マルチメディア通信と分散処理 58－5
（1992. 11. 19）

BANロジックによるアドレス登録プロトコルの検証

村山 優子
WIDE プロジェクト／慶應義塾大学

コンピューター・ネットワーク環境下でのひとつの問題は、ネーム・サーバなどネットワーク情報システムに入っている情報自体が、必ずしも信用できないということである。先のINET'92では、情報の一例として、ホスト・アドレスに着目し、信用できるアドレス情報のための登録プロトコルを提唱した。本論文は、そのプロトコルをBANロジックを用いてフォーマルな形での検証を試みた。この試みは、本来BANロジックが作られた背景、すなわち、認証プロトコルの検証とは、いささか異にし、情報の完全性を検証することが目的である。

Using BAN logic for the proof of a registration protocol

Yuko Murayama
WIDE Project/Keio University

The current problem of the computer network environment is that information held in the information systems such as a name server, is not necessarily trustworthy. We look particularly at an address as an example of the network configuration information. In this view, we proposed a registration protocol, which would preserve the credibility of an address at INET92. This paper examines the protocol in a formal manner using the logic introduced by Burrows, Abadi, and Needham, notably called BAN Logic. Our use is somewhat different from their original use for authentication protocols in the way that information integrity is examined.

1 Introduction

Computer networks have evolved to provide users with access to a wide range of information resources. Information has long been treated as data in computer networks, and much research has been carried out on how well networks can carry data from one end to another. Shannon identified this level of research as the engineering level [5]. This paper is concerned with the semantic level of information; i.e. is the information being carried trustworthy?

In our INET92 paper [3], we have proposed a registration protocol which preserves the credibility of network layer addresses of hosts. In this paper, we examine the protocol in a formal manner using the logic introduced by Burrows, Abadi, and Needham [2]. Their attempt is novel because the protocol flow is expressed in terms of the knowledge obtained at each participant of the transaction; with the traditional notation, one can only express the syntax and flow of the messages. Although the argument by Nessett [4] showed that the logic missed out the confidentiality aspect intentionally for simplifying the process [1], it is no problem with our application of the logic because our primary concern is to examine information integrity.

Next section describes our model of address information flow and a protocol for registration. Section 3 examines the protocol by using a formal method. Section 4 gives some conclusions.

2 Our registration procedure

2.1 Overview

Our idea presented at INET'92 was to prevent the threat of redirection of packets at the network level within a subnet by the certification of network and subnetwork address mapping. When a host starts up, it has to go through a registration procedure. During registration, the host configuration is verified. Upon a successful registration, a host is given an authorised token. This implies that a host which is misconfigured would not get the token. The token will be used in a secure network operation such as the secure resolution of a network address into a subnetwork address, allowing only authorised hosts to appear in the address resolution table at each host. This prevents an innocent host from sending a packet to a bogus host by unauthorised redirection, because the innocent hosts use the above secure address resolution to locate the subnetwork address of deceiver's server. We call an authorised token a certificate in the rest of this paper.

We propose the procedures on how a certificate associated with a network address can be issued. Our scheme includes adding two processes to the current address information flow; one is the off-line registration of the address of a network host with an authority during information generation, the other is the configuration verification of the host when it announces itself to a network. Upon successful verification, an authority will issue a certificate for the network address which will be registered with the information systems. Network operations make use of these information systems. This way, only confirmed information can appear in the network operations.

The information system may be a dynamically leaning system such as the Ethernet ARP system and a routing information exchange system. Alternatively it may be a sort of database system such as a directory service system and a name server.

The organisation of this section is as follows. The next subsection presents our address information flow highlighting two additional phases to the current address information flow. Section 2.3 describes the registration protocol.

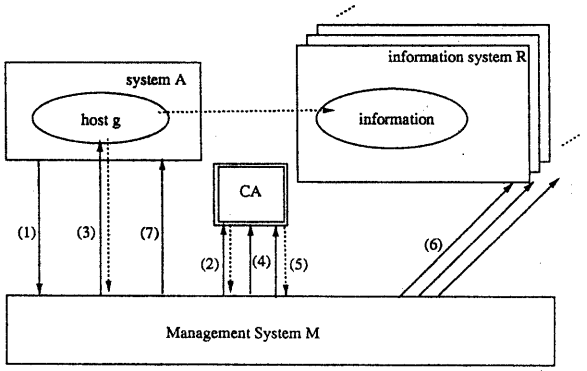
2.2 Our network address information flow

What is missing in the current addressing scheme is that there is no mechanism for verifying the credibility of an address and for binding between a host and the address. In our model, we add two procedures to the current address information flow; one is certificate registration, another is configuration confirmation. The information flow from the object generator to the information registration, therefore, has the following phases:

- Resource Admission Phase
- Naming Phase
- Certificate Registration Phase
- Configuration Phase
- Configuration Confirmation and Registration Phase

The order of the phases does not particularly matter. For example, in a real-time naming environment, naming takes place after an object is configured to join the network.

