

Presented to:
Prof. Bala Prakasa Rao Killi

Dept. of Computer Science and Engineering
National Institute of Technology, Warangal



Authenticated Key Agreement Scheme With User Anonymity and Untraceability for 5G-Enabled Softwarized Industrial Cyber-Physical Systems

Presented by:
Mohammed Junaid Anwar 21CSB0B36
Ashish Upre 21CSB0B62
Ashish Vedula 21CSB0B63

Acknowledgement

Gratitude to the Authors

We extend our heartfelt gratitude to the authors of the foundational work:

- ❖ Anil Kumar Sutrala
- ❖ Mohammad S. Obaidat, Life Fellow, IEEE
- ❖ Sourav Saha, Student Member, IEEE
- ❖ Ashok Kumar Das, Senior Member, IEEE
- ❖ Mamoun Alazab, Senior Member, IEEE
- ❖ Youngho Park, Member, IEEE

Their pioneering contributions have enabled valuable advancements in secure communication for 5G-enabled industrial cyber-physical systems, addressing critical challenges in information and communication technology.

Overview of CPS and Security Challenges

Growth and Applications of CPS

- ❖ The rise of Information and Communications Technology (ICT) has expanded the reach of Cyber Physical Systems (CPS).
- ❖ CPS now impacts fields like:
 - ❖ Smart grids, smart cities
 - ❖ Transportation, public safety, healthcare
 - ❖ Industrial manufacturing, and more

Security Concerns

- ❖ **5G and SDN Integration:** Communication via public channels within industrial CPS (ICPS) through 5G and Software-Defined Networking (SDN).
- ❖ **Security Threats:** Increased risk of potential security threats and attacks within ICPS environments.

Proposed Solution: UAKA-5GSICPS Scheme

Objective

- Introduces the **UAKA-5GSICPS** scheme, a three-factor user authentication and key agreement protocol.
- Specifically designed for 5G-enabled SDN-based ICPS environments.

Functionality

- Ensures **mutual authentication** between authorized users and IoT-based smart devices.
- Authentication process is mediated by the SDN controller node for secure real-time data access.

Motivation

Challenges in Securing ICPS

- ❖ **Modest Resource Limitations:** Traditional security paradigms are unsuitable for Industrial Cyber-Physical Systems (ICPS), where sensors and actuators operate with limited resources.
- ❖ **Heterogeneous Communication:** Ensuring secure communication with session key establishment between diverse, registered users and devices is a significant challenge.
- ❖ **Scalability Concerns:** Security measures must avoid becoming bottlenecks as ICPS devices scale up extensively.

Objective of the Work

- ❖ Proposes an efficient user authentication and session key establishment scheme for secure communication between ICPS users and devices.

Software-Defined/Softwarized Networking (SDN)

Key Concepts of SDN

❖ Control and Data Plane Separation:

- ❖ Control Plane: Manages control flow and network topology.
- ❖ Data Plane: Forwards packets based on routes determined by the control plane.

❖ Operational Advantages:

- ❖ Efficient error recovery, configuration backup, and operational ease.
- ❖ Ideal for large-scale deployments.

Protocols and Implementations

❖ **OpenFlow:** Standard defined by the ONF, facilitating communication between SDN controllers and switches.

❖ **OVS and OF-CONFIG:** Open vSwitch (OVS) is a popular implementation supporting additional protocols like OF-CONFIG.

❖ **Industry Adoption:** Companies like Google, Cisco, and IBM use SDN for improved data center efficiency.

5G-Enabled Infrastructure and Its Capabilities

Key Features of 5G

- ❖ **High-Speed & Low Latency:**
 - ❖ Data rate up to 10 Gbps and latency reduce to one-thousandth of a second
- ❖ **Advanced Technologies:**
 - ❖ Small cells, Massive MIMO, mmWave, and Li-Fi, 100 billion device support

Applications and Use Cases

- ❖ **Fixed Wireless Access:** Wireless broadband for home internet
- ❖ **Industrial IoT & ICPS:** Smart cities, asset tracking, utilities, agriculture, and more

Network Slicing

- ❖ Enables creating multiple virtual networks from one physical network.
- ❖ Supported by technologies like NFV, MEC, and SDN for managing high-traffic environments.

Network Model of UAKA-5GSICPS

Components

- ❖ **Smart Devices:** n_{sd} IoT-enabled smart devices $\{SD_j \mid j = 1, 2, 3, \dots, n_{sd}\}$ connected to the data plane switches.
- ❖ **Data Plane:** Switches responsible for forwarding data packets to connected devices.
- ❖ **Control Plane:** n_{cn} controller nodes $\{CN_i \mid i = 1, 2, 3, \dots, n_{cn}\}$ manage network topology and user authentication.

Authentication Process

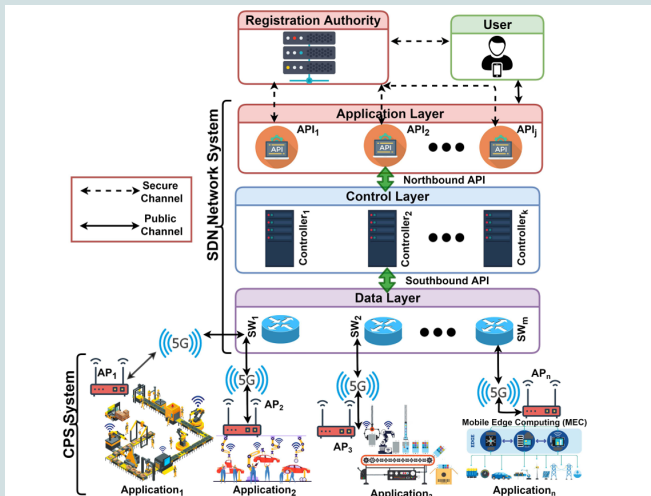
- ❖ Controller nodes authenticate registered users U_k for secure session establishment with smart devices.

Interfaces

- ❖ **Northbound Interface:** Connects control plane to higher-level entities (users).
- ❖ **Southbound Interface:** Connects data plane to lower-level entities (smart devices/access points).

Network Model

5G-Enabled Softwarized ICPS



Network Model of the proposed scheme.

Attack Model

Communication Vulnerabilities

- ❖ Communication between smart devices and users typically occurs via insecure channels. So, An adversary \mathcal{A} can compromise data shared between them.

Threat Models

- ❖ **Dolev-Yao Threat Model (DY Model):**
 - ❖ \mathcal{A} can eavesdrop, modify, delete, and inject fake information.
- ❖ **Canetti-Krawczyk Model (CK-Adversary Model):**
 - ❖ \mathcal{A} can hijack sessions and compromise session states, keys, and short-term secrets.

Security Considerations

- ❖ Session keys must be constructed from both short-term and long-term secrets to prevent Ephemeral Secret Leakage (ESL) attacks.
- ❖ The Registration Authority (RA) is considered a fully trusted entity in the ICPS environment.

Research Contributions

- ❖ **Design of UAKA-5GSICPS:** A new three-factor user authentication and key agreement scheme specifically for SDN-based ICPS.
- ❖ **Security Analysis:** The scheme is analyzed against various known attacks, proving its security.
- ❖ **Formal Security Verification:** Utilizes the AVISPA tool to demonstrate resilience against replay and man-in-the-middle attacks.
- ❖ **Comparative Analysis:** Detailed comparison with existing schemes shows:
 - ❖ Comparable communication and computation overhead.
 - ❖ Enhanced security features.
 - ❖ Practical implementation potential in real-world applications.

Related Works

- ❖ In 2015, NIST published an overview on the security of industrial control systems [1].
- ❖ In 2017, Molina et al. addressed cyber security in SDN for ICPS, focusing on access control [2].
- ❖ Taylor et al. and Chong et al. discussed security challenges in connecting IT with CPS devices [3, 4].

Key Contributions and Limitations

- ❖ Chen et al. proposed an efficient user authentication scheme but it suffers from privileged insider attack [5].
- ❖ Harishma et al. introduced a key agreement scheme for heterogeneous CPS, but it is vulnerable to ESL attacks [6].
- ❖ Eldefrawy et al. proposed an authentication protocol for industrial IoT, but it lacks mutual authentication [7].
- ❖ Renuka et al. offered a secure password-based authentication scheme for M2M networks, yet its adoption remains limited [8].

Summary and Proposed Contribution

- ❖ Many existing schemes are vulnerable or impractical for real-world applications.
- ❖ Proposed: A novel user authentication scheme for ICPS that:
 - ❖ Provides full anonymity and untraceability.
 - ❖ Is secure against modern attack strategies.
 - ❖ Supports user credentials update and dynamic IoT device addition.
- ❖ Viable for real-time ICPS applications with competitive computational and communication costs.

Introduction to UAKA-5GSICPS

- ❖ **Overview:** The proposed scheme, UAKA-5GSICPS addresses the growing security concerns in modern industrial systems.
- ❖ **Key Phases:**
 - ❖ System Bootstrap Phase: Initializes the system components.
 - ❖ Pre-deployment Phase: Prepares controller nodes and IoT smart devices for integration.
 - ❖ User Registration Phase: Allows users to register securely within the system.
 - ❖ Login Phase: Facilitates user access through secure login procedures.
 - ❖ Authentication and Key Agreement Phase: Establishes trust and secure communication between users and devices.
 - ❖ User Credentials Update Phase: Enables users to update their credentials as needed.
 - ❖ Dynamic Smart Device Addition Phase: Allows for the seamless integration of new IoT devices into the existing system.
- ❖ **Assumptions:**
 - ❖ Trusted entities: The Registration Authority (RA) and controller nodes are considered trusted.
 - ❖ Synchronized Clocks: All entities maintain synchronized clocks to prevent replay attacks.

