12-5-2006

Uncontrollable Privacy - The right that every attacker desires

Giannakis Antoniou
University of Melbourne

Stefanos Gritzalis
University of the Aegean

DOI: 10.4225/75/57b6552c34764
4th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia, 5th December, 2006

Uncontrollable Privacy - The right that every attacker desires

Giannakis Antoniou* and Stefanos Gritzalis†
*Dept. of Computer Science and Software Engineering
University of Melbourne
giannos@iprimus.com.au
†Dept. of Information and Communication Systems Engineering,
University of the Aegean, Samos, 83200 Greece
sgritz@aegean.gr

Abstract

The request of the Internet users enjoying privacy during their e-activities enforces the Internet society to develop techniques which offer privacy to the Internet users, known as Privacy Enhancing Technologies (PETs). Among the Internet users, there are attackers who desire more than anything else to enjoy privacy during their malicious actions, and a PET is what they were looking for. Thus, although a PET should offer privacy to the internet users, proper techniques should also be employed in order to help the victims during the investigation procedure and unveil the identification of the attackers. The paper summarizes the current design issues of PETs and introduces additional issues in order to offer forensic investigation services. To the best of our knowledge this is the first attempt which it proves (the obvious) that the existing PETs do not meet accountability requirements. By knowing explicitly the reasons the PETs are inefficient offering accountability, it is the most appropriate way to make PETs offering higher level of accountability without decreasing the level of the privacy offered.

Keywords

Privacy, Accountability, Privacy Enhancing Technology, Network Forensics.

INTRODUCTION

"Everyone has the right to freedom of opinion and expression; this right includes freedom to hold opinions without interference and to seek, receive and impart information and ideas through any media and regardless of frontiers." (Article 19, Universal Declaration of Human Rights). Internet users are humans who desire to use this right in the Internet community as well. The technology responsible to offer privacy to the Internet users is called Privacy Enhancing Technology (PET). Despite the large range of PETs (Gritzalis, 2004; Anonymizer, 2004; Reiter and Rubin, 1998; Dingledine et al, 2004; Shields and Neil Levine, 2000; Golle and Juels, 2004; Ulf et al, 2003; Rennhard and Plattner, 2004), none of them has been designed in such a way to offer forensic investigation techniques, in order to discourage potential attackers to use the PET as a shield of privacy protection.

The section 2 describes the PET framework and three PETs; section 3 summarizes the current design criteria for a PET; section 4 proposes additional design criteria that every PET should follow in order to offer forensic investigation techniques; section 5 concludes the paper giving directions for future research.

PRIVACY ENHANCING TECHNOLOGIES

The PETs have one or more privacy enhancing entities (PEE) participated in forwarding the messages from the user to the Server and backward. The Server can identify the sender of the message based on the IP Address of that message. However, the message received by the Server has the IP Address of the last PEE, because the PEE replaced the IP Address of the sender with his/her IP Address before forwarding it to the Server. By following this technique, the PET achieves to hide the identification of the user from the Server. From the large list of available PETs (offering communication anonymity), there is below a description of three of them, the Anonymizer (Anonymizer, 2003), the TOR (Dingledine et al, 2004) and the Crowds (Reiter and Rubin, 1998). These PETs have been selected because their characteristics represent the majority of PETs' characteristics.

Anonymizer

The Anonymizer (figure 1) uses a single intermediate node, which is responsible to forward the requests (like a proxy server) to the appropriate entity. The Anonymizer supports only http requests.

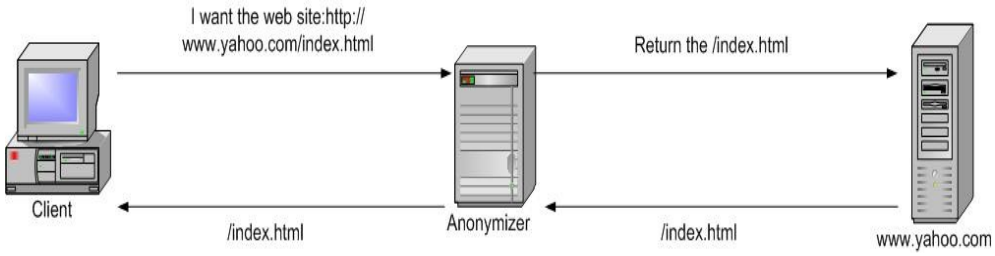

Figure 1 Anonymizer

In the example (figure 2) the Client A communicates with Amazon, the Client B communicates with Yahoo and the Client C communicates with Ups.

