Errors in Attacks on Authentication Protocols*

Anders Moen Hagalisletto
Department of Informatics, University of Oslo
Postbox 1080 Blindern, 0316 Oslo, Norway, Email: andersmo@ifi.uio.no

Abstract

A tool for automated validation of attacks on authentication protocols has been used to find several flaws and ambiguities in the list of attacks described in the well known report by Clark and Jacob. In this paper the errors are presented and classified. Corrected descriptions of the incorrect attacks are given for the attacks that can be easily repaired.

Keywords: Security protocols, attacks, validation

1 Introduction

The report "A Survey of Authentication Protocol Literature: Version 1.0" by Clark and Jacob [9] (in this paper denoted Clark/Jacob), has been used extensively by experts on security protocols as the main reference on authentication protocols.[1] This publication was a major achievement in security protocol design and analysis, and it was intended to be a "living document" that should be regularly updated with corrections, new protocols and attacks [9, p. 6]. Tools for analyzing protocols have used the attacks in the Clark/Jacob report as a benchmark for evaluating protocol analyzers ([6], [12], [5], [19], [4] [14], [10], [3]) or as a reference to protocol specifications and attacks ([8], [17]). It is therefore important to obtain correct knowledge about which attacks are correct. Typical claims have been:

> "So far, about 40 protocols from [9] have been analyzed on which all the previously known attacks are detected, as well as new ones." [5, p. 16]

A new tool for analyzing descriptions of attacks has revealed that 7 of 23 attacks in Clark/Jacob contain flaws. Four of these incorrect attacks contains more than one error. Most of the errors reported here have been discovered in a fully automated way by the static validator of the protocol simulator PROSA [15]. The validation is split into several steps: First an attack description is specified in a formal language for protocols. Then the formal attack description is refined in an automated way into a description including the assumptions about actions that the agents is required to perform and assertions that the agent should possess. Finally, the validator checks whether each element in the attack has been obtained in a legal way either by past communication or by the cryptographic operations. PROSA has been used to find ambiguities in two additional attacks in Clark/Jacob, by showing exactly where the attack descriptions need to be adjusted. This paper also shows how several errors and ambiguities in the attack descriptions can be resolved by slight modification of the attacks.

To the best of the author's knowledge, the severe errors reported in this paper have not been published before. Some of the minor errors reported here are corrected in the Security Protocol Open Repository[2] (SPORE) [13], and some attacks are left out from the webpage which might indicate that some researchers might be aware of the errors. Unfortunately, SPORE is not a systematic update of Clark/Jacob: Flawed but repairable attacks are left out, errors have migrated to SPORE from Clark/Jacob, and correct attacks have been left out. For the sake of completeness both the severe errors and the misprints are included in this paper.

The paper is organized as follows: In Section 2, a classification of errors is presented. The incorrect attacks and their analysis are presented in Section 3. Some attack descriptions in Clark/Jacob are incomplete although not incorrect, revisions proposed by the validator are presented in Section 3.3. The results are discussed and compared with SPORE in Section 4, and finally in Section 5 concluding remarks are given.

*Thanks to Peter Csaba Ölveczky, Olaf Owe, Chik How Tan, David Basin, Atle Refsdal, Thor Kristoffersen, Bjarte M. Østvold, Habtamu Abie, Wolfgang Leister, Joakim Bjørk and five anonymous referees for comments on earlier drafts of this paper.

[1] At 27 November 2006, the report had 50 citations by *citeseer*.

[2] The webpage SPORE is the self-proclaimed successor of the Clark/Jacob report. Since the Internet page might be updated continuously, this paper refers to the version 27 November 2006.

2 Classification of errors

Every attack in the report by Clark/Jacob Clark/Jacob has been checked using the validator. Errors in attack descriptions can be divided into four categories:

(a) *misprints*,
(b) *man-in-the-middle errors*,
(c) *incompleteness of assumptions*, and
(d) *protocol jumps*.

The *man-in-the-middle flaw* is a common type of specification error; it means that the intruder should have intercepted and forwarded a message earlier in the protocol session. In order to make the attack description precise a protocol clause

$$(P) \quad A \longrightarrow B \quad : \quad M$$

meaning "agent A sends a message M to the agent B", should be replaced by the two clauses

$$\begin{array}{lll} (P_1) & A \longrightarrow I(B) & : \quad M \\ (P_2) & I(A) \longrightarrow B & : \quad M \end{array}$$

where the notation $I(B)$ means that the intruder I impersonates the agent B. Hence in (P_1) intruder I intercepts the message intended to be received by B, while in (P_2), the intruder I sends a message pretending to be A.

Incompleteness of assumptions means that there are some data in the protocol, like keys, nonces, timestamps or ciphertexts that an agent is assumed to be aware of at a given point in the attack description, but these data have not been obtained through past communication and legal decryption.