A. System Bootstrap Phase

Overview: The System Bootstrap Phase is conducted by the Registration Authority (RA) to initialize system parameters for secure communication. Key steps include:

1. Parameter Selection:

- ❖ The RA selects a large prime number p .
- ❖ Defines a non-singular elliptic curve $E_p(a, b)$: $y^2 = x^3 + ax + b \pmod{p}$ over a Galois Field $GF(p)$ or Finite Field \mathbb{Z}_p .
- ❖ Specifies a base point P on $E_p(a, b)$ with order approximately p and includes a point at infinity O .
- ❖ Chooses a collision-resistant one-way hash function $h(\cdot)$.

2. Private and Public Key Generation:

- ❖ The RA selects a private key $\text{pr}_{RA} \in \mathbb{Z}_p^*$.
- ❖ The corresponding public key Pub_{RA} is computed as $\text{Pub}_{RA} = \text{pr}_{RA} \cdot P$.
- ❖ The private key pr_{RA} is kept secret, while Pub_{RA} and $h(\cdot)$ are made publicly available.

Purpose: These steps ensure the secure initialization of parameters necessary for the authentication and key agreement processes in UAKA-5GSICPS.

B. Pre-Deployment Phase

1. Controller Nodes Enrollment

Objective: Enroll controller nodes CN_i with secure identities and credentials.

Steps for Controller Nodes Enrollment

1. Generate Pseudo-Identity:

- ❖ RA computes $RID_{CN_i} = h(ID_{CN_i} || pr_{RA})$.
- ❖ Picks private key $pr_{CN_i} \in Z_p^*$ and calculates public key $Pub_{CN_i} = pr_{CN_i} \cdot P$.

2. Random Secret and Certificate Generation:

- ❖ Selects random secret $r_{CN_i} \in Z_p^*$, computes $R_{CN_i} = r_{CN_i} \cdot P$.
- ❖ Generates certificate

$$Cert_{CN_i} = pr_{RA} + h(RID_{CN_i} || Pub_{RA} || Pub_{CN_i}) \cdot r_{CN_i} \pmod{p}.$$

3. Preload and Secure Deletion:

- ❖ RA stores $\{RID_{CN_i}, pr_{CN_i}, Pub_{CN_i}, R_{CN_i}, Cert_{CN_i}\}$ in CN_i 's memory.
- ❖ Deletes $ID_{CN_i}, pr_{CN_i}, Cert_{CN_i}$ from its own database to avoid stolen verifier and privileged-insider attacks.

B. Pre-Deployment Phase

2. Smart Devices Enrollment

Objective: Enroll smart devices SD_j with secure identities and credentials.

Steps for Smart Devices Enrollment

1. Generate Pseudo-Identity and Key Pair:

- ❖ RA computes $RID_{SD_j} = h(ID_{SD_j} || pr_{RA})$.
- ❖ Creates private-public key pair (pr_{SD_j}, Pub_{SD_j}) , with $Pub_{SD_j} = pr_{SD_j} \cdot P$.
- ❖ Chooses random secret $r_{SD_j} \in Z_p^*$ and computes $R_{SD_j} = r_{SD_j} \cdot P$.

2. Certificate and Shared Secret Generation:

- ❖ Certificate
 $Cert_{SD_j} = pr_{RA} + h(RID_{SD_j} || Pub_{RA} || Pub_{CN_i} || Pub_{SD_j}) \cdot r_{SD_j} \pmod{p}$.
- ❖ Shared secret $s_{SD_j, CN_i} = h(RID_{SD_j} || RID_{CN_i} || r_{SD_j} || r_{CN_i} || RTS_{SD_j})$, where RTS_{SD_j} is the registration timestamp.

3. Preload and Secure Deletion:

- ❖ RA stores $\{RID_{SD_j}, RID_{CN_i}, R_{CN_i}, pr_{SD_j}, Pub_{SD_j}, R_{SD_j}, Cert_{SD_j}, s_{SD_j, CN_i}\}$ in SD_j 's memory.
- ❖ Erases $ID_{SD_j}, RTS_{SD_j}, r_{SD_j}, pr_{SD_j}$ from memory to avoid stolen verifier and privileged-insider attacks.

Additional Step: Load s_{SD_j, CN_i} of all associated devices into CN_i 's memory and delete r_{CN_i} .

C. User Registration Phase

Objective: User U_k registers with the RA in the SDN-based ICPS through a secure, one-time process.

Step 1: User Initialization

- ❖ User U_k picks an identity ID_{U_k} and password PW_{U_k} .
- ❖ U_k randomly selects a secret $a_k \in Z_p^*$ and computes their pseudo-identity $PID_{U_k} = h(ID_{U_k} || a_k)$.
- ❖ U_k generates a private-public key pair (pr_{U_k}, Pub_{U_k}) by selecting $pr_{U_k} \in Z_p^*$ and computing $Pub_{U_k} = pr_{U_k} \cdot P$.
- ❖ Sends registration request (PID_{U_k}, Pub_{U_k}) to RA over a secure channel.

Step 2: RA Processing

- ❖ RA receives the registration request and calculates $RID_{U_k} = h(PID_{U_k} || pr_{RA})$.
- ❖ Randomly selects $r_{U_k} \in Z_p^*$ and computes $R_{U_k} = r_{U_k} \cdot P$.
- ❖ Chooses a temporary identity TID_{U_k} for U_k for one-time use.

C. User Registration Phase

Step 3: Certificate Generation and Device Setup

- RA generates a certificate Cert_{U_k} for U_k as:

$$\text{Cert}_{U_k} = \text{pr}_{\text{RA}} + h(\text{RID}_{U_k} || \text{Pub}_{\text{RA}} || \text{Pub}_{U_k}) \cdot r_{U_k} \pmod{p}$$

- Issues a mobile device MD_{U_k} to U_k containing:
 - TID_{U_k} , R_{U_k} , and Cert_{U_k}
- RA deletes r_{U_k} and Cert_{U_k} from its memory.
- Maintains a mapping of TID_{U_k} , RID_{U_k} , R_{U_k} in the secure database of controller node CN_i .

Step 4: Biometrics and Computations

- U_k imprints biometrics BM_{U_k} to generate:
 - σ_{U_k} : Biometric key, τ_{U_k} : Public reproduction parameter
 using the fuzzy extractor probabilistic generation function $G(\cdot)$ as $\text{Gen}(\text{BM}_{U_k}) = (\sigma_{U_k}, \tau_{U_k})$.
- U_k then computes:
 - $L_{U_k} = \text{pr}_{U_k} \oplus h(\sigma_{U_k} || \text{PID}_{U_k} || \text{PW}_{U_k})$
 - $M_{U_k} = a_k \oplus h(\text{PW}_{U_k} || \sigma_{U_k} || \text{ID}_{U_k})$
 - $\text{Cert}_{U_k}^* = \text{Cert}_{U_k} \oplus h(\sigma_{U_k} || \text{PW}_{U_k})$
 - $W_{U_k} = h(\text{Cert}_{U_k} || \text{RID}_{U_k} || \text{PW}_{U_k} || R_{U_k} || \text{pr}_{U_k})$

C. User Registration Phase

Step 5: Final Storage and Cleanup

- ❖ U_k removes sensitive information from memory:
 - ❖ r_{U_k} and pr_{U_k} (private key)
 - ❖ $Cert_{U_k}$ from mobile device MD_{U_k}
- ❖ U_k stores the following in MD_{U_k} :
 - ❖ $Pub_{U_k}, L_{U_k}, M_{U_k}, W_{U_k}, Cert_{U_k}^*$
 - ❖ $h(\cdot), E_p(a, b), P, \tau_{U_k}$
 - ❖ $Gen(\cdot), Rep(\cdot)$
 - ❖ e_t : Error tolerance threshold for the fuzzy extractor reproduction function $Rep(\cdot)$

Controller node CN_i
$RID_{CN_i}, pr_{CN_i}, Pub_{CN_i}, R_{CN_i}, Cert_{CN_i},$ $\{(TID_{U_k}, RID_{U_k}, R_{U_k}) \mid 1 \leq k \leq n\}, \{(RID_{SD_j}, s_{SD_j, CN_i}) \mid 1 \leq j \leq m\}$
Smart device SD_j
$RID_{SD_j}, RID_{CN_i}, R_{CN_i}, (pr_{SD_j}, Pub_{SD_j}), R_{SD_j}, Cert_{SD_j}, s_{SD_j, CN_i}$
User U_k 's mobile device MD_{U_k}
$(\bar{TID}_{U_k}, \bar{RID}_{U_k}), L_{U_k}, M_{U_k}, W_{U_k}, Pub_{U_k}, R_{U_k}, Cert_{U_k}^*$ $h(\cdot), E_p(a, b), P, \tau_{U_k}, Gen(\cdot), Rep(\cdot), e_t$

Figure: Loaded Credentials in CN_i , SD_j , and User U_k 's Mobile Device MD_{U_k}

D. Login Phase

Step 1: User inputs data into mobile device and initial calculations.

