Assignment 2

Aryan Nath
Information Security

March 2, 2025

Problem 1

Write an expository note on the material discussed during Lecture 4 (February 11). We want you to develop and demonstrate your intuition for what motivated the technical definitions and theorems discussed in class.

Answer

In one of our previous lectures we discussed the protocol for establishing a handshake between users A and B using a random nonce n generated by A. The protocol for this handshake is:

$$\begin{aligned} A \to B &: \text{pair}(\text{"}A_{id}\text{"}, \text{aenc}(n, pk_B)) \\ B \to A &: \text{aenc}(n, pk_A) \end{aligned}$$

This protocol is secure if no one except A and B gets access to the nonce n. The assumptions we made are that encryption is bullet proof and communication is done over an adversarial channel, that is, all messages sent over the network can be accessed by an adversary.

Given these assumptions, an adversary I can get access to the nonce n using the following attack on the handshake:

$$\begin{aligned} A \to (I)B &: \text{pair}(\text{"}A_{id}\text{"}, \text{aenc}(n, pk_B)) \\ I \to B &: \text{pair}(\text{"}I_{id}\text{"}, \text{aenc}(n, pk_B)) \\ B \to (I)A &: \text{aenc}(n, pk_I) \end{aligned}$$

The adversary 'I' can now decrypt $\text{aenc}(n, pk_I)$ *with its secret key* sk_I *and get the nonce* n.

$$I \rightarrow A : \text{aenc}(n, pk_A)$$

A receives the response as they expected and has no idea that n *has been exposed.*

here $A \rightarrow (I)B$ means that I has deceived A to send messages to I while thinking that they are sending messages to B.

Since this is a simple protocol, it was easy for us to determine that the protocol is not safe. For longer and more complex protocols, we need a formal verification system which we can use to determine whether the adversary gets access to such a nonce n or a secret that only Alice and Bob are supposed to know.

This motivates the requirement for the Dolev Yao model, which allows us to formally define the 'knowledge base' of the adversary and finally answer the question $I \vdash n$?

The Dolev Yao Model and its preliminaries:

1. Knowledge Base: Term algebra is used to give the knowledge of the protocol actors as follows:

   (a) The grammar for a $term \coloneqq keys|messages|nonces|ids|$Any operator applied to the subterms of

   (b) Operator grammar is defined as $operator \coloneqq aenc|senc|sign|verify$.

   (c) Equations are defined as $adec(aenc(m,k),k^{-1}) = m, verify(sign(t,id'),t,id) = true$ if id = id'

2. How do the protocol actors infer knowledge?

   Given the knowledge base X of an agents / intruder, X can be used to infer knowledge using the following rules:

   (a) Axiom:

   $$\frac{}{\mathbb{X} \cup \{t\} \vdash t} \quad ax(t \in X) \qquad \text{(Destruction Rule)}$$

   (b) Pair:

   $$\frac{\mathbb{X} \vdash t_0 \quad \mathbb{X} \vdash t_1}{\mathbb{X} \vdash \text{pair}(t_0, t_1)} \quad pair \qquad \text{(Introduction Rule)}$$

   (c) Split:

   $$\frac{\mathbb{X} \vdash \text{pair}(t_0, t_1)}{\mathbb{X} \vdash t_i} \quad split_i \qquad \text{(Destruction Rule)}$$

   (d) Asymmetric Encryption:

   $$\frac{\mathbb{X} \vdash t \quad \mathbb{X} \vdash k}{\mathbb{X} \vdash \text{enc}(t, k)} \quad enc \qquad \text{(Introduction Rule)}$$

(e) Asymmetric Decryption:

$$\frac{\mathbb{X} \vdash \mathrm{enc}(t, k) \quad \mathbb{X} \vdash \mathrm{inv}(k)}{\mathbb{X} \vdash t} \; dec \qquad \text{(Destruction Rule)}$$

(f) Generate public key using parameter k:

$$\frac{\mathbb{X} \vdash k}{\mathbb{X} \vdash \mathrm{pk}(k)} \; pk \qquad \text{(Introduction Rule)}$$

So, in the **Dolev Yao model** we formally prove/disprove $X \vdash t$? by collecting information in the knowledge base X by appending all terms (messages) accessible by the adversary I over the network and performing inference using the above rules.