The final type of error, *protocol jump* means that an honest agent involved in the attack description does not follow the protocol that is supposed to be under attack.

Misprints and man-in-the-middle errors are easy to fix, and the attacks were revised and then validated as correct attacks. Most implementations of the Dolev-Yao model assumes that the attacker controls the network, hence every honest message is intercepted by the attacker (P_1). If an attack is discovered by an automated analysis tool for security protocols, then typically a proper subset of the interceptions is required in order for the attack to succeed. Based on the result of this inquiry we recommend that the exact interceptions should be stated explicit in any attack description. Two attacks in Clark/Jacob lacking man-in-the-middle clauses turned out to contain severe errors: it is likely that these errors could have been discovered if the attack description had been complete in the first place.

Incompleteness of assumptions and protocol jumps tend to indicate severe errors in the attacks, some of which are not easily repaired. Flaws of kind $(a-c)$ were typically found by the validator, while mistakes of kind (d) were discovered by the validator except the errors in the Shamir Rivest Adelman.

3 A collection of errors in attacks

In the following section we present the collection of errors found by the validator. First we present five attacks that contain severe errors. Then two attacks only containing misprints are discussed. Both the protocol specifications and attack descriptions are given in a notation similar to Clark/Jacob. The messages in the protocols consists of basic entities as follows:

A, B, C, S, I, $I(A)$	agent terms
K_{AB}	symmetric key shared by A and B
K_A	A's public key
K_A^{-1}	A's private key
N_A	nonce generated by agent A
T_A	timestamp generated by agent A

There are two composition operators in the notation: concatenation denoted by "," (comma) and encryption denoted by $E(K \ : \ M)$, where K denotes a key and M a message content.

3.1 Attacks containing severe errors

The attacks presented in this section contain at least one severe error each: either incompleteness of assumptions (c) or protocol jumps (d). Two of the protocols, the Wide Mouthed Frog and the Denning Sacco Public Key Protocol, additionally contain misprints (a) and man-in-the-middle errors (b). Later, in Section 4 (Table 1), an overview of the errors found is given.

3.1.1 The Wide Mouthed Frog

The protocol [9, p. 48], was proposed by Mike Burrows as "(...) perhaps the simplest protocol that uses shared-key cryptography and an authentication server" [7, p. 25]:

$$\begin{array}{lll} (\text{WMF}_1) & A \longrightarrow S & : \ A, E(K_{AS} : T_A, B, K_{AB}) \\ (\text{WMF}_2) & S \longrightarrow B & : \ E(K_{BS} : T_S, A, K_{AB}) \end{array}$$

An attack on the protocol was presented by Anderson and Needham in [2, p. 429], and formally described in Clark/Jacob. Two mistakes in the attack on the Wide Mouthed Frog protocol given in Clark/Jacob, were found by the validator. The attack by Clark/Jacob states:

$$\begin{array}{lll} (W.1.1) & A \longrightarrow S & : \ A, E(K_{AS} : T_A, B, K_{AB}) \\ (W.1.2) & S \longrightarrow B & : \ E(K_{BS} : T_S, A, K_{AB}) \\ (W.2.1) & I(B) \longrightarrow S & : \ B, E(K_{BS} : T_S, A, K_{AB}) \\ (W.2.2) & S \longrightarrow I(A) & : \ E(K_{AS} : T'_S, B, K_{AB}) \\ (W.3.1) & I(A) \longrightarrow S & : \ A, E(K_{AS} : T'_S, B, K_{AB}) \\ (W.3.2) & S \longrightarrow I(B) & : \ E(K_{BS} : T''_S, A, K_{AB}) \\ (W.4.1) & A \longrightarrow I(S) & : \ E(K_{AS} : T'_S, B, K_{AB}) \\ (W.4.2) & I(S) \longrightarrow B & : \ E(K_{BS} : T''_S, A, K_{AB}) \end{array}$$

The mistakes are not easy to spot at glance. The first mistake occurs in line $(W.2.1)$: The intruder I has not obtained $E(K_{BS} : T_S, A, K_{AB})$. The reason is that I is not intercepting the second message $(W.1.2)$. The second problem occurs with the transmission $(W.4.1)$: The agent A does not have any way of deducing the already created timestamp T'_S from what has happened previously in the attack. A corrected version of the attack can be given as follows:

$$
\begin{array}{lll}
(W.1.1) & A \longrightarrow S & : A, E(K_{AS} : T_A, B, K_{AB}) \\
(W.1.2.a) & S \longrightarrow I(B) & : E(K_{BS} : T_S, A, K_{AB}) \\
(W.1.2.b) & I(S) \longrightarrow B & : E(K_{BS} : T_S, A, K_{AB}) \\
(W.2.1) & I(B) \longrightarrow S & : B, E(K_{BS} : T_S, A, K_{AB}) \\
(W.2.2) & S \longrightarrow I(A) & : E(K_{AS} : T'_S, B, K_{AB}) \\
(W.3.1) & I(A) \longrightarrow S & : A, E(K_{AS} : T'_S, B, K_{AB}) \\
(W.3.2) & S \longrightarrow I(B) & : E(K_{BS} : T''_S, A, K_{AB}) \\
(W.4.1.a) & A \longrightarrow I(S) & : A, E(K_{AS} : T'_A, B, K_{AB}) \\
(W.4.2) & I(S) \longrightarrow B & : E(K_{BS} : T''_S, A, K_{AB})
\end{array}
$$

