International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)

International Conference on Information Systems and Computing (ICISC-2013), INDIA.

DETECTION OF E-MAIL SENDERS WITH SMTP EXTENSION

S.Vikash Koushik[1], G.Aparna[2], B.Thenkalvi[3]

[1,2] UG Student, Department of Computer Science and Engineering, Sri Muthukumaran Institute of Technology, Chennai, India;
[3] Assistant Professor, Department of Computer Science and Engineering, Sri Muthukumaran Institute of Technology, Chennai, India.

Email: [1] vikashkoushik@ieee.org, [2] aparnaganesh@ieee.org, [3] bthenkalvi@ieee.org

Abstract

E-mail spam is one of the most common problems that we face in our day to day life. There exists massive number of spams sent from justifiable home computers that comprises of a malware called as bot. The current scheme to identify mail senders called as Domain Keys Identified Mail (DKIM) which cannot identify such spamming bots. They are capable of only identifying email domains and not email addresses of the senders. A delicately complex problem when using no DNS is where to locate our trusted core, rather than the DNS. To cope up with this problem, we are proposing a design to detect the email addresses of senders that do not use DNS, so we embed our scheme into the Simple Mail Transfer Protocol (SMTP).

***Keywords*-- DNS, trusted core, Spam, bot, SMTP**

I. INTRODUCTION

Spam mails are one of the common social problems that almost everyone faces in our day to day life. The information collected by MessageLabs indicated that the spam rate is over 70 percent and persistently remains high [3]. There are two natures of compromised machines on the Internet – sheer volume and widespread – that render many existing security counter measures less effective and defending attacks involving compromised machines extremely hard [4]. Spam usually spoofs its address, username @ domain name of origin. In some cases, the address is source of both legitimate email as well as spam due to botnets which are controlled by bot masters. The botnets are commonly used to launch a very large amount of spam mails or messages [1]. In our scheme, an email service provider to which the users subscribe the email, register for a one time secret key at the trusted core which is host trusted by and local to the email service provider. We focus our attention to schemes for authorizing and identifying email senders such as Domain Key Identified mail (DKIM) and Sender Policy Framework (SPF) [7] that identify the senders based on their digital signatures and IP addresses of the sender's email respectively.

II. EXISTING SYSTEM

One of the dominant techniques for protecting against the email address spoofing is to authorize legitimate email domains to send email [8] [9]. In DKIM, the administrator registers the records to authorize the domain name and also prepares a server to save the domain's public key.

This would ensure that the sender's domain name is digitally signed and sent along with an email header. On the other end, the receiving side gets the public key through the DNS and verifies the digital signature. Even though there are no methods of spam detection in DKIM, the sending domain will most likely be genuine if the decrypted domain name is indistinguishable to the name in the FROM header that is obtained in plain text.

The DKIM scheme is not capable of identifying the spam bots since it identifies the email domains, and not the email addresses of the senders. In addition to this, the DNS has few security issues when used as a trusted core, as explained below:

First, the DNS cannot overcome the pharming attack that can exploit the major drawbacks in the DNS to return the IP address of the pharmer's server, instead of the legitimate IP address of a queried domain name. The DNS security extension (DNSSEC) [10] can protect from pharming as it ensures data origin through a verification chain from the root to the resolver.

Secondly, a system for authenticating a web server needs a trusted core such as Certified Authority (CA) or a scheme to collect the reputation of it, in its related communities, which may be considered as a distributed online CA. In this case, the DNS is neither a CA nor a reputed scheme. Hence, allowing the spammers to easily register themselves to the DNS. With the Simple Mail Transfer Protocol service extension for substantiation [12], the client can indicate an substantiation service for the server.

The fidelity of SMTP servers can be obtained through trusted administration with no centralized trusted cores [2].

To assuage this problematic situation, we propose a system for authorizing, detecting and identifying the email addresses of the senders which is easy to use, especially for home users and doesn't rely on the DNS. We embed our system into the Simple Mail Transfer Protocol (SMTP) [11] to detect the email address of the sender.

III. Proposed System

3.1 Overview of Proposed Framework:

An architectural diagram of our proposed scheme is shown in Fig. 1. The SMTP is the basic protocol that is used for the email transfer. The process of sending and receiving are referred to as client and server respectively. The email transfer services provided by client and server are also known as Mail Transfer Agent (MTA). Similarly, the process of source and target of email are called Mail User Agents (MUA).

