

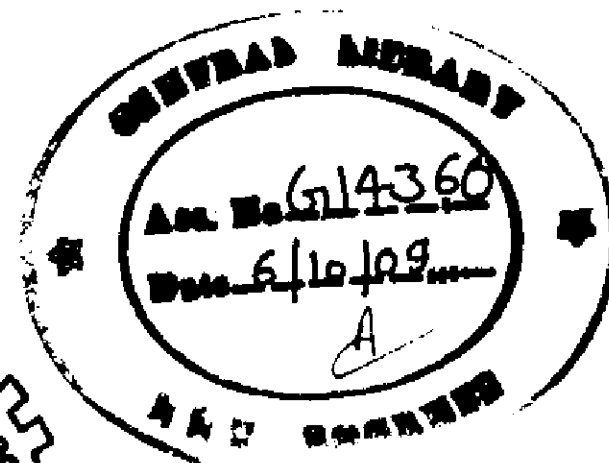
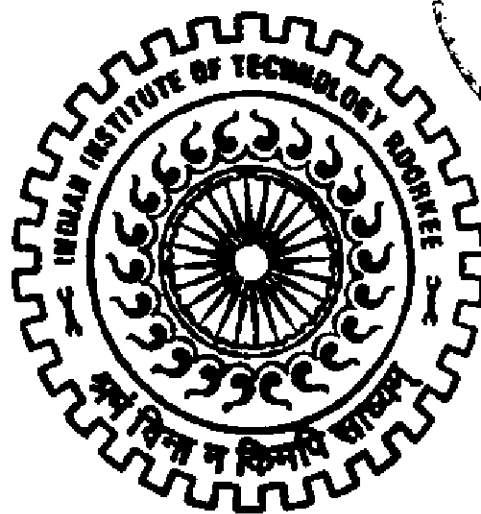
# THREE HOP FORWARDING PROTOCOL FOR ANONYMITY IN GRID COMPUTING SYSTEMS

## A DISSERTATION

*Submitted in partial fulfillment of the  
requirements for the award of the degree  
of*  
MASTER OF TECHNOLOGY  
*in*  
COMPUTER SCIENCE AND ENGINEERING

*By*

**V RAM SRUJAN POLINA**



DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE  
ROORKEE-247 667 (INDIA)

JUNE, 2009

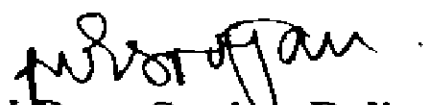
## Candidate's Declaration

---

I hereby declare that the work being presented in the dissertation report titled "**Three Hop Forwarding protocol for anonymity in grid computing systems**" in partial fulfillment of the requirement for the award of the degree of **Master of Technology in Computer Science and Engineering**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, is an authenticate record of my own work carried out under the guidance of Dr. Padam Kumar, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee.

I have not submitted the matter embodied in this dissertation report for the award of any other degree.

Dated: 30/06/09.  
Place: IIT Roorkee.

  
(V Ram Srujan Polina)

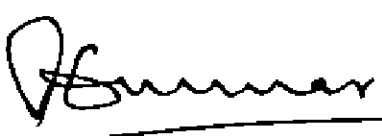
---

## Certificate

---

This is to certify that above statements made by the candidate are correct to the best of my knowledge and belief.

Dated: 30/06/09.  
Place: IIT Roorkee.

  
**Dr. Padam Kumar**  
Professor,  
Department of Electronics and  
Computer Engineering, IIT Roorkee,  
Roorkee -247667 (India).

# ACKNOWLEDGEMENTS

---

First of all, I would like to thank Prof. Dr. Padam Kumar, my supervisor, for getting me interested in the area of Grid Computing, and his consistent and sound critiques of my ideas and work. I am particularly grateful for his enthusiasm, constant support and encouragement. My dissertation could not have been done reasonably without insightful advice from him. Working under his guidance will always remain a cherished experience in my memory. I am also thankful to Indian Institute of Technology Roorkee for giving me this opportunity.

I am also grateful to Amit Agarwal(PhD) for his support and for helping me out with this thesis and Gaurav Tripathi(IDD), for helping me with simulation software. I am also thankful to all my friends who helped me directly and indirectly in completing this dissertation.

Most importantly, I would like to extend my deepest appreciation to my family and relatives for their love, encouragement and moral support. Finally I thank God for being kind to me and driving me through this journey.

**P V R Srujan**

## ABSTRACT

---

Grid computing is widely regarded as a technology of immense potential in both industry and academia. Recently, the high computing industries like finance, life sciences, energy, automobiles, rendering, etc. are showing a great amount of interest in the potential of connecting standalone and silo based clusters into a department and sometimes enterprise wide grid system. Even when the enterprises have considered grid as a solution, several issues have made them reconsider their decisions. Issues related to application engineering, manageability, data management, licensing, security, etc. have prevented them from implementing an enterprise-wide grid solution. As a technology, grid computing has potential beyond the high performance computing industries due to its inherent collaboration, autonomic, and utility based service behavior. To make this evolution possible all the above mentioned issues need to be solved. As an issue, security is perhaps the most important and needs close understanding as grid computing offers unique security challenges.

The work presented in this thesis is related to anonymous communication on grids. As grid computing scales up in size and diversity, anonymous communications will be desirable, and sometimes vital, for certain applications. However, existing anonymity protocols, when being applied to grid applications, either dramatically degrade system efficiency or cause severe performance bottlenecks. A distributed and highly efficient anonymity protocol can be designed if one considers the existing *trust* in grids. We have designed such a protocol based on controlled anonymity, which maintains an entity's anonymity against un-trustable entities. This protocol provides sender anonymity. We use the entropy based information theoretic based metrics to quantitatively analyze the degree of anonymity that could be offered by this protocol, and use a simulator to confirm its efficiency advantage over its predecessor, the 2-hop forwarding protocol.

# Contents

---

<b>Candidate's Declaration &amp; Certificate</b> .....	i
<b>Acknowledgements</b> .....	ii
<b>Abstract</b> .....	iii
<b>Contents</b> .....	iv
<b>List of Figures</b> .....	vii
<b>List of Tables</b> .....	viii
<b>1. Introduction and Statement of the Problem</b>	<b>1</b>
1.1 Introduction.....	1
1.1.1 Benefits of Grid Computing.....	1
1.1.2 Grid Computing Issues and Concerns.....	2
1.1.3 Privacy and Anonymity Issues.....	5
1.2 Motivation.....	6
1.3 Problem Statement and Contribution.....	8
1.4 Outline of the Report.....	8
<b>2. Grid Security Architecture</b>	<b>9</b>
2.1 Introduction.....	9
2.2 Grid Security Requirements.....	10
2.3 Grid Security Model.....	13
<b>3. Background and Literature Review</b>	<b>18</b>
3.1 Security and Privacy.....	18
3.2 Virtual Organizations.....	19
3.2.1 The Emergence of Virtual Organizations.....	20
3.2.2 VO Life Cycle.....	22
3.3 Definitions and Notations.....	23

3.4 Trust.....	23
3.4.1 Trust Management in Virtual Organizations.....	24
3.4.2 Trust Model for Grid Applications.....	26
3.5 Existing Approaches to Achieving Anonymity.....	27
3.6 2-Hop Forwarding Protocol.....	28
3.6.1 Analysis of the Degree of Anonymity.....	29
3.6.2 Disadvantages of 2-Hop Forwarding Protocol.....	31
3.6.3 Selection of Forwarder and Possible Attacks.....	32
<b>4. Anonymity Metrics</b>	<b>33</b>
4.1 Introduction.....	33
4.1.1 Defining Anonymity.....	33
4.2 Model.....	34
4.2.1 A Taxonomy of Attackers and Their Possible Attacks.....	35
4.3 Information Theoretic Anonymity Metrics.....	35
4.3.1 Entropy.....	36
4.3.2 Effective Anonymity Set Size.....	36
4.3.3 Degree of Anonymity.....	37
4.3.4 Average Degree of Anonymity.....	38
<b>5. 3-Hop Forwarding Protocol</b>	<b>39</b>
5.1 Protocol.....	39
5.1.1 Definitions and Notations.....	40
5.2 Analysis of Degree of Anonymity.....	40
<b>6. Results and Discussions</b>	<b>43</b>
6.1 OMNeT++ .....	43
6.2 Comparison to 2-Hop Protocol.....	47
6.3 Selection of the Forwarder and Possible Attacks.....	47

## List of Figures

---

Figure 2.1: Components of Grid Security Model.....	13
Figure 3.1: An example of a grid of multiple virtual organizations (VOs).....	20
Figure 3.2: 2-hop forwarding protocol.....	28
Figure 3.3: Minimum and maximum anonymity of 2-Hop Forwarding Protocol.....	30
Figure 3.4: Load imbalance due to selection of constant forwarder.....	31
Figure 3.5: Improved load balance but possibility of intersection attack.....	31
Figure 3.6: Load imbalance due to selection of forwarder which is a member of several VOs.....	31
Figure 4.1: Model for anonymity systems.....	34
Figure 4.2: Taxonomy of the Attackers.....	35
Figure 4.3: Anonymity set.....	37
Figure 5.1: 3-hop forwarding protocol.....	39
Figure 5.2: Maximum Degree of Anonymity.....	42
Figure 6.1: Simulation Model in OMNeT++.....	44
Figure 6.2: Degree of Anonymity of 2-hop protocol.....	45
Figure 6.3: Degree of Anonymity of 3-hop protocol.....	45
Figure 6.4: Delay in 2-hop protocol.....	46
Figure 6.5: Delay in 3-hop protocol.....	46

## List of Tables

---

Table 3.1: Privacy policy at site 6.....	20
Table 3.2: VO membership matrix for the entire grid.....	20



## 1.1 An Overview

A Computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high end computational capabilities. Computational Grids<sup>[1]</sup> couple computers, storage systems, and other devices to enable advanced applications such as distributed supercomputing, teleimmersion, computer enhanced instruments and distributed data mining. Grid applications are distinguished from traditional client-server applications by their simultaneous use of large numbers of resources, dynamic resource requirements and use of resources from multiple administrative domains, complex communication structures, and stringent performance requirements among others.

### 1.1.1 Benefits of Grid Computing

In recent years, the IT infrastructure of most enterprises is facing a huge amount of stress due to the significant increase in transaction volumes. In addition, there are requirements in terms of collaboration and virtualization of resources and policies. All these business requirements drive the need and deployment of the grid in enterprises. There are mainly four distinct benefits of using grids *viz. performance and scalability, resource utilization, management and reliability and virtualization.*

#### **Performance and Scalability:**

Many pharmaceutical and financial enterprises are on a constant lookout for solutions which can reduce their time to market their products. In some cases, even a 5 - 10% improvement results in huge cost savings. Grid computing solutions of having a shared infrastructure provide more computational capabilities and increase scalability of the IT infrastructure. Most of the enterprises are therefore currently looking at the grid as a more flexible and scalable versions of their cluster infrastructure. As a result, most of the applications running on the grid infrastructure are compute intensive or batch-type applications.

#### **Resource Utilization:**

Another pertinent grid imperative is the need to utilize the IT resources more efficiently. It has been found that most of the IT resources in medium to large scale

enterprises are grossly underutilized. The average utilization is as low as 5 - 10% for PCs and around 30 - 35% for servers. Grid computing offers a mechanism to utilize the resources more efficiently through the process of resource sharing.

### **Management and Reliability:**

As the IT infrastructure grows, the systems become more and more complex and heterogeneous. Therefore, the issue of management becomes extremely critical. Grid computing provides a single interface for managing the heterogeneous resources. The complexity of managing the heterogeneous resources separately is greatly reduced in such an integrated management environment. Another benefit of grid computing is that it can create more robust and resilient IT infrastructure through the use of decentralization, fail-over and fault tolerance to make the infrastructure better suited to respond to minor or major disasters.

### **Virtualization:**

With the growth of mergers and acquisitions in the enterprise world, heterogeneity is inevitable. Heterogeneity exists in the type of hardware, storage, operating systems, and policies within the enterprises. The grid provides virtualization of heterogeneous resources resulting in better management of the resources. It is to be noted that the problem is not entirely solved. As one will find out in the subsequent chapters of the dissertation that managing heterogeneous security policy is still a challenge and requires research attention.

### **1.1.2 Grid Computing Issues and Concerns**

The major concerns in grid computing are:

- Application and data engineering,
- Manageability,
- Licensing issues, and
- Security.

### **Application and Data Engineering:**

Most of the early adopters of the grid are users in the areas where there are huge amounts of data and computation involved like life sciences, finance, automotive and aerospace, energy etc. Most of the users have been using applications in high performance clusters or in some cases SMPs and find grid computing an excellent opportunity to move

their applications to a cluster of clusters or a combination of cluster and PCs available in the organization to get the performance benefits without putting in too much investment. The problem in the above scenario is that there is lack of tools, frameworks, or platforms to help users *gridize* (enable applications to run over grid) their applications.

Gridization has two aspects:

- Data can be manipulated, striped across the grid for enhanced performance. There are applications in life sciences like BLAST and in other domains where this type of technique will be useful. We call this *data engineering*.
- Another aspect involves manipulating the applications themselves so that they are able to extract maximum benefit out of the grid computing infrastructure. We call this *application engineering*.

In the data engineering space, where application data is split across the grid for enhanced performance, there are tools and technologies which partially achieve this. These tools provide some mechanisms for writing parallel applications or analysis of applications. However, most of these tools and techniques are insufficient for enterprise needs. Typically, enterprise applications are complex and have several business level and application level dependencies which cannot be efficiently handled by these tools. Therefore, significant research and development efforts need to be undertaken in this direction to develop tools for enterprise applications.

### **Grid Manageability:**

From its inception, IT systems were besotted with problems like scheduling, management, security, and other challenges. To solve these problems, substantial work has been carried out at different levels, for example in the form of infrastructure management systems, job schedulers, mechanisms for implementing security, etc. It is becoming clear that one key concern which this evolutionary growth of the technology has resulted in is complexity, and the ensuing problem of *manageability*. Manageability is often cited as one of the key issues in any real world grid implementation. The problem of manageability is closely related to that of integration. Since grids bring together software components, frameworks, middleware and hardware elements, integrating them together often entails gluing together systems which may not be designed and developed with that in mind.