In the Resource Admission Phase, an environmental policy is enforced to make the decision to admit an object to be configured. The object is named in the



Legend:

(1) : System A initiates the registration of host g

(2) : M makes a request to CA for the information associated with registration ID

(3) : M verifies that g is operational

(4) : M reports to CA that g is operational

(5) : M asks CA for notarisation

(6) : M distrivutes the updates to the information systems

(7) : M gives the notarised address to A

——→ : request or report

←····· : reply

Figure 1: The flow of configuration confirmation and registration

Naming Phase. Name and associated cryptographic system attributes of the object are registered at the Certificate Registration Phase. As a successful result, the registration ID is issued. The Configuration Phase is to introduce an object into the environmental operation.

In the Configuration Confirmation and Registration Phase, the object starts the registration of its associated information with the registration ID through the environmental operation. It is this phase where the verification to ensure that the object is bound to the name which was assigned in the Naming Phase, is carried out in our scheme.

In this paper, we are concerned with the Configuration Confirmation and Registration Phase. The idea is that when an object is configured in a network, it goes into the confirmation phase and the confirmed object is given a certificate of its configured address. The certificate is registered into the information system, so that the information system holds credible information. The next section will introduce a registration protocol, which is used in the Configuration Confirmation and Registration Phase.

2.3 A registration protocol

The purpose of configuration confirmation is two-fold: one purpose is to verify the configuration (e.g. binding of an allocated address to an object); another is to validate the allocated address. Configuration con-

firmation is carried out at *system generation.*

We presume that a system starts requesting for notarisation of information items with a registration ID, when it has been configured with a network interface. We call this stage *system generation.* The registration ID acts as a session key throughout the Configuration Confirmation Phase.

In the following we introduce a protocol for configuration confirmation as well as registration. Figure 1 shows the protocol flow.

The configuration confirmation sequence is as follows:

1. A host uses the network to inform the manager of the completion of its configuration using the registration ID
2. The manager gathers the host information, including the associated registration ID, which was registered at Certificate Registration.
3. The configuration of the host is verified using the previously registered information.
4. Upon successful verification, the manager recognises the addition of the host.
5. The host address is authorised.
6. The authorised address is registered by the information system.
7. The certificate of the address may be given to the host.

In steps 1 through 3, authentication of a registration session for a host is carried out. A registration ID is used as a shared secret between the system A and the CA. However, knowing the registration ID is not enough to authenticate the session. Only if the system is recognised also as being configured correctly in step 3, will the session continue.

In step 4 above, additional information about the new host is recognised. The associated information is certified in 5, and then passed to the information system in 6. A certificate for the network and link address mapping which could be used in the secure address resolution operation is issued at this stage.

Later in 7, the host system will receive the certificates.

3 Formal analysis of the protocol

3.1 Overview

We examine our registration protocol by the formal method introduced by Burrows, Abadi, and Needham [2]. We are interested particularly in how an allocated address is passed by its allocator, the Naming Authority, to the host and the information systems, and how its integrity is preserved.

In the next subsection, we introduce the notation. Section 3.3 describes the goal of our analysis, and Section 3.4 describes rules, and Section 3.5 gives the assumptions. Section 3.6 presents our analysis of the protocol.

3.2 Introduction to notation

The followings are the notations introduced by Burrows et al.:

$A \models X$: means that A believes that X is a true information item.

$A \Rightarrow X$: means that A has a jurisdiction over X.

$\sharp(X)$: originally means that the information item, X, is generated recently.

We modify this definition into that the production of X has been validated and verified recently; i.e.X is perceived as correct at a recent time, however, it is not clear whether its generation was recent, or not.

$A \triangleleft X$: A sees X.

$A \mid\sim X$: A once said X.

We introduced a new notation as follows, which indicates the information item has been notarised by a trustful authority:

$\langle\langle X \rangle\rangle_C$: X is certified by the authority C.

