BAN logic, the ‘mother of all authentication logics’, does not model hash functions in an appropriate way. It is possible to derive false statements from true assumptions, therefore, BAN logic is not ‘sound’ – even without a semantics. In the (limited and heavily debated) semantics that BAN logic has, this problem also shows up.

Chapter 5

‘Unsoundness’ of BAN logic

In this chapter we show a problem of BAN logic [BAN89b, BAN89a][1] that has, to our knowledge, not yet been identified, despite all research into formal protocol analysis. The problem is this: BAN logic is not ‘sound’. False statements can be obtained by ‘valid’ inference rules from true assumptions. This behavior is caused by a questionable inference rule. In Section 5.1 we will explain the reasoning mistake behind this questionable inference rule. As a result of the reasoning mistake, the inference rule does not have a *computational justification*, which is discussed in Section 5.2. Section 5.3 shows the protocol we use in our unsoundness proof and Sect. 5.4 shows all inference rules used in our proof. Section 5.5 shows the actual proof. In Sect. 5.6 we will give an alternative proof, but in the questionable semantics of BAN logic; therefore, we regard our proof of Sect. 5.5 more important. We close with some remarks on the relevance of our results.

5.1 Cryptographic Hash Functions and Justified Beliefs

A cryptographic hash function is a function $H\colon \{0,1\}^* \to \{0,1\}^k$ which is computationally feasible to compute, but for which the inverse is computationally infeasible. In particular, computing the inverse of a hash function takes $O(2^k)$ operations. Thus, a cryptographic hash function is *one-way*: it is computationally infeasible to construct a message x such that $H(x)$ yields a given value h [DH76]. For an extensive treatment of cryptographic hash functions, consult Chapter 3.

[1] If you are unfamiliar with BAN logic, you may wish to consult the previous chapter first.

We repeat the most relevant properties of cryptographic hash functions from Chapter 3 here. Cryptographic hash functions have a lot of applications, including password protection, manipulation detection and the optimization of digital signature schemes. Unfortunately however, the class of applications is sometimes overestimated. Consider for example the following quote from security expert Bruce Schneier [Sch96, page 31] (also quoted on page 43):

> "If you want to verify someone has a particular file (that you also have), but you don't want him to send it to you, then you ask him for the hash value. If he sends you the correct hash value, then it is almost certain that he has that file."

Unfortunately, this claim is false. The problem is that in the above situation sketch, there is no mention that the hash value should be kept totally secret. If there is somebody who is willing to publish the hash value of the file, anybody can 'prove' possession of the file.

The authors of BAN logic [BAN89b, BAN89a] made the same reasoning mistake as Bruce Schneier, and incorporated into their logic an inference rule reflecting the abovementioned questionable reasoning[2]. The name of the questionable rule is **H-BAN** and the rule will be shown in Sect. 5.2 on page 58. As a result of this, BAN logic is not 'sound'. Essential in our proof is the fact that belief in BAN logic is considered to be *justified belief*.

But first, let us recapitulate what soundness is. A proof procedure is sound if it proves only valid formulae. In particular, from justified ('true') formulae it should be impossible to infer an unjustifiable ('false') formula. A proof of soundness generally involves a formal system and a class of models (a *semantics*): a proof of soundness essentially shows that every formula that is *derivable* ($\vdash$) in the formal system is *observable* ($\models$) in all relevant models (i.e., $S \vdash X$ implies $s \models X$).

A related concept, 'soundness'[3] ($S \vdash P \mid\equiv X$ implies $S \vdash X$) relies on the definition of the modal operator *belief* ($\mid\equiv$) in BAN logic which denotes *true justified belief*. As opposed to beliefs in general, which may be ungrounded and false, a true justified belief should be true. To see what the authors of BAN logic consider belief, let us look at the following excerpt from [BAN94, page 7]:

> "More precisely, define knowledge as truth in all states (as in [HM84][4]); our notion of belief is a rudimentary approximation to knowledge, and it is simple to see that if all initial beliefs are knowledge then all final beliefs are knowledge and, in particular, they are true."

In this chapter, we will prove 'unsoundness' in Section 5.5 and unsoundness in Section 5.6. In our 'unsoundness' proof, all initial beliefs are clearly

[2] See Appendix A.1 for a detailed discussion of the papers presenting BAN logic, and which papers exactly contain the reasoning mistake.