### **Grid Licensing:**

Large scale information technology systems are undergoing transformational changes in the wake of technological developments and their adoption in scientific and business applications. Software-intensive systems are increasingly being developed using service orientation and virtualization of resources, as is evident in the growth and adoption of Web services and grid computing technologies. Since the software architectures and products are evolving rapidly to realize these visions, software' price and license is required to be aligned with this new reality. The grid allows the sharing of resources across different systems. For custom defined applications this vision holds good. However, for vendor applications the gain in sharing of resources is offset by the licensing needs of the applications sharing the resources. The technology challenge is there to develop suitable pricing and licensing infrastructure.

### **Grid Security:**

Security requires the three fundamental services: *authentication*, *authorization*, and *encryption*. A grid resource must be authenticated before any checks can be done as to whether or not any requested access or operation is allowed within the grid. Once the grid resources have been authenticated within the grid, the grid user can be granted certain rights to access a grid resource. This, however, does not prevent data in transit between grid resources from being captured, spoofed, or altered. The security service to insure that this does not happen is encryption.

There are different types of grid architectures <sup>[2]</sup> to fit different types of business problems. Some grids are designed to take advantage of extra processing resources, whereas some grid architectures are designed to support collaboration between various organizations. The type of grid selected is based primarily on the business problem that is being solved. Taking the goals of the business into consideration will help you choose the proper type of grid framework. A business that wants to tap into unused resources for calculating risk analysis within their corporate datacenter will have a much different design than a company that wants to open their distributed network to create a federated database with one or two of their main suppliers. Such different types of grid applications will require proportionately different designs, based on their respective unique requirements. The selection of a specific grid type will have a direct impact on the grid solution design. Additionally, it should be

mentioned that grid technologies are still evolving and tactical modifications to grid reference architecture may be required to satisfy a particular business requirement.

The grid was initiated as a way of supporting scientific collaboration, where many of the participants knew each other. In this case, there is an implicit trust relation. All partners have a common objective, for instance to realize a scientific experiment and it is assumed that resources would be provided and used within some defined and respected boundaries. However, when the grid is intended to be used for business purposes, it is necessary to share resources with unknown parties. Such interactions may involve some degree of risk since the resource user cannot distinguish between high and low quality resource providers on the grid. The inefficiency resulting from this asymmetry of information can be mitigated through trust mechanisms.

### **1.1.3 Privacy and Anonymity Issues**

Several times privacy is closely resembled with anonymity that demands the need of being unidentified or unobserved while transacting over public domain such as web or other public realm. Adequate level of privacy needs to be achieved through controlled disclosure of identity and associated information. *Anonymity* can ensure achievement of privacy needs. In general, anonymous message transmission requires that the transacting message would not carry any information about the original sender and intended receiver.

Anonymity refers to the state that an entity is not identified in the communications with others. Anonymous communications <sup>[3]</sup> may have one or more of the following properties: sender anonymity, receiver anonymity, and unlink-ability. Sender anonymity means that when a message is observed, the sender cannot be identified. Receiver anonymity means that the receiver cannot be identified. Unlink-ability means that the relationship between the sender and the receiver in the communication cannot be identified, even if sender anonymity or receiver anonymity cannot be guaranteed. An anonymity mechanism may provide anonymity against one type of threat but not against another type. For example, using a proxy between senders and receivers may provide sender anonymity against the receiver and vice versa, but cannot provide any anonymity against an eavesdropper who can observe all messages from and to the proxy. Most activities where anonymity is desired require sender anonymity.

Achieving anonymity in a network is very difficult. Although it usually makes use of encryption techniques, it is a fundamentally different problem than achieving data confidentiality. When speaking about encryption, we care only about protecting the data. In contrast, anonymity means protecting the communication itself in the sense that it should not be possible for an attacker to follow the path data packets take through the network. Since it is not possible to get rid of the data itself, we have to try to confuse attacker such that they cannot trace the data from the source to the destination. Sender anonymity is most commonly achieved by transmitting a message to its destination through one or more intermediate nodes in order to hide the true identity of the sender. The message thus is effectively *rerouted* along what is called a *rerouting path*.

For a given anonymous communication system, we measure the ability to protect sender anonymity by determining how much uncertainty this system can provide to hide the true identity of a sender. We call this measure the *anonymity degree*. There are many existing approaches such as crowds, mixes and onion routing to provide anonymity. However, the existing approaches, when applied to grid computing environment, will incur overheads that are too high to be acceptable. This is because most grid computing applications involve transfers of very large amounts of data, and system efficiency is much more important for grid applications than for the others. If a small number of proxies were used, the proxies would quickly become the bottleneck as the number of communications increases. Using rerouting of random length could avoid having the bottlenecks, but will greatly increase network traffics and the processing times at intermediate nodes, and thus significantly reduce the efficiency of the whole grid.

A distributed and highly efficient anonymity mechanism for grid computing can be designed if one takes use of the trust existing among certain sets of entities in a grid. We call those sets of entities “trust sets”. In a grid, a trust set may form if the entities have close collaborations or if they belong to the same organization, which are typical in grid applications. We design an anonymous forwarding protocol based on trust sets.

## **1.2 Motivation**

People increasingly use the grid for a wide range of activities: to share large amounts of data or for high end calculations which require very high computing power. While

performing a network activity, even if the confidentiality of the information being transmitted is protected through encryption, the source and destination of the communication are easily traceable. The information on who communicates with whom may reveal critical information that could be used against the user. The link-ability of all traffic information generated by a grid user (e.g., through the IP address, national ID number or social security number), allows for sophisticated profiling of each user.

In the current communication infrastructure, traffic data is available at moderate cost to anyone willing to harvest it, without the data subject being aware of it. There is already an emerging market of personal data that criminals use to impersonate their victims. In some cases, the damage inflicted to identity theft victims is huge. With the development of data collection technologies, data storage capacity and profiling techniques, these data become easier to get, cheaper to store and more profitable to use (either for legal or illegal practices). Software tools which can be used for privacy violations are increasingly available. These include spyware such as key loggers. In this scenario, large amounts of information about large numbers of people are under the control of a few data holders. Users effectively lose control over their own personal data, and at the same time all their online activity can be collected, aggregated and stored. These data can be used to take decisions that impact the data subjects. This asymmetric distribution of information creates dangerous imbalances, as those in control of the information may use their acquired power for many different purposes. This possibility is particularly disturbing in contexts where human rights are not respected. Totalitarian regimes may monitor electronic communication in order to identify (and punish) dissidents or journalists.

Technology can be designed to keep personal data under the control of the user. The user could disclose the minimal amount of information to the entities with which he interacts. Anonymity technologies serve as tools for the protection of privacy in electronic applications, and they are a key component of Privacy Enhancing Technologies (PETs). Anonymous communication networks protect the privacy of Internet users towards the other end of the communication and towards observers in the network. This is achieved by hiding the link between the initiator of the communication and the responder. For applications such as electronic voting and electronic payments, anonymity and privacy are strictly necessary. In a democratic society, public elections will be held anonymously and citizens have a fundamental right to privacy, for example when buying goods or subscribing to services.

### **1.3 Problem Statement and Contribution**

The scope of the research presented is anonymous communications over grid networks. Our contributions have been related to the field of multi-hop forwarding systems. We propose a protocol based on existing trust in grid. Sites in same virtual organization form a trust set. We use these trust sets to develop a protocol to provide anonymity in grid systems. Our major contribution consists of methods to measure anonymity, the design of new anonymous communication mechanism, a 3-hop forwarding protocol. This anonymous protocol is different from previous ones in that it provides controlled anonymity, i.e. the identity of an entity in a communication is hidden from potential adversaries but not from some trustable peer entities. We calculate the degree of anonymity, bandwidth and data transfer latency for the proposed protocol and compare with the 2-hop forwarding protocol. We study the existing protocols such as crowds and onion routing.

### **1.4 Outline of the report**

The outline of this dissertation is the following:

- Chapter 1 presents the motivation and context of the work performed for this thesis.
- Chapter 2 presents a brief overview of Grid security architecture.
- Chapter 3 presents background literature for the project. We present our study on virtual organizations and trust in grid computing. We also made brief study about existing anonymity protocol. We also present the underlying protocol on which our current protocol is based upon.
- Chapter 4 introduces information theoretic anonymity metrics. We provide a general model for anonymity systems and present two flavors of entropy based anonymity metrics: the effective anonymity set size and the degree of anonymity. We apply the metrics to an anonymity system and discuss the results obtained.
- Chapter 5 presents an analysis of the 3-hop forwarding protocol. We calculate degree of anonymity and analyze its merits and demerits.
- Chapter 6 discusses results of the above protocols and is compared with 2-hop forwarding protocol
- Chapter 7 presents conclusions and future work that can be done in this field.



## 2.1 Introduction

Research and development efforts within the grid community have produced protocols, services, and tools that address the challenges arising when we seek to build scalable *virtual organizations* (VOs). What distinguishes a VO from a classical organization is that it may gather individuals and/or institutions that have agreed to share resources and otherwise collaborate on an ad-hoc, dynamic basis, while they continue to belong to different real organizations, each governed by their own set of internal rules and policies. This poses a challenge when combined with the fact that an individual or institution may be a member of several VOs simultaneously. From a security point of view, one is thus confronted with protection domains that may superpose, straddle, and intersect one another in many different ways. Within this context, we require interoperability among domains while maintaining a clear separation of the security policies and mechanisms deployed by both virtual and real organizations.

The technologies that have evolved from the grid community include security solutions that support management of credentials and policies when computations span multiple institutions; resource management protocols and services that support secure remote access to computing and data resources and the co-allocation of multiple resources, information query protocols and services that provide configuration and status information about resources, organizations and services and data management services that locate and transport datasets between storage systems and applications.

Controlling access to services through robust security protocols and security policy is paramount to controlling access to VO resources and assets. Thus, authentication mechanisms are required so that the identity of individuals and services can be established, and service providers must implement authorization mechanisms to enforce policy over how each service can be used. The requirement for composition complicates issues of policy enforcement, as one must be able to apply and enforce policy at all levels of composition and to translate policies between levels of composition. For example, when running a data mining query against a distributed collection of databases, we might need to enforce not only

database-specific access control policies based on the identity of the requestor but also resource consumption policies associated with the VO.

## 2.2 Grid Security Requirements

OGSA <sup>[4]</sup> security must be seamless from edge of network to application and data servers, and allow the federation of security mechanisms not only at intermediaries, but also on the platforms that host the services being accessed.

The basic OGSA security model must address the following security disciplines:

- *Authentication:* Provide plug points for multiple authentication mechanisms and the means for conveying the specific mechanism used in any given authentication operation. The authentication mechanism may be a custom authentication mechanism or an industry-standard technology. The authentication plug point must be agnostic to any specific authentication technology.
- *Delegation:* Provide facilities to allow for delegation of access rights from requestors to services, as well as to allow for delegation policies to be specified. When dealing with delegation of authority from an entity to another, care should be taken so that the authority transferred through delegation is scoped only to the task(s) intended to be performed and within a limited lifetime to minimize the misuse of delegated authority.
- *Single Logon:* Relieve an entity having successfully completed the act of authentication once from the need to participate in re-authentications upon subsequent accesses to OGSA-managed resources for some reasonable period of time. This must take into account that a request may span security domains and hence should factor in federation between authentication domains and mapping of identities. This requirement is important from two perspectives:
  - It places a secondary requirement on an OGSA-compliant implementation to be able to delegate an entity's rights, subject to policy (e.g., lifespan of credentials, restrictions placed by the entity)

- If the credential material is delegated to intermediaries, it may be augmented to indicate the identity of the intermediaries, subject to policy.
- *Credential Lifespan and Renewal:* In many scenarios, a job initiated by a user may take longer than the life span of the user's initially delegated credential. In those cases, the user needs the ability to be notified prior to expiration of the credentials, or the ability to refresh those credentials such that the job can be completed.
- *Authorization:* Allow for controlling access to OGSA services based on authorization policies (i.e., who can access a service, under what conditions) attached to each service. Also allow for service requestors to specify invocation policies (i.e. who does the client trust to provide the requested service). Authorization should accommodate various access control models and implementation.
- *Privacy:* Allow both a service requester and a service provider to define and enforce privacy policies, for instance taking into account things like personally identifiable information (PII), purpose of invocation, etc. (Privacy policies may be treated as an aspect of authorization policy addressing privacy semantics such as information usage rather than plain information access).
- *Confidentiality:* Protect the confidentiality of the underlying communication (transport) mechanism, and the confidentiality of the messages or documents that flow over the transport mechanism in a OGSA compliant infrastructure. The confidentiality requirement includes point-to-point transport as well as store-and-forward mechanisms.
- *Message integrity:* Ensure that unauthorized changes made to messages or documents may be detected by the recipient. The use of message or document level integrity checking is determined by policy, which is tied to the offered quality of the service.
- *Policy exchange:* Allow service requestors and providers to exchange dynamically security (among other) policy information to establish a negotiated security context between them. Such policy information can contain authentication requirements, supported functionality, constraints, privacy rules etc.