In the following subsections, we also use the following abbreviations:

- Reg : the registration ID
- X_a : the address of the object
- X_d : the description of the object
- $X_c(Z)$: the cryptographic system attributes of Z
- T_Z : the time stamp of Z

The participants are expressed as follows:

- NA : the Naming Authority
- CA : the Certification Authority

- M : the Management System
- R : the Information System
- A : a system to be configured with the allocated address

3.3 The goal of analysis

As the notation which we use is originally intended for authentication, the successful result of a protocol between two principals, A and B, is that both principals know the shared secret, and each knows that other knows it as well. The notation for this result is as follows:

$$A \models X \text{ and } A \models B \models X$$

and

$$B \models X \text{ and } B \models A \models X$$

However, our goal is somewhat different, in that we need to ensure an information item on an object, g, e.g. a network address, X_a, originated by the *information originator*, NA, and verified by CA, is assigned to the *information provider*, A, as is intended, and arrives at an *information system*, R, eventually. Therefore, the successful result should be as follows:

(1) $NA \models X$ and (2) $NA \models A \mid\sim X$
(3) $CA \models NA \models X$ and (4) $CA \models A \mid\sim X$
(5) $A \models NA \models X$ and (6) $A \models X$
(7) $R \models CA \models X$ and (8) $R \models NA \models X$
(9) $R \models X$

We rewrite the above goals with more precise information items using the following notations:

- *Bind(Xa)=g* means the address, Xa is assigned for the object, g
- *Claim(g)=Xa* means g is claiming Xa

Then the goals are expressed as follows:

(1) $NA \models Bind(\mathrm{X}a)=g$
and (2) $NA \models Claim(g)=\mathrm{X}a$
(3) $CA \models NA \models Claim\ (Bind(\mathrm{X}a))=\mathrm{X}a$
and (4) $CA \models Claim(Bind(\mathrm{X}a))=\mathrm{X}a$
(5) $A \models NA \models Claim\ (Bind(\mathrm{X}a))=\mathrm{X}a$
and (6) $A \models Claim(Bind(\mathrm{X}a))=\mathrm{X}a$
(7) $R \models CA \models Claim\ (Bind(\mathrm{X}a))=\mathrm{X}a$
and (8) $R \models NA \models Claim\ (Bind(\mathrm{X}a))=\mathrm{X}a$
(9) $R \models Claim\ (Bind(\mathrm{X}a))=\mathrm{X}a$

3.4 The BAN Logic rules

The following rules are defined in BAN Logic.
The first rule is that if A sees the information item, X, encrypted with B's secret key, A believes that B once said X as follows:

$$\frac{A \models \overset{K}{\mapsto} B, A \triangleleft \{X\}_{K^{-1}}}{A \models B \mid\sim X}$$

The second rule is that if A believes X is uttered only recently and B once said X, then A believes that B has said X, recently as follows:

$$\frac{A \models \sharp(X), A \models B \mid\sim X}{A \models B \models X}$$

The third rule is that if A believes that B has jurisdiction over X then A trusts B on the truth of X as follows:

$$\frac{A \models \sharp(X), A \models B \mid\sim X}{A \models B \models X}$$

3.5 The assumptions

We have the following assumptions:
Assumption 1: $CA \models \sharp\ (Reg)$
Assumption 2: $CA \models NA \mapsto Bind(\mathrm{X}a)= g$
Assumption 3: The timers in all the agents of concern are synchronised, so that they believe their time stamps each other.

For $x \in \{ M, R, CA, A \}$,
$x \models (\sharp\ (T_M), \sharp\ (T_{CA}), \sharp\ (T_R), \sharp\ (T_A))$

Assumption 4: $\{ A, M, R \} \models CA \mapsto (Reg, \mathrm{X}a, \mathrm{X}d, \mathrm{X}c)$
Assumption 5:
$\{ A, M, R \} \models CA \mapsto \{ Claim(Bind(\mathrm{X}a))=\mathrm{X}a \}$
That is, if
$\{ A, M, R \} \models CA \mid\sim x$,
where $x \in \{ Claim(Bind(\mathrm{X}a))=\mathrm{X}a \}$,
then $\{ A, M, R \} \models x$.
Assumption 6: $CA \models \{ A, M, R \} \models CA \mapsto \{ (Xa) \}$
Assumption 7:
$\{ A, M, R \} \models NA \models Claim(Bind(\mathrm{X}a))=\mathrm{X}a$,
if and only if
$\{ A, M, R \} \models CA \models Claim(Bind(\mathrm{X}a))=\mathrm{X}a$.
Assumption 8:
$\{ CA \} \models M \mapsto$ (Add, object ID, information)
i.e. M has an authority to declare the addition of an object and it indicates that the verification has done.
We also assume that relevant public keys are distributed previously to the manager system M, the information system R, and the object site system A.

