

RPINA- Network Forensics Protocol Embedding Privacy Enhancing Technologies

Giannakis Antoniou* and Stefanos Gritzalis†

*Faculty of Information Technology
Monash University, Victoria, Australia
E-mail: gant2@student.monash.edu.au

†Info-Sec-Lab Laboratory of Information and Communication Systems Security,
Dept. of Information and Communication Systems Engineering,
University of the Aegean, Samos, 83200 Greece
E-mail: sgritz@aegean.gr

Abstract — Although privacy is considered to be the ultimate right for every user to enjoy intercommunications with security and anonymity, the provision for such a service could easily be adapted as a hiding cover by malicious users. Privacy Enhancing Technologies (PETs) should not only hide the identity of legitimate users but also provide means by which evidence of malicious activity can be gathered and revealed when necessary. This paper proposes a network forensics protocol called RPINA (Respect Private Information, Not Abuser) which may operate over PETs, without violating the privacy of innocent users, but only the privacy of abusers. This approach introduces a new dimension in the relation between these two opposite-goal technologies, which enhances their viability in the global network environment.

I. INTRODUCTION

Privacy Enhancing Technologies (PETs), which are responsible to hide the identity of users, have been designed in such a way preventing Network Forensics protocols from functioning and identifying the abusers. Offering unconditional privacy [1] to users, like the PETs [2-9] do, it minimizes the capabilities of network forensics protocols [10-12] and tools [13, 15], which are responsible to support the discovery of the identity of the abusers after an incident occurs. As a result, the abusers may take advantage and enjoy their privacy right, like every user, behind a PET just to perform malicious actions without being identified. Obviously, such an action should not be permitted. Although privacy is a respectable right for users, there is a reasonable and *sine qua non* need to violate the privacy of abusers, by allowing network forensics protocols functioning properly and identifying them only once an incident occurs. In order to achieve this goal, a mechanism is needed to:

- distinguish an innocent user from an abuser
- guarantee and respect the privacy right of innocent users offered by PETs
- gather strong lawful evidence about the actions of an abuser (in relation with an incident).

In order to achieve the above goals, this paper introduces the RPINA protocol, which respects the privacy of users as far as they do not attempt performing any malicious action

against Servers. An innocent user can continue enjoying the privacy right offered by a PET, while the abuser is not able to continue hiding behind a PET.

The paper is organized as follows: Section 2 describes related work; Section 3 listed the objectives of the RPINA protocol; Section 4 describes in detail the RPINA protocol; Section 5 explains the role of a ticket; Section 6 describes how the objectives introduced in section 3 have been achieved; Section 7 explains the effects of the RPINA protocol in different areas; Section 8 gives a number of brief case studies and Section 9 concludes the paper.

II. RELATED WORK

To the best of our knowledge, the only known attempt to bridge PET and network forensics is PPINA [14] protocol. However, it lacks of strong message integrity while a Server returns a message to the user through the PET. It has also an unnecessary layer of message confidentiality and protection against replay attack.

III. THE OBJECTIVES OF RPINA PROTOCOL

This section describes the objectives of the RPINA protocol. The RPINA protocol must operate over PETs and has the following characteristics:

1. *Be PET-independent*: The protocol must be PET protocol independent; therefore, the protocol should not affect the operation of a PET protocol but to support it. Also, no modification of a PET protocol should be needed in order to be embedded in the network forensics protocol.
2. *Add an authentication layer*: The protocol must provide a way to authenticate the user before using the PET, since, if the user tries to attack a Server, the abuser should be identified.
3. *Discourage a potential abuser to attack through PETs*: The discouragement of a potential abuser to attack through the PET is a layer, which offers protection to the Server.