3. Capabilities of the 'standard' adversary:

   - The adversary can see any message.
   - The adversary can block any message.
   - The adversary can redirect any message.
   - The adversary can masquerade as anyone.
   - The adversary can initiate new communication.
   - The adversary can mix-and-match messages.
   - The adversary can genearte new messages.
   - The adversary cannot break the underlying 'perfect' cryptography.

4. Given X, t how difficult is it to verify $X \vdash t$
   For a *passive intruder*, who can just listen to the network traffic and make inferences based on just the information they have access to, this problem can be solved in $\mathbf{O(|st(X \cup t)|^3)}$ ($\mathbf{st}$ is the subterm of $X \cup t$, which we will talk about after this part). The fastest way to solve this problem is linear in the size of $\mathbf{st}(X \cup t)$.

   However, the same problem for an *active intruder* who can modify intercepted message and masquerade as other parties is much more difficult to solve. The active adversary can (1) call unboundedly many sessions between other parties and can (2) inject unboudedly large terms into the protocol (since $|\mathbf{st(X)}| \propto |X|$, this makes the inference time for our Dolev Yao model also unbounded). (1) can be solved by bounding the number of sessions and given boundedly many sessions, (2) can be solved as well. Still, with boundedly many sessions, the problem for *active intruders* is *NP-complete* (polynomially many terms are injected by the adversary; hence, polynomiallly number of derivability checks are required and each check takes polynomial time).

5. The subterm $\mathbf{st(t)}$ of a term t is the smallest set such that:
   1)$t \in \mathbf{st(t)}$
   2)$pair(t_1, t_2) \in \mathbf{st(t)}$, then $\{t_1, t_2\} \subseteq \mathbf{st(t)}$

3)$aenc(t', k) \in \mathbf{st(t)}$, then $\{t', k\} \subseteq \mathbf{st(t)}$
4) for a set X, $\mathbf{st(X)} = \cup_{t \in X}\mathbf{st(t)}$. 5) $\mathbf{st(X)} \leq \cup_{t \in X}\mathbf{st(t)}$
(**overview of the proof: st(pair(t_1,t_2))** = { pair(t_1,t_2) ∪ **st**(t_1) ∪**st**(t_2) } $\leq 1 + |t\_1| + |t\_2|$ using induction hypothesis

st(aenc(t',k)) = { aenc(t',k) ∪**st**(t') ∪**st**(k) } ≤ 1+$|t'|$+$|k|$ by induction hypothesis)
We will visit subterms again once we reach the polynomial time derivation algorithm I mentioned above.

6. Do we have a shortest proof for $x \vdash t$?
Yes, we do. These are *normal proofs*. In fact, every *normal proof* of the form $X \vdash t$ is the shortest proof for $X \vdash t$ (provable by contradiction).

   Normal proof: Given a set X, a proof $X \vdash t$ is normal if the major premise of every elimination rule in the proof is the conclusion of an elimination rule (pk, pair, and split increase the length of the proof).

   Also, whenever there exists a proof for $X \vdash t$, there exists a normal proof for $X \vdash t$. Hence, *for every proof there exists a shortest, normal proof.*

7. **Subterm property**:
Before I mention the polynomial time algorithm for derivability, we need to first visit the subterm property in order derive the bound on the time complexity of the algorithm.

   Subterm Property: Let π be a normal derivation with conclusion $X \vdash t$ and last rule **str**. Let $X \vdash s$ occur in π. Then $s \in \mathbf{st(X{\cup}t)}$. Furthermore, if **str** is an elimination rule, then $s \in \mathbf{st(X)}$.