[3] Note the quotes, which distinguish 'soundness' from soundness.

[4] This is a reference to a preliminary paper. The final paper is [HM90] — WT.

knowledge, though one of the obtained final beliefs is not knowledge, in particular, it is false. Thus, by inferring an unjustified belief in BAN logic from true assumptions, we prove that BAN logic is not sound. In particular, this means that it is impossible to create a semantics in which BAN logic is sound.

5.2 On the Computational Justification of Beliefs

In the analysis of security protocols, if a principal obtains a new belief, there has to be a computational justification for the newly obtained belief. For example, if a principal sees a message cryptographically signed with private key K^{-1}, it is justified to believe that the message originates from the principal owning private key K^{-1}. The computational justification is in this case that it is computationally infeasible for principals other than the one owning private key K^{-1} to construct a message signed with this key. This type of justification is *essential* if security is of concern.[5]

With this consideration in mind, it is worth noting the following excerpt from page 266 of the BAN paper [BAN89b], (resp. pages 41–42 of [BAN89a]):

> "Obviously, trust in protocols that use hash functions is not always warranted. If H is an arbitrary function, nothing convinces one that when A has uttered $H(m)$ he must have also uttered m. In fact, A may never have seen m. This may happen, for instance, if the author of m gave $H(m)$ to A, who signed it and sent it. This is similar to the way in which a manager signs a document presented by a subordinate without reading the details of the document. However, the manager expects anyone receiving this signed document to behave as though the manager had full knowledge of the contents. Thus, provided the manager is not careless and the hash function is suitable, signing a hash value should be considered the same as signing the entire message."

This quote contains an assumption which is, in our opinion, unreasonable: The manager expects anyone receiving the signed document as though something would be the case *which may not be the case*. Of course, any principal including the manager may be free to desire any behavior from other principals. But is it reasonable to expect beliefs to be obtained which are not computationally justified?

It is reasonable to assume that any principal, upon seeing $\{H(N)\}_{K^{-1}}$ will believe the manager signed $H(N)$, since it is computationally too difficult for any principal *other than the manager* to construct the signature. However, it is not reasonable to assume that any principal, upon believing a manager signed $H(N)$, believes that the manager has seen N, as there is *no computational problem*

[5] Consider the alternative: we do not want principals to believe a message is sent by Santa Claus just because the name 'Santa Claus' is written beneath it; writing the name 'Santa Claus' is an exercise just as easy for Santa Claus himself as it is for anybody else.

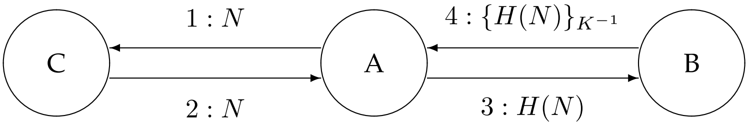

FIGURE 5.1: The two parrots protocol, graphical illustration.

that would justify such a belief. Anybody may have computed $H(N)$ from N, in particular someone may have told the manager $H(N)$ but not N. Therefore, the expectation of a manager that other principals should act as if the manager knows N, is not warranted.

In fact, the text quoted above is the justification of the inference rule **H-BAN** in BAN logic. We believe the identified problematic assumption explains the problems that arise from the inference rule **H-BAN**.

The **H-BAN** *hashing* inference rule reads, as given on page 266 of [BAN89b] (resp. page 42 of [BAN89a])[6]:

H-BAN $$\frac{P \models Q \mid\sim H(X), \quad P \lhd X}{P \models Q \mid\sim X}$$

This rule is problematic, as it essentially infers belief (by P) of "possession" (by Q) of the message X from P believing that Q once conveyed $H(X)$. This rule leads to the 'unsoundness' of BAN logic. Fortunately, none of the authentication logics that descend from BAN logic, adopts the **H-BAN** inference rule.

Because the most commonly used signature schemes use cryptographic hash functions, the **H-BAN** inference rule was added to BAN logic to facilitate the analysis of such signature schemes.

With this inference rule at hand, we can see how the two parrots protocol demonstrates the 'unsoundness' of BAN logic.