4. *Offer end-to-end message confidentiality between the user and the Server:* Nobody, even a malicious PET, should be able to read the exchanged messages, except the user and the Server.
5. *Offer end-to-end integrity of the exchanged messages:* Nobody should be able to modify undetected any exchanged message.
6. *Not violate the privacy of the users:* The protocol must not reduce the level of privacy offered by the PET itself.
7. *Support the forensic investigation by co-operating with the Server-victim:* The protocol must ensure the Server-victim that the abuser will be revealed and a strong body of evidence will be built up concerning the abuser's (non-repudiated) actions.
8. *Offer an efficient investigation method:* The investigation method must be easy and time-effective, without affecting the functionality of the Server-victim during the investigation procedure.
9. *Be cost-effective and easy to implement:* The easier and cheaper a technique is, the more acceptances may be from the Internet community.

The RPINA protocol, which is introduced in the next section, aims to achieve the above mentioned objectives.

IV. THE RPINA PROTOCOL

The RPINA protocol offers a proactive forensic investigation service. It adds end-to-end confidentiality and integrity layers and forensic investigation services. The RPINA protocol operates over a PET protocol and is therefore a general solution, not linked with a specific PET. It does not offer intrusion detection functionalities. The following paragraphs explain the notation used and the operation of the RPINA protocol.

A=Anonymous User (AU) C=PET D=Server
 B=Directory Service (DS) / Forensics Investigation Entity (FIE)

$As\{Data\}$ = The Data is signed by the private key of an AU, where the public key, of that private key, is published

$Ae\{Data\}$ = The Data is encrypted by the public key of an AU, where the public key is published

$s\{Data\}$ = The Data is signed by the private key, which is created for the needs of a session. The public key of that private key is not published. Only the Server and the AU know that public key.

$e\{Data\}$ = The Data is encrypted by a secret key

$aKey\{Data\}$ = The Data is encrypted by the aKey

ForensicReceipt = The digest of the received data of a Server

The whole communication process is divided into 3 phases: the Initialization phase, the Main phase and the Forensic Investigation phase.

A. Initialization Phase

Before the actual communication (main phase) begins, the initialization phase is required. During this phase:

- The AU generates a session pair of keys used for the purpose of a session

- the DS gathers the fingerprint [$s\{Token2\}$] for future communication actions of the AU
- the DS issues a ticket [$Bs\{s\{Token2\}\}$] to the AU
- the Server validates that fingerprint
- the AU exchanges a secret key (session public key) with the Server

The AU generates a session pair of keys (Public/Private), where the session public key plays also the role of a secret key (for symmetric encryption) in order to provide end-to-end data encryption (between the AU and the Server). This secret key is valid only during that session. After the end of that session, the secret key turns to be invalid, and a new session pair of keys should be generated for future sessions, even if the participating entities are the same. The AU signs with the session private key and encrypts/decrypts/verifies with the secret key. The Server is capable to verify and decrypt the data [$se\{Data\}$] with the secret key.

The communication protocols between the participated entities during the initialization phase are the following:

A → B: $As\{Be\{aKey, Token\}, aKey\{s\{Token2\}\}\}$ (Step 1)

The AU calculates the Token [$Token = HashFunction(secret\ key)$] and the Token2 [$Token2 = HashFunction(Token)$]. The AU signs the Token2 [$s\{Token2\}$] by using the session private key. In addition, the AU generates a symmetric key (aKey) to encrypt part of the data [$s\{Token2\}$] sent to the DS. The [$s\{Token2\}$] is encrypted in order to avoid a possible attack from the Server.

B → A: $Ae\{Bs\{s\{Token2\}\}\}$ (Step 2)

The DS is responsible for the verification of the validity of Token2, based on the given Token (the Token2 must be the digest of the Token). The DS stores the message from Step 1 in order to prove, in case of a forensic investigation, that the specific AU was intending to communicate with a Server by using a secret key, which had the specific Token2. The [$Bs\{s\{Token2\}\}$] is the ticket which is necessary for the AU to establish a communication with the Server

A → C: $Bs\{s\{Token2\}\}, De\{s\{secret\ key\}\}$ (Step 3)

The AU sends the ticket to the PET. The AU also sends encrypted the signed secret key of the communication session [$De\{s\{secret\ key\}\}$] to the PET.

