Towards a Secure Email Service for The Future Internet

Muhammad Shoaib Saleem, Eric Renault

Towards a Secure Email Service for The Future Internet

Muhammad Shoaib Saleem and Éric Renault
Institut Télécom - Télécom SudParis, Évry, France
Email: {shoaib.saleem, eric.renault}@it-sudparis.eu

***Abstract*—Email is one of the most used application over the Internet and it is vulnerable to cyber attacks or cyberwarfare on a fairly quotidian basis. Spam emails are another major nuisance for email users. The proposed email service presented in this article is based on the Future Internet architecture named Network of Information (NetInf). NetInf is an Information-Centric Network developed to address issues like ubiquitous content distribution, user privacy and security, naming and addressing, routing and name resolution with a clean-slate approach. The working scenario of the NetInf email service explains how all these features are inherited and utilized. The qualitative evaluation shows that the proposed service is secure, efficient and scalable.**

***Keywords*-NetInf Mail Service, Future Internet, Asymmetric Cryptography, Information Object, Bit Level Object.**

I. Introduction

The notion behind developing the Future Internet architecture is to improve the information availability at a global scale. There have been lots of efforts to develop the whole Future Internet architecture by redefining the basic building blocks of the Internet architecture. The Network of information [1] is the part of the Future Internet architectures where an information is considered as the prime unit of the network, not nodes. The TCP/IP-based Internet is a network of networks of interconnected nodes. However, the main interest of end users is information, not their location (i.e. the server or the IP address where it is stored). In this Information-Centric Networking paradigm, a piece of information is elevated as the primary candidate in the network, superseding the node-centric network approach. This idea is the cornerstone in the design and architecture of the Network of Information.

Users on the Internet always desire to experience the best Quality of Service (QoS). However, quality of experience (QoE) defines the parameters for designing (new services) or redesigning (existing services). Email is one of the widely used services on the Internet. The basic function of this service is the transfer of the text messages. Dedicated ports, servers and protocols are involved in this service which makes its installation and maintenance quite difficult. Spam email, privacy and security of users' emails and email ID are the main issues still addressed in the research community.

This paper presents a new email service based on the specificities of the Future Internet. It is based on the NetInf architecture and uses the services NetInf provides in terms of routing, name resolution, storing/retrieving information and content distribution. The use of asymmetric key cryptography as a user ID is a unique feature introduced in the NetInf architecture. This ID, with no domain name attached, makes this server-less service scalable, secure and reliable.

The major difference between the NetInf email service and the current one is its independence from dedicated servers, ports and protocols. Since each entity in NetInf is considered as an object with a unique ID, each email is considered as a separate object secured through asymmetric-key cryptography. There are some useful ideas which can help in reducing the management and an overall increase in the overhead resulted by using cryptography keys like the one presented in [2]. This new email service is first of its kind for the Future Internet.

The paper is organized as follows: Section II presents the motivation for this work. Section III gives a comprehensive overview about the design and the architecture of NetInf. Section IV explains the NetInf email message format followed by a discussion over the working scenario of the proposed service. The evaluation of this work is done in Section V while the concluding remarks and the future work are discussed in Section VI.

II. Motivation

This section addresses issues and problems related to contemporary email services. The following example depicted in Fig. 1 briefly explains the working principle of an email services used today.

A. Email Message Sending (A@X $\xrightarrow[Emails]{Message(M)}$ B@Y)

The simple scenario explained here involves the steps required to send an email message M from user $A@X$ (where A is the username or identifier and X is the domain name) to user $B@Y$. Variations exist in the message format and the steps involved depending upon the email content, attachments (if any) and the security (if a security protocol is used). [7] and [8] explain the complete format of an email message which is beyond the scope of this paper. Fig. 1 shows how an email is send from user $A@X$ to user $B@Y$ in the current Internet system. The steps followed are:-

- User $A@X$ writes message M to user $B@Y$ using its Mail User Agent (MUA).
- The MUA of A sends M using a submission protocol to the Mail Submission Agent (MSA).
- The MSA resolves domain name Y of user B through DNS and receives an ID from the Mail Transfer Agent (MTA) for B of Y domain.

