
Trusted Domains in Peer Groups

Fernando C. Colón Osorio
& Justin Whitney

Wireless System Security Research Laboratory
(WSSRL)

Problem Definition

- How to provide classical security concepts of
 - Confidentiality & Privacy
 - Data Integrity
 - Availability of Assets
- and
- Fairness of operation

On a P₂P, where

- No central hierarchy is present
- All nodes are given equal authority & Privileges
- There are a number of un-trustworthy potential members of the group

Basic Solution

- “Wisdom of Crowds” – from Social engineering

Plus....

- Threshold cryptography (e.g., TS-RSA)
- Lamport’s Byzantine Agreement Protocol for Initial group formation,
- Plus, some other elements...

Wisdom of Crowds - Background

"No one in this world, so far as I know, has ever lost money by underestimating (Undervaluing) the intelligence of the great masses of the plain people."

-H. L. Mencken (1880-1956)

Wisdom of Crowds Basic Concept (Surowiecki, circa 2004):

- H. L. Mencken was wrong!!!
- Large groups of people are smarter than an elite few, no matter how brilliant—better at solving problems, fostering innovation, coming to wise decisions, even predicting the future the future.
- Crowd – Law of Large Numbers!!!! As $N \approx \infty$

Wisdom of Crowds, where to find it?

- Reputation Systems – e.g., eBay
- Odd Making - Las Vegas
- Political Process – 2004 Presidential Election – IOWA Electronic Markets
- More ...

Previous Work

- Peer Groups
 - No Group Hierarchy
 - Cooperative
 - Mobile (or highly dynamic – members come and go)

- Peer Groups in MANETS – Mobile Ad Hoc Networks (sometimes called “Mesh” Networks)
 - Wireless
 - Peer-based (cooperative)
 - Highly Dynamic (mobile)
 - Independent of fixed infrastructure

- Trust, creation, management maintenance (many, many)

- Secret Sharing – Threshold Cryptography

- Admission Control problem & Consensus

Previous Work – Peer Groups In Manets

- “Self-Organized Network-Layer Security in Mobile Ad Hoc Networks”
by H. Yang, X. Meng, and S. Lu from UCAL, presented at WiSE '02

- In their case the “peer group” is a MANET
 - Goal was to provide secure network-layer services of routing and packet forwarding

Yang's major contributions

- Creation of a Trusted Domain (TD)
- Use of threshold-share cryptography (TS-RSA) to certify nodes as part of Trusted Domain
- Can tolerate up to $t-1$ malicious nodes

Limitations of Yang's work

- Depends on central authority to issue S_k/P_k private/public key pairs (in bootstrapping process).
 - Invalidates Ad-Hoc assumption

- Membership and capabilities are coarse: nodes have **full** privileges in the group or **none**

- Requires “uniqueness” but it uses MAC for it???
 - Forgeable – i.e. *not* unique
 - Huge problem
 - Intolerant of masquerading

- Neighbor requirements of protocol to join, makes t-1 tolerance of malicious nodes invalid.

Previous Work, contn...

“On the Utility of Distributed Cryptography in P2P and MANETs: the Case of Membership Control.

by Narasimha, M., Tsudik, G., and Yi, J.

- Presented IEEE International Conference on Network Protocols (ICNP'03), November 2003.

Major Contribution

- ❑ Created a general framework
- ❑ Adds Variable admission control policy & Charter for the group (GC)
- ❑ Implemented using different Threshold cryptographic schemes, and... analyzed their performance
- ❑ Addresses group size variability – i.e., group expands and contract in the number of members

Limitations of Narsimha's work

- Depends on central authority for S_k/P_k generation like Yang's
- Solution intolerant of masquerading
- Trust is binary – all or none

