Research in Information Security (CSE540)

Mid Semester Examination (Monsoon 2023)

International Institute of Information Technology, Hyderabad

Time: 1 hour and 30 minutes Total Marks: 40

Instructions: Answer ALL questions.

This is a closed book and notes examination.

Regular Calculator is allowed.

1. **(a)** Suppose two communicating entities, say A and B would like to establish a secret session key K_s using the following protocol.

   A sends the following message to B:

   $$A \rightarrow B : KU_A || ID_A.$$

   In reply, B sends the following message to A:

   $$B \rightarrow A : ID_B || E_{KU_A}(K_s).$$

   Here, KU_A is the public key of A, ID_A, ID_B the identifier of nodes A and B, and $E(\cdot)$ the encryption algorithm (for example, RSA). Node A finally decrypts the encrypted key using its own private key KR_A and stores the secret key K_s for future communication with node B.

   Show that the above protocol is vulnerable to the man-in-the-middle attack so that the same key K_s can be established among the attacker, node A and node B instead of establishing the secret key between A and B only.

   (b) Consider the following digital signature scheme. Let p be a prime and $\alpha \in Z_p^*$ be a primitive root. Let the key space be $\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod{p}\}$. The values (p, α, β) are the public key, and a is the private key. For the key $K = (p, \alpha, \beta)$, and for a secret random number k, the sender, being a signer, A generates a signature on a message x as $sig_k(x, k) = (\gamma, \delta)$, where $\gamma = \alpha^k \pmod{p}$ and $\delta = (x + a\gamma)k^{-1} \pmod{p-1}$, and then sends the signed message $(x, (\gamma, \delta))$ to the verifier B.

   Devise the corresponding verification algorithm ver_k for B in order to prove the authentication of message by the verifier B showing its correctness proof.

   [5 + 5 = 10]

2. **(a)** Explain three-way authentication mechanism with *user anonymity* property in X.509 authentication service. Here, *user anonymity* property means an adversary can not trace the identity of the user.

   (b) Explain the internal error control mechanism used for security in the datalink layer.

   (c) Better security can be provided by combining both Link-by-Link Encryption (LLE) and End-to-End Encryption (EEE) - Justify. Explain this combined approach with an example.

   [5 + 3 + 2 = 10]

3. **(a)** Explain the Encapsulating Security Payload (ESP) protocol under tunnel mode for providing security of IPv4 packets. Why do you need the Authentication Header (AH) though ESP is better than AH? Justify your answer.

(b) To protect replay attacks, IPSec (IP security) makes use of sliding receiving window and unique sequence number to each IP packet. To remedy the drawbacks found in this procedure, Zhao and Wu proposed an improved and effective solution to that method. Discuss the Zhao-Wu's method explaining how the anti-replay receiving window differs from the sliding receiving window in the original IPSec. Explain the mechanisms for dropping/discarding packets at the receiver side in case of Zhao-Wu's protocol.

[(4+1) + 5 = 10]

4. **(a)** Explain in detail the SSL record protocol architecture and its functionalities.

(b) In a SYN flood attack, a victim sees a dis-appropriate number of SYN packets compared to FIN packets. Define the following parameters. S_i = # of SYN packets arrivals in the i^{th} observation interval, F_i = # of FIN packets arrivals in the i^{th} observation interval, and D_i = normalized difference between S_i and F_i in the i-th observation interval. In most cases, the number of RST packets is a small fraction of the number of FIN packets, and due to this we ignore them.

Construct an anomaly detection system by devising a detection algorithm based on the 'modified cumulative sum' of the previous values of D. What are the disadvantages of this approach?

[5 + 5 = 10]

********** End of Question Paper **********

Research in Information Security (CS8.501)

Class Test 1 (Monsoon 2023)

International Institute of Information Technology, Hyderabad

Time: 70 Minutes Total Marks: 20

Aug 25, 2023

1. Suppose Alice wishes to send a text message to Bob using the RSA algorithm. Bob's public key is the pair $(e, n) = (7, 187)$. Note that $187 = 17 * 11$ and $7 * 23 = 160$.
Alice uses an alphabet set of 10 letters and encodes them as $A = 0, C = 1, D = 2, E = 3, I = 4, N = 5, O = 6, R = 7, T = 8, U = 9$. Alice transmits the message in blocks. Assume that each block corresponds to two letters which are encoded into their numerical equivalents, e.g., NO becomes 56 and then it is enciphered by using RSA.

   (a) If Alice wants to send the text "TA", what ciphertext will be received by Bob?

   (b) If Bob receives the ciphertext "79", what was the original message transmitted by Alice?

   [5 + 5 = 10]

2. The Linear Cipher encodes each plain letter P to a cipher letter C using the following encryption function:

$$C = a * P + b \pmod{26}$$

where the encryption key consists of the pair of integers (a, b). Here, we say a the factor key and b the shift key. For example, if $a = 3$ and $b = 4$, the encryption function becomes $C = 3 * P + 4 \pmod{26}$.

Consider the encoding scheme as follows: $A = 0, B = 1, C = 2, \ldots, Z = 25$.

   (a) If the plaintext is RXTZ, find the ciphertext.

   (b) If the ciphertext is REPLY, find the respective plaintext.

   [5 + 5 = 10]

********** End of Question Paper **********

Research in Information Security (CS8.501)

Class Test 2 (Monsoon 2023)

International Institute of Information Technology, Hyderabad

Time: 70 Minutes Total Marks: 20

Sep 29, 2023

---

1. Assume that there are N security classes in a user hierarchy, which form a partial-ordered set (poset) $SC = \{SC_1, SC_2, \ldots, SC_N\}$. $H(\cdot)$ is a secure collision-resistant one-way hash function (for example, SHA-1 hash function), Ω a symmetric key cryptosystem (for example, AES symmetric-key block cipher), and $E_k(\cdot)/D_k(\cdot)$ the symmetric-key encryption/decryption using key k.

   Consider the following linear polynomial-based dynamic key management scheme for hierarchical access control with the following phases. The scheme makes use of the linear polynomials (of degree 1) instead of the polynomials of higher degree (> 1) for computational efficiency.

   - ***Relationship building phase:*** In this phase, CA first needs to build the hierarchical structure for controlling access according to the given relationships among the security classes in the hierarchy.

     Let $SC_i \in SC$ and $SC_j \in SC$ be two security classes such that the relationship $SC_j \leq SC_i$ hold, that is, SC_i have a higher security clearance than that for SC_j. A legitimate relationship $(SC_i, SC_j) \in R_{i,j}$ between SC_i and SC_j will exist in the hierarchy if SC_i can access SC_j.
   - ***Key generation phase:*** Once the relationship building phase is completed by the CA, the CA can execute the following steps for distributing the secret and public keys to security classes in the given hierarchy:

     Step 1. CA first selects a secure collision-resistant one-way hash function $H(\cdot)$ and then a finite field $GF(m)$, where m is either odd prime or prime power. CA also chooses a symmetric key cryptosystem Ω (for example, AES).

     Step 2. CA randomly selects its own secret key k_{CA}. After that CA needs to select randomly the secret key sk_i and sub-secret key d_i for each security class SC_i ($1 \leq i \leq N$) in the hierarchy.

     Step 3. Once Step 2 is completed, for each security class SC_i, CA computes the signature $Sign_i$ on sk_i as $Sign_i = H(ID_{CA}||sk_i)$, where ID_{CA} is the identity of the CA. The purpose of signature is used later by the CA and security classes for verification of the secret key sk_i of SC_i. CA declares them as public.

     Step 4. For each SC_i such that $(SC_i, SC_j) \in R_{i,j}$, CA then constructs the linear polynomials $f_{i,j}(x) = (x - H(ID_{CA}||Sign_j||d_i)) + sk_j \pmod m$, and declares them publicly.

     Step 5. Finally, CA sends d_i to SC_i via a secure channel.

