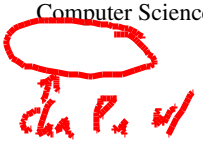
Chaotic Protocols*

Mohamed Mejri

Computer Science and Software Engineering Departement, Laval University,
Sainte-Foy, Qc,G1K 7P4, Canada.
mohamed.mejri@ift.ulaval.ca



Abstract. Generally, a piece of information (key, password, etc.) is classified as secret when it gives to its owner some privileges or protections. For instance, cryptographic keys can be used for several purposes such as to prove our identity (authentication), to keep our anonymity, to protect our data against malicious manipulation, etc. The secrecy property within cryptographic protocols is generally defined as follows : A given protocol P keeps a message M (M can be a key) secret, if an intruder will not be able to reveal the value of M during any execution of this protocol.
In this paper, we prove that some cryptographic protocols can involve the following amazing situation: **the intruder can never discover the value of a given key k but he is able to both encrypt and decrypt any message using this key k**. We baptize this kind of awful cryptographic protocols by **chaotic protocols**. This fact has been discovered when analyzing the one-way Woo and Lam authentication protocol using the Dymna approach which is specially developed for the verification of cryptographic protocols. Abadi and Needham present an attack against this protocol and proposed a new corrected version . Surprisingly, we prove in this paper that the new proposed version is also a chaotic protocol. Finally, some interesting features of chaotic protocols are discussed in this paper.

Keywords: Cryptographic Protocols, Secrecy, Inference System, Verification.

1 Motivations

Since the advent of Internet, the list of intrusions in computer systems, flights of information via this network and other security incidents, does not cease lengthening. Internet has not only given a perfect window to the tradesmen of the whole world who find the occasion to benefit from a world virtual market, but also gives many ideas to all those who look for easy money and those who find a great pleasure to ransack computer sites of others.

Cryptographic protocols (an orderly defined sequence of communication and computation steps using cryptography.) are the most effective and the widespread used means to meet security needs (confidentiality, integrity, authentication, non-repudiation,

* This research is supported by a research grant from the Natural Sciences and Engineering Council of Canada, NSERC, the "Fonds Québécois de la Recherche sur la Nature et les Technologies", FQRNT, and "Commission Permanente de Coopération Franco-Québécoise", CPCFQ.

A. Laganà et al. (Eds.): ICCSA 2004, LNCS 3043, pp. 938–948, 2004.

anonymity, goods and money atomicity, etc.). Therefore, the correctness of these protocols is paramount since the risks of their use are real, especially when they are involved in sensitive fields such as military (where there are human lives in danger) and banks (where there is a lot of money) and the least error can generate undesirable and often irreversible consequences.

Today, it is well-known that the design of cryptographic protocols is error prone. Several protocols have been shown flawed in computer security literature [4] many years after their publication and use. In spite of the interesting activities of research which led to correct a significant number of errors in the design of cryptographic protocols using different methods, the problem still not overcome and far from being controlled. This is due, on one hand, to the complexity and the subtlety of the cryptographic protocols themselves and, on the other hand, to the limitations of the current methods and techniques. A complete bibliography and a comparative study of these methods can be found in [2,3,4,8,9,10,11,12].

The main points addressed by this work are the following:

- we introduce a new class of cryptographic protocols called *chaotic protocols*. These protocols have the particularity to allow an intruder to both encrypt and decrypt any message he wants using unknown keys.
- we prove, using the Dymna approach [5,6,7], that the one-way Woo and Lam authentication protocol [13] is chaotic. Furthermore, we give the corrected version of this protocol proposed by Abadi and Needham in [1] and we prove that it is also chaotic.
- we exhibit some interesting features of chaotic protocols and we discuss their impact on the analysis of cryptographic protocols.

The remainder of this paper is organized as follows: In Section 2, we review the basic notation and terminology used within cryptographic protocols. In Section 3, we present how the Dymna approach can be used to analyze cryptographic protocols given in their standard notation. In Section 4, we introduce *chaotic protocols* (protocols that allow an intruder to encrypt and decrypt any message using unknown keys) and we use the Dyman approach to prove that both the original version Woo and Lam protocol together with the corrected one proposed by Abadi and Needham are chaotic. Finally, in Section 5, some concluding remarks and important features of chaotic protocols are ultimately sketched as a conclusion.