- MSA sends the message to MTA of B.
- MTA forwards the message to Mail Delivery Agent (MDA) which further relays the message M to user B inbox.
- User $B@Y$ retrieves message M using its MUA.

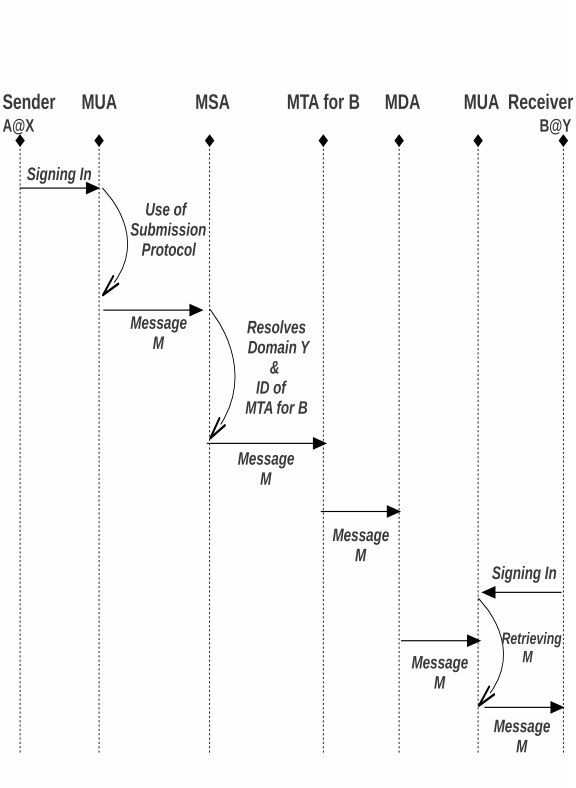

Fig. 1. **Existing Email Message Sending**

The email services today follow more or less the same procedure for sending and retrieving emails as discussed in the example above. Still there are issues that degrade the overall performance of this whole system. For example, issues like security and privacy related to the email content and it's user's credentials, email reputation systems for spam-email filtering as well as email system latency in case if the installed data centers of email system are centralized. This can also cause the service unavailability problem in case of system failure. These issues have been addressed by the research community over the years and a brief overview is discussed in this section.

As mentioned before, security is one of the major concerns in email service. In [2], public key certificate and credential trust management problems are targeted. The proxy based security scheme is an interesting solution as it reduces the cost of public key management. However, self security of the proxy service is required to handle any failure. Similarly, based on XML technology, [9] presents an advance email security mechanism. Its flexibility and its portability through web service gives it an edge over the existing system. The service is known as XML email which is based on the previous approaches plus novel features making this approach efficient with more security properties. The system can be developed to cater for spam e-mails also.

[10] is one step towards context based classification for email filtering. The survey conducted between different statistical approaches for email filtering is summarized along with their performances. The current deployment of e-mail filtering on web is mostly based on statistical approaches on text classification. However, the context based approach is new and there is a wide scope to work on it in future. Using context based text classification can further improve the spam filtering. On the other hand, an email reputation system evaluates the performance of the e-mail services. CARE [11] is one such system which is based on the inter-domain collaboration of different email services. The performances of these services are evaluated by rating both spam and anti-spam emails filtering. Results obtained through extensive simulation proves its effectiveness.

The above discussion concludes that the problems faced by email service can be mitigated by implementing various solutions proposed by the research community. However, the point of argument which motivates to propose the NetInf email service is to incorporate all these solutions in one system. The NetInf architecture has all the features required by an email service as discussed above. The next section explains the NetInf architecture and its features.