C → D: $Bs\{s\{Token2\}\}, De\{s\{secret\ key\}\}$ (Step 4)

The PET forwards the packet to the Server

D → C: $e\{s\{Token2\}\}$ (Step 5)

The Server is responsible to verify the validity and the authenticity of Token2, based on the given secret key. The Server stores the message from step 4 in order to prove later on (to the FIE) that the communication has used a secret key with a specific Token2. A successful verification of the DS by the Server means that the secret key is linked with the Token2 (secret key → Token → Token2), which Token2 is linked with the AU who has signed the message of step 1. However, the Server needs to verify that the AU also has the session private key by verifying the signature of the secret key [$s\{secret\ key\}$] and [$s\{Token2\}$] by using the secret key as a session public key. Once the Server verifies successfully the information, it encrypts the [$s\{Token2\}$] with the session secret key and send it to the PET.

C → A: e{s{Token2}} (Step 6)

The PET forwards the encrypted information to the AU.

Generating private/public key pairs is a computationally intensive procedure. However, an AU can generate several session pairs of keys during times of low system load and request tickets from the DS (one ticket for each session pair of keys). A ticket is server-independent and time-independent; therefore, a ticket can be used for future communication with any Server at any time. A ticket is useful only to the entity who knows the related session private key.

B. Main Phase

Once the AU and the Server have exchanged the secret key and the Server has validated and authenticated the value of the Token2, the session has been established and the Main Phase is ready to begin. The AU can enjoy the confidentiality, integrity, authenticity of the exchanged messages and the anonymity offered by the PET, while the Server is ensured that the identity of the AU will be revealed (from the FIE) in case the AU is an abuser:

A → C : se{Data,nonce} (Step 7)

The AU encrypts, with the session public key and signs, with the session private key, the message [Data, nonce]. Each message of Step 7 contains the Data and the nonce. The nonce is a counter, unique for each session, which helps to avoid a replay attack.

C → D : se{Data,nonce} (Step 8)

The PET forwards the packet to the Server

D → C : e{Ds{Data}} (Step 9)

If the Server receives the same encrypted message (data and nonce) twice, the Server rejects the last message. The nonce is also useful during the Forensic Investigation Phase, when the FIE tries to determine whether a series of packets are malicious or not. The Server stores the message of Step 8 in order to prove, in case of an attack, that the AU who has the session private key of the session public key (secret key) has sent the message. The Server decrypts and verifies the message [se{Data, nonce}] by using the secret key. In case that the Server wants to reply, it will encrypt the data with the secret key, and it will send the data to the PET.

C → A: e{Ds{Data}} (Step 10)

The PET forwards the packet to the AU

C. Forensic Investigation Phase

During this phase, the DS plays the role of the FIE. Once the Server detects an attack, the Server only needs to contact the FIE and send the malicious messages (Step 4 and Step 8). The FIE verifies the authenticity and the integrity of the messages as well as whether the messages are malicious or not. In case the FIE concludes that the messages are malicious, it replies with evidence (which proves the involvement of the abuser and cannot be repudiated) and the identity of the abuser. It is particularly important that in RPINA protocol, the Forensic Investigation phase can take place while the Server is still functioning. In current network forensic investigations, it is usual for the Server needs to stop its operation after an incident, while the forensic entity investigates the Server for

evidence. During the Forensics Investigation phase, none of the entities needs to reveal their private keys in order to prove their claims. However, the Server is required to reveal the exchanged secret key used during the communication with the malicious user.

D → B: Bs{secret key, Bs{s{Token2}}} (Step 11)

The Server sends the ticket (generated and signed by the DS) to the FIE including the secret key.

B → D: Bs{Case ID} (Step 12)

The FIE verifies the validity of the secret key (secret key → Token) and the authenticity of the Token2 [s{Token2}] and replies (Step 12) with a Case ID. This identity (Case ID) is the identification of the current investigation case.

D → B: Bs{Bs{se{Data,nonce}}} (Step 13)

The Server sends to the FIE all the communication messages for that session received by the AU.

