Protocols using Anonymous Connections: Mobile Applications

Michael G. Reed, Paul F. Syverson, and David M. Goldschlag [*]
Naval Research Laboratory

Abstract. This paper describes security protocols that use anonymous channels as primitive, much in the way that key distribution protocols take encryption as primitive. This abstraction allows us to focus on high level anonymity goals of these protocols much as abstracting away from encryption clarifies and emphasizes high level security goals of key distribution protocols. The contributions of this paper are (1) a notation for describing such protocols, and (2) two protocols for location protected communication over a public infrastructure.

1 Introduction

As mobile devices for communication and computation gain more widespread acceptance, where a person is located when processing digital information or sending and receiving messages or phone calls is increasingly under individual control. Relatedly, individuals no longer tied to an office have enjoyed increasing privacy over their location information. If one can conduct business from anywhere, then one can be anywhere when conducting business. However, this is not an entirely accurate picture. For example, mobile phones may not reveal one's location to the party at the other end of the line as readily as stationary ones, but currently implemented technology still requires tracking of the mobile phone itself.

The primary purpose of phones is to allow individuals to communicate. Where anyone happens to be, and even who they are, is simply coincidental to that communication. Technology that more precisely reflects the functional needs of the intended application would therefore provide anonymous channels of communication. The communication over such a channel need not be anonymous; parties typically will identify themselves over the channel, but the channel itself should not reveal their locations or identities to the network or observers of the network. This paper describes protocols that use anonymous channels as primitives. After sketching the requirements for a channel to be anonymous we use such channels to construct protocols for location protected mobile applications. One such application we have already mentioned. Specifically, our protocol allows a mobile phone to send and receive calls without revealing its location to

[*] Address: Naval Research Laboratory, Center For High Assurance Computer Systems, Washington, D.C. 20375-5337, USA, phone: +1 202.767.2389, fax: +1 202.404.7942, e-mail: {*last name*}@itd.nrl.navy.mil.

anyone, including the communications infrastructure on which it relies. A side benefit of our protocol is that its implementation would potentially extend the useful battery life of a phone in standby (listening) mode by orders of magnitude.

Another, less well known but increasingly important, application of mobile communication is in location tracking. This has already been implemented in the Lo-Jack system that allows police to track a car that has been reported stolen and in an active badge system implemented by Olivetti at the Computing Laboratory at Cambridge University. It is also an important component of ITS (intelligent transport systems, cf., below). As an example we describe active badge systems. Like current mobile phones, these do not fully protect location information. In fact quite the opposite. The purpose of such a system is to track the location of those wearing the badges. This can be useful, e.g., in an environment where individuals are not always in their offices but it is important to be able to find them when needed. While useful, active badges can have overtones of big brother, as they allow a company to track things such as how long an individual is in the cafeteria. One way to reduce this threat is to give individuals control over their location information. Such concerns have led to the proposal of protocols for doing just that [7]. These protocols allow individuals to keep their location information in a designated repository over which they have access control. Not even the tracking system is able to determine where an individual is without consulting the designated repository. This paper also presents protocols for individual control of location information in a tracking system. Our approach is related to that of Jackson in [7] but has important differences. One difference is that all versions proposed in [7] require the badge to produce or carry route information for each message it sends. Our approach requires only the production of an adequately random string of one-time identifiers. This means that the Jackson approach decentralizes the control of location information, which appears to be good, and makes the badge operation more complicated, which appears to be bad. Our approach allows for simple badges but requires a centralized database. This centralization might appear to be a vulnerability, but we shall see that it is not.

Intelligent transport systems are designed to track the movement of vehicles on appropriately structured public highways. Some of the advantages of such a system include route optimization for individual vehicles, traffic control for all vehicles so enabled, and traffic signal control for emergency vehicles. However, there is great potential for abuse in such a system [8]. In addition to civil rights and privacy abuses similar to the problems described above for badging systems, the potential exists for other, perhaps more serious threats. For example, truck hijackers could make use of a system that tracks the movement of a fleet of trucks to optimize their chances for a successful and lucrative hijacking. Kidnappers, terrorists, murderers, etc. could trace someone, even if, e.g., she intentionally took varying routes to work each day.