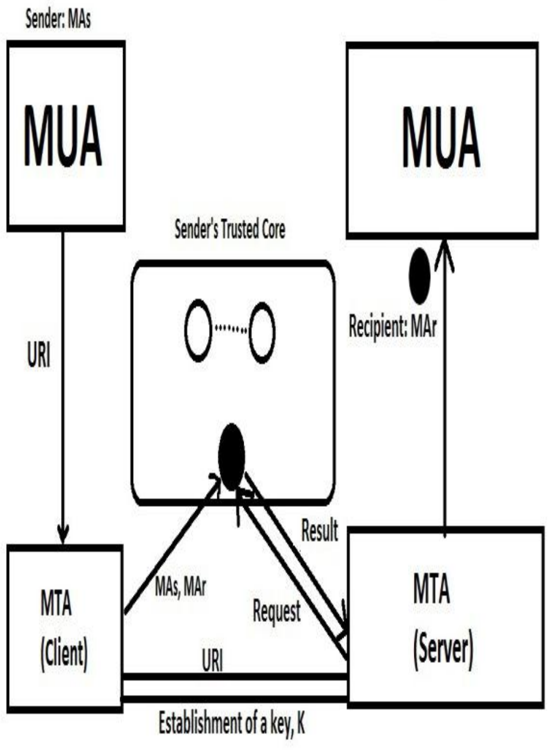

Fig 1. Proposed Framework

In the event of authorizing, the authorizer first registers the user's mail id MA_s at a number of trusted cores and stores the Uniform Resource Identifier (URI) of the trusted core into the user's computer. To recognize the sender's mail id MA_s, the server and the client need to set up a key K which is used as an encryption key. Following this process, the client need to give the sender's mail and receiver's mail id (MA_s, MA_r) to a trusted core which is chosen on random by the sender MUA and then sends the URI of the trusted core service to the server. The deposit will succeed if the trusted core has the same registered MA_s as received MA_s. Finally, the trusted core will receive a recognition request with (FROM, MA_r) from the server and replies the result with a success if FROM = MA_s.

The proposed system is sub-divided into three phases: phase 1 – Sender's authorization, phase 2 – Detection of E-Mail addresses of the senders and phase 3 – Embedding Shamir's Protocol in SMTP which are discussed below.

3.2 Sender's Authorization:

The Pretty Good Policy follows [6] the policy that the trust worthiness of a person is established through reputation in the communities to which they belong to. In DKIM, the administrator authorizes its employees. Regrettably, how to authorize the persons on an individual home computer are not mentioned in the DKIM [2].

In our scheme, an Employer Supported Policing (ESP) is used to authorize the home users when they subscribe to it, while the administrator in the organization authorizes its employees. The administrator prepares a number of trusted computers in the organization that are used as trusted cores, each for momentarily depositing one time secret key to ensure the trust worthiness of the sender's email address. In our paper, we denote a set of objects by {objects}, an object encrypted with key X using symmetric encryption system like AES [13] by $\{object\}_X$.

For the approval of a home user, Fig. 2(a), in reply to the subscription request from the user with their mail address MA, the ESP returns a set of n pairs $(\{(URI_i, D_i)\})P$ encrypted with key P which is then sent to the user through a secure channel such as HTTPS and the ESP also saves the (MA , D_i) in the trusted $core_i$.