   Now, we describe the poly time derivability algorithm.

8. **Polynomial time algorithm for derivability:**

   Given X, t we need to check whether $X \vdash t$. Let $N = \mathbf{st(X \cup t)}$. Now, a derivation of $X \vdash t$ can be done using just the subterms of $X \cup t$, which are exactly $\mathbf{st(X{\cup}t)}$, as anything outside their subterms would lead to a redundant conclusion that just adds complexity to our proof and is hence redundant. Furthermore, part on normal proofs we know that if $X \vdash t$ exists then there is a minimum size normal proof for $X \vdash t$. We can directly see that in any minimum size proof if $s \in \mathbf{st(X{\cup}t)}$, then $X \vdash s$ occurs at most once on each branch of the derivation as we could otherwise obtain a shorter derivation for s and hence t, contradicting the normality of the proof.

   Since the largest the number of subterms we can have in the derivation of $X \vdash t$ is $|\mathbf{st(X)}| + |\mathbf{st(t)}|$, a normal minimum sized proof of $X \vdash t$ is bounded by N. Hence, if we calculate each $t' \in \mathbf{st(X{\cup}t)}$ derivable at each level of the proof, starting from X and do not go beyond N levels. If t has been derived by then then $X \vdash t$? is yes, otherwise no (since every proof has a normal minimum sized proof).

Using the above the polynomial time algorithm for derivability is constructed as follows:

Algorithm 1 Cubic algorithm for computing $\mathit{derive}(X_0)$

```
$i \leftarrow 0$;
$Y \leftarrow X_0$;
while $i \leq N$ do
    $Z \leftarrow Y \cup \{(t_0, t_1) \in \mathbf{st} \mid t_0, t_1; \in Y\} \cup \{\{t'\}_k \in \mathbf{st} \mid t', k \in Y\}$;
    $Z \leftarrow Z \cup \{t' \mid \exists r : (t', r) \in Y \text{ or } (r, t') \in Y\}$;
    $Z \leftarrow Z \cup \{t' \mid \exists k : \{t'\}_k \in Y \text{ and } \mathit{inv}(k) \in Y\}$;
    $Y \leftarrow Z$;
end while
return $Y$;
```

Since any minium size proof by the subterms of $X \cup t$ is bounded N and we have derived all possible terms from X over N levels, if $X \vdash t$ is possible then it would have added to Y.

Problem 2

Test your understanding by solving these exercises.

a. What is the size of the term below? What are its subterms?

$$\mathsf{aenc}(\mathsf{pair}(\mathsf{aenc}(m, \mathsf{pk}(k_1)), \mathsf{pair}(n_1, n_2), \mathsf{pk}(k_2)))$$

b. Given the following X, can X derive $\mathsf{aenc}(m, \mathsf{pk}(k_3))$ using a normal proof?

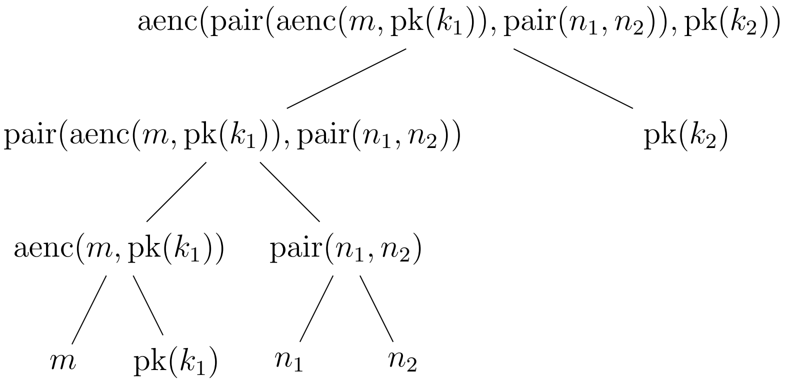
$$\begin{aligned} X = \{ & \\ & \mathsf{aenc}(m, \mathsf{pk}(k_1)), \\ & \mathsf{pair}(k_2, \mathsf{aenc}(\mathsf{pair}(m, k_1), \mathsf{pk}(k_3))), \\ & \mathsf{aenc}(k_3, \mathsf{pk}(k_2)), \\ & \} \end{aligned}$$

Answer:

(a) Given $t = aenc(pair(aenc(m, pk(k1)), pair(n1, n2)), pk(k2)))$.

