1. [20 points]

Protocol(A, B) {
  chan ← [];  
hst ← [];  
mKey ← random();
  startSystem(A, Client(A,B,mKey));
  startSystem(B, Server(B,A,mKey));
  startSystem(Attacker());
}

Attacker() {
  read and write chan
}

Client(A, B, mKey) {
  // atomicity points: 1
  nL ← 0;
  while (true) {
    nL ← nL + 5;
    tx([A,B,nL]);
 1: msg ← rx([B,A,...]);
    if (msg[3] = enc(mKey,nL)) {
      nR ← msg[2];
      hst.append([A,nL,nR]);
      tx([A,B,nL,enc(mKey,nR)]);
    }
  }
}

Server(B, A, mKey) {
  // atomicity points: 1,2
  nL ← 0;
  while (true) {
    msg ← rx([A,B,..]);
    nR ← msg[2];
    nL ← nL + 7;
    tx([B,A,nL,enc(mKey,nR)]);
 2: msg ← rx([A,B,...]);
      hst.append([B,nL,nR]);
    }
  }
}

For each assertion below, prove or disprove whether the assertion holds for Protocol. If you prove, present an invariably-complete predicate that implies the assertion’s predicate. If you disprove, present a counter-example evolution.

a. Inv (mKey ncf α)

b. Inv forall(i in hst.keys: hst[i] = [B,nB,nA] ⇒ [A,nA,nB] in hst[0..i-1])
Solution to part a [10 points]

Informal argument [4 points]

It suffices to show that any $mKey$-term in $\alpha$ is a secure encryption using $mKey$ (because of axiom 2). The messages sent by $A$ or $B$ contain three kinds of fields: ids ($A$, $B$), $nL$ values, and $\text{enc}(mKey, nR)$ values. Only the last kind involve $mKey$, and these are secure encryptions using $mKey$ because even though the attacker can write to $\text{chan}$, the $mKey$ values it can write are themselves secure encryptions using $mKey$.

Proof [6 or 10 (if Informal argument absent)]

The conjunction of $C_1$–$C_4$ is invariantly complete, and $C_1$ implies ($mKey \ ncf \ \alpha$).

\[
\begin{align*}
C_1 &: \ (y \ in \ \alpha . \ \text{inpts}(mKey)) \Rightarrow (y \ seu \ mKey) \quad [2 \ points] \\
C_2 &: \ (y \ in \ \text{chan}. \ \text{inpts}(mKey)) \Rightarrow (y \ seu \ mKey) \quad [2 \ points] \\
C_3 &: \ A.\text{nL}.\text{inpts}(mKey) = [] \quad [1 \ point] \\
C_4 &: \ B.\text{nL}.\text{inpts}(mKey) = [] \quad [1 \ point]
\end{align*}
\]

Details:

\[
\begin{array}{c|c|c|c|c}
\text{initial step} & C_1 & C_2 & C_3 & C_4 \\
\hline
A.1 & \text{true} & \text{true} & \text{true} & \text{true} \\
B.1 & C_2, C_3, C_1 & C_2, C_3 & \text{true} & C_4 \\
B.2 & C_1 & C_2 & C_3 & C_4 \\
\text{attacker write} & C_1 & C_1, C_2 & C_3 & C_4 \\
\end{array}
\]

Solution to part b [10 points]

We disprove the assertion.

Informal argument [4 points]

Before an execution of B.1, The attacker knows the B.1. When B is at 1, the attacker knows the challenge, say $nB$, that B.1 will issue next. So it writes message $[A, B, 1, nB]$ to $\text{chan}$. B responds with message $[B, A, nB, \text{enc}(mKey, nB)]$. The attacker, using the last field of this message, makes up the response expected by B.2, which causes B to add an entry to $\text{hst}$ without $A$ receiving any message.