III. Design and Architecture of Network of Information

The Information-Centric Networking is a concept where end users are only concerned with the data they are accessing. In this paradigm, the physical location of the information is irrelevant to the users. The Network of Information (NetInf) has been developed in the scope of the 4WARD project [3] (A European project for Future Internet architecture). The contemporary Internet architecture is composed of nodes interconnected by the TCP/IP protocol suite. NetInf addresses both wired and wireless communication issues that involve the Internet by proposing a major overhaul in the Internet architecture.

The use of mobile Internet has increased overwhelmingly because of the availability of Smart phones and the applications that are being used. This unprecedented increase of Internet usage over mobile phone has degraded the performance of different services in terms of overall QoS, privacy and security. Available solutions mitigate these problems temporarily because all the provided services are compatible with the current TCP/IP based Internet architecture. In NetInf, the hierarchical flat structure for the name resolution based on DHT makes content caching and data transfer faster than the current tree shaped Internet architecture. The other challenges addressed by NetInf have focused on the reduction of the overall traffic congestion, cache management (e.g., updating the version of the cache if it is changing frequently) as well as scalability of the architecture in the case of a global deployment.

Information on the Internet is not limited to simple text messages anymore. The complexity has become more visible when the four-decade old client-server model fails to exploit different attributes related to an information. So far, this model has worked brilliantly. However, the task of developing a framework where different network technologies can be integrated becomes cumbersome under a network with a host centric approach. In an Information-Centric Network (like NetInf), all objects, including services, have a unique identifier (or a name). Accessing these services through these identifiers make the system independent from particular applications required to access them.

NetInf considers every object, either virtual (e.g. text files, songs, videos, etc.) or physical (e.g. monuments, buildings, man, machine, etc.) as an information, and defines them as an Information Object (IO). The object model of NetInf represents the hierarchy of different types of objects. The higher level of the hierarchy consists in a set of IOs which are further classified into Semantic Information Objects (IO_s) (providing semantic details of an IO from the user perspective) and Management Information Objects (IO_m) (describing an IO from the NetInf point of view). The lower level of the hierarchy of the object model represents the information in terms of streams of bits and are defined as Bit-level Object (BO).

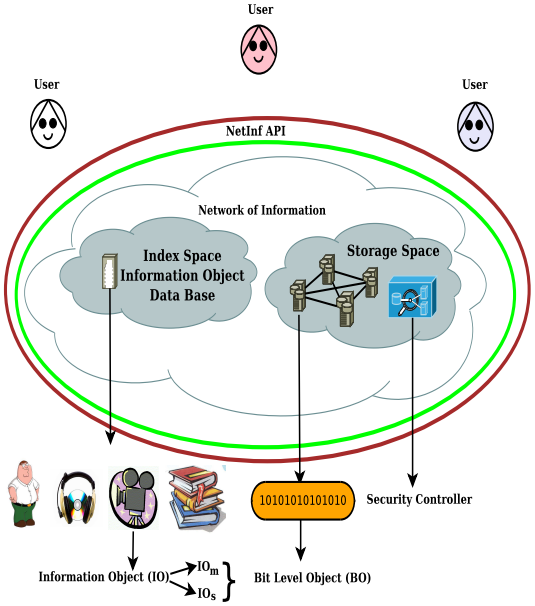

Fig. 2. **Network of Information Mail Service**

The Metalist model in [4] defines IO management for NetInf. This model serves different aspects by representing the metadata related to BO. For example, tagging objects with attributes makes the semantic search easier for the end user. However, in order to reduce the metadata redundancy, existing metalist can be used in this model. The indirection method is used in cases where a metalist with a different format can be included as an external metadata in the NetInf Metalist model. Information in NetInf are managed by users and NetInf mutually. Users publish IOs with *Access Rights* (i.e. right to read, delete or modify the content of an information) defined and attached to it while NetInf manages them through information dissemination within the network.

