Anonymous addresses and confidentiality of location

Ian W. Jackson

Cambridge University Computer Laboratory /
Olivetti Research Limited, Cambridge

7th November 1995

Abstract

I describe a scheme which will allow location information from devices like active badges to be collected by an agent controlled by the user, hiding the user's location from the network infrastructure, so that the user may distribute the information about their location as they see fit.

Anonymous addresses and confidentiality of location

Ian W. Jackson

Cambridge University Computer Laboratory /
Olivetti Research Limited, Cambridge

7th November 1995

1. Introduction

Suppose a location information system which involves something like active badges.

Many people, especially employees of large companies not noted for concern with things like employee privacy, are reluctant to accept this kind of technology because they fear that the management will use them to 'spy' on them.

To alleviate this fear I believe that it is necessary to give users strong control over the information gathered by such a system.

This control should continue to be effective even when large portions (but not all) of the computing infrastructure have been 'compromised', in the sense that the people who run them are doing monitoring.

Here is an outline of a possible solution to this problem.

2. Users' agents

Possible the most flexibly way of achieving the goal is to allow the user to maintain, on systems controlled by themselves or those that they trust, an 'agent' which is the only entity that has direct access to the information about their location. Anyone wanting the user's location must ask their agent.

The agent may be quite remote from the sensors doing the actual tracking.

How does the user's agent keep track of the user's location, without allowing other parts of the system to do so ? Ideally the badge (or other device carried by the user) would be able to send unobserved and unintercepted a message directly to the user's agent. The sensor then simply sends the user's agent the location data. Unfortunately this is impractical as it stands. Spread-spectrum radio might be an option, but would be rather large to implement in a badge-like device, and in any case restrictions on the types of devices and the ways of gathering location information are undesirable.

3. Anonymous address labels

Instead, I have borrowed some ideas from the cryptographically-based anonymous email remailer networks. Messages are sent via a series of remailers, all of whom must be compromised for a message to be traceable (other than by traffic analysis).

The badge sends to the sensor an ‘address label’, which the sensor can use to communicate with the user’s agent without knowing where that agent is or whose it is.

The address label consists of a key, with which to encrypt the location data, the name (or network address) of the next remailer, and an piece of encrypted instructions which is to be interpreted by that next remailer. This latter will usually consist of another address label.

The sending badge constructs the address label by repeated encapsulation, and broadcasts it for the benefit of any sensors present.

Each sensor which receives it use the label to send a message containing the location of the sighting.

Each remailer strips off, uses and discards one layer of addressing encapsulation. At each stage the key from that stage’s address label is used to reencipher the data, so that the destination receives a multi-layered encrypted message.

Address labels have to be constructed so that many different labels can be found for a single path or destination - this is so that the address label doesn’t become an alternative means of identifying the user. Preferably, each transmission from a badge will use a different label.

4. Protocols

4.1. Public key done by badge

Each remailer receives a message in two parts. The second part is the message itself (possibly already encrypted, perhaps more than once).

The first part is encrypted, using the remailer’s public key. It contains a symmetric key with which to encrypt the data, and information which can be added to the message to allow the final recepient to determine that key (the key must change with each new label, so that it cannot be used to trace the user and is not so easily compromised). The first part also contains the name of the next hop destination, as well as an encrypted lump that the next hop (presumably) knows how to unpick.

I propose to generate the ‘hop session key’, which is used to re-encrypt the data, by encrypting a nonce under a key shared between the badge and the agent. By encrypting the nonce in two different ways we can give the remailer the session key as well as allow the remailer to pass on without understanding information that will allow the agent to recover the session key.

Let us call the badge B and the user's agent A. Supposing that remailer P receives:

$$\boxed{\{\{N_{B,P}\}_{K_{AB}}, \{f\,(N_{B,P})\}_{K_{AB}}, Q\,,\ \{\textit{stuff for } Q\}_{K_Q}\}_{K_P}}\ ,\ \boxed{D}$$

Here the left hand box is the "anonymous address label", and the right hand side is the data to be sent.

It then sends to Q (which is either the next remailer or A, the final destination, but P doesn't need to know which):

$$\boxed{\{\textit{stuff for } Q\}_{K_Q}}\ ,\ \boxed{\{N_{B,P}\}_{K_{AB}}, \{D\}_{\{f(N_{B,P})\}_{K_{AB}}}}$$

A can now recover the original D by decrypting $N_{B,P}$, applying f (which might be some simple function such as exclusive or with a fixed bit-pattern), and hence deriving the key $\{f\,(N_{B,P})\}_{K_{AB}}$.

Before a message as shown above arrives at the first remailer it is assembled by the sensor. The sensor (S) is receives from the badge data equivalent to that used by a remailer (ie, *stuff for S*, which includes amongst other things P and $\{\textit{stuff for } P\}_{K_P}$). This data need not be encrypted, if we do not trust the sensor and do not want to spend effort encrypting things for it which we could better spend encrypting things for a more trusted party. S encrypts the location L using the rules for D above, and sends the result to P.