D → B: Bs{CaseID} (Step 14)

The Server sends a message to inform the FIE that there are no more messages to send

B → D: Bs{ForensicReceipt,Case ID} (Step 15)

After the FIE receives all the necessary messages, it replies with a ForensicReceipt. The ForensicReceipt is the hash value of the Step 11 and Step 13. The Server, also, calculates the ForensicReceipt. These two ForensicReceipts should have the same value. Otherwise, there is a problem with the integrity of the exchanged messages. The ForensicReceipt ensures that the FIE received all the data that the Server has sent. The FIE, also, cannot deny that the Server asked the FIE to investigate the case (The FIE signs the ForensicReceipt).

B → D: Bs{Bs{Bs{secret key,ForensicReceipt,aKey,IP Address,As{Bs{aKey,Token},aKey{s{Token2}}}}} (Step 16)

After the FIE concludes that the messages were malicious, the FIE sends the IP Address of the AU and evidence (from Step 1) which proves the involvement of the AU. Only an entity who knows the session private key could sign the Token2. The Server can now submit the message of Step 16, and the malicious communicated messages (Step 13) to the court of law as evidence of the AU's actions.

V. TICKET'S STRENGTH AND ITS IMPORTANCE

The ticket [Bs{s{Token2}}] is issued by the DS after a user's request [As{Bs{aKey,Token}, aKey{s{Token2}}}]. The ticket has the signature of the DS and ensures that:

- DS verifies the user's signature of the request
- DS verifies that Token2 = HashFunction(Token)

The ticket is meaningless for a user who does not have the related session private key because:

- The Server, during the initialized phase, tries to verify that the signature of [s{Secret Key}] can be validated by the Secret Key. If the secret key, which is actually the session public key, does not validate the signature, the validation fails and the Server reject the request.
- The Server, during the main phase, tries to verify the signature of [se{Data, nonce}] by using the secret key. This signature cannot be generated without the use of the related session private key.

We conclude that a successful attack to the ticket has no meaning without the knowledge of the session private key. A chain of evaluated information associates the sender of a message with the identity of the sender.

Message \rightarrow ticket \rightarrow request for a ticket \rightarrow sender's identity

Because of the high level of sensitivity and importance of those tickets and AU's requests, the DS stores them in a secure place. There is a plethora of available store media, however, for higher level of availability, the use of write-once media is recommended. Although the DS is a single point of failure, it has no risk from forensic investigation point of view. The AUs' requests as well as the requests of the Servers to investigate a case can be stored in a store media and after the DS goes on-line it continues with the investigation procedure.

VI. OBJECTIVE ACHIEVEMENTS

The goal of the RPINA protocol is to achieve the objectives mentioned in section III. This section explains one-by-one how the objectives have been achieved.

1. *Be PET-independent:* Any PET protocol can be embedded in the RPINA protocol, giving the flexibility to each PET to use different protocol, without affecting the operation and capabilities of the RPINA protocol.
2. *Add an authentication layer:* The user contacts with the DS, during the initialization phase, requesting a ticket. The request is signed by the user and the DS validates the request and issues a ticket. There is no doubt that the request came from a specific user (we assume that nobody has compromised the public/private key of the user). The Server does not authenticate directly the user, but indirectly by validating the ticket. If the ticket has the valid signature of the DS, the Server is sure that the DS authenticated that user, and in case of an attack (from that user), the DS will reveal the identity of the user.
3. *Discourage a potential abuser to attack through PETs:* The Server requires a ticket when a packet is coming from a PET. Otherwise, the Server rejects the packet; therefore, the option to attack through a PET is not wise because the abuser has to get a ticket from the DS. Although a PET will offer anonymity to the user, the RPINA protocol will reveal his/her identity. For this reason, a potential abuser has a reason to avoid attacking through PETs.
4. *Offer end-to-end message confidentiality between the user and the Server:* One of the goals of the initialization phase is for a user to exchange confidentially a secret key with the Server. This secret key is known only by the user and the Server and it is used for a symmetric encryption in order to offer end-to-end message confidentiality.
5. *Offer end-to-end integrity of the exchanged messages:* The secret key which is exchanged between the user and

the Server during the initialized phase, plays also the role of the session public key. The integrity of that public key is also verified. There are three ways to verify it