User U_k Login Attempt into ICPS environment

- ❖ U_k inputs:
 - ❖ Identity ID_{U_k}
 - ❖ Password PW_{U_k}
 - ❖ Personal biometrics BM'_{U_k}
- ❖ MD_{U_k} performs calculations:
 - ❖ Computes $\sigma_{U_k} = \text{Rep}(BM_{U_k}, \tau_{U_k})$ if the Hamming distance between registered and current biometric is within tolerance e''_t
 - ❖ Computes $a_k^* = M_{U_k} \oplus h(PW_{U_k} || \sigma_{U_k} || ID_{U_k})$
 - ❖ Generates pseudo-identity $PID_{U_k} = h(ID_{U_k} || a_k^*)$
 - ❖ Determines private key $pr_{U_k} = L_{U_k} \oplus h(\sigma_{U_k} || PID_{U_k} || PW_{U_k})$
 - ❖ Computes $\text{Cert}'_{U_k} = \text{Cert}^*_{U_k} \oplus h(\sigma_{U_k} || PW_{U_k})$

D. Login Phase

Verification and Authentication Request Generation

❖ Step 2: Verification

- ❖ MD_{U_k} calculates $W_{U_k}^* = h(\text{Cert}'_{U_k} || \text{RID}_{U_k} || \text{PW}_{U_k} || \text{R}_{U_k} || \text{pr}_{U_k})$
- ❖ If $W_{U_k}^* \neq W_{U_k}$, the session is aborted.
- ❖ Otherwise, it picks a random secret $k_1 \in Z_p^*$ and generates timestamp TS_1 .
- ❖ Computes:
 - ▶ $k'_1 = h(k_1 || \text{pr}_{U_k} || \text{TS}_1)$
 - ▶ $K'_1 = k'_1 \cdot P$
 - ▶ $\text{RID}_{SD_j}^* = \text{RID}_{SD_j} \oplus h(\text{R}_{U_k} || \text{RID}_{U_k} || \text{TS}_1)$
 - ▶ $\text{Cert}_{U_k}^+ = \text{Cert}'_{U_k} + h(\text{RID}_{SD_j}^* || \text{R}_{U_k} || K'_1 || \text{Pub}_{U_k} || \text{TS}_1) \cdot k'_1 \pmod{p}$
- ❖ Generates a temporary identity $\text{TID}_{U_k}^{\text{new}}$ and computes $\text{TID}_{U_k}^*$
- ❖ Computes:
 - ▶ $\text{TID}_{U_k}^* = \text{TID}_{U_k}^{\text{new}} \oplus h(\text{TID}_{U_k} || \text{RID}_{U_k} || \text{R}_{U_k} || \text{TS}_1)$
 - ▶ $\text{TC}_{U_k} = h(\text{TS}_1 || \sigma_{U_k} || \text{PW}_{U_k} || \text{PID}_{U_k})$
 - ▶ $B_{U_k} = \text{TC}_{U_k} \oplus h(\text{RID}_{U_k} || \text{TS}_1 || \text{R}_{U_k})$

❖ Step 3: Sending Authentication Request

- ❖ MD_{U_k} sends $\text{Msg}_1 = \langle \text{TID}_{U_k}, \text{TID}_{U_k}^*, \text{Cert}_{U_k}^+, K'_1, \text{RID}_{SD_j}^*, B_{U_k}, \text{TS}_1 \rangle$ to C_{N_i} via public channel.

E. Authentication and Key Agreement Phase (Part 1)

Controller Node C_{N_i} upon Receiving Authentication Request Msg_1 :

❖ Step 1: Timestamp Verification

- ❖ C_{N_i} checks if $|\text{TS}_1 - \text{TS}_1^*| \leq \Delta T$, where TS_1^* is the reception time of Msg_1 .
- ❖ If the condition fails, C_{N_i} halts further processing.
- ❖ Otherwise, C_{N_i} retrieves $(\text{RID}_{U_k}, R_{U_k})$ for user U_k using TID_{U_k} .

❖ Step 2: Authentication Verification

- ❖ C_{N_i} verifies the below equation, If it matches, C_{N_i} proceeds; otherwise, it aborts the process. $\text{Pub}_{RA} + h(\text{RID}_{U_k} || \text{Pub}_{RA} || \text{Pub}_{U_k}) \cdot R_{U_k} + h(\text{RID}_{SD_j}^* || R_{U_k} || K'_1 || \text{Pub}_{U_k} || \text{TS}_1) \cdot K'_1 = \text{Cert}_{U_k}^+ \cdot P$

E. Authentication and Key Agreement Phase (Part 1)

❖ Step 3: Retrieve and Calculate Required Values

- ❖ Retrieve $RID_{SD_j} = RID_{SD_j}^* \oplus h(R_{U_k} || RID_{U_k} || TS_1)$
- ❖ $Cert'_{C_{N_i}} = Cert_{C_{N_i}} \oplus h(TID_{U_k} || s_{SD_j, C_{N_i}} || TS_2)$
- ❖ $TC_{U_k} = B_{U_k} \oplus h(RID_{U_k} || TS_1 || R_{U_k})$
- ❖ $C_{U_k} = h(TC_{U_k} || TS_2) \oplus h(s_{SD_j, C_{N_i}} || TS_2 || TID_{U_k} || RID_{SD_j})$
- ❖ $X_i =$
 $h(TID_{U_k} || s_{SD_j, C_{N_i}} || K'_1 || C_{U_k} || R_{C_{N_i}} || Cert'_{C_{N_i}} || RID_{C_{N_i}} || RID_{SD_j} || TS_1 || TS_2)$
- ❖ Send key establishment request
 $Msg_2 = \langle TID_{U_k}, X_i, Cert'_{C_{N_i}}, K'_1, C_{U_k}, TS_1, TS_2 \rangle$ to SD_j .

❖ Step 4: Update Temporary Identity

- ❖ Calculate $TID_{U_k}^{new} = TID_{U_k}^* \oplus h(TID_{U_k} || RID_{U_k} || R_{U_k} || TS_1)$
- ❖ Update TID_{U_k} with $TID_{U_k}^{new}$ for U_k in C_{N_i} 's secure database.

E. Authentication and Key Agreement Phase (Part 2)

Service Device SD_j upon Receiving Key Establishment Request Msg_2 :

❖ Step 1: Timestamp Verification

- ❖ SD_j checks if $|TS_2 - TS_2^*| \leq \Delta T$, where TS_2^* is the current timestamp at SD_j . If the condition fails, SD_j discards the request as stale.
- ❖ Otherwise, SD_j retrieves $(RID_{SD_j}, RID_{C_{N_i}}, s_{SD_j, C_{N_i}})$ from memory and computes:

$$Cert_{C_{N_i}} = Cert'_{C_{N_i}} \oplus h(TID_{U_k} || s_{SD_j, C_{N_i}} || TS_2)$$

❖ Step 2: Authentication Verification

- ❖ Calculate: $X_i^* = h(TID_{U_k} || s_{SD_j, C_{N_i}} || K_1' || C_{U_k} || R_{C_{N_i}} || Cert_{C_{N_i}} || RID_{C_{N_i}} || RID_{SD_j} || TS_1 || TS_2)$
- ❖ Verify if $X_i^* = X_i$ and:

$$Pub_{RA} + h(RID_{C_{N_i}} || Pub_{RA} || Pub_{C_{N_i}}) \cdot R_{C_{N_i}} = Cert_{C_{N_i}} \cdot P$$
- ❖ If these conditions hold, SD_j authenticates C_{N_i} else terminates the session.

E. Authentication and Key Agreement Phase (Part 2)

❖ Step 3: Session Key Computation

- ❖ SD_j picks a random secret $k_2 \in Z_p^*$ and computes:

$$k'_2 = h(k_2 || s_{SD_j, C_{N_i}} || pr_{SD_j} || TS_3)$$

$$K'_2 = k'_2 \cdot P$$

$$Cert_{SD_j}^+ = Cert_{SD_j} + h(TID_{U_k} || RID_{SD_j} || K'_2 || Pub_{SD_j} || TS_3) \cdot k'_2 \pmod{p}$$

$$h(TC_{U_k} || TS_2) = C_{U_k} \oplus h(s_{SD_j, C_{N_i}} || TS_2 || TID_{U_k} || RID_{SD_j})$$

- ❖ Calculate the shared session key:

$$SK_{SD_j, U_k} = h(k'_2 \cdot K'_1 || h(TC_{U_k} || TS_2) || RID_{SD_j} || TID_{U_k} || TS_1 || TS_3)$$

- ❖ Compute session key verifier:

$$SKV = h(SK_{SD_j, U_k} || TS_2 || TS_3)$$

❖ Step 4: Sending Key Establishment Response

- ❖ SD_j sends the key establishment response

$Msg_3 = \langle Cert_{SD_j}^+, R_{SD_j}, K'_2, SKV, TS_2, TS_3 \rangle$ to U_k via a public channel.

E. Authentication and Key Agreement Phase (Part 3)

User U_k Session Key Establishment with SD_j

After receiving Msg_3 from SD_j , MD_{U_k} performs the following operations:

❖ Step 1: Timestamp Verification:

- ❖ MD_{U_k} verifies the freshness of the message by checking if $|\text{TS}_3 - \text{TS}_3^*| \leq \Delta T$, where TS_3^* is the current timestamp generated by U_k .
- ❖ If the condition is not met, U_k discards the response and stops further processing.

❖ Step 2: Signature Verification:

- ❖ MD_{U_k} checks the validity of the signature:

$$\text{Pub}_{RA} + h(\text{RID}_{SD_j} || \text{Pub}_{RA} || \text{Pub}_{CN_i} || \text{Pub}_{SD_j}) \cdot \text{R}_{SD_j} +$$

$$h(\text{TID}_{U_k} || \text{RID}_{SD_j} || K'_2 || \text{Pub}_{SD_j} || \text{TS}_3) \cdot K'_2 = \text{Cert}_{SD_j}^+ \cdot P$$
- ❖ If this condition fails, U_i discards the response and stops the session.

E. Authentication and Key Agreement Phase (Part 3)

User U_k Session Key Establishment with SD_j

❖ Step 3: Session Key Computation:

- ❖ MD_{U_k} computes the session key:

$$SK_{U_k, SD_j} = h(k'_1 \cdot K'_2 || h(TC_{U_k} || TS_2) || RID_{SD_j} || TID_{U_k} || TS_1 || TS_3)$$

- ❖ Computes the session key verifier:

$$SKV^* = h(SK_{U_k, SD_j} || TS_2 || TS_3)$$

- ❖ MD_{U_k} checks if $SKV^* = SKV$. If invalid, U_k aborts the session. Otherwise, U_k and SD_j successfully establish the session key.

❖ Step 4: Identity Update:

- ❖ U_k updates TID_{U_k} with $TID_{U_k}^{new}$ in MD_{U_k} .
- ❖ The overall phase is summarized in the image next slide.