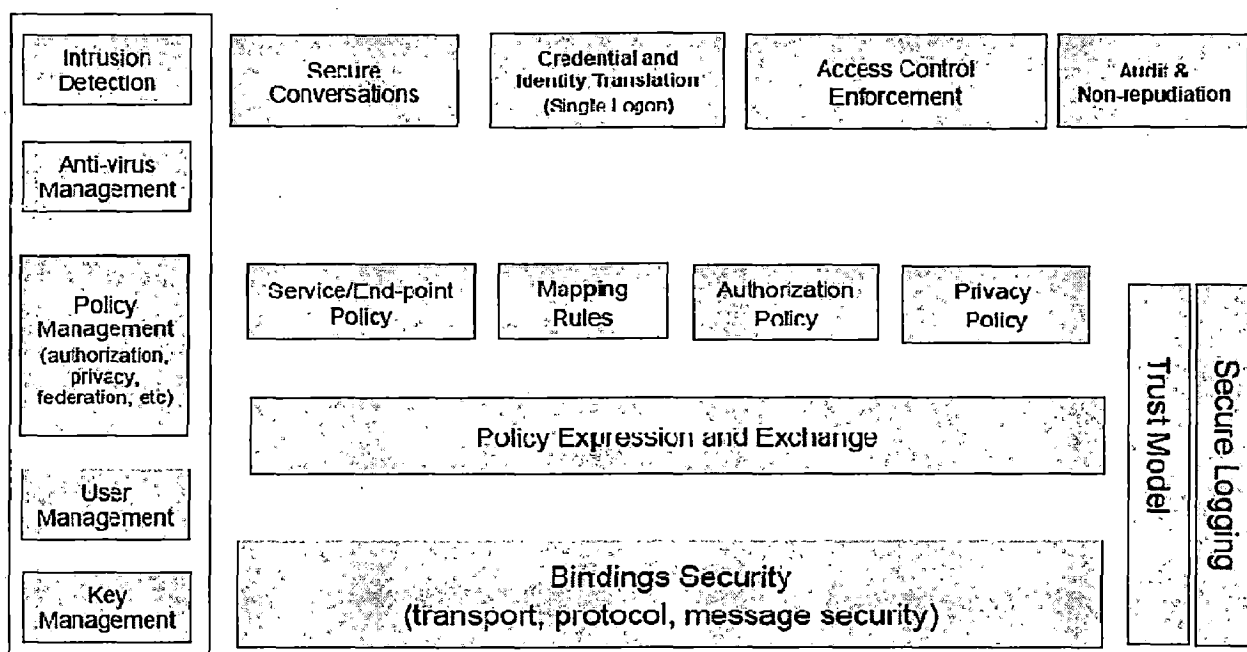
- *Secure logging:* Provide all services, including security services themselves, with facilities for time-stamping and securely logging any kind of operational information or event in the course of time - securely meaning here reliably and accurately, i.e. so that such collection is neither interruptible nor alterable by adverse agents. Secure logging is the foundation for addressing requirements for notarization, non-repudiation, and auditing.
- *Assurance:* Provide means to qualify the security assurance level that can be expected of a hosting environment. This can be used to express the protection characteristics of the environment such as virus protection, firewall usage for Internet access, internal VPN usage, etc. Such information can be taken into account when making a decision about which environment to deploy a service in.
- *Manageability:* Explicitly recognize the need for manageability of security functionality within the OGSA security model. For example, identity management, policy management, key management, and so forth. The need for security management also includes higher-level requirements such as anti-virus protection, intrusion detection and protection, which are requirements in their own rights but are typically provided as part of security management.
- *Firewall traversal:* A major barrier to dynamic, cross-domain grid computing today is the existence of firewalls. As noted above, firewalls provide limited value within a dynamic grid environment. However, it is also the case that firewalls are unlikely to disappear anytime soon. Thus, the OGSA security model must take them into account and provide mechanisms for cleanly traversing them, without compromising local control of firewall policy.
- *Securing the OGSA infrastructure:* The core grid service specification (OGSI) presumes a set of basic infrastructure services, such as handle map, registry, and factory services.

The OGSA security model must address the security of these components. In addition, securing lower level components that OGSI relies on would enhance the security of the OGSI Environment.

## 2.3 Grid Security Model

Industry efforts have rallied around Web services (WS) as an emerging architecture which has the ability to deliver integrated, interoperable solutions. Ensuring the integrity, confidentiality security of Web services through the application of a comprehensive security model is critical, both for organizations and their customers, which is the fundamental starting point for constructing virtual organizations. The secure interoperability between virtual organizations demands interoperable solutions using heterogeneous systems. For instance, the secure messaging model proposed by the Web Services Security roadmap [WSR] document supports both public key infrastructure (PKI) and Kerberos mechanisms as particular embodiments of a more-general facility and can be extended to support additional security mechanisms.

The security of a grid environment must take into account the security of various aspects involved in a grid service invocation. This is depicted in the figure 2.1 below.



**Fig 2.1:** Components of Grid Security Model.

### Binding Security:

The set of bindings to be considered includes SOAP (SOAP/HTTP, SOAP over a message queue or SOAP over any other protocol) and IIOP bindings. The security of a binding is based on the security characteristics of the associated protocol and message format.

If new protocols or message formats are introduced, care should be taken to address security requirements in those bindings so that, at a minimum, suitable authentication, integrity, and confidentiality can be achieved.

### **Policy Expression and Exchange:**

Web Services have certain requirements that must be met in order to interact with them. A hosting environment has access to policies associated with a hosted web service so that it can enforce the invocation requirements when the service is accessed. It is important for service requestors to know about the policies associated with a target service. Once the service requestor knows the requirements and supported capabilities of a target service, it can evaluate the capabilities and mechanisms that the service provider supports. At the end of the evaluation, both the service requestor and the service provider together select the optimal set of bindings to converse with one another. In a dynamic environment like the Grid, it is important for service requestors to discover these policies dynamically and make decisions at runtime. Such policies can be associated with the service definition, service data (i.e. part of grid service specification), or exchanged between service requestor and service provider.

Policy expression and exchange facilities will address the grid security requirements to exchange policy between participating end points, securing the OGSI infrastructure and play a critical part to achieve secure association between the end points.

### **Secure Association:**

A service requester and a service provider are likely to exchange more messages and submit requests subsequent to an initial request. In order for messages to be securely exchanged, policy may require service requester and service provider to authenticate each other. In that case, a mechanism is required so that they can perform authentication and establish a security context. This security context can be used to protect exchange of subsequent messages. The period of time over which a context is reused is considered a session or association between the interacting end points.

Facilitating secure association is required to establish the identity of a requester to the service provider (and vice versa) so that the service provider (and service requestor) can satisfy the requirements to authenticate the identity on the other end and then enforce authorization and privacy policies based on the established identity. The identities of the

requestor and service provider are required for auditing purposes, so that audit logs will contain information about accessing identity.

### **Identity and Credential Mapping/Translation:**

A grid environment consists of multiple trust (VOs) and security domains. Operations between entities in different domains will typically require mutual authentication. However the assumption that all domains may share a global user registry is unrealistic. Hence when operations between entities cross real domain as well as virtual organization boundaries, the identity of service requestors and providers, as well as their respective credentials as expressed in their home domain may not be syntactically or even semantically meaningful in their communication partner's domain. Enabling interoperation will thus require "federating" the involved domains and their respective security mechanisms, for example a Kerberos and a PKI domain.

This federation will typically be accomplished through mapping or translation of identities and/or credentials is required through proxies, gateways or trusted intermediaries. The mapping/translation components at this layer are responsible for implementing these functions as directed by corresponding policies. The definition of these policies is the subject of suitable management functions and trust models.

### **Privacy Enforcement:**

Maintaining anonymity or the ability to withhold private information is important in certain service environments. Organizations creating, managing, and using Grid services will often need to state their privacy policies and require that incoming service requests make claims about the service provider's adherence to these policies. The WS-Privacy specification will describe a model for how a privacy language may be embedded into WS-Policy descriptions.

The grid security model should adopt WS-Privacy in addition to WS-Policy to enforce privacy policies in a Grid environment. While the authorization and privacy functions in the grid security model build upon the WS-policy, WS-Authorization and WS-Privacy components, they do so by partitioning policy-related functions into specific functionality by abstracting the expression and exchange of policies from actual policy itself. Mechanisms to express, expose and exchange policies are covered by the policy expression and exchange

layer in the proposed grid security model. Enforcement of policies pertaining to service end points, federation, authorization and privacy should be built upon WS-Secure Conversation, WS-Federation, WS-Authorization and WS-Privacy in the WS-security architecture.

#### **Trust:**

Each member of a VO is likely to have a security infrastructure that includes authentication service, user registry, authorization engine, network layer protection and other security services. The security policies, authentication credentials and identities belonging to that member organization are likely to be managed, issued and defined within the scope of the organization – i.e., a security domain. In order to securely process requests that traverse between members of a VO, it is necessary for the member organizations to have established a trust relationship. Such trust relationships are essential for services accessed between the members to traverse network checkpoints (e.g., firewalls) and satisfy authorization policies associated with a service achieved by translating credentials from one domain to another (e.g., Kerberos to PKI) and mapping identities across security domains. Therefore, defining and establishing these trust relationships in a grid environment, i.e. defining VO membership, is a necessary foundation of the security model. Such a model needs to define direct or mutual trust relationships between two domains, as well as indirect trust relationships brokered through intermediaries. These relationships will then often materialize as rules for mapping identities and credentials among the involved organization domains.

The grid trust model should be based on the web services WS-Trust specification. Importantly, due to the dynamic nature of grids, trust relationships might also need to be established dynamically using trust proxies that act as intermediaries. Trust can be established and enforced based on trust policies defined either a-priori or dynamically. Once such a model is defined, this will play a role in defining how trust assertions are to be consumed by a service provider or a requester as the case may be. The model will also form the basis to satisfy the requirements to achieve single logon based on trust of asserting authority or trust on requesting member of a VO.

#### **Secure Logging:**

The grid security model explicitly calls for secure logging functionality as the necessary foundation for many higher-level audit-related services. Similar to trust model and



security management, secure logging is a basic service that is applicable to other components in the model.

**Management of Security:**

The grid security model groups all security management functions applicable to various aspects of binding, policy and federation. These include key management for cryptographic functions, user registry management, authorization, privacy and trust policy management and management of mapping rules which enables federation. It may also include the management of intrusion detection, anti-virus services and assurance information enabling service requestors to discover what security mechanisms and assurances a hosting environment can offer. Addressing the management of various aspects of the security infrastructure will satisfy the manageability requirement on the grid environment.

The grid environment and technologies address seamless integration of services with existing resources and core application assets. The grid security model is a framework that is extensible, flexible, and maximizes existing investments in security infrastructure. It allows use of existing technologies such as X.509 public-key certificates, Kerberos shared-secret tickets and even password digests. Therefore, it is important for the security architecture to adopt, embrace and support existing standards where relevant. Given grid services are based on web services, grid security model will embrace and extend the web services security standards proposed under the WS Security roadmap [WSR]. Specifically, given that OGSA is a service oriented architecture based on web services (i.e. WSDL based service definitions), the OGSA security model needs to be consistent with web services security model. The web services security roadmap provides a layered approach to address web services.

### 3.1 Security and Privacy

Security <sup>[5]</sup> and privacy are critical to grid systems. Typical features of security are:

**Authentication:** Authentication is the process of verifying the validity of a claimed individual and identifying who he or she is. Authentication is not limited to human beings; services, applications, and other entities may be required to authenticate also.

**Access control:** Assurance that each user or computer that uses the service is permitted to do what he or she asks for. The process of authorization is often used as a synonym for access control, but it also includes granting the access or rights to perform some actions based on access rights.

**Data integrity:** Data integrity assures that the data is not altered or destroyed in an unauthorized manner.

**Key management:** Key management deals with the secure generation, distribution, authentication, and storage of keys used in cryptography.

Privacy includes features such as:

**Data confidentiality:** Sensitive information must not be revealed to parties that it was not meant for.

**Anonymity:** Identity of sender or receiver is not identifiable in a network.

Security is essential feature of any communication over a network of users while privacy is not required in all cases. Privacy is a policy issue and is site specific. For example, Internet users who are concerned with censorship would like to publish or receive files without disclosing their identity. Privacy is becoming increasingly essential in today's world. For grid computing systems privacy is required, since data belongs to very large variety of domains.

### 3.2 Virtual organizations

The real and specific problem that underlies the Grid concept is *coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations*. The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a *Virtual Organization (VO)* <sup>[6]</sup>.

The following are examples of VOs: the application service providers, storage service providers, cycle providers, and consultants engaged by a car manufacturer to perform scenario evaluation during planning for a new factory; members of an industrial consortium bidding on a new aircraft; a crisis management team and the databases and simulation systems that they use to plan a response to an emergency situation; and members of a large, international, multiyear high energy physics collaboration. Each of these examples represents an approach to computing and problem solving based on collaboration in computation- and data-rich environments. As these examples show, VOs vary tremendously in their purpose, scope, size, duration, structure, community, and sociology.

Consider the grid in Figure 3.1. Suppose that six sites decide to share their resources and therefore form a virtual organization  $VO_1$ . Meanwhile, sites 4, 5 and 6 form another virtual organization  $VO_2$  for achieving a different task, sharing another subset of their resources for that task. We represent each virtual organization as  $VO_k = \{Entity\ set\}\{Task_k\}$ . Thus, in this example  $VO_2 = \{Site4, Site5, Site6\}\{T_2\}$ . The resource sharing policy can be defined at the entity level.

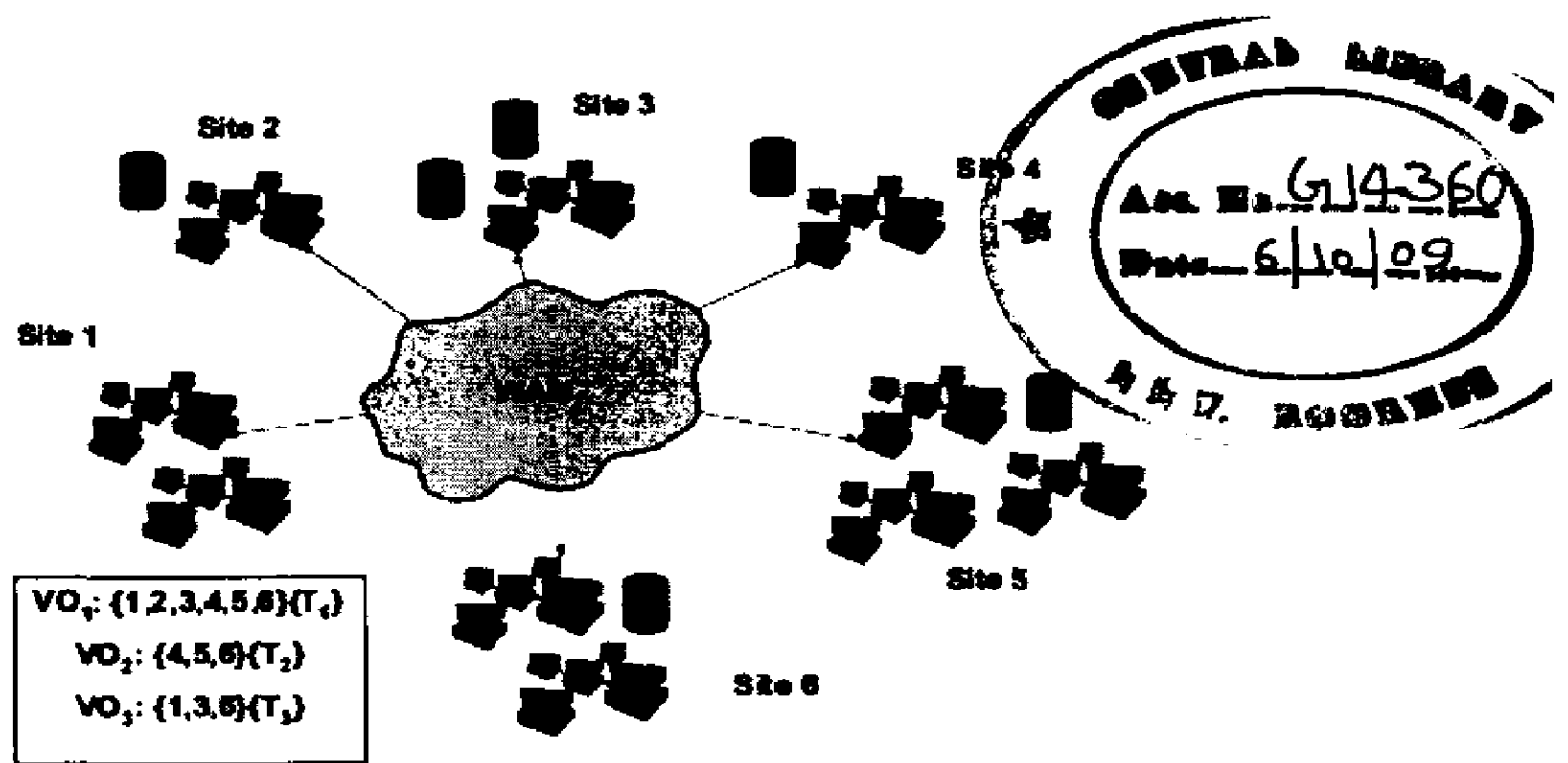


Fig 3.1: An example of a grid of multiple virtual organizations (VOs).