- the signature of the [s{Secret Key}] must be validated by the secret key
 - the ticket [Bs{s{Token2}}] contains the Token2, which is the digest of the digest of the secret key
 - the [s{Token2}] must also be validated by the secret key
6. *Not violate the privacy of the users:* One advantage of the RPINA protocol is also a downside because of the fact that an entity (DS) can associate the token with the identity of a user. However, the end-to-end message confidentiality which the RPINA protocol offers which enforces the privacy of the users, it balances the privacy violation of the RPINA protocol. Moreover, the RPINA protocol enforces the privacy of a Server during a forensic investigation procedure, because the strong evidence of the user's malicious actions is embedded in a series of messages; consequently, the Server does not need to make available the Server's computer-victim to the FIE or to the judge and violates the privacy of that Server.
 7. *Support the forensic investigation by co-operate with the Server-victim:* A ticket [Bs{s{Token2}}] contains the signature of the DS, which means that during that session if a user attacks, the DS is responsible to unveil the identification of that user. Otherwise, responsible for the attack is the DS because of issuing the ticket without authenticating the user. A valid ticket obligates the signer (DS) to unveil the identification of the user, as well as to provide evidence to the Server for that claim. This fact obligates the FIE to co-operate with the Server.
 8. *Offer an efficient investigation method:* The proactive controllable investigation method ensures the efficiency of the forensic investigation procedure. The DS collects the "fingerprints" of the future communication action of the user. Any action against the Server is identifiable, as far as the Server checks and validates (following the RPINA protocol requirements) the ticket and the exchanged messages of the user.
 9. *Be cost-effective and easy to implement:* The RPINA protocol has no complex or difficult implementation. However, the DS requires a relative large storage capacity in order to store the tickets' requests of the users (less than 1kb per ticket's request). The users and Servers require following the RPINA protocol, while no any modification of the PETs protocols is required. The RPINA protocol requires a Public Key Infrastructure (PKI). This is necessary because the digital signature derives from the use of asymmetric encryption, where the private key is known by only one entity (as far as the owner of the private key keeps it in secret), and only that

entity can sign a message and produce the signature. Based on this uniqueness, the digital signature has the known strength. Therefore, the expensive use of the PKI is unavoidable.

As a result for the achievement of the above objectives, there is an impact in a number of areas. The next section describes how these areas can be affected.

VII. EFFECTS

This section explains the effects of RPINA protocol in the area of PET and computer/network forensics. It describes also the impact of the RPINA protocol from the side of a user/abuser and a server.

PET environment: The PETs are not very popular because the main participated entities, the Server and a user, have significant drawbacks by using PETs

- At present, the PETs protocols have direct access to the exchanged messages and violate the right of the user to have a confidential communication. Because of that, users avoid using the PETs.
- Additionally, some servers avoid accepting requests coming from PETs because they know that a possible abuser is not only able to use the PET to hide his/her identification but also no mechanism exists to prove that a specific user was the abuser.

The RPINA protocol eliminates these issues of the users and servers because:

- The PET cannot access the exchanged messages
- The Server not only can identify an abuser, but also can gather evidence and prosecute him/her.

A future PET protocol embedded in the RPINA protocol can be designed without trying offering end-to-end services but only privacy protection to the user. This approach will result to build more efficient PETs focusing only in privacy aspect of the problem rather than trying to develop and embed forensic investigation techniques in each PET individually.

Computer/Network Forensic area: The proactive investigation method employed in the RPINA protocol, reduces the time and the effort of the investigators to find out and prove the involvement of an abuser in an attack, even though that abuser was anonymous during his/her communication. In addition to this, the methodology prevents the charging of an innocent user, while guarantees the accusation of the actual abuser. There is no large volume of data that the investigator needs to investigate, whereas the present techniques face this problem while the volume increases very rapidly and the investigators are seeking for techniques in order to manage the large volume of data.