5.3 The Two Parrots Protocol

To prove the 'unsoundness' of BAN logic, we rely on a protocol. The rather simple *two parrots protocol*[7], shown in shown in Figures 5.1 and 5.2, will demonstrate the 'unsoundness'. Alice (denoted A) chooses a random number N, sends it to Cecil (denoted C), who returns the number. Then Alice sends the cryptographic hash of the number to Bob (denoted B), and Bob signs this hash value and returns it to Alice. As Bob only sees the cryptographic hash value of N, and a cryptographic hash function is one-way, Bob does not learn N itself. Of course, Cecil might privately disclose N to Bob, but this does not happen

[6] If one would like to add inference rule **H-BAN** to GNY, albeit just for demonstration purposes, one could use exactly the same notation, as the formal languages of BAN logic and GNY logic coincide for the constructs used in this particular inference rule.

[7] The *two parrots protocol* is a variation on the *signing parrot protocol*, which was presented in Chapter 4, Table 4.1.

in the two parrots protocol. Thus, though by private channels Bob might learn N, the protocol certainly does not guarantee this.

Alice cannot, as a result of the protocol, conclude that Bob knows N. Neither can Alice conclude that Bob *does not* know N. However, according to the analysis of the two parrots protocol in BAN logic, Alice will believe that Bob knows N.

In the two parrots protocol, the message N is transmitted without protection. Thus, one can argue that Bob could learn N by mere eavesdropping. For the sake of simplicity, we use a *very simple protocol* that suffices to demonstrate our observation on BAN logic. Of course, protection of N can be achieved by encryption of the messages between Alice and Cecil. Our proof can be easily extended to obtain the same result for such an altered protocol. Moreover, our proof does not rely on Bob eavesdropping.

Thus, though Bob could learn N through either an assistant (Cecil disclosing N to Bob) or through eavesdropping, the communication in the two parrots protocol simply does not warrant Bob knowing N, and therefore also does not warrant Alice believing that Bob knows N.

When we want to formally analyze the protocol in BAN logic, we need to *transcribe* it into BAN logic. A summary of the protocol transcription is given in Figure 5.3. For illustrative purposes, we will also give the transcription into GNY logic in Figure 5.4.[8] First, we have the protocol assumptions which state that A knows the public key K of B, A knows N, and A believes N to be *fresh*. A newly generated random number is particularly fresh.

The protocol description itself is rather straightforward. To quickly see how the two parrots protocol interacts with the **H-BAN** inference rule, observe that message 2 ($C \rightarrow A\colon N$) can be used to obtain the second precondition of **H-BAN**, and that message 4 ($B \rightarrow A\colon \{H(N)\}_{K^{-1}}$) can be used to obtain the first precondition of **H-BAN**. Thus, messages 2 and 4 are the essential messages of the protocol. The other messages can be considered mere 'glue'.

What is achieved by a protocol can be stated in *claims*. For the two parrots protocol, the following claim is true:

$$\text{It will not be the case that } B \mid\equiv N$$

which essentially states that B will not know N. Note that this is true because

1. B only sees $H(N)$,
2. the inverse of $H(\cdot)$ is hard to compute ($H(\cdot)$ is a one-way function), and
3. B has only polynomially many computational resources.

[8] Note how idealization in BAN logic (Figure 5.3) differs from the idealization in GNY logic as given in Figure 5.4. The assumptions $A \ni +K$ ("A knows the public key $+K$") and $B \ni -K$ ("B knows his own private key $-K$") are omitted, as within BAN logic this is implied by $A \mid\equiv \overset{+K}{\mapsto} B$. In fact BAN logic does not even explicitly name public and private keys individually. Therefore, the signed message in protocol step 2, has the form $\{H(N)\}_{K^{-1}}$ (BAN logic) instead of $\{H(N)\}_{-K}$ (GNY logic). Moreover, as BAN does not distinguish between possession ($\ni$) and belief ($\mid\equiv$), the assumptions and claims are rewritten accordingly. ($A \ni N$ in GNY logic is $A \mid\equiv N$ in BAN logic; $A \mid\equiv B \ni N$ in GNY logic is $A \mid\equiv B \mid\equiv N$ in BAN logic.)

assumptions Alice knows Bob's public key, and Bob knows his own secret key.

the protocol itself Alice chooses a random number and sends it to Cecil. Cecil sends this very same number back to Alice. Alice computes the (cryptographic) hash value of this number and send the hash value to Bob. Bob signs the hash value and sends it back to Alice.

claims Alice knows that knows Bob received her message containing the hash value of the random number. Bob does not know the random number itself by means of the protocol, though Bob might learn it by other means (e.g. Cecil tells Bob the number in private). Alice has no stance on whether Bob knows the random number.