Privacy from site	Collaborative task	Privacy required?
1	$T_1, T_2, T_3$	No, Yes, No
2	$T_1, T_2, T_3$	No, Yes, Yes
3	$T_1, T_2, T_3$	No, Yes, No
4	$T_1, T_2, T_3$	No, No, Yes
5	$T_1, T_2, T_3$	No, No, Yes

Table 3.1: Privacy policy at site 6

	T1	T2	T3
Site-1	1	0	1
Site-2	1	0	0
Site-3	1	0	1
Site-4	1	1	0
Site-5	1	1	0
Site-6	1	1	1

Table 3.2: VO membership matrix for entire grid

Tables 3.1 and 3.2 shows the privacy policy of site 6 and the *VO membership matrix* of the entire grid. In the membership matrix  $M$ ,

$M[i][j] = 0$  : Site -  $i$  does not belong to  $VO_j$ .  
 $M[i][j] = 1$  : Site -  $i$  belongs to  $VO_j$ .

### 3.2.1 The Emergence of Virtual Organizations

Consider the following four scenarios:

1. A company needing to reach a decision on the placement of a new factory invokes a sophisticated financial forecasting model from an ASP, providing it with access to

appropriate proprietary historical data from a corporate database on storage systems operated by an SSP. During the decision-making meeting, what-if scenarios are run collaboratively and interactively, even though the division heads participating in the decision are located in different cities. The ASP itself contracts with a cycle provider for additional “oomph” during particularly demanding scenarios, requiring of course that cycles meet desired security and performance requirements.

2. An industrial consortium formed to develop a feasibility study for a next-generation supersonic aircraft undertakes a highly accurate multidisciplinary simulation of the entire aircraft. This simulation integrates proprietary software components developed by different participants, with each component operating on that participant’s computers and having access to appropriate design databases and other data made available to the consortium by its members.

3. A crisis management team responds to a chemical spill by using local weather and soil models to estimate the spread of the spill, determining the impact based on population location as well as geographic features such as rivers and water supplies, creating a short term mitigation plan (perhaps based on chemical reaction models), and tasking emergency response personnel by planning and coordinating evacuation, notifying hospitals, and so forth.

4. Thousands of physicists at hundreds of laboratories and universities worldwide come together to design, create, operate, and analyze the products of a major detector at CERN, the European high energy physics laboratory. During the analysis phase, they pool their computing, storage, and networking resources to create a “Data Grid” capable of analyzing petabytes of data.

These four examples differ in many respects: the number and type of participants, the types of activities, the duration and scale of the interaction, and the resources being shared. But they also have much in common. The dynamic nature of sharing relationships means that we require mechanisms for discovering and characterizing the nature of the relationships that exist at a particular point in time. For example, a new participant joining VO must be able to determine what resources it is able to access, the “quality” of these resources, and the policies that govern access.

Sharing relationships are often not simply client-server, but peer to peer: and sharing relationships can exist among any subset of participants. Sharing relationships may be combined to coordinate use across many resources, each owned by different organizations. For example, in VO, a computation started on one pooled computational resource may subsequently access data or initiate sub computations elsewhere. The ability to delegate authority in controlled ways becomes important in such situations, as do mechanisms for coordinating operations across multiple resources.

The same resource may be used in different ways, depending on the restrictions placed on the sharing and the goal of sharing. For example, a computer may be used only to run a specific piece of software in one sharing arrangement, while it may provide generic compute cycles in another. Because of the lack of a priori knowledge about how a resource may be used, performance metrics, expectations, and limitations (i.e., quality of service) may be part of the conditions placed on resource sharing or usage.

These characteristics and requirements define what we term a *virtual organization*. VOs enable disparate groups of organizations and/or individuals to share resources in a controlled fashion, so that members may collaborate to achieve a shared goal.

### 3.2.2 VO life cycle

VO lifecycle <sup>[7]</sup> include phases such as

- Identification,
- Formation,
- Operation/evolution, and
- Dissolution.

The identification phase deals with setting up the VO; this includes selection of potential business partners by using search engines or looking up registries. VO formation deals with partnership formation, including the VO configuration distributing information such as policies, agreements, etc., and the binding of the selected candidate partners into the actual VO. After the formation phase, the VO can be considered to be ready to enter the operation phase where the identified and properly configured VO members perform

according to their role. Membership and structure of VOs may evolve over time in response to changes of objectives or to adapt to new opportunities in the business environment. Finally, the dissolution phase is initiated when the objectives of the VO have been fulfilled. During the dissolution phase, the VO structure is dissolved and final operations are performed to annul all contractual binding of the partners. This involves the billing process for services used and an assessment of the performances of the respective participants.

### 3.3 Definitions and Notations

We define  $S$  to be the set of grid sites. Considering a grid of size  $n$ ,  $S = \{s_1, s_2, \dots, s_n\}$ . Let  $T$  be the total number of collaborative tasks in the grid. This is also equal to the total number of VOs in the grid. We represent the VO membership of a site  $k$  as the set  $VO(k)$  for all tasks and  $VO(k, t)$  for task  $t$ .  $VO'(k)$  represents the set of entities which are members of exactly the same VO(s) as  $k$ . The VO membership matrix is represented as  $M$ , where  $M[i][j] \in \{0, 1\} \forall 0 \leq i \leq n, 0 \leq j \leq T$ . For example, in Figure 3.1,  $VO(3) = \{1, 2, 4, 5, 6\}$ ,  $VO'(3) = \{1\}$  and  $VO(3, t_3) = \{1, 6\}$ .  $\Omega$  is the set of possible initiators of a job request. We call this the *initiator anonymity set*.  $\Psi$  is the set of possible forwarders for a given initiator and a job request. We call this the Forwarder Set. We also use a random variable  $X$ , which denotes the initiator for a given job request.

### 3.4 Trust

Trust<sup>[8]</sup> is an abstract concept that is highly context dependent<sup>[9]</sup>, but in Grid computing, is understood to mean competence in delivering or utilizing services in an agreed fashion. The trustworthiness of a Grid node is a function of its behaviors, and the trust value is necessarily subjective. Different nodes may have different degree of trust in a given node, depending on their current and past perception of the given node's behavior.

Two types of trust need to be addressed in Grid computing, for the resource/service providing node (provider), and the consuming node (consumer). From the consumer's point of view, we have *provider*<sup>[10]</sup> trust - i.e., trust in a resource/service provider that it will allocate resources to successfully deliver services promised. From the provider's point of view, we have *consumer trust* - this is the trust in a resource/service consumer that it will honestly utilize the requested service in the agreed way.

### **3.4.1 Trust Management in Virtual Organizations**

In this section we revisit the VO lifecycle, augmenting it with actions for trust management. Trust Management Systems (TMS) <sup>[11]</sup> can be divided into two main types: policy based TMS and reputation based TMS.

In policy-based TMS, the different entities that constitute the system exchange and maintain credentials to establish the trust relationships. The main goal in this kind of systems is to enable access control by verifying the credentials, certification trust and restricting access to credentials based predefined policies.

In reputation-based TMS, there exists a mechanism by which a system requesting a resource evaluates the trust of the system providing the resource. It is closely related to context trust. The trust values can be a function of the global and local reputation along with the different policies. Key elements in this type of systems are the reputation model, the metrics and how feedback is generated.

In relation to the VO lifecycle <sup>[12]</sup>, we distinguish four main phases: Identification, Formation, Operation, Evolution, and Dissolution.

#### **A. VO Identification:**

Before starting this phase, the creator of a VO should select the trust management systems to be used (policy-based or reputation-based TMSs) and the trust policies that will be used. Such information may be taken into account for searching for potential VO-partners. For instance, the parameters for the search may include in addition to service/resource descriptions, trust and reputation ratings, security grades, etc. The process may also involve metadata such as security and trust policies or Service Level Agreement (SLA) templates with ranges of possible values and/or dependencies between them. The identification phase ends with a list of candidates that potentially could perform the roles needed for the VO, taking into consideration trust and security information.

#### **B. VO Formation:**

In the formation phase, the list of potential VO candidates is reduced to the set of VO members. A central component in this phase is the VO manager, who negotiates with the VO



candidates their participation in the VO; selects the VO members, and distributes VO-level configuration information such as policies, SLAs, etc. The negotiation process include trust negotiation An important process in this phase is trust negotiation: the process by which all trust information, credential, reputation metrics, policies is negotiated between the VO manager and the VO members.

In principle, the intended formation may fail due to at least two reasons:

- No provider (or not enough providers) is able to fulfill all given requirements comes to SLA, trust, security etc., or
- Providers are not (fully) available at the specified time. In order to circumvent these problems, either the requirements may be reduced ("choose the best available") or the actual formation may be delayed to be re-launched at a more suitable time. Obviously there may be the case, where a general restructuring of the requirements led to a repetition of the identification phase

### **C. VO Operation:**

The operational phase could be considered the main lifecycle phase of a VO. During this phase the identified services and resources contribute to the actual execution of the VOs tasks by executing pre-defined business processes (e.g. a workflow of simulation processes and pre- and post processing steps). A lot of additional issues related to management and supervision are involved in this phase in order to ensure smooth operation of the actual task(s). Such issues cover recording of and reacting to participants' performance, updating and changing roles and therefore access rights of participants according to the current status of the executed workflow, carrying out financial arrangements (accounting, metering), etc.

In certain environments persistent information of all operations performed may be required to allow for later examination e.g. to identify fault-sources. Throughout the operation of the VO, service performance will be monitored. This will be used as evidence when constructing the reputation of the service providers. Any violation -e.g. an unauthorized access detected by the access control systems- and security threats -e.g. an event detected by an intrusion detection system- need to be notified to other members in order to take appropriate actions. Unusual behaviors may lead to both a trust re-assessment and a contract

adaptation. VO members will also need to enforce security at their local site. For example, providing access to services and adapting to changes and the violations.

#### **D. VO Evolution:**

Evolution is actually part of the operational phase: as participants in every distributed application may fail completely or behave inappropriately, the need arises to dynamically change the VO structure and replace such partners. This involves identifying new, alternative partners and services, as well as re-negotiating terms and providing configuration information as during identification. One of the main problems involved with evolution consists in re-configuring the existing VO structure so as to seamlessly integrate the new partner, possibly even unnoticed by other participants. Ideally, one would like the new service to take over the replaced partners' task at the point of its leaving without interruption and without having to reset the state of operation. There may be other reasons for participants joining or leaving the VO, mostly related to the overall business process, which might require specific services only for a limited period of time - since it is not sensible to provide an unused, yet particularly configured service to the VO for its whole lifetime, the partner may request to enter or leave the VO when not needed.

#### **E. VO Dissolution:**

During the dissolution phase, the VO structure is dissolved and final operations are performed to annul all contractual binding of the partners. This involves the billing process for used services and an assessment of the respective participants (or more specifically their resources) performances, like amount of SLA violations and reputation. The latter may of particular interest for further interactions respectively for other potential customers. Additionally it is required to revoke all security tokens, access rights, etc. in order to avoid that a participant may use its particular privileges. Generally the inverse actions of the formation phase have to be performed during Termination. Obviously partial termination operations are performed during evolution steps of the VO's operation phase.

### **3.4.2 Trust Model for Grid Applications**

We use the following trust model<sup>[13]</sup> in our protocol. A grid essentially consists of various virtual organizations, which represent different trust domains. Thus a many-to-many and symmetric trust relationships exist among the members of a VO. We use the term *trust*

set for the members of a virtual organization. In our trust model, we assume that if a site wants to send a job request to a remote site (such that  $job \in T_k$ ), then it can trust any forwarder in VO which corresponds to  $T_k$ . Moreover, since both  $I$  and  $F$  belong to the same VO (w.r.t  $T_k$ ), it is reasonable to assume that  $F$  would be aware of the type of job submissions at site  $I$ . We use this trust model in our analysis.

We use a simplified trust model in which each trust set is a virtual organization and therefore each entity in a VO trusts every other entity. Other possible variants can include cases where each trust set can have entities from different virtual organizations.

### 3.5 Existing Approaches to achieving anonymity

There are many approaches to achieving anonymity. Of these, most popular are crowds, onion routing and mixes.

**Crowds** <sup>[14]</sup>: aim at protecting the users' web-browsing anonymity. Like Onion Routing, the Crowds protocol uses a series of cooperating proxies (called jondos) to maintain anonymity within the group. Unlike Onion Routing, the sender does not determine the entire path. Instead, the path is chosen randomly on a hop-by-hop basis. Cycles are allowed on the path. Once a path is chosen, it is used for all the anonymous communication from the sender to the receiver within a 24-hour period. At some specific time instant, new members can join the crowd and new paths can be formed.

**Mix** <sup>[15]</sup>: is a store-and-forward device that accepts a number of fixed-length messages from different sources, discards repeats, performs a cryptographic transformation on the messages, and then outputs the message to the next destination in an order not predictable from the order of inputs. A Mix based approach then sends messages over a series of such independent mix.

**Anonymizer** <sup>[16]</sup>: provides fast, anonymous, interactive communication services. Anonymizer in this approach is essentially a web proxy that filters out the identifying headers and source addresses from web client requests. Instead of a user's true identity, a web server can only learn the identity of the *Anonymizer Server*. In this approach, all rerouting paths have a single intermediate node, which is the Anonymizer Server.

