

Extending BAN Logic for Reasoning with Modern PKI-based Protocols

Sufatrio
Temasek Laboratories,
National University of Singapore
5 Sports Drive 2, Singapore 117508, Singapore
tsslufat@nus.edu.sg

Roland H.C. Yap
School of Computing,
National University of Singapore
Law Link, Singapore 117590, Singapore
ryap@comp.nus.edu.sg

Abstract

BAN Logic is a well-known authentication logic which, despite other more recent logics and formal methods, remains popular with many protocol designers. BAN Logic however does not properly deal with the issues of certificates and the use of Public Key Infrastructure (PKI). This paper proposes an extension to BAN Logic which focuses on certificate processing within the PKI setting. Our extension is along the lines of the work by Gaarder and Sneekenes but better captures current aspects of PKI. In particular, our extension redresses the reasoning on the goodness of private keys, and considers certificate revocation. Common pitfalls in public-key based protocol design are due to insufficient attention placed on the “intended recipient” as well as the “stated sender” of a message. Our extension makes the recipient and sender explicit, which reduces the likelihood of introducing such flaws into the protocol and its subsequent proof using BAN Logic. In summary, our logic is primarily focused on making BAN Logic more concise yet practical to use on PKI-based protocols.

1. Introduction

Designing a correct protocol specification which satisfies certain security properties is well recognized as a non-trivial task. Many logics and formal techniques have been proposed for verifying cryptographic protocols. Among various authentication logics, BAN Logic [7, 8] is one of the best known and most widely used [18, 20, 19]. One reason for its popularity is that BAN Logic is comparatively easy to use. As pointed out by Meadows [18, 19], BAN Logic’s intentional avoidance of many advanced features makes it a simple and straightforward logic that is easy to apply yet of substantial use for detecting flaws. This may well explain the constant appearance of publications applying BAN Logic even till now [3, 5, 21, 25, 10, 9], with application domains as diverse as wireless network [25], mobile com-

munication [9] to voting [21].

BAN Logic however gives a rather simplified treatment of public-key authentication processing. It does not deal with deeper aspects of public-key authentication such as certificate processing, presumably because PKI was not well established when the logic was designed. The situation is now very different since PKI is common, and many modern real-world protocols rely on PKI. There exist some work such as [3, 12, 22] which attempted to extend BAN Logic to better reason with public-key authentication. Among previous work, we find the extension by Gaarder and Sneekenes [12] particularly interesting and useful. In our view, the extension does improve the expressiveness of BAN Logic while keeping the logic’s secret-key aspects intact for easy application. However, it still falls short in capturing many important concepts and practice of modern PKI usage.

In our work, we begin with the starting point of retaining the popularity of BAN Logic among protocol designers. We propose various public-key related enhancements to BAN Logic to allow for more concise reasoning on PKI-based protocols. We address various shortcomings of [12] by redefining as well as introducing a number of rules and structures to better capture PKI usage and its good practice. We also show how our extended BAN Logic can help avoid loopholes in real-world protocol specifications.

Our approach in presenting the results in this paper is ultimately pragmatic. We focus on the logic definition and its usage application while leaving theoretical analysis of the logic, e.g. the logic’s soundness and completeness with respect to some well-defined semantics, as a separate treatment beyond this paper’s scope. It is our goal here to expound an up-to-date yet accessible authentication logic which can be handily used by protocol designers who may not be expert in authentication logic or formal method.

The remainder of this paper is organized as follows. We first survey related work in Section 2. We then give a brief review of the original BAN Logic and the previous extension by Gaarder and Sneekenes in Section 3. Section 4 presents our new extended PKI-based BAN Logic, whereas

Section 5 gives insight on its application. We then discuss related topics in Section 6, and conclude in Section 7.

2 Related Work

There exist various work in the literature which apply formal methods to PKI [3, 12, 22, 17, 15, 14, 23]. We briefly survey work which extends authentication logics, particularly BAN Logic, to deal with public-key authentication. Our focus of comparison will be on: certificate processing, the notion of *time duration*, and rules on messages encrypted (signed) using public (private) keys.

As pointed out by many researchers, such as in [4], the original BAN Logic is known to have limitations in describing “serverless protocols”. In PKI setting, the limitations have to do with accepting the validity of a certificate. This may happen since the only way of promoting “once said” to “believe” is by use of the freshness property of a statement, which is typically in the form of nonce or timestamp. In a serverless protocol, such freshness guarantee however cannot be provided, because the server is not necessarily available at the time of communication. To work around the problem above, the original BAN Logic has chosen to ignore the initial handling of certificates by assuming that they have been previously distributed, checked, and accepted as valid. Aziz and Diffie, who applied BAN Logic in [4], alternatively assume a certificate to always be fresh. Hence, the required belief statements on certificate contents can somehow be derived.