The three Servers cannot determine the original sender of the requests, because the requests have passed through the Anonymizer and the Anonymizer has already altered information from layer 4 and below (including the source IP Address). However, a global eavesdropper can link the sender with the receiver of the packets.

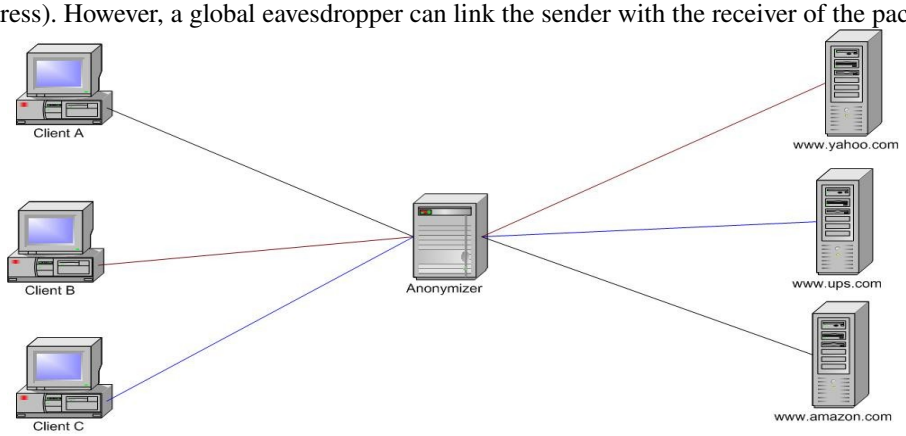

Figure 2 Example of an Anonymizer

Tor - The updated version of Onion Routing

The client, before sending the real message, exchanges secret keys with some privacy enhancing entities (OR1, OR2 and OR3). After the client has exchanged the keys of the necessary entities, it encrypts the message with the key of the last entity. Then the encrypted message is encrypted again with the key of the previous-last entity. This procedure continues until the encrypted message is encrypted with the key of the first entity. Then, the client sends the encrypted message to the first entity, which decrypts the message, and it sends the message to the next entity, until the message reaches the last entity. The last entity decrypts the encrypted message and then the clear message is sent to the original server as shown in figure 3

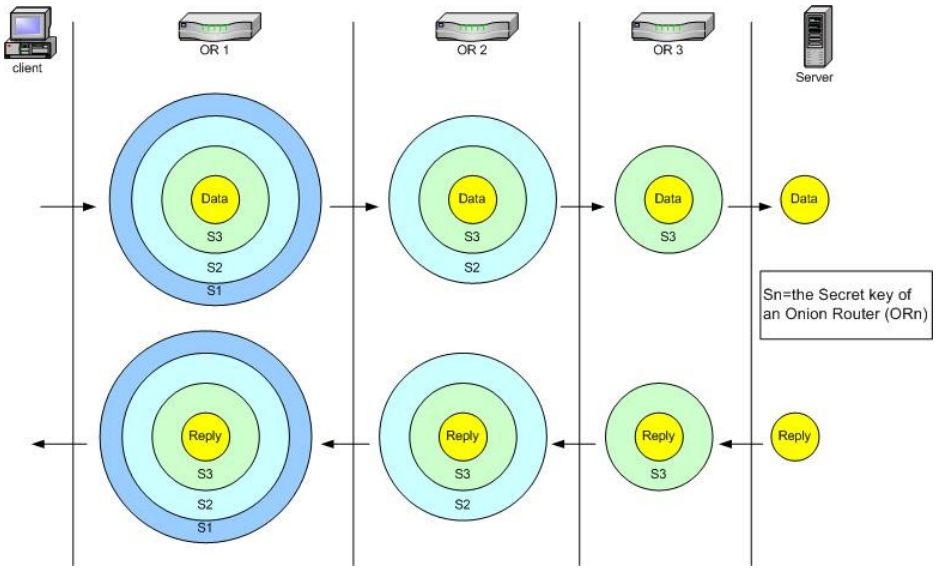

Figure 3 TOR protocol

Crowds

The philosophy of the structure and the actions of the Crowds is based on the fact that the more hosts (called "jondo") are participating in the forwarding process of a request, a higher level of anonymity (Pfitzmann and Hansen, 2006) is achieved. Each participant of the Crowds executes a process on his/her computer, which is called "jondo". Each jondo is responsible to forward requests coming from another jondo. When a new jondo is created (and added to the framework), it makes its identity public. The packet passes through a number of jondos before reaching the final destination. Each jondo selects randomly the next jondo. Once a virtual path is created, the jondo uses the same path for future connections. After 24 hours jondo discards the current path and creates a new one.

