# Towards Extensional Goals in Authentication Protocols

Colin Boyd
Information Security Research Centre
School of Data Communications
Queensland University of Technology
Brisbane Q4001
AUSTRALIA

**Abstract**

The importance of clarifying the goals of a cryptographic protocol is widely recognised. The majority of authors have addressed intensional goals which are concerned with correct operation within the protocol itself. Extensional goals are properties independent of the protocol and define what the protocol is designed to achieve. This paper reviews the previous literature on goals in protocols and classifies them as intensional or extensional goals. A hierarchy of extensional protocol goals is proposed which includes the major proposed goals for key establishment. It is shown how these extensional goals can be exploited to motivate design of entity authentication protocols.

## 1 Introduction

Research into cryptographic protocols has been increasing in momentum in recent years. A great deal of effort has been devoted to analysis techniques [15] while general principles for design of protocols are now much better understood [2]. Even so, the rate at which protocol faults are being reported does not appear to be slowing down [1, 14, 17, 16].

It is obvious that any attack on a protocol is only valid if it violates some property that the protocol was intended to achieve. In other words all attacks must be considered relative to the protocol goals. Many protocols are poorly designed because their authors are unclear what are the protocol goals they are trying to achieve. This is turn leads to disputes about whether protocol attacks are valid, since designers may regard the goals differently from analysers (as discussed by Gollman [13]).

Clarity in describing protocols goals is desirable for all parties concerned.

**Designers** should make use of the protocol goals to justify each message field and all cryptographic processing. Experience shows that protocols with well-defined goals are stream-lined and transparent to analyse.

**Analysers** should make use of protocol goals to direct their attempts to find attacks.

**Implementors** must be clear on exactly what a protocol is intended to achieve so that they protect the user from the correct threats. It is not the responsibility of the implementor to work out what a protocol achieves.

Many authors have considered the question of what are appropriate goals, mainly in the context of protocol analysis. A panel session at the 1996 Computer Security Foundations workshop was dedicated to the question: "What is an Attack on a Cryptographic Protocol?" [30], and showed that there are many questions yet to be resolved. A central issue in this paper is the division between *intensional* goals which are generally concerned with ensuring that the protocol runs correctly as specified, and *extensional* goals which are concerned with what the protocol achieves for its participants.

The view taken in this paper is that while intensional properties may be important for analysis, it is extensional properties that are appropriate to be considered by the protocol designer. In other words the protocol designer should be ensuring that the extensional protocol goals are achieved, by whatever means. Furthermore, an attack on a protocol must be measured against whether it defeats the extensional goals of the protocol. In addition it is shown that a clear view of extensional goals is of great benefit to the protocol designer.

In the next section the previous literature on goals in protocols is reviewed and goals are classified as intensional or extensional goals. Then a hierarchy of extensional protocol goals is proposed which includes the major proposed goals for key establishment. Examples are then considered to contrast the different features captured by the intensional and extensional views. This leads to consideration of a new attack on the widely studied Needham-Schroeder public key protocol. Finally it is shown how these extensional goals can be exploited to motivate design of key establishment protocols.

## 2 Goals Discussed in Previous Work

### 2.1 Intensional and Extensional Goals

Roscoe [25] has recently considered the difference between intensional and extensional specifications of cryptographic protocols. Although no formal definition of these terms is given, the general principle is whether specifications take into account the details of how the operates rather than what each protocol principal gains from the protocol. An *extensional* property is defined by Roscoe as one which is:

> ...independent of the details of the protocol and would apply to any other protocol designed to achieve the same effect.

Thus extensional properties cannot refer to any specific protocol messages. An example of an extensional property would be that user $A$ wishes to communicate with a user $B$ using a shared key.

In contrast, Roscoe defines an *intensional* property as one:

> ...whose primary purpose is to assert a property of the way, in terms of communications within the protocol, a particular state is reached.

Thus an example of an intensional property would be that user $A$ has responded to a specific message from user $B$ using a shared key. This intensional property could be used to provide the extensional property mentioned above. However there are other methods that can be used to achieve the extensional property; for example, a trusted server might inform $B$ that user $A$ has the key and wishes to use it to communicate with $B$.

Roscoe showed with examples that analysis of a protocol from an intensional specification is often more useful in finding protocol flaws than one from an extensional specification. He defines the *canonical intensional specification* as follows.

**Canonical intensional specification** is that an entity will believe a protocol run has completed only if a correct series of messages has occured up to and including the last message that entity communicates.

The canonical specification is a very strong requirement that fails for many protocols which are believed secure by different measures. It essentially says that if the protocol completes in the view of its participants then the protocol must have run as specified. Notice that this specification says nothing about what the protocol achieves, and in particular is satisfied by the (admittedly very secure) null protocol.

## 2.2 Key Establishment or Authentication?

In the early literature on cryptographic protocols it was common to refer to all protocols concerned with setting up session keys as 'authentication protocols'. This is not entirely satisfactory because some protocols which set up session keys provide no authentication of one party to the other while other protocols designed to provide entity authentication involve no session key. Therefore it has become common more recently to distinguish between protocols which provide only authentication, and call these *entity authentication protocols* while using the term *key establishment protocol* for one that involves setting up a new key, typically for a communications session.

One of the features of the hierarchy of goals presented below is an integration of goals concerning key establishment and entity authentication. It has been recognised by many authors that there is a problem defining in abstract terms what should be meant by entity authentication, although the meaning of key establishment seems easier to decide upon. Gollman [13] has put forward a number of different options for what could be meant by authentication. The first one is as follows.