In their work on formal verification of CCITT X.509 protocol [12], Gaarder and Snekkenes argue that important aspects of public-key authentication are lost when BAN Logic is used for protocol verification. To amend this deficiency, they propose enhancements to BAN Logic that take certificate checking into account as an integral part of the reasoning process. The extension defines the notion of *duration* to capture some time-related aspects. A principal can therefore claim a formulae is, was, or will be good in a time interval. In [22], Stubblebine and Wright however argue that the assumptions used are too restrictive for reasoning about long-lived security associations. Additionally, there exist issues on synchronization and synchronization bounds. Nevertheless, the simplicity of the logic proposed in [12], while improving the ability to reason with PKI-based protocols, is appealing. Our work here focuses on reworking the logic to be more accurately in line with current PKI practice.

The work of Syverson [23] also adds time to a logic of authentication. It incorporates a temporal formalism into a semantics model of BAN Logic developed in [2] using temporal notions of “all points in the run prior to the current one” and “at some point in the run prior to the current one.” Here in our work, we adopt the duration model of [12] which is relatively easier to use, yet enables the analysis of

subtle relationships in PKI-based protocols.

Stubblebine and Wright [22] also propose a logic extension for dealing with PKI. The logic supports the concept of synchronization, revocation and recency. In pursuing more expressiveness, it however becomes far more complex than the original BAN Logic. We view that the complexity is a drawback which makes it less likely to be used in practice.

3 BAN Logic and Extension by Gaarder-Snekkenes

Below, we review briefly the original BAN Logic with an emphasis on its notation, logic constructs and inference rules relevant to our extension. We then summarize the extension logic of Gaarder and Snekkenes, and pinpoint some problems with it.

3.1 The Original BAN Logic

BAN Logic [7, 8] is a modal-sorted logic constructed on several sorts of objects: principals, keys, messages and well-formed formulae. Predicate constructs are used to interpret organized objects into well-formed formulae. BAN Logic defines the following constructs:

$P \equiv X$: P believes X ;
$P \triangleleft X$: P sees X ;
$P \sim X$: P once said X ;
$P \Rightarrow X$: P has jurisdiction over X ;
$\#(X)$: X is fresh;
$\{X\}_{K_{PQ}}$: X encrypted with K_{PQ} ;
$P \stackrel{K_{PQ}}{\longleftrightarrow} Q$: P and Q may use a secret-key K_{PQ} .

For more discussion on BAN Logic, see also [13, 18]. Syverson and Cervesato [24] give a tutorial on the logic, and also puts it within a broader context of logics of authentication. For our extended logic in this paper, only the secret-key rules of the original BAN Logic are relevant since we redefine all the public-key related rules later in this paper.

3.2 The Extension by Gaarder-Snekkenes

3.2.1 Extension Summary

New Constructs for Public-Key Formalism

The following logic constructs were defined for public-key authentication (with some slight notational modification on keys):

- $\wp\kappa(P, K_P)$: P has associated a good public key K_P ;
- $\Pi(K_P^{-1})$: P has a good private key K_P^{-1} ;
- $\sigma(X, K_P^{-1})$: X signed with P 's private key K_P^{-1} ;
- $\{X\}_{K_P}$: X encrypted under P 's public key K_P .

In Gaarder-Snekkenes' extension, it is assumed that digital signature is always in *appendix mode*, which we also adopt here. Hence, the signature construct $\sigma(X, K_P^{-1})$ is actually a contracted form of the following:

$$\sigma(X, K_P^{-1}) = X, \text{Sign}(K_P^{-1}, \text{hash}(X)) \quad (1)$$

where: $\text{Sign}()$ is a signature function, and $\text{hash}()$ is a hash construction function.

Certificate and Its Idealization

As mentioned earlier, one of the extension's main contributions is the idealization of a certificate. It adopts the certificate based on the X.509 Standard, whose basic structure can be described as follows:¹

$$\text{Cert}_P = \sigma((N, I, \delta^P, P, K_P, A), K_I^{-1}) \quad (2)$$

where:

- N : unique serial number of the certificate;
- I : name of the issuer;
- δ^P : validity period of the certificate, which consists of: t_1^P (not before) and t_2^P (not after);
- P : the distinguished name of the principal;
- K_P : the certified public key of P ;
- A : identifier of the signature algorithm employed;
- K_I^{-1} : I 's private key which is used to sign the certificate.

Based on the certificate structure, Gaarder and Snekkenes gave a certificate the following idealization:

$$\text{Cert}_P = \sigma((\Theta(t_1^P, t_2^P), \wp\kappa(P, K_P)), K_I^{-1}). \quad (3)$$

In its idealized form, a certificate thus basically consists of two parts: its *validity period* (called duration-stamp in [12]) and *certificate statement*. The construct $(\Theta(t_1, t_2), X)$ was specifically introduced to say that "certificate statement X holds in the time interval (t_1, t_2) ". Hence, the issuer I who uttered the duration-stamped certificate in (3) claims that $\wp\kappa(P, K_P)$ is good in the time interval of t_1^P and t_2^P .

