Towards Symbolic Encryption Schemes

Ahmed, Naveed; Jensen, Christian D.; Zenner, Erik

Published in:
Computer Security – ESORICS 2012

Link to article, DOI:
10.1007/978-3-642-33167-1_32

Publication date:
2012

Document Version
Publisher's PDF, also known as Version of record

Towards Symbolic Encryption Schemes

Naveed Ahmed[1], Christian D. Jensen[1], and Erik Zenner[2]

[1] DTU Informatics, Denmark
{Naah,Christian.Jensen}@imm.dtu.dk
[2] University of Applied Sciences Offenburg, Germany
erik.zenner@hs-offenburg.de

Abstract. Symbolic encryption, in the style of Dolev-Yao models, is ubiquitous in formal security models. In its common use, encryption on a whole message is specified as a single monolithic block. From a cryptographic perspective, however, this may require a resource-intensive cryptographic algorithm, namely an authenticated encryption scheme that is secure under chosen ciphertext attack. Therefore, many reasonable encryption schemes, such as AES in the CBC or CFB mode[1], are not among the implementation options.

In this paper, we report new attacks on CBC and CFB based implementations of the well-known Needham-Schroeder and Denning-Sacco protocols. To avoid such problems, we advocate the use of refined notions of symbolic encryption that have natural correspondence to standard cryptographic encryption schemes.

Keywords: Encryption, Assumptions, Implementation.

1 Introduction

A private-key encryption scheme enables two honest parties that share a key to privately communicate over a network, in such a way that a dishonest man-in-middle, the adversary, is unable to gain any non-trivial information about the communication. The requirements of cryptographic encryption may include left-right indistinguishability (IND) and non-malleability (NM), which can be characterized in different attack settings [2].

Over the years, many abstractions of cryptographic encryption have been proposed. The most popular abstraction is the Dolev-Yao model [3]. In this symbolic model, two types of simplifications are introduced. Firstly, binary strings and functions are replaced by symbolic terms and derivation rules. In particular, this results in idealized encryption functions—either an adversary can decrypt a symbolic ciphertext (e.g., if he can derive the key) or the adversary gets absolutely no information about the plaintext. The second simplification is related to the capabilities of an adversary, namely the adversary is modelled as a non-deterministic strategy that is limited to selecting its actions from a small set of

[1] i.e., cipher block chaining (CBC) and cipher feedback mode of encryption (CFB) [1].

S. Foresti, M. Yung, and F. Martinelli (Eds.): ESORICS 2012, LNCS 7459, pp. 557–572, 2012.

(pre-defined) logic rules. The security models that use these two abstractions are commonly referred to as symbolic/formal security models.

A symbolic model is simpler than its cryptographic counterpart, and therefore one can avoid relatively complicated and long proofs of traditional cryptography. More importantly, computers can do the tedious job of proving (and similarly verifying) the proofs of security.

Unfortunately, any security assurance in a symbolic model does not automatically translate to the underlying computational cryptography and, therefore, to its hardware/software implementation. In any implementation of symbolic encryption, a system designer has to make certain security critical decisions, related to, e.g., mode of encryption, block alignment, and message authentication code. Many attacks targeting the implementation of encryption are known [4, 5].

One approach to address such issues is to always rely on the most stringent interpretation of encryption [2], i.e., an encryption scheme that is private and non-malleable against an adversary that has adaptive access to encryption and decryption oracles. Such strong requirements, however, often implies a resource-intensive implementation.

We note that encryption or decryption oracles are not present in many protocols. Moreover, the security of a protocol does not always depend on non-malleability or privacy of the encryption. Therefore, in our view, one should use symbolic encryption in such a way that it closely mimics an actual cryptographic encryption scheme. In this way, not only one can avoid many implementation related ambiguities but also a level of safe optimization can be achieved, e.g., if a protocol is secure with ECB based encryption.

Our contributions in this paper are summarized in the following.

We present new attacks on the CBC and CFB based implementations of the Needham-Schroeder symmetric-key (NSSK) protocol [6], without exploiting the previously known vulnerability [7]. These attacks also work with the seven-round version of the NSSK protocol [8], which is an improved version of the original NSSK protocol after the flaw [7] was discovered. Further, we report new attacks on CBC and CFB based implementations of the Denning-Sacco symmetric-key (DSSK) protocol [7], which is another improved version of the NSSK protocol, and which does not suffer any attacks to the best of our knowledge.

It is worth mentioning that the CBC mode is semantically secure in traditional CPA (chosen plaintext attack) model [9], and the CFB mode is secure against an even more powerful adversary who has an access to block-wise online encryption oracle [10]. Our attacks, although are CPAs, are against the protocol security and not against the CBC/CFB security, which indicates that these protocols entail more stringent requirements on encryption, such as non-malleability.

Further, we advocate a few refined ways of using symbolic encryption that have natural correspondence to standard cryptographic constructions. The refined notions require different implementation resources and, therefore, a level of safe optimization can be achieved while still relying on symbolic encryption.

The rest of the paper is arranged as follows. In Sect. 2, we briefly examine the prior art. Next, in Sect. 3 and Sect. 4, we present the new attacks. In Sect. 5,

we list down a few symbolic encryption schemes and show that these schemes provide different levels of security in a symbolic model. In Sect. 7, we discuss our contribution in a broader perspective, and in Sect. 8 we conclude our work. In the paper, exclusive-or ($\oplus$) is abbreviated as Xor and a distinction should be observed between *symbolic encryption* [11, 12] and *cryptographic encryption* [2, 13].

2 Related Work

Meadows [14] presents an extensive survey of the works that rely on symbolic encryption. We here do not discuss formal security analysis as such and only focus on the implementation perspective of symbolic encryption.

Moore [15] was probably the first to highlight the security problems that may occur in implementing symbolic encryption. Boyd [16] describes a few possible attacks on the NSSK protocol based on some strong assumptions such as the use of a stream cipher for encryption, however, the presentation does not come close to that of ours. Mao and Boyd [17] discuss some general vulnerabilities that may occur when using cipher-block-chaining mode for implementing encryption. Bellovin [4] reported vulnerabilities in the earlier versions of IPsec by exploiting CBC-mode encryption.

Stubblebine et al. [18] investigates modes of encryption for discovering *known pairs* and *chosen texts*, using the NRL Protocol Analyzer. Our attack makes use of chosen texts, in which a party can be used as an encryption oracle; this is then exploited by an adversary who obtains the ciphertext against a plaintext. In the same line of work, Kremer and Ryan [19] model ECB and CBC mode using Blanchet's protocol verifier. Interestingly, they use the NSSK protocol as a case study but stop after indicating the existence of chosen texts in the protocol. Nevertheless, the existence of chosen texts is quite common in cryptographic protocols and often does not lead to insecure encryption.