   Answer the following questions to this scheme:

(a) Consider the following small user hierarchy provided in Figure 1. Compute all the public parameters corresponding to each security class in the scheme to be stored in public domain.

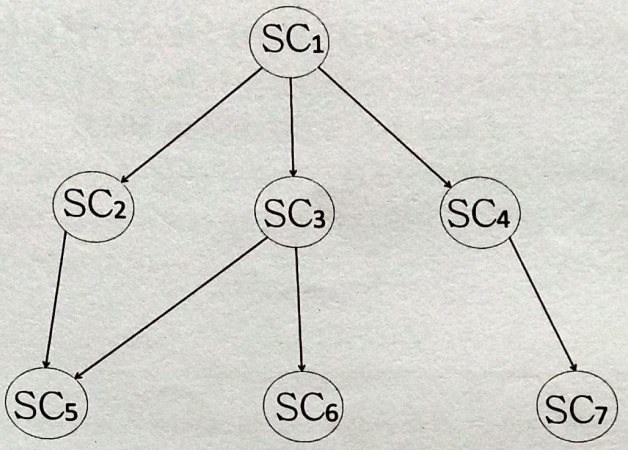

Figure 1: An example of a poset in a user hierarchy

(b) Design the key derivation phase related to this scheme in which a security class SC_i can derive the secret key sk_j of its successor SC_j with $(SC_i, SC_j) \in R_{i,j}$. Show with the designed key derivation phase, how the secret key sk_5 of security class SC_5 can be derived by its predecessor classes SC_1, SC_2 and SC_3.

[10 + 10 = 20]

Research in Information Security (CS8.501)

Class Test 3 (Monsoon 2023)

International Institute of Information Technology, Hyderabad

Time: 70 Minutes Total Marks: 20

Oct 17, 2023

---

1. Consider the polynomial-pool based key pre-distribution scheme in wireless sensor networks (WSNs) as follows.

   The key setup server (the base station, BS) selects a random pool $\mathcal{K}$ of s symmetric bivariate polynomials in $F_q[x, y]$ each of degree t in x and y, where F_q is a finite field of order q, q being a prime, and $F_q = Z_q = \{0, 1, 2, \ldots, q-1\}$. For example, $f(x, y) = 5x^7 + 3x^3y^3 + 2xy + 5y^7$ is a symmetric bivariate polynomial in F_7 of degree 7, because $f(x, y) = f(y, x)$.

   Some ids $id_{u_1}, id_{u_2}, \ldots, id_{u_n} \in F_q$ are generated for the sensor nodes u_i in the sensor network, where n is the network size. For each sensor node u to be deployed in the network, s' polynomials, say, $f_1(x, y)$, $f_2(x, y)$, $\ldots$, $f_{s'}(x, y)$ are randomly selected from $\mathcal{K}$ and the polynomial shares $f_1(id_u, y)$, $f_2(id_u, y)$, $\ldots$, $f_{s'}(id_u, y)$ are loaded in the key ring K_u of u. Thus, u is given the information $\{id_u, K_u\}$ prior to its deployment.

   Answer the following questions:
   (a) Design the direct key establish phase for the scheme.
   (b) Design the dynamic node addition phase for the scheme.
   (c) Derive the storage, communication and computational overheads required for a sensor node to pre-load the information in its memory prior to deployment and during the direct key establishment phase to establish a symmetric key with one of it neighbor nodes.
   (d) Derive the network connectivity of the scheme during the direct key establishment phase.
   (e) Finally, derive the resilience against sensor node capture attack of the scheme during the direct key establishment phase.

   [5 + 2 + 3 + 5 + 5 = 20]

Research in Information Security (CS8.501)

Class Test 4 (Monsoon 2023)
International Institute of Information Technology, Hyderabad
Time: 70 Minutes Total Marks: 20

Nov 7, 2023

---

1. Consider the following variant of user authentication scheme, which contains the following phases.

   • **Registration phase**
   In this phase, a legal user U_i registers or re-registers with the gateway node (GWN). In order to register to the GWN, the user U_i need to execute the following steps:

   Step 1. U_i first selects a unique identity ID_i and a chosen password PWD_i. After that U_i generates a random value r and computes $RPWD_i = h(r||PWD_i)$ and sends the registration request message $R = \langle ID_i, RPWD_i \rangle$ to the GWN via a secure channel.

   Step 2. After receiving the registration request message $R = \langle ID_i, RPWD_i \rangle$ from the user U_i, the GWN verifies ID_i and rejects this request if ID_i is invalid. Otherwise, the GWN computes the temporal credential $TCred_i = h\ (K_{GWN-U}||ID_i||TExp_i)$ and $PTCred_i = TCred_i \oplus RPWD_i$ for the user U_i, where $TExp_i$ is the expiration time of U_i's temporal credential. The GWN then initializes the temporary identity TID_i of the user U_i and stores the tuple $(TID_i, ID_i, TExp_i)$ in its verification table. The GWN issues a smart card containing to the information $\{h(\cdot), TID_i, TExp_i, PTCred_i\}$ to U_i and sends it via a secure channel.

   Step 3. After receiving the smart card, U_i stores the random number r in the smart card.

   The registration phase for all the deployed smart devices is as follows:

   Step 1. A smart device SN_j submits its identifier ID_{SN_j} to the GWN through a secure channel.

   Step 2. Upon receiving the message, the GWN computes the temporal credential for SN_j as $TC_j = h(K_{GWN-S}\ ||ID_{SN_j})$, where K_{GWN-S} is the GWN's private key only known to the GWN. The GWN then sends the message $\langle TC_j \rangle$ to SN_j.

   Step 3. When the smart device SN_j receives the message in Step 2, SN_j stores TC_j as its temporal credential.

   • **Login and authentication phase**
   In this phase, in order to login to the GWN and authenticate by the smart devices in WSN, the following steps are executed by the user U_i, the GWN and the login-smart device SN_j:

   Step 1. U_i first inserts his/her smart card to a terminal and then inputs his/her identity ID_i and password PWD_i. The smart card then generates a current timestamp TS_4 and also a random key Key_i. After that the smart card computes $TCred_i = PTCred_i \oplus h(r||PWD_i)$, $PKS_i = Key_i \oplus h(TCred_i\ ||TS_4)$, $C_i = h(ID_i||Key_i||TCred_i||TS_4)$. U_i then sends the login request message $\langle TID_i, C_i, PKS_i, TS_4 \rangle$ to the GWN via a public channel.

Research in Information Security (CSE540)

End Semester Examination (Monsoon 2023)
International Institute of Information Technology, Hyderabad
Time: 3 hours Total Marks: 60
Instructions: This is a closed books and notes examination.
Regular Calculator is allowed.

Answer ANY SIX questions from the following questions.

1. **(a) Define a multi-signature scheme. Explain a generalized multi-signature scheme with an example.**

(b) Consider the following bilinear pairing based proxy signature scheme as shown in Figure 1.

Assumption: CDHP is hard.

- Alice computes her public key $y_o = H(ID_o)$, where ID_o is her identity. Then, Alice obtains her private key $x_o \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_o)$ from KGC.
- Bob computes his public key $y_p = H(ID_p)$, where ID_p is his identity. Then, Bob obtains his private key $x_p \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_p)$ from KGC.
- Martin (a designated verifier) computes his public key $y_m = H(ID_m)$, where ID_m is his identity. Then, Martin obtains his private key $x_m \leftarrow \mathcal{KG}_{cdhp}(\textbf{params-cdhp}, y_m)$ from KGC.
- Delegation capability generation: Alice generates delegation capability σ_o as $\sigma_o = x_o H(\omega)$.
- Delegation capability verification: Bob accepts σ_o if and only if $\hat{e}(\sigma_o, P) = \hat{e}(H(\omega), y_o)$.
- Proxy key generation: Bob computes $\rho_p = \sigma_o + x_p H(\omega)$.
- Proxy signature generation: To generate a designated verifier proxy signature for Martin on message m, Bob does the following:
  - Picks $k_p \in \mathbb{Z}^*_{q-1}$ and computes $r_p = k_p y_m$.
  - Computes $\sigma_p = k_p(y_o + y_p) - H_2(m, r_p) \cdot \rho_p$, where $H_2 : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}^*_q$.
  - The proxy signature of message m is the tuple $(r_p, \sigma_p, (\omega, m))$.

