PPINA - A Forensic Investigation Protocol for Privacy Enhancing Technologies

Giannakis Antoniou[1], Campbell Wilson[1], and Dimitris Geneiatakis[2]

[1] Faculty of Information Technology, Monash University, Caulfield East Melbourne, 3145 Victoria, Australia
gant2@student.monash.edu.au, cambell.wilson@infotech.monash.edu.au
[2] Dept. of Information and Communication Systems Engineering, University of the Aegean, Karlovassi, Samos, 83200 Greece
dgen@aegean.gr

Abstract. Although privacy is often seen as an essential right for internet users, the provision of anonymity can also provide the ultimate cover for 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. This paper proposes a forensic investigation technique, which can be embedded in the framework of existing PETs , thereby adding network forensic functionality to the PET. This approach introduces a new dimension to the implementation of Privacy Enhancing Technologies, which enhances their viability in the global network environment.

Keywords: Network Forensics, Privacy Enhancing Technologies

1 Introduction

Privacy Enhancing Technologies (PETs) provide an environment for internet users within which connection anonymity [1] can be assured. However, the provision of such anonymity without proper control has the potential to cause chaos within the internet society rather than helping legitimate users. Generally, internet users want to be able to take advantage of the network services offered by the internet without having to necessarily reveal their identity. On the other hand, servers providing such services should also have a mechanism by which the identity of any malicious user (for example a user taking part in an attack on network resources) can be unveiled if necessary, and evidence of such a user's activity provided to the appropriate entity.

The field of network forensics involves the investigation of cyber-crimes, including establishing the identity of internet abusers and gathering evidence of malicious activity for presentation to law courts. A key component in network forensics, which provides strong evidence of identity, is the digital signature. In offline life, the written signature is an important aspect of an agreement between two people. In the digital society, the digital signature is an equivalent way of legally enforcing an agreement between two parties [15]. In the United States

[2–4] and the European Union [4, 5], written signatures and digital signatures have the same legal standing. The techniques we propose in this paper involve the use of the digital signature as a means of identifying users and increasing the level of non-repudiation bestowed on the actions of a user.

Research efforts into PETs and network forensics have essentially opposite respective goals. While PETs attempt to hide the identity of users, network forensics is an area primarily concerned with the revealing of the identity of abusers. The philosophy behind the PPINA (Protect Private Information, Not Abuser) technique presented in this paper involves bridging these two research areas in order to produce a harmonious combination, which can serve both legitimate internet users as well as law enforcement agencies. We have to make it clear at this point that the purpose of the paper is not to introduce a new PET, but to introduce a forensic investigation technique, which can be embedded in a PET framework.

The underlying scenario for our technique is that any user can be anonymous (i.e. protected by the PET) unless the Server requests a forensic investigation entity (FIE) to investigate a particular sequence of packets received by the Anonymous User (AU) through a PET. If the server has enough evidence to prove that somebody has tried to attack the server, then the FIE will further investigate and reveal the identity of the abuser. At the end of this process, a strong body of evidence will be built up concerning the abuser's (non-repudiated) actions. To the authors' knowledge, this is the first time network forensics and privacy enhancing technology have been combined in a single unified framework.

The paper is organized as follows: Section 2 examines the general framework of PETs; Section 3 outlines the motivations of the proposed solution; Section 4 introduces our proposed technique; Section 5 presents a hypothetical case study illustrating the application of our proposed technique and Section 6 concludes the paper.
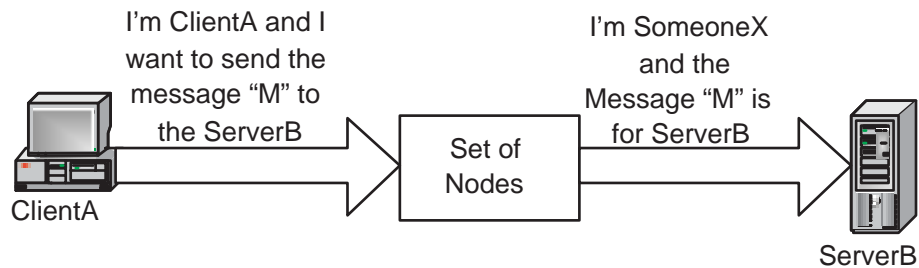
2 The General PET Framework

In the last 2 decades, several PET protocols [6–13] offering anonymity, have been proposed. Most of them do successfully offer privacy (at the network layer) by hiding the users' IP addresses. However, none of them includes techniques for revealing the identity of those users who are abusing the network resources and gathering supporting evidence of such activities.