At the last stage, where $Q = A$, *stuff for Q* is merely B's name, so that A knows which of the possibly many badges it is handling is involved. A strips off the many levels of encryption to recover the original D, ie L, the location.

4.2. No public key in badge

Public key cryptography has disadvantages for very small mobile entities; it requires much computational effort, and the messages it produces are large.

The required aspect of public key cryptography is the fact that many senders can encrypt messages to the same recipient in such a way that none of the senders can read each other's messages, nor can they relate different messages from the same sender. We can achieve much the same effect by using symmetric key with large numbers of public keys.

The recipient issues {nonce,key} pairs on request; these are constructed so that the recipient can derive the key from the nonce easily, but the nonces are unpredictable, and noone else can relate nonces and keys that they haven't explicitly been sent:

$$N_{Xi} = \{i\}_{K_{XX}}; K_{Xi} = \{f(i)\}_{K_{XX}}$$

We can now rewrite the remailing function using these pairs. Supposing that remailer P receives:

$$\boxed{N_{Pi}, \{N_{Ak}, K_{Ak}, Q, N_{Qj}, \{stuff\ for\ Q\}_{K_{Qj}}\}_{K_{Pi}}}, \boxed{D}$$

It then sends to Q:

$$\boxed{N_{Qj}, \{stuff\ for\ Q\}_{K_{Qj}}}, \boxed{N_{Ak}, \{D\}_{K_{Ak}}}$$

4.3. Delivery of keys to badge

The above scheme requires that the badge have a stock of anonymous address labels for its user's agent. Since the labels now only contain symmetric keys and symmetric key encrypted data they are a lot smaller - just over 3 times the key size per hop - and the badge is now no longer required to manipulate them. This means we can do away with the complication surrounding $N_{B,P}$ and K_{AB}.

Instead, we have the agent collect {nonce,key} pairs from the remailers, construct address labels for itself, and send them to the badge. It needs a way of sending the labels to the badge, anonymously.

We have to use public key cryptograpy here, in order to bootstrap the symmetric keys. Supposing that remailer Q receives

$$\{P, \{stuff\ for\ P\}_{K_P}\}_{K_Q}$$

it sends to P:

$$\{stuff\ for\ P\}_{K_P}$$

This is similar to the electronic mail remailers – the sender (here the user's agent) is responsible for constructing the message and nesting the encryptions.

4.4. Request for {nonce,key} pairs

Finally, we need a protocol for the user's agent to request {nonce,key} pairs. This has to be done anonymously too. We use the 'reverse' anonymity mechanism to transmit the request, and the anonymous address label scheme itself to return the reply.

When a remailer P receives:

$$\{ \boxed{N_{Qj}, \{stuff\ for\ Q\}_{K_{Qj}}}, N_{Ak}, K_{Ak}, n\}_{K_P}$$

it replies, via Q, with:

$$\boxed{N_{Qj}, \{stuff\ for\ Q\}_{K_{Qj}}}, \boxed{N_{Ak}, \{N_{P(i+1)}, K_{P(i+1)} \ldots N_{P(i+n)}, K_{P(i+n)}\}_{K_{Ak}}}$$

4.5. Freshness

The descriptions above do not include any timestamps or other means of producing freshness. However, because the user's agent knows when each of the requests for address labels was sent it can discard stale replies, and also stale address labels.

This is another improvement over the public-key based protocol, which would need the introduction of clocks or perhaps some kind of challenge-response.

4.6. Limitations and practicality

There are a number of unresolved issues here. In particular, problems will arise when the badge is out of touch for a long time and runs out of address labels. The usual tradeoff between security and functionality exists here: the badge can go off-air (or transmit much less frequently), until it is 'refilled' (perhaps by the user pressing a button to indicate that they're willing to have their location known), or it can reuse old labels in the hope that this will allow the agent to find it and send it some new ones.

I have not yet analysed the protocols above for weaknesses; I intend to use a version of the BAN logic which I've extended to check confidentiality and the correctness of the idealisation step.

There will be a certain vulnerability to traffic analysis. The key setup messages will provide cover traffic for the actual badge data, but they do tend to be rather large. In order to solve this problem it will probably be necessary to make explicit provision for cover traffic; this, combined with a certain degree of regular queueing of messages (rather than sending them instantaneously) may help. Further thought is required in this area.

5. 'Advanced' applications

5.1. 'Who is at location L ?'

It will frequently be the case that an enquirer will want to ask questions about locations, rather than users. How, then, will they know which agent(s) to ask ?

Such enquiries can be sent to the sensors in the relevant area. The sensors can keep a list of recently-seen address labels, and forward the enquiry to all of those agents. Each agent can then choose whether and how to reply (possibly anonymising the reply, for example to say 'there is someone here but they won't tell you who they are').

5.2. Unlocking doors and similar security-critical decisions

If the security policy of the door or whatever is such that the user can be trusted not falsely to say 'yes' then the sensor near a door (for example) could include in its message to the agent a request for credentials suitable for opening the door.

If the requirements are more complicated the door will have to use the anonymous address label to tell the agent to tell the badge to send its credentials, in some sense.

The exact details of these kind of applications need to be worked out.