Figure 1: Bilinear pairing based proxy signature scheme

Your task is to design the corresponding proxy signature verification phase. In this case, to verify the proxy signature of message m, the following verification equation needs to be satisfied:

$$\hat{e}(\sigma_p, P) \cdot \hat{e}(H(w), y_o + y_p)^{H_2(m, r_p)} = z.$$

Find the value of z. **[5 + 5 = 10]**

2. (a) Define an aggregate signature. Explain a generalized aggregate signature scheme with an example.

(b) Describe a generalized DLP-based proxy signature.

[5 + 5 = 10]

3. (a) A user hierarchy consists of a number n of disjoint security classes, say, $SC_1, SC_2, \ldots, SC_n$. Let this set be $SC = \{SC_1, SC_2, \ldots, SC_n\}$. A "binary relation $\geq$" is defined in SC as $SC_i \geq SC_j$, which means that the security class SC_i has a security clearance higher than or equal to the security class SC_j. Prove that the relation $\geq$ is a partial ordered relation.

(b) Consider the following hierarchical access control scheme for key management. It consists of three main phases, namely, the relationship building phase, the key generation phase, and the key derivation phase. In the relationship building phase, a central authority (CA) builds the hierarchical structure for controlling access according to the relationship between the nodes. In the key generation phase, CA chooses a pair of points and constructs the public polynomial using a one-way hash function. In the key derivation phase, the predecessor in the hierarchy can use its own secret key and the public information related to the successor(s) to derive the decryption key(s) for the successor(s) for accessing the authorized file(s). Further, it provides the solution of key management of dynamic access problems by means of providing inserting a new security class, removing an existing security class, creating a new relationship, revoking an existing relationship, and changing secret key of a security class in the hierarchy. We discuss the following phases.

Relationship building phase: In this phase, CA builds the hierarchical structure for controlling access according to the relationships among the nodes in the hierarchy. Let $U = \{SC_1, SC_2, \ldots, SC_n\}$ be a set of n security classes in the hierarchy. Assume that SC_i is a security class with higher clearance and SC_j a security class with lower clearance, that is, $SC_i \geq SC_j$. A legitimate relationship $(SC_i, SC_j) \in R_{i,j}$ between two security classes SC_i and SC_j exists in the hierarchy if SC_i can access SC_j.

Key generation phase: In this phase, CA performs the following steps:

Step 1: Selects randomly a large prime p, an elliptic curve $E_p(a, b)$ defined over Z_p such that the order of $E_p(a, b)$ lies in the interval $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$, and a one-way function $h(\cdot)$ to transform a point into a number and a base point G_j from $E_p(a, b)$, $1 \leq j \leq n$. Then, for each security class SC_j $(1 \leq j \leq n)$, selects a secret key sk_j and a sub-secret key s_j.

Step 2: For all $\{SC_i | (SC_i, SC_j) \in R_{i,j}\}$, computes $s_i G_j = (x_{j,i}, y_{j,i})$, and $h(x_{j,i} || y_{j,i})$, where $||$ is a bit concatenation operator.

Step 3: Finally, computes the public polynomial $f_j(x)$ using the values of $h(x_{j,i} || y_{j,i})$ as

$$f_j(x) = \prod_{SC_i > SC_j} (x - h(x_{j,i} || y_{j,i})) + sk_j \pmod p$$

Step 4: Sends sk_j and s_j to the security class SC_j via a secret channel, and announces $p, h(\cdot), G_j, f_j(x)$ as public.

Key derivation phase: In order to compute the secret keys sk_j of all successors, SC_j, the predecessor SC_i, for which the relationships $(SC_i, SC_j) \in R_{i,j}$ between SC_i and SC_j hold, proceeds as follows:

Step 1: For $\{SC_i | (SC_i, SC_j) \in R_{i,j}\}$, computes $s_i G_j = (x_{j,i}, y_{j,i})$, and $h(x_{j,i} || y_{j,i})$.

Step 2: Computes the secret key sk_j using $h(x_{j,i} || y_{j,i})$ as follows:

$$f_j(x) = \prod_{SC_i > SC_j} (x - h(x_{j,i} || y_{j,i})) + sk_j \pmod p,$$

$$f_j(h(x_{j,i} || y_{j,i})) = sk_j \pmod p.$$

Inserting new security classes phase: If a new security class SC_k is inserted into the hierarchy such that $SC_i \geq SC_k \geq SC_j$, then the relationships $(SC_i, SC_k) \in R_{i,k}$ for $SC_i \geq SC_k$ and $(SC_k, SC_j) \in R_{k,j}$ for $SC_k \geq SC_j$ need to be updated into the hierarchy. CA needs the following steps to manage the accessing priority of SC_k in the hierarchy.

Step 1: Updates the partial relationships R that follows when the security class SC_k joins the hierarchy, and randomly selects the secret key sk_k, the sub-secret key s_k and the base point G_k for the class SC_k.

Step 2: For all $\{SC_i | (SC_i, SC_k) \in R_{i,k}\}$ that satisfies $SC_i \geq SC_k$ when the new class SC_k is inserted in the hierarchy, computes $s_i G_k = (x_{k,i}, y_{k,i})$, and $h(x_{k,i} || y_{k,i})$.

Step 3: Computes the public polynomial $f_k(x)$ as follows:

$$f_k(x) = \prod_{SC_i > SC_k} (x - h(x_{k,i}||y_{k,i})) + sk_k \pmod p$$

Step 4: For all $\{SC_i | (SC_i, SC_k) \in R_{i,k}\}$ and $\{SC_j | (SC_k, SC_j) \in R_{k,j}\}$ that satisfy $SC_i \geq SC_k \geq SC_j$ when the new class SC_k is inserted in the hierarchy, computes $s_k G_j = (x_{j,k}, y_{j,k})$, $s_i G_j = (x_{j,i}, y_{j,i})$, $h(x_{j,k}||y_{j,k})$ and $h(x_{j,i}||y_{j,i})$.

Step 5: Computes the public polynomial $f'_j(x)$ as follows:

$$f'_j(x) = \prod_{SC_i > SC_k > SC_j} (x - h(x_{j,i}||y_{j,i}))(x - h(x_{j,k}||y_{j,k})) + sk_j \pmod p$$

Step 6: Replaces $f_j(x)$ with $f'_j(x)$, and sends sk_k and s_k to SC_k via a secure channel, and announces publicly G_k, $f_k(x)$ and $f'_j(x)$.

With respect to the above scheme, show that it is vulnerable to exterior root finding attack. Explain this attack with a small poset.

[4 + (3 + 3) = 10]

4. (a) Discuss the taxonomy of security protocols in Internet of Things (IoT).

   (b) Consider the path key establishment phase of the basic random key distribution scheme (EG scheme) in wireless sensor networks (WSNs). With respect to this, derive the network connectivity and resilience against sensor node capture attack.

   [4 + (3 + 3) = 10]

5. Consider the following authentication scheme. Table 1 lists various symbols and their meanings that are used in the proposed scheme (BSKMP-ICPS) for Industrial Cyber-Physical Systems (ICPS). Various phases related to BSKMP-ICPS are discussed below.

