True Anonymity Without Mixes

Carlos Molina–Jiménez and Lindsay Marshall
Department of Computing Science
University of Newcastle upon Tyne
Newcastle upon Tyne, NE1 7RU, UK
{*Carlos.Molina, Lindsay.Marshall*}*@ncl.ac.uk*

Abstract

Anonymity is an essential part of social structures. In the non–electronic world there are several services that are based on anonymous interactions between individuals. The migration of these services to the Internet world is unfeasible without the provision of anonymizers to guarantee anonymity.

Anonymizers based on mix computers interposed between the sender and the receiver of an e-mail message have been used in the Internet for several years by senders of e–mail messages who do not wish to disclose their identity. Unfortunately, the degree of anonymity provided by this paradigm is limited and fragile. First, the messages sent are not truly anonymous but pseudo–anonymous since one of the mixes, at least, always knows the sender's identity. Secondly, the strength of the system to protect the sender's identity depends on the ability and willingness of the mixes to keep the secret. If the mixes fail, the sender's anonymity is compromised.

In this paper, we propose a novel approach for sending truly anonymous messages over the Internet where the anonymous message is sent from a PDA which uses dynamically assigned temporary, non–personal, random IP and MAC addresses. Anonymous e–cash is used to pay for the service.

1. Introduction

Probably the single biggest fallacy about the Internet is that it is anonymous. Most computer users know pretty well how to surf the web and retrieve information, yet what most of them ignore is that while they visit web pages, they can be profiled by web page owners. Because of the design of the Internet, it is relatively easy for web pages' owners to keep log files to gather information about visitors and to convert such information into statistical data. Moreover, web page owners that does not want to know anything about log files can delegate this job to on–line companies like NBC Internet Inc. (NBCi) which in return for your personal data and a share of the collected statistics will monitor your web page for free [14]. Having installed a counter (a reference to NBCi's CGI script) in your web page, NBCi offers you sensitive information like the operating systems and the browsers used by visitors, time of visit, visitors' country, and the full host name of the last 100 visitors. Notice that this information is provided as bar graphics and is ready for use in marketing campaigns and all without the visitor being aware of it.

From the above discussion it follows that it would be desirable for web surfers to visit web pages anonymously. Moreover, there are other applications such as expression of political views, assistance with embarrassing diseases and electronic transactions where users do not wish to disclose their identity.

As a response to this concern, intensive research has been conducted in this direction which has resulted in several published papers suggesting solutions to the problem. An elegant solution to this problem was suggested by Chaum [5]. Unfortunately, this approach is not suitable for Internet applications as it is based on the sending of broadcast messages over a reliable network; consequently, we will not discuss it further. More practical approaches has been proposed and even implemented and made available for Internet users. However, all proposals known to us so far, mixed–oriented proposal for instance, are based on the use a trusted third party located between the sender and the receiver whose job is to blur the direct association between the former and the latter. We argue that anonymizers based on trusted third parties are not a satisfactory answer to the problem. Therefore, in this paper we suggest an innovative and simple approach for sending truly anonymous messages which does not relies on a trusted third party, nor does it uses a home IP address assigned by the home network operator to identify the sending device. In our proposal, the anonymous message is sent from a mobile device identified by a non–personal, temporary, random identifier IP address assigned by the Mobile Support Station (MSS). Anonymous e–cash is used to pay for the call.

As we will show later on, the idea is simple, clear, easy to understand, validated and implemented; surprisingly, probably because of its simplicity, it has been overlooked and has not been studied before.

It is worth noticing that in this paper we assume that Bob, the mobile user, is in possession of a Personal Digital Assistant (PDA), yet Bob's PDA represents any pocket–sized electronic device equipped with computational power and wireless communication facilities, i.e. a cross between a computer and a mobile phone which is called a *Mobile Internet Device* or a *Wireless Internet Device* by some vendors. Moreover, even if in Section 5 we talk specifically of PDAs that comply with the IEEE 802 standard, the main ideas of not using personalized identifiers to provide true anonymity are valid for any other wireless standard in the market.

2. Mobile hosts with and without home IP addresses

Several approaches have been suggested to provide host mobility in the Internet [12, 21, 22, 15, 20]. All of them are grounded on the assumption that when a mobile host, Bob's PDA for example, is away from its home network it still needs to access data and services (personal files, local data bases, local web pages, forward messages, etc.) available only in or through his home network. Because of this, Bob's PDA must be assigned a permanent IP address in its local network which remains constant regardless of its current physical and logical location. Taking into account that the IP address is uniquely assigned to the PDA and assuming that a message received from the PDA has not been meddled with, it is not difficult to see that the source IP address contained in messages coming from Bob's PDA uniquely identifies his device. An undesirable side effect of this is that the IP address can be used by the people Bob exchanges messages with to trace Bob's identity.