Although there is a plethora of PETs, most of them have the same framework (figure 1). Each PET protocol is distinguished based on the algorithm used to forward anonymously the messages from the ClientA (Anonymous User) to the ServerB (Server) and back again.

3 Motivations

There is no current technology that offers anonymity to a user and at the same time discourages that user to act maliciously against the server. In particular,

Fig. 1. General PET Framework

none of the related PET technologies offer:

a) Complete non-repudiation: In order for a Server to be able to accuse an abuser, the server needs strong evidence about the action of the abuser. Without such evidence, the abuser cannot be prosecuted. The ultimate evidence in the digital world is the digital signature [15] because it assures an action cannot be repudiated by the abuser.

b) Complete confidentiality and integrity: An internet user wants to be anonymous and only the destination server must read his or her messages. No one else, even the trusted Privacy Enhancing Entities, must be able to access them, i.e. the integrity of the messages must be protected. However, at present, PETs have at least one node that has access to the unencrypted messages and no technique has therefore been employed to offer complete integrity of the messages.

c) Complete authentication: Internet users have recently seen anonymity as an important facet of network communication [14]. At the same time, for many years, malicious intruders have also been looking for such anonymity. Intrusion is an illegal action and an intruder therefore wants to become anonymous (for different reason than an internet user) during his or her activities. Therefore, an intruder can use the PET to hide his or her identity, and the PET helps (unknowingly) the intruder. For this reason, the PET should be sure that the client is the person that it claims to be, before offering anonymity to the client. Although the PET may offer anonymity to the client without the necessity for identifying the client, the PET can become a very good tool for any intruder. Of course, an intruder can also hide his identity illegally, without using a PET framework (for example, by spoofing his or her IP address). However, the PET legitimates in a sense this identity hiding action. For this reason, a mechanism should be applied to identify only the abusers.

These above issues are essential considerations in the design of an appropriate framework, which offers network forensics services in a PET framework. The next section presents the proposed framework design and the communication protocol, which add network forensics services in the PET framework.

4 The *PPINA* FRAMEWORK

The PPINA (Protect Private Information Not Abuser) framework offers a proactive forensic investigation technique that can be embedded in any PET because it is independent of a PET protocol. It adds an end-to-end confidentiality and integrity layer (Issue (b) from our motivations) and forensic investigation service (Issues (a) and (c) from our motivations). In addition, the Server does not have its functionality compromised during the forensic investigation. The Server only needs to contact the Forensic Investigation Entity (FIE) and send the malicious messages. 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 attacker and cannot be repudiated) and the identity of the attacker. We emphasize again that the PPINA protocol operates over a PET protocol and is therefore a general solution, not linked with a specific PET. The following explains the operation of the PPINA protocol. We first introduce the notation used in the explanation.

Notation

A = Anonymous User
B = Directory Service
C = PET
D = Server
$As\{Data\}$ = The Data is signed by the private key of an Anonymous User, where the public key, of that private key, is published
$Ae\{Data\}$ = The Data is encrypted by the public key of an Anonymous User, 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. This public key plays, also, the role of a secret key
$e\{Data\}$ = The Data is encrypted by a secret key (symmetric encryption)
$bKey\{Data\}$ = The Data is encrypted by the bKey (symmetric encryption)
$ForensicReceipt$ = The digest of the received data from the Server