The (Luis von et al, 2006) describes a principle of transforming almost every PET to offer message traceability. The (Golle, 2004) introduces 3 protocols which offer reputability in mix networks. The (Antoniou et al, 2006) and (Antoniou and Gritzalis, 2006) introduce a new protocol called RPINA (Respect Private Information Not Abuser) which can embed any PET and offer network forensics functionality to the Server as far as the anonymous user does not perform any malicious action against the Server. Although the importance of the above papers, none of them define the necessary functionalities that a PET should offer in order to achieve message traceability.

CURRENT PET'S ISSUES

This section describes how the PET protocols approach the issues below, which are currently under PETs' consideration. The PET should be able to:

a. forward the message to the Server as quickly as possible

An Anonymous User (AU) wants to enjoy the anonymity offered by the PET without sacrificing the performance of the communication. However, the use of cryptography, which is used by the most PETs, reduces the communication's performance.

TOR offers a circuit-based communication. Before a user sends information to a Server, a circuit must be created. During the creation of the circuit, a user uses asynchronous encryption which is a time-consuming procedure. After the circuit is created, the user uses symmetric encryption, which is faster. In order to avoid the delay of the creation of a circuit, a user can use an existed circuit for more than one communication session. However, the extended use of the same circuit means that the participated entities use the same secret keys to encrypt more messages, which increases the possibility of an eavesdropper to find out the secret key. It is also

important the fact that the user is responsible to select the Onion Routers that will participate in the circuit; therefore, the user may balance the level of the privacy with the level of performance (the more Onion Routers are participated, the higher level of privacy and the lower level of performance is achieved)

In the Crowds, each jondo is responsible to decide whether to send the message to the destination server or to forward it to another jondo; thus, it decreases the flexibility of the user in order to control the level of privacy and the level of performance. Moreover, during the creation of the path, there is a delay due to the user needs, in order to get a list of the available jondos with their secret keys from the Blender.

The single-proxy PETs, like the Anonymizer, offer the limited possible delay, because the lack of encryption avoids delay of forwarding messages.

b. resist against outsider attackers

An attacker, who is not one of the privacy enhancing entities, is called an outsider attacker. A potential attacker may select a number of ways to attack and link the identification of the sender (user) with the receiver (server). The PET is responsible to resist, by confusing an outsider attacker, who tries to intrude the system. In (Argyrakis et al, 2003) there is a complete analysis of those attacks (Passive attack, Active attack, Trace Back Attack, Eavesdroppers, Message Attack, Timing attacks, Flooding attacks, Connection periods attacks and Cookies).

Without any encryption, the Anonymizer is not capable to offer high level of protection to a client against traffic analysers, or trace back attacker and other kind of attacks. They are also vulnerable from a global eavesdropper (an eavesdropper who has the ability to eavesdrop all the communications).

The TOR offers confidentiality between the user and the last privacy enhancing entity. An eavesdropper can only access the plain message during the communication of the last PEE (Privacy Enhancing Entity) and the Server. However, the attacker will not be able to link/match the plain message with the original sender of the message.

The TOR offers padding (a technique which adds dump data in order to create a fix-size message) to the messages, which protects against the trace-back attack as well as the passive attack. Also, the multiple layers of the encrypted message are an additional level of offering defence against those attacks.

The padding technique sacrifices the performance of the protocol because it sends dummy data. The dummy data not only travel from one node to another overloading the network, but it must also be encrypted and decrypted, which is a time-intensive procedure.

The Crowds, like the TOR, has an efficient level of protection against outsider attacks. However, both PETs are vulnerable against time analysis attack.

c. resist against insider attackers

The potential attacker is one of the privacy enhancing entities. The PET is responsible to protect the privacy of the user against compromised nodes (Malicious Collaborators - Argyrakis et al, 2003)

The TOR has a very efficient way to protect against an attack from inside. Even if one of the PEE participated in a circuit is honest, the attackers cannot join the sender's identity with the receiver's identity.

The Crowds has less efficient techniques, than the TOR, to resist against insider attackers because all the participated PEE in a path have access to the forwarded plain message, whereas in TOR, only the last entity has access to the plain message.

The Anonymizer is a single-proxy PET; therefore, there is no insider attacker other than the same single-proxy PEE itself.

d. offer fault tolerant to the privacy of the message

In a PET environment with several privacy enhancing entities (PEE), it is usual for an entity to stop functioning while forwarding a message. In such an event, the PET should have a recovery mechanism.