3.6 Protocol analysis

The protocol is summarised as follows:

Message 1:
$A \rightarrow M$: [Report, g's ID, $\{ Reg, T_A \} PK_M$]

Message 2.1:
$M \rightarrow CA$: [Request, g's ID, $\{\{ Reg, T_M \} SK_M \} PK_{CA}$]

Message 2.2:
$CA \rightarrow M$: [Reply, g's ID, $\{\{ Reg, \mathrm{X}a, \mathrm{X}d, \mathrm{X}c(A), T_M + 1 \} SK_C A \} PK_M$]

Message 3.1:
$M \rightarrow A$: [Request, $\{ g, \mathrm{ID}.\mathrm{X}a, \mathrm{ID}.\mathrm{X}d, T_M \} SK_M$]

Message 3.2:
$A \rightarrow M$: [Reply, $\{\{ \mathrm{XX}a, \mathrm{XX}d, T_M + 1 \} SK_A$]

If M perceives that $PK_A \in \mathrm{Xc(A)}$, $\mathrm{XX}a = \mathrm{X}a$,and $\mathrm{XX}d = \mathrm{X}d$, M sends the following packets, otherwise the procedure stops.

Message 4:
$M \rightarrow CA$: [{ Report, Add,g, Xa, Xd, Xc(A), $T_M \} SK_M$]

Message 5.1:
$M \rightarrow CA$: [{ Request, g, Xa, $T_M \} SK_M$]

Message 5.2:
$CA \rightarrow M$: [{ Reply,g, $\langle\langle \mathrm{X}a \rangle\rangle$ CA, $T_M + 1 \} SK_C A$]

Message 6:
$M \rightarrow R$: [{ Report,g, Xd, $T_M \} SK_M$]

Message 7:
$M \rightarrow A$: [{ Set,g, $\langle\langle \mathrm{X}a \rangle\rangle$ CA, Xc(group), $T_M \} SK_M$]

Protocol analysis is as follows.

From Message 1,

$$M \triangleleft Reg$$

From Message 2.1,

$$CA \models M \mid\sim Reg$$

From Message 2.2,

$$M \models CA \mid\sim (Reg, \mathrm{X}a, \mathrm{X}d, \mathrm{X}c(A)),$$

and

$$M \models CA \Rightarrow (Reg, \mathrm{X}a, \mathrm{X}d, \mathrm{X}c(A))$$

hence,

$$M \models (Reg, \mathrm{X}a, \mathrm{X}d, \mathrm{X}c(A))$$

After the receipt of Message 3.2 which is the reply to the request in Message 3.1,

$$M \models A \mid\sim (Claim(g) = XXa, XXd)$$

Then M checks locally whether XXa=Xa and XXd=Xd (and the success of decryption of the packet shows $PK_A \in \mathrm{Xc}(A)$) or not. If so, the configuring Xa with the object g is verified as follows:

$$M \models A \mid\sim (Claim(g) = Xa)$$

From Message 4,

$$CA \models M \models (A \mid\sim (Claim(g) = Xa))$$

Also,

$$CA \models M \Rightarrow (A \mid\sim (Claim(g) = Xa))$$

hence,

$$CA \models (A \mid\sim (Claim(g) = Xa))$$

On receipt of Message 5.1 of the request of certificate from M, CA remembers that it already knew the followings:

$$CA \models (Reg, Claim(g) = Xa)$$

and from Message 4,

$$CA \models (A \mid\sim (Claim(g) = Xa))$$

Thus CA could issue certificate straight away. However, here CA checks the inconsistency of addressing by looking through a list of addresses which CA has certified. NA will be informed if there is any double-allocation of an address. Moreover,

after the validation there needs to be a transaction (probably off-line) between *CA* and *NA* to confirm each other that the allocated name, X*a*, is indeed usable as follows:

$$CA \rightarrow NA\text{: } \langle\langle Claim(g)\text{=Xa}, \mathrm{T}_{CA} \rangle\rangle\ CA \qquad (5.1.add1)$$

If *NA* perceived that *Bind*(X*a*)=*g*, then it sends the following reply:

$$NA \rightarrow CA\text{: } \langle\langle\ \mathrm{T}_{CA}+1\ \rangle\rangle\ \mathrm{NA} \qquad (5.1.add2)$$

From (5.1.add1),

$$NA \models CA \mid\sim (Claim(g) = Xa)$$

From (5.1.add2), we can see that *NA* admits the followings:

$$NA \models Bind(Xa) = g \qquad (1)$$

and

$$NA \models Claim(g) = Xa \qquad (2)$$

As *CA* already knew that *Claim*(*g*)=X*a*, it deduces the following:

$$CA \models NA \models (Claim(Bind(Xa)) = Xa) \qquad (3)$$

CA issues a certificate for *Claim*(*Bind*(X*a*))=X*a* in Message 5.2.