2.1. Do mixes provide true anonymity?

So far, all mechanisms known to us for providing anonymity on the Internet rely on the use of a trusted third party. A trusted third party is a computer (or a set of them) located between the sender and the receiver of the anonymous message whose job is to blur the association between the sender and the receiver. For example, to conceal the sender's identity, the trusted third party replaces the sender's name and IP–address found on the head of the message with its own name an address, respectively; the result of this is that at the receiver's end the message appears as coming from the trusted third party rather than from the original sender. In the simplest schemes, the rôle of the trusted third party is played by a single computer, that is, only one computer is placed between the sender and the receiver (see [3], for example). In an attempt to decrease the dependence on a single trusted third party, Chaum introduced the concept of *mixes* in the early 80s [6]. Basically, the idea behind Chaum's proposal is that instead of relying on a single trusted third party, we rely on a set of trusted third parties which cooperate in blurring the association between the sender and the receiver. Each of the trusted third party is called a mix. On this account, if Bob wishes to send and anonymous message to Alice, Bob delivers his message to the first mix in the set, then Bob's message is bounced from one mix into another until it is eventually delivered to Alice. Of course, the algorithm assumes that mix j that receives the message from mix i and forwards it to mix k, does not tell mix k where the message came from. The result of this is that Bob's identity can be disclosed only by subversion or conspiracy of all the mixes in the set. Practical examples of mix–based anonymizers are discussed in [17, 10].

A serious flaw of mix–based anonymizers is that their degree of anonymity is limited since there are no means of hiding the IP address of the sender; one of the mixes, at least, will always know it. The problem here is that the sender is a computer with a personal and permanent IP address which can lead to the identity of the owner of the computer. Trying to send an anonymous message from a computer with a personal and permanent IP address is analogous to trying to make an anonymous call from a home telephone line by using the Calling Line Identification Blocking service (the 141 number in the UK). The calling number is hidden from the receiver by preceding the dialed number with the digits 141, but it is not hidden from the carrier, nor from anybody who has the means of persuading the carrier to disclose it nor from a miscreant with enough knowledge and resources to break the carrier's computer where the number is stored. To put it simply, the anonymous messages sent using mix–based anonymizers are not anonymous but only pseudo–anonymous.

Anonymity offered by mix–based anonymizers is fragile, i.e. it is still breakable. This is a consequence of relying on a trusted third party. Like the name suggests, a trusted third party is somebody whom we do not know and whom we have to trust. That is, we have to trust that by no means the computer (or computers) in the middle will not disclose any information that could lead to identify the sender; in Chaum's proposal, we have to trust the mixes. Certainly, it might sound difficult to think that all mixes in the set collude against Bob. Yet it is conceivable that all of them are seized or monitored by government's agents equipped with enough resources for breaking the keys used by the mixes and discover the IP address of Bob's PDA and then Bob's identity. Recall that at least one of the mixes knows this IP address. Likewise, it in certainly feasible that the administrator of each of the mixes receives a court order or a threat to disclose the all the information to identify Bob.

It is true that, from the point of view of security, the strength of a mix–based anonymizer increases as the number of mixes in the set increases, however, this makes the whole system more complicated. There are several issues involved here, we will name only the most relevant. For example, the length of the original message posted by Bob and the transmission time increases proportionally to the number of mixes. Some of the mixes might demand direct payment from Bob. The system is more prone to failure, thus, some mixes might fail, be unreachable, or refuse to cooperate while Bob is sending his message or waiting for Alice's answer.

To summarize, regardless of the number of mixes involved, the hole that one of the mixes, at least, knows the sender's identity is not fixed, that is, incrementing the number of mixes does not lead to complete anonymity.

2.2. Lack of anonymity in postpaid communication services

One of the attractions of having a mobile device is that Bob can freely travel all over the world (where coverage for his PDA is provided) enjoying continuity of his communication service, sending and receiving messages and being billed by a single bill at home at the end of the month and on a postpaid service basis.

For this to be possible, regardless of its current geographical location and before full access to the Internet resources is granted, Bob's PDA must contact its home network and authenticate itself so that his home network operator can collect all charges that Bob incurs.

For the sake of authentication, Bob's PDA can use any hardware (the serial number of his CPU for example) or software (an assigned number, for example) identifier previously agreed upon with the operator of its home network, as long as this identifier unmistakenly leads to Bob's home account. A side effect of this scheme is that Bob's identifier can readily be used by the operator of his home network, to find out all about Bob's whereabouts (his geographical location, services used, and so on).