**Gol1** The protocol shall establish a fresh session key, known only to the participants in the session and possibly some Trusted Third Parties.

This is clearly an extensional goal. Furthermore, it may be achieved even though each party knows nothing about even the existence of the other party, let alone whether the other party is willing to engage in a session. Thus this is a goal about key establishment rather than entity authentication.

The second goal suggested by Gollman is as follows, in which $A$ and $B$ are the protocol principals.

**Gol2** A cryptographic key associated with $B$ was used in a message received by $A$ during the protocol run. The protocol run is defined by $A$'s challenge or a current time stamp.

This is an intensional goal concerning entity authentication. It says nothing about a new session key and can clearly be satisfied by a protocol which is not concerned with key establishment. Gollman's other two goals are also intensional and say nothing about session keys.

This pattern, of using extensional specifications when considering key establishment and intensional ones when considering entity authentication, will be seen to be repeated many times by different authors. It may be posited that it is easier to use extensional goals when dealing with key establishment than when dealing with entity authentication. In section 3 extensional goals for entity authentication are proposed.

## 2.3 Goals in Logical Analysis

Various logics have been used successfully for analysis of cryptographic protocols. The first of these was the logic of Burrows, Abadi and Needham, the BAN logic [10]. An analysis of a protocol using the BAN logic results in a set of *beliefs* of each protocol principal and goals may be expressed as the desired final beliefs. In principle it is not necessary to know the protocol goals in order to perform such an analysis; indeed one of the strengths of a logical analysis is the ability to discover subtly differing final beliefs of the principals. However, the authors of the BAN logic do suggest what may be typical protocol goals. The first is as follows, where $A$ and $B$ are principals who wish to use a new key $K$. (The goal may be expressed in the formalism of the logic but that need not concern us here.)

**BAN1** $A$ believes ($K$ is a good key for $A$ and $B$)

This is an extensional goal which is essentially identical to Gollman's **Gol1**. (Although being a 'good' key is an atomic construct, its semantics are that the key will never be discovered by others. Furthermore only keys that are fresh are ever promoted to being believed good.) Normally $B$ would also be expected to establish a symmetrical belief. Principals in a BAN logic analysis can possess beliefs about beliefs and the other typical goal put forward is such a *second order* belief.

**BAN2** $A$ believes ($B$ believes ($K$ is a good key for $A$ and $B$))

Again this is an extensional goal concerned solely with key establishment. No general goals about entity authentication are discussed, but in the course of analysis of several protocols the BAN authors obtain certain properties that might have been termed protocol goals. For example they show that some protocols reveal that a certain principal is 'alive' because that principal has sent a message recently. We will expand on this idea later.

Numerous enhancements and alternatives to the BAN logic have been published. The logic of Syverson and van Oorschot, SVO logic [28, 29], aims to unify a number of previous logics including BAN. The authors identify what they term six 'Generic Formal Goals'. These are expressed in English below; for formal statements readers should refer to the papers.

**SVO1: Far-end Operative** $A$ believes $B$ recently 'said' something.

**SVO2: Entity Authentication** $A$ believes $B$ recently replied to a specific challenge.

**SVO3: Secure Key Establishment** $A$ has a certain key $K$ which $A$ believes is good for communication with $B$.

**SVO4: Key Confirmation** In addition to **SVO3**, $A$ has received evidence confirming that $B$ knows $K$.

**SVO5: Key Freshness** $A$ believes a certain key $K$ is fresh.

**SVO6: Mutual Understanding of Shared Key** $A$ believes that $B$ has recently confirmed that $B$ has a certain key $K$ which $B$ believes is good for communication with $A$.

There are clearly dependencies between various of these goals. Furthermore, it is not clear why these particular goals are important; for example it might be questioned whether Secure Key Establishment is useful without Key Freshness. To be fair to the authors they state that it is *not* intended as a "definitive list of *the* goals that a key agreement or key distribution protocol should meet".

Secure Key Establishment, **SVO3**, is an extensional goal essentially the same as **BAN1** and **Gol1**. **SVO4** and **SVO6** are also extensional. **SVO1** and **SVO2**, however, are intensional goals because they are concerned with particular message flows. Below the goal **SVO1** ('liveness') will be presented as an extensional goal in the absence of the particular requirement that a message has recently been uttered by $B$.

### 2.4 Goals in Algebraic Analysis

The most widely used alternative to using logic for analysis of cryptographic protocols is to specify them in an 'algebraic' specification language and perform analysis of the states reached, particularly by an attacker. Several research efforts have been made in this direction [15, 24, 27].

The NRL Protocol Analyzer [19] is a software tool implementing one such approach. Syverson and Meadows [31] have considered methods to specify formal requirements for the protocols which are analysed using the NRL Protocol Analyzer, rather than simply looking for specific flaws such as compromised keys. They specify slightly different goals for different protocol architectures, in particular differentiating between server based key distribution and key agreement protocols. The three requirements for the two party/one server key distribution protocols, in their informal versions, are as follows.

**SM1** If a key is accepted, it should not be learned by the intruder, except through a compromise event[1].

**SM2** If a key is accepted for communication between two parties, it should not have been accepted in the past, except by the other party.

**SM3** If a key is accepted for communication between two entities, then it must have been requested by the initiating entity and sent by the server for a communication between those two entitites.