Today

- Overview of our work
 - Created a model, developed an algorithm, and implemented a mechanism to form, manage, and maintain a trusted peer group
 - Same definition of peer group as in Narasimha
 - In this context, we extended the trust model to incorporate different levels of granularity, **not just all or none** as in Narasimha or Yang.
 - *No central point of control required*
 - *True Ad-hoc unlike Narasimha or Yang, but(more later)*
 - *Solution is tolerant of masquerading*
 - *Deals effectively with group expansion and contraction (fault tolerant)*
 - *Performs relatively well for large group sizes (admission under 0.5 – 2.9 secs).*

Model

- Assume an initial group of trusted members,
 $P_{\text{initial}} \quad p \in \{ p_0, p_1, \dots, p_k \}$
- A capabilities matrix $A\{a_0 \dots a_i | a_i < a_{i+1}\}$.
- A policy matrix $O\{o_0 \dots o_i\}$
- An Initial time period $[T_0, T_1]$ during which all nodes that are trusted remain trusted and do not fail

Example

□ Group of P₂P file sharing nodes

■ Capabilities (A):

level	send	receive	search
1	X		
2	X		X
3	X		X
4	X	X	X

■ Policy (O):

level	e_1	e_2	e_3	e_4	e_5
1	$\geq f_0$ files	rate $r_0 \forall f_0 \geq r_0$		$t_0=0$	
2	$\geq f_1$ files	rate $r_1 \forall f_1 \geq r_1$	$searches \leq 3/min$	$t_1=1$ min	$T_2 \geq$ two-days
3	$\geq f_2$ files	rate $r_2 \forall f_2 \geq r_2$	$searches \leq 5/min$	$t_2=5$ min	$T_3 \geq$ one week
4	$\geq f_3$ files	rate $r_3 \forall f_3 \geq r_3$	$searches \leq 10/min$	$t_3=10$ min	$T_4 \geq$ two weeks

Algorithm

□ Basic Premise

- Given we have a trusted peer group G_0
- With a Group Charter (GC) for G_0 specifies:
 - Capabilities matrix for G_0
 - Policy matrix for G_0

□ Then,

- We assume that non-malicious nodes adhere to the policy for their “level”
- All nodes are allowed capabilities for their “level”

Algorithm

- Initial TD (during $[T_0, T_1]$) formation, a group charter (GC) is formed, capability and policy matrix as well as to generate an initial group key.
 - Uses TS-RSA
 - Distributes shares to all nodes in the TD
 - Destroys its own copy of master Sk/Pk
 - t nodes can now sign using Sk

- A node n that wishes to join G_0
 - Requests the GC from the initial Trusted Domain
 - It submits GMCREQ (Request for new GMC) for level 0 to at least $T G_0$
 - It is certified by at least $T G_0$ – a certificate is issued to the node at hand
 - Certificates include an expiration time, of course, when the certificate is no longer valid

- For duration of certificate, the requesting node
 - **must** adhere to all $O[0]$
 - **must not** exceed $A[0]$ or $O[0]$

Algorithm, contn...

- t nodes (at Level 3) observe the behavior of n over a period of time. Upon expiration of the certificates nodes are required to renew

Node n desires to be admitted to the highest level:

- Each node in the Trusted domain at Level 3 issues its partial share to n .
- n assembles partial shares into share S_k
- n has now full membership at Level 3.

Implementation: Monitoring

- ❑ To be recertified, node n must have interacted with t TD nodes per renewal period in accordance with policy/capabilities matrices.
- ❑ n is rewarded for good behavior by an increase in level over time.
- ❑ Bad behavior is tolerated for the renewal period only

Observations

- Damage that n can do is limited by the capability matrix $A[\text{level}(n)]$.
- We can increase trust in n over time
- Maximal damage can be done at $\text{level}=\text{max}$
- We try to make it so difficult to reach $\text{level}=\text{max}$ so that this is undesirable

Experimental Setup

- ❑ An implementation based on “Bouncer” toolkit from U. California Irvine was developed (over 100,000 lines of code in total)
- ❑ 30 lab machines used, each equipped with an Intel Pentium ©III processor, 128 megs of RAM, running Linux kernel version 2.2.x.
- ❑ Group sizes of 5, 10, 15, 20, 25, and 30 were created on a 100 Mbs LAN
- ❑ A Test-bed with automatic scripts for test generation, data collection was created (source code available)