Table 1: Notations and their meanings

Symbol	Meaning
TA	Trusted authority
SD_i, ID_{SD_i}	i^{th} IoT smart device and its identity
GWN_j, ID_{GWN_j}	j^{th} gateway node and its identity
TID_{SD_i}	Temporary identity of each SD_i
RID_X	Pseudo identity of an entity X
FS_k, ID_{FS_k}	k^{th} fog server and its identity
CS_l, ID_{CS_l}	l^{th} cloud server and its identity
(pr_X, Pub_X)	Private-public keys pair of an entity X
$h(\cdot)$	A collision-resistant cryptographic one-way hash function
$E(\cdot)/D(\cdot)$	Public key encryption/decryption function
q	A large prime number
$GF(q)$	Galois (finite) field over large prime q
$E_q(\alpha, \beta)$	A non-singular elliptic curve: $y^2 = x^3 + \alpha x + \beta \pmod q$ over $GF(q)$ with $\alpha, \beta \in Z_q = \{0, 1, \cdots, q-1\}$
G	A base point G in $E_q(\alpha, \beta)$
$P + Q$	Elliptic curve point addition of two points $P, Q \in E_q(\alpha, \beta)$,
$k \cdot G$	Elliptic curve point multiplication; $k \cdot G = G + G + \cdots G$ (k times), $G \in E_q(\alpha, \beta)$, $k \in Z_q^*$
ECDSA	Elliptic curve digital signature algorithm
TS_x	Current system timestamp generated by an entity X
ΔT	Maximum transmission delay associated with a message
$x * y$	Modular multiplication of elements $x, y \in Z_q$
$\|, \oplus$	Data concatenation & exclusive-OR operators, respectively

Setup Phase: The TA being the trusted entity in the network is responsible for selecting a "non-singular elliptic curve $E_q(\alpha, \beta)$ of the form: $y^2 = x^3 + \alpha x + \beta \pmod q$" over a Galois (finite) field $GF(q)$ with constants $\alpha, \beta \in Z_q = \{0, 1, \cdots, q-1\}$ having $4\alpha^3 + 27\beta^2 \neq 0 \pmod q$, where q is sufficiently large prime number so that the "Elliptic Curve Discrete Logarithm Problem (ECDLP)" turns out to be intractable. The TA then picks a "base point $G \in E_q(\alpha, \beta)$ whose order is as large as q", "ECDSA signature generation and verification algorithms", and also a "collision-resistant cryptographic one-way hash function" (for instance, we use Secure Hash Algorithms (SHA-256) as hash function).

Next, the TA picks its own private-public keys pair (pr_{TA}, Pub_{TA}), where $pr_{TA} \in Z_q^* = \{1, 2, \cdots, q-1\}$ is randomly generated and $Pub_{TA} = pr_{TA} \cdot G$. It is assumed that all the fog servers FS_k $(k = 1, 2, \cdots, n_{fs})$

and cloud servers CS_l ($l = 1, 2, \cdots, n_{cs}$) are registered by the TA. After that each FS_k will generate its own private-public keys pair ($pr_{FS_k} \in Z_q^*$, $Pub_{FS_k} = pr_{FS_k} \cdot G$), and also each CS_l will generate its own private-public keys pair ($pr_{CS_l} \in Z_q^*$, $Pub_{CS_l} = pr_{CS_l} \cdot G$). FS_k and CS_l declare their public keys Pub_{FS_k} and Pub_{CS_l} as public, whereas the TA also declares its own public key Pub_{TA} as public. However, the private key of each entity is kept secret to that entity only.

Gateway Node Enrollment: For each gateway node GWN_j under a group Gr_j, the TA does the following:

Step GNE1. The TA first selects a unique identity ID_{GWN_j} and its corresponding pseudo identity as $RID_{GWN_j} = h(ID_{GWN_j} || pr_{TA})$, and a temporal credential as $TC_{GWN_j} = h(ID_{GWN_j} || RTS_{GWN_j} || pr_{TA})$ where RTS_{GWN_j} is the GWN_j's registration timestamp. Next, the TA generates a unique "t-degree bivariate symmetric polynomial of the form: $Poly_j(x, y) = \sum_{u=0}^{t} \sum_{v=0}^{t} a_{u,v} x^u y^v \pmod q$" where the co-efficients $a_{u,v} \in Z_q$ are chosen randomly, $Poly_j(x, y) = Poly_j(y, x)$ and $t >> n_{sd}$ in each Gr_j so that the proposed key management phase becomes "unconditionally secure and t-collision resistant" against IoT smart device physical capture attacks. Next, the TA calculates the polynomial share $Poly_j(RID_{GWN_j}, y)$ and sends the credentials $\{RID_{GWN_j}, TC_{GWN_j}, Poly_j(RID_{GWN_j}, y), h(\cdot), E_q(\alpha, \beta), G\}$ securely to each GWN_j.

Step GNE2. After receiving the credentials from the TA, each GWN_j generates its own private-public keys pair ($pr_{GWN_j} \in Z_q^*$, $Pub_{GWN_j} = pr_{GWN_j} \cdot G$), saves pr_{GWN_j} and makes Pub_{GWN_j} as public. The TA also deletes the generated credentials ID_{GWN_j}, RID_{GWN_j} and TC_{GWN_j} in order to thwart against stolen verifier attack. Thus, the GWN stores $\{RID_{GWN_j}, TC_{GWN_j}, Poly_j(RID_{GWN_j}, y), h(\cdot), E_q(\alpha, \beta), G, pr_{GWN_j}\}$ in its secure database so that the stolen-verifier attack is prevented to launch other attacks including impersonation attacks.

IoT Smart Device Enrollment: For each smart device SD_i in each group Gr_j, the TA executes the following steps:

Step SDE1. For each SD_i, the TA picks a unique identity ID_{SD_i}, a unique random temporary identity TID_{SD_i}, a registration timestamp RTS_{SD_i} and a private-public keys pair ($pr_{SD_i} \in Z_q^*$, $Pub_{SD_i} = pr_{SD_i} \cdot G$), and computes the temporal credential as $TC_{SD_i} = h(ID_{SD_i} || RTS_{SD_i} || pr_{TA})$ and pseudo-identity $RID_{SD_i} = h(ID_{SD_i} || pr_{TA})$, where RTS_{SD_i} is the SD_i's registration timestamp.

Step SDE2. The TA also calculates the polynomial share $Poly_j(RID_{SD_i}, y)$ for each SD_i corresponding to Gr_j and pre-loads the credentials $\{(TID_{SD_i}, RID_{SD_i}), RID_{GWN_j}, TC_{SD_i}, Poly_j(RID_{SD_i}, y), (pr_{SD_i}, Pub_{SD_i}), h(\cdot), E_q(\alpha, \beta), G\}$ into the memory of SD_i prior to its deployment in Gr_j.

Step SDE3. The TA sends securely $\{(TID_{SD_i}, RID_{SD_i}) | i = 1, 2, \cdots, n_{sd}\}$ to GWN_j residing in its group Gr_j. The TA also deletes the information ID_{SD_i}, TID_{SD_i} and TC_{SD_i} to thwart against stolen verifier attack. Furthermore, the TA makes Pub_{SD_i} as public for each SD_i.

Key Management Phase: This phase permits a registered IoT smart device (SD_i) to establish a secret key with its associated gateway node GWN_j residing in a group Gr_j for a particular ICPS application. The following are the steps needed for key management between SD_i and GWN_j:

Step KM1. SD_i as the initiator creates a random secret $rn_{SD_i} \in Z_q^*$ and a current timestamp TS_{SDi1} to calculate $x_{SD_i} = h(RID_{SD_i} || rn_{SD_i} || TC_{SD_i} || TS_{SDi1})$ and $RN_{SD_i} = x_{SD_i} \cdot G$. Next, SD_i generates a new temporary identity $TID_{SD_i}^{new}$, evaluates its own polynomial share $Poly_j(RID_{SD_i}, y)$ at $y = RID_{GWN_j}$ to have the secret value $Poly_j(RID_{SD_i}, RID_{GWN_j})$, and computes $TID_{SD_i}^* = TID_{SD_i}^{new} \oplus h(Poly_j(RID_{SD_i}, RID_{GWN_j}) || RN_{SD_i} || TS_{SDi1})$ and signature on rn_{SD_i} as $Sig_{SD_i} = x_{SD_i} + h(TID_{SD_i} || TID_{SD_i}^{new} || RN_{SD_i} || Pub_{SD_i} || TS_{SDi1}) * pr_{SD_i} \pmod q$. After that SD_i sends a key management request message $Msg_1 = \{TID_{SD_i}, TID_{SD_i}^*, RN_{SD_i}, Sig_{SD_i}, TS_{SDi1}\}$ to GWN_j via public channel.

