A PUF-based Secure Communication Protocol for IoT

Urbi Chatterjee
Ph.D. Scholar

Secure Embedded Architecture Laboratory (SEAL),
Department of Computer Science and Engineering,
Indian Institute of Technology Kharagpur.

Outline

- Objectives
- Background
- Proposed Authentication, Key exchange and Secure Communication Protocol
- Security Analysis
- Implementation
- Future Work

Objectives

- To develop a lightweight identity-based security protocol suitable for Internet of Things (IoT) framework, to enable secure authentication and message exchange among the smart devices.

Objectives

- To develop a lightweight identity-based security protocol suitable for Internet of Things (IoT) framework, to enable secure authentication and message exchange among the smart devices.
- Physically Unclonable Functions (PUFs) will be used for generating the public identity of each device.

Objectives

- To develop a lightweight identity-based security protocol suitable for Internet of Things (IoT) framework, to enable secure authentication and message exchange among the smart devices.
- Physically Unclonable Functions (PUFs) will be used for generating the public identity of each device.
- This identity will be used to generate the public key for each device for message encryption.

Objectives

- To develop a lightweight identity-based security protocol suitable for Internet of Things (IoT) framework, to enable secure authentication and message exchange among the smart devices.
- Physically Unclonable Functions (PUFs) will be used for generating the public identity of each device.
- This identity will be used to generate the public key for each device for message encryption.
- Formal proofs of security for the proposed protocol will be provided in the Session Key security and Universally Composable Framework.

Objectives

- To develop a lightweight identity-based security protocol suitable for Internet of Things (IoT) framework, to enable secure authentication and message exchange among the smart devices.
- Physically Unclonable Functions (PUFs) will be used for generating the public identity of each device.
- This identity will be used to generate the public key for each device for message encryption.
- Formal proofs of security for the proposed protocol will be provided in the Session Key security and Universally Composable Framework.
- Implementation of a low-overhead hardware/software co-design of the architecture.

Background

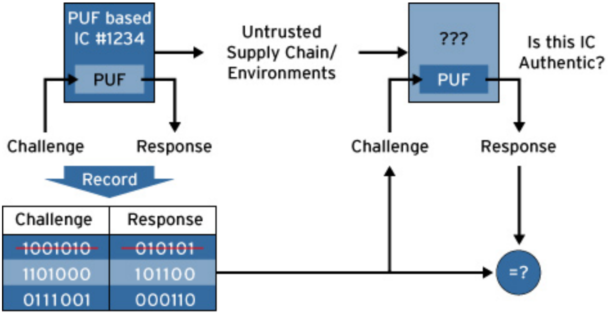
Physically Unclonable Functions (PUFs)

- PUF is a mapping $\gamma : \{0,1\}^p \longrightarrow \{0,1\}^q$, where the output q-bit words are unambiguously identified by both the p challenge bits and the unclonable, unpredictable (but repeatable) instance specific system behavior.
- Easy to design and fabricate, but infeasible to replicate, even if given the exact manufacturing process.
- PUF offloads the computational expense of cryptographic algorithms while having relatively low hardware overhead.

Figure : General block diagram of PUFs

PUF based Authentication

Figure : The mechanism of PUF based Authentication



Cryptographic Pairing

Definition 1: A *bilinear pairing* is a map $e: \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_3$ where $\mathbb{G}_1$, $\mathbb{G}_2$ are additive groups, $\mathbb{G}_3$ is a multiplicative group, and the map is linear in each component:

$$e(P+Q,R) = e(P,R)\cdot e(Q,R) \quad (1)$$

$$e(P,Q+R) = e(P,Q)\cdot e(P,R) \quad (2)$$

Notations:

- F_p : a prime field with characteristic p
- $E(F_p)$: an elliptic curve defined over F_p
- n : the order of $E(F_p)$
- r : a large prime dividing n
- k : the least positive integer such that $r|(p^k-1)$ and $r^2 \nmid (p^k-1)$. It is called the *embedding degree of r* with regard to F_p
- $[a]P$: the multiplication of a point $P \in E$ by a scalar $a \in \mathbb{Z}$
- $\mathcal{O} \in E$: the point at infinity

The r-torsion group of the curve is contained in $E(F_{p^k})$, while the r-th roots of unity are contained in F_{p^k}.