Two Classes of tests

- Performance:
 - Time between GMC Request and GMC Acquisition at each level.
 - Time taken by each peer node to partially sign each GMC Request.
 - Time taken between request of a share, receipt of the share, generation of share.
- Functional tests

Number	Behavior Type
1	Honest
2	Malicious
3	Greedy
4	Masquerading

Results

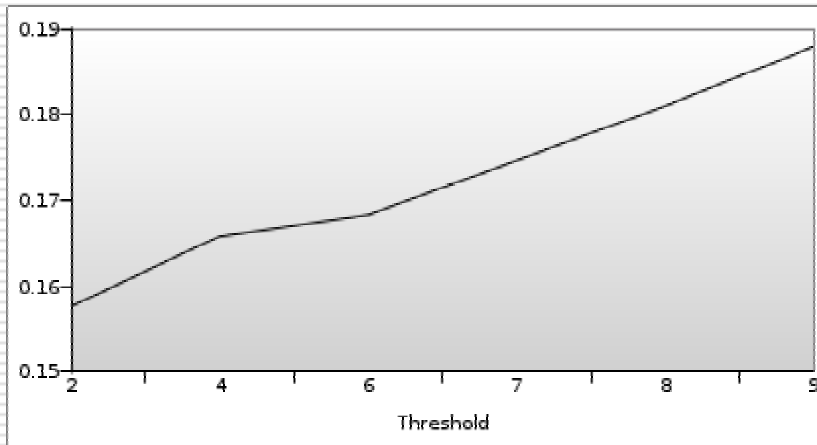


Figure 1a: Computation Time to sign Certificate from a TD (Median)

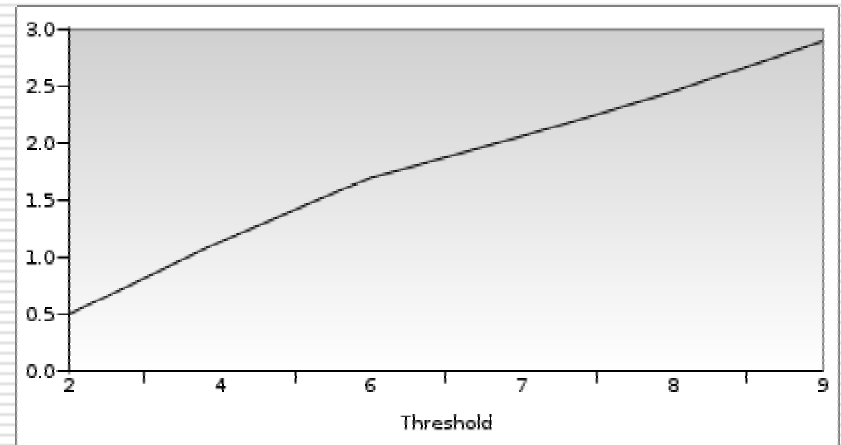


Figure 1b: Computation Time to obtain group Membership on a TD (Median)

More Results

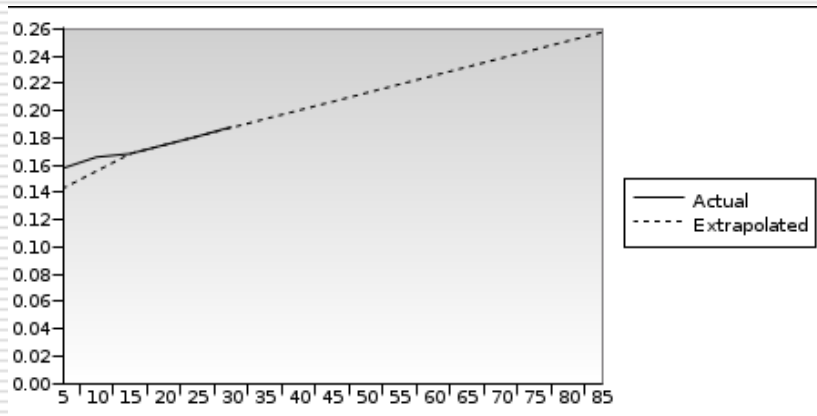


Figure 2a: Computation Time to sign Certificate from a TD (Median) - extrapolated