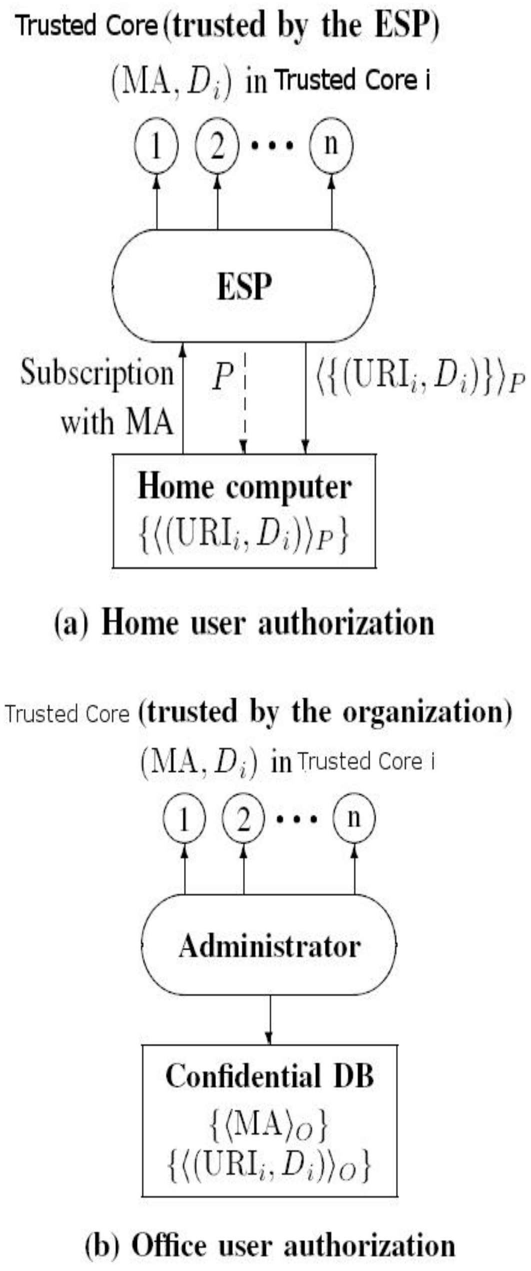
Trusted Core (trusted by the ESP)
(MA, D_i) in Trusted Core i
1
2
n
ESP
Subscription with MA
P
$\langle\{(\mathrm{URI}_i, D_i)\}\rangle_P$
Home computer
$\{\langle(\mathrm{URI}_i, D_i)\rangle_P\}$
(a) Home user authorization
Trusted Core (trusted by the organization)
(MA, D_i) in Trusted Core i
1
2
n
Administrator
Confidential DB
$\{\langle\mathrm{MA}\rangle_O\}$
$\{\langle(\mathrm{URI}_i, D_i)\rangle_O\}$
(b) Office user authorization

Fig 2. Sender Authorization for the users at home and office

On the contrary, to authorize the employees in an organization, (Fig. 2 (b)) the administrator will have to register their MAs and pairs {((URI$_i$, D$_i$))O} in a secret shared database of the organization since a particular employee may work on several office computers at different time. The administrator also stores (MA, D$_i$) in the trusted core$_i$.

When the home users sends a mail, its MUA aimlessly selects a pair (URI$_i$, D$_i$) from the saved pairs to specify the trusted core for the email, and sends the chosen pair to the SMTP client of the ESP. For office use, an SMTP client that deals with the outgoing mails, will verify the user whether he is registered in the database, and if so, it will arbitrarily retrieve a pair (URI$_i$, D$_i$) from the database.

3.2.1 Key Generation using Pipelined version of Shamir's Protocol:

Our identification protocol uses a pipelined version of Shamir's no key establishment protocol to establish a secret key [5]. With the inventive Shamir's protocol, the client and the server, exchange 3 messages over a public feed in order to establish a secret key K shared by them, where p is a large prime such that the computation of discrete logarithm modulus p is infeasible and x is a co-prime to y when their greatest common divisor equals to 1.

K1: The server would choose a random *K* $(1 <= K <= p-1)$ and a random a $(1 <= a <= p-2)$ that is a co-prime to $p-1$, and sends $K^a \bmod p$ to the client.
K2: The client chooses a *random* b $(1 <= b <= p-2)$ that is a co-prime to $p-1$, and sends $(K^a)^b \bmod p$ to the server.
K3: The server calculates $a^{-1} \bmod p-1$, and sends $(K^{ab})^{a-1} \bmod p = K^b \bmod p$ to the client.
K4: The client calculates $b^{-1} \bmod p-1$, and obtains the shared key $(K^{-b})^{b-1} \bmod p = K \bmod p$. [5]

In order to drive the protocol into the SMTP(Fig 3), as well as to increase the performance of generating a one - time secret and unique key, we pipeline the original protocol in order to produce the keys for various sessions in a pipelined manner where the parameters $\{K,a,b\}$ for the forth coming sessions are denoted as $\{K_n,a_n,b_n\}$.