In this description, message $(W.1.2)$ is replaced by a man-in-the-middle interception: $(W.1.2.a)$ and $(W.1.2.b)$. The application clause $(W.4.1)$ was not a sentence that could be interpreted as belonging to a session of the Wide Mouthed Frog protocol. In $(W.4.1.a)$ the agent A starts a re-authentication of the agent B with a new timestamp T'_A, and is fooled in message $(W.4.2)$ to believe that S has replied with the appropriate timestamp T''_S. The timestamp is not fresh, B falsely believes that S has been involved in the last session.

3.1.2 Yahalom

The Yahalom protocol [9, p. 49] uses symmetric keys, in order to establish a new session key K_{AB} to be shared by agent A and B. The protocol was invented by Raphael Yahalom and presented in [7, p. 30]:

$$
\begin{array}{ll}
(\mathrm{Y}_1) & A \longrightarrow B : A, N_A \\
(\mathrm{Y}_2) & B \longrightarrow S : E(K_{BS} : A, N_A, N_B) \\
(\mathrm{Y}_3) & S \longrightarrow A : E(K_{AS} : B, K_{AB}, N_A, N_B), \\
 & \qquad\qquad\qquad E(K_{BS} : A, K_{AB}) \\
(\mathrm{Y}_4) & A \longrightarrow B : B, E(K_{BS} : A, K_{AB}), E(K_{AB} : N_B)
\end{array}
$$

In the attack presented in Clark/Jacob, the attacker tries to make the respondent B believe that the concatenation of nonces N_A, N_B plays the role of the new secret session key, in other words the attack is a typical type flaw:

$$
\begin{array}{lll}
(Y.1) & I(A) \rightarrow B : & A, N_A \\
(Y.2) & B \rightarrow I(S) : & E(K_{BS} : A, N_A, N_B) \\
(Y.3) & & \text{Omitted} \\
(Y.4) & I(A) \rightarrow B : & B, E(K_{BS} : A, N_A, N_B), \\
 & & \qquad E(N_A, N_B : N_B)
\end{array}
$$

The validator immediately found that the intruder I does not possess $E(N_A, N_B : N_B)$ before entering $(Y.4)$. There are two reasons why the intruder I cannot build the sentence $E(N_A, N_B : N_B)$: I does not possess N_B, and can neither build the fake key N_A, N_B nor the content N_B to be encrypted. There is no obvious way to repair this attack. Note that Donovan et. al. considered the attack to be erroneous without giving an explanation or analysis of how they considered it flawed [12, p. 6].

3.1.3 Woo Lam Π

Woo Lam's Π protocol [9, p. 51] is the final of a series of one-way authentication protocols initially presented in [20]:

$$
\begin{array}{lll}
\mathrm{WL}_1 & A \longrightarrow B & : A \\
\mathrm{WL}_2 & B \longrightarrow A & : N_B \\
\mathrm{WL}_3 & A \longrightarrow B & : E(K_{AS} : N_B) \\
\mathrm{WL}_4 & B \longrightarrow S & : E(K_{BS} : A, E(K_{AS} : N_B)) \\
\mathrm{WL}_5 & S \longrightarrow B & : E(K_{BS} : N_B)
\end{array}
$$

Clark/Jacob presents two attacks on the protocol. The final and rather obscure attack on the protocol [9, p. 53] is given by two interleaving sessions:

$$
\begin{array}{lll}
(L.1.1) & B \longrightarrow I & : B \\
(L.2.1) & I(A) \longrightarrow B & : A \\
(L.2.2) & B \longrightarrow I(A) & : N_B \\
(L.1.2) & I \longrightarrow B & : E(N_B : K_{IS}) \\
(L.1.3) & B \longrightarrow I & : E(E(N_B : K_{IS}) : K_{BS}) \\
(L.2.5) & I(S) \longrightarrow B & : E(N_B : K_{BS})
\end{array}
$$