**Hordes:** employs multiple jondos similar to those used in the Crowds protocol to anonymously route a packet towards the receiver. It uses multicast services, however, to anonymously route the reply back to the sender instead of using the reverse path of the request. Similar to Crowds, Hordes also allows cycles on the forwarding path.

### 3.6 2-Hop Forwarding Protocol

In a 2-hop forwarding protocol <sup>[17]</sup> (figure 3.2) proposed by Sourik Ray *et al*, the initiator routes the job request (e.g. input data, executables) through an intermediate forwarder and the job output is forwarded along the reverse path back to the initiator. We assume the existence of a trusted server called the Trust Set Maintenance (TSM) server, which maintains and periodically updates the VO memberships of the different entities in the grid. The TSM server generates the forwarding path for the initiator of the job request. This function may be integrated into some grid service nodes, such as job schedulers. The criterion for selecting the forwarder is described in details in the analysis section.

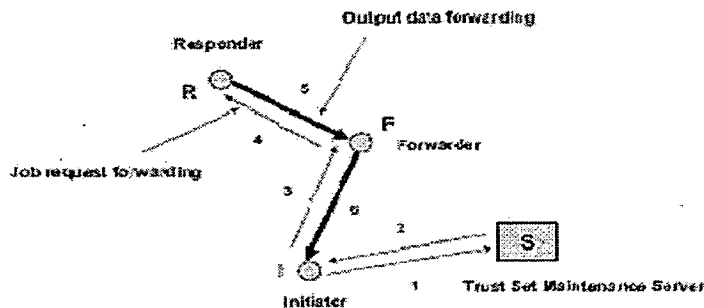


Fig 3.2: 2-hop forwarding protocol

The protocol works effectively in a passive, internal <sup>[18]</sup> threat model, in which the adversary is from the responder site, e.g. the administrator at the site or an intruder who has acquired the administrator privilege. For each incoming job request, the adversary will try to predict with a certain probability whether the site sending the request is the forwarder or the actual initiator (note that the forwarding protocol is used by the initiator only when it needs anonymity from the responder site). Moreover, the adversary may have partial or complete

knowledge of the VO memberships of all the virtual organizations in the grid and can use this information to predict the Initiator anonymity set.

### 3.6.1 Analysis of the Degree of Anonymity

We evaluate the entropies before and after the receipt of a job request at the responder site and analyze the degree of anonymity achieved for two extreme cases: (1) the responder site has information on only its own VO membership; and (2) the responder site knows the entire VO membership, i.e. the matrix  $M$ . We also assume a uniform probability distribution over the anonymity set.

#### Case 1: The responder site has the least information

In this case, the adversary (responder site) has membership information about the VOs to which it belongs, but it does not know the VO membership of other entities. According to the design of the protocol, a site uses a forwarder to route a job request only when it requires anonymity from the responder site. Moreover, all sites which are members of the same VO and collaborate for a common goal do not require anonymity from each other. Thus the responder can eliminate only itself from the anonymity set and therefore  $|\Omega| = n-1$ , and  $P(X=s) = 1/n-1$ , and then

$$\begin{aligned} H(X)_{\text{apriori}} &= -\sum_{s \in \Omega} P(X=s) \log_2 P(X=s) \\ &= \log_2(n-1) \end{aligned} \quad (3.1)$$

After receiving the job request, the adversary gets to know the identity of the forwarder and can possibly predict the collaborative task to which the job belongs (without loss of generality, we can assume that the responder can predict the task from the executables and/or input data file sent by the initiator). Since the adversary has no information on the VO membership of  $F$ , it cannot predict the anonymity set and therefore for a large grid, a posteriori entropy is approximately equal to the a priori entropy and complete anonymity is achieved.

$$\begin{aligned} H(X)_{\text{a posteriori}} &= \log_2(n-2) \\ d(\Omega) &= H(X)_{\text{a posteriori}} / H(X)_{\text{apriori}} \\ &= \log_2(n-2) / \log_2(n-1) \approx 1 \end{aligned} \quad (3.2)$$

**Case 2: The responder site has the maximum information**

In this case, the adversary knows the entire VO membership matrix  $M$ . Before observing the job request, the responder can eliminate only those entities from the anonymity set which belong to exactly the same VO(s) as the responder. Knowing the membership of  $F$ , it can derive the a posteriori anonymity set to be  $VO(F, T_k)$ , where  $T_k$  is the task to which the job belongs.

Thus, a posteriori entropy is given by:

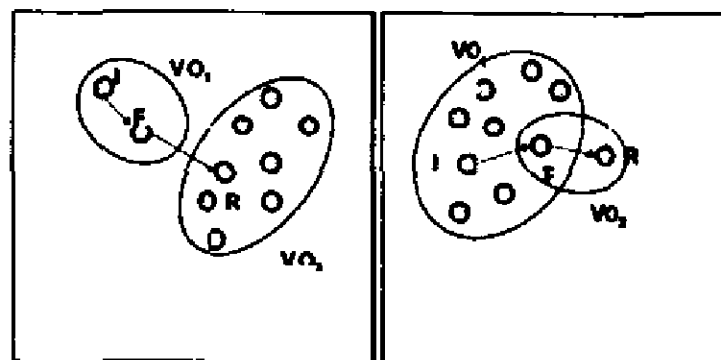
$$H(X)_{\text{a posteriori}} = \log_2 |VO(F, T_k)| \tag{3.3}$$

$$\begin{aligned} d(\Omega) &= H(X)_{\text{a posteriori}} / H(X)_{\text{apriori}} \\ &= \log_2 |VO(F, T_k)| / \log_2 (n - 1 - VO'(R)) \end{aligned} \tag{3.4}$$

We analyze the variation of the degree of anonymity for different grid configurations for a given grid size.

We have the following observations:

- The minimum degree of anonymity ( $d(\Omega) = 0$ ) corresponds to the case when the initiator is a member of a virtual organization such that  $|VO(I, T_k)| = 1$ . This means that  $I$  has to use the only other member as the forwarder for collaborative task  $T_k$ .
- The maximum anonymity ( $d(\Omega) = 1$ ) corresponds to the case when
  - There are exactly two virtual organizations, one corresponding to  $T_k$  and the other corresponding to some other collaborative task;
  - $F$  is the only common member of both the VOs; and
  - $|VO'(R)| = 0$ .



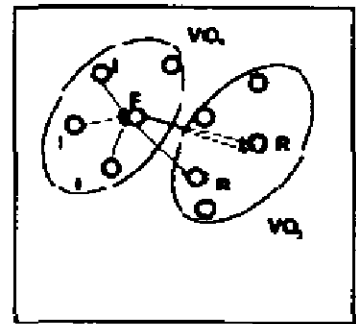
**Fig 3.3:** minimum and maximum anonymity for  $n=10$

The ratio of the size of the VO membership set of  $F$ , the  $VO'$  membership set of  $R$  and the number of virtual organizations influences the degree of anonymity that can be achieved. For a given  $|VO'(R)|$ , the anonymity increases as  $|VO(F, Tk)|$  increases. Intuitively we can say this observation is valid because if the initiator is a member of a VO which has many members, the anonymity set is larger and therefore a higher degree of anonymity can be achieved.

### 3.6.2 Disadvantages of 2-Hop Forwarding Protocol

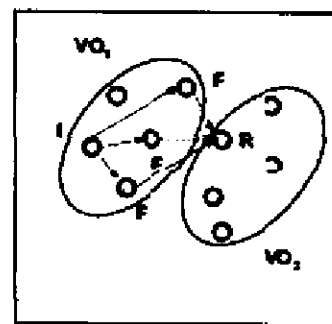
There are three main disadvantages of 2-hop forwarding protocol:

*Load imbalancing due to constant forwarder:* If same site is used as forwarder by many members of a VO then there will be load imbalances and delays.



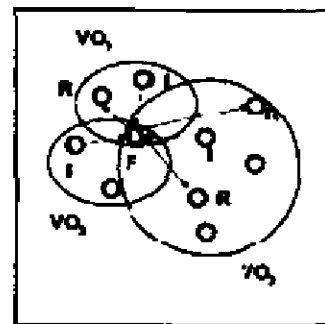
**Fig 3.4:** Load imbalance due to selection of constant forwarder

*Intersection attacks:* In an intersection attack, if the adversary knows that the initiator is in two different sets  $A$  and  $B$ , then the anonymity of the initiator is reduced to  $A \cap B$ .



**Fig 3.5:** improved load balance but possibility of intersection attack.

*Load imbalance due to selection of a forwarder which is a member of several VOs :*



**Fig 3.6:** load imbalance due to selection of a forwarder which is a member of several VOs.

### 3.6.3 Selection of Forwarder and Possible Attacks

From the analysis in the previous section, we observed that the size of the VO to which the initiator belongs w.r.t task  $T_k$  influences the degree of anonymity that can be achieved. Since in our trust model, we assume that the initiator trusts a forwarder only if it belongs to the same VO which corresponds to the collaborative task  $T_k$ , the selection of the forwarder is limited to the same VO. The selection of the same forwarder repeatedly for job request routing and data forwarding can lead to load imbalance and generation of hotspots. On the other hand, a random selection of the forwarder can lead to intersection attacks <sup>[19]</sup>. In an intersection attack, if the adversary knows that the initiator is in two different sets  $A$  and  $B$ , then the anonymity of the initiator is reduced to  $A \cap B$ . Moreover, a site which is a member of several VOs has a very high probability of being selected as the forwarder. Thus the selection of the forwarder is a tradeoff between load balance and higher degree of anonymity.



## 4.1 Introduction

The need for a metric to measure the performance of anonymity implementations appeared with the development of applications that enabled anonymous electronic transactions, such as untraceable email, electronic voting, anonymous e-coins or privacy-enhanced web browsing.

The work presented in this chapter is a contribution to the field of privacy enhancing technologies by Claudia Díaz *et al*, and was published in the 2nd Workshop on Privacy Enhancing Technologies 2002.

The anonymity metrics <sup>[20]</sup> presented here can be applied to concrete systems, adversaries and conditions. These metrics give a measure of the size and distinguish ability of the set of subjects potentially linked to a particular transaction, and attacked by a concrete adversary. In order to get an idea on the performance of an anonymity implementation under different conditions, multiple anonymity measurements must be made and analyzed.

Information theoretic metrics can be applied to a broad range of anonymity systems. It is thus important to understand the concepts behind entropy-based anonymity metrics in order to apply and interpret them correctly in concrete scenarios. The metrics must be adapted to the anonymity system under study, and the computation of probability distributions that lead to meaningful metric values is not always obvious.

### 4.1.1 Defining Anonymity

Pfitzmann and Hansen <sup>[21]</sup> defined anonymity as the state of being not identifiable within a set of subjects, the anonymity set. According to the Pfitzmann-Hansen definition of anonymity, the subjects who may be related to an anonymous transaction constitute the anonymity set for that particular transaction. A subject carries on the transaction anonymously if he cannot be distinguished (by an adversary) from other subjects. This definition of anonymity captures the probabilistic information obtained by adversaries trying to identify anonymous subjects.

## 4.2 Model

In this chapter, we present a general model for anonymity systems. Many anonymity systems can be modeled in terms of unlinkability. Unlinkability is defined by Pfitzmann and Hansen as follows: *unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, events, actions, ...)* means that within the system (comprising these and possibly other items), from the attacker's perspective, these items of interest are no more and no less related after his observation than they are related concerning his apriori knowledge. Our model can be applied both to sender and recipient anonymity. If we consider the sending and receiving of messages as Items of Interest (IOIs), anonymity may be defined as unlinkability of an IOI and a subject. More specifically, we can describe the anonymity of an IOI such that it is not linkable to any subject, and the anonymity of a subject as not being linkable to any IOI. In this context, unlinkability is achieved with high entropy values. Figure 4.1 presents a simplified anonymity model. The goal of anonymity systems is to hide the relationship between subjects and IOIs. Hiding these links is the basic mechanism behind anonymous transactions.

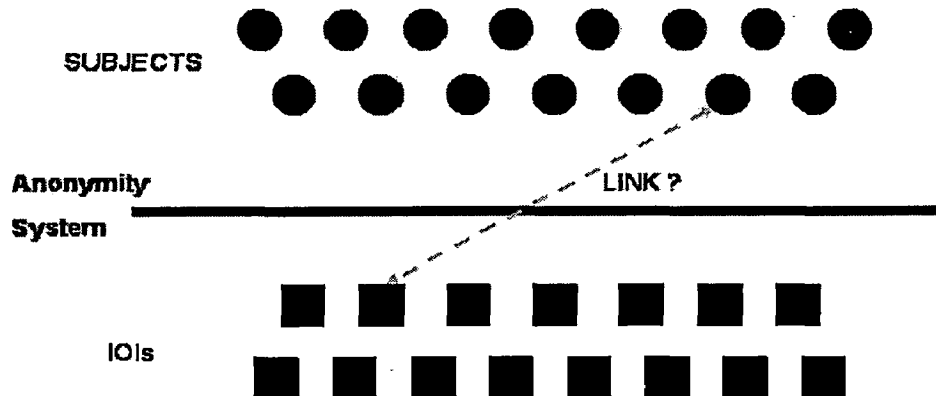


Fig 4.1: model for anonymity systems.

An observer of the system sees that a set of subjects are accessing the anonymity system. At the output of the system, they see IOIs which are hard to link to a particular subject. The set of subjects who might be linked to an IOI is called the anonymity set. The

### 4.2.1 Taxonomy of Attackers and Their Possible Attacks

In the context of this thesis, an attacker is an entity that deliberately tries to compromise the anonymity of one or more users of a computer network. Attackers can be classified according to which kind of attacks they are capable of launching. Attackers can be either *passive* or *active*. An active attacker can modify the traffic in a network, while a passive attacker (also called eavesdropper) is restricted to observing the traffic. Attackers can further be classified as either *local* or *global* attackers. Local attackers launch their attacks in a subset of the network while global attackers launch their attacks on the whole network.