An interesting case is that of encryption-only-mode of IPsec, for which Paterson and Yau [5] exploited CBC mode of encryption. Their attacks work if an implementation does not follow the standard strictly. Later, Degabriele and Paterson [20] published another attack that works only if an implementation strictly follows the standard.

Chevalier et al. [21] extend the Dolev-Yao intruder with the capability to exploit Xor operator, as used in CBC, and they show that the protocol insecurity problem is NP-complete. Küsters and Truderung developed a verification method that can reduce the protocol models that are *Xor-linear* to Xor free models, which then can be analysed using existing tools [22]; however, the CBC based NSSK protocol is not Xor-linear due to the nested encryption.

In our view, the multiplicity error of DSSK protocol [23] is not a valid attack because it does not violate the claimed goals [7], namely neither confidentiality of the session key nor the entity authentication of participants is violated. Similarly, a reported type flaw [24] is based on a somewhat dubious assumption: if $\{T\} \equiv \{T, \{B, K_{AB}, T\}_{SA}\}$. Even if this assumption holds, the session key remains confidential and there is no violation of authentication.

In a slightly bigger picture, an impressive amount of research has been done for establishing a theoretically sound link between symbolic cryptography and complexity-theoretic cryptography [25–27]. In the line of universal composability, Ran Canetti and Herzog [28] show that the Dolev-Yao model can be layered on top of the traditional universal composability framework. Currently, this approach is limited to so-called simple protocols: the protocols that use only those cryptographic schemes that have some standard symbolic counterparts.

Another related line of work is on the security of online ciphers started by Bellare et al. [29]. In an online cipher, encryption of a plaintext block only depends on the current block and the previous blocks of the plaintext. Note that the requirements of a cipher are more stringent than an encryption scheme, because one is not allowed to use random initializing vectors (iv) in the construction of a cipher. Without a random iv, CBC and CFB modes are the candidates of online cipher, for which Fouque et al. [10] show that the CFB mode is provably secure and the CBC mode is not secure. The CBC mode is provably secure with a randomly chosen iv [9].

The attacks presented in this paper are based on the actual construction of CBC and CFB modes (using random IVs), but we still use symbolic abstraction to model the underlying cipher. We believe this level of abstraction is a good compromise between computational cryptography (where a cipher is modelled as a pseudorandom permutation) and symbolic cryptography (where the whole encryption scheme is modelled as a perfect cipher). At this abstraction level, which probably has not been explored in the prior art, we present a few symbolic encryption schemes.

3 NSSK Protocol

The NSSK protocol [6] is a key establishment protocol, based on symmetric encryption and the notion of a trusted third-party (TTP). In this paper, we assume that when a session expires then the session key is safely discarded, because this assumption prohibits the previously known flaw [7] resulting in a "secure" NSSK protocol. The protocol narrations are listed in the following.

$$
\begin{aligned}
&(1)\ A \longrightarrow S : A, B, N_A \\
&(2)\ S \longrightarrow A : \{N_A, B, K_{AB}, \{K_{AB}, A\}_{SB}\}_{SA} \\
&(3)\ A \longrightarrow B : \{K_{AB}, A\}_{SB} \\
&(4)\ B \longrightarrow A : \{N_B\}_{AB} \\
&(5)\ A \longrightarrow B : \{N_B - 1\}_{AB}
\end{aligned}
$$

Here A and B represent the initiator and responder roles that parties can take during an execution of the protocol; S is the role of a trusted third-party (TTP). It is assumed that S knows the identities of all legitimate entities (principals), and shares a long-term secret key with each of them, namely, S shares K_{SA} and K_{SB} with A and B respectively. The term K_{AB} denotes a session key. The notation $\{\dots\}_{AB}$ stands for a ciphertext computed using a key K_{AB}.

The first message is a request from A to the TTP that A wishes to establish a key with B, by sending its identity, the identity of the peer entity and a nonce. On receiving the request the TTP generates a random session key K_{AB}. The TTP replies with a message encrypted with A's long-term key, K_{SA}. This message includes a session key K_{AB}, and another encrypted message containing the same session key but encrypted with B's long term key, which A sends to B in the next step.

When B receives the message, it decrypts it using K_{SB}, then verifies that it contains B's identity, and if successful, then B considers K_{AB} as a valid session key. To verify the freshness of the session key, B sends a nonce, N_B, to A encrypted using the session key. On receiving the message in Step 4, A decrypts it and sends $N_B - 1$ to B encrypted using the same session key. This completes the protocol. If both parties terminate without generating any error then A and B assume that K_{AB} is a valid session key for the subsequent communication.

As per the standard cryptographic assumption, the initializing vectors (iv) in CBC and CFB modes are public values. We assume that the attacker is an insider, i.e., $\mathcal{I}$ is a legitimate network entity and shared $K_{S\mathcal{I}}$ with the TTP. An attacker $\mathcal{I}$ in the role of A is denoted by $\mathcal{I}(A)$.

For the simplicity of exposition, we assume that each term of the protocol is encoded in a separate block, e.g., the implementation of $\{N_1, N_2\}_{AB}$ using CBC mode of encryption results in the following ciphertext: $iv, c_1 = \{N_1 \oplus iv\}_{AB}, \{N_2 \oplus c_1\}_{AB}$. If blocks are not encoded with this perfect alignment then less efficient versions of the reported attacks may exist that require more computation and communication on the part of the adversary[2]. Nevertheless, the cryptographic security guarantees [9, 10] are valid independent of the block alignments in a plaintext.

In the following we describe the attacks against CBC and CFB based implementations. These attacks are also applicable on the seven-round version of the NSSK protocol [8], which does not suffer from the old-session-key attack [7].

NSSK with CBC Mode of Encryption

The attack is shown in Fig. 1, which consists of three setup phases followed by the main attack phase. The superscripts in iv^a, iv^b and iv^c are labels used to easily distinguish between initialization vectors in Setup-(a), Setup-(b) and Setup-(c) respectively; a subscript, such as '1' in iv_1, is used to distinguish different values of initialization vectors. The notation '=' is used to introduce intermediate terms to simplify the description of the attack.

In Setup-(a), $\mathcal{I}$ obtains the term $\{iv_2^a \oplus K_1\}_{SB}$, which he sends as a nonce in Setup-(b) to obtain c_1^b. In Setup-(c), $\mathcal{I}$ obtains the term c_1^c by sending K_1 as a nonce; K_1 can be computed by $\mathcal{I}$ in Setup-(a).

[2] For instance, in an attack on IPsec [5] that is based on address rewriting, the first phase of the attack succeeds with a probability of 2^{-17}, due to a specific block alignment of IPsec. This means that an attacker may have to repeat the first phase 2^{17} times in order to succeed.