The attack involves five type conversions. The two main conversions regard interpreting the nonce N as a key in $(L.1.2)$ and $(L.2.5)$, and interpreting the cipher-text $E(N : K_{IS})$ as a key in $(L.1.3)$. The validator reported that agent B had no reason to believe that $E(N_B : K_{IS})$ is a key. This problem can be solved by assuming that the equation $E(\text{Key} : M) = E(M : \text{Key})$ (†) holds. Then in clause $(L.1.3)$ agent B is encrypting with key K_{BS} and sending $E(K_{BS} : E(N_B : K_{IS}))$, while the intruder I is receiving $E(E(N_B : K_{IS}) : K_{BS})$ and decrypting with the key $E(N : K_{IS})$. But then $(L.2.5)$ turns out to be a problem, since B's interaction requires two missing intermediate protocol events $(L.2.3)$ and $(L.2.4)$. One may reinterpret the attack as one single session to avoid this protocol jump:

$$
\begin{array}{lll}
(L.1) & I(A) \longrightarrow B & : A \\
(L.2) & B \longrightarrow I(A) & : N_B \\
(L.3) & I(A) \longrightarrow B & : E(N_B : K_{IS}) \\
(L.4) & B \longrightarrow I(S) & : E(K_{BS} : E(N_B : K_{IS})) \\
(L.5) & I(S) \longrightarrow B & : E(N_B : K_{BS})
\end{array}
$$

The clause $(L.4)$ is not part of the protocol according to WL_4, agent B is not following the protocol, by omitting the agent name A. Instead it is possible to return to the original attack by including the two missing messages:

$$(L.2.3) \quad I(A) \longrightarrow B \quad : \quad M$$
$$(L.2.4) \quad B \longrightarrow I(S) \quad : \quad E(K_{BS} : A, M)$$

Since Woo and Lam require that the agents can detect replays, M must be chosen different from any of the previous messages and such that B can not decrypt according to (†). One such example is $M = E(K_{IS} : E(K_{IS} : N_B))$: M is no replay, and both K_{IS} and $E(K_{IS} : N_B)$ are kept secret to agent B.

3.1.4 Denning Sacco Public Key

The protocol [9, p. 63] uses certificates to establish a secure connection between two agents A and B:

$$(\text{DS}_1) \quad A \longrightarrow S \ : \ A, B$$
$$(\text{DS}_2) \quad S \longrightarrow A \ : \ \text{C}_A, \text{C}_B$$
$$(\text{DS}_3) \quad A \longrightarrow B \ : \ \text{C}_A, \text{C}_B, E(K_B : E(K_A^{-1} : K_{AB}, T_A))$$

The agent A uses two certificates, denoted C_A and C_B, that are distributed from a trusted server S in order to securely deliver a new session key K_{AB} to the agent B. The session key and a timestamp is signed by A's private key, in order to assure authenticity, and then encrypted with B's public key in order to provide secrecy. In the attack by Clark/Jacob the bad agent B is fooling an honest agent C to believe that B is running a session with agent A:

$$(D.3) \quad B(A) \longrightarrow C : \text{C}_A, \text{C}_C, E(K_C : E(K_A^{-1} : K_{AB}))$$

There is an obvious misprint, the timestamp T_A is left out in $(D.3)$. In the original attack by Abadi and Needham [1], the timestamp is included:

$$(D.3.a) \quad B(A) \longrightarrow C \ : \ \text{C}_A, \text{C}_C, E(K_C : E(K_A^{-1} : K_{AB}, T_A))$$

The attack relies on B's capabilities to initiate sessions and intercept any previous runs of the session, as described informally in the original paper by Abadi and Needham [1]:

$$(D.1.1) \quad A \longrightarrow S \quad : \quad A, B$$
$$(D.1.2) \quad S \longrightarrow A \quad : \quad \text{C}_A, \text{C}_B$$
$$(D.1.3) \quad A \longrightarrow B \quad : \quad \text{C}_A, \text{C}_B, E(K_B : E(K_A^{-1} : K_{AB}, T_A))$$
$$(D.2.1) \quad B(A) \rightarrow S : \quad A, C$$
$$(D.2.2) \quad S \rightarrow B(A) : \quad \text{C}'_A, \text{C}'_C$$
$$(D.2.3) \quad B(A) \rightarrow C : \quad \text{C}_A, \text{C}'_C, E(K_C : E(K_A^{-1} : K_{AB}, T_A))$$

But even this attack description which is derived from [1] is flawed. The final clause $(D.2.3)$ is a protocol jump for the honest agent C. This can be seen by examining the certificates ([11, p. 534] or [9, p. 63]) involved in the two protocol runs:

$$\text{C}_A = E(K_S^{-1} : A, K_A, T_S) \quad \text{C}_B = E(K_S^{-1} : B, K_B, T_S)$$
$$\text{C}'_A = E(K_S^{-1} : A, K_A, T'_S) \quad \text{C}'_C = E(K_S^{-1} : C, K_C, T'_S)$$

The agent C will not accept the certificates C_A, C'_C as belonging to a Denning Sacco session, since the certificates received are not synchronized with respect to the timestamp:

$$\text{C}_A, \text{C}'_C = E(K_S^{-1} : A, K_A, T_S), E(K_S^{-1} : C, K_C, T'_S)$$

This is an explicit part of the protocol, hence C will abort her session after decrypting the certificates. The attack can be repaired by replacing $(D.2.3)$ with the clause $(D.2.3.b)$, in the previous description:

$$(D.2.3.b) \ B(A) \longrightarrow C \ : \ \text{C}'_A, \text{C}'_C, E(K_C : E(K_A^{-1} : K_{AB}, T_A))$$

In this message both the certificates of A and C are synchronized on the timestamp T'_S. Agent C can not detect that B is the originator: Hence C is successfully fooled to believe that A sent her a new session K_{AB} key shared by A and C.