2 Basics

In this section, we introduce the basic notations that will be used throughout this paper. This protocol notation, which we refer to as the *standard notation*, is based on a fairly standard informal notation used by the security protocol community. A message is composed of one or more primitive words. A message m encrypted with key k is written $\{m\}_k$ and forms a word by itself. Concatenated messages are separated by commas. Message contents (words) have the following naming conventions: Encryption keys and nonces are respectively written k and N. Principals are written A, B, S and I, where A and B stand for principals who wish to communicate, S for a trusted server and I for a

potential intruder. Subscripts will be used to denote an association to a principal; thus, for example N_a is a nonce that belongs to A and k_{as} is a shared key between A and S. Here is the BNF syntax of messages:

$$\begin{array}{rl} m ::= & A \quad \text{Principal Identifier} \\ \mid & N_a \quad \text{Nonce} \\ \mid & k \quad \text{Key} \\ \mid & n \quad \text{Numeral} \\ \mid & X \quad \text{Message Variable} \\ \mid & \{m\}_k \quad \text{Encrypted Message} \\ \mid & m, m' \quad \text{Message Concatenation} \end{array}$$

A protocol P is a sequence of communication steps. Each step has a unique identifier and specifies the sender, the receiver and the transmitted message. More precisely P has to respect the following BNF grammar:

$$P ::= \langle i.\ A \rightarrow B : m \rangle \mid P.P$$

As an example, we give in Table 1 the one-way Woo and Lam authentication protocol [13,14]. This protocol relies on symmetric-key cryptography and allows a principal A to prove his identity to principal B. The description of the protocol can be read as follows: (1) A initiates the protocol and claims his identity to B; (2) B replies by sending the nonce N_b and asking A to encrypt it under k_{as} in order to prove what he claimed; (3) A returns the nonce N_b encrypted under k_{as}; (4) B forwards the response encrypted, together with A's identity, under k_{bs} for verification; (5) S decrypts the received message using B's key, extracts the encrypted component and decrypts it using A's key and re-encrypts under B's Key. If S replies $\{N_b\}_{k_{bs}}$, then B will find N_b after decrypting it and he should be convinced that A is really running this session with him.

Table 1. The Woo and Lam Protocol

$\langle 1.\ A \rightarrow B : A \rangle.$
$\langle 2.\ B \rightarrow A : N_b \rangle.$
$\langle 3.\ A \rightarrow B : \{N_b\}_{k_{as}} \rangle.$
$\langle 4.\ B \rightarrow S : \{A, \{N_b\}_{k_{as}}\}_{k_{bs}} \rangle.$
$\langle 5.\ S \rightarrow B : \{N_b\}_{K_{bs}} \rangle$

In the following section we describe the Dymna approach that will be used to analyze cryptographic protocols.

3 Approach

The main idea underlying the Dymna approach [5,6,7] is to come up with a model that captures in a finite way all the intruder abilities. Basically, the intruder's abilities are

formally captured by an inference system that take into consideration both the intruder's traditional abilities (encryption/decryption, composition/decomposition, etc.) together with additional abilities extracted from the analyzed protocol itself. Once generated, this inference system is used in a goal-directed way to search for a potential security flaw. More precisely, the verification process needs the following steps :

3.1 Role Extraction

A role is a protocol abstraction where the emphasis is put on a particular principal. For instance, in the case of the Woo and Lam protocol of Table 1, three roles, denoted $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{S}$, can be extracted. They respectively correspond to principals A, B and S and are defined as following:

$$\begin{aligned}
\mathcal{A} = \; & \langle i.1, A \rightarrow I(B) : A \rangle. \\
& \langle i.2, I(B) \rightarrow A : N_b^i \rangle. \\
& \langle i.3, A \rightarrow I(B) : \{N_b^i\}_{k_{as}} \rangle
\end{aligned}$$

$$\begin{aligned}
\mathcal{B} = \; & \langle i.1, I(A) \rightarrow B : A \rangle. \\
& \langle i.2, B \rightarrow I(A) : N_b^i \rangle. \\
& \langle i.3, I(A) \rightarrow B : \{N_b^i\}_{k_{as}} \rangle. \\
& \langle i.4, B \rightarrow I(S) : \{A, \{N_b^i\}_{k_{as}}\}_{k_{bs}} \rangle. \\
& \langle i.5, I(S) \rightarrow B : \{N_b^i\}_{k_{bs}} \rangle
\end{aligned}$$

$$\begin{aligned}
\mathcal{S} = \; & \langle i.4, B \rightarrow S : \{A, \{N_b^i\}_{k_{as}}\}_{k_{bs}} \rangle. \\
& \langle i.5, A \rightarrow B : \{N_b^i\}_{k_{bs}} \rangle
\end{aligned}$$