	Messages
	Setup-(a)
(1)	$\mathcal{I} \longrightarrow S : \mathcal{I}, B, N_{\mathcal{I}}$
(2)	$S \longrightarrow \mathcal{I} : iv_1^a, iv_2^a, c_1^a = \{iv_1^a \oplus N_{\mathcal{I}}\}_{S\mathcal{I}}, c_2^a = \{c_1^a \oplus B\}_{S\mathcal{I}}, c_3^a = \{c_2^a \oplus K_1\}_{S\mathcal{I}}, c_4^a = \{c_3^a \oplus \{iv_2^a \oplus K_1\}_{SB}\}_{S\mathcal{I}}, c_5^a = \{c_4^a \oplus \{\{iv_2^a \oplus K_1\}_{SB} \oplus \mathcal{I}\}_{SB}\}_{S\mathcal{I}}$
	Setup-(b)
(1)	$\mathcal{I}(A) \longrightarrow S : A, B, \{iv_2^a \oplus K_1\}_{SB}$
(2)	$S \longrightarrow \mathcal{I}(A) : iv_1^b, iv_2^b, c_1^b = \{iv_1^b \oplus \{iv_2^a \oplus K_1\}_{SB}\}_{SA}, c_2^b = \{c_1^b \oplus B\}_{SA}, c_3^b = \{c_2^b \oplus K_2\}_{SA}, c_4^b = \{c_3^b \oplus \{iv_2^b \oplus K_2\}_{SB}\}_{SA}, c_5^b = \{c_4^b \oplus \{\{iv_2^b \oplus K_2\}_{SB} \oplus A\}_{SB}\}_{SA}$
	Setup-(c)
(1)	$\mathcal{I}(A) \longrightarrow S : A, B, K_1$
(2)	$S \longrightarrow A : iv_1^c, iv_2^c, c_1^c = \{iv_1^c \oplus K_1\}_{SA}, c_2^c = \{c_1^c \oplus B\}_{SA}, c_3^c = \{c_2^c \oplus K_3\}_{SA}, c_4^c = \{c_3^c \oplus \{iv_2^c \oplus K_3\}_{SB}\}_{SA}, c_5^c = \{c_4^c \oplus \{\{iv_2^c \oplus K_3\}_{SB} \oplus A\}_{SB}\}_{SA}$
	Attack
(1)	$A \longrightarrow S : A, B, N_A$
(2a)	$S \longrightarrow \mathcal{I}(A) : iv_1, iv_2, c_1 = \{iv_1 \oplus N_A\}_{SA}, c_2 = \{c_1 \oplus B\}_{SA}, c_3 = \{c_2 \oplus K_4\}_{SA}, c_4 = \{c_3 \oplus \{iv_2 \oplus K_4\}_{SB}\}_{SA}, c_5 = \{c_4 \oplus \{\{iv_2 \oplus K_4\}_{SB} \oplus A\}_{SB}\}_{SA}$
(2b)	$\mathcal{I}(S) \longrightarrow A : iv_1, iv_2, c_1 = \{iv_1 \oplus N_A\}_{SA}, c_2 = \{c_1 \oplus B\}_{SA}, c_1^c = \{iv_1^c \oplus K_1\}_{SA}, c_1^b = \{iv_1^b \oplus \{iv_2^a \oplus K_1\}_{SB}\}_{SA}, c_1^b = \{iv_1^b \oplus \{iv_2^a \oplus K_1\}_{SB}\}_{SA}$
(3a)	$A \longrightarrow \mathcal{I}(B) : c_1^c \oplus iv_1^b \oplus \{iv_2^a \oplus K_1\}_{SB}, c_1^b \oplus iv_1^b \oplus \{iv_2^a \oplus K_1\}_{SB}$
(3b)	$\mathcal{I}(A' = c_6 \oplus iv_2^a \oplus K_1) \longrightarrow B : iv_2^a, c_6 = \{iv_2^a \oplus K_1\}_{SB}, \{iv_2^a \oplus K_1\}_{SB}$
(4a)	$B \longrightarrow \mathcal{I}(A') : \{N_B\}_{K_1}$
(4b)	$\mathcal{I}(B) \longrightarrow A : \{N_B\}_{c_2 \oplus iv_1^c \oplus K_1}$
(5a)	$A \longrightarrow \mathcal{I}(B) : \{N_B - 1\}_{c_2 \oplus iv_1^c \oplus K_1}$
(5b)	$\mathcal{I}(A') \longrightarrow B : \{N_B - 1\}_{K_1}$

Fig. 1. Attack on CBC-version of NSSK Protocol

	Messages
	Attack
(1)	$A \longrightarrow S : A, B, N_A$
(2a)	$S \longrightarrow \mathcal{I}(A) : iv_1, iv_2, c_1 = \{iv_1\}_{SA} \oplus N_A, c_2 = \{c_1\}_{SA} \oplus B, c_3 = \{c_2\}_{SA} \oplus K_4, c_4 = \{c_3\}_{SA} \oplus \{iv_2\}_{SB} \oplus K_4, c_5 = \{c_4\}_{SA} \oplus \{\{iv_2\}_{SB} \oplus K_4\}_{SB} \oplus A$
(2b)	$\mathcal{I}(S) \longrightarrow A : iv_1, iv_2, c_1 = \{iv_1\}_{SA} \oplus N_A, c_2 = \{c_1\}_{SA} \oplus B, R_1, c_2 = \{c_1\}_{SA} \oplus B, R_2$
(3)	$A \longrightarrow \mathcal{I}(B) : iv_2, \{R_1\}_{SA} \oplus c_2, \{c_2\}_{SA} \oplus R_2$
(4)	$\mathcal{I}(B) \longrightarrow A : iv_4, \{iv_4\}_{K_1'} \oplus N_{\mathcal{I}}$, where $K_1' = \{c_2\}_{SA} \oplus R_1$
(5)	$A \longrightarrow \mathcal{I}(B) : iv_5, \{iv_5\}_{K_1'} \oplus (N_{\mathcal{I}} - 1)$

Fig. 2. Attack on CFB-version of NSSK Protocol

In the main phase of the attack, $\mathcal{I}$ replays the two terms, c_1^b and c_1^c, in place of c_3 and c_4 in the step (2b). This completes the attack on A, i.e., $\mathcal{I}$ can know impersonate as B to A with a known session key $c_2 \oplus iv_1^c \oplus K_1$.

The attack can be further extended to B if $c_6 \oplus iv_2^a \oplus K_1$ represents some valid identity. If this is the case then, in the step (3b), $\mathcal{I}$ replays $iv_2^a, \{iv_2^a \oplus K_1\}_{SB}$, which he obtains in Setup-(a). In this way B believes in K_1 as a new session key shared with a party whose identity is $c_6 \oplus iv_2^a \oplus K_1$. At this stage, $\mathcal{I}$ has successfully deceived both A and B into accepting the session keys that he knows. There are two different session keys, namely, A's session key is $c_2 \oplus iv_1^c \oplus K_1$ and B's session key is K_1. Now, $\mathcal{I}$ can play a man-in-middle role in any subsequent communication.

