2 Cryptoverif

2.1 Exercise 1

We consider the public-key encryption scheme RSA-OAEP defined as follows:

\[ E(m, pk) = \]
\[ r \overset{R}{\leftarrow} \{0, 1\}^{k_0} \]
\[ X \leftarrow (m \parallel 0) \oplus G(r) \]
\[ Y \leftarrow H(X) \oplus r \]
\[ \text{return } f_{pk}(X \parallel Y) \]

where \( m \) is the message to encrypt, \( pk \) is the public key, \( r \) is chosen randomly in \( \{0, 1\}^{k_0} \), \( \oplus \) is exclusive or, \( \parallel \) is the concatenation of bitstrings, \( G \) and \( H \) are hash functions in the random oracle model, and \( f \) is a one-way trapdoor permutation (in practice, \( f \) is RSA).

In this scheme, the message \( m \) has \( n - k_0 - k_1 \) bits; the constant 0 contains \( k_1 \) bits; \( r, H(X), \) and \( Y \) have \( k_0 \) bits; \( G(r), m \parallel 0, \) and \( X \) have \( n - k_0 \) bits; \( f_{pk} \) is permutation on bitstrings of size \( n \). Integers \( k_0 \) and \( k_1 \) are large enough so that the probability that a uniformly distributed random bitstring of size \( k_0 \) (resp. \( k_1 \)) be equal to a fixed constant is small.

(1) What is the decryption function of this scheme? What can be checked to make sure that decryption succeeds?

(2) How does one specify in Cryptoverif that a random bitstring is chosen in a set large enough so that the probability of collision with a fixed constant is small?

(3) We would like to show that this scheme is IND-CPA. Which game should be given as input to Cryptoverif for this proof?

(4) Among the available cryptographic transformations (random oracle assumption on \( G \) and \( H \), replacing \( x \oplus r \) with \( r \) when \( r \) is a random value not used elsewhere, and one-wayness), which one(s) can be applied directly on the first game given in question (3)? Justify. Can some of these assumptions be applied after simple syntactic transformations? If yes, which one(s), after which transformation(s)? Can applying some of the cryptographic transformations help apply others? Explain why.

(5) How should the game of question (3) be modified in order to prove that the scheme is IND-CCA2?
2.2 Exercise 2

We consider the following protocol:

\[ A \rightarrow B : E_{pk_B}(pk_A, k, S_{sk_A}(k)) \]

In this protocol, \( A \) has a secret signature key \( sk_A \) and a corresponding public verification key \( pk_A \), and \( B \) has a public encryption key \( pk_B \) and a corresponding secret decryption key \( sk_B \). \( E \) denotes encryption and \( S \) denotes signature.

\( A \) chooses a fresh shared-key encryption key \( k \), signs it under \( sk_A \), and sends to \( B \) the encryption under \( pk_B \) of the triple containing \( A \)'s public key, the key \( k \), and the signature. \( B \) decrypts, obtains \( k \), and verifies the signature.

\( A \) is willing to run this protocol with \( B \), but also with dishonest participants that are supposed to be included in the adversary. Similarly, \( B \) is willing to run this protocol with \( A \), but also with dishonest participants.

The signature scheme is supposed to be UF-CMA (unforgeable under chosen-message attacks). The public-key encryption scheme is supposed to be IND-CCA2 (indistinguishable under chosen-ciphertext attacks). All keys are supposed to have the same length.

Our goal is to prove the secrecy of the key \( k \) in a successful run of the protocol between \( A \) and \( B \).

(1) Give the first game, which represents the protocol, in the CryptoVerif input language.

(2) Is the key \( k \) exchanged between \( A \) and \( B \) really secret? Justify. If not, give an attack and propose a fix to the protocol.