The TOR has no recovery method after a PEE stops functioning. The tcp protocol will be responsible to inform the user of a time-out delay.

The case with the Crowds, is similar to this, where the unlinkability (Pfitzmann and Hansen, 2006) offered by the PET has a result to lack of fault tolerant techniques.

The Anonymizer as a single-proxy PET, could not be able to offer a fault tolerant mechanism.

e. serve as many Internet services as possible

For a user, it is not acceptable to use a different PET for each Internet service. Therefore, a PET should support as many protocols as possible (i.e. Telnet, Ftp, Http, Https, IRC, and Email). However, some PETs have been designed to offer specific Internet services to the client (i.e. Web Services).

The TOR supports all the services which run over TCP protocol.

The Crowds and the Anonymizer support only limited protocols of the 7th OSI Layer, including the HTTP protocol.

f. be adapted easily

A user, who wants to use a PET without the need of modifying the existed system infrastructure or Operating System, is more likely to use it. The distribution of software which should offer integrity is not an easy task. In case a user executes software, it is a risky procedure, especially when the integrity of that software is not guaranteed. The software may be a Trojan horse or a Virus.

A user, who wants to use the TOR, requires executing software at his/her computer. Although the program does not need administrator's privileges, it may be harmful. The same issue appears in the Crowds, where a user executes the software (called jondo).

In the case of Anonymizer, what is required to be performed is just to configure the Browser to support this PET. This is actually its biggest strength among the other issues.

g. be acceptable for the sender (client) and the receiver (Server)

A PET, which is acceptable neither from clients nor from Servers, is quite a useless technology. The services and functionalities of a PET must respect the participated entities (client and Server).

In a TOR environment, the user gives authority to the last PEE of the circuit to have access to the plain message. On the other side, a Server knows that in case of a malicious action of an anonymous user, no mechanism exists to unveil the identification of that user.

FORENSIC INVESTIGATION ISSUES

It is interesting to study the behaviour of the PETs in a case where an abuser uses the PETs to attack, and what techniques are employed to guarantee abuser's identity. In a scenario where an attacker uses a PET to hide his/her identity, the victim-server requests from that PET to help with the investigation. The malicious message has the identification of the PET; therefore, there is a possibility for the message to have been originally sent by the PET and not from an Internet user. Thus, the PET should be accused if PET does not provide evidence proving the involvement of another entity (Internet user) in that malicious action.

A proper PET should offer not only privacy to the user but also forensic investigation techniques to the Server.

During a forensic investigation, there is a need to consider the below questions that a PET should be able to answer.

a. Can a PET know that a message was sent by itself?

After a Server detects a malicious message coming from a PET, it requests from that PET the identification of the user. The Server, also, sends the message received by the PET. How can the PET be sure that the message has

been sent by itself? A precondition to answer this question is that the message should be linked with the PET. One technique to link the PET with a message is that the PET signs the message before sending it to the Server. Therefore, if the Server sends the signed message to the PET, the PET cannot deny that the message was sent by itself.

If the message is encrypted with the Public Key of the Server, the PET cannot sign the unencrypted message, but the encrypted message. The below scenario describes the way that the PET proceeds with the investigation while the message is encrypted with the public key of the Server.

The AU sends a message encrypted with the public key of the Server

AU ➔ PET: encryptedMessage

The PET signs the encrypted message and forwards it to the server.

PET ➔ Server: sign{encryptedMessage}

The Server detects that the message is malicious and sends both the encrypted and the decrypted messages back to the PET.

Server ➔ PET: sign{encryptedMessage}, plainMessage

There is an issue. The PET has signed only the encrypted message, not the plain message. How can the PET know that the plain message has been sent by that PET?

Although the PET knows that it has sent the encryptedMessage to the Server, it needs to prove that the plainMessage is the decrypted message of the encryptedMessage.

The PET encrypts the plainMessage with the public key of the Server.

x=encrypt(plainMessage, Server's Public Key)

If the x is equal to the encryptedMessage, then the unencrypted message of the encryptedMessage is the plainMessage; therefore, the PET knows that the plainMessage has been forwarded by itself, without any signature on this specific message.

The PET has an alternative option to determine if the message was sent by itself or even sent without any signature. The PET can generate the digest message of each message and store it for future use. However, the PET will need to have large store media to store all this information. In case the PET wants to verify that a message was sent by itself, the PET can generate the digest message of that message and compare it with the stored digest messages. The PET can determine if the message was sent by itself, however it can deny this fact.