Step KM2. After receiving the message Msg_1 at time TS_{SDi1}^*, the timeliness of TS_{SDi1} is checked by GWN_j with the verifying condition: $|TS_{SDi1} - TS_{SDi1}^*| < \Delta T$, where the significance of ΔT is the "maximum transmission delay". If it is valid, GWN_j fetches RID_{SD_i} of SD_i corresponding to TID_{SD_i} from its secure database. GWN_j then calculates $Poly_j(RID_{GWN_j}, RID_{SD_i})$ using its own RID_{GWN_j}, which is essential same as $Poly_j(RID_{SD_i}, RID_{GWN_j})$ because the polynomial $Poly_j(x, y)$ is symmetric, and $TID_{SD_i}^{new} = TID_{SD_i}^* \oplus h(Poly_j(RID_{GWN_j}, RID_{SD_i}) || RN_{SD_i} || TS_{SDi1})$. Next, GWN_j verifies the signature Sig_{SD_i} by the condition: $Sig_{SD_i} \cdot G = RN_{SD_i} + h(TID_{SD_i} || TID_{SD_i}^{new} || RN_{SD_i} || Pub_{SD_i} || TS_{SDi1}) \cdot Pub_{SD_i}$. If the signature is valid, the next step is executed; otherwise, the phase is immediately terminated by GWN_j.

Step KM3. GWN_j creates current timestamp TS_{GWNj} along with random secret $rn_{GWN_j} \in Z_q^*$ in order to compute $y_{GWN_j} = h(rn_{GWN_j} || TC_{GWN_j} || TS_{GWNj})$, $RN_{GWN_j} = y_{GWN_j} \cdot G$, the Diffie-Hellman type secret key $DHK_{GWN_j, SD_i} = y_{GWN_j} \cdot RN_{SD_i}$, secret key shared with SD_i as

$SK_{GWN_j,SD_i} = h(Poly_j(RID_{GWN_j}, RID_{SD_i}) \,||DHK_{GWN_j,SD_i} \,||Sig_{SD_i})$ and also signature on rn_{GWN_j} and SK_{GWN_j,SD_i} as $Sig_{GWN_j} = y_{GWN_j} + h(RID_{GWN_j} \,||Pub_{GWN_j} \,||TS_{GWNj} \,||SK_{GWN_j,SD_i}) * pr_{GWN_j} \pmod q$. After these calculations, GWN_j sends the key management response message $Msg_2 = \{RN_{GWN_j}, Sig_{GWN_j}, TS_{GWNj}\}$ to SD_i via open channel. It is worth noticing that we need not to include GWN_j's pseudo-identity RID_{GWN_j} because GWN_j has prior knowledge about its all IoT smart devices deployed in Gr_j.

Step KM4. After reception of Msg_2 at time TS^*_{GWNj}, SD_i checks the timeliness of the TS_{GWNj} by the verifying condition: $|TS_{GWNj} - TS^*_{GWNj}| < \Delta T$. If it passes, SD_i will proceed to calculate the Diffie-Hellman type key $DHK_{SD_i,GWN_j} = x_{SD_i} \cdot RN_{GWN_j}$, which is same as $DHK_{GWN_j,SD_i} = y_{GWN_j} \cdot RN_{SD_i}$, that is, $DHK_{SD_i,GWN_j} = DHK_{GWN_j,SD_i}$ and session key shared with GWN_j as $SK_{SD_i,GWN_j} = h(Poly_j(RID_{SD_i}, RID_{GWN_j}) \,||DHK_{SD_i,GWN_j} \,||Sig_{SD_i})$. Next, SD_i verifies the validity of received signature Sig_{GWN_j} as $Sig_{GWN_j} \cdot G = RN_{GWN_j} + h(RID_{GWN_j} \,||Pub_{GWN_j} \,||TS_{GWNj} \,||SK_{SD_i,GWN_j}) \cdot Pub_{GWN_j}$. If the signature is valid, SD_i authenticates GWN_j. At the same time, computed session key SK_{SD_i,GWN_j} $(= SK_{GWN_j,SD_i})$ is considered as valid one and SD_i proceeds for the next step.

Step KM5. SD_i generates a current timestamp TS_{SDi2} and calculates the session key verifier $SKV_{SD_i,GWN_j} = h(SK_{SD_i,GWN_j} \,||TS_{SDi2})$. SD_i then sends the final message as the acknowledgment $Msg_3 = \{SKV_{SD_i,GWN_j}, TS_{SDi2}\}$ to GWN_j via open channel, and updates its old TID_{SD_i} with $TID^{new}_{SD_i}$ in its memory, which was calculated in Step KM1.

Step KM6. After receiving the message Msg_3 at time TS^*_{SDi2}, the timeliness of TS_{SDi2} is checked by GWN_j with the verifying condition: $|TS_{SDi2} - TS^*_{SDi2}| < \Delta T$. If the timestamp is valid, GWN_j checks the verifying condition: $SKV_{SD_i,GWN_j} = h(SK_{GWN_j,SD_i} \,||TS_{SDi2})$ with the help of its previously calculated session key SK_{GWN_j,SD_i}. If the condition is satisfied, GWN_j authenticates SD_i as valid entity and also considered SK_{GWN_j,SD_i} as authentic. GWN_j updates the old TID_{SD_i} of SD_i with new calculated $TID^{new}_{SD_i}$ in Step KM2 in its secure database.

Finally, the overall key management phase is briefed in Fig. 2.

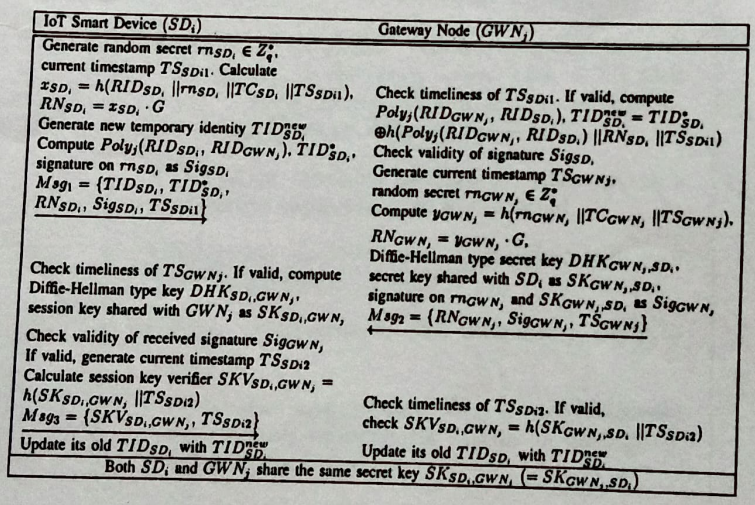

Figure 2: Overview of key management phase

With respect to the above scheme, answer the following:

(a) Show the correctness proof of the establishment session key SK_{GWN_j,SD_i} by both gateway node GWN_j and its IoT device SD_i.

(b) Show that the proposed scheme is secure against IoT device impersonation attack.

[5 + 5 = 10]

6. **(a)** Explain the layer-wise attacks on wireless sensor network (WSN) stack.

(b) Explain the tasks related to an access control scheme in IoT. In an access control scheme, a threat model usually involves the Dolev-Yao (DY) threat model as well as CK-adversary model. Discuss briefly the differences between these two threat models.

[5 + 5 = 10]

7. **(a)** Explain the role of blockchain-based AI/ML techniques for Big data analytics with a diagram.

(b) Explain the need for secure data delivery and collection in an Internet of Drones (IoD) environment. In such scenario, the blockchain technology can be useful. If we use the private blockchain, what will be the structure of a block in such private blockchain case?