As we can see, we have added a session identifier (i) to each communication step. Furthermore, we have associated this identifier to each fresh message to capture the fact that values of these messages change from one session to another.

3.2 Role Generalization

From the roles, we extract what we call generalized roles. A generalized role is an abstraction of a role where some messages are replaced by variables. Intuitively, we replace a message or a component of message by a variable, if the receiver of this message could not do any verification on it. For instance, in the Woo and Lam protocol, the principal A receives, at the second step, a nonce N_b^i that he is not able to verify its value. Then, we replace the nonce N_b^i by a variable X and we obtain the following A's generalized role:

$$\begin{aligned}
\mathcal{A}_G = \; & \langle i.1, A \rightarrow I(B) : A \rangle. \\
& \langle i.2, I(B) \rightarrow A : X \rangle. \\
& \langle i.3, A \rightarrow I(B) : \{X\}_{k_{as}} \rangle
\end{aligned}$$

The principal B receives at the third step a message encrypted by an unknown key ($\{N_b^i\}_{k_{as}}$). Therefore, he cannot do any verification on this message and we can replace

it by a variable X. Therefore, the generalized role associated to B is:

$$\begin{array}{rl} \mathcal{B}_G = & \langle i.1,\, I(A) \rightarrow B \quad : A \rangle. \\ & \langle i.2,\, B \quad \rightarrow I(A) : N_b^i \rangle. \\ & \langle i.3,\, I(A) \rightarrow B \quad : X \rangle. \\ & \langle i.4,\, B \quad \rightarrow I(S) : \{A, X\}_{k_{bs}} \rangle. \\ & \langle i.5,\, I(S) \rightarrow B \quad : \{N_b^i\}_{k_{bs}} \rangle \end{array}$$

Finally, since S does not previously know the values of N_b^i, then this message can be replaced by X. In summary, the generalized role associated to S is the following:

$$\begin{array}{rl} \mathcal{S}_G = & \langle i.4,\, I(B) \rightarrow S \quad : \{A, \{X\}_{k_{as}}\}_{k_{bs}} \rangle. \\ & \langle i.5,\, S \quad \rightarrow I(B) : \{X\}_{k_{bs}} \rangle \end{array}$$

3.3 Proof System Generation

Starting from the generalized roles we extract a set of inference rules. Each rule corresponds to an output (message sent by an honest agent) in a generalized role. The rules premisses contain all the message received by the role until the corresponding output, whereas the conclusion contains the output message. The general form of an inference rule is:

$$\frac{p_1 \ldots p_n}{c}$$

where $p_1, \ldots, p_n$ and c are messages. Here is the way such an inference rule should be read: the intruder could supply the protocol with the messages $p_1, \ldots, p_n$ in order to get the message c from the protocol. Furthermore, we will endow each inference rule with a sequence of protocol steps (a scenario) showing *how* the intruder could instrument the protocol with the information $p_1, \ldots, p_n$ so as to get the message c.

The inference rules that could be extracted from the Woo and Lam protocol of Table 1 are given in Table 2. The rules R_1 and R_3 are extracted from the generalized role of A, the rule R_2 and R_4 from the generalized role of B, and the rule R_5 from the generalized role of the server. The rule R_2, for instance, states that the intruder could instrument the protocol so as to get the message $\{X\}_{k_{as}}$ provided that he supplies the protocol with X, whereas the scenario attached to this rule shows how this goal can be achieved.