FIGURE 5.2: The two parrots protocol, plain description

---

assumptions $A \mid\equiv \overset{K}{\mapsto} B$, $A \mid\equiv N$, $A \mid\equiv \sharp(N)$

the protocol itself

1. $A \rightarrow C\colon N$
2. $C \rightarrow A\colon N$
3. $A \rightarrow B\colon H(N)$
4. $B \rightarrow A\colon \{H(N)\}_{K^{-1}}$

claims It will not be the case that $B \mid\equiv N$.

problem $A \mid\equiv B \mid\equiv N$ can be inferred.

FIGURE 5.3: BAN idealization of the two parrots protocol. In general, the message X cryptographically signed with the private key corresponding to public key K is denoted as $\{X\}_{K^{-1}}$. Thus, any agent that knows K can verify the signature and read X. The assumptions are the *true premises* that lead to the *false belief* which is shown under 'problem'.

---

assumptions $A \ni +K$, $A \mid\equiv \overset{+K}{\mapsto} B$, $B \ni -K$, $A \ni N$, $A \mid\equiv \sharp(N)$

the protocol itself

1. S_1 $A \rightarrow C\colon N$
2. S_2 $C \rightarrow A\colon N$
3. S_3 $A \rightarrow B\colon H(N)$
4. S_4 $B \rightarrow A\colon \{H(N)\}_{-K}$

claims $A \mid\equiv B \ni H(N)$ and it will not be the case that $B \ni N$.

remark If rule **H-BAN** would be added to GNY logic, $A \mid\equiv B \ni N$ would be inferrable, which is undesirable.

FIGURE 5.4: GNY idealization of the two parrots protocol. To verify that, if **H-BAN** is adopted, a legal annotation of this protocol exists where $A \mid\equiv B \ni N$ is inferred, see that the protocol is highly similar to the signing parrot protocol shown in Figure 4.2, and a legal annotation can be derived by adaptation of Figure 4.4.

The problem that we identify in BAN logic (see Sect. 5.5) has the effect that due to inference rule **H-BAN** the following statement can also be inferred in BAN logic:

$$A \models B \models N$$

which states that A will believe that B will know N. This belief of A is not computationally justified (see Sect. 5.2).

5.4 Used Inference Rules

The proof of 'unsoundness' in Sect. 5.5 involves three inference rules of BAN logic[9]. Inference rule **H-BAN** has already been given on page 58, the other two rules are:

1. the *message meaning* inference rule number *ii* as given on page 238 of [BAN89b] (resp. page 6 of [BAN89a])[10]:

   MM $$\frac{P \models \overset{K}{\mapsto} Q, \quad P \triangleleft \{X\}_{K^{-1}}}{P \models Q \mid\!\sim X}$$

   This rule formalizes that if P knows Q's public key, and P receives a message X signed with Q's private key, P may infer that Q once sent X.[11]

2. the *nonce-verification* inference rule as given on page 238 of [BAN89b] (resp. page 6 of [BAN89a])[12]:

   NV $$\frac{P \models \sharp(X), \quad P \models Q \mid\!\sim X}{P \models Q \models X}$$

   This rule formalizes that if P believes X to be *fresh* (it originates in the current session), and P believes Q once conveyed X, then P may infer that Q believes X (in the current session).[13]

5.5 Proof of 'Unsoundness' of BAN logic

In this section, we will present our formal proof. In our proof, we use the term 'false belief'. This might be perceived as unnecessarily harsh or misleading, but

[9] These names of these inference rules have been given by the writer of this text.

[10] The GNY equivalent of this inference rule is **I4**.

[11] Inference rule **MM** has been questioned by Wedel and Kessler, as it is invalid if interpreted according to their semantics [WK96]. However, they point out that it is unclear whether BAN logic itself or their semantics of BAN logic is to blame for that.

[12] The GNY equivalent of this inference rule is **I6**.

[13] This rule relies on the assumption that only beliefs are communicated.

we will argue that this is the right formulation, even in lack of a clear semantics of BAN logic as a whole. The central construct of BAN logic, $\models$, is defined as follows on page 236 of [BAN89b] (resp. page 4 of [BAN89a]):

> "$P \models X$: *P believes* X, or P would be entitled to believe X. In particular, the principal P may act as though X is true. This construct is central to the logic."