New Inference Rules

To reason with the certificate and public-key constructs, [12] introduced the following inference rules:

- The message meaning (for public-key) Rule:

$$\frac{P \equiv \wp\kappa(Q, K_Q), P \equiv \Pi(K_Q^{-1}), P \triangleleft \sigma(X, K_Q^{-1})}{P \equiv Q \sim X} \quad (4)$$

¹To simplify the formalism, [12] considers the certification path to be of length one. We also take similar approach here.

- The see signed-message Rule:

$$\frac{P \triangleleft \sigma(X, K_Q^{-1})}{P \triangleleft X} \quad (5)$$

- The certificate duration-stamp Rule:

$$\frac{P \equiv Q \sim (\Theta(t_1^R, t_2^R), C^R), P \equiv Q \equiv \Delta(t_1^R, t_2^R)}{P \equiv Q \equiv C^R} \quad (6)$$

In the above rule, Q acts as the issuer of a certificate for R , whose certificate statement is denoted by C^R . The rule thus provides a way of promoting *once said* to *believe* about a certificate statement. For this purpose, P needs to believe that Q holds a belief on "**good time interval**" t_1^R and t_2^R (denoted by $\Delta(t_1^R, t_2^R)$). In other words, P must believe that the validity period of (t_1^R, t_2^R) still holds according to the current time in Q .

Message-Recipient Construct

Gaarder and Snekkenes also introduced a notion of "intended recipient" of a message. They argued that such assurance needs to be explicitly stated as one of the goals in their X.509 protocol formalism. Moreover, they also claimed that the original BAN Logic provides no means of expressing it in the language available. Therefore, the construct " $\mathfrak{R}(X, P)$ " was introduced to say that " P is the intended recipient of message X ".

The construct is defined to appear in the following form:

$$P \rightarrow Q : X, \mathfrak{R}(X, Q) \quad (7)$$

which then should be interpreted as " P sends to Q a particular message X together with a statement telling that Q is the intended recipient of X ". This effort clearly represents a step forward in capturing the notion of *explicit principal naming* [1]. In doing so, however, [12] requires the receiver to hold an assumption about the sender's jurisdiction over message recipient statement (i.e. $Q \equiv P \Rightarrow \mathfrak{R}(X, Q)$), an approach which we will examine more closely below.

3.2.2 Problems and Limitations

Despite its improvements on BAN Logic, there exist some key aspects of the extension in [12] which we find rather unsettling:

- **Assumption on $\Pi(K_Q^{-1})$** : In verifying a protocol using [12], a (supplied) assumption needs to be added to a principal that he/she believes the goodness of the private key of another principal involved. That is, a formula such as " $P \equiv \Pi(K_Q^{-1})$ " needs to be vacuously held at the start of the verification process. Such stipulation is needed so that [12] can process message-meaning Rule (4). We however view the inclusion

of such assumption is an unsound practice. Such formula should never be supplied as an assumption, but rather it must be logically derived from certificate reasoning within the logic. The only assumption needed is the goodness of public and private key of the CA in addition to his/her own. A principal then should be able to derive the goodness of other principal's keys from the certificates issued by the CA. If some keys are no longer to be considered good, revocation mechanisms like Certificate Revocation List (CRL) [11] should then be incorporated in the defined logic rule(s).

- **Assumption for message recipient construct:** One of the consequences of message-recipient construct in [12] is that it requires a statement $\mathfrak{R}(X, P)$ for every “tagged” message to be made as a *protocol goal*. In our opinion, such assurance does not need to be stipulated as a formalism goal since we are mainly interested on deriving goals about the goodness of the generated session key(s). In our view, such construct should instead be incorporated as part of logic rules. Moreover, as already mentioned, the construct requires the receiver to hold an assumption about the sender's jurisdiction over message recipient statement. In our new extension, by integrating the intended-recipient requirement into “message-meaning” rule, we manage to eliminate the need for sender's jurisdiction, thus simplifying the reasoning on message processing.
- **Omission of certificate revocation process:** The work [12] apparently has chosen to ignore the incorporation of certificate revocation issue in their logic. Instead, it assumes that each principal always maintains the goodness of its private key. As a result, we need to believe that a certificate, once issued, is *always good* for the time-interval specified in its validity period. Considering the importance of certificate revocation in public-key authentication, its omission in the formalization represents a limitation of the extension.

4 Our Extended PKI-based BAN Logic

Motivated by the drawbacks of extension in [12], we propose the following extensions to BAN Logic.