3.4 Verification

Informally, the verification process consist on checking wether a protocol satisfies its security goals in the presence of powerful intruder. The intruder computation abilities are captured by a proof system that contains two parts:

- The computation abilities given by the protocol itself. This part contains all the rules extracted from the analyzed protocol. For the Woo and Lam protocol (Table 1), these rules are given by Table 2.
- Usual computation abilities: The intruder has an initial knowledge K_I generally made of the keys that he shares with other principals, nonces, the server identity and other principal identities. Furthermore, the intruder can encrypt and decrypt

Table 2. The Deductive Proof System Associated with the Woo and Lam Protocol

	Inference Rules	Scenarios
R_1	$\frac{}{A}$	$\langle i.1 \quad A \quad \longrightarrow \quad I(B) : A\rangle$
R_2	$\frac{A}{N_b}$	$\langle i.1\ I(A) \longrightarrow \quad B \quad : A\rangle.$ $\langle i.2 \quad B \quad \longrightarrow \ I(A) : N_b^i\rangle$
R_3	$\frac{X}{\{X\}_{k_{as}}}$	$\langle i.1 \quad A \quad \longrightarrow \ I(B) : A\rangle.$ $\langle i.2\ I(B) \longrightarrow \quad A \quad : X\rangle.$ $\langle i.3 \quad A \quad \longrightarrow \ I(B) : \{X\}_{k_{as}}\rangle$
R_4	$\frac{A, X}{\{A, X\}_{k_{bs}}}$	$\langle i.1\ I(A) \longrightarrow \quad B \quad : A\rangle.$ $\langle i.2 \quad B \quad \longrightarrow \ I(A) : N_b^i\rangle.$ $\langle i.3\ I(A) \longrightarrow \quad B \quad : X\rangle.$ $\langle i.4. \quad B \quad \longrightarrow \ I(S) : \{A, X\}_{k_{bs}}\rangle$
R_5	$\frac{\{A, \{X\}_{k_{as}}\}_{k_{bs}}}{\{X\}_{k_{bs}}}$	$\langle i.4\ I(B) \longrightarrow \quad S \quad : \{A, \{X\}_{k_{as}}\}_{k_{bs}}\rangle.$ $\langle i.5 \quad S \quad \longrightarrow \ I(B) : \{X\}_{k_{bs}}\rangle$

any message under known keys (rules R_e and R_d). In addition, he has the ability to compose (concatenate) and decompose messages (rules R_{c_1}, R_{c_2} and R_{c_3}). All these usual abilities are formalized by the rules given in Table 3.

Table 3. The Usual Abilities of the Intruder

Encryption	Decryption	Knowledge
$R_e : \frac{X \quad Y}{\{X\}_Y}$	$R_d : \frac{\{X\}_Y \quad Y}{X}$	$R_K : \frac{X \in K_I}{X}$
Left Decomposition	**Right Decomposition**	**Concatenation**
$R_{c_1} : \frac{X.Y}{X}$	$R_{c_2} : \frac{X.Y}{Y}$	$R_{c_3} : \frac{X \quad Y}{X.Y}$

The sequent $K_I \models_{\mathcal{P}} m$ is used, in the rest of this paper, to state that the intruder is able to know the message m, using his initial knowledge K_I, the rules of his usual abilities and the ones extracted from the protocole $\mathcal{P}$.

Now, it remains only to formalize security properties to be able to analyze cryptographic protocols. Within our approach, a security property has to be specified in terms of a set of constraints $\{K_I^1 \models_{\mathcal{P}} m^1, \ldots, K_I^n \models_{\mathcal{P}} m^n\}$, meaning that if the intruder is able to prove the sequents $K_I^1 \models_{\mathcal{P}} m^1, \ldots, K_I^n \models_{\mathcal{P}} m^n$, then the security property is not satisfied. For instance, the verification of the secrecy property (does a protocol $\mathcal{P}$ keeps a message m secret?) turns to check whether the sequent $K_I \models_{\mathcal{P}} m$ can be proved or not. Other interesting properties such authentication can also be amounted to constraints verification problem.