Take an example for searching an information through NetInf architecture. A user connected to NetInf will look for the relevant information by feeding semantic information through an interface. The NetInf system will retrieve the result in the form of metadata tagged with the semantic information provided by the end user. Using this information, the required information can be accessed in NetInf. Here, the provided semantic information can be considered as IO_s. The metadata provided by the NetInf system is analogous to IO_m and the retrieved information is BO.

Since privacy and security of information in NetInf is one of the prime objectives, the scenario for information retrieval is not as simple as explained above. Within the NetInf architecture, a Security Controller [5] facilitates any kind of secure information transfer as shown in Fig. 2. The owner of an information in NetInf has the right to declare it either as a public or protected entity. Information notified as a public entity is accessible by any user. However, the *Access Rights* define what rights they have. Generally, read only right is granted. On the other hand, an information declared protected by its owner, the Security Controller challenges the users, accessing it, to undergo verification tests. This is done by the consent of the user who owns this protected information and distributes its public key to a limited number of users. Whenever, one such user accesses NetInf, it undergoes through the Security Controller challenge by providing public key granted by the information owner. In NetInf architecture, the Security Controller is located in the storage space (discussed below).

The NetInf architecture has the following distinct components as shown in Fig. 2:

- **Storage Space** is the part of NetInf where IOs and BOs are stored. The Security Controller [5] in the Storage Space takes care of access to both kinds of data objects. The Storage space can be considered as a set of data centers. Information once published in the NetInf system is managed by NetInf and may be replicated in different data centers. In NetInf, information are cached in data centers that come along the routing path adopted for delivery. This helps in data dissemination and reduction of the overall latency.
- **Index Space** indexes the IO_s. Semantic details of BO are pushed in the Index space by the users. The Metalist model is implemented in the Index space of NetInf. However, the semantic details can be indexed using other mechanisms e.g. RDF framework. This defines the flexibility of NetInf architecture to quickly assimilate any change and to implement and apply it immediately. In NetInf, the XML format is used for tagging metadata. Hence, the Index space maintains an XML based database of IO.

- **NetInf API** was developed for NetInf and is based on the REST architectural style [6]. It is the only part through which users interact with NetInf. It provides basic functions to publish, retrieve, delete (and other functions) IO using $Push$,Put, Get, $Publish$ and $Remove$ operational commands. As an example, in NetInf, command $Push$ is used to store BOs in the storage part. Once BOs are stored, the IDs generated by the Storage Space results in the creation of IO_m by NetInf which manages all these IDs. Upon storing this IO_m in the Storage Space, the generated ID is forwarded to the user. Similarly, while using the command Get, the same ID is used to retrieve IO_m. The contents of IO_m represents the IDs for BOs stored in the Storage Space. Using these IDs, the required information is extracted from the Storage Space.

IV. Network of Information Email Service

A. NetInf Email Message Format

In the NetInf email service, an email message is composed of the content part unlike to the current email message format [7]. The header and envelope terms describe the part of the content which carries the information that include sender/receiver identity, date and time at which the message was sent and timestamp. The message body contains the message written by the sender of the email.

Fig. 3 shows an email message format. The header and the message parts are considered as objects in NetInf. Each object has a unique ID, OID_H and OID_M for the header and the message parts respectively. *Access Rights* are attached to both objects.

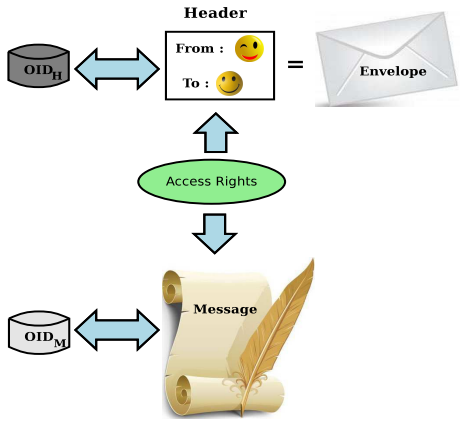

Fig. 3. **NetInf Email Message Format**

B. NetInf Email Working Scenario

The scenario explains the end-to-end user procedures involved in sending and receiving emails in NetInf. In summary, the procedure indulges all the three components of NetInf (explained in Section II) to be the part in the process of email exchange. From an end user to the storage space of NetInf, the data flow is ensured to be secured by the grace of asymmetric key cryptography. The NetInf interface provides simple commands like *Push*, *Put* and *Publish* for information insertion within the system. The retrieval involves a phase in which the user is challenged to prove his authenticity using his unique private key. The procedures, both at the sender and the receiver ends, are two phased processes. The list of notations used during these processes is shown in Table I. The details of each phase are as follow:

TABLE I
List of Notations

M	Email Message
$\bar{M}$	Encrypted Email Message
H	Message Header
$\bar{H}$	Encrypted Message Header
PK	Asymmetric Public Key
$\bar{PK}$	Asymmetric Private Key
IO_S	Information Object(s)
IO_m	Management Information Object
IO_m^M	Management Information Object for Message M
IO_s	Information Object with Semantic Details
IO_m^H	Management Information Object for Message Header H
$AR\|_D^R$	Access Rights for Read and Delete
ID	Storage Space Generated Identifier
OID_H	Header Object Identifier
OID_M	Message Object Identifier

1) @Sender End: In Fig. 4, the sending function format of an email is represented as $send(PK, H, M)$.

Phase #I

In Phase #I, the message part ***(M)*** of ***send(PK,H,M)*** is processed.

(a) The email message consists of three fields: (i) The header ***(H)***, holds the details mentioned earlier. (ii) The receiver's Public Key ***(PK)*** and (iii) The content of the message ***(M)***.

(b) The Mail User Agent (MUA) encrypts, the message ***(M)*** with the receiver's public key ***(PK)*** and the *Access Rights* (Read, Delete) ***(AR $|_D^R$)*** are attached with it. The encrypted message ***($\bar{M}$)*** along with *Access Rights* ***(AR $|_D^R$)*** are pushed into NetInf through the NetInf API.

$$\textbf{\textit{Push}}\ (\bar{M} = M \oplus PK\ ,\ AR\ |_D^R)$$

(c) Within the NetInf API, ***($\bar{M}$)*** is stored using the ***Put*** function. Each new object is acknowledged by NetInf which sends back a unique identifier ***(ID)*** generated by the Storage Space. This results in creating an IO_m^M managing the generated ***ID*** for ***($\bar{M}$)***.

(d) With the ***Put*** IO_m^M operation within the Storage Space, NetInf returns IO_m^M ***ID***

Phase #II

In Phase #II, the Header ***(H)*** part of ***send(PK,H,M)*** is treated by NetInf.

(a) The header ***(H)*** is concatenated with $IO_m^M ID$ and is encrypted using ***PK*** as follows:

$$\bar{H} = (H\ ||\ IO_m^M\ ID) \oplus PK$$

(b) The encrypted Header ***($\bar{H}$)*** along with ***AR*** $|_D^R$ are published in the NetInf. NetInf stores the encrypted header ***($\bar{H}$)*** in