Cryptographic Pairing

Definition 2: Let $P, Q \in E(F_{p^k})[r]$ and let D_P and D_Q be degree zero divisors with disjoint supports such that $D_P \backsim (P)(O)$ and $D_Q \backsim (Q)(O)$. There exist functions f and g such that $(f) = rD_P$ and $(g) = rD_Q$. The Weil pairing ê is a map ê $: E(F_{p^k})[r] \times E(F_{p^k})[r] \to \mu_r$, where μ_r is the order r subgroup of $E(F_{p^k})$ and it is defined as:

$$\hat{e}(P, Q) = f(D_Q)/g(D_P) \tag{3}$$

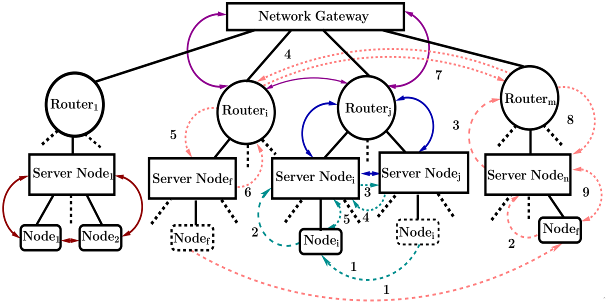
The Weil pairing of order r has the following important properties:

- Non-degeneracy: for each $P \neq \mathcal{O}$, there exists $Q \in E(F_{p^k})[r]$ such that $\hat{e}(P, Q) \neq 1$.
- Bilinearity: for any integer t, $\hat{e}([t]P, Q) = \hat{e}(P, [t]Q) = \hat{e}(P, Q)^t$ for all $P \in E(F_{p^k})[r]$ and $Q \in E(F_{p^k})[r]$.
- Computability: there exists an efficient algorithm to compute $\hat{e}(P, Q)$ given P and Q.

Proposed Authentication, Key exchange and Secure Communication Protocol

This work was accepted in *ACM Transaction on Embedded Computing Systems (TECS)* and was partially funded by an SGDRI Research Grant from IIT Kharagpur, and a research grant from Wipro Limited.

Secure Communication Mechanism in Different Levels of IoT Architecture



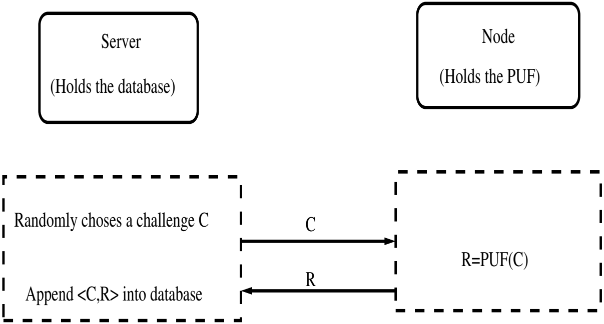
Public Mathematical Parameters

For some large prime value p, two groups $\mathbb{G}_1$, $\mathbb{G}_2$ of order p are generated, and an admissible bilinear map ê: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is defined over these two groups. We also need to choose four secure cryptographic hash functions:

- H_1: $\{0,1\}^n \rightarrow \mathbb{G}_1^*$
- H_2: $\mathbb{G}_2 \rightarrow \{0,1\}^n$
- H_3: $\{0,1\}^n \times \{0,1\}^n \rightarrow Z_p^*$
- H_4: $\{0,1\}^n \rightarrow \{0,1\}^n$