The size of a term is equal to the number of its subterms, or equivalently the number of nodes in the term tree of the term.

By just counting the number of subterms of t we can determine the size of t as 9 (assuming $pk(k)$ is an atomic term). Let's verify this using the term tree for $aenc(pair(aenc(m, pk(k1)), pair(n1, n2)), pk(k2)))$.


$\mathrm{aenc}(\mathrm{pair}(\mathrm{aenc}(m, \mathrm{pk}(k_1)), \mathrm{pair}(n_1, n_2)), \mathrm{pk}(k_2))$
$\mathrm{pair}(\mathrm{aenc}(m, \mathrm{pk}(k_1)), \mathrm{pair}(n_1, n_2))$
$\mathrm{pk}(k_2)$
$\mathrm{aenc}(m, \mathrm{pk}(k_1))$
$\mathrm{pair}(n_1, n_2)$
m
$\mathrm{pk}(k_1)$
n_1
n_2

We can see that the number of nodes (subterms) is precisely 9.

All subterms of $aenc(pair(aenc(m, pk(k1)), pair(n1, n2)), pk(k2)))$ have been shown in the above term tree.

(b) First we recall that in a normal proof the major premise of the elimination rules *adec,split, ax* has to be from an elimination rule. Let's verify that X can derive $aenc(m, pk(k3))$ by constructing a normal proof (also remember that if $X \vdash aenc(m, pk(k3))$ then we definitely have a normal proof).

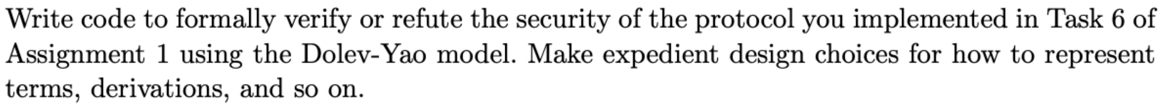
The proof is as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{pair(k_2, aenc(pair(m, k_1), pk(k_3))) \in X}{X \vdash pair(k_2, aenc(pair(m, k_1), pk(k_3)))}\text{ ax [E]}}{X \vdash aenc(pair(m, k_1), pk(k_3))}\text{ split [E]} \qquad \dfrac{\dfrac{\dfrac{pair(k_2, aenc(pair(m, k_1), pk(k_3))) \in X}{X \vdash pair(k_2, aenc(pair(m, k_1), pk(k_3)))}\text{ ax [E]}}{X \vdash k_2}\text{ split [E]} \qquad \dfrac{aenc(k_3, pk(k_2)) \in X}{X \vdash aenc(k_3, pk(k_2))}\text{ ax [E]}}{X \vdash k_3}\text{ adec [E]}}{X \vdash pair(m, k_1)}\text{ adec [E]}}{X \vdash m}\text{ split [E]}$$

$$\dfrac{X \vdash m \qquad \dfrac{X \vdash k_3}{X \vdash pk(k_3)}\text{ pk [I]}}{X \vdash aenc(m, pk(k_3))}\text{ aenc [I]}$$

(Assuming k_3 is a parameter which can be used in the rule pk to derive $pk(k_3)$, else we cannot get $pk(k_3)$ and X cannot derive $aenc(m, pk(k_3))$)

Problem 3



Write code to formally verify or refute the security of the protocol you implemented in Task 6 of Assignment 1 using the Dolev-Yao model. Make expedient design choices for how to represent terms, derivations, and so on.

Answer

The code and output have been submitted.