From Message 5.2 as *CA* has sent the certificate, it shows that

$$CA \models Claim(Bind(Xa)) = Xa \qquad (4)$$

$$M \models CA \models Claim(Bind(Xa)) = Xa,$$

and from Assumption 5,

$$M \models Claim(Bind(Xa)) = Xa$$

On issue of the certificate, *CA* should expire the *Reg*. Also, the timeout limit should be set in *Reg* in case that for some reason the registration would never be invoked.

From Assumption 6,

$$CA \models M \models CA \models Claim(Bind(Xa)) = Xa$$

From Message 6.3,

$$R \models CA \models Claim(Bind(Xa)) = Xa \qquad (7)$$

From Assumption 7,

$$R \models NA \models Claim(Bind(Xa)) = Xa \qquad (8)$$

From Assumption 8,

$$R \models Claim(Bind(Xa)) = Xa \qquad (9)$$

From Message 7.1,

$$A \models M \mid\sim (Set, g, \langle\langle Claim(Bind(Xa)) = Xa \rangle\rangle CA, Xc(group))$$

and

$$A \models M \mapsto (Set, g, \langle\langle Claim(Bind(Xa)) = Xa \rangle\rangle CA, Xc(group))$$

hence,

$$A \models (Set, g, \langle\langle Claim(Bind(Xa)) = Xa \rangle\rangle CA, Xc(group))$$

$$A \models CA \models Claim(Bind(Xa)) = Xa,$$

and also

$$CA \mapsto Claim(Bind(Xa)) = Xa$$

hence,

$$A \models Claim(Bind(Xa)) = Xa \qquad (6)$$

and from Assumption 7,

$$A \models NA \models Claim(Bind(Xa)) = Xa \qquad (5)$$

Now that A receives the certificate, $A \models Reg.ID$ and it knows that the registration has been completed.

Q.E.D.

4 Conclusion

The current problem in networks is that there is no way to know how reliable the information is. How

reliable the information-based system such as a network operates, depends on how reliable the information is. The information system is responsible for the information used for network operations. We looked particularly at information registration.
The protocol consists of two levels of operation. One is an *engineering level* check and another is a *semantic level* check. The basic verification at the *engineering level* is to maintain the data integrity. The authentication would be exercised further to authenticate a sender, however, our main purpose is to ensure that a given information is correct at the *semantic level*. This is done by verification and authorisation.
We have introduced our protocol in terms of transactions between management agents. The protocol was analysed formally by making use of the logic recently introduced by Burrows, Abadi, and Needham. We have proved how the integrity of the information, an address, is maintained.
Our attempt was to use the formal notation and logic, intended originally for proving authentication protocols, to prove the integrity of information flow. We have managed to use the logic system to prove the goals; however, we have found a difficulty in this use. In an authentication protocol, the number of the parties involved in the operation would be usually at most three; two sites and an authority. Since our protocol involves more than three agents, the structure of information flow is more complex. In particular, at the Certificate Registration Phase, there are multiple paths for information flow; one goes though to a system which will have the object configured; another goes to the on-line Certification Authority (CA). We have no clear way to differentiate an address, X_a, which is set to the object at the Configuration Phase, from the one, which is in the knowledge of CA. In this sort of situation, we may need some more explicit way to express how an information item comes through. Apart from that, our trial of use of the logic has shown that it is useful when one wants to express the flow of information on the semantics level.

Acknowledgements

I would like to thank Peter T. Kirstein at UCL for his support during the research presented in this paper. I would also like to thank Mike Burrows at SRC/DEC for giving me the TEXmacros for the logic which I used here.

References

[1] M. Burrows, M. Abadi, and R. Needham. Rejoinder to nessett. *ACM Operating Systems Review*, Vol. 24, No. 2, pp. 39–40, April 1990.

[2] Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. Technical Report 39, DEC Systems Research Center, February 1989.

[3] Y. Murayama. Configuration detection in computer networks within an organisation. *Proc. of INET'92*, pp. 307–316, June 1992. Kobe, Japan.

[4] D. M. Nessett. A critique of the burrows, abadi, and needham logic. *ACM Operating Systems Review*, Vol. 24, No. 2, pp. 35–38, April 1990.

[5] C. Shannon. *The Mathematical Theory Of Communication*. The University Of Illinois Press: Urbana, 1949.