**SM1** and **SM2** are extensional and correspond closely to **SVO3** and **SVO5**. **SM3**, however, is intensional and places restrictions on the protocol format; for example it precludes the possibility that one of the two users generates the session key, or that the responder alone contacts the server.

Syverson and Meadows also provide requirements for the case of key agreement between two users without the help of a server. These requirements are also partly extensional and partly intensional. In an extension of their work they also consider requirements for re-authentication of a key [31]; as might be expected, these involve different properties from those relevant for new session keys.

Another method for algebraic analysis uses CSP specifications together with a tool called FDR [24]. Lowe has used this technique to derive a variety of new protocol attacks and recently has considered what are the possible goals for authentication protocols [18]. These all concern properties provided to an initiator $A$ communicating with a responder $B$.

**Low1: Aliveness** Whenever $A$ completes a run of the protocol, apparently with $B$, then $B$ has previously been running the protocol.

**Low2: Weak Agreement** Whenever $A$ completes a run of the protocol, apparently with responder $B$, then $B$ has previously been running the protocol, apparently with $A$.

**Low3: Non-injective Agreement** on a set of data items $ds$. Whenever $A$ completes a run of the protocol, apparently with $B$, then $B$ has previously been running the protocol, apparently with $A$, and $B$ was acting as responder in his run, and the two agents agree on the data values corresponding to all the variables in $ds$.

**Low4: Agreement** on a set of data items $ds$. Whenever $A$ completes a run of the protocol, apparently with responder $B$, then $B$ has previously been running the protocol, apparently

[1] This is a formally defined event in the model, representing compromise of a session key.

with $A$, and $B$ was acting as responder in his run, and the two agents agreed on the data values corresponding to all the variables in $ds$ and each such run of $A$ corresponds to a *unique* run of $B$.

It can be seen that each property is stricter than the previous one. Lowe also provides formal versions in CSP for which he shows that the hierarchy holds formally. All the properties are intensional. Take for example **Low1**; this is different from **SVO1** because it demands that the responder is engaged in the same protocol as the initiator, rather than that a value is returned as in **SVO1**. **Low1** does not provide assurance that $B$ has responded recently, so it provides liveness at some time, rather than the liveness 'now' of **SVO1**. However Lowe does discuss how to extend the properties to include recentness.

Another approach using CSP has been made by Schneider [27]. He enumerated nine different 'flavours' of authentication in his analysis of the well-known Needham-Schroeder public key protocol [21]. These fascinatingly subtle variations reveal the microscopic detail with which protocol goals may be differentiated. Each property relates matters such as who initiated the run, whether a nonce is associated with a specific party, or whether a nonce was received by a specific party. All the properties are intensional.

There have been a number of other algebraic analysis techniques proposed. It has recently been recognised that, in order to ensure that the same protocol specified by a designer is presented to different analysis tools in a uniform way, there should be some standard method for presenting protocol specifications that includes all the information required by each tool. To this end Millen has initiated a Common Authentication Protocols Specification Language (CAPSL) [20] which does just this. A CAPSL specification consists of a number of sections concerning different protocol elements. One of these is a section on protocol assumptions and goals.

A CAPSL protocol specifier is free to choose the protocol goals within what may be expressed in the language. This requires use of a set of keywords, principally `BELIEVES`, `HOLDS`, `KNOWS` and `SECRET`. For examples, a goal might be that the session key $K$ is secret to certain principals, or that a principal believes that another principal holds $K$. These are extensional goals. At present CAPSL only appears to have the ability to specify goals concerning key establishment and not entity authentication.

### 2.5 Goals for Provable Protocols

Protocol analysis techniques are not (currently) able to provide a proof of the security, or otherwise, of an arbitrary protocol. However, Bellare and Rogaway, and others following them, have been able to design particular protocols which are proven secure in a specific sense [3, 5, 7]. Security in these models is based on the notion of *matching conversations*, an idea which seems to have been first introduced by Bird *et al* [6]. Roughly, a protocol is secure if a principal will successfully complete a protocol run only when the protocol partner has a 'matching' conversation. The idea was also used by Diffie, Van Oorschot and Wiener [12] in their definition of a secure protocol. They give the following definition.

**DVW** A *secure protocol* is a protocol for which the following conditions hold in all cases where one party, say Alice, executes the protocol faithfully and accepts the identity of the other:

- At the time that Alice accepts the other party's identity (before she sends or receives a subsequent message) the other party's record of the partial or full run matches Alice's record.

- It is computationally infeasible for the exchanged key if accepted by Alice to be recovered by anyone other than Alice and possibly the party whose identity Alice accepted. (This condition does not apply to authentication without key exchange.)

As well as this definition a detailed definition of what it means for protocols runs to match is also given. The first condition is intensional and concerns entity authentication. The second is extensional and concerns key establishment.

Bellare and Rogaway [3] consider a powerful (formally defined) adversary. They define a secure mutual authentication protocol as one in which two principals 'accept' if they have matching conversations, but the probability of an unmatched conversation (in the presence of the powerful adversary) is negligible. The protocol that the authors present satisfies **SVO1** and **SVO2** but it is not clear that this is a minimal example that satisfies their definition. (The null protocol does satisfy the definition but should perhaps be excluded on the grounds that there is no possibility for any principal to accept.)