4.1 Inclusion of $\Pi(K_P^{-1})$ into Idealized Certificate

In our formalism work, we idealize a certificate's statement as follows:²

$$Cert_P = \sigma((\Theta(t_1^P, t_2^P), \wp\kappa(P, K_P), \Pi(K_P^{-1})), K_I^{-1}) \quad (8)$$

²Note that the *complete* idealized certificate definition will include message-recipient construct as defined in (12).

As can be seen, we now include $\Pi(K_P^{-1})$ into the idealized certificate's statement in contrast with previous definition of (3). Hence, a valid certificate now assures that *both* public and private key of a principal are good. With this modification, we thus eliminate the need to have an assumption about the goodness of other principal's private key as in [12]. This modification is crucial as it helps formalize the close relationship between certificate validity and the goodness a private key. The rule makes it clear that a belief on $\Pi(K_P^{-1})$ should be derived from a valid P 's certificate. Consequently, should the private key of a principal ever be compromised, the principal must notify and ask the CA to revoke his/her certificate, a requirement that is consistent with current PKI practice.

4.2 New Use of Message-Recipient

Different from [12], message-recipient construct is defined in our extension to be part of signature construct (σ), which now appears in the following form:

$$\sigma(\mathfrak{R}(X, P), K_Q^{-1}). \quad (9)$$

Unlike the use of message-recipient construct in [12], we incorporate it into our new Message-meaning Rule (defined below) as one of its premises. A principal thus needs to ensure the existence of a valid recipient tag in the signed message in order to proceed with the rule. With this, we also manage to eliminate the requirement for stating message-recipient as a verification goal, as well as the requirement for introducing an assumption about sender's jurisdiction over message-recipient as in [12].

4.3 New Message-Meaning Rule (Private-Key Signed Message)

In symmetric-key based authentication, the intended recipient of a message can usually be inferred from the shared secret-key employed in the encryption or Message Authentication Code (MAC) generation. Unfortunately, this does not apply in private-key signed messages where the same private key is used to sign messages regardless of their intended recipient. Hence, in public-key authentication, there is a greater need to follow the “naming principle” [1].³ Surprisingly, the same mistake due to disregarding this principle seems to be made time after time in many published protocols (see [16]). Given this concern, we redefine the “Message-meaning for signed-message” Rule as follows:

$$\frac{P \equiv \wp\kappa(Q, K_Q), P \equiv \Pi(K_Q^{-1}), P \triangleleft \sigma(\mathfrak{R}(X, P), K_Q^{-1})}{P \equiv Q \vdash X}. \quad (10)$$

³It is possible to argue that in a few situations, due to privacy considerations, protocol designers might aim to withhold identity information as long as possible; thus conflicting with the naming principle. We however focus here on general situations where ensuring secure authentication take higher precedence over privacy concerns.

With (10), we thus integrate message-recipient construct into the message-meaning rule. Our requirement for the third premise above is strongly motivated by the work of Meadows [18] and Boyd and Mathuria [6], which still managed to successfully find a loophole in Aziz-Diffie protocol [AD94] despite the application of BAN Logic to the proposed protocol. We will examine this later in Section 5 where we show how the new rule and some cautionary note on BAN Logic can help pinpoint the problem with the protocol and its flawed “proof”.

4.4 All-Recipient See Rule

Having defined the new message-meaning rule as above, we further note that it is possible for a message to be actually intended for *all* principals in the protocol. A good example is a certificate, which is meant to be accepted by any principal as long as he/she trusts the issuer. We thus define a special principal name called “*all*”, and define the the following rule:

$$\frac{P \triangleleft \sigma(\mathfrak{R}(X, all), K_Q^{-1})}{P \triangleleft \sigma(\mathfrak{R}(X, P), K_Q^{-1})}. \quad (11)$$

4.5 Certificate and Certificate-Validation

In line with the use of *all recipient* definition above, a certificate is now to be idealized as follows:

$$Cert_P = \sigma(\mathfrak{R}((\Theta(t_1^P, t_2^P), \wp\kappa(P, K_P), \Pi(K_P^{-1})), all), K_I^{-1}). \quad (12)$$

This certificate definition now correctly includes the message-recipient construct, and subsumes the previous interim definition given in (8).

To derive a belief on a certificate, this Certificate Validation Rule is used:

$$\frac{P \equiv Q \sim (\Theta(t_1^R, t_2^R), C^R), P \equiv Q \equiv \Delta(t_1^R, t_2^R), P \equiv Q \equiv \Phi(C^R)}{P \equiv Q \equiv C^R}. \quad (13)$$

Here C^R denotes a certificate statement, consisting of $\wp\kappa(P, K_P), \Pi(K_P^{-1})$. This rule thus supersedes rule (6) previously defined in [12]. Our addition of the third premise is done to emphasize the need for “*certificate revalidation step*” before deriving any belief of a certificate. P must ensure the premise by checking that Q still believes that the uttered certificate statement remains valid ($Q \equiv \Phi(C^R)$).⁴ In the CRL model, this step is done by checking the absence of the certificate in question in Q ’s recent CRL.