2.3. True anonymity in prepaid communication services

The arguments outlined in in Section 2.2 about the need of a home IP address are certainly justifiable, however, there are situations where Bob might want to access Internet services under strict anonymity; i.e without disclosing his identity neither to Alice (the receiver of his message) nor to the operator of his home network or to the operators of the communication infrastructure located between him and Alice. If this is the case, the use of a PDA identified to its home network by an IP address is certainly not suitable and not necessary as long as the PDA accesses Internet services that do not require support from its home network; examples of such services are reading local news, posting of messages to electronic lists, e–mailing anonymous messages and so on.

Probably the simplest way for accessing Internet services under complete anonymity is the use of a prepaid PDA. Although they are not yet in the market, a prepaid PDA would work as a prepaid GSM phone does: Bob would get his prepaid PDA in the supermarket and without the need to sign any contract at all with the communicator provider he would load it with a certain amount of money from which the cost of his calls is deducted. When his prepaid credit runs out he would recharge his device by purchasing a top–up anonymous scratch lottery–like card similar to the ones used by current prepaid phones [23, 7, 16]. Since Bob does not need to give away any personal data to buy his device and since he can recharge it anonymously, a prepaid PDA can be used for accessing Internet services under complete anonymity. It can be used for example for sending truly anonymous messages.

2.4. Anonymous and non–anonymous Internet access

From the discussion presented in Sections 2.2 and 2.3 one can conclude that the provision of either non–anonymous and anonymous Internet access is a simple question as long as Bob uses two PDAs: one for each service. Although PDAs are pocket–sized, light, and easy to carry, using different PDAs for accessing different services in neither optimal nor practical. The question that immediately arises here is whether it would be possible to have true anonymity from a postpaid PDA. In other words, can Bob use his IP addressed PDA for sending true anonymous messages? Fortunately, the answer is yes. The crucial idea here is to make Bob's PDA communicate with the MSS in two different modes: non–anonymous and anonymous.

In non–anonymous mode the PDA communicates with its MSS in the traditional way, i.e. it uses its home IP address and authenticates to its home network before being accepted by its current network. Whenever Bob wishes to protect his identity, he switches his PDA into anonymous mode.

In anonymous mode Bob's PDA does not use its IP home address, neither does it authenticate to its home network; it does not need to contact its home network at all; it might contact it (under the cover of its anonymous hood) but only if Bob wishes to and its home network accepts anonymous visitors.

To communicate with its current MSS in anonymous mode, Bob's PDA uses a non–personal, temporary, random IP address assigned by the MSS on a per–communication session basis. It is of great importance to notice that for the functionality of our protocol (discussed in Section 4) this IP address is just an identifier that serves to identify Bob's PDA within the confines of the MSS, for this reason we call it a *TmpId*. The nature of this TmpId is irrelevant for our protocol. It can certainly have an standard IP address format so that it functions like a care–of–address in the IP mobile protocol [15] or like a reusable address in the Network Address Translator protocol [9]. Likewise, how this TmpId is generated and administered by the MSS is not relevant to our protocol. For example, it can be assigned by means of the Dynamic Host Configuration Protocol [8].

The use of a TmpId to identify Bob's PDA is what makes our approach strong enough to resist attacks aimed at wiretapping the communication links between Bob and Alice, and at breaking or seizing the MSS. Although the MSS serves as a relay between Bob and Alice, it knows nothing about Bob's identity. In contrast with mix–oriented anonymizers, in our approach, Bob's identity is never transmitted to the MSS. Because of this, the communication lines and the MSS will not reveal Bob's identity even if they are taken under the control of Ebe the meddler.

At this stage one can ask what motivations would a MSS have to provide anonymous communication services to Bob. The answer is money. The MSS will not be bothered about Bob's identity as long as he or somebody else pays for the service. This issue is discussed in the next section.

3. Anonymous payment for the communication service

The method of payment for the anonymous communication services offered by the MSS play an important part in the algorithm for anonymous communication presented in Section 4. However, since this issue is not intrinsic to the algorithm we will discuss it separately.

There are two possible cases Bob can be faced with when he comes to the MSS to request an anonymous communication service. First, the MSS can offer free anonymous communication services to Bob. This means that somebody else is paying for Bob's anonymous call. For example, the government or the called party (in return for commercial advertisements) is paying for it. Why, how and whoever pays the MSS for the service is irrelevant to our algorithm.

Secondly, following the general rule which states that the calling party pays, the MSS can charge Bob on–line on a pay–for–time–used basis. If this is the case, the MSS must support a mechanism for anonymous payment. Bob can use an anonymous prepaid card (see Section 2.3). Alternatively, Bob can use anonymous e–cash [4, 18].