[4 + (3 + 3) = 10]

8. **(a)** Explain a confusion matrix. With respect to it, define accuracy, recall, precision and F1-score.

(b) Explain various attacks related to AI/ML.

(c) Discuss the Ripple Protocol Consensus Algorithm (RPCA) used for mining the blocks in a blockchain network.

[3 + 3 + 4 = 10]

********** End of Question Paper **********

Research in Information Security (CSE540)

Mid Semester Examination (Monsoon 2022)

International Institute of Information Technology, Hyderabad

Time: 1 hour and 30 minutes Total Marks: 40

Instructions: Answer ALL questions. This is a closed book and notes examination.
Calculator is allowed.
You may attempt the BONUS question too.

1. **(a)** Verify whether the elliptic curve of the form: $E_{19}(-1, 6)$ is singular or non-singular.

   (b) Define the formally the elliptic curve discrete logarithm problem (ECDLP). What is the complexity required to solve ECDLP?

   (c) What are the differences among "multi-signature", "proxy signature", "blind signature" and "ring signature"? [1 + 3 + 4 = 8]

2. **(a)** Write the security properties of proxy signature.

   (b) Describe a generalized DLP-based proxy signature model, which is based on the discrete logarithm problem (DLP).

   (c) Consider the following proxy signature scheme based on the bilinear pairings:

   - Step 1. Alice as an original signer computes her public key $y_o = H(ID_o)$, where ID_o is her identity. Then, Alice generates her private key $x_o \leftarrow KG_{cdhp}(params - cdhp, y_o)$. Here, $H(\cdot)$ represents map-to-point hash function which will map a string to an elliptic curve point.
   - Step 2. Bob as a proxy signer computes his public key $y_p = H(ID_p)$, where ID_p is his identity. Then, Bob generates his private key $x_p \leftarrow KG_{cdhp}(params - cdhp, y_p)$.
   - Step 3. **Delegation capability generation:** Alice computes $\sigma_o = (s_o + b_o H'(w, y_o, y_p)$, and $\psi_o = b_o P$. Here, b_o is secret to Alice only, $H' : \{0,1\}^* \times G_1 \times G_1 \rightarrow G_1$, P is a base point in elliptic curve group G_1, and w is a warrant. Suppose G_1 is an additive cyclic group generated by P of prime order q, and G_2 is a multiplicative cyclic group of the same order. We define the map $e : G_1 \times G_1 \rightarrow G_2$ as a bilinear map.
   - Step 4. **Delegation capability verification**: Bob accepts σ_o if and only if

     $$e(s_o, P) = e(\psi_o, H'(w, y_o, y_p)) \cdot e(y_o, Reg_o),$$

     where $Reg_o = sb_oP$, registration token published by the KGC, s is the KGC's master key and $Pub_{KGC} = sP$ is the corresponding public key of the KGC.
   - Step 5. **Proxy key generation**: Bob computes $\rho_p = s_o + s_p + b_p H'(w, y_o, y_p)$. Here, b_p is secret to the proxy signer only.
   - Step 6. **Proxy signature generation**: To sign a message m, Bob does the following.
     - Selects a random $r \in Z_q^*$ and computes $R = rP$.
     - Computes $a = h(m, R, y_p)$ and $\psi_p = b_p P$, where $h : \{0,1\}^* \times G_1 \times G_1 \rightarrow \{0,1\}^*$ is a hash fuunction.
     - Computes $\sigma_p = (r+a)^{-1}\rho_p$. The proxy signature of message m is $(w, m, R, \sigma_p, \psi_o, \psi_p, y_o, y_p)$.
   - Step 7. **Proxy signature verification**: The proxy signature is valid if and only if

     $$e((R + h(m, R, y_p)P, \sigma_p) = e(\psi_o + \psi_p, H'(w, y_o, y_p)) \cdot e(y_o, Reg_o) \cdot e(y_p, Reg_p).$$

With respect to the above scheme, answer the following:

(i) Show the correctness proof of the **delegation capability verification** in Step 4.

(ii) Show the correctness proof of the **proxy signature verification** in Step 7.

[2 + 3 + (3 + 4) = 12]

3. (a) Explain wormhole and sinkhole attacks with examples in the context of wireless sensor networks (WSNs).

   (b) Consider the random key distribution scheme [EG scheme] (L. Eschenauer and V. D. Gligor, "A Key Management Scheme for Distributed Sensor Networks," in 9th ACM Conference on Computer and Communication Security, pages 41-47, November 2002).
   Answer the following two parts associated with the EG scheme:
   (i) Derive the network connectivity of the EG scheme during the path key establishment phase.
   (ii) Derive the resilience against sensor node capture attack during the direct key establishment phase.

   [(2 + 2) + (4 + 2) = 10]

4. (a) Explain the location-aware closest polynomials pre-distribution scheme (CPPS) in wireless sensor networks (WSNs) by explaining the pre-deployment and direct key establishment phases. [Donggang Liu, Peng Ning, "Improving key predistribution with deployment knowledge in static sensor networks," in ACM Transactions on Sensor Networks, vol. 1, no. 2, pp. 204-239, 2005]

   (b) Derive the resilience against node capture attack probability for the enhanced CPKS (ECPKS) scheme that applies the key prioritization technique using both pre-deployment and post-deployment locations of the deployed sensor nodes in WSNs.

   [5 + 5 = 10]

5. **[BONUS Question:]** Consider the following variant of random key distribution in wireless sensor networks, known as *polynomial-pool based key pre-distribution* scheme:

   Let $F_q = GF(q)$ be a finite field with a q (either a prime or 2^m for some positive integer m) just big enough to accommodate a symmetric cryptographic key. Let $f(x, y) \in F_q[x, y]$ be a t-degree symmetric bivariate polynomial i.e., $f(x, y) = f(y, x)$. The coefficients of the polynomial $f(x, y)$ are chosen from the finite field F_q. A *polynomial share* of $f(x, y)$ is a univariate polynomial $f(u, y)$ for some $u \in F_q$. We have, $f(u, v) = f(v, u)$. Thus, if two shares $f(u, y)$ and $f(v, y)$ of the same polynomial $f(x, y)$ are given to two nodes, say, u and v, they can come up with the common value $f(u, v) \in F_q$ as a shared key between them.

   The key setup server selects a random pool $\mathcal{K}$ of s symmetric bivariate polynomials in $F_q[x, y]$ each of degree t in x and y. Some ids $u_1, u_2, \ldots, u_n \in F_q$ are also generated for the sensor nodes in the network, where n is the network size. For each sensor node u to be deployed in the network, s' polynomials, say, $f_1(x, y), f_2(x, y), \ldots, f_{s'}(x, y)$ are randomly selected from $\mathcal{K}$ and the polynomial shares $f_1(u, y), f_2(u, y), \ldots, f_{s'}(u, y)$ are loaded in the key ring K_u of u. Immediately after deployment, each sensor u transmits the ids of the polynomial shares residing in its key ring. Two physical neighbors u and v having shares of some common polynomial(s) can establish a pairwise key between them. For instance, if f_i is the common polynomial share between u and v, they establish pairwise secret key as $k_{u,v} = f_i(u, v)$ (by node u) and $f_i(v, u)$ $(= f_i(u, v) = k_{u,v})$ (by node v).

   Answer the following questions about the above scheme:

   (i) Compute the network connectivity probability of establishing a secret pairwise key between two neighbor sensor nodes u and v.

   (ii) Derive the resilience against physical sensor node capture attack probability when some sensor nodes are physically compromised by an attacker.

   (iii) What is about the scalability?

   [4 + 4 + 2 = 10]

********** End of Question Paper **********

Research in Information Security (CSE540)

End Semester Examination (Monsoon 2022)

International Institute of Information Technology, Hyderabad

Time: 3 hours Total Marks: 60

Instructions: Answer *ALL* questions. This is a closed book and notes examination. Calculator is allowed.

