

Solution of Exercise Sheet 1

Out: Thu, Oct 22, 2009

Due: Tue, Nov 3, 2009, 10am

Problem 1: Designing a symbolic model (10 Points)

In the lecture we have seen various symbolic models for analyzing protocols involving group arithmetic. For each of these models we have seen three examples of protocols that are secure with respect to the Dolev-Yao model but insecure when implemented cryptographically (see Definition 7, Definition 8, and Definition 10 in the lecture notes).

- Design a symbolic model (i.e., a set of messages M and relations \approx, \vdash) in which all three protocols are insecure. (Of course, you should not do this by adding a rule that gives the adversary too much power such as the rule $\frac{S \vdash g(M)}{S \vdash M}$.)
- For each of the three protocols from the lecture notes, prove that it is insecure (in the sense of Lemma 1, Lemma 2, and Lemma 3 in the lecture notes).
- You can get extra points if you can find additional problems with the symbolic models presented in the lecture and solve these problems in your symbolic model.

Solution. The attack on the protocol from Definition 10 in the lecture notes is based on the fact that the adversary can do addition and multiplication. We therefore extend the symbolic model from Definition 9 in the lecture notes to contain addition and multiplication.

The set M of messages is given by the following grammar:

$$M ::= N | N^A | enc(M, M) | g(M) | f(M, M) | zero | one | add(M, M) | mul(M, M)$$

The relation \approx is the smallest equivalence relation satisfying:

$$\begin{array}{c}
\frac{M_1 \approx M_2 \quad C \text{ is a context}}{C[M_1] \approx C[M_2]} \quad f(g(M_1), M_2) \approx f(g(M_2), M_1) \\
f(x, \text{one}) \approx x \quad f(x, \text{zero}) \approx \text{one} \quad g(\text{zero}) \approx \text{one} \\
\text{add}(M_1, M_2) \approx \text{add}(M_2, M_1) \quad \text{add}(M_1, \text{zero}) \approx M_1 \\
\text{add}(M_1, \text{add}(M_2, M_3)) \approx \text{add}(\text{add}(M_1, M_2), M_3) \\
\text{mul}(M_1, M_2) \approx \text{mul}(M_2, M_1) \quad \text{mul}(M_1, \text{one}) \approx M_1 \\
\text{mul}(M_1, \text{mul}(M_2, M_3)) \approx \text{mul}(\text{mul}(M_1, M_2), M_3) \\
f(\text{mul}(M_1, M_2), M_3) \approx \text{mul}(f(M_1, M_3), f(M_2, M_3)) \\
f(M_1, \text{add}(M_2, M_3)) \approx \text{mul}(f(M_1, M_2), f(M_1, M_3)) \\
g(\text{add}(M_2, M_3)) \approx \text{mul}(g(M_2), g(M_3)) \\
f(g(M_1), M_2) \approx g(\text{mul}(M_1, M_2)) \\
f(f(M_1, M_2), M_3) \approx f(M_1, \text{mul}(M_2, M_3))
\end{array}$$

(I do not claim that these are all possible rules.)

The deduction relation we use is the smallest relation \vdash satisfying:

$$\begin{array}{c}
\frac{M \in S}{S \vdash M} \quad \frac{N \in N_A}{S \vdash N} \quad \frac{S \vdash M_1, \text{enc}(M_1, M_2)}{S \vdash M_2} \quad \frac{S \vdash M \quad M \approx M'}{S \vdash M'} \\
\frac{S \vdash \text{mul}(M_1, M_2) \quad S \vdash M_1}{S \vdash M_2} \text{DIV} \quad \frac{S \vdash f(M_1, M_2) \quad S \vdash M_2}{S \vdash M_2} \text{ROOT} \\
\frac{S \vdash M_1, M_2}{S \vdash \text{enc}(M_1, M_2), g(M_1), f(M_1, M_2), \text{zero}, \text{one}, \text{mul}(M_1, M_2), \text{add}(M_1, M_2)}
\end{array}$$

The rule DIV models the fact that the adversary can divide. The rule ROOT models the fact that the adversary can compute M_2 -th roots. (An alternative to these rules would be to add a constructor *inv* for inverting group elements and exponents.)

Insecurity of the protocol from Definition 7 in the lecture notes. Let $S := \{X, f(g(X), Y), \text{enc}(g(Y), N)\}$. We have to show that $S \vdash N$. (Compare to Lemma 1 in the lecture notes.)

This is shown using the following derivation:

$$\frac{\frac{\frac{f(g(X), Y) \in S}{S \vdash f(g(X), Y)} \quad f(g(X), Y) \approx f(g(Y), X)}{S \vdash f(g(Y), X)} \quad \frac{X \in S}{S \vdash X}}{S \vdash g(Y)} \text{ROOT} \quad S \vdash \text{enc}(g(Y), N)}{S \vdash N}$$

Insecurity of the protocol from Definition 8 in the lecture notes. We have to show that there exists a message X^* with $\emptyset \vdash X^*$ and $S \vdash N$ where $S := \{f(g(Y), X^*), enc(g(Y), N)\}$. (Compare to Lemma 2 in the lecture notes.)