E. Login and Authentication and Key Agreement Phases

User (U_k)/mobile device (MD_{U_k})	Controller node (CN_i)	Smart device (SD_j)
<p>Input ID_{U_k}, PW_{U_k} and biometrics BM_{U_k}. Calculate $\sigma_{U_k} = \text{Rep}(BM_{U_k}, \tau_{U_k})$. $\alpha_{U_k}^* = M_{U_k} \oplus h(PW_{U_k} \parallel \sigma_{U_k} \parallel ID_{U_k})$. $PID_{U_k} = h(ID_{U_k} \parallel \alpha_{U_k}^*)$, $pr_{U_k} = L_{U_k} \oplus h(\sigma_{U_k} \parallel PID_{U_k} \parallel PW_{U_k})$, $Cert_{U_k}^* = Cert_{U_k}^* \oplus h(\sigma_{U_k} \parallel PW_{U_k})$, $W_{U_k} = h(Cert_{U_k}^* \parallel RID_{U_k} \parallel PW_{U_k} \parallel R_{U_k} \parallel pr_{U_k})$ and check if $W_{U_k} = W_{U_k}$? If valid, select a desired smart device SD_j. Pick random secret $k_1 \in \mathbb{Z}_p^*$, timestamp TS_1, compute $K_1^* = h(k_1 \parallel pr_{U_k} \parallel TS_1)$, $K_1^* = K_1^* \cdot P$, $RID_{SD_j}^* = RID_{SD_j} \oplus h(R_{U_k} \parallel RID_{U_k} \parallel TS_1)$, $Cert_{U_k}^* = Cert_{U_k}^* + h(RID_{SD_j}^* \parallel R_{U_k} \parallel K_1^* \parallel Pub_{U_k} \parallel TS_1) * k_1^* \pmod{p}$, picks a new random temporary identity $TID_{U_k}^{new}$ and compute $TID_{U_k}^* = TID_{U_k}^{new} \oplus h(TID_{U_k} \parallel RID_{U_k} \parallel R_{U_k} \parallel TS_1)$. $TC_{U_k} = h(TS_1 \parallel \sigma_{U_k} \parallel PW_{U_k} \parallel PID_{U_k})$. $B_{U_k} = TC_{U_k} \oplus h(RID_{U_k} \parallel TS_1 \parallel R_{U_k})$. $Msg_1 = (TID_{U_k}, TID_{U_k}^*, Cert_{U_k}^*, K_1^*, RID_{SD_j}^*, B_{U_k}, TS_1)$</p> <p>→ (to controller node CN_i)</p> <p>Check if $TS_3 - TS_1 \leq \Delta T$? If valid, verify if $Pub_{RA} + h(RID_{SD_j}) \parallel Pub_{RA} \parallel Cert_{CN_i} \parallel Pub_{SD_j} \cdot R_{SD_j} + h(TID_{U_k} \parallel RID_{SD_j} \parallel TS_1) \parallel K_2^* \parallel Pub_{SD_j} \parallel TS_3$. $K_2^* = Cert_{SD_j}^* \cdot P$? If valid, compute $SK_{U_k, SD_j} = h(K_1^* \cdot K_2^* \parallel h(TC_{U_k} \parallel TS_2) \parallel RID_{U_k} \parallel TS_1 \parallel TS_3)$ and $SKV^* = h(SK_{U_k, SD_j} \parallel TS_2 \parallel TS_3)$. Check if $SKV^* = SKV$? If valid, update TID_{U_k} with $TID_{U_k}^{new}$ in MD_{U_k}.</p>	<p>Check if $TS_1 - TS_1^* \leq \Delta T$? If valid, fetch (RID_{U_k}, R_{U_k}) for TID_{U_k}. Verify if $Pub_{RA} + h(RID_{U_k} \parallel Pub_{RA} \parallel Pub_{U_k}) \cdot R_{U_k} + h(RID_{SD_j}^* \parallel R_{U_k} \parallel K_1^* \parallel Pub_{U_k} \parallel TS_1) \cdot K_1^* = Cert_{U_k}^* \cdot P$? If so, compute $RID_{SD_j} = RID_{SD_j}^* \oplus h(R_{U_k} \parallel RID_{U_k} \parallel TS_1)$, $Cert_{CN_i}^* = Cert_{CN_i} \oplus h(TID_{U_k} \parallel s_{SD_j, CN_i} \parallel TS_2)$, $TC_{U_k} = B_{U_k} \oplus h(RID_{U_k} \parallel TS_1 \parallel R_{U_k})$, $C_{U_k} = h(TC_{U_k} \parallel TS_2) \oplus h(s_{SD_j, CN_i} \parallel TS_2 \parallel TID_{U_k} \parallel RID_{SD_j})$, $X_1 = h(TID_{U_k} \parallel s_{SD_j, CN_i} \parallel K_1^* \parallel C_{U_k} \parallel R_{CN_i} \parallel Cert_{CN_i} \parallel RID_{CN_i} \parallel RID_{SD_j} \parallel TS_1 \parallel TS_2)$. Prepare key establishment request Msg_2. $Msg_2 = (TID_{U_k}, X_1, Cert_{CN_i}^*, K_1^*, C_{U_k}, TS_1, TS_2)$</p> <p>→ (to smart device SD_j)</p> <p>Calculate $TID_{U_k}^{new} = TID_{U_k}^* \oplus h(TID_{U_k} \parallel RID_{U_k} \parallel R_{U_k} \parallel TS_1)$. Update TID_{U_k} with $TID_{U_k}^{new}$ for U_k in its database.</p>	<p>Check if $TS_2 - TS_2^* \leq \Delta T$? Abort otherwise. Compute $Cert_{CN_i} = Cert_{CN_i}^* \oplus h(TID_{U_k} \parallel s_{SD_j, CN_i} \parallel TS_2)$, $X_1^* = h(TID_{U_k} \parallel s_{SD_j, CN_i} \parallel K_1^* \parallel R_{CN_i} \parallel C_{U_k} \parallel Cert_{CN_i} \parallel RID_{CN_i} \parallel RID_{SD_j} \parallel TS_1 \parallel TS_2)$, $h(TC_{U_k} \parallel TS_2) = C_{U_k}$, $\oplus h(s_{SD_j, CN_i} \parallel TS_2 \parallel TID_{U_k} \parallel RID_{SD_j})$, Check if $X_1^* = X_1$, $Pub_{RA} + h(RID_{CN_i} \parallel Pub_{RA} \parallel Pub_{CN_i}) \cdot R_{CN_i} = Cert_{CN_i} \cdot P$. If both valid, generate random secret $k_2 \in \mathbb{Z}_p^*$. Compute $K_2^* = h(k_2 \parallel s_{SD_j, CN_i} \parallel pr_{SD_j} \parallel TS_3)$, $K_2^* = K_2^* \cdot P$, $Cert_{SD_j}^* = Cert_{SD_j} + h(TID_{U_k} \parallel RID_{SD_j} \parallel K_2^* \parallel Pub_{SD_j} \parallel TS_3) * k_2^* \pmod{p}$, $SK_{SD_j, U_k} = h(K_2^* \cdot K_1^* \parallel h(TC_{U_k} \parallel TS_2) \parallel RID_{U_k} \parallel TS_1 \parallel TS_3)$, session key verifier $SKV^* = h(SK_{U_k, SD_j} \parallel TS_2 \parallel TS_3)$. Prepare key establishment response Msg_3. $Msg_3 = (Cert_{SD_j}^*, R_{SD_j}, K_2^*, SKV, TS_2, TS_3)$</p> <p>← (to user U_k)</p>
Both U_k and SD_j agree on the session key SK_{U_k, SD_j} ($= SK_{SD_j, U_k}$).		

Figure: Login and authentication and key agreement phases.

F. User Credentials Update Phase (Part 1)

Overview: Registered user U_k may wish to change credentials, such as password or biometrics. This phase facilitates updating any or all credentials: *identity* ID_{U_k} , *password* PW_{U_k} , and *biometrics* BM_{U_k} .

❖ Step 1: Input Current Credentials

- ❖ U_k inputs current credentials $ID_{U_k}^{cur}$, $PW_{U_k}^{cur}$, and biometrics $BM_{U_k}^{cur}$ into MD_{U_k} .
- ❖ MD_{U_k} computes $\sigma_{U_k}^{cur} = \text{Rep}(BM_{U_k}^{cur}, \tau_{U_k})$.
- ❖ Calculates $a_k^* = M_{U_k} \oplus h(PW_{U_k}^{cur} || \sigma_{U_k}^{cur} || ID_{U_k}^{cur})$.
- ❖ Derives:
 - ▶ $PID_{U_k}^{cur} = h(ID_{U_k}^{cur} || a_k^*)$
 - ▶ $pr_{U_k}^* = L_{U_k} \oplus h(\sigma_{U_k}^{cur} || PID_{U_k}^{cur} || PW_{U_k}^{cur})$
 - ▶ $\text{Cert}_{U_k}^* = \text{Cert}_{U_k} \oplus h(\sigma_{U_k}^{cur} || PW_{U_k}^{cur})$
- ❖ Checks if $W_{U_k} = h(\text{Cert}_{U_k}^* || RID_{U_k} || PW_{U_k}^{cur} || R_{U_k} || pr_{U_k}^*)$ if it doesn't satisfy it stops proceeding further.