While the private location tracking protocol described in this paper is explained in terms of the active badging example, it is a general protocol for private location tracking using a public infrastructure. Thus, it applies equally well to vehicle tracking in ITS or an enhanced Lo-Jack system. In fact, this protocol is a special case of location protected communication using a public infrastructure, where what is communicated is itself location information.

The remainder of the paper is structured as follows: In section 2 we give an overview of anonymous channels and present our notation for describing protocols that make use of them. In section 3 we set out two protocols, a protocol for location protected communication over cellular phones in section 3.1 and a protocol for private location tracking in section 3.2. In section 4 we present background information. In particular, we briefly describe onion routing, a system we have implemented for anonymous communication over the Internet. In section 5 we present our conclusions.

2 Anonymous Channels

For us, an *anonymous channel* is a communication channel for which it is infeasible to determine both endpoints. The principal initiating the connection is the *initiator*, and the principal to whom he connects is the *responder*. These are not merely theoretical constructs; we have implemented a mechanism for anonymous channels (in fact near real-time anonymous connections) that operates below the application layer and supports a variety of Internet applications [10, 11]. The design of our mechanism can be applied to non-Internet applications such as are described in this paper. We will give more background on our design in section 4.1. Just as the strength of an encryption algorithm is relative to assumptions about everything from special restrictions on the key space and on other inputs to the capabilities and collateral information of a potential attacker, so too what we mean by "infeasible" will have many caveats and limitations for any given mechanism to implement an anonymous channel. Fortunately, for protocol purposes we need not specify these any more than we need to specify properties of cryptographic algorithms and their implementations when we describe, for example, a protocol for authenticated key distribution.

In practice, we do assume that the initiator knows the responder: since he initiates the connection, he presumably knows to whom he intends to connect. Note that this assumes either a means to authenticate the responder or, in the case of one way communication, a means to guarantee that only the responder can receive the message, e.g., public key encryption. In theory, the initiator may be sending out a proverbial note in a bottle, destined for where he does not know. However, unlike a shipwrecked sailor on a desert island, our initiator will have reason to believe that his messages can ordinarily be tracked back to him the moment he releases them. Therefore, he will need to assure himself that his messages have drifted far enough away from him before anyone can begin to track them. Thus, even if he wishes to establish an anonymous channel with whomever will respond, he needs to determine a point away from himself before which the channel will not emerge.

The initiator may build an anonymous connection all the way to the responder. This would protect both of them from association with the channel by all but the initiator. However, since the initiator often needs only to hide that communication is coming from or going to him, in practice we may only have half-anonymous channels. In other words, the initiator produces a channel which cannot be traced to him and uses this to contact the responder. From the end of the anonymous part of the channel to the responder anyone can see what the responder is sending and receiving. If end-to-end encryption is piped through this half-anonymous channel, it can effectively be made fully anonymous. Nobody can tell what the responder is sending or receiving or to whom the responder is connected. The only thing that can be observed from the outside is that the responder is talking to someone. (This too can be hidden if the channel is maintained even when not in use and dummy traffic sent over it. However, such countermeasures are quite expensive.)

Note that anonymous channels are not explicitly required to be confidential channels. However, cleartext is obviously trackable. And, even ciphertext that appears the same everywhere is trackable. Thus, for the reasons we have been describing, an anonymous channel must be encrypted in a changing manner at least to a point where it is acceptable that the communication be tracked.

Before going further, we contrast anonymous channels with a related but distinct form of channels, specifically subliminal or covert channels. In theory, all of these are channels for which it is infeasible to detect the existence of the channel. Thus, our distinction deals more with the expected environment for a channel rather than the channel's undetectability in that environment. In practice, channels called 'subliminal' typically piggyback on legitimate channels between the principals. Covert channels either piggyback similarly or exist in a medium that is not explicitly a communications medium at all. So, covert and subliminal channels are channels that rely on some other type of channel or computation to hide them. (In most applications the covert or subliminal channel will be somehow illegitimate and the cover channel or computation legitimate.) By contrast, anonymous channels rely on other anonymous channels to hide them. The cover for these channels are other channels like them, and the hiding comes from the indistinguishability of these channels from each other. Another important contrast between these types of channels is their relative efficiency. Though not always the case, covert and subliminal channels are typically inefficient as compared to the legitimate communication they parallel. Anonymous channels are expected to be roughly comparable in efficiency to their non-anonymous counterparts in the same medium. We have found this to be the case in our implementation.