# 3 A Hierarchy of Extensional Goals

It is clear from section 2 that there are many similarities between the different protocol goals specified in the literature. While many authors have given extensional goals for key establishment only intensional goals seem to have been proposed for entity authentication. If protocol designers are to provide secure entity authentication protocols they might be helped greatly by knowing extensional specifications for what they are trying to achieve.

The aim in this section is to consider what are the reasonable and desirable extensional goals for cryptographic protocols. A hierarchy is proposed which incorporates key establishment on one side and entity authentication on the other, as well as extended goals which include both key establishment and entity authentication. These are certainly not the only possible extensional goals and may not be the most useful, but they do seem to fit together in a logical way and cover a broad range of reasonable protocol goals. The approach taken is to consider the fundamental elements used in practical protocols and to abstract the properties that are obtained by employing those elements.

## 3.1 Key Oriented Goals

As has been seen in the last section, there is broad agreement in the research community about the extensional goals in key establishment protocols. These goals may be reached by considering what may be achieved with typical message components. There are only three types of message components that are conventionally used in cryptographic protocols for key establishment and entity authentication. These are:

1. **secrets** which include long-term keys and session keys.
2. **identifiers** for protocol principals.
3. **nonces** which may be random values, timestamps or counters.

These components are combined and processed with cryptographic mechanisms to provide confidentiality and/or authentication.

Consider key establishment. A new session key $K$ may be associated with a nonce, or with identifiers of protocol principals. In practice a session key is not of any use unless it is known to be fresh and it is known which other entities possess it. Comparison with the definitions in section 2 shows that most authors agree that secure key establishment should require the two extensional goals that the key is known to be fresh and is known only to the other protocol participant(s), possibly including trusted third parties. This is often referred to as establishing a *good key*.

**Good Key** for use with $B$. $A$ accepts the key for use with $B$ only if:

- the key is fresh (**key freshness**).
- the key is known only to $A$ and $B$ (**key exclusivity**).

### 3.2 User Oriented Goals

When it comes to entity authentication authors seem to have had a harder time deciding what should be an extensional goal, and in section 2 it was seen that most resort to intensional specifications. One reason for this may be that it is difficult to be clear on the purpose of entity authentication in the absence of key establishment. In fact Bellare and Rogaway [5] have stated:

> ...entity authentication is rarely useful in the absence of an associated key distribution, while key distribution, all by itself, it not only useful, but it is not appreciably more so when an entity authentication occurs along side ...by the time you become aware of [an entity authentication] there will be no particular reason to believe that the partner is still "out there" anyway.

There are situations when entity authentication by itself may be useful, such as when using a secured communication channel. But it is important to appreciate exactly what it provides. Imagine user $A$ having received some messages in an entity authentication protocol. What is it that she can hope to have learned from those messages? One aspect is that user $B$ is really out there now, somewhere on the network. This is the liveness property we have already seen. The only other assurance that seems relevant is to know that $B$ is ready to engage in communication with $A$.

Considering again the fundamental elements used in authentication protocols this seems to be all that can be achieved. A session key is no longer relevant; therefore messages can convey freshness, or principals with which communication is desired. Combining these leads to a proposed extensional definition of entity authentication. (There are several alternative ways of expressing this property which all indicate that $A$ is authenticated to $B$ only if $A$ is prepared to engage in communications with $B$.)

**Entity Authentication** of $A$ to $B$. $B$ accepts $A$ only if principal $A$ wishes to communicate with $B$.

The two subgoals of Entity Authentication are that $A$ once wished to communicate with $B$, and that $A$ wishes to communicate with an unknown principal. The latter of these is the liveness property discussed before, similar to goal **SVO1**. Notice that it is straightforward to extend this definition to a multi-party goal of entity authentication of $A$ to a group of users $\mathcal{U}$: the principal $A$ wishes to communicate with the principals in $\mathcal{U}$.

### 3.3 Enhanced Goals

It is possible to consider any combination of the goals or subgoals from key establishment and from entity authentication. There are also goals which go beyond both good key and entity authentication which are termed here *enhanced goals*. Which are the useful goals to aim for? To answer this question we must examine what is the purpose to go beyond key establishment. A protocol that provides only key establishment gives no assurance that the partner with whom communication is desired even exists. Thus key establishment only provides the *ability* to engage in secure communication. Enhanced goals seek to establish the *readiness* of the partner to engage in secure communication. Since the extensional goal for entity authentication proposed above deals with exactly this concern it is natural that enhanced goals should include entity authentication together with key establishment.

Key confirmation provides evidence that the partner has the same key but leaves open the possibility that the key is intended by the partner for a different communication session (with the assumption that the partner may be engaged in several conversations). Key confirmation provides evidence that the partner wishes to communicate with some entity, so implies liveness but may not include entity authentication.

**Key confirmation** $B$ accepts $A$ with key $K$ only if $K$ is a good key to communicate with $A$ and principal $B$ has received $K$.

Mutual belief in the key, following **SVO6**, adds to key confirmation that $B$ associates key $K$ with $A$. (Actually, **SVO6** does not require the good key property, but seems of little value if it does not also hold.) It provides both key confirmation and entity authentication since if the partner has acknowledged that the key is good for the communication this can be taken as a confirmation that the partner is willing to communicate.

**Mutual Belief in Key** $B$ accepts $A$ with key $K$ only if $K$ is a good key for use with $A$, and $B$ wishes to communicate with $A$ using key $K$ which $B$ believes is good for that purpose.