F. User Credentials Update Phase (Part 2)

❖ Step 2: Input New Credentials

- ❖ U_k inputs new credentials $ID_{U_k}^{new}$, $PW_{U_k}^{new}$, $BM_{U_k}^{new}$.
- ❖ MD_{U_k} computes:
 - ▶ $\sigma_{U_k}^{new}, \tau_{U_k}^{new} = \text{Gen}(BM_{U_k}^{new})$
 - ▶ $PID_{U_k}^{new} = h(ID_{U_k}^{new} || a_k^*)$
 - ▶ $L_{U_k}^{new} = pr_{U_k}^* \oplus h(\sigma_{U_k}^{new} || PID_{U_k}^{new} || PW_{U_k}^{new})$
 - ▶ $M_{U_k}^{new} = a_k^* \oplus h(PW_{U_k}^{new} || \sigma_{U_k}^{new} || ID_{U_k}^{new})$
 - ▶ $Cert_{U_k}^{new} = Cert_{U_k}^* \oplus h(\sigma_{U_k}^{new} || PW_{U_k}^{new})$
 - ▶ $W_{U_k}^{new} = h(Cert_{U_k}^{new*} || RID_{U_k} || PW_{U_k}^{new} || R_{U_k} || pr_{U_k}^*)$

❖ Step 3: Update and Replace

- ❖ MD_{U_k} replaces $L_{U_k}, M_{U_k}, Cert_{U_k}, W_{U_k}, \tau_{U_k}$ with the new values: $L_{U_k}^{new}, M_{U_k}^{new}, Cert_{U_k}^{new}, W_{U_k}^{new}, \tau_{U_k}^{new}$.

The summary of this phase is provided in Figure next slide

User Credentials Update Phase

User(U_k)	Mobile Device(MD_{U_k})
<p>Input current credentials $ID_{U_k}^{cur}$, $PW_{U_k}^{cur}$.</p> <p>Imprint current biometrics $BM_{U_k}^{cur}$.</p>	<p>Compute $\sigma_{U_k}^{cur} = Rep(BM_{U_k}^{cur}, \tau_{U_k})$, $a_k^* = M_{U_k} \oplus h(PW_{U_k}^{cur} \parallel \sigma_{U_k}^{cur} \parallel ID_{U_k}^{cur})$, $PID_{U_k}^{cur} = h(ID_{U_k}^{cur} \parallel a_k^*)$, $pr_{U_k}^* = L_{U_k} \oplus h(\sigma_{U_k}^{cur} \parallel PID_{U_k}^{cur} \parallel PW_{U_k}^{cur})$, $Cert_{U_k}^* = Cert_{U_k} \oplus h(\sigma_{U_k}^{cur} \parallel PW_{U_k}^{cur})$. Check if $W_{U_k} = h(Cert_{U_k}^* \parallel RID_{U_k} \parallel PW_{U_k}^{cur} \parallel R_{U_k} \parallel pr_{U_k}^*)$? If valid, request for new credentials.</p>
<p>Input new credentials $ID_{U_k}^{new}$, $PW_{U_k}^{new}$ and imprint new personal biometrics $BM_{U_k}^{new}$.</p>	<p>Compute, $Gen(BM_{U_k}^{new}) = (\sigma_{U_k}^{new}, \tau_{U_k}^{new})$, $PID_{U_k}^{new} = h(ID_{U_k}^{new} \parallel a_k^*)$, $L_{U_k}^{new} = pr_{U_k}^* \oplus h(\sigma_{U_k}^{new} \parallel PID_{U_k}^{new} \parallel PW_{U_k}^{new})$, $M_{U_k}^{new} = a_k^* \oplus h(PW_{U_k}^{new} \parallel \sigma_{U_k}^{new} \parallel ID_{U_k}^{new})$, $Cert_{U_k}^{new} = Cert_{U_k}^* \oplus h(\sigma_{U_k}^{new} \parallel PW_{U_k}^{new})$, $W_{U_k}^{new} = h(Cert_{U_k}^* \parallel RID_{U_k} \parallel PW_{U_k}^{new} \parallel R_{U_k} \parallel pr_{U_k}^*)$. Replace L_{U_k}, M_{U_k}, $Cert_{U_k}$, W_{U_k} and τ_{U_k} with $L_{U_k}^{new}$, $M_{U_k}^{new}$, $Cert_{U_k}^{new}$, $W_{U_k}^{new}$ and $\tau_{U_k}^{new}$ respectively from its memory.</p>

G. Dynamic Smart Device Addition Phase

The proposed scheme allows adding new smart devices to an ICPS post-deployment and further participating in the authentication and key agreement phase to establish secure sessions with a user U_k . The RA performs the following steps to add a new smart device SD_n :

❖ Step 1:

- ❖ Generate a pseudo-identity $RID_{SD_n} = h(ID_{SD_n} || pr_{RA})$.
- ❖ Randomly choose a secret $r_{SD_n} \in Z_p^*$ to compute $R_{SD_n} = r_{SD_n}.P$.
- ❖ Prepare a personalized private-public key pair (pr_{SD_n}, Pub_{SD_n}) by randomly selecting $pr_{SD_n} \in Z_p^*$ and computing $Pub_{SD_n} = pr_{SD_n}.P$.

G. Dynamic Smart Device Addition Phase

❖ Step 2:

- ❖ Generate a certificate

$$Cert_{SD_n} = (pr_{RA} + h(RID_{SD_n} || Pub_{RA} || Pub_{CN_i} || Pub_{SD_n}) \cdot r_{SD_n}) \pmod{p}.$$

- ❖ Generate the shared secret

$$s_{SD_n, CN_i} = h(RID_{SD_n} || RID_{CN_i} || r_{SD_n} || r_{CN_i} || RT_{SD_n}).$$

- ❖ Send the tuple $(RID_{SD_n}, s_{SD_n, CN_i})$ to C_{N_i} via a secure channel.

❖ Step 3:

- ❖ Load

$\{RID_{SD_n}, RID_{CN_i}, R_{CN_i}, pr_{SD_n}, Pub_{SD_n}, R_{SD_n}, Cert_{SD_n}, s_{SD_n, CN_i}\}$
into the memory of SD_n and deploy it in the ICPS environment.

- ❖ RA erases $ID_{SD_n}, RT_{SD_n}, r_{SD_n}$, and pr_{SD_n} from memory.

The overall phase is summarized in the figure next slide.

G. Dynamic Smart Device Addition Phase

Registration Authority (RA)	Controller node (CN_i)
<p>Generate pseudo-identity $RID_{SD_n} = h(ID_{SD_n} pr_{RA})$.</p> <p>Randomly choose secret $r_{SD_n} \in Z_p^*$ and compute $R_{SD_n} = r_{SD_n} \cdot P$.</p> <p>Generate public-private key pair (pr_{SD_n}, Pub_{SD_n}) by randomly choosing $pr_{SD_n} \in Z_p^*$ and computing the public key $Pub_{SD_n} = pr_{SD_n} \cdot P$.</p> <p>Generate $Cert_{SD_n} = (pr_{RA} + h(RID_{SD_n} Pub_{RA} Pub_{CN_i} Pub_{SD_n}) * r_{SD_n}) \pmod{p}$ and $s_{SD_n, CN_i} = h(RID_{SD_n} RID_{CN_i} r_{SD_n} r_{CN_i} RTS_{SD_n})$.</p> <p>$\langle RID_{SD_n}, s_{SD_n, CN_i} \rangle$ (via secure channel)</p> <p>Load $\{RID_{SD_n}, RID_{CN_i}, R_{CN_i}, pr_{SD_n}, Pub_{SD_n}, R_{SD_n}, Cert_{SD_n}, s_{SD_n, CN_i}\}$ into SD_n's memory and erase $ID_{SD_n}, RTS_{SD_n}, r_{SD_n}$ and pr_{SD_n} from its memory.</p>	<p>Store $\langle RID_{SD_n}, s_{SD_n, CN_i} \rangle$ in its secure memory.</p>

Figure: Dynamic smart device (SDn) addition phase.

IV. Security Analysis

- ❖ Discussion on the resilience of the proposed scheme (UAKA-5GSICPS) against various attacks.
- ❖ Focus on the formal security analysis under the Random Oracle (ROR) model.

A. Formal Security Analysis Under ROR Model

- ❖ Analysis through Random Oracle (ROR) model to prove:
 - ❖ Semantic security
 - ❖ Session key security (SK-security)
- ❖ Discussion on the ROR model and the SK-security of the proposed scheme in Theorem 1.

Entities and Hash Function in the ROR Model

❖ Entities representation:

- ❖ User: \mathcal{E}_{Uk}
- ❖ Controller Node: \mathcal{E}_{CNI}
- ❖ Smart Device: \mathcal{E}_{SDj}

❖ Instances representation:

- ❖ $\mathcal{E}_{t1}^{Uk}, \mathcal{E}_{t2}^{CNI}, \mathcal{E}_{t3}^{SDj}$

❖ Collision-resistant one-way hash function $h(\cdot)$:

- ❖ Modeled as a random oracle H
- ❖ Publicly available to all entities in the ROR model

Adversary Queries

❖ List of queries for the adversary A :

- ❖ $Q_{Read}(\varepsilon_{t1}^{Uk}, \varepsilon_{t2}^{CNI}, \varepsilon_{t3}^{SDj})$: An adversary A uses this query to eavesdrop the publicly exchanged messages $Msg1, Msg2, Msg3$ among $\varepsilon_{t1}^{Uk}, \varepsilon_{t2}^{CNI}$, and ε_{t3}^{SDj} during the authentication and session key establishment. This is analogous to an “eavesdropping attack”.
- ❖ $Q_{Send}(\varepsilon_t, Msg)$: This query allows A to send a message Msg to ε_t and in turn receive the response from ε_t . It is analogous to an “active attack”.
- ❖ $Q_{CorruptMD}(\varepsilon_{t1}^{Uk})$: By querying this, A can extract the parameters of MD_{Uk} , which is the registered mobile device of a user Uk . This is analogous to an “active attack”.
- ❖ $Q_{RevealSK}(\varepsilon_t)$: With this query, the shared secret session key $SK_{Uk,SDj} = (SK_{SDj,Uk})$ between Uk and SDj is revealed to the adversary A .
- ❖ $Q_{Test}(\varepsilon_t)$: The output of this query is based on the outcome of an unbiased coin “ c ”:
 - ▶ If “Flip(c) = HEAD”, it returns the shared session key $SK_{Uk,SDj}$ between Uk and SDj , if it is freshly generated.
 - ▶ If “Flip(c) = TAIL”, it randomly selects the session key $SK_{Uk,SDj} \in Z_p^*$ and returns $SK_{Uk,SDj}$.