2.1 Anonymous Connections

The channels we consider for these applications are connection based. Thus, we will typically speak of anonymous connections rather than anonymous channels in general. We denote the sending of message M along an anonymous connection from X to Y by '$X \Rightarrow_X Y : M$'. It may be important to know if a message is being sent over an anonymous connection from initiator to responder or vice versa. Specifically, it may be important to know whose identity is protected from association with the channel. This is the purpose of the subscript in the just introduced notation. If Y sends X a message M' on an anonymous channel that X initiated, this is denoted '$Y \Rightarrow_X X : M'$'. Sending M on an ordinary (non-anonymous) connection from X to Y is denoted '$X \rightarrow Y : M$'.

Observation: $X \Rightarrow_X Y \rightarrow Z : M$ implies $X \Rightarrow_X Z : M$.

2.2 Replies to Anonymous Connections

It is also possible for the initiator to make available information that allows a specific or arbitrary responder to establish a connection back to the initiator. This connection will be anonymous just as if it were a connection established from the initiator. We call such a connection a reply-to-anonymous (RTA) connection. The data structure that allows a principal to make an RTA connection to X is denoted '$\langle \Rightarrow_X X \rangle$'. Note that if the responder builds the RTA connection on the end of an anonymous connection in which he is the initiator, the result is a connection in which neither principal can be identified (unless he sends identifying information through the connection).

3 Mobile Applications

We now consider some applications of anonymous channels. Specifically, we describe how anonymous connections may be used to hide location information in cellular phone and location tracking systems.

3.1 Location Hiding for Cellular Phones

First we will describe how to make calls to a cellular phone without requiring the phone to reveal its location. Then we will describe how to hide the location of a caller's cellular phone from the both the network and external eavesdroppers. There are other solutions with similar anonymity goals that contain many of the elements in our protocol [2, 3]. Our protocol focuses on simple yet anonymous communication on top of an energy efficient call-back architecture.

In current cellular phone systems, the location of a phone is tracked, so calls to that phone can be routed through the base station controlling the phone's current cell. This tracking has two disadvantages: One is that the system knows where phones are. The other is that phones must transmit frequently to update their locations. This drains the phone's battery quickly.

In our proposal, instead of tracking a phone's location, phones will be paged. When such a phone is called, the network invents a temporary number and pages the phone. The phone's response to the page will be to make a call back to the temporary number in the page. The phone network will then mate the two connections. In addition to overcoming the disadvantages just mentioned, this simplifies our protocol because we effectively need to describe only how to initiate a call from a cell phone; the phone never receives a call in the ordinary sense of 'receive'. We will return to discuss paging briefly below.

The principals specified in our protocol are the caller's cell phone P, the central switch S, and the callee intended to receive the call R. We now present our protocol for initiating a call from a cell phone.

1. $P \Rightarrow_P S : \textit{Payment info.}, N$
2. $S \Rightarrow_P P : \textit{Ack}$ or $\textit{Nack}$
3. $P \Leftrightarrow_P S \leftrightarrow R : \textit{Conversation}$

To make a call from a cellular phone without revealing location, the phone makes an anonymous connection to a central switch. It then sends to the switch the number it is trying to reach, together with some payment information to cover billing. The payment information may be the phone's subscriber ID or a credit card number or even anonymous e-cash of some sort. N is either R's phone number in the outgoing case or the temporary number from the page in the incoming case. Assuming that the payment information is acceptable the switch allows the call to be completed. In the outgoing case, the switch completes the call to R and patches this to the anonymous connection from P. In the incoming case, the switch allows the the connection from R to be patched to the anonymous connection from P.

Since we are only trying to hide location, the anonymous connection need not be made all the way from the caller to the callee. Rather, the anonymous connection is made to some central switch in the network, from which it can be passed along in the clear. This switch will not know from where the call is coming; however, it will not complete the call unless the phone sends identifying information or some guarantee of payment. (We do not here discuss how identification is authenticated.)

