A Lightweight Authentication and Inter-cloud Payment Protocol for Edge Computing

Mohamed Seifelnasr, Mouna Nakkar, Amr Youssef
Concordia Institute for Information Systems Engineering
Concordia University, Montreal, QC Canada

Riham AlTawy
Department of Electrical and Computer Engineering
University of Victoria, Victoria, BC, Canada

Abstract—In this paper, we propose a lightweight mutual authentication and inter-cloud redeemable payment protocol which allows IoT devices to subscribe with their home cloud service providers for roaming coverage. More precisely, such devices acquire authenticated payment tokens in order to benefit from the computation offloading services from edge nodes deployed by foreign cloud service providers. Hence, IoT devices are continuously serviced even when outside of their home cloud providers coverage. The protocol makes use of tree of secrets, hash chains, and Merkle trees. It requires sharing a Merkle tree root and a 128-bit secret key for constructing the tree of secrets among cloud admins. Our protocol provides mutual authentication, confidentiality, and easy charge redemption from the home server. For $N$ subscribed IoT devices, the storage at the hosting clouds is limited to $2 \times (N_s + 1) \times 16$ bytes and $32 \times \log N$ bytes for the IoT device, where $N_s$ is the maximum number of devices served by the IoT gateway per payment redemption period.

Index Terms—Internet of Things, Edge Computing, Computation Offloading, Mutual Authentication, Micropayment

I. INTRODUCTION

The broad deployment of the Internet of Things (IoT) in many applications, e.g., healthcare systems [1], [2], transportation [3], and smart homes [4] has prompted the adoption of the three-tier Cloud-Edge-IoT architecture [5], [6]. Such a paradigm enables the limited-resource IoT devices to handle huge data volumes and computation requirements through higher-tier computation offloading (CO). Edge computing is a distributed computing paradigm that lies between cloud servers and IoT devices which can then outsource some of their computation to the edge nodes. However, this new paradigm faces new security challenges that need to be addressed by new mutual authentication protocols between IoT devices and the edge nodes. For example, given the mobility of such devices, roaming service coverage are essential to adhere to the expected quality of service which introduces challenges to achieving mutual authentication. Additionally, if IoT devices are able to utilize foreign edge nodes for computation offloading, we need to ensure that such nodes get fairly compensated, and most importantly, using computationally efficient techniques. Our contributions can be summarized as follows: (i) We propose an efficient lightweight mutual authentication protocol that enables mobile IoT devices (e.g., wearable devices) to enjoy CO service from other hosting cloud admins. IoT devices subscribe with home cloud admin through acquiring authenticated payment tokens. Then, when an IoT device requires a service outside of the home cloud server coverage, the IoT provides edge nodes from the foreign hosting cloud admins with a proof of subscription and a payment token in return for the required CO units. The protocol achieves confidentiality, integrity, conditional traceability, and guaranteed redemption of CO charges to the serving entities. (ii) The proposed scheme reduces the storage needed at the IoT gateways by sharing only one secret key among cloud admins for building the tree of secrets. Additionally, we utilize one-way chains for tracking the number of CO services provided by the IoT gateway such that the correct charges are billed to the user.

II. RELATED WORK

Authentication protocols for edge computing can be divided into elliptic curve cryptosystems (ECC) based protocols and efficient-lightweight symmetric key based protocols [7]–[10]. ECC-based protocols rely on the hardness of the discrete log problem. These protocols incur pairing operations which makes them suitable for applications like smart vehicles, and smart grid [11]–[14]. Efficient and lightweight protocols are suitable for limited-resource IoT devices. Ibrahim proposed Octopus [15] for lightweight mutual authentication between the edge user and edge server. However, Octopus suffers from the linkability problem between the messages of the edge user. Wang et al. proposed LAMANCO [16] where IoT device is anonymously authenticated to the edge node. LAMANCO allows the registered IoT device to subscribe for a predefined computation units with the cloud admin (CA), and the preimage value in the IoT hash chain serves as a payment token to the edge nodes. However, the installation of the tamper-proof devices in the IoT devices and the edge nodes limits the practical deployment of the protocol.

III. SYSTEM MODEL, ENTITIES, AND DESIGN GOALS

System Model: Our system model adopts a three-tier hierarchy system as shown in Fig. 1. The entities in our model are (i) CA who is deployed by the cloud service provider and is responsible for the registration of the IoT devices and managing the subscriptions of IoT-edge CO services, (ii) IoT gateways who are deployed by the cloud service provider, and (iii) IoT devices who are resource-constrained devices with Internet accessibility. Occasionally, these devices need
to outsource some of their computation and storage to the IoT gateway to meet its application quality-of-service.

Fig. 1: The considered IoT-Edge-Cloud paradigm.

**Threat Model:** We consider (i) a trusted and secure CA, (ii) a trusted and secure IoT gateway which is securely connected to CA, (iii) a malicious IoT device which tries to obtain CO services without a valid subscription or impersonate other devices, and (iv) external adversaries who maybe curious to breach the transacting entities identities and information or impersonate other legitimate entities or perform replay attacks for achieving denial-of-service over the IoT gateway of the hosting cloud.

**Design Goals:** The design goals of our proposed protocol are (i) mutual authentication between the IoT device and the IoT gateway of the hosting cloud, (ii) confidentiality and integrity of the exchanged messages, (iii) traceability of the misbehaved IoT device to its real identity, (iv) efficiency of the protocol and its freedom of complex public key cryptographic operations, and (v) guaranteed token redemption where IoT gateway of the hosting cloud has a verifiable valid payment token for the CO service offered to the requesting IoT.