4 Chaotic Protocols

In this section we show some very important facts from the analysis of the Woo and Lam protocol using the previously described approach. The first one concerns the definition of the secrecy property which is the fact of keeping secret a given piece of information. This aspect of security is considered as the oldest and the best known. Even if they are different formalisations of this property, almost all of them lead to the same informal meaning: we say that a protocol preserves the secrecy of a message m, if it does not leak the value of m during its execution. The messages (parameters) of the protocol that have to be kept secret are generally cryptographic keys and other sensitive data. Now suppose that our secret information is a key k, then the questions that we address are : if we are sure that the intruder can never reveal the value of k, is that a guarantee that the intruder can never use k to both encrypt and decrypt any message he wants? If, it is not (i.e., even if the intruder does not know the value of a key, he still is able to both encrypt and decrypt any message he want using this key), then does keeping the key secret has any sense?

It is commonly known that the main use of cryptographic keys is to perform encryption and decryption. Surprisingly, the fact of being sure that the value of a key cannot be known by the intruder is not a sufficient protection for this key. In fact, as we will show with the Woo and Lam protocol, if the secret keys are involved in a badly designed protocols, then that may lead to a serious problem that consists in giving an intruder the power of both encrypting and decrypting any message he wants using keys that he does not know. Protocols that endow the intruder with this extra power are called *chaotic*.

Notice that there are many protocols, specially challenge-response authentication protocols, which may allow an intruder to encrypt or decrypt messages. But they will not allow him to both encrypt and decrypt messages.

Definition 1 (Chaotic Protocol). *Let $\mathcal{P}$ be a protocol and $\mathcal{K}$ a set of secret keys involved in $\mathcal{P}$. We say that $\mathcal{P}$ is chaotic with respect to $\mathcal{K}$, if it allows an intruder to both encrypt and decrypt any message he wants using any key in $\mathcal{K}$.*

Within our approach this definition can be formalized as follows : Let $\mathcal{P}$ be protocol and $\mathcal{K}$ a set of secret keys involved in $\mathcal{P}$ and K_I be the initial knowledge of the intruder ($K_I \cap \mathcal{K} = \emptyset$). The protocol $\mathcal{P}$ is chaotic, if the following two constraints can be resolved:

$$K_I \cup \{m\} \models_{\mathcal{P}} \{m\}_k$$
$$K_I \cup \{\{m\}_k\} \models_{\mathcal{P}} m$$

Now, let's prove that the Woo and Lam protocol given by Table 1 is a chaotic protocol with respect to $\{k_{as}\}$.

Theorem 1. *The Woo and Lam protocol given by Table 1 is a chaotic protocol with respect to $\{k_{as}\}$.*

Proof. **– The intruder can encrypt any message using the key k_{as}**: This goal can be formalized as follows:

$$K_I \cup \{m\} \models_{\mathcal{P}} \{m\}_{k_{as}}$$

The rule R_3 of Table 2 gives an immediate answer to our question. In fact, this rule states that the intruder can obtain any message $\{X\}_{k_{as}}$ provided that he can supply X.

Table 4. Proof Attached to $K_I \cup \{m\} \models_{\mathcal{P}} \{m\}_{k_{as}}$

$$R_3[m/X] : \frac{R_K : \dfrac{m \in K_I \cup \{m\}}{m}}{\{m\}_{k_{as}}}$$

The scenario attached to this rule explains how this possible. After applying the same substitution ($\{X \mapsto m\}$) to this scenario the proof of "$K_I \cup \{m\} \models_{\mathcal{P}} \{m\}_{k_{as}}$" is as shown by Table 5.

Table 5. Scenario associated with $K_I \cup \{m\} \models_{\mathcal{P}} \{m\}_{k_{as}}$

$\langle i.1 \quad A \quad \longrightarrow \quad I(B) : A \rangle.$
$\langle i.2. \; I(B) \quad \longrightarrow \quad A \quad : m \rangle.$
$\langle i.3. \quad A \quad \longrightarrow \quad I(B) : \{m\}_{k_{as}} \rangle$

– **The intruder can decrypt any message using the key** k_{as}: This goal can be formalized as follows:

$$K_I \cup \{\{m\}_{k_{as}}\} \models_{\mathcal{P}} m$$

Actually, this goal can be reached and the proof is given in Table 6. The scenario showing how this is possible is illustrated by Table 7.

The Woo and Lam protocol given in Table 1 is known to be flawed many years ago. In particular, Abadi and Needham present an attack to the protocol in [1] and suggest the new corrected version given in Table 8. Surprisingly, the new proposed version is also a chaotic protocol. Hereafter, we give the proof.