3.1.5 Shamir Rivest Adelman Three Pass (SRA)

The Shamir Rivest Adelman protocol [9, p. 64] assumes that encryption is commutative, which means the following: $E[k_1 : E[k_2 : F]] = E[k_2 : E[k_1 : F]]$ (‡).

$$(\text{SRA}_1) \quad A \longrightarrow B \quad : \quad E(K_A : M)$$
$$(\text{SRA}_2) \quad B \longrightarrow A \quad : \quad E(K_B : E(K_A : M))$$
$$(\text{SRA}_3) \quad A \longrightarrow B \quad : \quad E(K_B : M)$$

The second attack presented in the report contains two protocol jumps. The attack involves two interleaving sessions.

$$(R.1.1) \quad A \longrightarrow I(B) \quad : \quad E(K_A : M)$$
$$(R.2.1) \quad I(B) \longrightarrow A \quad : \quad E(K_A : M)$$
$$(R.2.2) \quad A \longrightarrow I(B) \quad : \quad M$$
$$(R.1.2) \quad I(B) \longrightarrow A \quad : \quad bogus$$
$$(R.1.3) \quad A \longrightarrow I(B) \quad : \quad E(K_A : bogus)$$

The first mistake in this attack occurs in event $(R.2.2)$ as a reply to $(R.2.1)$. Agent A is not following the protocol, she should have been replying $E(K_A : E(K_A : M))$ according to (SRA_2). In $(R.1.2)$ the attacker I may well send the message containing $bogus$, since A is expecting a message encrypted with B's public key. But it is incorrect to claim that A is sending $E(K_A : bogus)$ in $(R.1.3)$. Since it receives bogus, in its (first) session $R.1.2$, it is not able to do any decryptions, and the session aborts. The clause $R.1.3$ is nothing that A can do as the final step in the (first) session, since it is not in accordance with A's local understanding of the protocol. It could have been the start of a third session. This is the second protocol jump in the description. This attack is not easily repaired either.

3.2 Two attacks containing only misprints

3.2.1 Neuman Stubblebine

The protocol is split into two parts, exchanging tickets ($\text{NS}_1 - \text{NS}_4$) and repeated authentication ($\text{NS}_5 - \text{NS}_7$):

$$
\begin{array}{lll}
(\text{NS}_1) & A \longrightarrow B & : A, N_A \\
(\text{NS}_2) & B \longrightarrow S & : B, E(K_{BS} : A, N_A, T_B), N_B \\
(\text{NS}_3) & S \longrightarrow A & : E(K_{AS} : A, N_A, K_{AB}, T_B), \\
 & & \quad E(K_{BS} : A, K_{AB}, T_B), N_B \\
(\text{NS}_4) & A \longrightarrow B & : E(K_{BS} : A, K_{AB}, T_B), E(K_{AB} : N_B) \\
(\text{NS}_5) & A \longrightarrow B & : N'_A, E(K_{BS} : A, K_{AB}, T_B) \\
(\text{NS}_6) & B \longrightarrow A & : N'_B, E(K_{AB} : N'_A) \\
(\text{NS}_7) & A \longrightarrow B & : E(K_{AB} : N'_B),
\end{array}
$$

In the first attack on Neuman Stubblebine protocol [9, p. 57], the intruder I tries to fool B to accept the nonce N_A as a session key:

$$
\begin{array}{lll}
(N.1) & I(A) \longrightarrow B & : A, N_A \\
(N.2) & B \longrightarrow I(S) & : B, E(K_{BS} : A, N_A, T_B), N_B \\
(N.3) & & \text{Omitted} \\
(N.4) & I(A) \longrightarrow B & : E(K_{BS} : A, N_A, T_B), E(K_{AB} : N_B) \\
(N.5) & I(A) \longrightarrow B & : N'_A, E(K_{BS} : A, N_A, T_B) \\
(N.6) & B \longrightarrow I(A) & : N'_B, E(K_{AB} : N'_A) \\
(N.7) & I(A) \longrightarrow B & : E(K_{AB} : N'_B),
\end{array}
$$

The attack is not valid because of the final sentence $(N.7)$: The intruder I is not able to form $E(K_{AB} : N'_B)$, because the key K_{AB} is unknown to I. This might be a mistyped key by the authors. If we instead follow their own convention for notation and write

$$
\begin{array}{lll}
(N.6.a) & B \longrightarrow I(A) & : N'_B, E(N_A : N'_A) \\
(N.7.a) & I(A) \longrightarrow B & : E(N_A : N'_B),
\end{array}
$$

then the specification represents a correct attack.