Since the use of anonymous e–cash to pay for the anonymous communication service is the most general case and because Bob might need an anonymous method of payment to pay for other services apart from the MSS's, we consider the use of anonymous e–cash in our approach presented in the next section and originally introduced in [13].

4. Protocol for sending true anonymous messages

In our system shown in Fig. 1, Bob is a PDA user wishing to send an anonymous message; Alice is the recipient of the anonymous message on her work station (WS); and Doug is the owner of the MSS and offers communication services on a pay–for–time–used basis. Clare is a bank owner and offers support for anonymous e–cash payments to her account holders (Bob and Doug). Finally, Ebe is another PDA user, an evil one.

For the description of our protocol we will adopt the widely accepted BAN–style notation (see [2]). Thus, here, and throughout, A, B, C and D represent Alice, Bob, Clare, Doug and Ebe respectively. K_{BD} is a secret key shared or intended to be shared by Bob and Doug; K_D^{pu} and K_D^{pr} are Doug's public and private keys, respectively. The notation $\{M\}$ indicates the string M in plain text; while the notation $\{M\}_K$ indicates the string M encrypted using the key K. Finally, a right arrow indicates the operation $sends$, thus $B \rightarrow A : \{M\}$ indicates that B sends to A the message $\{M\}$.

The protocol for sending an anonymous message consists of the following steps:

1. Bob turns on his PDA and learns K_D^{pu} from Doug by listening to the MSS advertisement:
   $D \rightarrow B : \{K_D^{pu}\}$
2. Bob creates K_{BD}, encrypts it using K_D^{pu}, and sends it to Doug for approval. He waits t units of time for a reply:
   $B \rightarrow D : \{K_{BD}\}_{K_D^{pu}}$
3. Doug checks that the K_{BD} suggested by Bob is correct and not in use. If so, the MSS creates a TmpId for Bob's PDA, encrypts it using K_{BD}, and sends it to Bob as a reply. If the K_{BD} suggested by Bob is incorrect, Doug does not reply. If it is correct but has been assigned to an existing user, Doug does not reply to Bob and additionally asks the user of the existing K_{BD} to renew it (for simplicity, this operation is not shown in the BAN–notation). After t units of silence, Bob can try again. The approved K_{BD} is used then to encrypt and decrypt messages between the PDA and the MSS until either the end of the session or until it has to be renewed. Messages encrypted with K_{BD} can

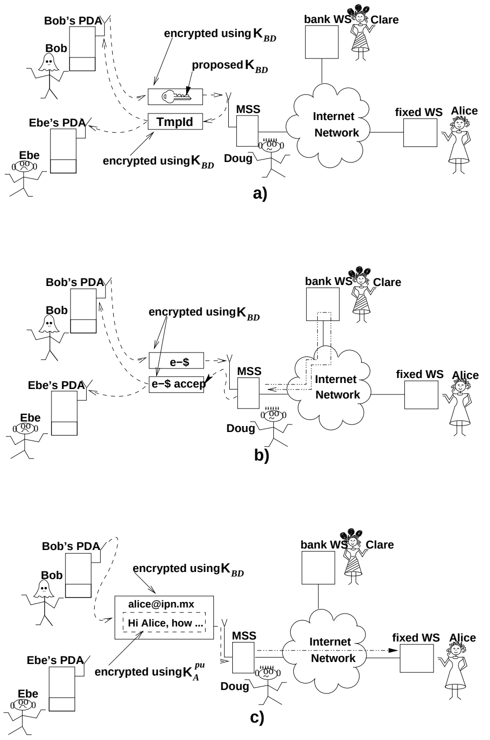


Figure 1. Anonymous and confidential call from a PDA

be overheard by Ebe but he will not be able to decrypt them:
$D \rightarrow B : \{TmpId\}_{K_{BD}}$

4. Bob sends an anonymous e–coin to Doug to pay for the communication session. Before accepting or refusing the coin as a payment, Doug consults Clare; thus the coin is forwarded to her in a message encrypted using Clare's public key:
$B \rightarrow D : \{e - \$\}_{K_{BD}}$
$D \rightarrow C : \{e - \$\}_{K_C^{pu}}$

5. If Bob wishes to anonymously e–mail Alice, he encrypts the message body using Alice's K_A^{pu}, appends the encrypted text to Alice's address, encrypts the result using K_{BD}, and sends it to Doug:
$B \rightarrow D : \{alice@ipn.mx \mid \{Hi, A...\}_{K_A^{pu}}\}_{K_{BD}}$

6. Doug decrypts the message and forwards to Alice the enclosed and encrypted message body together with Bob's TmpId:
$D \rightarrow A : \{\{Hi, A...\}_{K_A^{pu}} \mid TmpId\}$