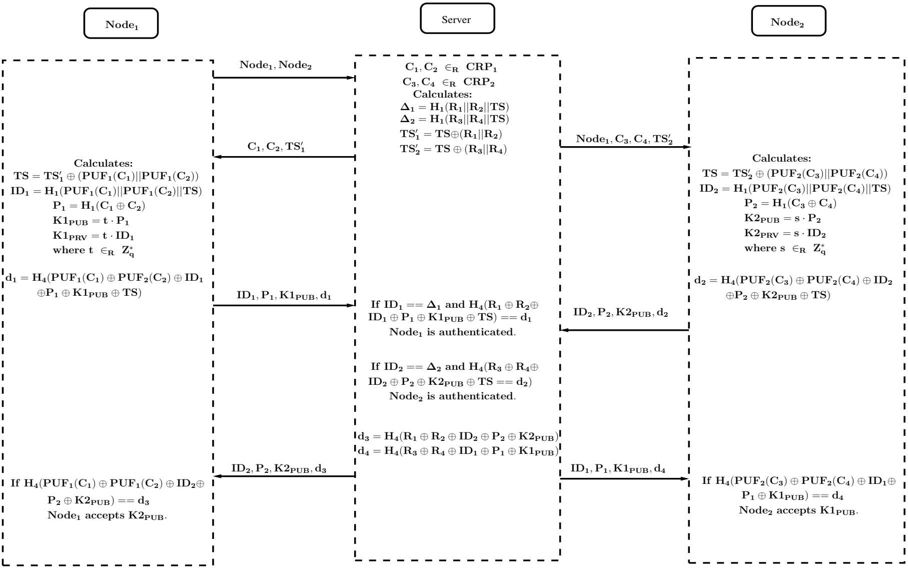
where n is the bit length of the message. So the public mathematical parameters are: $<p, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, H_1, H_2, H_3, H_4>$.

Enrolment Phase

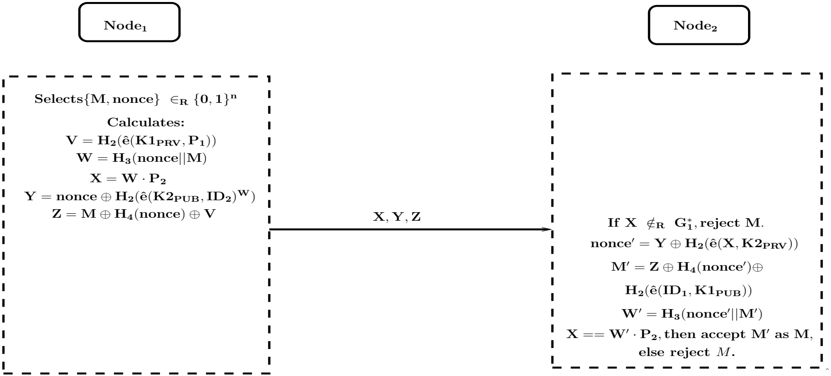


(The process is repeated for pre–defined K times)

Authentication and Key Sharing Phase



Secure Communication Phase



- since $K1_{PRV} = t \cdot ID_1$, $\hat{e}(K1_{PRV}, P_1) = \hat{e}(t \cdot ID_1, P_1) = \hat{e}(ID_1, P_1)^t \in \mathbb{G}_2$.
- $Y = nonce \oplus H_2(\hat{e}(K2_{PUB}, ID_2)^W) = nonce \oplus H_2(\hat{e}(s \cdot P_2, ID_2)^W) = nonce \oplus H_2(\hat{e}(P_2, ID_2)^{s \cdot W})$
- $nonce' = Y \oplus H_2(\hat{e}\ (X, K2_{PRV}) = Y \oplus H_2(\hat{e}\ (W \cdot P_2, s \cdot ID_2) = Y \oplus H_2(\hat{e}(P_2, ID_2)^{s \cdot W})$
- $M' = Z \oplus H_4(nonce') \oplus H_2(\hat{e}(ID_1, K1_{PUB})) = Z \oplus H_4(nonce') \oplus H_2(\hat{e}(ID_1, t \cdot P_1)) = Z \oplus H_4(nonce') \oplus H_2(\hat{e}(ID_1, P_1)^t)$

Security Analysis

Definition of Session-Key Security

Definition 3: *(Security by indistinguishability)* Suppose, two games Game 1 and Game 2 are constructed in which the adversary communicates with the protocol under consideration. If no feasible adversary can distinguish between whether she is interacting with Game 1 or Game 2 , then the protocol is said to be indistinguishable and secure. usually two adversarial models are considered in this framework:

- **The Unauthenticated-link Adversarial Model (UM)**
- **The Authenticated-link Adversarial Model (AM)**

Definition of Session-Key Security

To define UM, first an "experiment" is defined where the attacker Λ chooses to attack a session under "test", and is asked to distinguish between the real value of the session key and a random value. Let κ be the shared session key of the session under test. We consider the result of a coin toss b, where $b \in \{0, 1\}$. If $b = 0$, the value κ is given to the attacker Λ, otherwise a random value r, randomly chosen from the probability distribution of keys generated by the protocol π, is provided. The attacker have the permission to act as a regular UM attacker, and at the end of its run, outputs a bit b'.