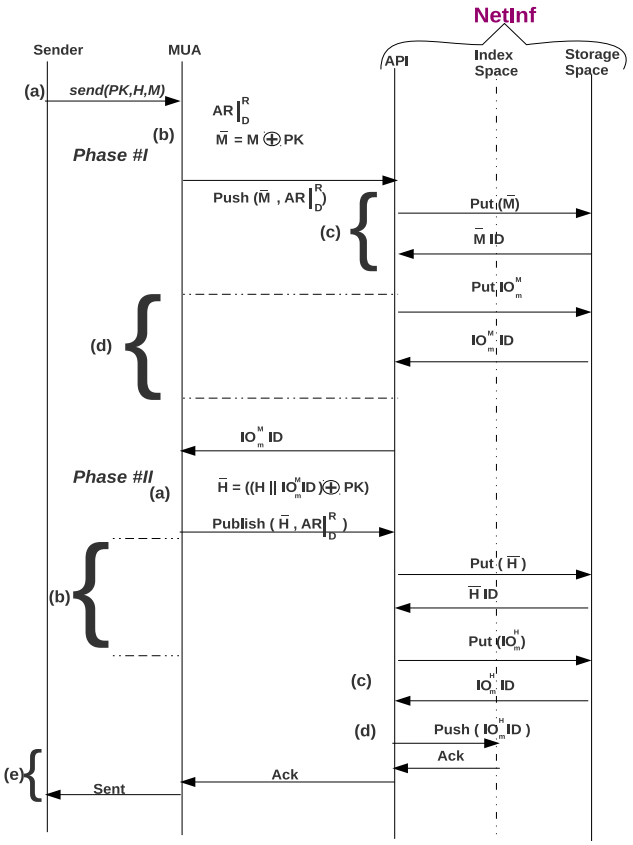

Fig. 4. **NetInf Email Sending Procedure**

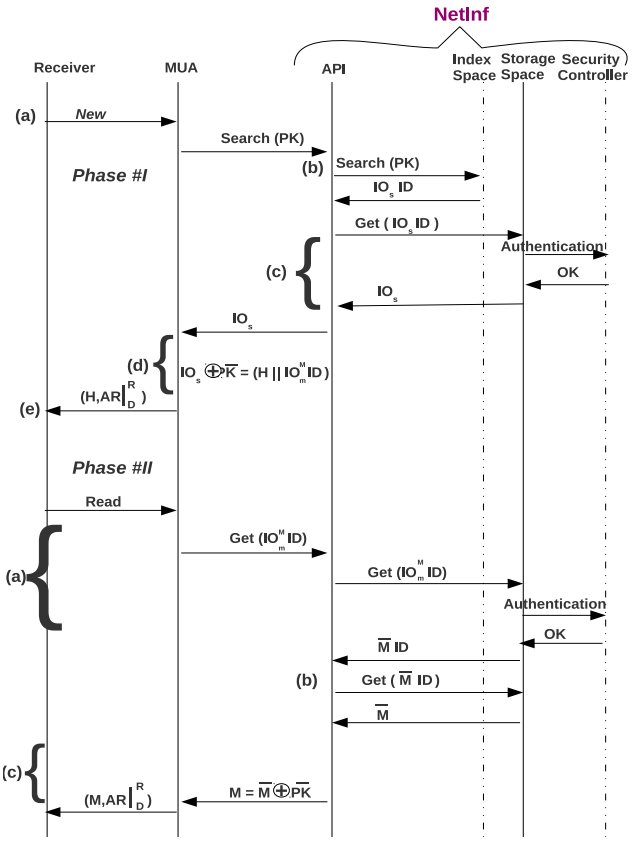

Fig. 5. **NetInf Email Receiving Procedure**

the Storage Space. For the management of the generated ***ID***, NetInf creates $\boldsymbol{IO}_m^H$.

(c) ***Put*** $\boldsymbol{(IO_m^H)}$ operation returns the ***ID*** for $\boldsymbol{IO}_m^H$.

(d) ***Push*** $\boldsymbol{(IO_m^H ID)}$ is the step where the whole email with all the contents is published in the Index space.

(e) In the end, an acknowledgement is received by the user at the sender end that the message has been sent successfully.

2) @Receiver End: The receiver end user performs a request to identify the list of messages he has received, i.e. the list of objects that have been encrypted with his ***PK***. At the receiver end, an email message is retrieved in two phases as shown in Fig. 5. The first phase recovers the metalist of Semantic Information Object $\boldsymbol{(IO_s)}$ with their ***(IDs)***. The $\boldsymbol{(IO_s)}$ is equivalent to $\boldsymbol{(IO_m^H)}$ as it recovers the ***(H)*** of the email. The second phase allows to extract the message from the Header ***(H)*** depending upon the *Access Rights* ***(AR)***.