3.2.2 Encrypted Key Exchange

In the protocol [9, p. 65], a password P is used as symmetric key to distribute a randomly generated public key K_A and a new session key R. The session key R is a secret key shared by A and B:

$$
\begin{array}{lll}
(\text{E}_1) & A \longrightarrow B & : E(P : K_A) \\
(\text{E}_2) & B \longrightarrow A & : E(P : E(K_A : R)) \\
(\text{E}_3) & A \longrightarrow B & : E(R : N_A) \\
(\text{E}_4) & B \longrightarrow A & : E(R : N_A, N_B) \\
(\text{E}_5) & A \longrightarrow B & : E(R : N_B)
\end{array}
$$

The attack is given as follows [9, p. 65]:

$$
\begin{array}{lll}
(E.1.1) & A \longrightarrow I(B) & : E(P : K_A) \\
(E.2.1) & I(B) \longrightarrow A & : E(P : K_A) \\
(E.2.2) & A \longrightarrow I(B) & : E(P : E(K_A : R)) \\
(E.1.2) & I(B) \longrightarrow A & : E(P : E(K_A : R)) \\
(E.1.3) & A \longrightarrow I(B) & : E(R : N_A) \\
(E.2.3) & I(B) \longrightarrow A & : E(R : N_A) \\
(E.2.4) & A \longrightarrow I(B) & : E(R : N_A, N_B) \\
(E.1.4) & I(B) \longrightarrow A & : E(R : N_A, N_B) \\
(E.1.5) & A \longrightarrow B & : E(R : N_B) \\
(E.2.5) & I(B) \longrightarrow A & : E(R : N_B)
\end{array}
$$

The validator reports that in between the transmission $(E.1.5)$ and $(E.2.5)$, the agent B had not obtained the key R in a valid manner. The reason is that R is a shared (symmetric) key between A and B. The agent B expects to be able to know R, yet B has not created R, or received R during the session. So B is correct in claiming ownership to the key, and the validator is correct in stating that B's claim is unjustified, since R is specified to be freshly generated in the protocol run! Protocol clause $(E.1.5)$ is a simple misprint, by instead replacing it with the clause

$$
(E.1.5.a) \quad A \longrightarrow I(B) \ : \ E(R : N_B),
$$

the attack becomes valid.

3.3 Concise descriptions of attacks

Two of the attacks are not directly incorrect, since the attacks lacked man-in-the-middle clauses that were, however, described informally in the text: Below necessary and sufficient criteria for making the attacks precise are given:

3.3.1 Andrew Secure RPC

The protocol is given in [9, p. 45] (initially presented by M. Satyanarayanan in [18, p. 256]).

$$
\begin{array}{lll}
(\text{A}_1) & A \longrightarrow B & : A, E(K_{AB} : N_A) \\
(\text{A}_2) & B \longrightarrow A & : E(K_{AB} : N_A + 1, N_B) \\
(\text{A}_3) & A \longrightarrow B & : E(K_{AB} : N_B + 1) \\
(\text{A}_4) & B \longrightarrow A & : E(K_{AB} : K'_{AB}, N'_B)
\end{array}
$$

The attack by Clark/Jacob [9, p. 24] states that the intruder intercepts the third message $(A.3)$, and then impersonates as Bob and replays message $(A.2)$ in $(A.4)$. Hence the intruder tries to fool A to believe that the nonce $N_A + 1$ is the new session key K'_{AB}.

$$
\begin{array}{lll}
(A.1) & A \longrightarrow B & : A, E(K_{AB} : N_A) \\
(A.2) & B \longrightarrow A & : E(K_{AB} : N_A + 1, N_B) \\
(A.3) & A \longrightarrow I(B) & : E(K_{AB} : N_B + 1) \\
(A.4) & I(B) \longrightarrow A & : E(K_{AB} : N_A + 1, N_B)
\end{array}
$$

Validation of the attack reveals that second message must be intercepted and forwarded by the intruder, hence message $(A.2)$ should be replaced by:

$(A.2.a) \quad B \longrightarrow I(A) \quad : \quad E(K_{AB} : N_A + 1, N_B)$
$(A.2.b) \quad I(B) \longrightarrow A \quad : \quad E(K_{AB} : N_A + 1, N_B)$

3.3.2 Neuman Stubblebine

The second attack on the protocol [9, p. 58] involves one previous session of the initiation phase, corresponding to $(N.2.1 - N.2.4)$, where the third message (NS_3) is intercepted and forwarded $(N.2.3.a)$ and $(N.2.3.b)$, hence Clark/Jacob's attack should be modified as follows:

$(N.2.1) \quad A \longrightarrow B \quad : A, N_A$
$(N.2.2) \quad B \longrightarrow S \quad : B, E(K_{BS} : A, N_A, T_B), N_B$
$(N.2.3.a) \; S \rightarrow I(A) \; : E(K_{AS} : A, N_A, K_{AB}, T_B),$
$E(K_{BS} : A, K_{AB}, T_B), N_B$
$(N.2.3.b) \; I(S) \rightarrow A \; : E(K_{AS} : A, N_A, K_{AB}, T_B),$
$E(K_{BS} : A, K_{AB}, T_B), N_B$
$(N.2.4) \quad A \longrightarrow B \quad : E(K_{BS} : A, K_{AB}, T_B),$
$E(K_{AB} : N_B)$
$(N.2.5) \quad I(A) \rightarrow B : N'_A, E(K_{BS} : A, K_{AB}, T_B)$
$(N.2.6) \quad B \rightarrow I(A) : N'_B, E(K_{AB} : N'_A)$
$(N.3.5) \quad I(A) \rightarrow B : N'_B, E(K_{BS} : A, K_{AB}, T_B)$
$(N.3.6) \quad B \rightarrow I(A) : N''_B, E(K_{AB} : N'_B)$
$(N.2.7) \quad I(A) \rightarrow B : E(K_{AB} : N'_B),$

4 Discussion

The validity of 23 attacks on 15 protocols have been analyzed. Table 1 gives an overview of the results of the analysis: All the four error types described in Section 3.3 are represented. The validator reported problems with 9 attacks, that can be divided into three groups: incomplete attacks, misprints, and severe flaws. Two of the attacks only contained misprints: the first attack on Neuman Stubblebine and the attack on Encrypted Key Exchange. Both attacks are easily adjusted to form real attacks. The remaining five attacks included 11 errors, six severe, three misprint and two man-in-the-middle flaws. These included two attacks that are not easily repaired, the one on Yahalom and the second on Shamir Rivest Adelman (containing two severe errors). The attack of the Wide Mouthed Frog could be redesigned in conformance with the intentions of the informal description given by Clark/Jacob, by making the implicit interception explicit, and modifying the two final clauses. The attack on the Woo Lam Π protocol was interpreted in three ways, each gives rise to errors. Fortunately the attack was possible to repair in conformance with Clark/Jacobs requirements. The attack on the Denning Sacco Public Key protocol could be made explicit and the erroneous certificate was replaced successfully. It is interesting to observe that the error occured in specifications lacking explicit descriptions of intermediate interceptions (the missing session) and where essential parts of the cryptographic primitives (the certificates) were suppressed.

Protocol attacks *Errors from Section 3*	Kind of flaw	SPORE	Repairable
Wide Mouthed Frog	(b), (c/d)	not corrected	yes
Yahalom	(c)	not included	no
Woo Lam Π	(c) or (b) or (d)	not included	yes
Denning Sacco Public Key	(a), (b), (d)	not included	yes
Shamir Rivest Adelman	(d), (d)	not included	no
Neuman Stubblebine (1)	(a)	corrected	yes
Encrypted Key Exchange	(a)	not included	yes
Section 3.3			
Andrew Secure RPC	(b)	not included	yes
Neuman Stubblebine (2)	(b)	not corrected	yes

Table 1. Analysis result.

It turns out that SPORE is not a systematic update of Clark/Jacob, as Table 1 shows. In the following each of the attacks presented in this paper is compared with the descriptions on the webpage [13]. In the SPORE description of the Wide Mouthed Frog the first error is not corrected, and the website does not mention the final two clauses where the second severe error occurs. Neither the Yahalom attack nor the two attacks on Shamir Rivest Adelman is mentioned, although the first attack on Shamir Rivest Adelman is correct. The original attack on Neuman Stubblebine in [16] and the specification in SPORE are both correctly described. The Encrypted Key Exchange protocol and Denning Sacco Public Key are not included in SPORE, nor are the attacks, even though the attacks could be easily repaired. The attack on Andrew Secure RPC occurring in Clark/Jacob is replaced in SPORE with the original attack from [7]. Finally the Neuman Stubblebine description on the webpage contains the same man-in-the-middle flaw as in Clark/Jacob. Hence there does not seem to be uniform criteria for transferring attacks and protocols from Clark/Jacob into SPORE.

5 Conclusion

The errors in the Shamir Rivest Adelman attack were not discovered by the tool. The reason is that the validator can not discover every possible protocol jump in every attack.A recent result [**?**] shows that the errors in the Shamir Rivest Adelman attack described in this paper can be detected by simulation. The model contained two honest agents Alice and Bob that possessed the Shamir Rivest Adelman protocol. An attacker Malice was configured to control the network, and executed the Clark/Jacob attack with Alice and Bob. Simulations showed that the attack failed to succed in exactly the same state described in the paper, and reachability analysis of the model showed that the attack failed in any simulation.