4.1 The Three Phases

The whole communication process can be divided into 3 phases: the Initialization phase (Figure-2, Steps 1-6), the Main phase (Figure-2, Steps 7-10) and the
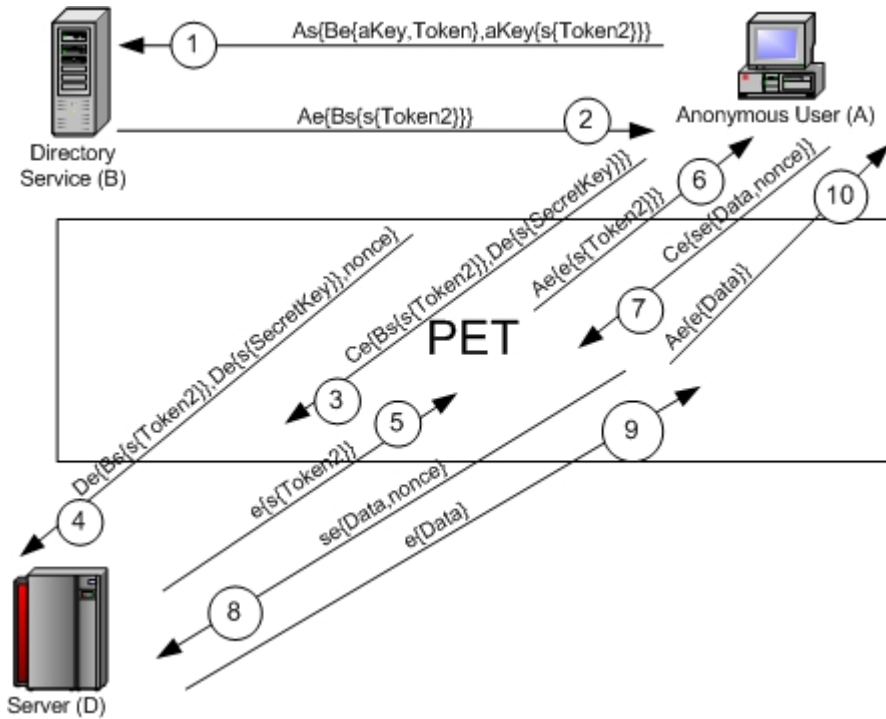
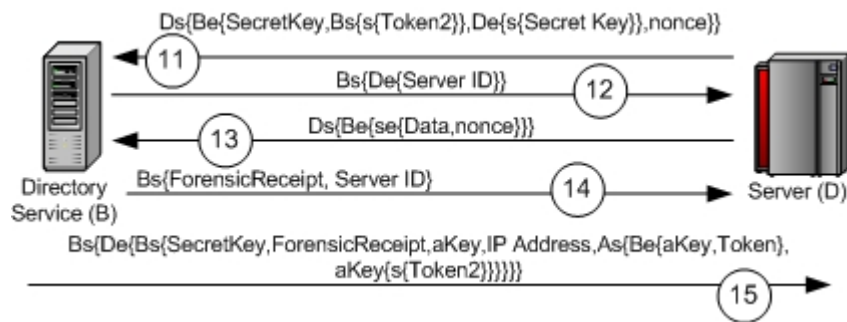**Fig. 2.** PPINA (Initialization and Main Phase)

Fig. 3. PPINA (Forensic Investigation Phase)

Forensic Investigation phase (Figure-3, Steps 11-15).

Initialization Phase
Before the real communication (main phase) begins, the initialization phase is required. The DS gathers the fingerprint [s{Token2}] of the future communication actions of the AU, and the Server validates that fingerprint:

A→B: As{ Be{ aKey, Token}, aKey{ s{ Token2 } } } **(Step 1)**
B→A: Ae{ Bs{ s{ Token2} } } **(Step 2)**
A→C: Ce{ Bs{ s{ Token2} }, De{ s{ Secret Key} } } **(Step 3)**
C→D: De{ Bs{ s{ Token2} }, De{ s{ Secret Key} }, nonce} **(Step 4)**
D→C: e{ s{ Token2} } **(Step 5)**
C→A: Ae{ e{ s{ Token2} } } **(Step 6)**

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

The AU calculates the Token [Token= Hash_Function(Secret Key)] and the Token2 [Token2= Hash_Function(Token)]. The AU signs the Token2 [s{Token2}] by using the private key. In addition, the AU generates a symmetric key (aKey) to encrypt part of the data sent to the DS. The [s{Token2}] is encrypted (Step 1) in order to avoid a possible attack from the Server. The DS is responsible for verifying the validity of Token2, based on the given Token. The DS stores the message (Step 1) in order to prove, in case of a forensic investigation, that the specific AU was going to communicate with a Server by using a secret key, which has the specific Token2. The [Bs{s{Token2}}] is the ticket which is forwarded to the Server through the AU and the PET. The AU also sends [De{s{Secret Key}}] to the Server. The Server is responsible to verify the validity and authenticity of Token2, based on the given secret key. The Server stores the message (Step 4) in order to prove later on (to the Forensic Entity) that the communication has used a secret key with a specific Token2. 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 (Step 1). However, the Server needs to verify that the AU also has the private key by verifying the signature of the secret key [s{Secret Key}] and [s{Token2}] by using the secret key as a public key.

Generating private/public key pairs is a computationally intensive procedure. However, an AU can generate several pairs of keys during times of low system load and request tickets from the DS (one ticket for each pair of key). A ticket is server-independent and time-independent; therefore, a ticket can be used for

future communication with any Server and any time. A ticket is useful only to the entity who knows the related private key.