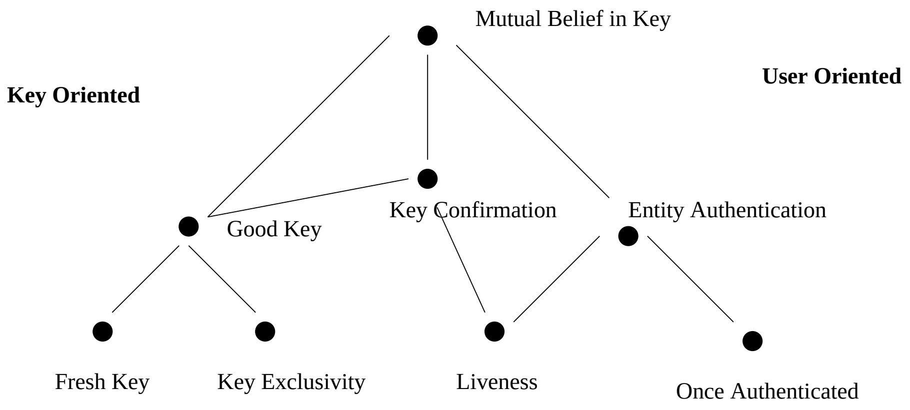


Figure 1: Hierarchy of Extensional Goals

The hierarchy of goals is shown in figure 1 as a lattice. Entity authentication and its two subgoals are classed as user oriented goals, while good key and its subgoals are key oriented. Mutual belief and key confirmation are classed as extended goals which concern both keys and users. Of course this hierarchy does not show all possible extensional goals. The ones shown appear to be some of the most important ones considered in the literature. As an example of a goal which is not included, there is an enhanced goal that lies between key confirmation and mutual belief, which provides key confirmation and entity authentication but does not provide assurance that the key is known by the partner to be good.

# 4 Examples

Two example protocols are now considered in the light of the goals examined above. These have been chosen particularly because they have given rise to controversy in the literature about what they achieve and what attacks are valid.

## 4.1 STS Protocol

The station-to-station (STS) protocol [12] uses a signature in the exchanged messages to add authentication to the well-known Diffie-Hellman protocol [11]. This uses arithmetic in the multiplicative group of a finite field of prime order with generator $\alpha$. Exponents $x$ and $y$ are chosen randomly by $A$ and $B$ respectively and are used to form the session key $K = \alpha^{xy}$. The messages in a successful protocol run are as follows.

1. $A \longrightarrow B : A, B, \alpha^x$
2. $B \longrightarrow A : B, A, \alpha^y, \{S_B(\alpha^y, \alpha^x)\}_{K_{AB}}$
3. $A \longrightarrow B : A, B, \{S_A(\alpha^x, \alpha^y)\}_{K_{AB}}$

Here $S_X(.)$ represents the signature by the principal $X$ on the string in the brackets, while $\{M\}_K$ denotes encryption of message $M$ using key $K$. The particular signature algorithm chosen does not matter for the protocol. Consider how the good key goal is achieved for $A$.

1. The signature in message 2 can only be formed by $B$.
2. It is not a replay from an old protocol run since $A$ knows that $\alpha^x$ was fresh.
3. The signature alone does not imply that $B$ knows $K_{AB}$. Therefore the encryption with $K_{AB}$ is necessary to provide assurance that $B$ really knows $K_{AB}$.

Thus it appears that $A$ gains key confirmation, as well as good key with $B$, from message 2. With regard to user oriented goals, it seems clear that both users achieve liveness of the other, since each receives a signed message containing a value it knows to be fresh. Entity authentication is more problematic since there is no explicit inclusion of identifiers in the signed messages which could be used to deduce the desired communications partner. Recently Lowe [17] has proposed an attack on the STS protocol. The attack does not affect the key establishment properties but is addressed at whether entity authentication is achieved.

Suppose $I$ is an intruder who wishes to attack the protocol.

- $I$ intercepts a protocol run started by $A$ and masquerades as $B$.
- In parallel $I$ starts a protocol run with $B$ while masquerading as $A$.

The attack runs as follows, where $I_X$ denotes $I$ masquerading as principal $X$.

1. $A \longrightarrow I_B : A, B, \alpha^x$

1'. $I \longrightarrow B : I, B, \alpha^x$

2'. $B \longrightarrow I : B, I, \alpha^y, \{S_B(\alpha^y, \alpha^x)\}_{K_{AB}}$

2. $I_B \longrightarrow A : B, A, \alpha^y, \{S_B(\alpha^y, \alpha^x)\}_{K_{AB}}$

3. $A \longrightarrow I_B : A, B, \{S_A(\alpha^x, \alpha^y)\}_{K_{AB}}$

The attack is very simple; $I$ is doing little more than relaying each message that passes between $A$ and $B$. What is the result? $B$ has no indication that $A$ has engaged in the protocol and yet $A$ has completed a successful run, apparently with $B$.

Is this a successful attack on the STS protocol? The answer must be that is depends what it was believed that STS achieves.

- After the attacking run it is clear that the good key goal has not been broken.
- Key confirmation has indeed been achieved: $A$ can be sure that $B$ knows the shared key.
- $A$ does not know that $B$ knows the key is good for use with $A$. In other words the *mutual belief in key* goal does not apply.
- The attack shows that $A$ would be wrong to conclude, after a successful run, that $B$ wishes to communicate with her. Thus entity authentication (using the proposed extensional definition) is not achieved.