Although none of the current PETs sign the message before sending it to the Server, this functionality can be embedded in the future, without reducing the level of the PETs' offered privacy. In addition to this, none of the PETs store the forwarded messages. Thus, the PETs have lack of techniques for identifying the messages that they forward.

b. Can a PET decide if a packet is malicious or not?

The PET must adopt a mechanism capable to detect whether a message is malicious or not. In order to accomplish that, the message must be decrypted.

The current PETs do not attempt to offer forensics investigation services; therefore, none of the PETs has employed such a mechanism. Furthermore, the incorporation of this service in the PETs does not lead to privacy violation or any other kind of threat.

c. Can a PET identify the sender (Anonymous User) of a message?

The PET must be sure that the message has been originally sent by a specific user even after the communication session has been completed. Without the prior authentication of the user, it is not possible to be sure about the

originality of the message. The authentication can be achieved with several ways. For instance, one way is that the AU can sign the message before sending it to the PET. In that case, the PET must remove the signature of the AU and forwards only the unsigned message to the Server, in order to hide the AU's identification.

Another way for the AU to be authenticated by the PET is that the AU can send a login/password with the message to the PET. However, a prior registration procedure must be applied.

The TOR protocol has a build-in authentication technique, even without the use of a digital signature. The first privacy enhancing entity (PEE) in the circuit can be sure about the identity of the AU. However, a victim-Server cannot contact directly that PEE requesting the identity of the AU, but from the last PEE of the circuit. The last PEE must request information from the previous PEE, until the request reaches the first PEE. Thus, a broken PEE results in breaking the chain between the first PEE and the last PEE of the circuit, consequently, the identity of the attacker remains unknown.

The Crowds protocol authenticates (login/password technique) the AU through the Blender before commencing sending information. Although the authentication procedure, takes place, it does not link the exchanged messages with the identity of the actual sender (AU). Therefore, the originality of a message cannot be identified or traced.

The Anonymizer does not offer any technique to identify the sender (AU) of a message.

d. Can a PET identify the sender of a message even with a compromised intermediate PEE?

In the case of a single-proxy PET, a compromised PEE leads to catastrophic results, not only for the forensic investigation procedure but also for the privacy violation of all the users using that PET.

In the case of the Mix-networks, like TOR and Crowds, where they offer unlinkability, all their privacy enhancing entities participated in a circuit/path must co-operate to reveal the identification of the abuser. Even if one of the privacy enhancing entities has been compromised, is not possible to identify the abuser because each PEE knows only its successor and predecessor PEE of the specific circuit. Each PEE is a link in a chain which links an AU with a Server. Even if one link stops functioning, the chain breaks, consequently no link between an AU and a Server exists.

However, if a PET does not offer strong unlinkability (including data anonymity), there is a way to find out the identification of a user, even with an intermediate compromised PEE. The only condition is that the PEE (PEE_entry) that has direct communication with the AU, as well as the PEE (PEE_exit) that has direct communication with the Server are not compromised. The PEE_exit may ask all the PEE if they know the origin of the specific message. The PEE_entry may respond with the identification of the user.

In case of TOR, only the latest PEE (PEE_exit) of the circuit knows the plain message, whereas in Crowds, the weakest data anonymity is provided, due to the reason that all the participated PEE of the path, have access to the plain message.

e. Can a PET present evidence to the Server regarding the involvement of a user?

The Server does not only need to know the identification of the abuser but also needs evidence about the PET's claims. Possibly, the PET desires to protect the abuser, and instead of the abuser's identification, the PET reveals the identification of an innocent user.

Appropriate evidence requires a strong relationship between the user's identity and the malicious message. An appropriate way to link the message with the user is the use of the digital signature. The AU must sign the message before sending it to the PET and the PET must store the signed message for future use in case of an attack. However, the PET must remove the signature of the AU before forwarding the message to the Server because the Server can identify the AU from that signature.

Such a technique has practical problems as the large volume of gathered information is a significant issue. Beyond this issue, each exchanged message will demand more cpu-time to be proceeded. The current PETs do

nothing for gathering evidence since they have not been designed to offer such services to the Server but to the user.

f. Can a PET guarantee the Server that the PET will co-operate in the investigation?

The Server knows that the PET has the responsibility to co-operate only in a case that the PET will not be able to refuse that the message was sent by that PET. If the PET signs the message before sending it to the Server, the Server can be sure that the PET cannot refuse this action.