Theorem 2. *The Woo and Lam protocol given by Table 8 (corrected version) is a chaotic protocol with respect to* $\{k_{as}\}$.

Proof. – **The intruder can encrypt any message using the key** k_{as} : Using the inference system, we want to know whether the intruder can encrypt any message using the key k_{as}. This goal can be formalized as follows:

$$K_I \cup \{m\} \models_{\mathcal{P}} \{m\}_{k_{as}}$$

Table 6. Proof Attached to $K_I \cup \{\{m\}_{k_{as}}\} \models_{\mathcal{P}} m$

$$R_{c_3\sigma_4}\dfrac{R_K : \dfrac{A \in K_I \cup \{m\}}{A} \qquad R_K : \dfrac{\{m\}_{k_{as}} \in K_I \cup \{m\}}{\{m\}_{k_{as}}}}{A, \{m\}_{k_{as}}}$$

$$R_{d\sigma_1} : \dfrac{R_{5\sigma_2} : \dfrac{R_{e\sigma_3} : \dfrac{A, \{m\}_{k_{as}} \quad k_{is}}{\{A, \{m\}_{k_{as}}\}_{k_{is}}}}{\{m\}_{k_{is}}} \qquad R_K : \dfrac{k_{is} \in K_I \cup \{m\}}{k_{is}}}{m}$$

Where:
$\sigma_1 = [X \mapsto m; Y \mapsto k_{is}]$ $\sigma_2 = [X \mapsto m; B \mapsto I]$
$\sigma_3 = [X \mapsto A, \{m\}_{k_{as}}; Y \mapsto k_{is}]$ $\sigma_4 = [X \mapsto A; Y \mapsto \{m\}_{k_{as}}]$

Table 7. Scenario Attached to $K_I \cup \{m\}_{k_{as}} \models_{\mathcal{P}} m$

$\langle i.4\ I \longrightarrow S : \{A, \{m\}_{k_{as}}\}_{k_{is}} \rangle.$
$\langle i.5\ S \longrightarrow I : \{m\}_{k_{is}} \rangle$

Table 8. The Corrected Version of the Woo and Lam Protocol

$\langle 1.\ A \to B : A \rangle.$
$\langle 2.\ B \to A : N_b \rangle.$
$\langle 3.\ A \to B : \{N_b\}_{k_{as}} \rangle.$
$\langle 4.\ B \to S : A, B, \{N_b\}_{k_{bs}} \rangle.$
$\langle 5.\ S \to B : \{A, N_b\}_{K_{bs}} \rangle$

Table 9. Corrected Version: Scenario associated with $K_I \cup \{m\} \models_{\mathcal{P}} \{m\}_{k_{as}}$

$\langle i.1 \quad A \longrightarrow I(B) : A \rangle.$
$\langle i.2.\ I(B) \longrightarrow A : m \rangle.$
$\langle i.3. \quad A \longrightarrow I(B) : \{m\}_{k_{as}} \rangle$

The proof is same to the one given for the original version of the protocol and the attack scenario is as shown by Table 9.

– **The intruder can decrypt any message using the key** k_{as}
Using the inference system, we want to know whether the intruder can decrypt any message encrypted by the key k_{as}. This goal can be formalized as follows:

$$K_I \cup \{\{m\}_{k_{as}}\} \models_{\mathcal{P}} m$$

The proof is same to the one given for the original version of the protocol and the attack scenario is as shown by Table 10.

Table 10. Corrected Version: Scenario Attached to $K_I \cup \{m\}_{k_{as}} \models_{\mathcal{P}} m$

$\langle i.4\ I \longrightarrow S : \{A, \{m\}_{k_{as}}\}_{k_{is}} \rangle.$
$\langle i.5\ S \longrightarrow I : \{m\}_{k_{is}} \rangle$

5 Conclusion

We reported in this paper a new class of protocols, named chaotic protocols. Intuitively a protocol is chaotic if it allows an intruder to both encrypt and decrypt any message using keys without knowing the values of these keys. This serious problem has been discovered using the Dymna approach when analyzing the Woo and Lam one-way authentication protocol. Abadi and Needham present an attack against this protocol and proposed a new corrected version in [1]. However, we have proved that the new proposed version is also a chaotic protocol.