In the rule above, we note that the resulting belief statement (C^R) can be argued as an “unstable” statement [8]. That is, the statement is valid only at the time of validation but not necessarily thereafter, as the corresponding certificate might be revoked at some point of time in the future.

⁴Note that although we put C^R as the parameter of $\Phi()$, in practice the matching is done based on the unique certificate’s serial number N .

A more elaborate logic would include a more general time-related reasoning, an approach that is taken for example by [22]. To keep our extension simple, however, we avoid doing so. In fact, both BAN Logic [8] and [12] implicitly made a similar simplification with respect to beliefs derived from secret-key reasoning, as in practice, a secret key will eventually cease to be valid due to expired lifetime or a possible security breach.

4.6 Message-Sender Construct

Realizing the important of naming principle, we take another step to additionally define a *message-sender* construct. The construct introduces the notion of “stated sender” of a message, and is defined to appear in one of the following forms:

$$\{\mathcal{S}(X, Q)\}_{K_P} \quad \text{or} \quad \mathcal{S}(\{X\}_{K_P}, Q). \quad (14)$$

$\mathcal{S}(X, Q)$ specifically says “message X together with Q as the stated sender of the message”. The first form occurs when the encryption is employed with an additional function of authentication.⁵ P , who receives the message from Q , ensures this construct by first decrypting the message, and then ensuring that it correctly contains Q as the sender ID together with X . The second form takes place when the encrypted message ($\{X\}_{K_P}$) and the clear sender ID (Q) come in a message signed by Q (see rule (16) below).

4.7 New Message-Meaning Rule (for Public-Key Encryption Message)

In the case of a private-key signed message, it is important to ensure the intended recipient of the message. When dealing with a message encrypted with the public key of a recipient, it is the *identity of the sender* that matters.

To capture two possible forms of message-sender, we define two following rules of message-meaning for public-key encryption message:

$$\frac{P \equiv \wp\kappa(P, K_P), P \equiv \Pi(K_P^{-1}), P \triangleleft \{\mathcal{S}(X, Q)\}_{K_P}}{P \equiv Q \sim X} \quad (15)$$

$$\frac{P \equiv \wp\kappa(P, K_P), P \equiv \Pi(K_P^{-1}), P \equiv Q \sim \mathcal{S}(\{X\}_{K_P}, Q)}{P \equiv Q \sim X}. \quad (16)$$

Rule (15) deals with situation where the stated sender is concealed within the encrypted message. We later show in Section 5 how this rule could have helped deal with a loophole in Needham-Schroeder Public-Key protocol whose attack was outlined by Lowe [16]. The rule (16) is to be employed where P has previously seen both $\{X\}_{K_P}$ and Q , perhaps as parts of longer message statement, in a signed message previously validated using rule (10).⁶

⁵It is important to make clear of the role of encryption in a protocol specification (see [1]).

⁶In this case of signed encrypted message, both message-recipient and

4.8 Redefined Message-Meaning Rule for Secret-Key

For completeness, we redefine here a new construct for message authentication using secret-key based MAC:

$$\mu(X, K_{PQ}) = X, H(K_{PQ}, X). \quad (17)$$

Similar to signature construction in (1), MAC generation is performed by applying a chosen keyed hash-construction function $H()$ to message X using K_{PQ} , and then appending the resulting hash-image to X . The related *message meaning (for secret-key using MAC) Rule* is defined as follows:

$$\frac{P \equiv Q \xleftrightarrow{K_{PQ}} P, P \triangleleft \mu(X, K_{PQ})}{P \equiv Q \sim X}. \quad (18)$$

Here we omit the requirement for intended recipient and stated sender by assuming that the shared secret key is good and the message X is unambiguously formatted.

4.9 Additional Rules for See Operator

- The **See hashed-message** Rule:

$$\frac{P \triangleleft \mu(X, K_{PQ})}{P \triangleleft X} \quad (19)$$

- The **See signed-message** Rule:

$$\frac{P \triangleleft \sigma(X, K_Q^{-1})}{P \triangleleft X} \quad (20)$$

- The **See recipient-tagged-message** Rule:

$$\frac{P \triangleleft \mathfrak{R}(X, P)}{P \triangleleft X} \quad (21)$$

- The **See sender-tagged-message** Rule:

$$\frac{P \triangleleft \mathfrak{S}(X, Q)}{P \triangleleft X} \quad (22)$$

5 Using the New Extended Logic

We now show how our extension could have helped prevent problems in flawed published protocols.