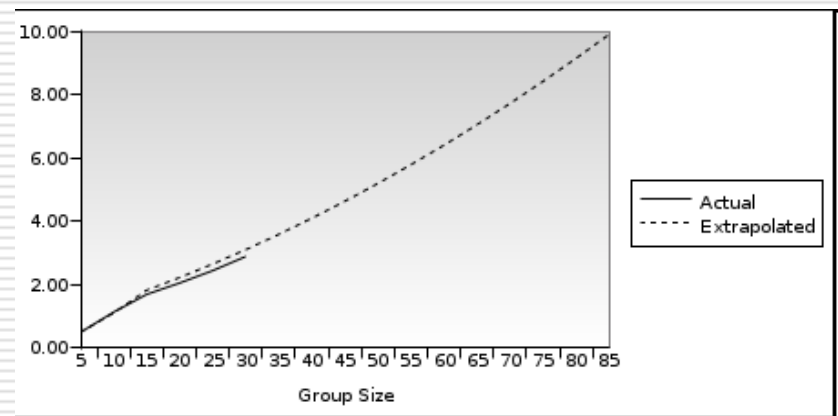


Figure 2b: Computation Time to obtain group Membership on a TD (Median) - extrapolated

Malicious Behaviors

Test Number	When	Behavior	Expected Result
1 2 3	T_0 to T_1 (Bootstrap)	Request early Start Request excessive Expiration Request inappropriate Level	GMC Request Denied, Blacklisted
4 5 6 7	Any Time After T_1	Attempt to Bootstrap Request early Start Request excessive Expiration Request inappropriate Level	GMC Request Denied, Blacklisted
8 9 10 11	Prior to any GMC Request	Attempt to Search at Level 0 Attempt to Download at Level 0 Attempt to Download at Level 1 Attempt to Download at Level 3	GMC Request Denied, Blacklisted

Conclusions

- A new Model & Implementation that provides security and trust in a P₂P network was presented.
 - Granular trust for some member
 - Differing costs for membership
 - Increasing cost for higher levels
- The model and Implementation have the property that it discourages bad behavior (malicious, greedy, others)
- Preserves ad-hoc assumption, unlike previous approaches
- Deals effectively with the Masquerading problem

Questions???

Backup Slides

Key Attributes of Crowd

- ❑ **Diverse** - so that people are bringing different pieces of information to the decision.
- ❑ **Decentralized** - so that no one at the top is dictating the crowd's answer. (Ideal in P₂P networks)
- ❑ **Local Control** - people in the crowd need to be independent, so that they pay attention mostly to their own information, and not worrying about what everyone around them thinks.
- ❑ **Summary** - need a way of summarizing people's opinions into one collective verdict



Example # 2

- Assume G_1 is a group of developers
- New developers wish to join G_1
 - We increase *patches/week* over time
- We only give write access at *level=max*
- If *patches/week* is set appropriately, no one person can masquerade
- Collusion is mitigated by t as before

Example # 2, contn..

□ Group of P₂P file sharing nodes

■ Capabilities (A):

Level	read	Submit -stable	Submit- unstable	Commit
0	X			
1	X	X		
2	X	X	X	
3	X	X	X	X

■ Policy (O): Based on patch submittals (see paper)

Formalizing Creation of G_0

- Assume we have $3m+1$ honest nodes
 - If we don't, probably don't want to form a trusted domain
 - m malicious nodes
- At time T_0 , $3m+1$ collaborates to bootstrap Sk/Pk
- At T_1 , G_0 is formed
 - Between T_0 and T_1 , no honest node becomes malicious
 - Via. BGP we know this can succeed