NSSK with CFB Mode of Encryption

In the attack on CFB version of the protocol, $\mathcal{I}$ is able to impersonate as B to A. In the step (2a), an adversary $\mathcal{I}$ intercepts the server's reply to A. The adversary replaces the terms c_3, c_4, and c_5 of the step (2a) with R_1, c_2, and R_2 and sends the resultant message to A in the step (2b); here R_1 and R_2 are any adversary's generated values.

When A receives the message in the step (2b), it decrypts R_1 to obtain a session key, which results in the value $K_1' = \{c_2\}_{SA} \oplus R_1$. Although the term $\{c_2\}_{SA}$ is not known to $\mathcal{I}$ at this stage, $\mathcal{I}$ can obtain $\{c_2\}_{SA}$ in the step 3, in which the term $\{c_2\}_{SA} \oplus R_2$ occurs in its second half. As R_2 is known to $\mathcal{I}$, the session key K_1' can be derived.

4 DSSK Protocol

Denning and Sacco [7] improves the NSSK protocol using time-stamps. The modified protocol is as follows:

$$
\begin{array}{l}
(1)\ A \longrightarrow S : A, B \\
(2)\ S \longrightarrow A : \{B, K_{AB}, T, \{A, K_{AB}, T\}_{SB}\}_{SA} \\
(3)\ A \longrightarrow B : \{A, K_{AB}, T\}_{SB}
\end{array}
$$

The protocols works essentially in the same way as the NSSK protocol. The new term T represents a time-stamp, and it is assumed that the local clocks of all network parties are loosely synchronized.

DSSK with CBC Mode of Encryption

This attack is listed in Fig 3, in which the adversary succeeds in impersonating B to A, i.e, at the end of the attack $\mathcal{I}$ in the role of B has a shared key with A. In the setup phase, $\mathcal{I}$ sends a request to S for establishing a connection with A, and as a result, $\mathcal{I}$ receives the terms $\bar{c}_5^a$ and $\bar{c}_6^a$, which are later used in the main phase of the attack.

In the main phase of the attack, $\mathcal{I}$ intercepts the reply from S and replace c_2 and c_3 with $\bar{c}_5^a$ and $\bar{c}_6^a$ respectively. Consequently, the last three messages will decrypt to some random data when A later sends them to B, however, $\mathcal{I}$ can pretend to be B. The session key for A and $\mathcal{I}(B)$ is $K_1 \oplus \bar{c}_4^a \oplus c_1$. Clearly, this term is computable by $\mathcal{I}$ because K_1, $\bar{c}_4^a$ and c_1 are known to $\mathcal{I}$.

The term $\bar{c}_6^a$ is decrypted to T_1. The Setup phase of the attack needs to be in real-time (in a loose sense) so that the difference between T_1 and T_2 is tolerable. As per the authors of the protocol, the definition of real-time is quite relaxed, namely a delay up to $\triangle t_1 + \triangle t_2$ is tolerable, where $\triangle t_1$ is the interval representing normal time-shift between A's local clock and the server clock, and $\triangle t_2$ is the expected network delay. This value is typically equal to a few seconds for most of the networks, such as the Internet.

	Messages
	Setup
(1)	$\mathcal{I} \longrightarrow S : \mathcal{I}, A$
(2)	$S \longrightarrow \mathcal{I} : iv_1^a, iv_2^a, c_1^a = \{A \oplus iv_1^a\}_{S\mathcal{I}}, c_2^a = \{K_1 \oplus c_1^a\}_{S\mathcal{I}}, c_3^a = \{T_1 \oplus c_2^a\}_{S\mathcal{I}}, c_4^a = \{(\bar{c}_4^a = \{\mathcal{I} \oplus iv_2^a\}_{SA}) \oplus c_3^a\}_{S\mathcal{I}}, c_5^a = \{(\bar{c}_5^a = \{K_1 \oplus \bar{c}_4^a\}_{SA}) \oplus c_4^a\}_{S\mathcal{I}}, c_6^a = \{(\bar{c}_6^a = \{T_1 \oplus \bar{c}_5^a\}_{SA}) \oplus c_5^a\}_{S\mathcal{I}}$
	Attack
(1)	$A \longrightarrow S : A, B$
(2a)	$S \longrightarrow \mathcal{I}(A) : iv_1, iv_2, c_1 = \{B \oplus iv_1\}_{SA}, c_2 = \{K_2 \oplus c_1\}_{SA}, c_3 = \{T_2 \oplus c_2\}_{SA}, c_4 = \{\{A \oplus iv_2\}_{SB} \oplus c_3\}_{SA}, c_5 = \{\{K_2 \oplus \{A \oplus iv_2\}_{SB}\}_{SB} \oplus c_4\}_{SA}, c_6 = \{\{T_2 \oplus \{K_2 \oplus \{A \oplus iv_2\}_{SB}\}_{SB} \oplus c_5\}_{SA}$
(2b)	$\mathcal{I}(S) \longrightarrow A : iv_1, iv_2, c_1 = \{B \oplus iv_1\}_{SA}, \bar{c}_5^a, \bar{c}_6^a, c_4, c_5, c_6$
(3)	$A \longrightarrow \mathcal{I}(B)$: random data

Fig. 3. Attack on CBC-version of DSSK Protocol

	Messages
	Setup
(1)	$\mathcal{I} \longrightarrow S : \mathcal{I}, A$
(2)	$S \longrightarrow \mathcal{I} : iv_1^a, iv_2^a, c_1^a = \{iv_1^a\}_{S\mathcal{I}} \oplus A, c_2^a = \{c_1^a\}_{S\mathcal{I}} \oplus K_1, c_3^a = \{c_2^a\}_{S\mathcal{I}} \oplus T_1, c_4^a = \{c_3^a\}_{S\mathcal{I}} \oplus (\bar{c}_4^a = \{iv_2^a\}_{SA} \oplus \mathcal{I}), c_5^a = \{c_4^a\}_{S\mathcal{I}} \oplus (\bar{c}_5^a = \{\bar{c}_4^a\}_{SA} \oplus K_1), c_6^a = \{c_5^a\}_{S\mathcal{I}} \oplus (\bar{c}_6^a = \{\bar{c}_5^a\}_{SA} \oplus T_1)$
	Attack
(1)	$A \longrightarrow S : A, B$
(2a)	$S \longrightarrow \mathcal{I}(A) : iv_1, iv_2, c_1 = \{iv_1\}_{SA} \oplus B, c_2 = \{c_1\}_{SA} \oplus K_2, c_3 = \{c_2\}_{SA} \oplus T_2, c_4 = \{c_3\}_{SA} \oplus \{iv_2\}_{SB} \oplus A, c_5 = \{c_4\}_{SA} \oplus \{\{iv_2\}_{SB} \oplus A\}_{SB} \oplus K_2, c_6 = \{c_5\}_{SA} \oplus \{\{\{iv_2\}_{SB} \oplus A\}_{SB} \oplus K_2\}_{SB} \oplus T_2$
(2b)	$\mathcal{I}(S) \longrightarrow A : iv_1, iv_2, c_1 = \{iv_1\}_{SA} \oplus B, \bar{c}_5^a, \bar{c}_6^a, c_1, R_1, R_2$
(3)	$A \longrightarrow \mathcal{I}(B) : iv_2, \{\bar{c}_6^a\}_{SA} \oplus c_1, \{c_1\}_{SA} \oplus R_1, \{R_1\}_{SA} \oplus R_2$