Application of Zipf's Law [9]

- ❖ User passwords are not uniformly distributed.
- ❖ Application of Zipf's law to prove SK-security of UAKA-5GSICPS.

Theorem 1

❖ Let $Adv_A^{UAKA5GSICPS}(t)$ be the advantage function of adversary A running in polynomial time t :

❖ Definitions:

- ▶ q_h : Number of Hash queries
- ▶ q_r : Number of Read queries
- ▶ $|H|$: Range space of $h(\cdot)$
- ▶ $Adv_A^{ECDDHP}(t)$: Advantage of breaking the ECDDHP
- ▶ l_σ : Number of bits in the biometric secret key σ_{Uk}
- ▶ C', s' : Zipf's parameters

❖ Result:

$$Adv_A^{UAKA5GSICPS}(t) \leq \frac{q_h^2}{|H|} + 2 \left[\max\{C' \cdot q_r^{s'}, \frac{q_r}{2^{l_\sigma}}\} + Adv_A^{ECDDHP}(t) \right]$$

Proof Outline

- ❖ Follows a similar proof structure as in [10, 11, 12].
- ❖ Define four games: G_0, G_1, G_2, G_3 .
- ❖ Event $Succ_A^{G_j}$: Adversary correctly guesses the coin flip outcome.
- ❖ Advantage of winning the game:

$$Adv_A^{G_j} = Pr[Succ_A^{G_j}]$$

Game G_0

- ❖ **Description:** This game corresponds to the actual attack executed by adversary A against our proposed protocol in the ROR model.
- ❖ **Key Definition:** The outcome of the coin flip c is selected randomly at the beginning of G_0 .
- ❖ **Advantage Calculation:**

$$Adv_A^{UAKA-5GSICPS} = |2 \cdot Adv_A^{G_0} - 1|$$

Game G_1

- ❖ **Description:** Modeled as an “eavesdropping attack” where A tries to read public messages exchanged during the authentication and key agreement phase.
- ❖ **Messages Involved:**
 - ❖ $Msg_1 = \langle TID_{U_k}, TID_{U_k}^*, Cert_{U_k}^+, K'_1, RID_{SDj}^*, BU_k, TS_1 \rangle$
 - ❖ $Msg_2 = \langle TID_{U_k}, X_i, Cert'_{C_{Ni}}, K'_1, CU_k, TS_1, TS_2 \rangle$
 - ❖ $Msg_3 = \langle Cert_{SDj}^+, R_{SDj}, K'_2, SK_V, TS_2, TS_3 \rangle$
 - ❖ Sent from U_k to C_{Ni} , C_{Ni} to SDj , and SDj to U_k , respectively, during the authentication and key agreement phase III-E using Q_{Read} query.
- ❖ **Query Usage:** Adversary invokes Q_{Read} to read messages and later invokes $Q_{RevealSK}$ and Q_{Test} to check if the session key $SK_{U_k, SDj}$ is a legitimate key or a random number $SK_{SDj, U_k} = h(k'_2 \cdot K'_1 \parallel h(TC_{U_k} \parallel TS_2) \parallel RID_{SDj} \parallel TID_{U_k} \parallel TS_1 \parallel TS_3) = h(k'_1 \cdot K'_2 \parallel h(TC_{U_k} \parallel TS_2) \parallel RID_{SDj} \parallel TID_{U_k} \parallel TS_1 \parallel TS_3) = SK_{U_k, SDj}$. Hence, the adversary A cannot distinguish a valid session key SK_{SDj, U_k} from a random number.
- ❖ **Indistinguishability:** Since G_0 and G_1 are indistinguishable,

$$Adv_A^{G_1} = Adv_A^{G_0}$$

Game G_2

- ❖ **Description:** Models an “active attack” by simulating the H oracle.
- ❖ **Security Properties:**
 - ❖ Messages Msg_1, Msg_2, Msg_3 are protected with the collision-resistant one-way hash function $h(.)$.
 - ❖ Extracting sensitive parameters is computationally infeasible due to the one-way property of $h(.)$.
 - ❖ The values $TID_{Uk}^*, Cert_{Uk}^+, RID_{SDj}^*, X_i, Cert_{CNi}, Cert_{SDj}^+, SK_V$ included in the network messages are indistinguishable.
 - ❖ Due to the inclusion of timestamps TS_k where $k \in [1, 3]$, and the short-term keys k_1 and k_2 (which are for one-time use), collision resistance is assured.
- ❖ **Indistinguishability:** G_1 and G_2 are indistinguishable, except that G_2 includes the H query simulation.
- ❖ **Advantage Relation:**

$$|Adv_A^{G_1} - Adv_A^{G_2}| \leq \frac{q^2}{2|H|} + Adv_A^{ECDHP}(t)$$

Game G_3

- ❖ **Description:** The adversary A attempts to tamper with the smart device MD_{U_k} of a user U_k using $Q_{CorruptMD}$.
- ❖ **Challenges for A:**
 - ❖ Extracting sensitive parameters is computationally infeasible without knowing $ID_{U_k}, PW_{U_k}, \sigma_{U_k}$.
 - ❖ The probability of guessing the biometric key σ_{U_k} is approximately $\frac{1}{2^{l_\sigma}}$.
- ❖ **Indistinguishability:** G_2 and G_3 are identical with no password/biometric guessing attacks.
- ❖ **Advantage Relation:** Hence, with the Zipf's law on passwords [9], we have:

$$|Adv_A^{G_2} - Adv_A^{G_3}| \leq \max\{C' \cdot q_r^{s'}, \frac{q_r}{2^{l_\sigma}}\}$$

- ❖ using all the equations from the previous games we get:

$$Adv_A^{UAKA-5GSICPS}(t) \leq \frac{q_h^2}{|H|} + 2 \left[\max\{C' \cdot q_r^{s'}, \frac{q_r}{2^{l_\sigma}}\} + Adv_A^{ECDDHP}(t) \right]$$

Informal Security Analysis

In this section, we demonstrate that the proposed scheme possesses the ability to resist various potential attacks. The following informal methods highlight the robustness of the system against specific threats:

1. User Impersonation Attack

❖ Attack Overview:

- ❖ An adversary A attempts to impersonate a legitimate user U_k by crafting a valid authentication request.
- ❖ Required message format:

$$Msg_1 = TID_{U_k}, TID_{U_k}^*, Cert_{U_k}^+, K_1, RID_{SD_j}^*, B_{U_k}, TS_1$$

❖ Adversary Capabilities:

- ❖ A can generate a random secret k_A and compute $K_A = k_A \cdot P$.
- ❖ The timestamp TS_A can be selected as the time of sending the fabricated message.

❖ Limitations of A :

- ❖ Cannot generate a valid certificate $Cert_{U_k}^+$ due to unknown parameters $R_{U_k}, RID_{U_k}, Cert_{U_k}, pr_{U_k}$.
- ❖ Even if A captures the user's smart device MD_{U_k} , the parameters $Cert_{U_k}$ and pr_{U_k} are masked using $PID_{U_k}, PW_{U_k}, \sigma_{U_k}$.
- ❖ The adversary cannot fabricate $TID_{U_k}^*$ due to unknown values RID_{U_k} and R_{U_k} .

❖ Conclusion:

- ❖ Therefore, A cannot impersonate a registered user in UAKA-5GSICPS.

2. Controller Node Impersonation Attack

❖ Attack Overview:

- ❖ An adversary A tries to impersonate a controller node C_{N_i} by sending a fabricated message.
- ❖ Required message format:

$$Msg_2 = TID_{U_k}, X_i, Cert_{C_{N_i}}, K_1, C_{U_k}, TS_1, TS_2$$

❖ Adversary Capabilities:

- ❖ A attempts to compute a valid X_i based on:

$$X_i = h(TID_{U_k} || s_{SD_j, C_{N_i}} || K_1 || C_{U_k} || R_{C_{N_i}} || Cert_{C_{N_i}} || RID_{C_{N_i}} || RID_{SD_j} || TS_1 || TS_2)$$

❖ Limitations of A :

- ❖ Cannot compute X_i due to unknown shared secret $s_{SD_j, C_{N_i}}$.
- ❖ Even if A compromises SD_j and extracts $s_{SD_j, C_{N_i}}$, the shared secret is distinct for each device, limiting the impact.
- ❖ Compromising SD_j does not expose sensitive information of other devices or registered users.

❖ Conclusion:

- ❖ The scheme effectively withstands impersonation attempts against the controller node.

3. Smart Device Impersonation Attack

❖ Attack Overview:

- ❖ The adversary A attempts to impersonate a smart device SD_j by fabricating an authentication response message.
- ❖ Required message format:

$$Msg_3 = Cert_{SD_j}^+, R_{SD_j}, K_2, SK_V, TS_2, TS_3$$

❖ Adversary Capabilities:

- ❖ A seeks to generate a valid $Cert_{SD_j}^+$.
- ❖ Requires knowledge of:

$$Cert_{SD_j} = (pr_{RA} + h(RID_{SD_j} || Pub_{RA} || Pub_{C_{N_i}} || Pub_{SD_j}) \cdot r_{SD_j}) \mod p$$

❖ Limitations of A :

- ❖ Cannot extract $Cert_{SD_j}$ without compromising SD_j .
- ❖ Compromising SD_j does not impact the entire ICPS environment or expose critical information about other nodes and users.

❖ Conclusion:

- ❖ The proposed scheme is resilient against smart device impersonation attacks.