Several errors have been found in the most frequently cited library on security protocols. A close investigation re-

vealed that errors migrate from original papers to the report by Clark/Jacob, and from the report to SPORE and papers on protocol analysis. This show that flaws in protocol attacks do occur and that they can be hard to discover by humans. Our experience indicates that attack descriptions should be described as accurately as protocols and their correctness should be analyzed as formally as protocols.

References

[1] Martín Abadi and Roger Needham. Prudent Engineering Practice for Cryptographic Protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, 1996.

[2] Ross Anderson and Roger Needham. Programming Satan's Computer. In *Computer Science Today*, volume 1000 of *Lecture Notes in Computer Science*, pages 426–441. Springer-Verlag, 1995.

[3] Michael Backes, Sebastian Mödersheim, Birgit Pfitzmann, and Luca Viganò. Symbolic and Cryptographic Analysis of the Secure WS-Reliable Messaging Scenario. In *Proceedings of FOSSACS 2006*, volume 3921 of *Lecture Notes in Computer Science*, pages 428–445. Springer-Verlag, 2006.

[4] David Basin, Sebastian Mödersheim, and Luca Viganò. OFMC: A symbolic model checker for security protocols. *International Journal of Information Security*, 4(3):181–208, June 2005.

[5] Chiara Bodei, Pierpaolo Degano, Riccardo Focardi, Roberto Gorrieri, and Fabio Martinelli. Techniques for Security Checking: Non-Interference vs Control Flow Analysis,. In *Proceedings of the Theory of Concurrency, Higher Order and Types Workshop*, volume 62. Electronic Notes in Computer Science, Elsevier, 2001. Udine (Italy).

[6] Stephen H. Brackin. Evaluating and Improving Protocol Analysis by Automatic Proof. In *IEEE Computer Security Foundations Workshop (CSFW)*, pages 138–152, 1998.

[7] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. Technical report, february 1989. Report 39, Digital Systems Research Center.

[8] Carlos Caleiro. Deconstructing Alice and Bob. In P. Degano and L. Viganò, editors, *Proceeding of the Second Workshop on Automated Reasoning for Security Protocol Analysis*, pages 3–22. Electronic Notes in Computer Science, 2005.

[9] John Clark and Jeremy Jacob. A Survey of Authentication Protocol Literature, 1997. Version 1.0, Unpublished Report, University of York, `http://cs.york.ac.uk/~jac/papers/drareview.ps.gz`.

[10] Ricardo Corin, Sandro Etalle, and Ari Saptawijaya. A logic for constraint-based security protocol analysis. In *Security and Privacy*, pages 155–168. IEEE Computer Society Press, 2006.

[11] Dorothy E. Denning and Giovanni M. Sacco. Timestamps in Key Distribution Protocols. *Comm. of the ACM*, 24(8):533–536, 1981.

[12] Ben Donovan, Paul Norris, and Gavin Lowe. Analyzing a Library of Security Protocols using Casper and FDR. In *Proceedings of the Workshop on Formal Methods and Security Protocols*. IEEE Computer Society Press, 1999.

[13] Laboratoire Spécification et Vérification. SPORE Security Protocol Open REpository. `http://www.lsv.ens-cachan.fr/spore/`.

[14] Andrew Gordon and Alan Jeffrey. Types and effects for asymmetric cryptographic protocols. In *15th IEEE Computer Security Foundations Workshop (CSFW 15), Cape Breton*, pages 77–91. IEEE Press, 2002.

[15] Anders Moen Hagalisletto. An executable operational semantics for high level specifications of security protocols, 2006. Technical Report, University of Oslo.

[16] Tzonelih Hwang, Narn-Yih Lee, Chuan-Ming Li, Ming-Yung Ko, and Yung-Hsiang Chen. Two attacks on Neuman-Stubblebine authentication protocols. *Information Processing Letter*, 53(2):103–107, 1995.

[17] Romain Janvier, Yassine Lakhnech, and Laurent Mazaré. Relating the symbolic and computational models of security protocols using hashes. In *Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis*, pages 67–89, 2006.

[18] Mahadev Satyanarayanan. Integrating security in a large distributed system. *ACM Trans. Comput. Syst.*, 7(3):247–280, 1989.

[19] Dawn Song, Sergey Berezin, and Adrian Perrig. Athena, a Novel Approach to Efficient Automatic Security Protocol Analysis. *Journal of Computer Security*, 9(1,2):47–74, 2001.

[20] Thomas Y. C. Woo and Simon S. Lam. A Lesson on Authentication Protocol Design. *Operating Systems Review*, 28(3):24–37, 1994.