Thus the attack is valid if mutual belief in the key was a protocol goal. It may also be valid if entity authentication was a goal. However, it is interesting to note that Syverson and Van Oorschot prove in their logic [28] that the protocol satisfies their goal **SVO2**, which they term entity authentication. Lowe proposes [17] that the identity of the other party be included in the signatures in order to overcome the attack. This also allows an informal argument that the extensional definition of entity authentication is achieved, if the included identifier is interpreted as the name of the entity with which communication is desired.

## 4.2 Needham-Schroeder Public Key Protocol

The Needham-Schroeder public key protocol [21] is another example that has been widely examined by protocol researchers. Users $A$ and $B$ own public keys, $K_A$ and $K_B$ respectively, which, it is assumed here, are authentically known by the other. In the following, $N_A$ and $N_B$ are random values chosen by $A$ and $B$ respectively, while $\{X\}_K$ denotes encryption of the value $X$ with the public key $K$.

1. $A \longrightarrow B : \{N_A, A\}_{K_B}$
2. $B \longrightarrow A : \{N_A, N_B\}_{K_A}$
3. $A \longrightarrow B : \{N_B\}_{K_B}$

Although this protocol is nearly 20 years old it has aroused quite some interest recently. An attack of Lowe [16] shows that $B$ cannot be sure that the final message came from $A$. Gollman [13] points out that the protocol fails, because of this, to achieve his goal **Gol2**. Notice that $A$ has never explicitly declared her intention to converse with $B$.

In order to fix the protocol against his attack, Lowe proposed the following variation which simply includes the identifier of $B$ in the second message.

1. $A \longrightarrow B : \{N_A, A\}_{K_B}$

2. $B \longrightarrow A : \{N_A, N_B, B\}_{K_A}$

3. $A \longrightarrow B : \{N_B\}_{K_B}$

Let us consider whether the extensional goals for entity authentication proposed above are satisfied. Consider the point of view of $A$. When she receives the message 2 she wants to use it to verify that $B$ wishes to communicate with her. Immediately we run into a problem here because it is not clear what she can tell from receiving an encrypted message. What she really requires is an authenticated message, but encryption with her public key may not provide this. Indeed, encryption is a way to provide the confidentiality service to the message *sender*. It seems that the protocol designers (and most analysers) have assumed that inclusion of the nonce $N_A$, which $A$ sent confidentially to $B$, is sufficient to provide authentication. Unfortunately this need not be the case, even with the most well known public key encryption algorithms.

To see the point, consider the Blum-Goldwasser public key encryption algorithm [8]. In this algorithm the plaintext is added modulo 2 to a string formed by iterative squaring. Consequently it is trivial for an attacker to change the known parts of the ciphertext, which are the identifiers in message 1 (and message 2 in Lowe's fix). This results in a simple attack on the protocol, or on Lowe's fix.

In practice, use of the Blum-Goldwasser algorithm is not very reasonable, since it is known to be vulnerable to a chosen ciphertext attack which is eminently possible in the protocol. On the other hand, there is a reasonable scenario in which use of the well-known RSA algorithm [23] is insecure. Suppose the attacker knows (or chooses) an identifier $C$ such that $C$ is the same bit string as identifier $A$ shifted left one place. Then the attacker can capture $\{N_A, A\}_{K_B}$ and multiply it by $\{2\}_{K_B}$. The effect of this is to change the encrypted message by shifting it to the left, due to the well-known multiplicative property of RSA, so that it becomes $\{2*(N_A, A)\}_{K_B} = \{2*N_A, C\}_{K_B}$. By this process, $A$'s name changes into $C$, even though the attacker has no knowledge of $N_A$. With the collaboration of $C$ (or we may assume the attacker is $C$) this allows a run of the protocol, or Lowe's fix, where $B$ only ever wanted to communicate with $C$, but $A$ believes $B$ wants to communicate with $A$. An attacking run on Lowe's fix is as follows.

1. $A \longrightarrow C_B : \{N_A, A\}_{K_B}$

1'. $C \longrightarrow B : \{2 * N_A, C\}_{K_B}$

2. $B \longrightarrow C : \{2 * N_A, N_B, B\}_{K_C}$

2'. $C_B \longrightarrow A : \{N_A, N_B, B\}_{K_A}$

3. $A \longrightarrow B : \{N_B\}_{K_B}$

In order to allow the receivers of messages 2 and 3 to authenticate the messages the encryption functions needs to act like a message authentication code (MAC) which guarantees that the message was written with knowledge of a shared secret ($N_A$ or $N_B$ in this case). This leads to the following alternative protocol.

1. $A \longrightarrow B : \{N_A\}_{K_B}, A$

2. $B \longrightarrow A : \{N_B\}_{K_A}, h(N_A, B)$

3. $A \longrightarrow B : h(N_B, A, B)$

Here $h(K, .)$ may be a keyed one-way hash function such that need $K$ is needed to calculate it, and it does not give away $K$. Several constructions for such functions exist in the literature [22]. Further considerations on protocol design for entity authentication are discussed below.

It should be noted that the above attack is probably prevented by including strict formatting in the RSA encrypted messages, such as is recommended by the RSA Encryption Standard PKCS #1 [26] or by Bellare and Rogaway [5]. Such formatting is intended to ensure that the encrypting agent must be aware of the whole plaintext in order to form a valid ciphertext. In other words encryption of a shared secret (such as the nonce $N_A$ or $N_B$ in the Needham-Schroeder protocol) gives the ciphertext the essential property of a MAC. Notice also that the attack is *not* a 'typing' attack, since inclusion of, say, one bit typing tags would still allow the attack to succeed with high probability.

