

1 Demo4 Diagrams

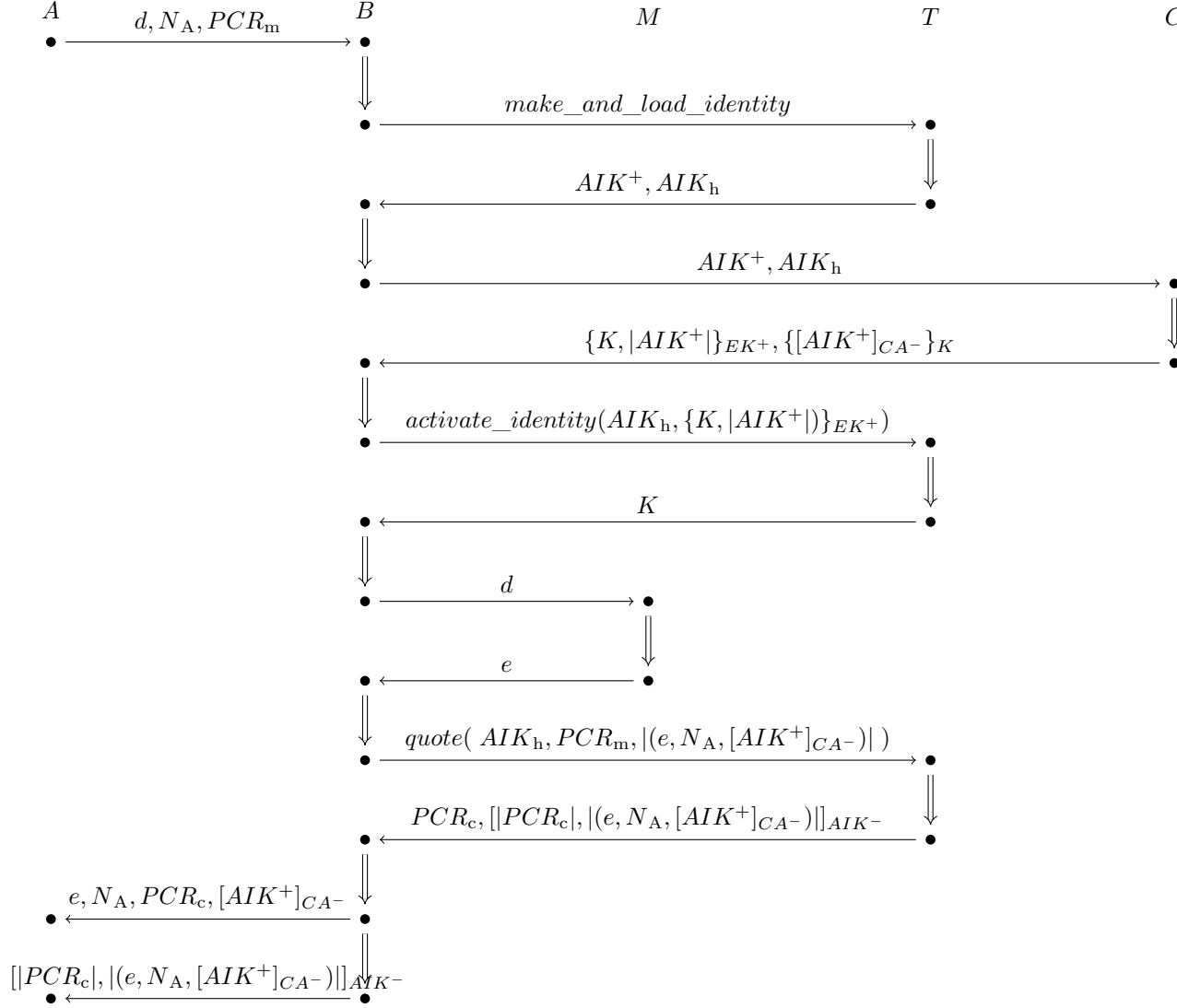
1.1 Message Sequence Diagram

$A \rightarrow B : d, N_A, PCR_m \text{ on } C_{AB}$
 $B \rightarrow T : \text{make_and_load_identity on } C_{BT}$
 $T \rightarrow B : AIK^+, AIK_h \text{ on } C_{TB}$
 $B \rightarrow C : B, AIK^+ \text{ on } C_{BC}$
 $C \rightarrow B : \{K, |AIK|\}_{EK^+}, \{[AIK^+]_{CA^-}\}_{K^+} \text{ on } C_{CB}$
 $B \rightarrow T : \text{activate_identity}(AIK_h, |AIK|) \text{ on } C_{BT}$
 $T \rightarrow B : K \text{ on } C_{TB}$
 $B \rightarrow M : d \text{ on } C_{BM}$
 $M \rightarrow B : e \text{ on } C_{MB}$
 $B \rightarrow T : \text{quote}(AIK_h, PCR_m, |(e, N_A, [AIK^+]_{CA^-})|) \text{ on } C_{BT}$
 $T \rightarrow B : PCR_c, [|PCR_c|, |(e, N_A, [AIK^+]_{CA^-})|]_{AIK^-} \text{ on } C_{TB}$
 $B \rightarrow A : e, N_A, PCR_c, [AIK^+]_{CA^-} \text{ on } C_{BA}$
 $B \rightarrow A : [|PCR_c|, |(e, N_A, [AIK^+]_{CA^-})|]_{AIK^-} \text{ on } C_{BA}$