7. Alice has no means to discover the identity of the TmpId holder. Yet she can reply to Bob by addressing her response to the MSS and including Bob's TmpId. Upon receiving Alice's reply, Doug forwards the body message and its headers to the PDA user with the TmpId, that is, to Bob. In the notation below, the header of Alice's reply are not shown, :
$A \rightarrow D : \{Hi, anonymous friend... \mid TmpId\}$
$D \rightarrow B : \{Hi, anonymous friend...\}_{K_{BD}}$

8. Bob's session ends when he turns off his PDA, leaves his current MSS, or his MSS times-out his session.

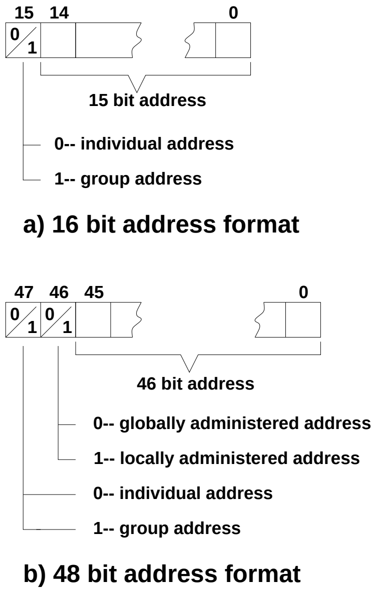
Notice that in the above protocol Alice's is sent in plain text; consequently, Doug can read it. To prevent Doug from reading Alice's reply, Bob can create and append a session key for him and Alice (K_{BA}) to Alice's message body and ask Alice to encrypt the message body of her reply using this key. A comment about confidentiality and anonymity at Bob's side is in order. It can be argued that, to protect Bob's identity, it is not strictly necessary to encrypt Bob's anonymous message and Alice's reply. The two messages can be transmitted in plain text without disclosing Bob's identity. If Ebe reads them, he will learn that an anonymous PDA user identified with a TmpId is exchanging e–mails with Alice, yet Ebe does not know the identity of such PDA user. In our protocol, we encrypt both, Bob's anonymous message and Alice's reply, to make sure that an analysis of the plain texts does not give Ebe, a piece of evidence about Bob's identity.

5. Can the MAC address reveal Bob's identity?

When communication takes place among multiple stations on a common communication channel (an Ethernet cable, a radio frequency channel, etc.) there is a need to identify both the sender and the receiver uniquely. In LANs that comply with the IEEE 802 standards (see [1] for example), a station is identified by a string of either 16 or 48 bits assigned to its network interface controller (NIC) and called the Media Access Control address (MAC address for short).

16 bit MAC addresses can be administered locally only, conversely, 48 bit MAC addresses can be administered either locally or globally. The main difference between local and global MAC address administration is that in the first case the addresses assigned to a NIC are determined by the administrator of the LAN, who is responsible for guaranteeing that no two NICs in his LAN have the same MAC address at the same time. On the other hand, globally administered addresses are assigned by the IEEE standardization body in coordination with the NIC's manufacturer. As their name implies, globally administered MAC addresses are globally unique. This means that no two NICs in the world can have the same MAC address.

The 802 standard does not encourage any specific implementation of the MAC address, however, in practice this



a) 16 bit address format

b) 48 bit address format

Figure 2. Address field format of a IEEE 802 standard.

address is normally stored in what is called the unicast address register of the NIC. For locally administered addresses the value of the 46th bit in the unicast address register is 1. Conversely, for globally administered addresses, this bit is set to 0 (see Fig. 2–b). At any time the actual MAC address of a NIC is determined by the contents of this register which can be read and written by the device driver software. Normally, at initialization time, the device driver software loads the unicast address register with a default value. Since 16 bit MAC addresses are always administered locally, the default value is always read from a set of switches configured manually by the LAN administrator. Similarly, the default value for a 48 bit MAC address administered locally is read from a set of switches configured manually by the LAN administrator. However, if the 48 bit MAC address is administered globally, the default value is read from a ROM chip embedded in the NIC.

During initialization, the default value of the MAC address can be ignored by the device driver software, so that the unicast address register can be loaded with a different value determined by the LAN administrator and under his responsibility [19]. Also, the contents of this register can be changed at any time (see Fig. 3).

The use of globally administered MAC addresses significantly reduces the LAN administrator's work. To initialize a NIC with a globally–unique 48 bit MAC address, it is enough to load the unicast address register of the NIC with the value read from the ROM chip.