## 5 Designing Entity Authentication Protocols

It was observed in section 2 that authors have generally found it easy to give extensional goals for key establishment, but that for entity authentication only intensional goals are usually found. It may be more than a coincidence that the majority of recent attacks on protocols seem to have been concerned with authentication rather than key establishment [1, 14, 17]. A clear view of what it means to achieve key establishment has allowed protocol designers to more systematically incorporate the correct mechanisms.

An informal, but successful, method to design new key establishment protocols has been to use the extensional properties of key freshness and exclusivity in combination with abstract notions of secure channels [9]. The purpose of this section is to suggest that a similar process can be done for entity authentication using the extensional properties established in section 3. The two properties that are of interest are liveness and entity authentication.

An abstract version of protocols intended to achieve liveness is as follows, where $A \xrightarrow{a} B$ denotes an abstract *authentication channel* which provides authenticity of everything received by $B$ [9]. $N_B$ is any value which can be verified by $B$ as fresh.

$$A \xrightarrow{a} B : N_B$$

This can be made concrete in a variety of ways. Mutual liveness between $A$ and $B$ who share a key $K_{AB}$ can be achieved in the following protocol.

1. $A \longrightarrow B : N_A$
2. $B \longrightarrow A : MAC_{K_{AB}}(B, N_A), N_B$
3. $A \longrightarrow B : MAC_{K_{AB}}(A, N_B)$

The inclusion of the identifiers in messages 2 and 3 ensure that messages of each entity can be recognised by themselves (which is merely a way of saying that the authentication channels are correctly implemented). The intended semantics of, say, message 2 is 'I am $B$ and I am alive'.

To extend this to provide entity authentication it is necessary to convey the semantics: 'I am $B$ and I wish to speak with $A$'. This can be achieved by adding the intended partner to the abstract protocol.

$$A \xrightarrow{a} B : B, N_B$$

Again, for the concrete version the name of the sender must be included to secure the authentication channel.

1. $A \longrightarrow B : N_A$
2. $B \longrightarrow A : MAC_{K_{AB}}(B, A, N_A), N_B$
3. $A \longrightarrow B : MAC_{K_{AB}}(A, B, N_B)$

Protocols similar to this one have been published in the literature (although it is not known whether this exact one has been suggested before). An attack on the above protocol is possible[2] which is very similar to some previously published attacks [6, 12]. In this attack $A$ is used as an 'oracle' by the attacker $C$.

1. $C_A \longrightarrow B : N_C$
2. $B \longrightarrow C_A : MAC_{K_{AB}}(B, A, N_A), N_B$

1'. $C_B \longrightarrow A : N_B$

2'. $A \longrightarrow C_B : MAC_{K_{AB}}(A, B, N_B), N_A$

3. $C_A \longrightarrow B : MAC_{K_{AB}}(A, B, N_B)$

Such an attack certainly violates the canonical intensional specification (as well as many other intensional ones) since $B$ accepts but the protocol has not run correctly. On the other hand has the extensional specification failed? $B$ believes that $A$ is prepared to communicate with him, and indeed we see that $A$ was sent a challenge by someone purporting to be $B$ and indeed replied with a message to the effect that she was prepared to communicate with $B$. Thus $B$ has not been deceived and the extensional goal is not violated.

Protocols using public key signatures may also be derived by similar arguments, as can ones using confidentiality. Consider the following which uses signatures of $A$ and $B$ in place of the $MAC$s used above.

1. $A \longrightarrow B : N_A$
2. $B \longrightarrow A : S_A(\{B, A, N_A\}), N_B$
3. $A \longrightarrow B : S_B(\{A, B, N_B\})$

In order to illustrate the ease with which new protocols may be designed using extensional goals, consider the following *conference* authentication protocol. The idea of such a protocol would be that all users $U_i$ in a set $\mathcal{U}$ should have confidence that all other users are ready to participate in a conference now. So far as is known, no such protocol is published previously to solve this problem.

Each user needs to authenticate a message to each other user with the semantics: 'I am $U_i$ and I want to communicate with the group $\mathcal{U}$'. This is most easily accomplished using a digital signature so that all users may authenticate the same message. Each user, $U_i, 1 \leq i \leq n$, choose a fresh random value $N_i$. In the following, $X \longrightarrow *$ denotes that the user $X$ broadcasts the message to all users, while the function $h$ is any one-way hash function. The protocol consists of two phases, in each of which each user broadcasts one message.

[2] I am very grateful to Anish Mathuria for pointing out this attack.

1. $U_i \longrightarrow * : N_i$
2. $U_i \longrightarrow * : Sig_{U_i}(\mathcal{U}, h(N_1|N_2|\ldots|N_n))$

Each user on receipt of the second set of messages verifies the signature. The protocol ensures to each user that the signature is fresh because the input to $h$ is fresh and hence the value $h(N_1|N_2|\ldots|N_n)$ is fresh.

## 6 Conclusion

Extensional goals seem more important for the protocol designer than intensional ones. It has been suggested in this paper that attacks should be measured by whether or not they violate extensional specifications, even if intensional specifications have been used to find the attacks in the first place. It has been shown that it is possible to find extensional specifications for entity authentication. It is a challenge to formalise the work in this paper to provide ways prove that extensional goals are satisfied. It may be useful to find extensional specifications for other more complex protocols, such as those being proposed for electronic commerce payments.