4. User Anonymity and Untraceability

❖ Attack Overview:

- ❖ The proposed scheme maintains user anonymity and untraceability during the login and authentication phases.
- ❖ User's real identity ID_{U_k} is never included in network messages.

❖ Adversary Capabilities:

- ❖ Assume A collects the authentication request:

$$Msg_1 = TID_{U_k}, TID_{U_k}^*, Cert_{U_k}^+, K_1, RID_{SD_j}^*, B_{U_k}, TS_1$$

- ❖ A captures the mobile device MD_{U_k} and extracts values such as $TID_{U_k}, RID_{U_k}, L_{U_k}, M_{U_k}, W_{U_k}, Pub_{U_k}, R_{U_k}, Cert_{U_k}^*, h(\cdot)$, etc.

❖ Limitations of A :

- ❖ Due to the collision-resistant property of the hash function $h(\cdot)$, guessing ID_{U_k} from L_{U_k} and M_{U_k} is infeasible without knowledge of $pr_{U_k}, PW_{U_k}, \sigma_{U_k}$, and a_k .
- ❖ Authentication requests from the same user are untraceable as TID_{U_k} differs for each request.

❖ Conclusion:

- ❖ UAKA-5GSICPS ensures both user anonymity and untraceability, even if the registered mobile device MD_{U_k} is compromised or stolen.

5. Privileged Insider Attack

❖ Attack Overview:

- ❖ Insider adversary A reads user registration request PID_{U_k}, Pub_{U_k} sent to the Registration Authority (RA).
- ❖ A accesses the registered user's mobile device MD_{U_k} post-registration.

❖ Adversary Capabilities:

- ❖ A can extract stored credentials from MD_{U_k} .
- ❖ A cannot guess ID_{U_k} or a_k due to the collision-resistant hash function $h(\cdot)$.

❖ Limitations of A :

- ❖ A cannot deduce sensitive parameters $pr_{U_k}, a_k, Cert_{U_k}$ without $ID_{U_k}, PW_{U_k}, \sigma_{U_k}$.

❖ Conclusion:

- ❖ The proposed scheme is resilient to privileged insider attacks, ensuring user security.

6. Stolen Registered Mobile Device Attack

❖ Attack Overview:

- ❖ An adversary A steals the mobile device MD_{U_k} of a registered user U_k .

❖ Adversary Capabilities:

- ❖ A can access the device but cannot derive sensitive attributes $a_k, pr_{U_k}, Cert_{U_k}$ without knowing $ID_{U_k}, PW_{U_k}, \sigma_{U_k}$.

❖ Limitations of A :

- ❖ Any modification of R_{U_k} or TID_{U_k} results in validation failures during authentication.

❖ Conclusion:

- ❖ The proposed scheme protects sensitive information even in the case of a stolen mobile device.

7. Physical Smart Device Capture Attack

❖ Attack Overview:

- ❖ An adversary A captures a smart device SD_j and extracts stored values.

❖ Adversary Capabilities:

- ❖ A retrieves various parameters unique to SD_j .
- ❖ Parameters $k_2, K_1, TID_{U_k}, TS_1$, and TS_3 are session-specific and independent.

❖ Limitations of A :

- ❖ Compromising SD_j does not reveal session keys for other devices; only the session key for the compromised device is affected.

❖ Conclusion:

- ❖ The proposed scheme maintains security against physical device capture attacks.

8. Password Guessing Attacks

❖ Attack Overview:

- ❖ An adversary A attempts to guess the user's password PW_{U_k} through various means.

❖ Adversary Capabilities:

- ❖ A can capture MD_{U_k} and attempt to extract values, but cannot guess PW_{U_k} without additional information.

❖ Limitations of A :

- ❖ The password is never transmitted over the network, making online guessing attacks infeasible.
- ❖ Deriving pr_{U_k} from Pub_{U_k} relies on the hardness of the ECDLP.

❖ Conclusion:

- ❖ The proposed scheme is secure against both online and offline password guessing attacks.

9. Replay Attack

❖ Attack Overview:

- ❖ An adversary A attempts to replay previously captured messages during authentication.

❖ Adversary Capabilities:

- ❖ A captures messages that include timestamps to authenticate sessions.

❖ Limitations of A :

- ❖ Messages are validated against their timestamps, ensuring freshness and preventing replay.
- ❖ Messages older than the maximum transmission delay T are rejected.

❖ Conclusion:

- ❖ The proposed scheme effectively protects against replay attacks through timestamp validation.

10. Man-in-the-Middle (MITM) Attack

❖ Attack Overview:

- ❖ An adversary A intercepts and tries to modify authentication messages between U_k and CNi .

❖ Adversary Capabilities:

- ❖ A attempts to create valid authentication requests but lacks critical parameters $pr_{U_k}, k_1, RID_{U_k}, R_{U_k}$.

❖ Limitations of A :

- ❖ Even if A is a legitimate user, they cannot generate valid messages for others without knowing specific identifiers.

❖ Conclusion:

- ❖ The scheme is robust against MITM attacks due to stringent parameter requirements for message validation.

11. Ephemeral Secret Leakage Attack

❖ Attack Overview:

- ❖ An adversary A tries to exploit leaked ephemeral session keys from compromised devices.

❖ Adversary Capabilities:

- ❖ A may obtain short-term keys k_1, k_2 but cannot compute the session key SK_{SD_j, U_k} without long-term secrets.

❖ Limitations of A :

- ❖ Knowing long-term secrets is insufficient for computing the session key without the ephemeral keys.
- ❖ Session keys are independent and do not impact future keys.

❖ Conclusion:

- ❖ The proposed scheme provides SK-security and preserves forward and backward secrecy against ephemeral secret leakage attacks.

V. Formal Security Verification Using AVISPA

- ❖ The proposed UAKA-5GSICPS is verified against **replay** and **man-in-the-middle (MitM)** attacks.
- ❖ We utilize the **AVISPA tool**, a push-button validation tool for security protocols.
- ❖ AVISPA provides the **High-Level Protocol Specification Language (HLPSL)** for specifying protocols and properties.
- ❖ It combines four backends for various automatic analysis techniques:
 - ❖ On-the-Fly Model-Checker (OFMC)
 - ❖ Constraint Logic-based Attack Searcher (CL-AtSe)
 - ❖ SAT-based Model Checker (SATMC)
 - ❖ Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)

Implementation and Simulation Setup

- ❖ The proposed UAKA-5GSICPS is implemented for:
 - ❖ **Registration phase** through a secure channel.
 - ❖ **Login and authentication key agreement phases** through a public channel.
- ❖ Roles defined in HLPSSL:
 - ❖ Registration Authority (RA)
 - ❖ Registered user U_k
 - ❖ Controller node CN_i
 - ❖ Smart device SD_j
 - ❖ **Session and goal and environment** roles.
- ❖ The Dolev-Yao (DY) threat model is implemented, with the intruder i participating actively during communication.

Simulation Results and Conclusion

- ❖ Simulations conducted using **OFMC** and **CL-AtSe** backends.
- ❖ **Exclusion of TA4SP and SATMC** due to the use of XOR operation:
 - ❖ These backends cannot support XOR, making results inconclusive.
- ❖ Simulation outcomes:
 - ❖ Both backends produce **SAFE output**, confirming security against replay and MiTM attacks.

Simulation Results

<p>SUMMARY SAFE</p> <p>DETAILS BOUNDED_NUMBER_OF_SESSIONS</p> <p>PROTOCOL /home/sourav/Desktop/span/ testsuite/results/UAKA-5GSICPS.if</p> <p>GOAL as specified</p> <p>BACKEND OFMC</p> <p>STATISTICS TIME 3224 ms parseTime 0 ms visitedNodes: 576 nodes depth: 7 plies</p>	<p>SUMMARY SAFE</p> <p>DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL</p> <p>PROTOCOL /home/sourav/Desktop/span/ testsuite/results/UAKA-5GSICPS.if</p> <p>GOAL As specified</p> <p>BACKEND CL-AtSe</p> <p>STATISTICS Analysed : 7 states Reachable : 1 states Translation: 0.39 seconds Computation: 0.01 seconds</p>
---	---

Figure: Simulation results under OFMC and CL-AtSe backends.

VI. Experiment Setup for MIRACL

❖ Cryptographic Library Used:

- ❖ Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) [13].

❖ Platforms:

❖ Server Environment:

- ▶ Model: MacBook Pro (2019)
- ▶ CPU Architecture: 64-bit
- ▶ Processor: 2.3 GHz Intel Core i9
- ▶ Memory: 32 GB
- ▶ OS: macOS Mojave 10.14.6

❖ Smart Device Environment:

- ▶ Model: Raspberry Pi 3 B+ Rev 1.3
- ▶ CPU: 64-bit, 1.4 GHz Quad-core
- ▶ Memory: 1 GB
- ▶ OS: Ubuntu 20.04 LTS, 64-bit

Measurement of Execution Time

❖ Cryptographic Primitives:

- ❖ T_{ecm} : Elliptic curve point multiplication
- ❖ T_{eca} : Elliptic curve point addition
- ❖ T_h : One-way hash function (SHA-256)
- ❖ T_e : Modular exponentiation
- ❖ T_{bp} : Bilinear pairing operation
- ❖ T_{senc} : Symmetric encryption
- ❖ T_{sdec} : Symmetric decryption
- ❖ $T_{ibe-keygen}$: Key generation for ID-based encryption
- ❖ $T_{ibe-enc}$: ID-based encryption
- ❖ $T_{ibe-dec}$: ID-based decryption

❖ Methodology:

- ❖ Each cryptographic primitive executed for 100 runs.
- ❖ Average run-time measured in milliseconds.