Proof [6 points]

Counter-example evolution:

\[
\begin{align*}
& \text{Initial step} \\
& \text{After: } [A, B, 1] \text{ in } \text{chan}; \text{hst} = []. \\
& \text{Attacker changes field 2 in the above message to 7.} \quad [2 \ points] \\
& \text{B.1} \\
& \text{After: } [B, A, 7, \text{enc}(mKey, 7)] \text{ in } \text{chan}; B.\text{nL} = B.\text{nR} = 7; \text{hst} = []. \\
& \text{Attacker, using enc}(mKey, 7) \text{ field in above message, sets chan to } [[B, A, \text{enc}(mKey, 7)]] . \quad [2 \ points] \\
& \text{B.2} \\
& \text{After: } \text{hst} = [[B, 7, 7]]. \text{ Assertion’s predicate not satisfied.} \quad [1 \ points]
\end{align*}
\]

–6 if attacker does not come up with $\text{enc}(mKey, B.\text{nL})$
2. [20 points]

An organization has a PKI (public-key infrastructure) for its users consisting of a single certification authority (CA) and a single directory server (DS) that serves certificates and CRLs to any user. Certificates have an expiry time of 1 year. CRLs are issued weekly. Answer the following questions. Be brief and precise.

a. What is the minimum information that a user in the organization must remember. Is any of it secret.

b. What is the minimum information a certificate must have.

c. Describe the steps that a user A takes to establish a connection to a user B with a shared session key.

**Solution to part a [3 points]**

A user must remember at the minimum
- its own private key [1 point]
- the CA’s public key [1 point]
Its private key is secret, shared with no one. [1 point]

**Solution to part b [7 points]**

A certificate has the following at a minimum:
- serial number [1 point]
- user id [1 point]
- user public key [2 point]
- expiry date [1 point]
- issuer (CA) id [1 point]
- issue date [1 point]
- signature on above by CA [2 point]

**Solution to part b [10 points]**

- A asks DS for B’s certificate, say certB, and latest CRL, say CRL (and A’s certificate if needed). [2 points]
- A verifies certificate(s) using the CA’s public key. A verifies CRL using the CA’s public key, checks certB not in CRL [1 point]
- A generates a session key, say k, signs it (with its private key), encrypts it (with B’s public key). [3 points]
- A sends \([A,B,\text{enc}(\text{pubB},[k,\text{enc}(\text{prvA},k)]),\text{certA},\text{CRL}]). [3 points]
- B gets and verifies certA and CRL (just as A did with certB and CRL) [3 points]
- B decrypts enc(pubB,k) to get k. [1 points]
Function $\text{enc}(\ldots)$ can encrypt arbitrary values, including integers (e.g., $\text{enc}(K,3)$) and structures (e.g., $\text{enc}(K,[3,5])$). Note that $\text{enc}(K,3)$ is different from $\text{enc}(K,[3])$.

In the program below, the following happens repeatedly if the attacker does nothing:

- $A$ sends $[A,B,1,nA]$
- $B$ responds with $[B,A,\text{enc}(\text{mKey},[nB,nA+1])]$
- $A$ responds with $[A,B,\text{enc}(\text{mKey},[nA,nB+1])]$

```
Protocol(A, B) {
    chan ← []; // connection history
    hst ← []; // connection history
    mKey ← random();
    startSystem(A, Client(A,B,mKey));
    startSystem(B, Server(B,A,mKey));
    startSystem(Attacker());
}
```

```
Client(A, B, mKey) {
    // atomicity points: 1
    sKey ← random();
    while (true) {
        nL ← random();
        tx([A,B,1,nL]);
        1: msg ← rx([B,A,1,\ldots]);
        z ← dec(mKey, msg[2]);
        if (z.size = 2 and z[1] = nL+1) {
            nR ← z[0];
            hst.append([A,nL,nR]);
            sKey ← enc(mKey, nL+nR);
            tx([A,B,2,enc(mKey,[nL,nR+1])]);
        }
    }
}
```

```
Server(B, A, mKey) {
    // atomicity points: 1,2
    sKey ← random();
    while (true) {
        1: msg ← rx([A,B,1,\ldots]);
        nR ← msg[3];
        nL ← random();
        tx([B,A,enc(mKey,[nL,nR+1])]);
        2: msg ← rx([A,B,2,\ldots]);
        z ← dec(mKey, msg[3]);
        if (z.size = 2 and z[0] = nR and z[1] = nL+1) {
            hst.append([B,nL,nR]);
            sKey ← enc(mKey, nL+nR);
        }
    }
}
```

For each assertion below, prove or disprove whether the assertion holds for Protocol. If you prove, present an invariantly-complete predicate that implies the assertion’s predicate. If you disprove, present a counter-example evolution.

a. $\text{Inv} \ (\text{mKey ncf } \alpha)$

b. $\text{Inv} \ (A.sKey ncf } \alpha)$

c. $\text{Inv} \ \forall i \in \text{hst.keys: } \text{hst}[i] = [B,nB,nA] \Rightarrow [A,nA,nB] \text{ in hst}[0..i-1]$
Solution to part a [3 points]

The conjunction of the following predicates is invariantly complete and implies $\text{mKey} \ ncf \ \alpha$.