## Acknowledgements

I am very grateful to Anish Mathuria for many constructive critical comments.

## References

[1] M. Abadi, "Explicit Communication in Authentication: Two New Examples", *IEEE Transactions on Software Engineering*, 1997, *to appear.*

[2] M. Abadi and R. Needham, "Prudent Engineering Practice for Cryptographic Protocols" *DEC SRC Research Report 125*, June 1994.

[3] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution", *Advances in Cryptology - Crypto'93* Springer-Verlag, 1994, pp.232-249.

[4] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption", *Advances in Cryptology - Eurocrypt 94*, Springer-Verlag, 1995, pp. 92-111.

[5] M. Bellare and P. Rogaway, "Provably Secure Session Key Distribution — the Three Party Case", *Proceedings of the 27th ACM Symposium on the Theory of Computing*, 1995.

[6] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva and M. Yung, "Systematic Design of a Family of Attack-Resistant Authentication Protocols", *IEEE Journal on Selected Areas in Communications*, 11, 5, pp.679-693, June 1993.

[7] S. Blake-Wilson and A. Menezes, "Security Proofs for Entity Authentication and Authenticated Key Transport Protocols Employing Asymmetric Techniques", Security Protocols Workshop, 1997.

[8] M. Blum and S. Goldwasser, "An Efficient Probabilistic Public-Key Encryption Scheme which Hides All Partial Information", *Advances in Cryptology - Crypto 84*, pp.289-299, Springer-Verlag, 1985.

[9] C. Boyd, "A Framework for Design of Key Establishment Protocols", *Information Security and Privacy*, LNCS 1172, pp.146-157, Springer-Verlag, 1996.

[10] M. Burrows, M. Abadi and R. Needham, "A Logic of Authentication", *Proceedings of the Royal Society*, Series A, 246, (1989), pp.233-271.

[11] W. Diffie and M. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, 22, pp.644-654, 1976.

[12] W. Diffie, P. van Oorschot and M. Wiener, "Authentication and Authenticated Key Exchange", *Designs, Codes and Cryptography*, 2, 1992, pp.107-125.

[13] D. Gollman, "What do we Mean by Entity Authentication", *IEEE Symposium on Security and Privacy*, pp.46-54, 1996.

[14] J.W. Gray III, "On the Clark-Jacob Version of SPLICE/AS", *Information Processing Letters*, 1997, *to appear.*

[15] R. Kemmerer, C. Meadows and J. Millen, "Three Systems for Cryptographic Protocol Analysis", *Journal of Cryptology*, 7,2, pp.79-130, 1994.

[16] G. Lowe, "Breaking and Fixing the Needham-Schroeder Public Key Protocol using FDR", *Tools and Algorithms for the Construction and Analysis of Systems*, Springer-Verlag, 1996, pp.147-166.

[17] G. Lowe, "Some New Attacks upon Security Protocols", *9th IEEE Computer Security Foundations Workshop*, IEEE Press 1996, pp.162-169.

[18] G. Lowe, "A Hierarchy of Authentication Specification", *10th IEEE Computer Security Foundations Workshop*, IEEE Press, 1997.

[19] C. Meadows, "The NRL Protocol Analyzer: An Overview", *Journal of Logic Programming*, 26,2, 1996, pp.113-131.

[20] J. Millen, "CAPSL: Common Authentication Protocol Specification Language" Document maintained at web site: http://www.mitre.org/research/capsl/.

[21] R. Needham and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers", *Communications of the ACM*, 21, pp.393-399, 1978.

[22] B. Preneel and P. van Oorschot, "MDx-MAC and Building Fast MACs from Hash Functions", *Advances in Cryptology - Crypto'95*, Springer-Verlag, 1995, pp.1-14.

[23] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", *Communications of the ACM*, 21, pp.120-126, 1978.

[24] A. W. Roscoe, "Modelling and Verifying Key Exchange Protocols using CSP and FDR", *8th IEEE Computer Security Foundations Workshop*, pp.98-107, IEEE Press 1995.

[25] A. W. Roscoe, "Intensional Specifications of Security Protocols", *9th IEEE Computer Security Foundations Workshop*, pp.28-38, 1996.

[26] RSA Laboratories, "PKCS #1: RSA Encryption Standard", Version 1.5, November 1993.

[27] S. Schneider, "Verifying Authentication Protocols with CSP", *10th IEEE Computer Security Foundations Workshop*, IEEE Press, 1997.

[28] P. Syverson and P. van Oorschot, "On Unifying Some Cryptographic Protocol Logics", *1994 IEEE Symposium on Research in Security and Privacy*, pp. 14-28, IEEE Computer Society Press, 1994.

[29] P. Syverson and P. van Oorschot, "A Unified Cryptographic Protocol Logic", Draft available from the authors, 1996.

[30] P. Syverson (Moderator), "Panel: What is an Attack on a Cryptographic Protocol?", *9th IEEE Computer Security Foundations Workshop*, p.188, IEEE Press, 1996.

[31] P. Syverson and C. Meadows, "Formal Requirements for Key Distribution Protocols", *Advances in Cryptology - Eurocrypt'94*, Springer-Verlag, 1995, pp.320-331.

[32] R. Yahalom, "Optimality of Asynchronous Two-Party Data-Exchange Protocols", *Journal of Computer Security*, 2, 2-3, 1993, pp.191-209.