There is an argue [2] that currently the techniques used in order to track down an abuser and reconstruct the events during the attack have the below disadvantages:

- Large volume of Data
- Incompleteness of Logs
- Long Response Times
- Lack of Mechanisms to Share Logs
- Unreliable Logging Mechanisms

The RPINA protocol addresses these issues in a very efficient way:

- **Large volume of Data:** The two entities who need to store information are the DS and the Server. The DS needs to store only the request of the user while the Server needs to store all the messages received by that user during a session. The Server can delete the messages after the session finishes and no incident has occurred.
- **Incompleteness of Logs:** The Logs contains (with the collaboration of the Server and the FIE) all the necessary information proving the involvement of a user in the attack.
- **Long Response Times:** The RPINA gathers all the necessary information (evidence) to accuse the abuser. Despite the fact that the FIE may need to inspect manually the messages before unveiling the identity of the abuser, an automatic mechanism can be employed to proceed immediately and unveil automatically the identity of the abuser.
- **Lack of Mechanisms to Share Logs:** The RPINA protocol guarantees that the logs of the FIE will be used to help the Server in the investigation.
- **Unreliable Logging Mechanisms:** All the information/messages stored in the logs have been signed and any modification of these log files is identifiable.

User/Abuser: The abusers have a reason to avoid attacking through PETs because of the fear of being identified, while users enjoy their e-activities (i.e. e-commerce, e-banking, and e-health) securely (confidentiality, integrity, authenticity) and anonymously.

Server: Currently, the cost performing a forensic investigation is high. The companies developing internal expertise in computer/network forensics are very few. Thereby, the RPINA protocol can save money to the company-victim as the forensic investigation starts immediately after the incident occurred and the location of the evidence is known (DS/FIE has the evidence).

VIII. CASE STUDY: AN IMPERSONATE ATTACK

In this scenario, the AU is an innocent user who wants to communicate with the Server. There is an abuser who eavesdrops their communication and tries to attack with the message that contains the signature of the AU. The AU requests a ticket from the DS. The DS checks the request and issues the ticket. The AU receives the ticket and forwards it to the Server through the PET. Once the Server receives the initialized packets and makes the necessary checks, it returns the [e{s{Token2}}], which is encrypted with the shared secret key. During the initialized phase, the abuser does not try to do anything more than eavesdrops the communication. After the AU receives the necessary packet [e{s{Token2}}], the Main phase begins. The AU encrypts the data and the nonce with the secret key and then signs the message. The nonce is a counter of the messages. Each message has its unique nonce. The AU sends the message to the PET and the PET forwards

it to the Server. During this communication session, the abuser eavesdrops the message [se{Data,nonce}]. The abuser tries to use the eavesdropped message in order to cause a problem to the AU and to the Server.

- A. *Option 1:* The abuser tries to modify the message, and especially the data. However, the message is not only encrypted with a secret key, but it is also signed by the AU (only the AU has the private key); therefore any modification will be detected by the Server (The Server has the public key and it can verify the signature of the AU).
- B. *Option 2:* Another option for the abuser is to re-send the message to the Server, which includes the signature of the AU, without any modification. In such a case, the Server ignores the message of the abuser because the AU has already sent it to the Server. The Server does not accept twice the same nonce in a session.
- C. *Option 3:* An alternate option that the abuser may apply is to catch the message without letting it reach the Server. By acting like this, the abuser can send the message in a later time and the Server will not be able to notice that the message came from an abuser. However, in the meantime, the session may be expired and the Server will request from the abuser to start the initialized phase before accepting the message.
- D. *Option 4:* The abuser uses the eavesdropped message to do a Denial of Service (DoS) attack. The abuser sends that message to the Server multiple times. The Server has only one defense against the abuser's attack. The Server decrypts the message with the secret key (Symmetric key) and once the nonce has already received by the Server, the message will be ignored. The RPINA protocol does not offer protection against this type of attack, and the Server may be a victim without discovering the identity of the abuser. Although the abuser uses the RPINA protocol to attack the Server, the abuser could also attack by using any other protocol, for instance the Http protocol. Responsible for this attack is not a vulnerability of a protocol. However, a protocol which is processor-intensive is more vulnerable to the DoS. In the case of the RPINA protocol during the Main Phase, the Server needs to decrypt the message with a symmetric encryption and then to check if the nonce has already been used before.