As it has already been mentioned in [a], none of the current PETs sign the message before sending it to the Server; therefore a PET can refuse that a message has been forwarded through it.

The table 1 summarizes the level that the PETs approach the issues

Table 1 Forensic services offering by the PETs

	TOR	CROWD	ANONMIZER
Can a PET know that a message was sent by itself?	NO	NO	NO
Can a PET decide if a packet is malicious or not?	NO	NO	NO
Can a PET identify the sender (Anonymous User) of a message?	M	L	L
Can a PET identify the sender of a message even with a compromised intermediate PEE?	NO	L	N/A
Can a PET provide evidence to the Server about the involvement of a user?	NO	NO	NO
Can a PET provide a guarantee to the Server that the PET will co-operate in the investigation?	NO	NO	NO
Level of achievement: H= high M=medium L=low NO= not offered N/A=not applicable			

CONCLUSION

Based on the current PETs' design considerations, it is an issue the fact that the PETs offer privacy to the users without any design consideration to discourage attackers to attack through PETs. Thus, PETs' design goal should also include/suggest a proper way to prevent the potential attackers to use maliciously these technologies.

The lack of advanced authentication techniques and non-repudiation actions of the client gives the potential attackers the opportunity to use the PETs and take advantage of these technologies. Besides, an Internet user should have privacy as far as he/she does not violate the rights of others. If, at the end of the day, we (internet society) have the perfect PET based on the current criteria, which does not prevent or discourage an attacker to attack through this technology, we will have bigger problems to solve, like the e-anarchy.

REFERENCES

Anonymizer (2003), URL http://www.anonymizer.com, Accessed at 16th of November 2006

Antoniou, G., Wilson, C. and Geneiatakis, D. (2006) A Forensic Investigation Protocol for Privacy Enhancing Technologies. Proceedings of the 10th IFIP CMS'06 Communications and Multimedia Security Conference, Iraklion, Greece, H. Leitold and E. Markatos (Eds.): CMS 2006, LNCS 4237, pp. 185–195

Antoniou, G. and Gritzalis, S. (2006) RPINA: Network Forensics Protocol Embedding Privacy Enhancing Technologies. Proceedings of the ISCIT 2006 International Symposium on Communications and Information Technologies, A. Taguchi et al. (Eds.), October 2006, Bangkok, Thailand, IEEE Press

Argyrakis, J., Gritzalis, S. and Kioulafas, C. (2003) Privacy Enhancing Technologies: A Review. EGOV 2003, LNCS 2739, pp.282-287, 2003, Springer-Verlag Berlin Heidelberg 2003

Dingledine, R., Mathewson, N. and Syverson, P. (2004) Tor: The Second-Generation Onion Router. In Proceedings of the 13th USENIX Security, Symposium, August 2004

Golle P. (2004) Reputable Mix Networks. Proceedings of the 2004 Workshop on Privacy Enhancing Technologies.

Golle, P. and Juels, A. (2004) Parallel Mixing. October 2004 Proceedings of the 11th ACM conference on Computer and communications security

Gritzalis, S. (2004) Enhancing Web Privacy and Anonymity in the Digital Era. Information Management and Computer Security, Vol.12, No.3, pp.255-288, 2004, Emerald

Luis von, A., Bortz, A., J. Hopper, N. and O'Neill, K. (2006) Selectively Traceable Anonymity. Proceedings of the 2006 Workshop on Privacy Enhancing Technologies

Pfitzmann, A. and Hansen, M. (2006) Anonymity, Unlinkability, Unobservability, Pseudonymity and Identity Management - A Consolidated Proposal for Terminology, URL http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.28.doc, Accessed at 16th of November 2006

Reiter, M., Rubin, A. (1998) Crowds: Anonymity for web transactions. ACM Transactions on Information and System Security (TISSEC), Vol. 1 , Issue 1 (Nov 1998), Pages: 66 - 92

Rennhard, M. and Plattner, B., (2004) Practical anonymity for the masses with morphmix, In Ari Juels, editor, Financial Cryptography. Springer-Verlag, LNCS 3110, 2004.

Shields, C. and Neil Levine, B. (2000) A Protocol for Anonymous Communication Over the Internet. November 2000 Proceedings of the 7th ACM conference on Computer and communications security

Ulf, M., Cottrell, L., Palfrader, P. and Sassaman, L., (2003) Mixmaster Protocol, Version 2. Draft, July 2003, URL http://www.abditum.com/mixmaster-spec.txt, Accessed at 16th of November 2006

COPYRIGHT