Fig. 4. Attack on CFB-version of DSSK Protocol

DSSK with CFB Mode of Encryption

This attack is similar to the attack on the CBC version. The adversary $\mathcal{I}$ obtains the terms $\bar{c}_5^a$ and $\bar{c}_6^a$ in the setup phase. In the main phase, $\mathcal{I}$ intercepts the reply from the server in the step (2a) and replaces the terms c_2, c_3, c_4, c_5, and c_6 with $\bar{c}_5^a$, $\bar{c}_6^a$, c_1, R_1, and R_2 respectively; here, R_1 and R_2 are any values known to $\mathcal{I}$.

When A receives the modified message in the step (2b), the session key is computed to be $\{c_1\}_{SA} \oplus \bar{c}_5^a$. The term $\{c_1\}_{SA}$ is not known to $\mathcal{I}$; that is why c_4 and c_5 of the step (2a) were replaced by c_1 and R_1. The decryption of c_1 and R_1 results in $\{\bar{c}_6^a\}_{SA} \oplus c_1$ and $\{c_1\}_{SA} \oplus R_1$, which A sends supposedly to B in the step (3). In this way, $\mathcal{I}$ can derive the term $\{c_1\}_{SA}$. The decryption of $\bar{c}_6^a$ results in T_1, which is a valid time-stamp based on the same arguments presented for the CBC version.

5 Private-Key Symbolic Encryption Schemes

The reported attacks cannot be produced in a security model in which encryption is specified as one monolithic ciphertext, which hides the structure of a ciphertext. On the other hand, a ciphertext resulting from a block-cipher based encryption scheme always has a semantic structure.

We propose that symbolic encryption should be specified using the abstraction of a block-cipher, because the output of a block cipher can be safely assumed as

a monolithic ciphertext. Further, a block cipher is the most natural abstraction of actual implementation and can be instantiated, e.g., with an appropriate algorithm from Advance Encryption Standard (AES).

In a formal security model that supports the Xor operator, it is quite straight forward to specify commonly used cryptographic encryption schemes, such as CBC and CFB. In our proposal, however, we do not assume that the support for the Xor operator is available. The motivation of this exclusion is contemporary and based on the observation that properly incorporating the Xor operator in formal security models is a long-standing open problem [21, 22]. The Xor operator is not supported by most of the verification tools, e.g., OFMC [11], LySa [12], and Spi-calculus [30]. On the other hand, the proposed Xor-free encryption schemes can be seamlessly used in existing formal security models.

Our assumptions are as follows. We consider a block-cipher as a family of pseudo random permutations (PRP) [31]. We consider three types of adversaries that are computationally bounded in a sense that their attacks strategies terminate in a polynomial time. The three types are passive adversary, CPA-adversary (i.e, an adversary who can access an encryption oracle), and CPA/CCA-adversary (i.e., an adversary who can access both encryption and decryption oracles.)

A few notations used in the following sections are as follows. An *overline* on a variable name, such as $\overline{M}_1$, indicates that the variable is on binary strings. We use the the notation $\mathcal{U}_s$ to represent the uniform distribution on strings of size s. The notation $\overline{U}_s$ is used to represent a random variable on the uniform distribution: $\overline{U}_s \leftarrow \mathcal{U}_s$. The concatenation of random variables $\overline{U}_s^1, \ldots, \overline{U}_s^n$ is just another random variable $\overline{U}_{ns}$, where $\overline{U}_{ns} \leftarrow \mathcal{U}_{ns}$. The notation $dist[.]$ represents the probability distribution of its argument, e.g., $dist[\overline{U}_s^1]$ is $\mathcal{U}_s$.
In the following, we define a minimal symbolic encryption system.

Definition 1 (Symbolic Encryption System). *On the set of all base terms $\mathcal{V}$, with a security parameter $s = log_2(|\mathcal{V}|)$, we define a private-key symbolic encryption system as follows.*

- $\mathtt{M} ::= \mathtt{M}, \mathtt{M} \mid \mathtt{V} \mid \{\mathtt{M}\}_\mathtt{K} \mid \{\mathtt{C}\}_\mathtt{K}^{-1}$
- $\mathtt{V} ::= x \in \mathcal{V}$
- $\mathtt{K} ::= \mathtt{M}$ *(Syntactic sugar to indicate that the term K is being used as a key)*
- $\mathtt{C} ::= \{\mathtt{M}\}_K$ *(Syntactic sugar to indicate that the term is a ciphertext)*
- *Cancellation Rule :* $M = \{\{M\}_K\}_K^{-1} = \{\{M\}_K^{-1}\}_K$
- *Encryption Rule : Given K and M, $\{M\}_K$ can be derived.*
- *Decryption Rule : Given K and C, $\{C\}_K^{-1}$ can be derived.*

Here $\mathtt{M}$, $\mathtt{K}$, $\mathtt{C}$ and $\mathtt{V}$ are the formal expression; while M, K and C are the corresponding meta-variables.

To define the semantics of the symbolic encryption system, we use the notion of variadic ciphers [32], which can take binary strings of different lengths as inputs. The reason for employing a variadic cipher is based on the fact that M in Def. 1 consists of a variable number of base terms. To model the arbitrary size of a key

(K in Def. 1), we extend the notion of a variadic cipher to an idealized variadic cipher (IVC), namely a family of functions that contains an infinite number of variadic functions. We represent the i-th idealized variadic cipher as $\Pi_i(.)$.

The notion of an IVC can be compared to a traditional cipher, which is modelled as a family containing a fixed number of PRPs (pseudo random permutations) and all of the PRPs are of the same size, e.g., AES with 256 bit key is a family containing 2^{256} PRPs of size 128 bit. Note, however, that the use of IVCs is just for a simpler exposition. Later, we define our symbolic schemes in such a way that only the restricted forms of IVCs occur that can be instantiated with traditional ciphers.