5.1 Needham-Schroeder Public-Key Authentication Protocol

In [16], Lowe published an attack on Needham-Schroeder public-key protocol. The protocol proceeds as follows:

1. $A \rightarrow S : A, B$
2. $S \rightarrow A : \{K_B, B\}_{K_S^{-1}}$

message-sender are checked. Although it looks rather strict, here we opt to do so as to help non-specialist protocol designers to avoid any unforeseen “small” mistake, which can prove costly once it is found. Alternatively, the extension logic can be made to relax this requirement.

3. $A \rightarrow B : \{N_A, A\}_{K_B}$
4. $B \rightarrow S : B, A$
5. $S \rightarrow B : \{K_A, A\}_{K_S^{-1}}$
6. $B \rightarrow A : \{N_A, N_B\}_{K_A}$
7. $A \rightarrow B : \{N_A\}_{K_B}$.

Despite the assumption that each principal has each other’s public key correctly, Lowe managed to find an attack on the protocol. The problem with the protocol has to do with the encryption using public key of the recipient without clear identity of the sender. Lowe proposed the modification of message 6 into:

$$6'. B \rightarrow A : \{B, N_A, N_B\}_{K_A}.$$

In our new logic, with our new message-meaning (for public-key encrypted message) rule, the derivation of the flawed beliefs would then be impossible. This is the case since the requirement of $\mathcal{S}(\dots, B)$ is unfulfilled for message 6. This example highlights the value of integrating message-sender construct into message-meaning rule.

5.2 Aziz-Diffie Protocol

We analyze below the protocol by Aziz and Diffie [4] which was still broken despite the use of the original BAN Logic by the authors to verify it.

The protocol uses public-key cryptography for securing the wireless link between a Mobile (M) and a Base (B). In the following, *alg_list* denotes a list of flags representing potential secret-key algorithms chosen by M . The flag *sel_alg* represents the particular algorithm selected by B from the list *alg_list*, and is to be employed to encrypt the subsequent data call. The protocol for providing the connection setup between M and B is as follows ([6]):

1. $M \rightarrow B : Cert(M), N_M, alg_list$
2. $B \rightarrow M : Cert(B), \{X_B\}_{K_M}, sel_alg, \{hash(\{X_B\}_{K_M}, sel_alg, N_M, alg_list)\}_{K_B^{-1}}$
3. $M \rightarrow B : \{X_M\}_{K_B}, \{hash(\{X_M\}_{K_B}, \{X_B\}_{K_M})\}_{K_M^{-1}}$

Here N_M is a nonce from M . X_B and X_M denote the particular session key values chosen by B and M , respectively. The final session key x is calculated as $X_M \oplus X_B$.

The protocol was verified in [4] using BAN Logic. In our analysis, the given proof apparently contains a serious flaw. The flaw is introduced in the error-prone idealization step of the formalism. [4] idealized message 2 as: $\{\{\xrightarrow{K_B} B\}_{K_{C_a}^{-1}}, M \xrightarrow{X_B} B, N_M\}_{K_B^{-1}}$. The problem with this is that what can actually be derived from the message is $\{M \xrightarrow{X_B} B\}_{K_M}$, and not $M \xrightarrow{X_B} B$. This formalism pitfall allowed [4] to incorrectly derive the desired goals despite the loophole in the protocol.

Subsequently, both [18] and [6] managed to mount an attack on the protocol. The attack outlined in [6] makes use of two parallel open sessions. Impersonating M in the first session, an attacker C is able to obtain $\{X_B\}_{K_M}$ from the

message 2. In the second session, C then replays $\{X_B\}_{K_M}$ to the initiating M when it plays a role as Base. [6] correctly pointed out the source of the problem is that C can construct message 2 without the knowledge of X_B . To fix the protocol, [6] proposed the modification of message 2 and 3 into:

$$\begin{aligned} 2'. B \rightarrow M : & \text{Cert}(B), N_B, \{X_B\}_{K_M}, \text{sel_alg}, \\ & \{ \text{hash}(X_B, M, N_M, \text{alg_list}) \}_{K_B^{-1}} \\ 3'. M \rightarrow B : & \{X_M\}_{K_B}, \{ \text{hash}(X_M, B, N_B) \}_{K_M^{-1}} \end{aligned}$$

Nonce N_B now provides freshness assurance, taking the role of $\{X_B\}_{K_M}$ in the original protocol. Note that message 2' contains M in the signed hash's arguments. Likewise, message 3' now contains B .⁷ Such inclusions are thus in line with our requirement of *message-recipient* in message meaning rule (for signed message), highlighting the need for such assurance in robust protocol design.