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 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: Ce{ se{ Data, nonce} } **(Step 7)**
C→D : se{ Data, nonce} **(Step 8)**
D→C : e{ Data} **(Step 9)**
C→A: Ae{ e{ Data} } **(Step 10)**

The AU encrypts, with the public key (secret key), and signs, with the private key, the message [Data, nonce]. Each message (Step 7) contains the Data and the nonce. The nonce is a counter, which helps to avoid a replay attack. If the Server receives an 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 (Step 8) in order to prove, in case of an attack, that the AU who has the private key of the 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 AU (Step 10) through the PET (Step 9).

Forensic Investigation Phase

During this phase, the DS plays the role of the Forensic Investigation Entity (FIE). In case the Server receives inappropriate data from an abuser, the Server informs the appropriate entity (FIE), which will investigate the incident and identify the abuser. The FIE needs evidence of the abuser's action in order to continue the investigation. The Server must provide such evidence. The Server should have saved the communication messages between the PET and Server to prove that the messages came from the specific PET. The Server accuses the PET, until the PET provides evidence that an Anonymous User generated these messages. The FIE should provide evidence about the identity of the Anonymous User.

It is particularly important that in our framework, 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 functioning for days 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: Ds{ Be{ Secret Key, Bs{ s{ Token2} }, De{ s{ Secret Key} }, nonce} } **(Step 11)**
B→D: Bs{ De{ Server ID} } **(Step 12)**
D→B: Ds{ Be{ se{ Data, nonce} } } **(Step 13)**
B→D: Bs{ ForensicReceipt, Server ID} **(Step 14)**
B→D: Bs{ De{ Bs{ Secret Key, ForensicReceipt, aKey, IP Address, As{ Be{ aKey, Token}, aKey{ s{ Token2} } } } } } **(Step 15)**

The Server sends (Step 11) the ticket (generated and signed by the DS) to the FIE including the secret key. 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 Server ID. This identity (Server ID) is the identification of the current investigation case. The Server sends to the FIE (Step 13) all the communication messages for that session received by the AU. After the FIE receives all the necessary messages, it replies (Step 14) 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).

After the FIE concludes that the messages were malicious, the FIE sends (Step 15) the IP Address of the AU and evidence (Step 1) which proves the involvement of the AU. Only an entity who knows the private key could sign the Token2. The Server can now submit [Bs{ Secret Key, ForensicReceipt, aKey, IP Address, As{ Be{ aKey, Token}, aKey{ s{ Token2 } } } }] from (Step 15), and the malicious communicated messages from the AU to the server (Step 13) to the court as evidence of the AU's actions.

We have described the operation of the PPINA protocol. The PPINA technique embeds in a PET framework a number of characteristics, which are described in detail in the next section.

4.2   Characteristics of the PPINA protocol

Provision of Strong Evidence: Since the messages have been signed by the AU, it can be confirmed the actions of the AU/abuser cannot be repudiated by the AU. The digital signature provides the ultimate legal means of evidence verification in the digital era; therefore, no entity is able to doubt about the integrity and the authenticity of the evidence.

Non-stop Server/victim operation: In a classic scenario, during an investigation, the Server/victim needs to be investigated closely by experts in order to gather evidence about the actions of the abuser. During the investigation, the Server/victim is typically not in online fully operational mode. However, in this case, the Server's computer does not need to be investigated, because the DS has the necessary evidence to accuse the abuser.

Related cheap and fast investigation procedure: The PPINA protocol forces every AU to provide evidence to the DS of the AU's future actions before contacting the Server. Therefore, the Server knows that the DS has the necessary evidence, making the investigation not only fast but also cheap.

No privacy violation exists during the investigation: As part of the computer forensic investigation procedure, the Server does not need to make available any storage media (Hard Disk, Tapes, CD-Rom, etc) which took place during the cyber-crime, which may also contains private sensitive information of the Server.

Respect the goal of the underlying PET: The PPINA can be embedded in a PET framework without to affect the level of offered privacy of the PET.

Our proposed framework offers significant advantages. However, it should be acknowledged that the level of encryption involved and the imposition of the authentication layer might decrease the level of performance of the communication between AU and Server. In addition, if the Server somehow compromises the DS, the Server can identify the users who have contacted with that Server.