Definition 4:A key-exchange (KE) protocol π is called SK-secure if the following properties hold for any KE-adversary Λ in the UM:

1. Protocol π satisfies the property that if two uncorrupted parties successfully complete a session then they both output the same key, and,
2. the probability that Λ guesses correctly the bit i.e., $b' = b$ is more than $\frac{1}{2}$ by only a *negligible quantity*.

The Uniqueness Property of Physical Unclonable Functions

- The uniqueness property of the PUF circuit embedded in a chip provides the capability of uniquely identify it from a set of PUF instances of the same type, which have gone through the same manufacturing process.

The Uniqueness Property of Physical Unclonable Functions

- The uniqueness property of the PUF circuit embedded in a chip provides the capability of uniquely identify it from a set of PUF instances of the same type, which have gone through the same manufacturing process.
- The uniqueness metric is defined as:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100 \quad (4)$$

The Uniqueness Property of Physical Unclonable Functions

- The uniqueness property of the PUF circuit embedded in a chip provides the capability of uniquely identify it from a set of PUF instances of the same type, which have gone through the same manufacturing process.
- The uniqueness metric is defined as:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100 \tag{4}$$

- The ideal value is 50%.

The Uniqueness Property of Physical Unclonable Functions

- The uniqueness property of the PUF circuit embedded in a chip provides the capability of uniquely identify it from a set of PUF instances of the same type, which have gone through the same manufacturing process.
- The uniqueness metric is defined as:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100 \tag{4}$$

- The ideal value is 50%.
- It is infeasible to physically clone a given PUF instance, for most PUF types.

The Uniqueness Property of Physical Unclonable Functions

- The uniqueness property of the PUF circuit embedded in a chip provides the capability of uniquely identify it from a set of PUF instances of the same type, which have gone through the same manufacturing process.
- The uniqueness metric is defined as:

$$Uniqueness = \frac{2}{k(k-1)} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \frac{HD(R_i, R_j)}{n} \times 100 \tag{4}$$

- The ideal value is 50%.
- It is infeasible to physically clone a given PUF instance, for most PUF types.
- We will prove that our proposed protocols are secure as long as the underlying problem of replicating (either physically or mathematically) the challenge-response mapping of a given PUF instance is hard.

Definition 5: *(Decisional Uniqueness Problem (DUP))* Given a PUF instance PUF_{Adv}, a challenge C and an n-bit string $z \in \{0,1\}^n$, the DUP aims to decide whether $z = PUF_N(C)$ for a PUF instance PUF_N, or a random n-bit string.

Definition 6: *(2-Decisional Uniqueness Problem (2-DUP))* Given a PUF instance PUF_{Adv}, two challenges C_1, C_2, and two n-bit strings $z_1, z_2 \in \{0,1\}^n$, the problem aims to find out whether $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ for another PUF instance PUF_N, or two random n-bit strings.

The computational indistinguishability refers to the *probability ensembles* which are infinite sequence of probability distributions.

The Uniqueness Property of Physical Unclonable Functions

Definition 7: *(Decisional Uniqueness Problem Assumption)* The problem of fabricating a PUF instance PUF_N using another instance PUF_{Adv} is hard, and for all probabilistic, polynomial time algorithm $\mathcal{A}$, there exists a *negligible function* $negl(\cdot)$ such that:

$$\mid Pr[\mathcal{A}(C, PUF_{Adv}, z) = 1] - Pr[\mathcal{A}(C, PUF_{Adv}, PUF_N(C)) = 1] \mid \leqslant negl(n) \qquad (5)$$

Definition 8: *(2-Decisional Uniqueness Problem Assumption)* The problem of fabricating a PUF instance PUF_N using another instance PUF_{Adv} is hard, and for all probabilistic, polynomial time algorithm $\mathcal{B}$, there exists a *negligible function* $negl(\cdot)$ such that:

$$\mid Pr[\mathcal{B}(C_1, C_2, PUF_{Adv}, z_1, z_2) = 1] - Pr[\mathcal{B}(C_1, C_2, PUF_{Adv}, PUF_N(C_1), PUF_N(C_2)) = 1] \mid \leqslant negl(n)$$

Claim: The 2-DUP problem is at least as hard as DUP.