In message 2', $\{X_B\}_{K_M}$ is however sent without sender-tag assurance, seemingly playing down the *message-sender* requirement in encrypted message. However, we can notice that X_B , instead of $\{X_B\}_{K_M}$, is now part of the signed message portion. Thus, after decrypting $\{X_B\}_{K_M}$ into X_B , M is required to check that the hash is correctly constructed with X_B as its input, and is subsequently signed by B . This provides B 's "authorship" (sender-tag) assurance on X_B . However, since designing a protocol is an error-prone activity, we opt to make the assurance explicit, either by suggesting $\{X_B, B\}_{K_M}$, or adding B in the signed hash portion to enable rule (16). Such inclusion will increase the protocol assurance while incurring small extra overheads.

6 Discussion

We have presented an extension of BAN Logic which deals with PKI. It addresses a number of issues in [12] which are vital to a more accurate reasoning with certificate-based protocols. In summary, our main contributions are:

- We present an improved idealization of certificate, in which the assurance of a private key is also derived from certificate. This eliminates the need to have an assumption about the goodness of the other principal's private key as in [12].
- We define a new message-meaning for the private-key signed message rule, which contains the message-recipient construct. By doing so, we manage to eliminate the requirement for stating message-recipient as a goal, as well as for introducing an assumption about sender's jurisdiction over message-recipient as in [12].

⁷Although M and B are not included in clear portion of message 2' and 3' respectively, they are actually present from the message transfer context. As such, we should take their presence into account in our idealized protocol and verification.

- We also modify the certificate-validation rule, which now includes the third premise to highlight the need for certificate revalidation step. In CRL model, the new rule thus makes explicit of two requirements in certificate validation: time synchronization with certificate issuer, and the check with issuer's recent CRL.
- We define the message-meaning rule for the public-key encrypted message which now requires message-sender construct. This modification is vital as it prevents a common mistake in public-key protocol design, as clearly illustrated among others by Lowe's attack on Needham-Schroeder public-key protocol [16].

Although some of the modifications above may look simple, they are however crucial for better reasoning with PKI-based protocols. Table 1 contrasts several constructs and rules from [12] with our new extension.

Lastly, we also would like to address a subtle issue in BAN Logic, namely the reasoning of: (i) *signed encrypted message*: a signed message contains encrypted message portion(s); and (ii) *encrypted signed message*: a signed message is sent encrypted.

We already consider case (i) when defining rule (16). To be more complete, we can additionally define:

$$\frac{P \models \wp \kappa(P, K_P), P \models \Pi(K_P^{-1}), P \models Q \models \mathcal{S}(\{X\}_{K_P}, Q)}{P \models Q \models X}. \quad (23)$$

This rule applies when the freshness assurance comes in the signed message rather than the encrypted portion. Hence, we simply derive a belief about the encrypted message. For case (ii), we can similarly define:

$$\frac{P \models \wp \kappa(Q, K_Q), P \models \Pi(K_Q^{-1}), P \models \sigma(\mathfrak{R}(X, P), K_Q^{-1})}{P \models Q \models X}. \quad (24)$$

7 Conclusion

We have presented our Extended BAN Logic built upon the previous work by Gaarder and Snekkenes [12] for better reasoning with certificate-based public-key authentication. Our extensions removes various limitations of [12] to make the logic more in line with concepts and practice in modern PKI setting. Examples on the usage of our extension are given. These help prevent common mistakes in public-key protocol design and verification. Given that BAN Logic is a well-understood and popular logic, we envisage that our extension provides a practical and valuable tool without requiring the users to manipulate substantially complex formalism. It is indeed our aim to make formal analysis on PKI-based protocols become more handily accessible to a wider range of protocol designers, thus allowing more parties to improve the security of their protocols.

Category	Gaarder-Sneekenes' Extension [12]	Our Extended Logic
Idealized certificate	$\sigma((\Theta(t_1^P, t_2^P), \wp\kappa(P, K_P)), K_I^{-1})$	$\sigma(\mathfrak{R}((\Theta(t_1^P, t_2^P), \wp\kappa(P, K_P), \Pi(K_P^{-1})), all), K_I^{-1})$
Certificate-validation	$\frac{P \models Q \sim (\Theta(t_1^R, t_2^R), C^R), P \models Q \models \Delta(t_1^R, t_2^R)}{P \models Q \models C^R}$	$\frac{P \models Q \sim (\Theta(t_1^R, t_2^R), C^R), P \models Q \models \Delta(t_1^R, t_2^R), P \models Q \models \Phi(C^R)}{P \models Q \models C^R}$
Message-recipient	$\mathfrak{R}(X, P)$	$\sigma(\mathfrak{R}(X, P), K_Q^{-1})$
Stated-sender	—	$\{S(X, Q)\}_{K_P}$ or $S(\{X\}_{K_P}, Q)$
Msg-meaning (signed)	$\frac{P \models \wp\kappa(Q, K_Q), P \models \Pi(K_Q^{-1}), P \triangleleft \sigma(X, K_Q^{-1})}{P \models Q \sim X}$	$\frac{P \models \wp\kappa(Q, K_Q), P \models \Pi(K_Q^{-1}), P \triangleleft \sigma(\mathfrak{R}(X, P), K_Q^{-1})}{P \models Q \sim X}$
Msg-meaning (encrypted)	—	$\frac{P \models \wp\kappa(P, K_P), P \models \Pi(K_P^{-1}), P \triangleleft \{S(X, Q)\}_{K_P}}{P \models Q \sim X}$ $\frac{P \models \wp\kappa(P, K_P), P \models \Pi(K_P^{-1}), P \models Q \sim S(\{X\}_{K_P}, Q)}{P \models Q \sim X}$

Table 1. Comparison of relevant constructs and rules from [12] and our new extension.