The IEEE 802 standardization body is in charge of administering the 2^{48} address space. Upon request, it issues to NIC manufacturers what is called an Organization–Unique Identifier (OUI) which is a string of 24 bits to be used for the left 24 bits of the MAC address wired in the ROM of all NICs manufactured by him. The right 24 bits are assigned by the manufacturer.

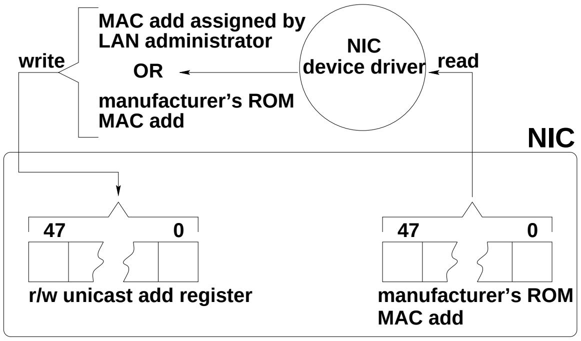


Figure 3. Initialization of the unicast address register.

5.1. Tracing Bob through the MAC address of his PDA

A side effect of the scheme used for assigning global MAC addresses is that by looking at the left 24 bits of the source address in a MAC frame, the receiver of the frame can always learn the name of the manufacturer of NIC being used by the sender's computer.

In the algorithm presented in Section 4 we assumed that messages exchanged between the MSS and its PDA are sent to a broadcast address, consequently, no MAC addresses were involved in the communication. However, it might be the case that Bob's PDA is required to use a MAC address to communicate with the MSS. If this is true, as stated above, the use of a globally administered MAC address would give the MSS the opportunity to trace Bob's identity with the help of the NIC's manufacturer and the shop assistant who sold to Bob the PDA. This would be certainly possible if the latter keeps records that link Bob to the PDA he bought.

A possible solution to this problem would be for the PDA to avoid using a globally unique MAC address when sending anonymous messages; for this purpose it can use a locally administered MAC address.

5.2. Random MAC addresses

Locally administered MAC addresses are assigned to NICs by the LAN administrator. As stated in Section 5, at run time what really matters is the value stored in the unicast address register of the NIC. If the LAN administrator decides to use locally administered MAC addresses, this value can be changed by him at will as long as it complies with the format presented in Fig. 2 and each NIC has a unique address within the extend of his LAN.

On this basis, to guarantee that Bob's identity is not traced by Doug (the owner of the MSS) after reading the MAC address of the messages sent from Bob's PDA to the MSS, Bob can load the unicast address register of his PDA with a temporary, non–personal, random MAC (TmpMAC) address.

To assign a unique (within the confines of Doug's MSS) TmpMAC address to Bob's PDA we can use the main ideas of the algorithm presented in Section 4 for negotiating a unique TmpId. The algorithm is basically the same, thus we will not present it step–by–step in BAN–style notation. When Bob turns on his PDA, the latter receives the public key of Doug's MSS at its broadcast address. Next, Bob generates a TmpMAC address and a K_{BD} key; he concatenates them, encrypts the result using Doug's public key and sends the message $\{TmpMAC \mid K_{BD}\}_{K_D^{pu}}$ to the MSS. Upon receiving this message, the MSS checks that the suggested TmpMAC is acceptable. The answer of the MSS (*accepted* or *refused*) is encrypted using K_{BD} and broadcast to the air. Bob's PDA receives the message at his broadcast address and decrypts it. If the answer received is *refused* he tries again. Otherwise he loads the unicast address register of his PDA (see Section 5) with the TmpMAC address and engages in an anonymous session. He can be confident that the NIC of his PDA is not revealing anything about his identity. Needless to say, the TmpMAC address is valid only for the duration of one session.

6. Attacks, defences and further research

In this section, we discuss the attacks to which our system might appear vulnerable and suggest possible defences. It is not in our ambitions to discuss the proposed solutions at large in this paper. Our sole intention is to raise the issue and to point to the future research directions we are currently exploring.

6.1. Traffic analysis

The main contribution of our work is that we present an extremely simple scheme for providing sender's anonymity. It is not our intention to provide receiver's anonymity. This means that in our scheme, Doug knows that, at certain time, Alice received an anonymous message. Likewise, we do not make any attempt to provide connection anonymity. This means that in our scheme, Alice and Doug know the communication path that the anonymous message travelled through before reaching her.

We are aware that the information about the receiver's identity and the communication path can be used by Doug, Alice and by anybody else (Ebe for example) with means to gain access to this information. The issue here is about traffic analysis. Let us assume that Doug is honest but his MSS is seized by Ebe. If this happens, Ebe can know that Alice, let us say the counsellor at the National Drug Centre, is being consulted by somebody located not far from Doug's MSS. This information reveals that somebody who roams around Doug's MSS is concerned about drug addiction. If the number of possible PDA users around Doug's MSS is small, Ebe have good chances to infer who the anonymous caller is.