In our proof, we obtain a result of the form $P \models X$, where X is *not warranted*. It *might* be the case that X were true, if some more communication were to occur than considered in our proof. Therefore, and in this way, we deem "false belief" the appropriate term for such an X. With this explanation given, let us formulate our main theorem:

Theorem 5.1 ('Unsoundness' of BAN logic)**.** *Within BAN logic (as defined in [BAN89b, BAN89a]) it is possible to derive unjustifiable beliefs. More precisely, a statement of the form $A \models X$ can be derived while the statement X itself cannot be derived.*

Proof (derivability). Consider the two parrots protocol, whose BAN idealization is given in Sect. 5.3. It is trivial to verify that A, C and B are capable of sending the messages they ought to send in the two parrots protocol.

As a result of protocol step 2 (S_2), the following statement is inserted:

$$A \triangleleft N \tag{5.1}$$

As a result of protocol step 4 (S_4), the following statement is inserted:

$$A \triangleleft \{H(N)\}_{K^{-1}} \tag{5.2}$$

Using inference rule **MM**, assumption $A \models \overset{K}{\mapsto} B$ and (5.2), we can infer:

$$A \models B \mid\sim H(N) \tag{5.3}$$

Using inference rule **H-BAN**, (5.3) and (5.1), we can infer:

$$A \models B \mid\sim N \tag{5.4}$$

Using inference rule **NV**, assumption $A \models \sharp(N)$ and (5.4), we can infer:

$$A \models B \models N \tag{5.5}$$

This inference is also depicted in Figure 5.5 as a *heavy annotation* of the protocol. Statement (5.5) should definitely not be derivable from the two parrots protocol. With all protocol assumptions satisfied and only valid inferences applied, an unjustifiable belief is established. More precisely, A believes $B \models N$, while it also consistent to assume that B does not know N, and nobody tells B about N. Therefore, $A \models B \models N$ is unjustified. □

1	$A \mid\equiv \overset{K}{\mapsto} B$			$(A \rightarrow B\colon H(N))$	
2	$A \mid\equiv N$		6	$B \lhd H(N)$	[3]
3	$A \mid\equiv \sharp(N)$			$(B \rightarrow A\colon \{H(N)\}_{K^{-1}})$	
	$(A \rightarrow C\colon N)$		7	$A \lhd \{H(N)\}_{K^{-1}}$	[4]
4	$C \lhd N$	[1]	8	$A \mid\equiv B \mid\sim H(N)$	**MM**(1, 7)
	$(C \rightarrow A\colon N)$		9	$A \mid\equiv B \mid\sim N$	**H-BAN**(8, 5)
5	$A \lhd N$	[2]	10	$A \mid\equiv B \mid\equiv N$	**NV**(3, 9)

FIGURE 5.5: Heavy BAN annotation of the two parrots protocol. This annotation actually shows that BAN logic is not 'sound', because statement 10 should not be derivable, as it is false.

The culprit is the inference rule **H-BAN**. This problem cannot be fixed by adding inference rules in such a way that $B \mid\equiv N$ can be inferred, as this would thwart the definition of a cryptographic hash function: then N would be derivable from $H(N)$. Such a 'fix' would increase the number of computationally unjustified inference rules from (at least) one to two.

Note that one more inference step is needed after application of the **H-BAN** rule before a false belief is established. This is because we need to obtain *belief of belief*, which cannot be directly inferred from **H-BAN**.[14]

5.6 The Semantic Approach

In the original BAN papers [BAN89b, BAN89a], a rather limited semantics is given for a part of the formal language of BAN logic. This semantics has been subject to an enormous amount of criticism. For one thing, the semantics is *very* closely tied to the formal language of BAN logic: what is derivable in the logic is by definition observable in the semantics. Arguably, the semantics is *so* closely tied to the formal language that it is of no additional value. Except for it being the subject of criticism, the semantics has hardly ever been used.