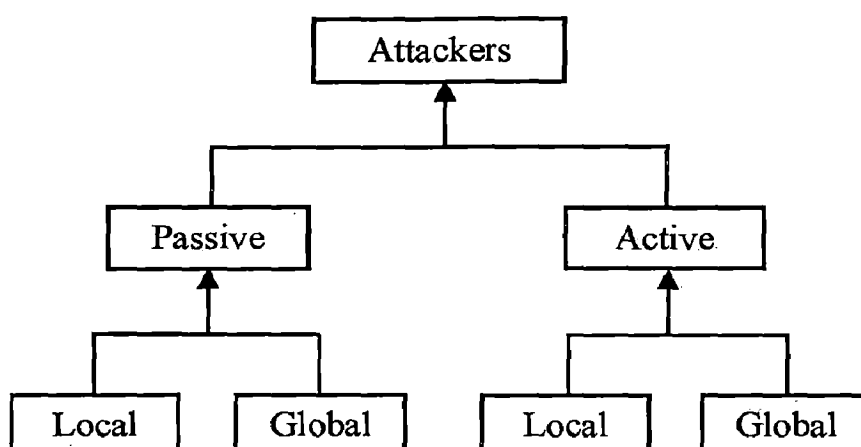


Fig 4.2: Taxonomy of the attackers.

The general strategy of an attacker is to obtain probabilistic relationships between input and output messages of an anonymity proxy to be able to narrow down the set of possible senders or recipients. The result of an attack could be that one sender appears to be the originator of a message with a high probability. If the attacker succeeds in reducing the size of the anonymity set into a singleton, the sender is unambiguously identified.

### 4.3 Information Theoretic Anonymity Metrics

In this section, we first introduce the concept of entropy, on which information theoretic anonymity metrics are based. Then, we explain how the effective anonymity set size and the degree of anonymity can be computed.

### 4.3.1 Entropy

The information theoretic concept of entropy <sup>[22]</sup> provides a measure of the uncertainty of a random variable. Let  $X$  be the discrete random variable with probability mass function  $p_i = \Pr(X = i)$ , where  $i$  represents each possible value that  $X$  may take with probability  $p_i > 0$ . In this case, each  $i$  corresponds to a subject of the anonymity set; i.e.,  $p_i$  is the probability of subject  $i$  being linked to the IOI.

We denote by  $H(X)$  the entropy of a random variable, and by  $N$  the number of subjects in the anonymity set.  $H(X)$  can be calculated as:

$$H(X) = - \sum_{i=1}^N p_i \log_2 (p_i), \quad i=1 \text{ to } N \quad (4.1)$$

### 4.3.2 Effective Anonymity Set Size

The effective anonymity set size is an intermediate step to compute the degree of anonymity. Serjantov and Danezis proposed in <sup>[23]</sup> the use of the effective anonymity set size as metric. As mentioned in Sect 4.1.1, anonymity was defined by Pfizmann and Hansenin as the state of being not identifiable within a set of subjects, the anonymity set. Anonymity metrics aim at giving a meaningful measure of the anonymity set size.

After deploying an attack on an anonymity system, the adversary typically obtains a distribution of probabilities that link subjects to the particular IOI of the attack. The probabilities are shown in Fig. 4.3 with the arrows that connect the IOI to the subjects of the anonymity set. Different subjects may appear as having a higher or lower probability  $p_i$  of having a link with the IOI, depending of the information obtained by the adversary using the attack.

Let  $N$  denote the total number of subjects which are linked to the IOI with a non-zero probability ( $p_i > 0, i = 1 \dots N$ ). The effective anonymity set size is defined as the entropy  $H(X)$  of the distribution  $X$  of probabilities that link the subjects of the anonymity set to the IOI.

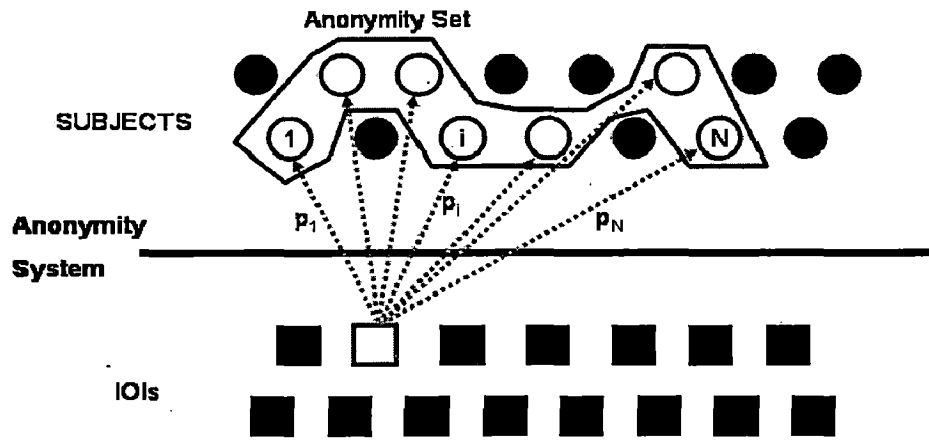


Fig 4.3: Anonymity set.

Entropy-based anonymity metrics give a measure of the uncertainty of the adversary on the subject who is related to the IOI. The effective anonymity set size takes into account the number of potential subjects linked to the IOI, and the probabilities assigned to the subjects. The more equally distributed the probabilities assigned to the subjects of the anonymity set, the higher the entropy (i.e., the higher the effective anonymity set size).

### 4.3.3 Degree of Anonymity

The degree of anonymity is a normalized version of the effective anonymity set size, which tells how good the system is performing on a 0 – 1 scale. The maximum effective anonymity set size for  $N$  subjects is reached when all subjects are linked to the IOI with equal probability (i.e.,  $p_i = 1/N$ ). In this case, all subjects are indistinguishable towards the adversary with respect to the IOI. For a given number  $N$  of users, the maximum achievable anonymity corresponds to the entropy of a uniform distribution.

We denote the maximum entropy by  $H_M$ :

$$H_M = \log_2 (N) \quad (4.2)$$

If we assume that the adversary has no a priori information on the system (i.e., the a priori anonymity of an IOI is  $H_M$ ), the amount of information gained by the adversary with an attack is the difference in the entropy before and after the attack, that is:  $H_M - H(X)$ .

The degree of anonymity is defined as the normalized value of this difference

in knowledge of the adversary:

$$d = 1 - (H(X) - H_M) / H_M = H(X) / H_M \quad (4.3)$$

$d = 0$  when a user appears as being the originator of a request with probability 1.

$d = 1$  when all users appear as being the originator with the same probability.

As we can observe in the formula, the degree of anonymity is obtained dividing the effective anonymity set size by the maximum entropy for a given number of subjects. This degree evaluates how much anonymity is provided by a system independently from the number of users. Given a certain number of subjects, the computation of the degree of anonymity gives an idea on how close the anonymity is to the maximum achievable.

Both metrics are computed using the same information, and one can trivially be computed from the other. The difference is, however, that the effective anonymity set size ties the anonymity to the actual number of users in the system; while the degree of anonymity makes abstraction on the number of users and focuses on the performance of the system (i.e., how close it is to the maximum achievable anonymity).

#### 4.3.4 Average degree of anonymity

The proposed model allows us to calculate the degree of anonymity obtained for each request. Given that during the attack  $R$  requests have been produced, we define the average degree of anonymity as:

$$D = \sum d_j / R, \quad j=1 \text{ to } R; \quad (4.4)$$

This gives an accurate idea on the degree of anonymity provided by the system for request on average.

We use a 3-hop forwarding protocol (figure 5.1) in which the initiator routes the job request (e.g. input data, executables) through an intermediate forwarder of same VO and this forwarder then routes the request through a member of any VO (same or different) and the job output is forwarded along the reverse path back to the initiator. We assume in our protocol that intermediate forwarder in different VO has probability  $P_f$  of being corrupted. As in 2-hop forwarding protocol we assume the existence of a trusted server called the Trust Set Maintenance (TSM) server, which maintains and periodically updates the VO memberships of the different entities in the grid. The TSM server generates the forwarding path for the initiator of the job request. This function may be integrated into some grid service nodes, such as job schedulers.

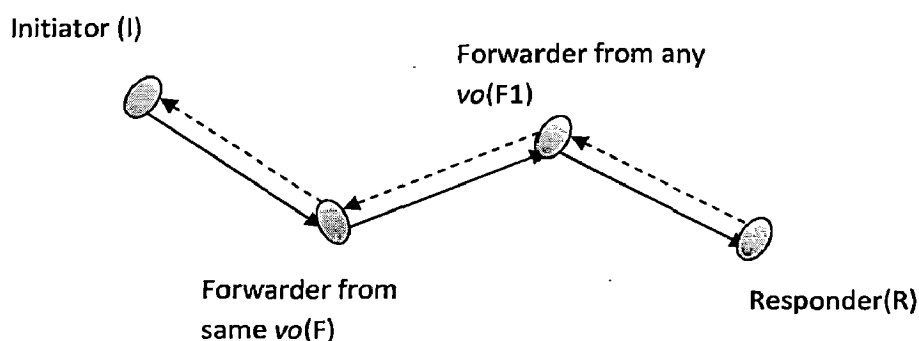


Fig 5.1: 3-hop forwarding protocol

## 5.1 Protocol

The procedure of creating a forwarding path is as follows:

**Step 1:** The initiator contacts the TSM server  $S$  for forwarder selection. We assume that the initiator contacts a job scheduler to determine the identity of the responder.

**Step 2:** The trusted server sends the forwarding path to  $I$ , which is encrypted with the public key of  $I$ . (We assume the existence of a public key infrastructure.) A randomly generated number  $n$  can be used to identify the transaction.

$$S \rightarrow I: \{n, F\}K_I$$

**Step 3:**  $I$  sends the job request to  $F$  and also signs the job request message using a certificate issued by a certificate authority for authentication. The job may consist of input data files, executables etc.

$$I \rightarrow F: \{\{Job, n, R\}K_F\}C_I$$

**Step 4:**  $F$  sends the job request to  $F1$  and also signs the job request message using a certificate issued by a certificate authority for authentication. The job may consist of input data files, executables etc.

$$F \rightarrow F1: \{\{Job, n, R\}K_{F1}\}C_f$$

**Step 5:**  $F1$  authenticates itself to  $R$  and forwards the job request on  $I$ 's behalf.

$$F1 \rightarrow R: \{\{Job, n\}K_R\}C_{F1}$$

**Steps 6,7,8:** The output data is forwarded back to the initiator through the forwarder  $F1$ .

$$R \rightarrow F1: \{Output, n\} K_{F1}$$

$$F1 \rightarrow F: \{Output, n\} K_{F1}$$

$$F \rightarrow I: \{Output, n\} K_{F1}$$

### 5.1.1 Definitions and notations

We use notations mentioned in section 3.3 for 2-hop forwarding protocol in our analysis. We define notations for the intermediate forwarder  $F1$ . We denote the VO to which  $F1$  belongs to as  $VO(F1, T_k)$  for task  $k$ .

## 5.2 Analysis of Degree of Anonymity

We evaluate the entropies before and after the receipt of a job request at the responder site and analyze the degree of anonymity achieved for two extreme cases:

- (1) The responder site has information on only its own VO membership; and
- (2) The responder site knows the entire VO membership, i.e. the matrix  $M$ . We also assume a uniform probability distribution over the anonymity set.



**Step 3:**  $I$  sends the job request to  $F$  and also signs the job request message using a certificate issued by a certificate authority for authentication. The job may consist of input data files, executables etc.

$$I \rightarrow F: \{\{Job, n, R\}K_F\}C_I$$

**Step 4:**  $F$  sends the job request to  $F1$  and also signs the job request message using a certificate issued by a certificate authority for authentication. The job may consist of input data files, executables etc.

$$I \rightarrow F: \{\{Job, n, R\}K_{F1}\}C_f$$

**Step 5:**  $F1$  authenticates itself to  $R$  and forwards the job request on  $I$ 's behalf.

$$F \rightarrow R: \{\{Job, n\}K_R\}C_{F1}$$

**Steps 6,7,8:** The output data is forwarded back to the initiator through the forwarder  $F1$ .

$$R \rightarrow F: \{Output, n\}K_{F1}$$

$$F1 \rightarrow F: \{Output, n\}K_{F1}$$

$$F \rightarrow I: \{Output, n\}K_{F1}$$

### 5.1.1 Definitions and notations

We use notations mentioned in section 3.3 for 2-hop forwarding protocol in our analysis. We define notations for the intermediate forwarder  $F1$ . We denote the VO to which  $F1$  belongs to as  $VO(F1, T_k)$  for task  $k$ .

## 5.2 Analysis of Degree of Anonymity

We evaluate the entropies before and after the receipt of a job request at the responder site and analyze the degree of anonymity achieved for two extreme cases:

- (1) The responder site has information on only its own VO membership; and
- (2) The responder site knows the entire VO membership, i.e. the matrix  $M$ . We also assume a uniform probability distribution over the anonymity set.

### Case 1: The responder site has the least information

In this case, the adversary (responder site) has membership information about the VOs to which it belongs, but it does not know the VO membership of other entities. According to the design of the protocol, a site uses a forwarder to route a job request only when it requires anonymity from the responder site. Moreover, all sites which are members of the same VO and collaborate for a common goal do not require anonymity from each other. Thus the responder can eliminate only itself from the anonymity set and therefore  $|\Omega| = n-1$ , and  $P(X=s) = 1/n-1$ , and then.

$$\begin{aligned} H(X)_{\text{apriori}} &= -\sum_{s \in \Omega} P(X=s) \log_2 P(X=s) , \\ &= \log_2(n-1) \end{aligned} \quad (5.1)$$

After receiving the job request, the adversary gets to know the identity of the forwarder and can possibly predict the collaborative task to which the job belongs (without loss of generality, we can assume that the responder can predict the task from the executables and/or input data file sent by the initiator). Since the adversary has no information on the VO membership of  $F$ , it cannot predict the anonymity set and therefore for a large grid, a posteriori entropy is approximately equal to the a priori entropy and complete anonymity is achieved.

$$H(X)_{\text{a posteriori}} = \log_2(n-2) \quad (5.2)$$

$$\begin{aligned} d(\Omega) &= H(X)_{\text{a posteriori}} / H(X)_{\text{apriori}} \\ &= \log_2(n-2) \\ &= \log_2(n-1) \approx 1 \end{aligned} \quad (5.3)$$

### Case 2: The responder site has the maximum information

In this case, the adversary knows the entire VO membership matrix  $M$ . Before observing the job request, the responder can eliminate only those entities from the anonymity set which belong to exactly the same VO(s) as the responder. In our model we assumed that intermediate forwarder  $F_1$  has probability  $P_f$  to be corrupted or  $1-P_f$  to be not corrupted.