The origin of the problem we are discussing is the association between the final forwarder of a message and its final destination (in this case Doug and Alice respectively). If the delivery of the message is guaranteed final forwarder always knows to whom he delivers the anonymous message. It is a well known fact that in practice it is extremely difficult to break the association between the final forwarder of a message and the final destination. The only mean known to us for delivering a message to an anonymous receiver is broadcast. Unfortunately, in terms of network traffic, broadcast is unbearable expensive when the ratio of the intended receivers to the total number of receivers is close to zero. Besides, broadcast is unreliable, there is not guarantee that Alice will receive the anonymous message. We admit that from the point of view of traffic analysis, our system is still open to improvement. Perhaps, it can be strengthened by using an ad–hoc network of wireless mixes at Bob's side so that he does not deliver his anonymous message directly to the MSS. Similarly, it might be helpful to use a set of mixes between the MSS and Alice. However, we do not know yet whether it is worth sacrificing the simplicity of the original system in return for additional strength. We are acutely aware that adding mixes will only help, that is, for our ambitions it is not a complete solution, since, as discussed in Section 2.1, the security offered by mixes is not complete.

6.2. What if you are caught red–handed

In the system we are presenting in this work, there is the threat that Bob (the anonymous sender) is caught red–handed, that is, with his PDA loaded with its TmpId and TmpMAC, by Alice (the receiver) and forced to reveal the content of the memory of his PDA. The catch might happen when Bob is in anonymous session or shortly after, that is, before Bob clear up his PDA. If Bob is caught red–handed, he will find it hard to disavow that he is the author of the anonymous messages received by Alice. This kind of social engineering attack is hard to prevent completely, simple because for the anonymous message to be received by Alice, Bob has to send it at some point. Being caught red–handed sending an anonymous message from a PDA at a MSS is equivalent to being caught making an anonymous call at a public phone box, or posting an anonymous letter at a post office. There is no protection against it. In all the three cases, the best Bob can do is to minimise the time of performing his anonymous job as much as possible so that the chances of being caught are minimise as well.

In our system, Bob can greatly minimise the chance of being caught red–handed if he sends the anonymous message to Alice and immediately after that he clears up the memory of his PDA so that no traces (TmpId, TmpMAC, keys, etc.) of his anonymous activities are left. To guarantee safety, he should erase all evidences from his PDA so that no file–recovery software can compromise him. Since we are talking about high levels of paranoia, it is worth noting that sending files to the bin is not enough, every single bit of compromising information must be destroyed properly (see [18], p. 228).

The limitation of this solution is that it works only for one–way communication, that is, it prevents Bob from receiving a reply from Alice. A possible solution to this problem is to receive Alice's reply off–line. Together with the anonymous message sent to Alice, Bob can send two more pieces of information: a secret key to encrypt the reply and a public Web address to place it. The basic idea is that Alice encrypts her reply and places it in an agreed upon public place so that Bob can collect it anonymously at some point in the future. Naturally, it is assumed the existence of a public Web site that accepts anonymous messages left for future collection and that such site accepts anonymous visitors. When Bob suspects that Alice's answers is waiting for him, he can anonymously connect to the Web site, download the encrypted reply, go home, load from a floppy disk the proper key and decrypt Alice's reply. Again, the risk of being caught red–handed while collecting Alice's reply can be minimise by enlarging the period of time elapsed between the sending of the anonymous message to Alice and the collection of the reply.

6.3. Hand–off of anonymous senders

The chances of catching the anonymous sender red–handed can be minimise by enhancing our protocol with support for hand–off of anonymous senders so that Bob is not forced to initiate and terminate his anonymous session at the same MSS, that is, at the same geographical location. This possibility involves issues of user location, routing of messages; and keeping Bob's TmpId, cryptographic keys and payment consistent when he migrates from one MSS into another. Host mobility in the Internet is a well researched topic. As a result, several protocols have been proposed. What is not clear yet is how the proposed protocols support mobile users' anonymity. It would be interesting to investigate how our anonymizer works together with protocols like [15, 20].

7. Conclusions

In this work, we argued that anonymizers based on mix computers and similar systems that rely on a third party interposed between the sender and the receiver cannot provide true anonymity; moreover, they provide only fragile anonymity. To address this issue, we presented a new approach for sending truly anonymous and confidential messages from a PDA served by an MSS. In our paradigm we send anonymous messages from a PDA which is not identified to its home network by a permanent home IP address but by a random, temporary, non–personal dynamically assigned identifier. Similarly, it uses a random, temporary, non–personal dynamically MAC address instead of the global one embedded in its NIC. Anonymous e–cash is used to pay the MSS for the communication service. To prove the feasibility the proposal and its correctness, the protocol was specified in Promela specification language, and its basic safety properties and proper end–states were validated using the Spin validator[11].