Correctness Proof of the Proposed Scheme

Let data node N along with its PUF instance PUF_N is running the protocol π with the server node S at timestamp TS. Now,

$$output_{N,\pi}(C_1, C_2, (PUF_N(C_1)||PUF_N(C_2)) \oplus TS) = H_1(PUF_N(C_1)||PUF_N(C_2)||TS) \quad (6)$$

and

$$output_{S,\pi}(C_1, C_2, (R_1||R_2) \oplus TS) = H_1(R_1||R_2||TS) \quad (7)$$

Definition 9: A protocol π for authentication and key exchange is called *correct* if there exists a negligible function $negl(\cdot)$, such that for every possible value of n:

$$\Pr[output_{N,\pi}(C_1, C_2, (PUF_N(C_1)||PUF_N(C_2)) \oplus TS) \neq output_{S,\pi}(C_1, C_2, (R_1||R_2) \oplus TS)] \leqslant negl(n)$$

The Eavesdropping Authentication and Key Exchange Experiment $\mathbf{Auth}_{\mathbf{adv},\pi}(\mathbf{n}, \zeta, \mathbf{PUF}_{\mathbf{Adv}}, \mathbf{ID_0}, \mathbf{ID_1})$:

1. The adversary *Adv* is provided:

Security Proof of the Proposed Scheme

The Eavesdropping Authentication and Key Exchange Experiment $\mathbf{Auth_{adv,\pi}(n, \zeta, PUF_{Adv}, ID_0, ID_1)}$:

1. The adversary *Adv* is provided:
   1. $\zeta =< C_1,\ C_2,\ TS' >$ where $TS' = ((PUF_N(C_1)||PUF_N(C_2)) \oplus TS)$.

Security Proof of the Proposed Scheme

The Eavesdropping Authentication and Key Exchange Experiment $\mathbf{Auth_{adv,\pi}(n, \zeta, PUF_{Adv}, ID_0, ID_1)}$:

1. The adversary *Adv* is provided:
    1. $\zeta =< C_1,\ C_2,\ TS' >$ where $TS' = ((PUF_N(C_1)||PUF_N(C_2)) \oplus TS)$.
    2. A PUF instance PUF_{Adv}.

Security Proof of the Proposed Scheme

The Eavesdropping Authentication and Key Exchange Experiment $\mathbf{Auth_{adv,\pi}(n, \zeta, PUF_{Adv}, ID_0, ID_1)}$:

1. The adversary *Adv* is provided:
    1. $\zeta =< C_1, C_2, TS' >$ where $TS' = ((PUF_N(C_1)||PUF_N(C_2)) \oplus TS)$.
    2. A PUF instance PUF_{Adv}.
    3. two identities ID_0 and ID_1, which are calculated as: a random bit $b \in \{0, 1\}$ is chosen and the followings have been calculated.

$$ID_b = H_1(PUF_N(C_1)||PUF_N(C_2)||TS)$$
$$ID_{1-b} = h \in_R G_1^*$$

Security Proof of the Proposed Scheme

The Eavesdropping Authentication and Key Exchange Experiment $\mathbf{Auth_{adv,\pi}(n, \zeta, PUF_{Adv}, ID_0, ID_1)}$:

1. The adversary *Adv* is provided:
   1. $\zeta =< C_1,\ C_2,\ TS' >$ where $TS' = ((PUF_N(C_1)||PUF_N(C_2)) \oplus TS)$.
   2. A PUF instance PUF_{Adv}.
   3. two identities ID_0 and ID_1, which are calculated as: a random bit $b \in \{0, 1\}$ is chosen and the followings have been calculated.

$$ID_b = H_1(PUF_N(C_1)||PUF_N(C_2)||TS)$$
$$ID_{1-b} = h \in_R G_1^*$$

2. The adversary *Adv* will output a value b'.

Security Proof of the Proposed Scheme

The Eavesdropping Authentication and Key Exchange Experiment $\mathbf{Auth_{adv,\pi}(n, \zeta, PUF_{Adv}, ID_0, ID_1)}$:

1. The adversary Adv is provided:
   1. $\zeta =< C_1,\ C_2,\ TS' >$ where $TS' = ((PUF_N(C_1) || PUF_N(C_2)) \oplus TS)$.
   2. A PUF instance PUF_{Adv}.
   3. two identities ID_0 and ID_1, which are calculated as: a random bit $b \in \{0, 1\}$ is chosen and the followings have been calculated.

$$ID_b = H_1(PUF_N(C_1) || PUF_N(C_2) || TS)$$

$$ID_{1-b} = h \in_R G_1^*$$

2. The adversary Adv will output a value b'.
3. The adversary Adv succeeds in the experiment if she can distinguish between the "correct" ID and the random one.