Let $X^* := one$. Since $\emptyset \vdash one$, we have $\emptyset \vdash X^*$. We show $S \vdash N$ using the following derivation:

$$\frac{\frac{\frac{f(g(Y), one) \in S}{S \vdash f(g(Y), one)}}{S \vdash g(Y)} \quad f(g(Y), one) \approx g(Y)}{S \vdash g(Y)} \quad \frac{enc(g(Y), N) \in S}{S \vdash enc(g(Y), N)}}{S \vdash N}$$

Insecurity of the protocol from Definition 10 in the lecture notes. We have to show that there exist messages X^*, Y^* with $\emptyset \vdash X^*, Y^*$ and $X^*, Y^* \neq zero$ and $X^*, Y^* \neq one$ and $X^* \neq Y^*$ and $S \vdash N$ where $S := \{f(g(W), X^*), f(g(W), Y^*), enc(f(g(W), Z), N), Z\}$. (Compare to Lemma 2 in the lecture notes.)

We let X^* be an adversary-nonce (i.e., $X^* \in N_A$). Then $\emptyset \vdash X^*$. We let $Y^* := add(X^*, one)$. Then

$$\frac{\emptyset \vdash X^* \quad \emptyset \vdash one}{\emptyset \vdash add(X^*, one)}$$

Thus $\emptyset \vdash Y^*$.

We have $S = \{f(g(W), X^*), f(g(W), add(X^*, one)), enc(f(g(W), Z), N), Z\}$. We show $S \vdash N$ using the following derivation:

$$\frac{\frac{\frac{f(g(W), add(X^*, one)) \in S}{S \vdash f(g(W), add(X^*, one))} \quad \begin{array}{l} f(g(W), add(X^*, one)) \\ \approx mul(f(g(W), X^*), f(g(W), one)) \\ \approx mul(f(g(W), X^*), g(W)) \end{array}}{S \vdash mul(f(g(W), X^*), g(W))} \quad \begin{array}{c} \vdots \\ \vdots \end{array}}{S \vdash g(W)} \quad \frac{f(g(W), X^*) \in S}{S \vdash f(g(W), X^*)} \text{ Div} \quad \frac{Z \in S}{S \vdash Z}}{S \vdash f(g(W), Z)} \quad \begin{array}{c} \vdots \\ \vdots \end{array} \quad \frac{enc(f(g(W), Z), N) \in S}{S \vdash enc(f(g(W), Z), N)}}{S \vdash N}$$

Note: When we write $\frac{\frac{A}{B} \quad C}{D}$ we mean $\frac{\frac{A}{B} \quad C}{D}$. The dots are only used to allow for a more flexible layout.

More problems with the symbolic models from the lecture. Further problems (that are not solved by the symbolic model presented above) include:

- One can apply the operations g and f to arbitrary messages, even messages that are of the wrong type. For example, what is the meaning of $g(g(w))$ (note that $g(w)$ is an element of some group, while $g(\dots)$ expects an integer as an argument). And what is the meaning of $f(g(w), enc(K_1, K_2))$? One possibility to solve this issue is to introduce the concept of a well-typed message (e.g., the argument of g must be a message containing only *add*, *mul*, *zero*, *one* and nonces) and restrict \vdash so that only well-typed messages can be deduced.
- Related to the previous point: We have the constant *one* which has different interpretation in different contexts. For example, in $g(one)$, *one* is interpreted as the integer 1, and in $g(zero) \approx one$, *one* is interpreted as the unit of the group. The solution would be to introduce two constructors one_{int} and one_{group} to distinguish between the two interpretations. Similarly, *mul* is sometimes used as the multiplication on integers, and sometimes as the multiplication in the group.
- There should be operations modeling the fact that we can invert group elements, as well as negate integers. In case of a known order group, we can also invert integers modulo the group order (e.g., $f(g(W), inv(W)) \approx g(one)$).
- The symbolic model so far implies collision-resistance of g . That is, one cannot deduce M_1, M_2 such that $M_1 \not\approx M_2$ and $g(M_1) \approx g(M_2)$. In a known order group, however, the adversary can choose M_1 at random and $M_2 := M_1 + \text{ord } G$ where $\text{ord } G$ is the group order. Then $M_1 \neq M_2$ as bitstrings, but $g^{M_1} = g^{M_2}$. A solution for this would be to introduce a symbol *order* and rules like $g(order) \approx g(zero)$.

Unfortunately, when implementing all the above suggestions, we get a very complex symbolic model.

Problem 2: A security proof in the Dolev-Yao model (10 Points)

Consider the following set of messages:

$$M ::= N | N_A | enc(M, M) | hash(M).$$

Here N and N_A represent countably infinite sets.

The interpretation of $enc(M_1, M_2)$ is a symmetric encryption of M_2 with key M_1 , the interpretation of $hash(M)$ is a hash value of M .