Aside from its obvious advantages, anonymity has several serious and negative side effects that make its deployment in the Internet a controversial issue. There are strong arguments for and against it. We believe that before saying that anonymity is good or bad, legal or illegal, we have to bring it into practice and test it rather than blindly approve or banish it.

8. Acknowledgement

Work funded in part by UK EPSRC grants GR/M94168 and GR/N35953. Part of this research was conducted when the first author was a PhD student with a grant from the National Autonomous University of Mexico. Thanks to Prof. Santosh Shrivastava for comments and support. Thanks to the pseudo–anonymous referees for revising the original draft of this paper.

References

[1] ISO/IEC 8802–3. ANSI/IEEE Std 802.3. *International Standard. Information Technology–Local and metropolitan area networks. Part 3: Carrier sense with multiple access with collision detection CSMA/CD access method and physical layer specification.* IEEE, fourth edition, 1993.

[2] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1), Jan. 1996.

[3] Inc. Anonymizer. Anonymizer. http://www.anonymizer.com/main.html, Apr. 1998.

[4] N. Asokan, P. A. Janson, M. Steiner, and M. Waidner. The state of the art in electronic payment systems. *Computer*, 30(9), Sept. 1997.

[5] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1(1), 1988.

[6] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), 1981.

[7] M. Collins. Telecommunications crime —part 2. *Computer & Security*, 18(8), 1999.

[8] D. E. Comer. *Internetworking with TCP/IP. Principles, Protocols, and Architecture*, volume 1. PRENTICE HALL, third edition, 1995.

[9] K. Egevang and P. Francis. The IP network address translator (NAT). RFC 1631, The Internet Engineering Task Force, May 1994. available at: http://www.rfc-editor.org/.

[10] D. Goldschlag, M. Reed, and P. Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2), Feb. 1999.

[11] G. J. Holzmann. The model checker Spin. *IEEE Transactions on Software Engineering*, 23(5), May 1997. also available at: http://netlib.bell-labs.com/netlib/spin/whatispin.html.

[12] J. Ioannidis, D. Duchamp, and J. G. Q. Maguire. IP-based protocols for mobile internetworking. In *SIGCOM'91 Conference. Communications Architecture and Protocols*, Zurich, Swizerland, Sept. 3-6 1991. ACM.

[13] C. Molina-Jiménez and L. Marshall. Anonymous and confidential communications from an IP addressless computer. In *Handheld and Ubiquitous Computing, Proceedings of the First International Symposium, HUC'99*, Karlsruhe, Germany, Sep 1999. Springer. Lecture Notes in Computer Science, 1707.

[14] INBC Internet. Xoom counter. http://counter.xoom.com/, July 2000.

[15] C. Perkins. IP mobility support. RFC 2002, The Internet Engineering Task Force, Oct. 1996. available at: http://www.rfc-editor.org/.

[16] S. M. Redl, M. K. Weber, and M. W. Oliphant. *GSM and Personal Communication Handbook*. Mobile Communications Series. Artech House, 1998.

[17] M. K. Reiter and A. D. Rubin. Anonymous web transactions with Crowds. *Communications of the ACM*, 42(2), Feb. 1999.

[18] B. Schneier. *Applied Cryptography*. John Wiley & Sons, Inc., second edition, 1996.

[19] R. Seifert. *The Switch Book. The complete Guide to LAN Switching Technology*. Jonh Wiley & Sons, Inc., 2000.

[20] A. C. Snoeren and H. Balakrishnan. An end–to–end approach to host mobility. In *Proceedings of the ACM MOBICOM 2000: The Sixth Annual International Conference on Mobile Computing and Networking*, August 6–11, Boston Massachusetts, 2000.

[21] F. Teraoka, Y. Yokote, and M. Tokoro. A network architecture providing host migration transparency. In *SIGCOM'91 Conference. Communications Architecture and Protocols*, Zurich, Swizerland, Sept. 3-6 1991. ACM.

[22] H. Wada, T. Yozawa, T. Ohnishi, and Y. Tanaka. Mobile computing environment based on internet packet forwarding. In *Proceeding of Winter Usenix*, San Diego CA, Jan. 25-29 1993. USENIX Association.

[23] Yi-Bing, M.-F. Chang, and H. C.-H. Rao. Mobile prepaid phone services. *IEEE Personal Communications*, 7(3), Jan. 2000.