Phase #I

(a) The process is initiated when the user at the receiver end uses its MUA to fetch new messages. The MUA initializes the search for the emails addressed with the receiver's public key ***(PK)*** using the ***search (PK)*** function.

(b) NetInf relays this request on behalf of the MUA to the Index Space. The request is replied with the list of metalist of $\boldsymbol{(IO_s)}$ with the identifiers $\boldsymbol{(IO_s ID)}$.

(c) Using the $\boldsymbol{Get(IO_s ID)}$ command, the Storage Space is accessed to retrieve a specific message. The Security Controller challenges this access by testing users' credibility. If the user proves to be the actual owner of the public key ***(PK)*** by providing his $\boldsymbol{(\bar{PK})}$, the Security Controller grants access to the Storage Space. Otherwise, the access is denied.

(d) Using private key $\boldsymbol{\bar{PK}}$, the user extracts $\boldsymbol{H \,||IO_m^M ID}$

$$\boldsymbol{IO_s \bigoplus \bar{PK} \rightarrow H \,||IO_m^M ID}$$

(e) Once decrypted, the user can access the Header ***(H)*** contents.

Phase #II

(a) In order to read the contents of the message ***M***, user's query of ***Read*** invokes the MUA to extract the bit level object from the Storage Space. The $\boldsymbol{Get(IO_m^M\ ID)}$ command makes this happen by accessing the Storage Space. Here again, the Security Controller verifies user's authentication and returns $\boldsymbol{(\bar{M}\ ID)}$.

(b) With $\boldsymbol{Get(\bar{M}\ ID)}$ operation, encrypted $\boldsymbol{(\bar{M})}$ is extracted.

(c) Finally, the encrypted message $\boldsymbol{\bar{M}}$ becomes readable message ***M*** when decrypted using receiver's private key $\boldsymbol{\bar{PK}}$.

$$\boldsymbol{\bar{M} \bigoplus \bar{PK} \rightarrow M}$$

V. Qualitative Evaluation

For the evaluation, we collected a sample of $3 * 10^5$ emails which were received between the years 2006 and 2010. The size of emails varies, ranging from 100 bytes to 50 MB. Fig. 6 presents the distribution of the number of emails together with their cumulative distribution. This clearly shows that the size of emails follows the Gaussian law centered around 3KB where 90% of emails have size smaller than or equal to 30KB. It means that most of the emails have small size on average i.e. they occupy small memory space. This qualitative analysis is an estimation to the possible outcome that will effect the size of the emails after encrypting emails before sending. We estimate that since most of the emails being exchanged on the Internet have, on average, small size; the performance of an email service in terms of latency will not be much affected. The performance criteria selected in this work is primarily privacy and security of users' identity and email content. The possible increase in the overall overhead that will be added after encrypting emails, by using the asymmetric keys, would be minimal.

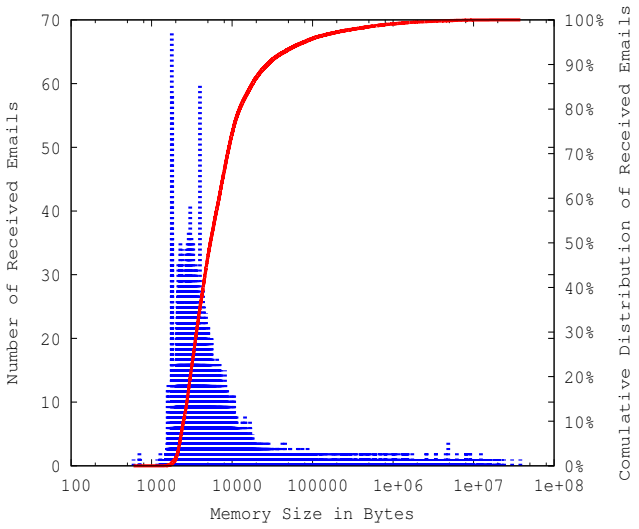

Fig. 6. **Number of Received Emails with Different Memory Sizes**

However, still in the Fig. 6, there are number of emails that require large memory space. This increase in the size can result into overall system latency while sending or fetching an email. NetInf architecture addresses this problem with better content distribution mechanism. Therefore, we envisage a trade-off between these two factors i.e. system latency (which will be minimal) and security and privacy of the content.