There is nothing in the protocol description to indicate that we are dealing with mobile phones. Of course with stationary phones, location protection of a given phone is moot. But, the protocol still protects the location origin of a call made from that phone. In fact, this sort of protection for stationary phones is discussed in [9]. It is helpful to have a notation abstract enough to cover anonymity in both these stationary phone connections and the mobile connections of [2, 3]. This is as it should be because the means to establish anonymous connections is separate from the basic communication medium that underlies it. For example, in our Internet implementation (cf. section 4.1) the underlying network is free to make whatever dynamic routing choices between points that it ordinarily does, provided that it connects to the points we do specify. Thus, the usual mobile phone procedures for connection to local base station and hand-offs between base stations as a phone changes cells are unaffected by our anonymous connections. The fact that there is movement in the cellular phone network is not hidden from the network; however, who is talking and where they are is hidden. So, the network is untrusted in this sense.

Our combination of anonymous connections and paging has two advantages. The locations of inactive phones do not need to be tracked within a paging region. Also phones never need to transmit except when they are involved in a call. This greatly reduces battery drain. For example, pagers last for a few months on a single battery, while cell phones last about a day in standby mode.

Many people would like to carry a cell phone to call or be called only in emergencies. Right now this is only convenient for outgoing calls. For incoming calls this is tedious at best since it still requires virtually daily charges of the battery. Our combination would make this more feasible since the phone could be carried for a month or more without recharging. It also has advantages over carrying a pager and a switched off cell phone. Aside from the advantage of needing to carry only one small device, calls from stations that do not allow incoming calls, e.g., payphones, can be taken.

One could imagine a variety of subscription prices for incoming calls based on the type of paging that is made available. Basic pagers typically operate in a large region (relative to cell phone cells), but basic service will not cover a large country like the US. Someone who regularly travels nationally might opt for a more expensive national (or international) paging service. In between these two extremes, one could have a phone-pager that operates regionally but updates the paging region it is in the first time it is turned on in that region. (Once it changes regions, it cannot receive incoming calls until it updates the region.)

3.2 Private Location Tracking

The next application we discuss is a location tracking service for which the user can control access to his location information. An active badging system can provide location information for individuals by sending badge identifiers to room sensors. Such a system, for example, has been implemented by Olivetti at the Cambridge University Computing Laboratory. While this information is useful for tracking people down, it may be a little too useful, making people hesitant to use it willingly. If control over access to an individual's location information can be placed in the hands of that individual, the system becomes much less threatening. The goal is then to provide a trusted home machine that can track the location of its user without centralized tracking information arising anywhere else and with limited computation power necessary for the wearable tracking device.

Here is a basic description of such a system. In terms of computing power of the wearable device (badge), it requires only that the user's wearable device share a pseudo-random number generator (PRNG) with the user's home machine. The PRNG is used to produce a sequence of tags that will serve roughly as one-time passwords. The badge sends a new tag every time it detects a different sensor. Once it receives an acknowledgement from the sensor it advances to the next tag. The badge must assume that the tag will be sent to the home machine at that point. The sensor opens an anonymous connection to the database and sends the current tag encrypted for the database and sends a (symmetric) key for encrypting the reply. The database looks up the RTA data structure associated with that tag (which the home machine has deposited there) and sends it back to the sensor. The sensor then uses this to construct an RTA connection to the home machine and sends the home machine his name (i.e., location information) and the tag. The principals specified in our protocol are: the wearable badge B, the room sensor S, the central database D, and the user's home machine H.