Definition 2 (Cryptographic Semantics). *The cryptographic semantics of the symbolic encryption system in Def. 1 are as follows.*

- $V \stackrel{def}{=} \overline{V} \in \{0,1\}^s$ *(Each base term is encoded as a bit string of a fixed length s, such that $s = log_2(|\mathcal{V}|)$)*
- $M_1, M_2 \stackrel{def}{=} \overline{M_1}, \overline{M_2}$ *(Concatenation of two bit strings)*
- $\{M\}_K \stackrel{def}{=} \overline{\{M\}_K} = \Pi_{\overline{K}}(\overline{M})$,
  where $\Pi_{\overline{K}}(\overline{M})$ is the $\overline{K}$th cipher in a family of IVCs.
- $\{C\}_K^{-1} \stackrel{def}{=} \overline{\{C\}_K^{-1}}$, *such that* $\overline{C} = \Pi_{\overline{K}}(\overline{\{C\}_K^{-1}})$.

Definition 3 (Security). *We define the following three security properties, assuming that K is not known to the adversary.*

WP-security *(Weak Privacy Against Passive Attack)* $\stackrel{def}{=}$ *It is infeasible for a passive adversary $\mathcal{I}$ to compute $\overline{C}$ for a known $\overline{M}$, s.t., $C = \{M\}_K$. Further, it is also infeasible for $\mathcal{I}$ to compute $\overline{M}$ for a known $\overline{C}$, s.t., $M = \{C\}_K^{-1}$.*

NM-security *(Non-malleability Against CPA/CCA)* $\stackrel{def}{=}$ *It is infeasible for a CPA/CCA-adversary to compute $\overline{C'}$ for a known $\overline{C}$, s.t., a pre-specified relation $\mathcal{R}(\overline{M}, \overline{M}')$ holds[3], where $M = \{C\}_K^{-1}$ and $M' = \{C'\}_K^{-1}$.*

IND-security *(Indistinguishability Against CPA)* $\stackrel{def}{=}$ *It is infeasible for a CPA-adversary to distinguish between the probability distributions $dist[\overline{\{M\}_K}]$ and $\mathcal{U}_l$, for all values of M, where $l = |\overline{\{M\}_K}|$.* [4]

Clearly, WP-security is implied by IND-security, because if an adversary can recover the plaintext from a ciphertext then he can always distinguish between the ciphertext and a random bit string. In our proofs, we also use the fact that if an encryption function is deterministic then it cannot be IND-secure [13]. Note that IND-security and NM-security are not comparable in our model, because we use the abstraction of a cipher, which is a deterministic encryption algorithm for a fixed key. As shown by Katz and Yung [2], for probabilistic encryption, there are well-defined relations between NM-security and IND-security under different attack models.

[3] E.g., $\overline{M} = \overline{M}_1, \overline{M}_2$ and $\overline{M}' = \overline{M}_2, \overline{M}_1$

[4] Equivalently, $\mathcal{I}$ can only succeeds in the indistinguishability experiment (IND-P2-C0) [2] with a negligible probability.

Claim 1 (Soundness of Symbolic Encryption System). *The symbolic encryption function $\{M\}_K$ is WP,NM-secure if a CPA/CCA-adversary cannot derive K.*

Proof (Sketch). As per the semantics, $\{M\}_K$ is a PRP corresponding to $\Pi_{\overline{K}}(\overline{M})$. For a secret K, mapping from $\overline{M}$ to $\Pi_{\overline{K}}(\overline{M})$ is secret, which implies weak privacy for a polynomial-time adversary. The mapping from an input to the output is random, which implies non-malleability. The formal proof is trivial (but tedious) and is left out. □

In the following, we introduce four symbolic encryption schemes. The direct implementations of these schemes, as per the semantics, assume the existence of one of the two ciphers corresponding to the key size s and $2s$, e.g., AES-128 and AES-256.

Definition 4 (Symbolic Encryption Schemes). *Let M and K be the two variable of symbolic encryption system with semantics $\overline{M}$ and $\overline{K}$, such that $|\overline{M}| \leq s^c$ and $|\overline{K}| = s$, for a constant c. Let $\overline{M}_1, \ldots, \overline{M}_N$ be the parsing of $\overline{M}$, such that $|\overline{M_i}| = s$, for $1 \leq i \leq N$. ECB symbolic encryption (ECB-SE), bulk symbolic encryption (BLK-SE), randomized symbolic encryption (RND-SE), and randomized-bulk symbolic encryption (RNB-SE), are defined by the ECB-rule, BLK-rule, RND-rule, and RNB-rule respectively.*

$$\textit{ECB-rule: } \frac{\{M\}_K}{\{M_1\}_K, ..., \{M_i\}_K, ..., \{M_N\}_K}$$

$$\textit{BLK-rule: } \frac{\{M\}_K}{\{M_1, ..., M_i, ..., M_N\}_K}$$

$$\textit{RND-rule: } \frac{\{M\}_K}{V_1, \{M_1\}_{K,V_1}, ..., V_i, \{M_i\}_{K,V_i}, ..., V_N, \{M_N\}_{K,V_N}}$$

$$\textit{RNB-rule: } \frac{\{M\}_K}{V_1, \{M_1, ..., M_i, ..., M_N\}_{K,V_1}}$$

In the above definition, s^c stands for a polynomial in s; without a polynomial length restriction, none of the existing cryptographic encryption schemes is secure. In the above rules, the base terms $V_1, ..., V_N$ appear as free variables, therefore these variables are assumed to be instantiated with unique values in each instance of a protocol. Also note that, e.g., the key used to create ciphertext $\{M_1, ..., M_i, ..., M_N\}_{K,V_1}$ is K, V_1, which semantically corresponds to the concatenation of $\overline{K}$ and $\overline{V_1}$. The main motivation for the above division is their correspondence to some of the existing cryptographic schemes, as described in the following sections.

6 Security Analysis

In the following, we analyse these schemes for the security properties in Def. 3.

Claim 2. *The ECB-SE is WP-secure, but it is neither NM-secure nor IND-secure.*

Proof. Each encrypted term $\{M_i\}_K$ in the ECB scheme is a PRP and is WP,NM-secure (Claim 1). From WP,NM-security of the terms, we derive the security properties of the whole scheme.

It is clear that the WP-security of the scheme can be reduced to the WP-security of its terms, because if a passive adversary can recover the plaintext $\overline{M_1}, \ldots, \overline{M_N}$ then he can invert a PRP on N different values. The same observation holds for deriving ciphertexts from plaintexts.

The scheme is not IND-secure because it is a deterministic function [13]. The ECB scheme is not NM-secure due to a simple attack. In the attack, an adversary permutes the individual encrypted terms. For example, given a ciphertext $\{M_1\}_K, \{M_2\}_K$, the adversary can produce another valid ciphertext $\{M_2\}_K, \{M_1\}_K$ that has a related plaintext to the plaintext of the first ciphertext. This completes the proof. □

Claim 3. *The output distribution of RND-SE is* $\mathcal{U}_{Ns}$.