The deduction relation \vdash is the smallest relation satisfying the following rules:

$$\frac{M_1 \in S}{S \vdash M_1} \text{ELEM} \quad \frac{N \in N_A}{S \vdash N} \text{ADVNONCE} \quad \frac{S \vdash M_1 \quad S \vdash enc(M_1, M_2)}{S \vdash M_2} \text{DEC}$$

$$\frac{S \vdash M_1, M_2}{S \vdash enc(M_1, M_2), hash(M_1)} \text{CONS}$$

Here M_1, M_2 are messages.

Consider the following protocol:

- Bob chooses nonces $N_{secret}, K_1, K_2 \in N$.
- Bob sends $hash(K_1), enc(K_1, K_2)$ to the adversary.
- Bob expects a message M_1 from the adversary.
- If M_1 is of the form $enc(K_1, K')$ for some message K' , Bob sends $enc(K', N_{secret})$ to the adversary.

Show that this protocol does not leak the nonce N_{secret} . More precisely, show the following claim:

Claim 1 *Let $S_1 := \{hash(K_1), enc(K_1, K_2)\}$. Assume that $S_1 \vdash enc(K_1, K')$. Let $S_2 := S_1 \cup \{enc(K', N_{secret})\}$. Then $S_2 \not\vdash N_{secret}$.*

(Although you do not need to do so for solving this problem, I strongly recommend that you make it clear for yourself why Claim 1 indeed expresses that the nonce N_{secret} is not leaked in the protocol.)

Hint: To show Claim 1, first try to find some invariant (1) that holds for all M with $S_1 \vdash M$. For every rule from the definition of \vdash , show that if M_1, M_2 satisfy the invariant (1), then the message in the conclusion of the rule also satisfies invariant (1). Then you can use your invariant to show what K' must be. Then try to find some invariant (2) that holds for all M with $S_2 \vdash M$. For every rule from the definition of \vdash , show that if M_1, M_2 satisfy the invariant (2), then the message in the conclusion of the rule also satisfies invariant (2). Finally, use that N_{secret} does not satisfy invariant (2) and you are done.

Solution. We first define a set A of terms by the following grammar:

$$A ::= N_A | enc(A, A) | hash(A) | hash(K_1) | enc(K_1, K_2)$$

Note that other than in the grammar for messages, N does not occur in this grammar.

Invariant (1): For each message M with $S_1 \vdash M$, we have that $M \in A$.

To prove this, we perform an induction over the rules defining \vdash . That is, we show that for each rule, if all messages occurring in the premise of the rule satisfy invariant (1), we have that the message in the conclusion of the rule also satisfies invariant (1).

For the rules ELEM and ADVNONCE, this follows immediately because $S_1 \subseteq A$ and $N_A \subseteq A$.

For the rule CONS, this follows because the grammar is closed under applications of the constructors *hash* and *enc*.

In rule DEC, we have the premise $S_1 \vdash M_1, enc(M_1, M_2)$. By induction hypothesis, we have that $M_1, enc(M_1, M_2) \in A$. Thus $M_1 \neq K_1$. (Note: $K_1 \in N$.) Hence $enc(M_1, M_2) \neq enc(K_1, K_2)$. Thus, by the grammar of A , $enc(M_1, M_2)$ matches the pattern $enc(A, A)$, i.e., $M_1, M_2 \in A$. The conclusion of DEC is $S_1 \vdash M_2$, thus we are done.

We have shown invariant (1). By assumption of Claim 1, $S_1 \vdash enc(K_1, K')$. With invariant (1), it follows that $enc(K_1, K') \in A$. By the grammar of A , this implies that $enc(K_1, K')$ matches $enc(A, A)$ or $enc(K_1, K_2)$. Since $K_1 \notin A$, the first case cannot occur. Thus $enc(K_1, K') = enc(K_1, K_2)$ and hence $K' = K_2$. Then $S_2 = \{hash(K_1), enc(K_1, K_2), enc(K_2, N_{secret})\}$.

We define the following grammar:

$$B ::= N_A | enc(B, B) | hash(B) | hash(K_1) | enc(K_1, K_2) | enc(K_2, N_{secret})$$

Invariant (2): For each message M with $S_2 \vdash M$, we have that $M \in B$.

To prove this, we perform an induction over the rules defining \vdash . That is, we show that for each rule, if all messages occurring in the premise of the rule satisfy invariant (2), we have that the message in the conclusion of the rule also satisfies invariant (2).

For the rules ELEM and ADVNONCE, this follows immediately because $S_2 \subseteq B$ and $N_A \subseteq B$.

For the rule CONS, this follows because the grammar is closed under applications of the constructors *hash* and *enc*.

In rule DEC, we have the premise $S_2 \vdash M_1, enc(M_1, M_2)$. By induction hypothesis, we have that $M_1, enc(M_1, M_2) \in A$. Thus $M_1 \neq K_1, K_2$. (Note: $K_1, K_2 \in N$.) Hence $enc(M_1, M_2) \neq enc(K_1, K_2), enc(K_2, N_{secret})$. Thus, by the grammar of B , $enc(M_1, M_2)$ matches the pattern $enc(A, A)$, i.e., $M_1, M_2 \in A$. The conclusion of DEC is $S_2 \vdash M_2$, thus we are done. **.noituloc**