1. Consider the following hierarchical access control scheme.

We assume that there are N security classes in the hierarchy which form a set $SC = \{SC_1, SC_2, ..., SC_N\}$. We use the following notations for describing our scheme. $H(\cdot)$ is a secure one-way hash function (for example, SHA-1 hash function, Ω a symmetric key cryptosystem (for example, AES symmetric-key block cipher, $E_k(\cdot)/D_k(\cdot)$ the symmetric-key encryption/decryption using key k, ID_{CA} the identity of the certificate (central) authority (CA), and $||$ the bit concatenation operator.

The proposed scheme consists the following three phases, namely the relationship building phase, key generation phase, and key derivation phase.

Relationship Building Phase: CA builds the hierarchical structure for controlling access according to the given relationships among the security classes in the hierarchy. Assume that $SC_i \in SC$ and $SC_j \in SC$ be two security classes such that $SC_j \leq SC_i$, that is, SC_i has a higher security clearance than that for SC_j. We say that a legitimate relationship $(SC_i, SC_j) \in R_{i,j}$ between SC_i and SC_j exists if SC_i can access SC_j.

Key Generation Phase: CA executes the following steps in order to complete this phase:

Step 1. CA chooses a secure hash function $H(\cdot)$, a finite field $GF(m)$ with m is either odd prime or prime power, and a symmetric key cryptosystem Ω.

Step 2. CA randomly selects its own secret key k_{CA}. CA then selects randomly the secret key sk_i and sub-secret key d_i for each security class SC_i $(1 \leq i \leq N)$ in the hierarchy.

Step 3. For each security class SC_i, CA computes the signature $Sign_i$ on sk_i as $Sign_i = H(ID_{CA}||sk_i)$ for the purpose of signature verification of the secret key sk_i. CA then publicly declares them.

Step 4. For each SC_i such that $(SC_i, SC_j) \in R_{i,j}$, CA constructs the linear polynomials $f_{i,j}(x) = (x - H(ID_{CA}||Sign_j||d_i)) + sk_j \pmod m$, and declares them publicly.

Step 5. Finally, CA sends d_i to SC_i via a secure channel.

At the end of this phase, CA encrypts d_i of SC_i as $S_i = E_{k_{CA}}(d_i)$, computes its signature Sd_i as $Sd_i = H(ID_{CA}||d_i)$ for the signature verification of d_i and stores the pair (S_i, Sd_i) in the public domain. CA then deletes all the secret keys sk_i and d_i. Note that whenever CA wants to update the secret keys sk_i's, CA first obtains d_i's from public parameters S_i's by decrypting them with its secret key k_{CA} and then verifies signatures by calculating the hash values as $Sd_i' = H(ID_{CA}||d_i)$, and checks if $Sd_i' = Sd_i$. If it matches, CA confirms that derived secret key d_i is legitimate.

Key Derivation Phase: If the security class SC_i wants to derive the secret key sk_j of its successor SC_j with $(SC_i, SC_j) \in R_{i,j}$, SC_i needs to proceed the following steps:

Step 1. SC_i first computes the hash value $H(ID_{CA}||Sign_j||d_i)$ using its own sub-secret key d_i, signature $Sign_j$ and ID_{CA} publicly available in the public domain.

Step 2. SC_i obtains secret key sk_j of SC_j's (including SC_i) as $sk_j = f_{i,j}(H(ID_{CA}||Sign_j||d_i))$. CA then verifies signature of sk_j as follows. CA computes $Sign_j^{'} = H(ID_{CA}||sk_j)$ and checks if $Sign_j^{'} = Sign_j$. If it holds, SC_i assures that the derived secret key sk_j is correct.

Based on the above scheme, explain the following two dynamic key management problems:

(a) Suppose a security class SC_l with $SC_j \leq SC_l \leq SC_i$ be added into the hierarchy. Explain the "Adding a New Security Class" phase in this context.

(b) Suppose the security class SC_l with $SC_j \leq SC_l \leq SC_i$ be removed from the hierarchy. Explain the "Deleting an Existing Security Class" phase so that the CA needs to remove SC_l so that the forward security is preserved.

[5 + 5 = 10]

2. **(a)** Explain with an architecture for Renewable Energy-Based Smart Grid for IoT applicaions.

(b) Explain the Fuzzy Extractor technique for biometric verification and its connection with a user authentication scheme in IoT applications.

(c) What is the significance of IoT smart device capture attack in connection with user authentication scheme in IoT applications?

[3 + 4 + 3 = 10]

3. **(a)** Explain the importance of the Dolev-Yao (DY) and Canetti and Krawczyk's adversary model (CK-adversary model) in an access control mechanism for IoT applications.

(b) Consider the following device access control scheme in IoT applications, which consists of the following phases.

Setup phase: The setup phase is executed by the "gateway node (GWN)" in order to select the system parameters with the following steps. It is worth noting that the GWN can be considered as a "Public Key Infrastructure (PKI)" in the proposed scheme (DACS-IoT).

S1. The GWN fixes a "collision-resistant one-way cryptographic hash function" $h(\cdot)$ defined by h: $\{0,1\}^* \rightarrow \{0,1\}^l$, which takes an arbitrary length input string and produces a fixed-length (l bits) hash output, called the message digest or hash value. $h(\cdot)$ can be considered as Secure Hash Standard (SHA-1) algorithm or it can be SHA-2 for more security achievement [**?**].

S2. Next, the GWN picks a "non-singular elliptic curve $E_p(a,b)$ of the form $y^2 = x^3 + ax + b \pmod p$ having a large prime p and two constants $a, b \in Z_p = \{0, 1, \ldots, p-1\}$ such that the necessary and sufficient condition $4a^3 + 27b^2 \neq 0 \pmod p$ is satisfied".

S3. The GWN then picks a base point G on $E_p(a,b)$ whose order is as large as p, say n such that $n.G = G + G + \cdots + G\ (n\ times) = \mathcal{O}$, where $\mathcal{O}$ is called the point at infinity or zero point.

S4. Finally, the GWN generates a public-private key pair (P_{gwn}, p_{gwn}) by selecting a private key p_{gwn} randomly & then calculating its correspondingly public key as $P_{gwn} = p_{gwn}.G$.

At the end of this phase, the GWN publishes the public parameters $\{h(\cdot), E_p(a,b), G, P_{gwn}\}$ and keeps its private key p_{gwn} as secret.

Device enrollment phase: In this phase, all the IoT sensing devices need to be registered in offline mode by the GWN before these are installed in the IoT environment. The following steps are essential to serve this purpose:

E1. For each IoT sensing device SD_i, the GWN fixes a unique ID ID_{dev_i} & generates a long-term random private key $k_{dev_i} \in Z_p^*$ to calculate the corresponding long-term public key $K_{dev_i} = k_{dev_i}.G$, where $Z_p^* = \{a \mid 0 < a < p, \gcd(a,p) = 1\} = \{1, 2, \ldots, p-1\}$.

E2. For each IoT sensing device SD_i, the GWN also picks a certificate random private key $ck_{dev_i} \in Z_p^*$ to calculate the corresponding certificate public key $CK_{dev_i} = ck_{dev_i}.G$. The GWN makes CK_{dev_i} as public.

E3. The GWN then creates a certificate $cert_{dev_i}$ for every sensing device SD_i as $cert_{dev_i} = p_{gwn}.h(ID_{gwn}||CK_{dev_i}) + ck_{dev_i} \pmod p$ using its own private key p_{gwn}. It is worth noting that since the certificate $cert_{dev_i}$ contains p_{gwn}, it is only created by the GWN.

E4. Finally, the GWN pre-loads the materials $\{ID_{dev_i}, ID_{gwn}, k_{dev_i}, K_{dev_i}, CK_{dev_i}, h(\cdot), E_p(a,b), G, cert_{dev_i}, P_{gwn}\}$ in the memory of each SD_i. The GWN erases the certificate random private key ck_{dev_i} from its database for security reasons.