Proof. The size of each i-th term in a ciphertext of RND-SE is $|\overline{\{M_i\}_{K,V_i}}| = s$, as per Def. 4. Therefore, the number of plausible ciphertexts for the i-th term is 2^s. In each term, V_i is used as part of the key. Being V_i a free variable, each application of RND-rule uses a new value. With a secret $\overline{K}$, there are 2^s equally probable values for the key $\overline{K}, \overline{V}_i$, in every application of the RNB-rule. Consequently, the output of $\Pi_{\overline{K},\overline{V}_i}(\overline{M})$ is evenly distributed on 2^s plausible ciphertexts for a known value of $\overline{M}$.

Therefore, we have $\overline{U_s^i} = \overline{\{M_i\}_{K,V_i}}$ where $dist[\overline{U_s^i}] = \mathcal{U}_s$, for $1 \leq i \leq N$. The distribution of complete ciphertext is $dist[\overline{U_s^1}, \ldots, \overline{U_s^N}] = \mathcal{U}_{Ns}$. □

Claim 4. *The RND-SE is WP,IND-secure, but it is not NM-secure.*

Proof. The RND-SE is clearly not NM-secure, because the same permutation attack of Claim 2 also works for the RND-SE. Since WP-security is implied by IND-security, we only need to prove that RND-SE is IND-secure. For IND-security, a CPA-adversary cannot distinguish between a ciphertext corresponding to the adversary's plaintext and a random string $\overline{U_{Ns}}$, where $\overline{U_{Ns}} \leftarrow \mathcal{U}_{Ns}$. From Claim 3, the output distribution of RND-SE is $\mathcal{U}_{Ns}$, which is same as that of the random bit string. Hence, RND-SE is WP,IND-secure but is not NM-secure. □

Claim 5. *The BLK-SE is WP,NM-secure, but it is not IND-secure.*

Proof. This scheme represents one variadic PRP and is therefore WP,NM-secure (Claim 1). The scheme is deterministic, therefore it cannot be IND-secure [13]. □

Claim 6. *The scheme RNB-SE is WP,IND,NM-secure.*

The ciphertext of the scheme is also a single variadic PRP and is therefore WP,NM-secure. To show that it is IND-secure, similar to Claim 4, the probability distribution of a ciphertext corresponding to a known plaintext must be computationally indistinguishable from a random bit string for a CPA-adversary.

The size of the ciphertext in RNB-SE is Ns, therefore the domain corresponding to plausible ciphertexts is of size 2^{Ns}. Since the term $\overline{V}$ is assigned a new value on each application of RNB-rule, there are 2^s uniformly distributed values for the key $\overline{K}, \overline{V}$. Consequently, there are 2^s uniformly distributed values for the ciphertext.

To violate IND-security, a CPA-adversary is required to distinguish between the following two distributions: the uniform distribution on 2^s strings each of size Ns (corresponding to the ciphertext); and the uniform distribution $\mathcal{U}_{Ns}$ (corresponding to a random string.) The most efficient known technique to distinguish between two uniform distributions is to compare the number of collisions in the lists of values drawn from the respective distributions.

From the birthday problem, we know that in a list of q ciphertexts computed from the same plaintext, an upper bound on the probability of any collision is $0.5q(q-1)2^{-s}$, and for random strings drawn from $\mathcal{U}_{Ns}$ a lower bound on the probability of any collision is $0.3q(q-1)2^{-Ns}$. Although the maximum difference between the probabilities of collisions, namely $0.5q(q-1)2^{-s} - 0.3q(q-1)2^{-Ns}$, is relatively large, but the difference is only noticeable if an adversary is able to generate at least one collision.

Since $0.5q(q-1)2^{-s} > 0.3q(q-1)2^{-Ns}$ and $0.5q(q-1)2^{-s}$ is negligible in s assuming q is polynomial in s, the probability of occurrence of a collision is negligible. For a polynomial-time adversary, q must be a polynomial in s. Therefore, we conclude that adversary cannot distinguish between the two distributions. Hence, RNB-SE is IND,NM,WP-secure. □

The results presented in this section are listed in Table 1. In all of the encryption schemes in Def. 4 the key size is fixed: it is s for ECB-SE and BLK-SE, and it is $2s$ for RND-SE and RNB-SE. Further, for ECB-SE and RND-SE, the block size is also fixed. This means that ECB-SE and RND-SE can be implemented with traditional block ciphers, and BLK-SE and RNB-SE schemes can be implemented with variadic ciphers [32]. Besides such semantic-oriented implementations, other cryptographic algorithms can be chosen for an implementation using the security properties (Claim 2-6) of each scheme.

We know that a cryptographic message authentication code (MAC) can be used to provide the non-malleability of a plaintext under encrypt-then-MAC method (i.e., MAC of ciphertext) [33]. Further note that CBC/CFB mode of encryption provide IND-security under CPA [9, 10].

Therefore, e.g., ECB-SE can be implemented using AES in ECB mode of encryption; RND-SE can be implemented using AES in CBC mode of encryption; BLK-SE can be implemented using AES in ECB mode along with a message authentication code (MAC); and RNB-SE can be implemented using AES in CBC mode along with a MAC. It is certainly possible (perhaps after extending the system of Def. 1) to define symbolic schemes that correspond to other forms of cryptographic encryption, such as the counter mode of encryption.

7 Discussion

In practice, it is nonetheless dangerous to assume that a system developer will actually discover and use the correct cryptographic scheme that meets the security requirements of a particular use of symbolic encryption. System developers often use an implementation instance that seems appropriate, e.g., in this paper, the CBC implementations of encryption in NSSK and DSKK protocols indeed guarantee privacy in a strong sense [13]; however, non-malleability of the ciphertexts, an implicit assumption, is also required for the security of these protocols.

One may always choose to employ a strong encryption scheme meeting the requirements of RNB-SE however, the cost associated with such an overly cautious approach cannot be ignored in practice. For example, if the symbolic model of a protocol that uses ECB-SE is secure then this means that the protocol can be implemented in a relatively efficient manner: a random number generator is not required; the algorithm for message authentication code (MAC, used to guarantee non-malleability) is not required; and communication bandwidth is reduced because we do not need to transmit initialization vectors and MAC codes. Moreover, parallelisation of the encryption process is straight forward.

In many applications, such optimizations can make a huge difference, e.g., for a hypervisor which has to process millions of requests per second, and a sensor node in which memory, computational power and energy are scarce resources. Many symbolic protocols remain secure when encryption requirements are met by a weaker symbolic encryption scheme, such as ECB-SE and CBC mode of encryption; in this way a level of safe optimization can be achieved.