In first case the responder has information of the first forwarder. So this case reduces to that of two hop forwarding protocol. Degree of Anonymity in this case is

$$H(X)_{\text{a posteriori}} = \log_2 |VO(F, T_k)| \quad (5.4)$$

$$d_1(\Omega) = \frac{H(X)_{\text{a posteriori}}}{H(X)_{\text{a priori}}} = \frac{\log_2 |VO(F, T_k)|}{\log_2 (n-1-VO'(R))} \quad (5.5)$$

In latter case the degree of anonymity achieved is

$$H(X)_{\text{a posteriori}} = \log_2 (n-1-VO'(R)-VO'(F1)) \quad (5.6)$$

$$H(X)_{\text{a priori}} = \log_2 (n-1-VO'(R)) \quad (5.7)$$

$$d_2(\Omega) = \frac{\log_2 (n-1-VO'(R)-VO'(F1))}{\log_2 (n-1-VO'(R))} \quad (5.8)$$

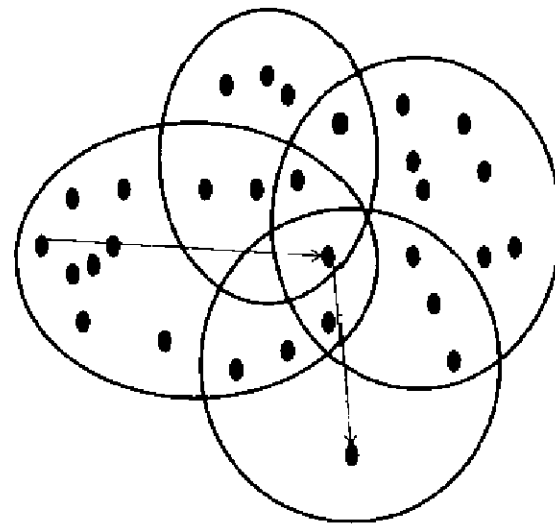
Therefore the total degree of anonymity can be calculated as

$$\text{Degree of anonymity } d = P_f d_1(\Omega) + (1-P_f) d_2(\Omega) \quad (5.9)$$

We analyze the variation of the degree of anonymity for different grid configurations for a given grid size and for different cases.

We have the following observations:

- The minimum degree of anonymity ( $d(\Omega) = 0$ ) corresponds to the case when the forwarder  $F1$  is compromised and the initiator is a member of a virtual organization such that  $|VO(I, T_k)| = 1$ . This means that  $I$  has to use the only other member as the forwarder for collaborative task  $T_k$ .
- The maximum anonymity ( $d(\Omega) = 1$ ) corresponds to the case when
  - Forwarder  $F1$  belongs to all virtual organizations, and
  - $|VO'(R)| = 0$ .



**Fig 5.2:** Maximum Degree of Anonymity

We use OMNeT++ to evaluate performance of our protocol. We use data transfer latency and bandwidth consumption as metrics for evaluating the overhead associated with the protocol as compared to the case when initiator anonymity is not required. Note that system overhead at the forwarders is roughly proportional to the bandwidth consumption. In our simulations we generate random transactions, where each transaction is the routing of a job request from the initiator to the responder through a forwarder. For ease of implementation, we assume that a remote job is executed at the responder in its entirety. For each transaction we measure the degree of anonymity, the file transfer latency and bandwidth consumption.

### 6.1 OMNeT++

OMNeT++ is an object-oriented modular discrete event network simulator. The simulator can be used for Traffic modeling of telecommunication networks, protocol modeling, modeling queuing networks, modeling multiprocessors and other distributed hardware systems, validating hardware architectures, evaluating performance aspects of complex software systems and modeling any other system where the discrete event approach is suitable.

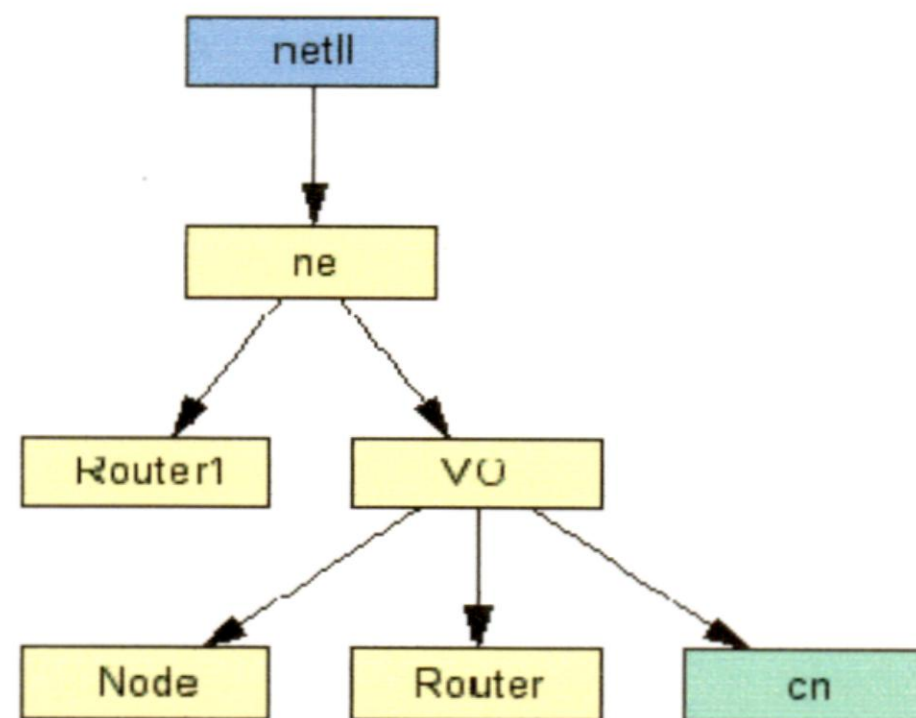
OMNeT++ provides efficient tools for the user to describe the structure of the actual system.

The topology of a model is specified using the NED language. The NED language facilitates the modular description of a network. This means that a network description may consist of a number of component descriptions (channels, simple/compound module types). The channels, simple modules and compound modules of one network description can be reused in another network description.

An OMNeT++ model consists of hierarchically nested modules. Modules communicate through message passing. Messages can contain arbitrarily complex data structures. Modules can send messages either directly to their destination or along a predefined path, through gates and connections. Modules can have their own parameters. Parameters can be used to customize module behaviour and to parameterize the model's

topology. Modules at the lowest level of the module hierarchy encapsulate behaviour. These modules are termed simple modules, and they are programmed in C++ using the simulation library.

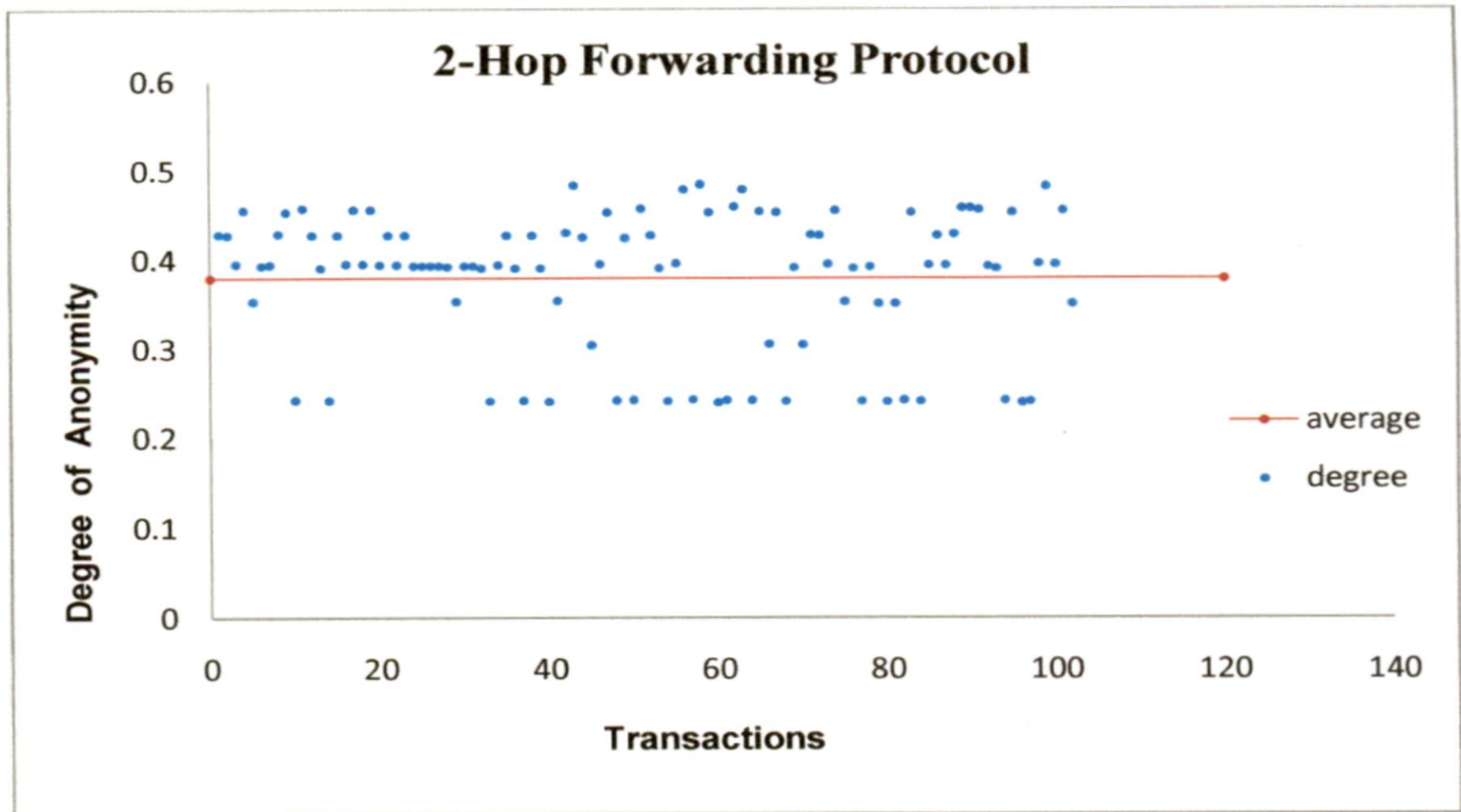
For this simulation we created a network named netll which contains VO modules and simple module external router which connects all VO's .Module VO consists of the simple modules node and router.cn is the channel with parameter delay. Module VO has parameter “n” which



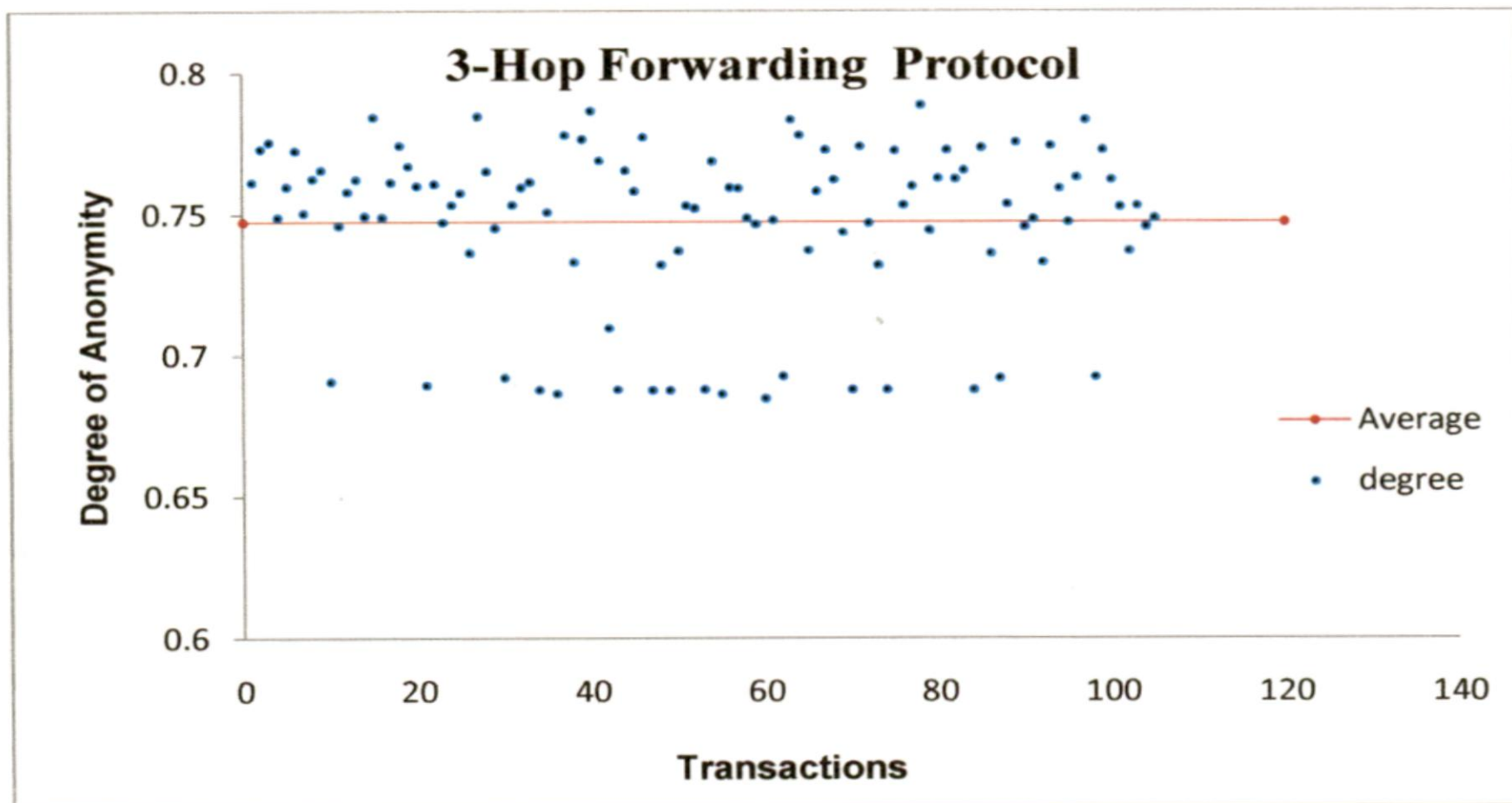
**Fig 6.1:** Simulation Model in OMNeT++

gives the number of nodes in each VO. This value is set to generate random number of nodes. We generate network configuration consisting of 15 VOs, where each VO contains 4-10 nodes and a router connecting all these nodes internally .Router is connected to the module interface. We use the 3-hop forwarding protocol when the adversary has the *maximum information*. We present below the Degree of Anonymity and delay graphs of 2-hop and 3-hop forwarding protocols.