IX. CONCLUSIONS

The provision for privacy and anonymity to users can also provide an environment within malicious users can hide their actual identity. A key driver of the wider adoption of privacy enhancing technologies would be the ability of forensic entities to gather and reveal evidence of malicious activity while legitimate users are still offered anonymity. In this paper, we have provided an environment through which the anonymity of users not engaged in malicious activity is

protected while such evidence can be gathered when network abuses occur. The bridging of the two opposite/complement-goal technologies, PET and Network Forensics, has been achieved by the use of RPINA protocol. The RPINA protocol offers non-repudiation actions of the users, by using the digital signature, adds a layer of message confidentiality, by using a secret key, respects the privacy information of the Server during the investigation, and decreases significantly the duration and the complexity of an investigation procedure. As future work, we aim to study related economics issues for the proposed solution.

REFERENCES

- [1] A. Pfitzmann, M. Hansen (2006), "Anonymity, Unlinkability, Unobservability, Pseudonymity and Identity Management - A Consolidated Proposal for Terminology", available at: <http://dud.inf.tu-dresden.de/literatur/>
- [2] S. Gritzalis (2004), "Enhancing Web Privacy and Anonymity in the Digital Era", Information Management and Computer Security, Vol.12, No.3, pp.255-288, Emerald
- [3] Anonymizer (2003), available at <http://www.anonymizer.com>
- [4] M. Reiter, A. Rubin, "Crowds: Anonymity for web transactions" (1998), ACM Transactions on Information and System Security (TISSEC), Vol. 1, No. 1, pp. 66 – 92
- [5] R. Dingledine, N. Mathewson, and P. Syverson (2004), Tor: The Second-Generation Onion Router. In Proceedings of the 13th USENIX Security, Symposium,
- [6] C. Shields, B. N. Levine, "A Protocol for Anonymous Communication Over the Internet" (2000), Proceedings of the 7th ACM conference on Computer and Communications Security
- [7] P. Golle, A. Juels, "Parallel Mixing" (2004), Proceedings of the 11th ACM conference on Computer and Communications Security
- [8] U. Moller, L. Cottrell, P. Palfrader, L. Sassaman (2003) "Mixmaster Protocol", Version 2. Draft, available at <http://www.abditum.com/mixmaster-spec.txt>
- [9] M. Rennhard, B. Plattner (2004), "Practical anonymity for the masses with morphmix", In A. Juels (Ed.), Financial Cryptography. Springer-Verlag, LNCS 3110
- [10] B. Carrier, C. Shields (2004), "The Session Token Protocol for Forensics and Traceback", ACM Transactions on Information and System Security, Vol.7, No. 3, pp. 333-362
- [11] K. Shanmugasundaram, A. Savant, H. Bronnimann, N. Memon (2003) "Fornet: A distributed forensics network", In The Second International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, St. Petersburg, Russia
- [12] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S. T. Kent, W. T. Strayer (2002), "Single-Packet IP Traceback", IEEE/ACM Transaction on Networking, Vol.10, No.6
- [13] B. Hards, "A guided tour of ethereal" (2004), Linux Journal, Volume 2004, No. 118, p.7
- [14] G. Antoniou, C. Wilson, D. Geneiatakis (2006) "A Forensic Investigation Protocol for Privacy Enhancing Technologies", In Proceedings of the 10th IFIP CMS'06 Communications and Multimedia Security Conference, Iraklion, Greece
- [15] B. Nelson, A. Phillips, F. Enfinger, C. Stuart (2004) Guide to Computer Forensics and Investigations