1. B detects new S
2. $B \rightarrow S : \mathit{Tag}$
3. $S \rightarrow B : \mathit{Ack}$
4. $S \Rightarrow_S D : \{\mathit{Tag}, K_{ds}\}_{K_d}$
5. $D \Rightarrow_S S : \{\langle \Rightarrow_H H \rangle\}_{K_{ds}}$ $\qquad$ ($\langle \Rightarrow_H H \rangle$ stored at D under Tag)
6. $S \Rightarrow_H H : S,\ \mathit{Tag}$

A few assumptions are necessary. We assume that the database is query only. This prevents an attacker from reading random RTA data structures from the database and confusing the home machine about the whereabouts of the badge. Even if attackers could read random RTAs, the attack would not reveal any information (but might confuse the home machine with bogus tags). Specifically, this attack would not allow an attacker to frame a badge wearer by sending a sequence of bogus locations to the home machine since there is no way to identify successive tags or matching RTA structures. Despite this, requiring that the database be query only will mean that the only way to mount such an attack would be to guess tags or to grab them from a badge as it sends them out, presumably requiring hardware and a more concerted attack. The home machine is assumed to have deposited (over an anonymous connection) an RTA structure for each tag. This should not be done in a batch unless the user wants to trust the database to know that all of the deposited tags and RTA structures are from the same home machine. However, many RTA structures can be deposited in advance. In fact, depositing several RTAs in advance makes this protocol resistant to attack via spurious Acks. Since each Ack makes the badge move to the next tag, spurious Acks might cause the badge and home station to drift out of synch. More specifically a query to the central database may not find a matching RTA. However, submitting enough RTAs in advance solves the spurious Ack problem, provided that the home machine checks ahead whenever it receives a tag that does not match the next one expected. The protocol is also resilient to lost Acks, even without depositing more than one RTA in advance. If the badge does not receive a sent Ack but the matching RTA is used, then only the next location update is lost.

There are some vulnerabilities associated with this protocol. Corrupt sensors could fail to acknowledge receipt of the tag. They could then cooperate to track the user who would be sending the same tags as it moved about. This attack is limited in that the tags will change every time the device encounters a properly functioning and uncompromised sensor. The device could also keep track of the number of times (or number of successive times) it fails to receive an *Ack*. If a threshold number is exceeded it could either cease to operate or flash or beep to indicate an error. This is useful for more than prevention of attacks since failure to properly receive *Ack*s could indicate a malfunction in the wearable device or in the locating system. Note that corrupt sensors cannot otherwise cooperate with each other or with a corrupt centralized database since there is nothing to correlate successive connections to the database. (If a badge moves from the range of one corrupt sensor to another, they could of course observe the successive connections made. However, even then it might take a good deal of analysis to determine which badge is likely to have followed which path, especially if a user passes through rooms with several other individuals.)

An alternative that would prevent the above attack is for a badge to send out a new tag at regular intervals, whether it encounters a new sensor or not. This probably entails more overhead since individuals are likely to spend extended intervals at given locations, e.g., in their offices. Notice that the database cannot infer that someone is sedentary (much less who) because there is no way to link successive queries as coming from or not coming from the same badge. And, the connection to it from the sensors is always anonymous, so it cannot tell whence queries come.

Another alternative protocol avoids the use of anonymous connections entirely. Here there is no centralized database. Instead sensors simply broadcast tags (and sensor IDs) to all home machines. Home machines then pick up those tags that are meant for them to track their users. There are a variety of tradeoffs between this protocol and those that make use of anonymous connections, e.g., cost of anonymous connection set-up vs. cost of broadcast. Which is the better approach is likely to be highly contextual.

4 Background

Chaum [1] defines a layered object that routes data through intermediate nodes, called *mixes*. These intermediate nodes may reorder, delay, and pad traffic to complicate traffic analysis. Chaum's mixes and related work are the basis for almost all subsequent work on anonymous communication. Other approaches to anonymity in mobile phone systems occur in [2] and [3]. Another approach to private location tracking occurs in [7]. The approach to anonymous connections that we have implemented is called *onion routing*. Onion routing shares many anonymity mechanisms with Babel [6] but Babel uses them specifically for e-mail, while onion routing uses them to build (possibly long lived) application independent connections. We now give a basic description of onion routing; more details can be found in [4, 10, 11, 5].

4.1 Onion Routing

Traffic analysis can be used to infer who is talking to whom over a public network. For example, in a packet switched network like the Internet, packets have a header used for routing, and a payload that carries the data. The header, which must be visible to the network (and to observers of the network), reveals the source and destination of the packet. Even if the header were obscured in some way, the packet could still be tracked as it moves through the network. Encrypting the payload is similarly ineffective, because the goal of traffic analysis is to identify who is talking to whom and not (to identify directly) the content of that conversation.