References

- [1] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, 1996.
- [2] M. Abadi and M. R. Tuttle. A semantics for a logic of authentication. In *ACM Symposium on Principles of Distributed Computing (PODC)*, 1991.
- [3] N. Agray, W. van der Hoek, and E. P. de Vink. On BAN logics for industrial security protocols. In *From Theory to Practice in Multi-Agent Systems*. LNAI Vol. 2296, 2002.
- [4] A. Aziz and W. Diffie. Privacy and authentication for wireless local area networks. *IEEE Personal Communication*, 1(1):25–31, 1994.
- [5] K. Bacakci and N. Baykal. One-time passwords: security analysis using BAN logic and integrating with smartcard authentication. In *International Symposium on Computer and Information Sciences (ISCIS)*. LNCS Vol. 2869, 2003.
- [6] C. Boyd and A. Mathuria. Key establishment protocols for secure mobile communications: a selective survey. In *Australasian Conference on Information Security and Privacy (ACISP)*. LNCS Vol. 1438, 1998.
- [7] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *Proceedings of the Royal Society*, 426(1871), 1989.
- [8] M. Burrows, M. Abadi, and R. Needham. *A logic of authentication, revised*. SRC Technical Report 39. Digital Systems Research Centre, 1990.
- [9] C. Chang, H. Pan, and H. Jia. A secure short message communication protocol. *International Journal of Automation and Computing*, 5(2):202–207, 2008.
- [10] L. Chen, G. Zhang, and X. Li. Efficient identity authentication protocol and its formal analysis. In *International Conference on Computational Intelligence and Security Workshops (CISW)*, 2007.
- [11] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. *Internet X.509 Public Key Infrastructure certificate and Certificate Revocation List (CRL) profile*. IETF RFC 5280, 2008.
- [12] K. Gaarder and E. Sneekenes. Applying a formal analysis technique to the CCITT X.509 strong two-way authentication protocol. *Journal of Cryptology*, 3(2):81–98, 1991.
- [13] S. Gritzalis, D. Spinellis, and P. Georgiadis. Security protocols over open networks and distributed systems: formal methods for their analysis, design, and verification. *Computer Communications*, 22(8):697–709, 1999.
- [14] J. Howell and D. Kotz. A formal semantics for SPKI. In *European Symposium on Research in Computer Security (ESORICS)*. LNCS Vol. 1895, 2000.
- [15] R. Kohlas and U. Maurer. Reasoning about public-key certification: on bindings between entities and public keys. *Journal on Selected Areas in Communications*, 18:551–560, 2000.
- [16] G. Lowe. Some new attacks upon security protocols. In *IEEE Computer Security Foundations Workshop*, 1996.
- [17] U. Maurer. Modelling a public-key infrastructure. In *European Symposium on Research in Computer Security (ESORICS)*. LNCS Vol. 1146, 1996.
- [18] C. Meadows. Formal verification of cryptographic protocols: a survey. In *Advances in Cryptology - Asiacrypt'94*. LNCS Vol. 917, 1994.
- [19] C. Meadows. Formal methods for cryptographic protocol analysis: emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, 21(1), 2003.
- [20] B. Schneier. *Applied cryptography: protocols, algorithms, and source code in C*. Wiley, New York, 2nd edition, 1996.
- [21] T. Storer, U. Martin, and I. Duncan. *BAN logic analysis of the UK postal voting system*. Research report. University of St. Andrews, 2003.
- [22] S. Stubblebine and R. Wright. An authentication logic with formal semantics supporting synchronization, revocation, and recency. *IEEE Transactions on Software Engineering*, 28(3):256–285, 2002.
- [23] P. Syverson. Adding time to a logic of authentication. In *ACM Conference on Computer and Communications Security (CCS)*, 1993.
- [24] P. Syverson and I. Cervesato. The logic of authentication protocols. In *Foundations of Security Analysis and Design*. LNCS Vol. 2171, 2001.
- [25] S. Xu and C. Huang. Attacks on PKM protocols of IEEE 802.16 and its later versions. In *International Symposium on Wireless Communication Systems (ISWCS)*, 2006.