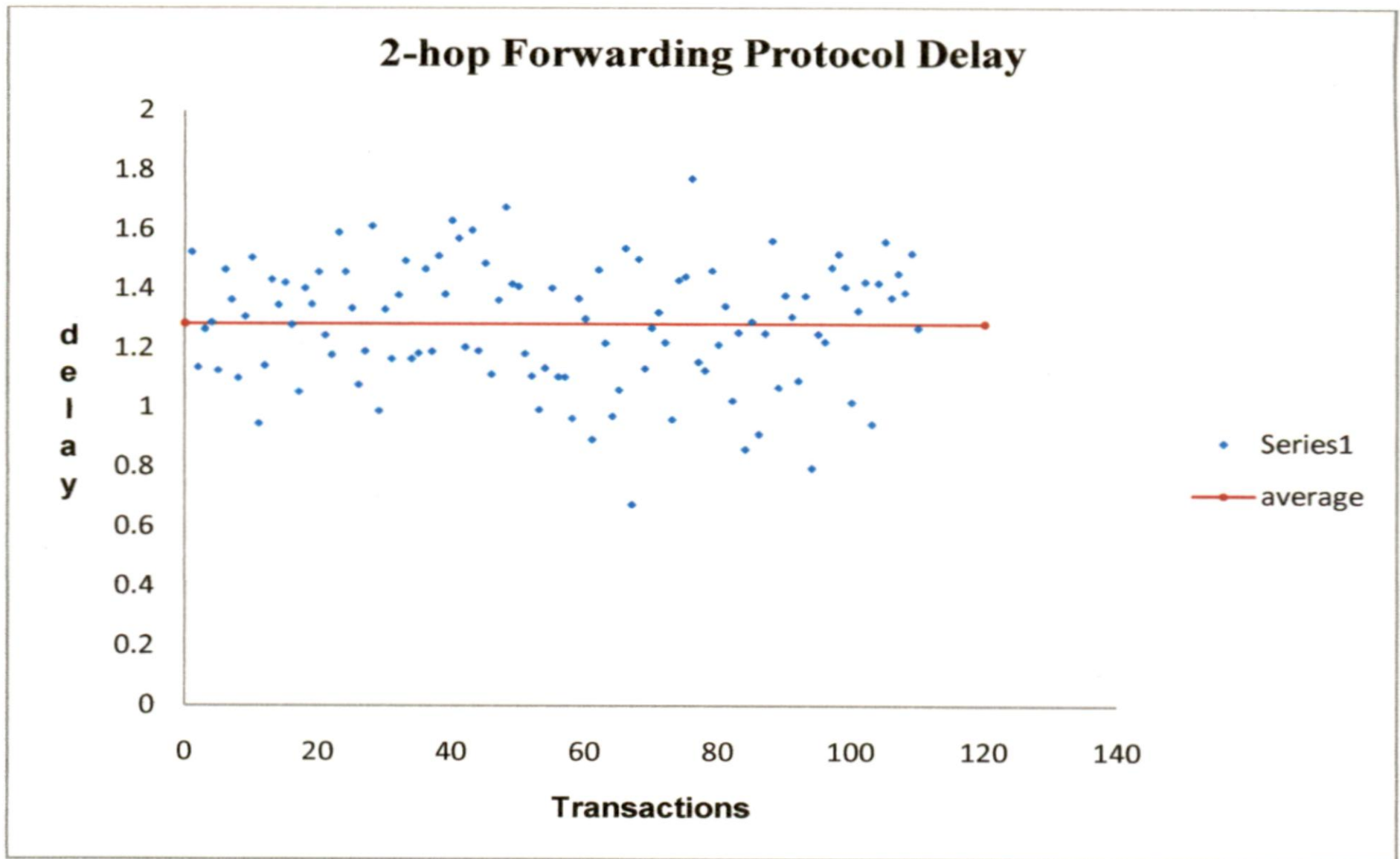
As we can see from graphs presented next the average degree of anonymity(Figs 6.2 & 6.3) achieved for a 2-hop forwarding protocol is approx. 0.4 while that of 3-hop forwarding protocol is 0.7 .While average delay(Figs 6.4 & 6.5) is 1.2 in case of 2- hop protocol it is 1.8 in case of 3-hop protocol. . Thus this protocol is applicable for grid systems that require anonymity but can only accommodate modest overhead in achieving initiator anonymity.



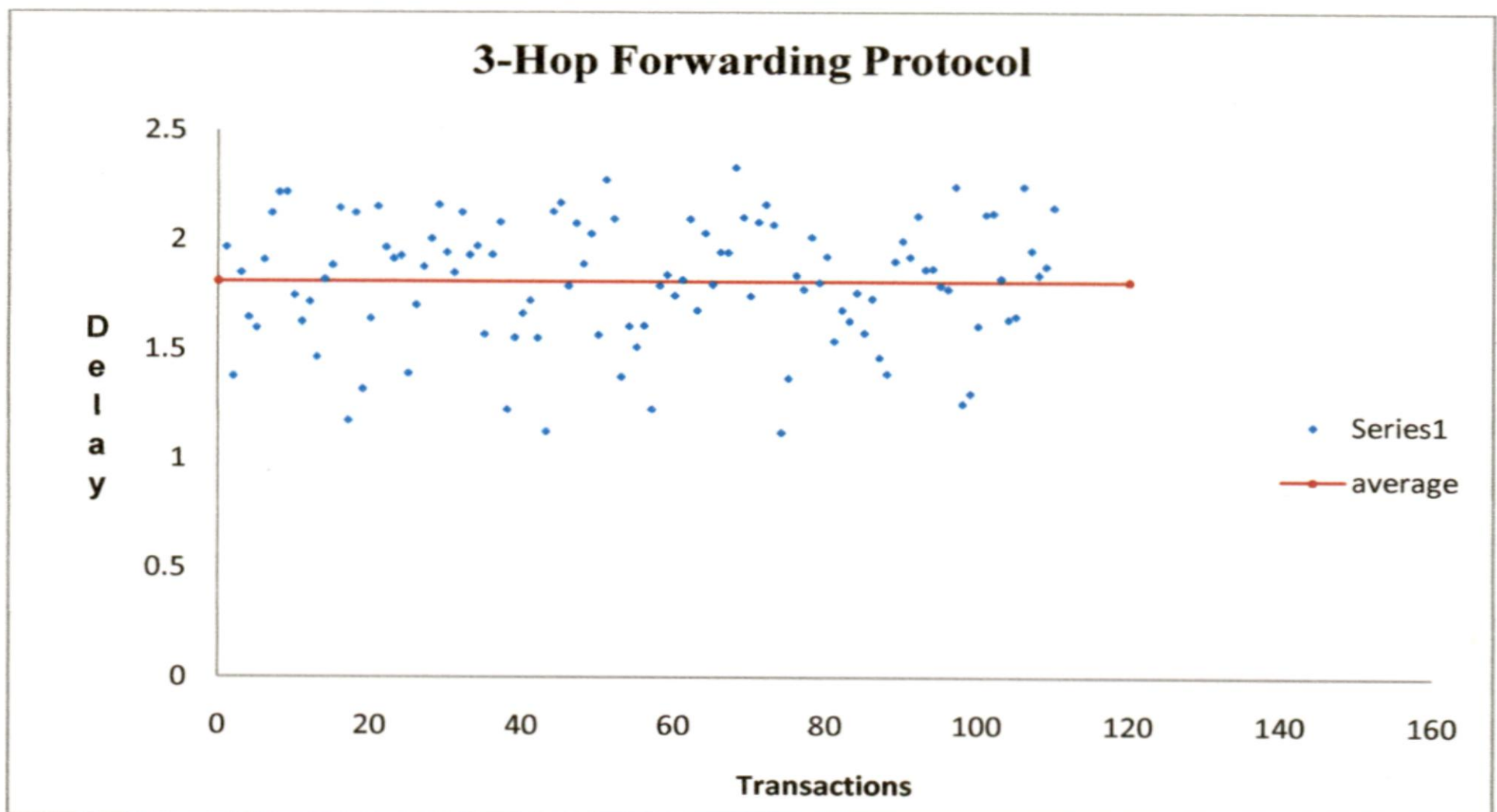
**Fig 6.2:** Degree of Anonymity of 2-hop protocol



**Fig 6.3:** Degree of Anonymity of 3-hop protocol



**Fig 6.4:** Delay in 2-hop protocol



**Fig 6.5:** Delay in 3-hop protocol

## 6.2 Comparison to 2-hop protocol

- By using 3- hop forwarding protocol we can eliminate intersection attacks and thus load imbalance due to selection of a constant forwarder.
- However the bandwidth usage of 3-hop forwarding protocol is high compared to 2-hop forwarding protocol.
- For small networks it is preferable to use 2-hop protocol than 3-hop protocol because the degree of anonymity to bandwidth ratio of latter is low compared to the former.

## 6.3 Selection of the Forwarder and Possible Attacks

From the analysis in the previous section, we observed that the size of the VO to which the initiator belongs w.r.t task  $T_k$  influences the degree of anonymity that can be achieved if the forwarder site is compromised. The selection of the same forwarder repeatedly for job request routing and data forwarding can lead to load imbalance and generation of hotspots. A random selection of the forwarder will not lead to an intersection attacks as in 2-hop forwarding protocol .In an intersection attack, if the adversary knows that the initiator is in two different sets  $A$  and  $B$ , then the anonymity of the initiator is reduced to  $A \cap B$ . Moreover, a site which is a member of several VOs has a very high probability of being selected as the forwarder. Thus the selection of the forwarder is a tradeoff between load balance and higher degree of anonymity.

## 6.4 Multihop protocol

Protocol may be extended to any number of hops. But for a grid configuration 3-hop protocol provides optimal degree of anonymity that can be achieved assuming that nodes are not compromised. So increasing hops beyond three greatly reduces efficiency due to high bandwidth consumptions without any considerable increase in degree of anonymity. Depending on size and anonymity requirements 3-hop protocol and 2- hop protocol can be used in conjunction for achieving better system efficiency.



We have studied a new anonymity approach for grid computing and designed the 3-hop anonymity protocol. We also studied about information theoretic approach to measure degree of anonymity. Based on the existing trust in grid application, this protocol is more efficient and stronger than the existing protocols. The degree of anonymity is quantitatively analyzed. Nevertheless, the study is limited to cases that trust relationship can simply be derived from virtual organization membership. With a more practical and complicated trust model, this protocol could be enhanced by considering varying trust levels, the possibility of attacks to trusted sites, and more trade-offs between anonymity and load balance.

We can also conclude that for a given grid configuration 3-hop protocol provides optimal degree of anonymity. Increasing number of hops beyond three will only add to more bandwidth consumption and delays.

In our protocol we didn't consider the trust existing between members of different Virtual Organizations. This can be studied and routing can be made more flexible instead of selecting at random.

## References

---

1. I. Foster and C. Kesselman, editors. "*Computational Grids: The Future of High Performance Distributed Computing*". Morgan Kaufmann, 1998.
2. J. Joseph, M. Ernest, C. Fellenstein, "Evolution of grid computing architecture and grid adoption models" IBM systems journal vol. 43, Number 4, 2004
3. Pfitzmann and M. Waidner. "*Networks without user observability*". *Computers & Security*, 6(2):158–166, Apr. 1987.
4. Nataraj Nagaratnam, Philippe Janson, John Dayka, Anthony Nadalin, Frank Siebenlist, Von Welch, Ian Foster, Steve Tuecke. "*The Security Architecture for Open Grid Services*". Global Grid Forum, July 2002.
5. "*Introduction to grid computing with globus*", [www.ibm.com/redbooks](http://www.ibm.com/redbooks).
6. I. Foster, C. Kesselman, and S. Tuecke. "*The anatomy of the Grid: Enabling scalable virtual organizations*". *International Journal of Supercomputer Applications*, 15, 2001.
7. Xiao. Yang. "*Security in distributed, grid, mobile, and pervasive computing*". Auerbach Publications 2007.
8. F. Azzedin and M. Maheswaran. "*Evolving and managing trust in grid computing systems*". In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2002.
9. V. Cahill et al. "*Using Trust for Secure Collaboration in Uncertain Environment*". *IEEE Pervasive Computing*, July-September 2003.

10. Huu Tran, Paul Watters, Michael Hitchens, Vijay Varadharajan, “*Trust and Authorization in the Grid: A Recommendation Model*” IEEE International Conference on Pervasive Services 2005.
11. A. Chakrabarti. “*Grid computing security*”. Springer, 2007.
12. Alvaro Arenas, Michael Wilson, Brian Matthews. “*On trust management in grids*”, Proceedings of the 1st international conference on Autonomic computing and communication systems, ICST, 2007
13. M. K. Reiter and A. D. Rubin. “*Crowds: Anonymity for Web Transactions*”, ACM Transactions on Information and System Security, v. 1, 66-92, 1998.
14. The Anonymizer, <http://www.anonymizer.com/>.
15. D. Chaum, “*Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*”, CACM, v. 24, 1981.
16. C. Shields and B. N. Levine, “*A Protocol for Anonymous Communication Over the Internet*”, Proceedings of the 7th ACM Conference on Computer and Communication Security, Nov. 1-4, 2000.
17. Souvik Ray ,Zhao Zhang, “*An Efficient Anonymity Protocol for Grid Computing*”, Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing ,GRID 2004.
18. J.F. Raymond. “Traffic analysis: Protocols, attacks, design issues and open problems”. In H.Federrath, editor, Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability, volume 2009 of LNCS, pages 10–29. 2001.
19. R. Sherwood, B. Bhattacharjee, and A. Srinivasan. “*P5: A protocol for scalable anonymous communication*”. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002.

20. Claudia D'iaz, Stefaan Seys, Joris Claessens, and Bart Preneel. "*Towards measuring anonymity*". In *Designing Privacy Enhancing Technologies, Proceedings of PET'02*, pages 54–68. Springer-Verlag, 2003.
21. Andreas Pfitzmann and Marit Hansen. "*Anonymity, unobservability, and pseudonymity: A proposal for terminology*". Draft, v0.21, September 2004.
22. Claude Shannon. "*A mathematical theory of communication*". *The Bell System Technical Journal*, 1948.
23. Andrei Serjantov and George Danezis. "*Towards an information theoretic metric for anonymity*". In *Designing Privacy Enhancing Technologies, Proceedings of PET'02*, pages 41–53. Springer-Verlag, 2002.