Device access control phase: For secure communication between two neighbor IoT sensing devices SD_i and SD_j, node authentication needs to be successful before establishment of the secret pairwise key between them. This is achieved by executing the following steps:

A1. $SD_i \rightarrow SD_j$: $msg_1 = \{cert_{dev_i}, sig_{dev_i}, TS_i, R_i, K_{dev_i}\}$

A1.1. SD_i first generates the current timestamp TS_i and a random secret $r_i \in Z_p^*$ to calculate the corresponding public $R_i = h(r_i||\ ID_{dev_i}||\ k_{dev_i}||\ TS_i).G$ using its own stored private key k_{dev_i} and identity ID_{dev_i}.

A1.2. SD_i calculates the signature sig_{dev_i} on r_i and ID_{dev_i} as

$$\begin{aligned} sig_{dev_i} &= h(r_i||ID_{dev_i}||k_{dev_i}||TS_i) \\ &\quad + k_{dev_i}.h(cert_{dev_i}||R_i||K_{dev_i}||ID_{gwn}||TS_i) \pmod p. \end{aligned}$$

A1.3. SD_i then dispatches the message authentication request message $msg_1 = \{cert_{dev_i}, sig_{dev_i}, TS_i, R_i, K_{dev_i}\}$ to its neighbor SD_j over open channel.

A2. $SD_j \rightarrow SD_i$: $msg_2 = \{cert_{dev_j}, sig_{dev_j}, TS_j, R_j, K_{dev_j}, SKV_{ij}\}$

After receiving the msg_1 from SD_i, SD_j performs the following sequence of verification steps:

A2.1. SD_j firstly checks if the verification condition $|TS_i - TS_{current}| < \Delta T$ holds or not, where $TS_{current}$ is the time when the message was received by SD_j and ΔT is the "maximum allowable transmission delay". If this verification does not hold, SD_j discards the message and discontinues the phase promptly. Once this condition is satisfied, the certificate $cert_{dev_i}$ verification is done by checking the condition:

$$cert_{dev_i}.G = h(ID_{gwn}||CK_{dev_i}).P_{gwn} + CK_{dev_i}.$$

Note that

$$\begin{aligned} cert_{dev_i}.G &= h(ID_{gwn}||CK_{dev_i})(p_{gwn}.G) + ck_{dev_i}.G \\ &= h(ID_{gwn}||CK_{dev_i}).P_{gwn} + CK_{dev_i}. \end{aligned}$$

After the successful certificate verification, the signature verification is done by SD_j to check if the condition $sig_{dev_i}.G = R_i + h(cert_{dev_i}||\ R_i||\ K_{dev_i}||\ ID_{gwn}||\ TS_i).K_{dev_i}$ holds true. If the condition fails, the phase is discontinued promptly. It is also worth noting that

$$\begin{aligned} sig_{dev_i}.G &= h(r_i||ID_{dev_i}||k_{dev_i}||TS_i).G \\ &\quad + (k_{dev_i}.G)h(cert_{dev_i}||R_i||K_{dev_i}||ID_{gwn}||TS_i) \\ &= R_i + h(cert_{dev_i}||R_i||K_{dev_i}||ID_{gwn}||TS_i).K_{dev_i}. \end{aligned}$$

If the signature validation fails, the phase is also discontinued promptly.

A2.2. SD_j now generates a random secret $r_j \in Z_p^*$ and the current timestamp TS_j, and calculates $R_j = h(r_j||\ ID_{dev_j}||\ k_{dev_j}||\ TS_j).G$. After that SD_j calculates the signature

$$\begin{aligned} sig_{dev_j} &= h(r_j||ID_{dev_j}||k_{dev_j}||TS_j) \\ &\quad + k_{dev_j}.h(cert_{dev_j}||R_j||K_{dev_j}||ID_{gwn}||TS_j) \pmod p. \end{aligned}$$

Using the generated r_j, $cert_{dev_j}$ and TS_j, SD_j calculates the secret key shared with SD_i as

$$SK_{ij} = h(h(r_j||ID_{dev_j}||k_{dev_j}||TS_j).R_i||cert_{dev_i}||cert_{dev_j}||ID_{gwn})$$

and its verifier as $SKV_{ij} = h(SK_{ij}\ ||TS_j)$. Next, SD_j dispatches the authentication reply message $msg_2 = \{cert_{dev_j}, sig_{dev_j}, TS_j, R_j, K_{dev_j}, SKV_{ij}\}$ to SD_i over an open channel.

A3. After receiving the msg_2 from SD_j, SD_i performs the following sequence of verification stages to check the authenticity of the message msg_2 as well as SD_j.

A3.1. SD_i verifies the validity of $|TS_j - TS_{current}| < \Delta T$, where $TS_{current}$ represents the time when the message was received. Once this condition is fulfilled, the certificate $cert_{dev_j}$ verification is the next step by SD_i by checking the condition:

$$cert_{dev_j}.G = P_{gwn}.h(ID_{gwn}||CK_{dev_j}) + CK_{dev_j}.$$

After this certificate verification, the signature verification takes place using the condition $sig_{dev_j}.G = R_j + h(cert_{dev_j}||R_j||K_{dev_j}||ID_{gwn}||TS_j).K_{dev_j}$. If the condition fails, SD_i discontinues this phase promptly. Note that

$$\begin{aligned} sig_{dev_j}.G &= h(r_j||ID_{dev_j}||k_{dev_j}||TS_j).G \\ &\quad + (k_{dev_j}.G)h(cert_{dev_j}||R_j||K_{dev_j}||ID_{gwn}||TS_j) \\ &= R_j + h(cert_{dev_j}||R_j||K_{dev_j}||ID_{gwn}||TS_j).K_{dev_j}. \end{aligned}$$

Otherwise, the next step is executed by SD_i.

A3.2. SD_i calculates the secret key shared with SD_j as

$$SK'_{ij} = h(h(r_i||ID_{dev_i}||k_{dev_i}||TS_i).R_j||cert_{dev_i}||cert_{dev_j}||ID_{gwn})$$

and the secret key verifier as $SKV'_{ij} = h(SK'_{ij}||TS_j)$, and checks if the computed secret key verifier SKV'_{ij} matches with the received verifier SKV_{ij}. If the condition fails, SD_j is not authenticated by SD_i. Otherwise, SD_i treats SK'_{ij} $(= SK_{ij})$ as the valid secret key shared with SD_j. In a similar way, SD_j also keeps the secret key SK_{ij} shared with SD_i.

With respect to the above scheme, answer the following questions:

(i) Provide your argument behind the use of the certificate and the certificate key used in the generation of the certificate.

(ii) Show the correctness proof of the session key establishment by the two communicating IoT devices SD_i and SD_j.

[4 + (3+3) = 10]

4. (a) Explain the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm used in the mining of a block into the blockchain.

   (b) Construct a block structure used in a private blockchain with the use of Merkle tree root.

   [5 + 5 = 10]

5. (a) With respect to the scheme discussed in the class: "Blockchain-Envisioned Secure Data Delivery and Collection Scheme for 5G-Based IoT-Enabled Internet of Drones Environment," in *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 8, pp. 9097-9111, 2020, DOI: 10.1109/TVT.2020.3000576, explain the need for the hour for Secure Data Delivery and Collection in an Internet of Drones (IoD) environment.

   (b) With respect to the scheme discussed in the class: "Blockchain-Enabled Certificate-Based Authentication for Vehicle Accident Detection and Notification in Intelligent Transportation Systems," in *IEEE Sensors Journal*, Vol. 21, No. 14, pp. 15824-15838, July 2021, DOI: 10.1109/JSEN.2020.3009382, explain the overall process of Blockchain-enabled edge computing based Intelligent Transportation System (ITS) with a diagram.

   [4 + 6 = 10]

6. (a) Explain all possible attacks related to AI/ML security.

   (b) Explain briefly the impact on blockchain-based AI/ML-enabled Big data analytics in an IoT application.

   [5 + 5 = 10]

********** End of Question Paper **********