Onion routing protects against traffic analysis attacks from both the network and observers. Onion routing works in the following way: The initiating application, instead of making a connection directly to a responding server, makes a connection to an application specific *onion routing proxy.* That onion routing proxy builds an anonymous connection through several other *onion routers* to the destination. Each onion router can only identify adjacent onion routers along the route. When the connection is broken, even this limited information about the connection is cleared at each onion router. Data passed along the anonymous connection appears different at and to each onion router, so data cannot be tracked en route and compromised onion routers cannot cooperate. An onion routing network can exist in several configurations that permit efficient use by both large institutions and individuals. The onion routing proxy defines a route through the onion routing network by constructing a layered data structure called an *onion* and sending that onion through the onion routing network. Each layer of the onion is public key encrypted for the intended onion router and defines the next hop in a route. An onion router that receives an onion peels off its layer, reads from that layer the name of the next hop and the cryptographic information associated with its hop in the anonymous connection, pads the embedded onion to some constant size, and sends the padded onion to the next onion router.

Before sending data over an anonymous connection, the initiator's onion routing proxy adds a layer of encryption for each onion router in the route. As data moves through the anonymous connection, each onion router removes one layer of encryption, so it finally arrives as plaintext. The last onion router forwards data to another type of proxy, called the *responder's proxy,* whose job is to pass data between the onion network and the responding server. This layering occurs in the reverse order for data moving back to the initiator. So data that has passed backward through the anonymous connection must be repeatedly decrypted to obtain the plaintext.

5 Conclusion

This paper presents a means for talking about anonymous connections and protocols that make use of them. We demonstrate the usefulness of describing anonymous connections at this level of abstraction by using our notation to describe two protocols for location protected communication over a public infrastructure.

References

1. D. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms, *Communications of the ACM*, v. 24, n. 2, Feb. 1981, pages 84-88.
2. H. Federrath, A. Jerichow, D. Kesdogan,and A. Pfitzmann. Security in Public Mobile Communication Networks, *Proceedings of the IFIP TC 6 International Workshop on Personal Wireless Communications*, Verlag der Augustinus Buchhandlung Aachen, 1995, pages 105–116.
3. H. Federrath, A. Jerichow, and A. Pfitzmann. MIXes in Mobile Communication Systems: Location Management with Privacy, in *Information Hiding*, Ross Anderson ed., Springer-Verlag, LNCS vol. 1174, June 1996, pages 121–135.

4. D. Goldschlag, M. Reed, and P. Syverson. Privacy on the Internet, *INET '97,* Kuala Lumpur, Malaysia, June, 1997.
5. D. Goldschlag, M. Reed, and P. Syverson. Hiding Routing Information, in *Information Hiding*, Ross Anderson ed., Springer-Verlag, LNCS vol. 1174, June 1996, pages 137–150.
6. C. Gülcü and G. Tsudik. Mixing Email with *Babel*, *1996 Symposium on Network and Distributed System Security,* San Diego, February 1996.
7. I. Jackson. Anonymous Addresses and Confidentiality of Location, in *Information Hiding*, Ross Anderson ed., Springer-Verlag, LNCS vol. 1174, June 1996, pages 115–120.
8. P. Karger and Y. Frankel. Security and Privacy Threats to ITS, *The Second World Congress on Intelligent Transportation Systems*, Yokohama, Japan, November 1995, pages 2452–2458.
9. A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDN-Mixes: Untraceable Communication with Very Small Bandwidth Overhead, *GI/ITG Conference: Communication in Distributed Systems*, Mannheim Feb, 1991, Informatik-Fachberichte 267, Springer-Verlag, Heildelberg 1991, pages 451-463.
10. M. Reed, P. Syverson, D. Goldschlag. Proxies for Anonymous Routing, *Proceedings of the* 12^{th} *Annual Computer Security Applications Conference*, IEEE CS Press, December, 1996, pages 95–104.
11. P. Syverson, D. Goldschlag, and M. Reed. Anonymous Connections and Onion Routing, *Proceedings of the Symposium on Security and Privacy*, Oakland, CA, May 1997, pages 44–54.

This article was processed using the LaTeX macro package with LLNCS style