The step K1 of the second session starts at step K3 of the first session to produce partly the K_n and save the produced partial key $K_n^{an\ bn}$ at the server. At the second session, the server calculates, $K^{bn}{}_n$ and sends it as K^{-b} for the second session to the client with K_n^{an} for the third session. The server saves the received partial key $K_n^{an\ bn}$ in each session. Hence, the client can actually calculate the key K for the current session on receiving K^b in step K1 of every session other than the very first session.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459 (Online), An ISO 9001:2008 Certified Journal, Volume 3, Special Issue 1, January 2013)
International Conference on Information Systems and Computing (ICISC-2013), INDIA.

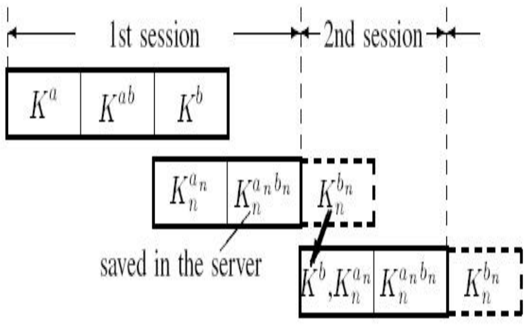

Fig 3. Pipelined execution of Shamir's Protocol for key generation.

3.3 *Detection of E-Mail Address of the Senders:*

In Fig.4, MA_s is the sender's mail id, where *(MA_s, D)* is a pair stored in the trusted core in the authorization stage, MA_r is the receiver's mail id. The objects in the braces in server and client are meant for the existing session. But they have been stored in the preceding session in the pipelined key establishment. Since a deposit to the trusted core is momentary, it expires at the time of TTL and the time when the server retrieves the object.

The operation proceeds as follows:

1: The server starts the detection process with *StartID*(K^b,K_n^{an})

2: The client determines the key K from the received K^b and deposits MA_r and K to the trusted core.

3: The client returns back the $(URI)_k$ and $K_n^{an\ bn}$ to the server.

4: The server then requests for sender's mail id MA_5, by sending the detection request ReqID(FROM,MA_r) to the trusted core which is mentioned by the received URI.

5: The trusted core then compares the received pair (FROM, MA_r) with the recorded pairs (MAs,MA_r) and returns (MA_s ,K, Match) to the server, where Match is equal to 1 if the matching succeeds, else 0.

The sender detection will succeed if the server obtains a message with Match of 1, else, it fails. The MA_s and K in the reply that is received from the trusted core in the 5th step is used for the robustness.

If a bot uses the user's mail address, the sender's detection will succeed. But in this case, the recipient can classify the received mail as spam, if necessary it can detect the bot from where the mail originates which is undesirable for the bot master. Hence, it provokes not to send a spam mail with the email addresses of legitimate users and will conceal the routing information.

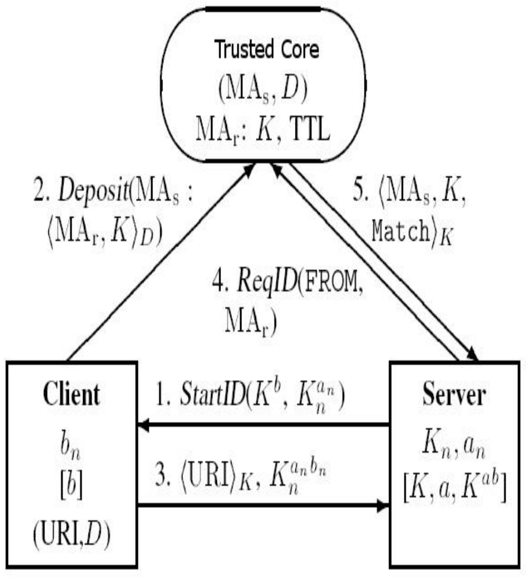
Trusted Core
(MA_s, D)
MA_r: K, TTL
2. Deposit(MA_s : $\langle MA_r, K\rangle_D$)
5. $\langle MA_s, K$, Match$\rangle_K$
4. ReqID(FROM, MA_r)
Client
b_n
[b]
(URI,D)
1. StartID(K^b, $K_n^{a_n}$)
3. $\langle$URI$\rangle_K$, $K_n^{a_n b_n}$
Server
K_n, a_n
$[K, a, K^{ab}]$

Fig 4. Sender detection protocol

3.4 *Embedding Shamir's Protocol in SMTP*

We have named our service as XSENDERID, which the server declares in the 1st step. When the client selects the service in the 2nd step, the server responds with a message for starting the service in the 3rd step. Later on, the server will send a key value pairs, CKEYb=Kb and NKEYa=Kna as the OK message for MAIL FROM and the client will send the pairs SENDER=URI and NKEYab=Knab as the rcpt-parameter of the RCPT TO, where Kb, Kna, Knab and URI represent K^{-b}, K_n^{an}, $K_n^{an\ bn}$ and $(URI)_k$ respectively.