Security Proof of the Proposed Scheme

Theorem

The authentication and key exchange protocol π is secure in the presence of eavesdropping adversaries if the 2-Decisional Uniqueness Problem Assumption holds.

We can show that the protocol π is secure if:

$$Pr[Auth_{adv,\pi} = 1] \leqslant \frac{1}{2} + negl(n)$$

Let the adversary *Adv* has some non-negligible advantage ε in breaking the protocol π. Then we can construct an algorithm $\mathcal{B}$ which will have the same advantage ε in breaking the 2-Uniqueness problem. But, due to the hardness 2-Uniqueness Problem, ε should be negligible.

Security Proof of the Proposed Scheme

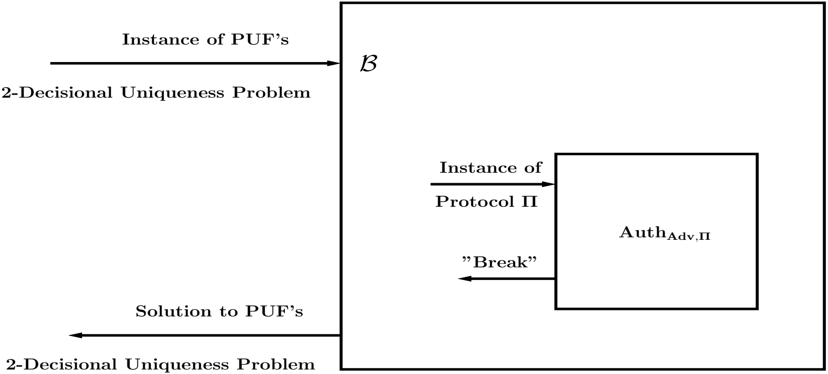


Figure : The view of $Auth_{adv,\pi}$ when it is run as a sub-routine of $\mathcal{B}$ (referred to [Katz and Lindell 2007]).

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide *Adv* with PUF_{Adv}.

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide Adv with PUF_{Adv}.
2. **Input tuple**:

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide Adv with PUF_{Adv}.
2. **Input tuple**:
    1. It randomly chooses TS.

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide *Adv* with PUF_{Adv}.
2. **Input tuple**:
    1. It randomly chooses TS.
    2. $TS' = (z_1||z_2) \oplus TS$.

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide Adv with PUF_{Adv}.
2. **Input tuple**:
    1. It randomly chooses TS.
    2. $TS' = (z_1||z_2) \oplus TS$.
    3. $\zeta =<C_1, C_2, TS'>$, random to the adversary Adv.

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide Adv with PUF_{Adv}.
2. **Input tuple**:
    1. It randomly chooses TS.
    2. $TS' = (z_1 || z_2) \oplus TS$.
    3. $\zeta = < C_1, C_2, TS' >$, random to the adversary Adv.
    4. $b \in \{0,1\}$.

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide Adv with PUF_{Adv}.
2. **Input tuple**:
    1. It randomly chooses TS.
    2. $TS' = (z_1||z_2) \oplus TS$.
    3. $\zeta =<C_1, C_2, TS'>$, random to the adversary Adv.
    4. $b \in \{0,1\}$.
    5. $ID_b = H_1(z_1||z_2||TS)$ and
       $ID_{1-b} = h \in_R G_1^*$

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide *Adv* with PUF_{Adv}.
2. **Input tuple**:
    1. It randomly chooses TS.
    2. $TS' = (z_1||z_2) \oplus TS$.
    3. $\zeta =< C_1, C_2, TS' >$, random to the adversary *Adv*.
    4. $b \in \{0,1\}$.
    5. $ID_b = H_1(z_1||z_2||TS)$ and
       $ID_{1-b} = h \in_R G_1^*$
    6. $\mathcal{B}$ provides *Adv* the input tuple $< \zeta, ID_0, ID_1 >$. If $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$, then ID_b will be equal to $H_1(PUF_N(C_1)||PUF_N(C_2)||TS)$. Otherwise, ID_0, ID_1 both will be some random element of G_1^*.

