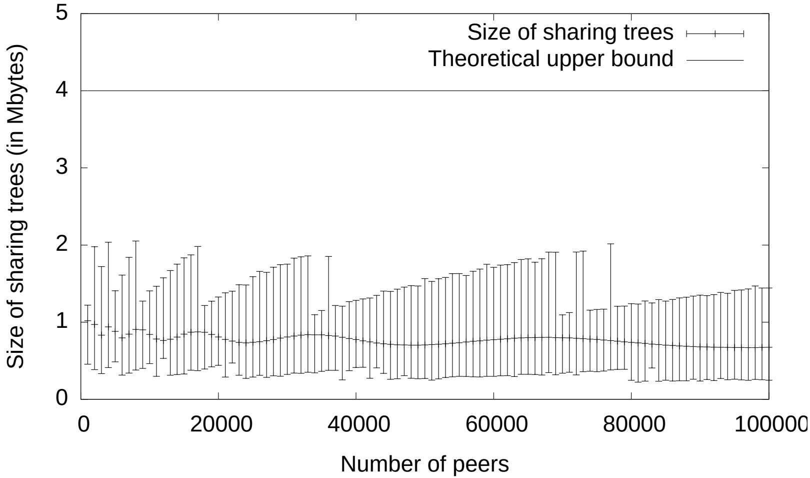
Simulations use PeerSim:

- Up to 100 000 online peers
- Peers are online 10% of the time
- Groups are composed of 20 to 40 members $\Rightarrow$ Tolerates 20% of attackers

Security: Size of shares



Efficiency: Size of sharing trees



Efficient Distributed PKI

Provided Service

- Cryptographic proof of agreement of a fixed ratio of the nodes
- Ratio is enforced with distributed protocols

Efficiency

- Maintenance is local to one or two groups
- Decisions are local to each node, no byzantine agreements
- Sharing trees remain small

Applications

- Protection from Sybil Attack
- Exclusion of attackers
- Secure naming of resources

An Efficient Distributed PKI for Structured P2P Networks

François Lesueur, Ludovic Mé, Valérie Viet Triem Tong
firstname.lastname@supelec.fr

SUPÉLEC, SSIR Group (EA 4039), France

IEEE P2P, September 9 2009
Seattle, US