The server responds with the detected result, so that if the result indicates a failure, the client has to close the channel without sending the message by sending QUIT instead of DATA. Since, the QUIT command removes the wasteful traffic for spam messages, the amount of communication resources required for email system will greatly be reduced.

IV. Conclusion And Future Work

We have presented a scheme to detect the email senders through trusted cores to the local client. The goal of the scheme is to provide the mail users, especially home users with an easy to install and usable service to detect the sender. This scheme consists of a process for authorizing the users to send mail, and a protocol to identify the mail addresses of the senders.

To authorize users, each administrator of each organization prepares a number of trusted cores for sender's detection.

These trusted cores are computers that are located in and by the organization. This mail service provider authorizes the users on the home computers when they subscribe to the mail service provider.

In our sender's identification protocol, the SMTP server and client will establish a single-time secret key K. Then pipelined version of Shamir's key establishment protocol [5] is used to improve the performance of the protocol and also to embed the protocol into the SMTP effectively. After the key establishment, the client momentarily deposits the key and the recipient's mail id MA_r to one of the trusted cores and informs the URI of the selected trusted core's services to the server. The server then sends the mail address of the sender that is obtained from the SMTP and the recipient address to the trusted core to retrieve the key K. The trusted core then returns the result of matching the pair (MA_s, MA_s) in the trusted core with the pair received from the server. In future, we intend to analyze this protocol with further enhancements that will make it cost efficient and less prone to attacks to the server.

REFERENCES

1. *Journal Papers*

[1] F. Li and M.-H. Hsieh, "An Empirical Study of Clustering Behaviorof Spammers and Group-Based Anti-Spam Strategies," 3rd Conf. on Email and Anti-Spam (CEAS), 2006.

[2] J. McGibney and D. Botvich, "A Trust Overlay Architectures and Protocol for Enhanced Protection against Spam," Int. Conf. on Availability, Reliability and Security (ARES), 2007.

[3] Chi-Yao Tseng, Pin-Chieh Sung and Ming-Syan. "Cosdes: A Collborative Spam Detection System with a Novel E-Mail Abstraction Scheme." IEEE Transactions on Knowledge and Data Engineering, VOL. 23, NO.5, MAY 2011.

[4] Zhenhai Duan, Peng Chen, Fernando Sanchez, Yingfei Dong, Mary Stephenson, and James Michael Barker. "Detecting Spam Zombies by Monitoring Outgoing Messages." IEEE Transactions on Dependable and Secure Computing, VOL. 9, NO. 2, MARCH/APRIL 2012.

2. *Text Book*

[5] P. van Oorschot, A. Menezes, and S. Vanstone, A Handbook of Applied Cryptography, Chapter 12, *CRC Press*, 1996.

[6] "An Introduction to Cryptography," *Network Associate, Inc* .

3. *Generic Information*

[7] M. Wong and W. Schlitt, "Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1," RFC 4408, 2006.

[8] J. Goodman, "Spam: Technologies and Policies," Microsoft White Paper, 2003.

[9] OECD Task Force on Spam, "Anti-Spam Toolkit of Recommended Policies and Measures," Report of the OECD Task Force on Spam, 2006.

[10] R.Austein, R. Arends, S. Rose M. Larson, and D. Massey,"Protocol Modifications for the DNS Security Extensions," RFC 4035, 2005.

[11] J. Klensin,"Simple Mail Transfer Protocol", RFC 2821, 2001.

[12] R. Siemborski and A. Melnikov, "SMTP Service Extension for Authentication," RFC 4954,

[13] Federal Information Processing Standards, "Announcing the Advanced Encryption Standard (AES)," FIPS-197, 2001.