Security Proof of the Proposed Scheme

Input to Algorithm $\mathcal{B}$: $(C_1, C_2, PUF_{Adv}, z_1, z_2)$ (where $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$ or two random string belongs to $\{0,1\}^*$).

1. **SetUp**:Provide *Adv* with PUF_{Adv}.
2. **Input tuple**:
    1. It randomly chooses TS.
    2. $TS' = (z_1||z_2) \oplus TS$.
    3. $\zeta =<C_1, C_2, TS'>$, random to the adversary *Adv*.
    4. $b \in \{0,1\}$.
    5. $ID_b = H_1(z_1||z_2||TS)$ and $ID_{1-b} = h \in_R G_1^*$
    6. $\mathcal{B}$ provides *Adv* the input tuple $< \zeta, ID_0, ID_1 >$. If $z_1 = PUF_N(C_1)$ and $z_2 = PUF_N(C_2)$, then ID_b will be equal to $H_1(PUF_N(C_1)||PUF_N(C_2)||TS)$. Otherwise, ID_0, ID_1 both will be some random element of G_1^*.
3. **Guess**: *Adv* returns b', a guess of b. If $b = b'$, $\mathcal{B}$ returns 1, otherwise, it returns 0.

Security Proof of the Proposed Scheme

- Once the authentication is done successfully, the data node N calculates its $\{$public,private$\}$ key pair $K1_{PUB} = t \cdot P_1$ and $K1_{PRV} = t \cdot ID_1$ and sends $K1_{PUB}$ to the server over the authenticated link.

Security Proof of the Proposed Scheme

- Once the authentication is done successfully, the data node N calculates its $\{$public,private$\}$ key pair $K1_{PUB} = t \cdot P_1$ and $K1_{PRV} = t \cdot ID_1$ and sends $K1_{PUB}$ to the server over the authenticated link.
- Now assuming the complexity of the *Computational Discrete Log Problem*, the probability that the adversary *Adv* can retrieve the value of t from $K1_{PUB}$, knowing the value of ID_1 is negligible.

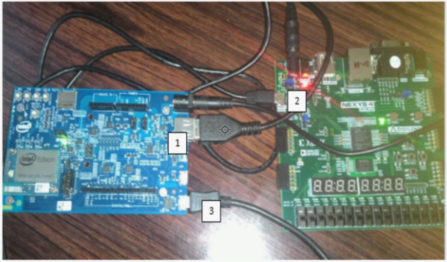
Security Proof of the Proposed Scheme

- Once the authentication is done successfully, the data node N calculates its $\{$public,private$\}$ key pair $K1_{PUB} = t \cdot P_1$ and $K1_{PRV} = t \cdot ID_1$ and sends $K1_{PUB}$ to the server over the authenticated link.
- Now assuming the complexity of the *Computational Discrete Log Problem*, the probability that the adversary *Adv* can retrieve the value of t from $K1_{PUB}$, knowing the value of ID_1 is negligible.

Theorem

Based on the complexity assumption of the Computational Discrete Log Problem and that the hash function is collision resistant, the authentication and key-exchange protocol π is SK-secure in AM as well as in UM model.

Experimental Setup



Components	No. of Slices	No. of Registers	No. of LUTs
6-Stage 64 Bit LSPUF	776	12	986
64 Bit LFSR	16	48	6
48 Bit Shift Register	15	48	3

Components	Execution Time (in sec)	Clock Cycles
Tate Pairing	0.02019	100950
H_1	0.071886	359430
H_2	0.000043	22
H_3	0.000134	670
H_4	0.00007	350

Future Work

1. To design the security protocol which will ensure mutual authentication even if the server is compromised.
2. To design new test beds for emerging IoT applications, explore the vulnerabilities in them and merge our proposed security protocols to provide an overall robust and secure solution.


TIME FOR QUESTIONS
12
1
2
3
4