As a consequence, this find has arisen some important questions about how cryptographic protocols should be analyzed. The first one concerns the secrecy property. In fact, when analyzing cryptographic protocols this property is in almost all cases formalized in term of whether an intruder can directly know the supposed secret information. However, chaotic protocols show that this definition of secrecy may be inadequate since even it is satisfied some serious problems (intruder still be able to both encrypt and decrypt messages using keys that he never know their values) may persist.

Furthermore, we can extract at least the following important features of chaotic protocols:

- Any chaotic protocol with respect to a set of secret keys $\mathcal{K}$ fail to satisfy any security gaol (secrecy, authentication, integrity, anonymity, etc.) build on the top of keys in $\mathcal{K}$. Therefore, given a protocol and a security property we can analyze it by first understand on which keys these properties are based and then we check whether this protocol is chaotic with respect to these keys or not.
- If a protocol P is chaotic with respect to a set of secret keys $\mathcal{K}$ then any other protocol P' that run in parallel with P is also chaotic with respect to $\mathcal{K}$. Suppose for instance that P is the Woo and Lam of Table1 and P' another protocol that it is proved to be correct when analyzed alone. Suppose also that P' use the same key k_{ab} used in P, then an intruder can employ the protocol P as a cryptographic system to encrypt and decrypt any message using the key k_{ab} and use the results to attack P'. From

this fact, it follows that it could be dangerous to use same keys in different protocols as many persons do (using the same password for different purposes). Besides, the correctness of a given protocol cannot be ensured by simply analyzing the protocol alone without taking into consideration its environment (the other protocols that will be executed in parallel with it).

References

1. M. Abadi and R. Needham. Prudent Engineering Practice for Cryptographic Protocols. Technical report, SRC DIGITAL, June 1994.
2. L. Buttyan. Formal methods in the design of cryptographic protocols (state of the art). Technical Report No. SSC/1999/38, Swiss Federal Institute of Technology (EPFL), Lausanne, November 1999.
3. U. Carlsen. *Formal Specification and Analysis of Cryptographic Protocols*. PhD thesis, Thèse d'Informatique soutenue à l'Université PARIS XI, October 1994.
4. J. Clark and J. Jacob. A Survey of Authentication Protocol Literature. Unpublished Article Available at `http://dcpu1.cs.york.ac.uk/~jeremy`, August 1996.
5. M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi. A New Algorithm for Automatic Verification of Authentication Cryptographic Protocols. In *Proceedings of the DIMACS Workshop on Design and Formal Verification of Security Protocols, DIMACS Center, Core Building, Rutgers University, New Jersy, USA*, Sep 1997.
6. M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi. Formal Automatic Verification of Authentication Cryptographic Protocols. In *Proceedings of the First IEEE International Conference on Formal Engineering Methods, Hiroshima, International Convention Center, Japan*. IEEE Press, November 1997.
7. M. Debbabi, M. Mejri, N. Tawbi, and I. Yahmadi. From Protocol Specifications to Flaws and Attack Scenarios: An Automatic and Formal Algorithm. In *Proceedings of the Second International Workshop on Enterprise Security, Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, USA*. IEEE Press, June 1997.
8. R. Kemmerer, C. Meadows, and J. Millen. Three Systems for Cryptographic Protocol Analysis. *Journal of Cryptology*, 7(2):79–130, 1994.
9. A. Liebl. Authentication in Distributed Systems: A Bibliography. *Operating Systems Review*, 27(4):122–136, October 1993.
10. C. Meadows. Formal Verification of Cryptographic Protocols: A Survey. In *Proceedings of Asiacrypt 96*, 1996.
11. A. D. Rubin and P. Honeyman. Formal Methods for the Analysis of Authentication Protocols. Technical Report Technical report 93–7, Technical Report, Center for Information Technology Integration, 1993. University of Michigan. Internal Draft.
12. P. Syverson. Knowledge, Belief, and Semantics in the Analysis of Cryptographic Protocols. *Journal of Computer Security*, 1(3):317–334, 92.
13. T. Y. C. Woo and S. S. Lam. Authentication for Distributed Systems. *Computer*, 25(1):39–52, January 1992.
14. T. Y. C. Woo and S. S. Lam. A Lesson on Authentication Protocol Design. *Operating Systems Review*, pages 24–37, 1994.