KEY

A: Appraiser
B: Attestation Agent
T: TPM
C: Certificate Authority
M: Measurer
d : desired evidence
e : gathered evidence
 N_A : nonce generated by A
 PCR_m : pcr mask indicating desired pcr registers
 PCR_c : pcr composite structure containing select pcr register values
 AIK_h : AIK key handle(used by TPM to reference loaded keys)
K: Session key created by C

1.2 Strand Space Diagram



Appraisal

```

main = do
  putStrLn "START main of Appraiser\n"
  (pcrSelect, nonce) ← mkTPMRequest ([0..23]::[Word8])
  let mReq = mkMeasureReq [0..2]
      req = (Request mReq pcrSelect nonce)
  putStrLn $ show req
  putStrLn $ "Press enter to send Request"
  getChar
  chan ← sendRequest req
  putStrLn "\nSENT REQUEST TO ATTESTATION AGENT...\n"
  putStrLn "\nRECEIVING RESPONSE...\n"
  result ← receiveResponse chan
  case (result) of
    (Left err) →
      putStrLn "Error getting response. Error was: " ++ err
    (Right response) → do
      putStrLn $ "Received: "
      putStrLn $ show response ++ "\n"
      putStrLn "Evaluating Response: "
      result ← evaluate req response
      showDemoEvalResult result

  putStrLn "END main of Appraiser"

```

Attestation

```
main = do
  putStrLn "START main of Attestation"

  apprChan ← server_init appId
  measChan ← client_init meaId

  publicEK ← tpm_key_pubek tpm
  tkShn ← tpm_session_oiap tpm
  tpm_takeownership tpm tkShn publicEK ownerPass srkPass
  putStrLn "\nTPM OWNERSHIP TAKEN\n"

  req ← receiveRequest apprChan
  resp ← mkResponse req
  sendResponse apprChan resp

where
  ownerPass = tpm_digest_pass oPass
  srkPass = tpm_digest_pass sPass
```

```

mkResponse :: Either String Request → Att Response
mkResponse (Right Request desiredE pcrSelect nonce) = do
  enterP "request AIK from TPM"
  x@(iKeyHandle, iSig) ← liftIO $ createAndLoadIdentKey
  liftIO $ putStrLn "AIK CREATED AND LOADED. "
  liftIO $ putStrLn $ "Handle: " ++ show iKeyHandle ++ "\n"

  caChan ← getPriChan
  pubAIK ← liftIO $ attGetPubKey iKeyHandle aikPass
  let caRequest = mkCARequest aikPass pubAIK iSig
  liftIO $ putStrLn $ show caRequest
  enterP "send CA Request:"
  r@(CAResponse caCertBytes actIdInput) ← liftIO $
    converseWithScottyCA caRequest

  liftIO $ putStrLn $ "SENT CA REQUEST"
  liftIO $ putStrLn $ "\nRECEIVING CA RESPONSE... \n"
  liftIO $ putStrLn $ "Received: " ++ show r

  iShn ← liftIO $ tpm_session_oiap tpm
  oShn ← liftIO $ tpm_session_oiap tpm
  enterP $ "release session key K by calling tpm_activate_identity" ++ ...
  sessionKey ← liftIO $ tpm_activateidentity tpm iShn oShn iKeyHandle
    aikPass ownerPass actIdInput
  liftIO $ putStrLn $ "Released K: " ++ ...
  let keyBytes = tpmSymmetricData sessionKey
      decryptedCertBytes = decryptCTR aes ctr (toStrict caCertBytes)
      caCert = (decode decryptedCertBytes) :: CACertificate

  meaChan ← getMeaChan
  evidenceList ← liftIO $ mapM (getEvidencePiece meaChan) desiredE

  let evBlob = ePack evidenceList qNonce caCert
      evBlobShal = bytestringDigest $ sha1 evBlob
  enterP $ "call tpm_quote with arguments:\n" ++ ...
  quote ← liftIO $ mkQuote iKeyHandle aikPass pcrSelect evBlobShal
  evPack = (EvidencePackage evidenceList qNonce eSig)
  liftIO $ tpm_flushspecific tpm qKeyHandle tpm_rt_key Evict Loaded key

  return (Response evPack caCert quote)

where
  aikPass = tpm_digest_pass "i"
  ownerPass = tpm_digest_pass oPass

```