In Sect. 5.5 we have explained why we used the formulation 'false belief' in a proof that does not rely on any formal semantics. Therefore, we have consistently used quotes around the term *unsoundness*. In this section we will provide a proof based on a semantics: therefore, we may omit the quotes around unsoundness. However, for this proof we need to disregard all criticisms of the semantics of BAN logic. Therefore, we regard our proof in the previous section as more important. But it is of course up to the reader to choose what he likes best:

1. to agree with our use of 'unjustified belief' in the previous section, and with it agree with the semantics-free proof of 'unsoundness' (shown in the previous section), or

[14] Note that in BAN logic, the semantics of *belief* ($\mid\equiv$) is defined, while the semantics of *once said* ($\mid\sim$) is still "largely a mystery" (literal quote from [BAN89b, BAN89a, BAN88]).

2. to accept the semantics of BAN logic, regardless of all its shortcomings, and with it agree to our proof of unsoundness (shown in this section).

Before we show a run of the two parrots protocol in the semantics of BAN logic, it is appropriate to summarize this semantics:

- A *local state* of a principal P is a tuple $(\mathscr{M}_P, \mathscr{B}_P)$, where $\mathscr{M}_P$ is the set of messages seen ($\triangleleft$) by P, and $\mathscr{B}_P$ is the set of beliefs ($\mid\equiv$) of P. These sets enjoy closure properties which correspond to the inference rules of the logic. For compactness and ease of reading, we have only included elements in these sets which are relevant for our purposes.
- A *global state* s is a tuple containing the local states of all principals. If s is a global state, then s_P is the local state of P in s and $\mathscr{M}_P(s)$ and $\mathscr{B}_P(s)$ are the corresponding sets of seen messages and beliefs. In our case the principals are A, B and C, and a global state s is the triple (s_A, s_B, s_C).
- A *run* is a finite sequence of of global states $s_0, \ldots, s_n$.
- A *protocol run* of a protocol of n steps of the form $(P_i \rightarrow Q_i : X_i)$ is a run of length $n+1$, where s_0 corresponds to the protocol assumptions and where $X_i \ni \mathscr{M}_{Q_i}(s_i)$ for all i such that $0 < i \leq n$.

To be able to show a run of the two parrots protocol which is convenient to read, we will first name and give all local states. Then, we will give the full protocol run in which the names of these local states are used. For naming the local states, we adhere to the following convention: $s_P^{n,\cdots,n'}$ is the local state of principal P in the global states $n, \cdots, n'$.

The local states of principals A, B and C are as follows:

$$\begin{array}{lllll}
 & & \mathscr{M}_A & \mathscr{B}_A & \\
s_A^{0,1} & = (& \emptyset, & \{\overset{K}{\mapsto} B, N, \sharp(N)\} &) \\
s_A^{2,3} & = (& \{N\}, & \{\overset{K}{\mapsto} B, N, \sharp(N)\} &) \\
s_A^{4} & = (& \{N, \{H(N)\}_{K^{-1}}\}, & \{\overset{K}{\mapsto} B, N, \sharp(N), & \\
 & & & B \mid\!\sim H(N), B \mid\!\sim N, B \mid\equiv N\} &) \\
 & & & & \\
 & & \mathscr{M}_B & \mathscr{B}_B & \\
s_B^{0,1,2} & = (& \emptyset, & \emptyset &) \\
s_B^{3,4} & = (& \{H(N)\} & \{H(N), \{H(N)\}_{K^{-1}}\} &) \\
 & & & & \\
 & & \mathscr{M}_C & \mathscr{B}_C & \\
s_C^{0} & = (& \emptyset, & \emptyset &) \\
s_C^{1,2,3,4} & = (& \{N\} & \{N\} &)
\end{array} \tag{5.6}$$

The following is a *run* of the two parrots protocol:

$$s_0, s_1, s_2, s_3, s_4 \tag{5.7}$$

where s_i are the global states after the consecutive steps of the protocol:

$$\begin{array}{rcccl}
 & s_A & s_B & s_C & \\
s_0 = (& s_A^{0,1} & s_B^{0,1,2} & s_C^{0} &) \\
s_1 = (& s_A^{0,1} & s_B^{0,1,2} & s_C^{1,2,3,4} &) \\
s_2 = (& s_A^{2,3} & s_B^{0,1,2} & s_C^{1,2,3,4} &) \\
s_3 = (& s_A^{2,3} & s_B^{3,4} & s_C^{1,2,3,4} &) \\
s_4 = (& s_A^{4} & s_B^{3,4} & s_C^{1,2,3,4} &)
\end{array} \tag{5.8}$$