Various Cryptographic Primitives Execution Time

Cryptographic primitive	Server average time (ms)	User/smart device average time using Raspberry PI 3 (ms)
T_{ecm}	0.382	2.288
T_{eca}	0.002	0.016
T_h	0.024	0.309
T_e	0.039	0.228
T_{bp}	6.353	32.084
T_{senc}	0.001	0.018
T_{sdec}	0.001	0.014
$T_{ibe-keygen}$	0.097	0.451
$T_{ibe-enc}$	3.549	20.332
$T_{ibe-dec}$	13.169	67.962

Figure: Execution time of various cryptographic primitives using MIRACL

VII. Comparative Analysis

- ❖ This section presents a comparative study of the proposed UAKA-5GSICPS scheme against existing competing schemes:
 - ❖ Harishma et al. [12]
 - ❖ Chen et al. [13]
 - ❖ Chen et al. [14]
- ❖ We focus on:
 - ❖ Security and functionality features
 - ❖ Computational costs
 - ❖ Communication costs
- ❖ The analysis demonstrates the superiority of UAKA-5GSICPS in various aspects compared to the existing schemes.

1. Security and Functionality Features Comparison

- ❖ Table in the next slide compares essential security and functionality features (F1–F14) among UAKA-5GSICPS and competing schemes.
- ❖ Key Observations:
 - ❖ UAKA-5GSICPS demonstrates superior performance across all essential features.
 - ❖ Features include:
 - ▶ Anonymity and untraceability
 - ▶ Resistance to replay and MiTM attacks
 - ▶ Robust key agreement protocols
- ❖ Conclusion:
 - ❖ The comprehensive security functionalities provided by UAKA-5GSICPS position it as a leading solution in the context of ICPS environments.

Security and Functionality Features Comparison

Feature	Harishma <i>et al.</i> [27]	Chen <i>et al.</i> [26]	Chen <i>et al.</i> [29]	UAKA-5GSICPS
$\mathcal{F}1$	×	✓	×	✓
$\mathcal{F}2$	×	×	×	✓
$\mathcal{F}3$	✓	✓	✓	✓
$\mathcal{F}4$	✓	✓	✓	✓
$\mathcal{F}5$	✓	✓	✓	✓
$\mathcal{F}6$	✓	✓	×	✓
$\mathcal{F}7$	✓	×	✓	✓
$\mathcal{F}8$	✓	✓	×	✓
$\mathcal{F}9$	×	✓	×	✓
$\mathcal{F}10$	N/A	✓	✓	✓
$\mathcal{F}11$	×	×	✓	✓
$\mathcal{F}12$	✓	✓	✓	✓
$\mathcal{F}13$	N/A	✓	N/A	✓
$\mathcal{F}14$	×	✓	✓	✓

$\mathcal{F}1$: “user anonymity”; $\mathcal{F}2$: “user untraceability”; $\mathcal{F}3$: “offline guessing attacks”; $\mathcal{F}4$: “fast wrong input detection”; $\mathcal{F}5$: “mutual authentication and session key agreement”; $\mathcal{F}6$: “impersonation attacks”; $\mathcal{F}7$: “privileged-insider attack”; $\mathcal{F}8$: “replay attack”; $\mathcal{F}9$: “man-in-the-middle attack”; $\mathcal{F}10$: “stolen smart card/mobile device attack”; $\mathcal{F}11$: “ESL attack under CK-adversary model”; $\mathcal{F}12$: “smart device physical capture attack”; $\mathcal{F}13$: “freely and locally password/biometric changing facility”; $\mathcal{F}14$: “dynamic smart device addition”; ✓: “a scheme is secure or supports a functionality feature”; ×: “a scheme is insecure or does not support a feature”; N/A: “not applicable”.

2. Computational Costs Comparison

- ❖ Focus on the computational costs during the login and authentication key agreement phases.
- ❖ Costs for UAKA-5GSICPS:
 - ❖ User U_k : $T_{fe} + 16T_h + 5T_{ecm} + 2T_{eca}$
 - ❖ Controller Node C_{Ni} : $9T_h + 3T_{ecm} + 2T_{eca}$
 - ❖ Smart Device SD_j : $8T_h + 4T_{ecm} + T_{eca}$
- ❖ Comparisons:
 - ❖ UAKA-5GSICPS is comparable to Chen et al. [12] and outperforms Harishma et al. [13] in terms of controller node/server overhead.
 - ❖ Slightly higher overhead on users' mobile devices, but remains within acceptable limits.
- ❖ Conclusion:
 - ❖ The security functionalities offered by UAKA-5GSICPS justify the computational costs incurred.

3. Communication Costs Comparison

- ❖ Communication cost analysis based on bit-sizes for various components:
 - ❖ Identity: 160 bits
 - ❖ Random nonce (secret): 160 bits
 - ❖ Current timestamp: 32 bits
 - ❖ Hash output (SHA-1): 160 bits
 - ❖ Elliptic curve point: 320 bits
- ❖ Transmission Requirements:
 - ❖ User U_k : 1152 bits
 - ❖ Controller Node C_{Ni} : 1024 bits
 - ❖ Smart Device SD_j : 1024 bits
- ❖ Total Communication Cost: 3200 bits for three messages.
- ❖ Comparison with Competing Schemes:
 - ❖ UAKA-5GSICPS is comparable to Chen et al. [12] and Chen et al. [14].
 - ❖ Outperforms Harishma et al. [13] in communication efficiency.
- ❖ Conclusion:
 - ❖ UAKA-5GSICPS balances security and efficiency in communication costs.

Comparison Tables

TABLE V
COMPARATIVE STUDY ON COMPUTATIONAL COSTS

Scheme	Smart device/user mobile device	Controller node/Server
Harishma <i>et al.</i> [27]	$6T_h + 2T_e + T_{senc}$ ≈ 2.33 ms	$5T_h + 2T_e + T_{ibe-keygen}$ $+ T_{ibe-dec} \approx 13.46$ ms
Chen <i>et al.</i> [26]	$8T_h + 5T_{ecm}$ ≈ 13.91 ms	$7T_h + T_{ecm}$ ≈ 0.55 ms
Chen <i>et al.</i> [29]	$8T_h + 7T_{ecm} + 2T_{eca}$ $+ 2T_e + 4T_{bp} \approx 147.31$ ms	—
UAKA-5GSICPS	$T_{fe} + 24T_h + 9T_{ecm} + 3T_{eca}$ ≈ 30.344 ms	$9T_h + 3T_{ecm} + 2T_{eca}$ ≈ 1.366 ms

TABLE VI
COMMUNICATION COST COMPARISON

Scheme	User	Server	Smart device	Total cost device (in bits)
Harishma <i>et al.</i> [27]	1344	$160k + 2720$	—	$160k + 4064$
Chen <i>et al.</i> [26]	832	320	1824	2976
Chen <i>et al.</i> [29]	1152	—	960	2112
UAKA-5GSICPS	1152	1024	1024	3200

k : number of challenge vectors used in Harishma *et al.*'s scheme [27].

Concluding Remarks (Part 1)

❖ Overview of the Study:

- ❖ Discussed security aspects of SDN-based ICPS environments.
- ❖ Designed a novel secure user authentication and key agreement scheme (UAKA-5GSICPS).

❖ Key Features of UAKA-5GSICPS:

- ❖ **Three-factor authentication:**
 - ▶ User password
 - ▶ Mobile device
 - ▶ Personal biometrics
- ❖ Supports mutual authentication and session key establishment.
- ❖ Allows dynamic addition of smart devices and user credential changes without RA communication.

Concluding Remarks (Part 2)

❖ Security Analysis:







- ❖ Formal and informal analyses demonstrate resilience against modern attacks.
- ❖ AVISPA tool verification confirms security against replay and MiTM attacks.
- ❖ Offers functionalities such as user anonymity and traceability.

❖ Comparative Analysis:

- ❖ UAKA-5GSICPS is comparable in computational and communication costs to other schemes.
- ❖ Feasible for practical applications, including medical CPS and vehicular transportation.

❖ Future Work:

- ❖ Explore secure user authentication and session key establishment in SDNs with distributed control plane architecture.

-  NIST, “Guide to industrial control systems (ics) security,” 2015, accessed on Jun. 2020. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.80082r2.pdf>
-  E. Molina and E. Jacob, “Software-defined networking in cyber physical systems: A survey,” *Comput. Electr. Eng.*, vol. 66, pp. 407–419, 2018.
-  J. M. Taylor and H. R. Sharif, “Security challenges and methods for protecting critical infrastructure cyber-physical systems,” in *Proc. Int. Conf. Sel. Topics Mobile Wireless Networking (MoWNeT)*, May 2017, pp. 1–6.
-  M. S. Chong, H. Sandberg, and A. M. H. Teixeira, “A tutorial introduction to security and privacy for cyber-physical systems,” in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 968–978.
-  Y. Chen, J.-F. Martínez, P. Castillejo, and L. López, “A privacy protection user authentication and key agreement scheme tailored for the internet of things environment: Priauth,” *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–17, Dec 2017.
-  B. Harishma, S. Patranabis, U. Chatterjee, and D. Mukhopadhyay, “Poster: Authenticated key-exchange protocol for heterogeneous cps,” in

Proceedings of the Asia Conference on Computer and Communications Security, May 2018, pp. 849–851.



K. Eldefrawy et al., “User authentication protocol for industrial iot,” *International Journal of Information Management*, vol. 48, pp. 192–203, 2019.



R. Renuka et al., “Secure password-based authentication for m2m networks,” *Journal of Network and Computer Applications*, vol. 136, pp. 87–97, 2019.



D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, “Zipf’s law in passwords,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, November 2017.



S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, “Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 824–839, September 2018.



S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, “Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things,” *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2884–2895, August 2018.



M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, March 2020.



"Miracl cryptographic sdk: Multiprecision integer and rational arithmetic cryptographic library," 2020, accessed: Apr. 2020. [Online]. Available: <https://github.com/miracl/MIRACL>



Y. Chen, J. Martínez, P. Castillejo, and L. López, "A bilinear map pairing based authentication scheme for smart grid communications: Pauth," *IEEE Access*, vol. 7, pp. 22 633–22 643, 2019.