- $C_1$: $(y \in \alpha.\text{inpts}(\text{mKey})) \Rightarrow (y \text{ seu } \text{mKey})$ [1 points]
- $C_2$: $(y \in \text{chan.}\text{inpts}(\text{mKey})) \Rightarrow (y \text{ seu } \text{mKey})$ [1 points]
- $C_3$: $\text{A.nL.}\text{inpts} = \emptyset$ [1 points]

Solution to part b [8 points]

We prove it.

Informal argument [4 points] Suppose A.1 sets $\text{A.sKey}$ to $\text{enc}(\text{mKey}, nA+nB)$. The attacker has $nA$, because it was previously sent in the open in a $[A,B,1,\ldots]$ message. But $nB$ has not been sent in the open. So $\text{A.sKey}$ is not computable by the attacker.

Proof [5 or 8 (if no informal argument) points] The conjunction of the following predicates is invariantly complete and implies $\text{A.sKey} \ ncf \ \alpha$ (assuming $\text{Inv} (\text{mKey} \ ncf \ \alpha)$.

- $C_1$: $(y \in \alpha.\text{inpts}(\text{mKey})) \Rightarrow (y \text{ seu } \text{mKey})$ and $y = \text{enc}(\text{mKey}, [\text{int}, \text{int}])$ [2 points]
- $C_2$: $(y \in \text{chan.}\text{inpts}(\text{mKey})) \Rightarrow (y \text{ seu } \text{mKey})$ and $y = \text{enc}(\text{mKey}, [\text{int}, \text{int}])$ [2 points]
- $C_3$: $\text{A.nL.}\text{inpts} = \emptyset$ [1 points]

Can also replace $y = \text{enc}(\text{mKey}, [\text{int}, \text{int}])$ with $y$ is not $\text{enc}(\text{mKey}, \text{int})$. Without saying this, you get close to zero points. Saying $y = \text{enc}(\text{mKey}, nA+nB)$ is no good.

Solution to part c [9 points]

We prove it.

Informal argument [4 points] Suppose $[B,nB,nA]$ is added to $\text{hst}$ at time $t_0$. Then just before $t_0$, B.2 received message $[A,B,2,\text{enc}(\text{mKey}, [nA,nB+1])]$ and $B.\text{nL} = nB$, $B.\text{nR} = nA$, and B at 2 held. Let $B.\text{nL}$ be set to $nB$ at time $t_1$. So B does nothing between $t_1$ and $t_0$. The attacker does not have $\text{mKey}$. So message $[A,B,2,\text{enc}(\text{mKey}, [nA,nB+1])]$ was sent by A.1 at some time $t_2$ between $t_1$ and $t_0$. The same execution of A.1 also entered $[A,nA,nB]$ into $\text{hst}$.

Proof [5 points] The conjunction of the following predicates is invariantly complete and implies $\text{A.sKey} \ ncf \ \alpha$ (assuming $\text{Inv} (\text{mKey} \ ncf \ \alpha)$.

- $E_0$: $(i \in \text{hst.keys}) \text{ and } \text{hst}[i] = [B,nB,nA] \Rightarrow ([A,nA,nB] \text{ in } \text{hst}[0..i-1])$
- $E_1$: $(B \text{ at } 2)$ and $B.\text{nL} = nB$ and $B.\text{nR} = nA$ and $([A,B,\text{enc}(\text{mKey}, [nA,nB+1])] \text{ in } \text{chan})$ $\Rightarrow ([A,nA,nB] \text{ in } \text{hst})$
- $E_2$: $(B \text{ at } 2)$ and $B.\text{nL} = nB$ and $B.\text{nR} = nA$ and $(\text{enc}(\text{mKey}, [nA,nB+1]) \text{ in } \alpha)$ $\Rightarrow ([A,nA,nB] \text{ in } \text{hst})$