It is important to remember that safely instantiating a symbolic encryption scheme with a cryptographic encryption scheme does not mean that the resultant protocol will be secure, because there are many attacks that do not rely on encryption, e.g., Lowe's attack [34] on public-key version of Needham-Schroeder protocol relies on the assumption of a corrupt insider, Denning-Sacco's attack [7] relies on the availability of a compromised old session key. Moreover, there are many security vulnerabilities that are outside the realm of (mathematical) cryptography, e.g., buffer-overflow.

Table 1. Summary of Results

Scheme	WP	IND	NM	Instantiation Examples
ECB-SE	√	×	×	AES-128 in ECB mode of encryption
BLK-SE	√	×	√	(1) AES-128 in ECB mode of encryption with SHA-256 as MAC (2) Variadic cipher
RND-SE	√	√	×	(1,2) AES-128 in CBC/CFB mode of encryption
RNB-SE	√	√	√	(1,2) AES-128 in CBC/CFB mode of encryption with SHA-256 as MAC (3) Variadic cipher with a randomized key

8 Conclusion

In this paper, we reported new attacks on reasonable implementations of well-known protocols. It appears that there is no inherent limitation in symbolic models which may have prevented detecting these attacks. We notice that encryption on multiple terms is traditionally specified as one big monolithic encrypted block, which, however, is not a good way of specifying it for practice-oriented security analysis. We presented four refined ways in which encryption can be specified in a symbolic model. The proposed specifications not only help to avoid many implementation vulnerabilities similar to the reported attacks, but they also provide a degree of safe optimization. We hope that our work will bring symbolic encryption closer to the secure implementation of encryption.

References

1. Dworkin, M.: Recommendation for block cipher modes of operation. methods and techniques. Technical report, DTIC Document (2001)
2. Katz, J., Yung, M.: Complete characterization of security notions for probabilistic private-key encryption. In: Theory of Computing. ACM (2000)
3. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Trans. on Information Theory (1983)
4. Bellovin, S.: Problem areas for the IP security protocols. In: USENIX UNIX Security Symp. (1996)
5. Paterson, K.G., Yau, A.K.L.: Cryptography in Theory and Practice: The Case of Encryption in IPsec. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 12–29. Springer, Heidelberg (2006)
6. Needham, R., Schroeder, M.: Using encryption for authentication in large networks of computers. Communications of the ACM 21(12), 993–999 (1978)
7. Denning, D., Sacco, G.: Timestamps in key distribution protocols. Communications of the ACM 24(8), 533–536 (1981)
8. Needham, R., Schroeder, M.: Authentication revisited. Operating Systems Review 21(1) (1987)
9. Goldwasser, S., Bellare, M.: Lecture notes on cryptography. Course "Cryptography and computer security" at MIT 1999 (1996) 1999
10. Fouque, P.-A., Martinet, G., Poupard, G.: Practical Symmetric On-Line Encryption. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 362–375. Springer, Heidelberg (2003)
11. Basin, D., Mödersheim, S., Vigano, L.: Ofmc: A symbolic model checker for security protocols. International Journal of Information Security 4(3), 181–208 (2005)
12. Bodei, C., Buchholtz, M., Degano, P., Nielson, F., Nielson, H.: Static validation of security protocols. Journal of Computer Security 13(3), 347–390 (2005)
13. Goldwasser, S., Micali, S.: Probabilistic encryption. J. of Computer and System Sciences 28(2) (1984)
14. Meadows, C.: Formal methods for cryptographic protocol analysis: Emerging issues and trends. Selected Areas in Communications 21(1), 44–54 (2003)
15. Moore, J.: Protocol failures in cryptosystems. Proc. of the IEEE 76(5), 594–602 (1988)
16. Boyd, C.: Hidden assumptions in cryptographic protocols. In: Proc. of Computers and Digital Techniques, vol. 137, pp. 433–436. IET (1990)

17. Mao, W., Boyd, C.: On the use of encryption in cryptographic protocols. In: Codes and Cyphers (1995)
18. Stubblebine, S., Meadows, C.: Formal characterization and automated analysis of known-pair and chosen-text attacks. IEEE J. on Selected Areas in Communications 18(4), 571–581 (2000)
19. Kremer, S., Ryan, M.: Analysing the vulnerability of protocols to produce known-pair and chosen-text attacks. Electronic Notes in Theoretical Computer Science 128(5), 87–104 (2005)
20. Degabriele, J., Paterson, K.: Attacking the IPsec standards in encryption-only configurations. In: IEEE Symp. S&P, pp. 335–349. IEEE (2007)
21. Chevalier, Y., Kusters, R., Rusinowitch, M., Turuani, M.: An NP decision procedure for protocol insecurity with XOR. In: Logic in CS. IEEE (2003)
22. Küsters, R., Truderung, T.: Reducing protocol analysis with xor to the xor-free case in the horn theory based approach. In: Proc. of CCS. ACM (2008)
23. Lowe, G.: A family of attacks upon authentication protocols. Technical Report 1997/5, University of Leicester (1997)
24. Chevalier, Y., Vigneron, L.: Automated Unbounded Verification of Security Protocols. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 125–171. Springer, Heidelberg (2002)
25. Abadi, M., Rogaway, P.: Reconciling two views of cryptography. In: TCS: Exploring New Frontiers of Theoretical Informatics, pp. 3–22 (2000)
26. Herzog, J.C., Liskov, M., Micali, S.: Plaintext Awareness via Key Registration. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 548–564. Springer, Heidelberg (2003)
27. Micciancio, D., Warinschi, B.: Soundness of Formal Encryption in the Presence of Active Adversaries. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 133–151. Springer, Heidelberg (2004)
28. Canetti, R., Herzog, J.: Universally composable symbolic security analysis. J. of Cryptology (2011)
29. Bellare, M., Boldyreva, A., Knudsen, L.R., Namprempre, C.: Online Ciphers and the Hash-CBC Construction. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 292–309. Springer, Heidelberg (2001)
30. Abadi, M., Gordon, A.D.: Reasoning About Cryptographic Protocols in the Spi Calculus. In: Mazurkiewicz, A., Winkowski, J. (eds.) CONCUR 1997. LNCS, vol. 1243, pp. 59–73. Springer, Heidelberg (1997)
31. Luby, M., Rackoff, C.: How to Construct Pseudo-random Permutations from Pseudo-random Functions. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 447–447. Springer, Heidelberg (1986)
32. Bellare, M., Rogaway, P.: On the Construction of Variable-Input-Length Ciphers. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 231–244. Springer, Heidelberg (1999)
33. Bellare, M., Namprempre, C.: Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000)
34. Lowe, G.: Breaking and fixing the Needham-Schroeder public-key protocol using FDR. Tools and Algos. for the Construction and Analysis of Systems (1996)