5 Case Study - AU attacking a Server

In this section, we explain in more detail the specific operation of the PPINA protocol using a simple case study whereby an anonymous user attempts to attack a Server via the PET. Suppose Alice (Anonymous User) wants to communicate with Bob (Server). Firstly, she generates a pair of keys (Public/Private) and then contacts the DS in order to get the necessary ticket [Bs{s{Token2}}], which is mandatory for the communication between Alice and Bob. Before the DS issues the ticket, it verifies that the Token2 is the hash value of the Token. Otherwise, the DS does not issue the ticket. Alice, through the PET, forwards the ticket to Bob. Alice, also, creates [De{s{Secret Key}}] and sends it to Bob. The Secret key (public key) is needed to offer confidentiality of the messages and also to verify the signed messages of Alice. Bob verifies that the Token2 is the hash value of a token, where token is the hash value of the secret key. Also Bob verifies the signature of [s{Token2}] and [s{Secret Key}] with the Secret Key, which here plays the role of the Public Key. If the verifications are valid, Bob encrypts the [s{Token2}] and sends it back to Alice, otherwise Bob terminates the communication. After the Initialization Phase is completed, the Main Phase begins whereby Alice wants to compromise Bob's computer. Alice signs the message with the private key and encrypts it(via symmetric encryption) with the secret key. Bob receives the message and makes the necessary verification, whereby he decrypts the message and verifies the signature with

the secret key. Once Bob realizes the attack, he stops the communication with Alice, contacts the DS and sends the appropriate evidence (all the information received by Alice including the secret key). The DS cannot deny the existence of Alice because the DS has issued the ticket (there is a signature of the DS on the ticket). The DS cannot also accuse an innocent AU (i.e. another anonymous user other then Alice) because Bob has the ability to verify the [s{Token2}] via the secret key. Bob expects to receive, from the DS, a signed message that includes a specific [s{Token2}], which can be verified by the secret key. The DS decrypts and verifies the messages with the secret key and examines the information [se{Data,nonce}]. If the DS detects that the information was malicious, it replies with the evidence [As{Be{aKey, Token}, aKey{s{Token2}}}] and additionally sends the IP Address of the user as well as the aKey. aKey will be used to decrypt the encrypted message in order Bob verifies the signature [s{Token2}] and is therefore sure about the identity of the abuser. It is possible that a compromised DS can send wrong IP Address. However, the user who has signed the message [As{Be{aKey,Token},aKey{s{Token2}}}] is the abuser, because only this AU knows the Private Key (AU signed Token2). The Server now has the strongest evidence to prove the involvement of that particular user in the attack.

6   Conclusion

The provision of privacy and anonymity to internet users can also provide an environment within which malicious users can hide. A key driver of the wider adoption of privacy enhancing technologies would be the ability of forensic entities to gather evidence of malicious activity while legitimate users are still offered anonymity. In this paper, we have provided a framework through which the anonymity of users not engaged in malicious activity is protected while such evidence can be gathered when network abuses occur. The PPINA framework offers non-reputation 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 the cost and the duration of an investigation.

References

1. Andreas Pfitzmann (2005), "Anonymity, Unlinkability, Unobservability, Pseudonymity and Identity Management - A Consolidated Proposal for Terminology"
2. http://slis.cua.edu/ihy/fall01/tpedoc/pl106229.pdf
3. http://www.ntia.doc.gov/ntiahome/ntiageneral/esign/105b/esign7.htm
4. Stephen E. Blythe, Digital Signature Law of the United Nations, European Union, United Kingdom and United Stats: Promotion of Growth in E-Commerce with Enhanced Security, 11 RICH. J.L. & TECH. 2 (2005), at http://law.richmond.edu/jolt/v11i2/article6.pdf

5. http://europa.eu.int/eur-lex/pri/en/oj/dat/2000/l_013/l_01320000119en00120020.pdf

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

7. Anonymizer (2003), available at http://www.anonymizer.com

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

9. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In Proceedings of the 13th USENIX Security, Sympo-sium, August 2004

10. Clay Shields, Brian Neil Levine, "A Protocol for Anonymous Communication Over the Internet", November 2000 Proceedings of the 7th ACM conference on Computer and communications security

11. Philippe Golle, Ari Juels, "Parallel Mixing", October 2004 Proceedings of the 11th ACM conference on Computer and communications security

12. Ulf Moller, Lance Cottrell, Peter Palfrader, and Len Sassaman. "Mixmaster Protocol", Version 2. Draft, July 2003, available at http://www.abditum.com/mixmaster-spec.txt

13. Marc Rennhard, Bernhard Plattner, "Practical anonymity for the masses with morphmix", In Ari Juels, editor, Financial Cryptography. Springer-Verlag, LNCS 3110, 2004.

14. Alessandro Acquisti (2005), "Privacy in Electronic Commerce and the Economics of Immediate Gratification"

15. Patrick W. Brown , "Digital signatures: can they be accepted as legal signatures in EDI?", December 1993, Proceedings of the 1st ACM conference on Computer and communications security