Now that we have specified a protocol run of the two parrots protocol, we can give our alternative proof of unsoundness:

Proof (observability). As shown in statement (5.5) of the *derivability* proof in Section 5.5, we can derive in BAN logic the sentence $A \mid\equiv B \mid\equiv N$ in *every* run S_1, S_2, S_3, S_4 of the two parrots protocol. Thus, we have:

$$S_1, S_2, S_3, S_4 \vdash A \mid\equiv B \mid\equiv N \tag{5.9}$$

Global state s_4 corresponds to the semantics after *a particular* protocol run S_1, S_2, S_3, S_4 of the two parrots protocol. When we take the model as given in equations (5.6)–(5.8), we can observe that 'A believes B knows N': $B \mid\equiv N \in \mathscr{B}_A(s_4)$, which gives us:

$$s_4 \models A \mid\equiv B \mid\equiv N \tag{5.10}$$

On the other hand, we can also observe in our model that 'B does not know N': $N \notin \mathscr{B}_B(s_4)$, which gives us:

$$s_4 \not\models B \mid\equiv N \tag{5.11}$$

Thus, the belief of A as given in (5.10) is not true in *a particular* protocol run as shown in (5.11). The false belief of A as given in (5.10), is nevertheless derivable (5.9) in *every* protocol run. Thus, it is possible to derive a false belief within BAN logic. □

Let us quote one last excerpt from Section 13, on page 269 of [BAN89b] (resp. pages 47–48 of [BAN89a]):

> "Clearly, some beliefs are false. This seems essential to a satisfactory semantics. [...] Most beliefs happen to be true in practice, but the semantics does not account for this coincidence. To guarantee that all beliefs are true we would need to guarantee that all initial beliefs are true."

The existence of false beliefs in the semantics as such is not a problem, the problem is that some false beliefs are derivable from true ones.

5.7 Conclusion

The formal approach to protocol analysis essentially started with BAN logic. Many critiques of BAN logic have appeared, mentioning its incompleteness (i.e., inability to detect some obvious problems, cf. [Nes90]) and its poor semantics (among many others, see [AT91]). Nevertheless, these critiques have not been a reason to abandon the *way of thinking* introduced by BAN logic [HPvdM03]. The many augmentations to BAN logic (most notably, AT [AT91], GNY [GNY90], AUTLOG [KW94, WK96], VO [vO93], SVO [SvO94, SvO96] and SVD [Dek00]) show the trust in the formal approach which originates from BAN logic. In our opinion, this consensual trust in the *way of thinking* introduced by BAN logic is justified. While obtaining completeness has long been regarded as impossible, the soundness of BAN logic itself has never been seriously doubted. Wedel and Kessler identified rules in BAN, AT and GNY which are invalid in their semantics, but they point out that it is unclear whether the inference rules or their semantics are to blame for that [WK96]. Various more recent results [AR02, CD05a, CD05b, Syv00] provide directions on how completeness could be obtained for formal protocol analysis.

Our unsoundness result does not at all invalidate the formal approach to protocol analysis. It should merely count as a warning to those who wish to *complete* their logic. All augmentations of BAN logic are incomplete in the sense that they do not accommodate all cryptographic primitives known to date. These logics are essentially 'just big enough' to capture the problems the authors intend to capture. And to be fair, this has been difficult enough already. Just a few BAN-descendant logics accommodate cryptographic hash functions, none of them accommodate fancy primitives like (to name just an example) oblivious transfer.

The fact that none of the hash-accommodating BAN-descendant logics adopts the **H-BAN** inference rule, can probably be explained by the observation that constructing a good logic is already so difficult that none of the authors will have felt the urge to include an inference rule into their logic that was not needed to capture the problem the author intended to capture. Nevertheless, it is remarkable that we are apparently the first to find this result on a paper which has been so extensively studied and is 17 years old.

So far, we know of only one publication which relies on the faulty **H-BAN** inference rule [AvdHdV01]. In this publication, the SET protocol[15] is analyzed in BAN logic. It remains open whether the authors' assessment of SET holds in a BAN logic with the inference rule **H-BAN** omitted.

[15] SET stands for *Secure Electronic Transactions* [MV97a, MV97b, MV97c]. The protocol was introduced by VISA and Mastercard for online payments, but it has never been widely adopted or deployed.