VI. Conclusion and Future Work

This paper presented an innovative NetInf-based email service. The objective of this work is to introduce a novel email service. The unique features of this new service supports the idea of having a network with no dedicated ports and servers. The approach used is based on the Future Internet architecture known as NetInf. This email service works on the top of NetInf architecture. The privacy and security of the email content is ensured by using asymmetric keys. This idea is not new for securing information data but the the use of public/private keys as a user ID is a new concept in the context of NetInf. The email format defines the content of the message. The content or rather IO in the context of NetInf have unique IDs along with defined *Access Rights*. The scenario of email exchange presented makes easy to comprehend the mechanism of the proposed service. The evaluation section discussed the analysis of the email data used and is linked with the system performance in terms of data size and the latency experienced while retrieving this data.

A clean slate approach for a new Internet architecture, like NetInf, is clearly an alternative. The proposed email service presented in this work is in the context of the NetInf. A service that is secure, reliable and where an information is created, stored and retrieved without requiring dedicated infrastructure. The basic NetInf prototype has already been implemented. Our ongoing work includes the development of an interface for the email service. The performance evaluation of the service will have to include various tests with different parameters measuring latency, reducing the overall overhead due to the use of asymmetric key cryptography and reputation tests to take care of Spam emails. Most of Spam emails are related to companies for advertisement purposes. One solution is to make the such senders pay the subscription fee to use email service for their product advertisement. This will possibly reduce the bulk of Spam emails which consume a big part of users mailbox Storage Space.

References

[1] Bengt Ahlgren, Matteo D'Ambrosio, Marco Marchisio, Ian Marsh, Christian Dannewitz, Börje Ohlman, Kostas Pentikousis, Ove Strandberg, René Rembarz and Vinicio Vercellone, *"Design Considerations for a Network of Information"*, Proceedings of the 2008 ACM CoNEXT Conference,CoNEXT '08, Madrid, Spain, 2008.

[2] Chen Tieming and Ma Shilong,*"A Secure Email Encryption Proxy Based on Identity-Based Cryptography"*, International MultiMedia and Information Technology, MMIT '08,Three Gorges, China,2008

[3] *http://www.4ward-project.eu*

[4] E.Renault and Djamal Zeghlache,*"The Metalist Model: A Simple and Extensible Information Model for the Future Internet"*, 15th Open European Summer School and IFIP TC6.6. Workshop on The Internet of the Future,Barcelona, Spain 2009

[5] E.Renault, A.Ahmad and M.Abid,*"Toward a Security Model for the Future Network of Information"*, 4th International Conference on Ubiquitous Information Technologies Applications, ICUT '09,Fukuoka, Japan,2009

[6] Roy T. Fielding, Richard Taylor and N Richard,*"Principled design of the modern Web architecture"*, Proceedings of the 22nd International Conference on Software Engineering,Limerick, Ireland, 2000

[7] P. Resnick, Ed.*"Internet Message Format"*, RFC: 5322,October 2008

[8] J. Klensin,*"Simple Mail Transfer Protocol"*, RFC: 5321,October 2008

[9] Lijun Liao and Jorg Schwenk,*"Secure Emails in XML Format Using Web Services"*, ECOWS '07. Fifth European Conference on Web Services,2007.

[10] Upasana and S.Chakravarty,*"A Survey on Text Classification Techniques for E-mail Filtering"*, Second International Conference on Machine Learning and Computing (ICMLC),2010

[11] Mengjun Xie and Haining Wang,*"CARE:A Collaboration-based Autonomous Reputation System for Email Services"*, INFOCOM, 2010.