**IV. PROTOCOL DESIGN AND IMPLEMENTATION**

We consider two cloud admins: a home cloud admin $CA_h$ where the IoT devices register and subscribe, and a hosting cloud admin $CA_v$ as shown in Fig. 1. When an IoT device $IoT_h$ registered with $CA_h$ moves to another district out of the coverage service of $CA_h$, but with some deployed IoT gateways for $CA_v$, the $IoT_h$ authenticates itself to the IoT gateway of the hosting cloud. To validate the IoT subscription with the home cloud admin, $CA_h$, $IoT_h$ provides a verifiable payment token and a subscription proof with the home cloud admin to $CA_v$. $CA_v$ verifies that the payment token covers the required CO units by the IoT device and validates its subscription proof. In the redemption interval, $CA_v$ collects the tokens from the serving IoT gateways and redeems the CO charges from $CA_h$. Our protocol runs as follows.

1) **Registration and Set-Up Phase:** $CA_h$ generates two random keys $S$, and $K$ where $S$ is a secret key used to generate the tree of secrets [17] and $K$ is a key known only by $CA_h$. Subsequently, $CA_h$ builds up a tree of secrets for the registered IoT devices as shown in Fig. 2. For each $IoT_h$, $CA_h$ sends the secrets along the path from the IoT device leaf to the root (i.e., for $IoT_{010}$ of index $i = 010$, $CA_h$ sends the derived keys $S_0$, $S_{01}$, $S_{010}$).

![Tree of secrets for 8 IoT devices pseudo-identities](image)

These secrets are used later to generate temporary pseudo-identities of $IoT_h$ for authentication with the IoT gateway of $CA_v$. Additionally, for each IoT device, the cloud admin $CA_h$ creates a hash chain [18] with seed $C_i = H(K|||S_i)$ and the tip $T_i = H^L(C_i)$ where $L$ is the length of the hash chain. Then, $CA_h$ accumulates the tips of all the IoT devices hash chains in a single Merkle tree [19]. Fig. 3 shows an example for accumulating the tips of 8 IoT devices in a single Merkle tree. $CA_h$ sends to each IoT device $C_1$, $T_i$, $π_i$ and the secrets along the path to the root on the tree of the secrets where $π_i$ is the Merkle proof associated with $T_i$ in the $CA_h$ Merkle tree, e.g., for IoT device, $IoT_{010}$ in Fig. 2, the secrets are $S_0$, $S_{01}$, $S_{010}$, the seed is $S_{010}$ and Merkle path is $010 = (H(T_{011}), H_{00}, H_{11})$. On the other side, $CA_h$ sends to $CA_v$ the secret key $S$ and the Merkle root $M_r$. $CA_v$ uses the secret key $S$ to decode the temporary pseudo-identities of the IoT devices in the mutual authentication phase, while $M_r$ is used for verifying the Merkle proof of the IoT tip $T_i$ [20].

2) **Mutual Authentication Phase:** The IoT device $IoT_{010}$ sends a Hello Message to the IoT gateway $G_v$ as shown in Fig. 4. Then, the IoT gateway challenges the IoT device with a random $r$, to which $IoT_{010}$ responds with the temporary pseudo-identity $P$ and challenges the IoT gateway with a nonce $N_1$, where $(P = E_{S_0}(r), E_{S_{01}}(r), E_{S_{010}}(r))$ is the encryption of the challenge $r$ with the secrets of the IoT device. When the IoT gateway receives $P$, the IoT gateway decodes the temporary pseudo-identity of the IoT device starting from the top, root $S$, and going downward by checking the encryption of the challenge $r$ under each of its child nodes keys, following the matching path until it reaches the corresponding pseudo-identity of the IoT device (e.g. $S_{010}$). Subsequently, the IoT gateway increments the IoT device nonce, $N_1$, and sends it back to the IoT device, $IoT_{010}$, encrypted with the secret $S_{010}$.

3) **Message Exchange Phase:** Using the nonce $N_2$, the IoT device and the IoT gateway derive a session key $k_s = H(N_2|||S_i)$ as shown (e.g. for IoT device $IoT_{010}$, $S_i = S_{010}$) in Fig. 5. Then, the IoT device and IoT gateway exchange the message encrypted using an authenticated encryption scheme with the session key $k_s$.

4) **CO Charges Redemption Phase:** At the end of each redemption interval, $CA_v$ sends the set of the pseudo-identities...
Hvix | Sk′, verify hic)′H) | s (Msv010010N
TNTTHTOH010π(s)|Xr)P|r=N(n

Compute (C o m p u t e ) = H( || )K s N2 S 010
||{N2 M 1}K s
ComputeX
s. t. Hn(X) = T010 n||E s010(π010|T010
verify( ) rootπ010 =?
( X )H n =? T010

Fig. 5: Authenticated encryption over exchanged messages.

Fig. 6: Token X similic redemption in return for n′ CO units.

V. SECURITY ANALYSIS

1) Mutual Authentication: The IoT is authenticated to the IoT gateway of CAh by sending its temporary pseudo-identity P which is generated using the tree of secrets. Paths of two IoT devices are different in at least one secret. Thus, the probability of IoT impersonating IoT is negligible unless it knows all the secrets. Similarly, the IoT authenticates the IoT gateway by challenging it with N1. Legitimate IoT gateway decodes the IoT temporary pseudo-identity P and knows the corresponding IoT secret Si and encrypts its response N1 + 1 using the corresponding IoT device secret.

2) Confidentiality and Integrity: Assume a semantic secure authenticated encryption scheme, the confidentiality and the integrity of the messages between IoT device and the IoT gateway are guaranteed.

3) Guaranteed Token Redemption: In the mutual authentication phase, IoT sends a pre-image value in its IoT hash chain as a payment token in addition to the Merkle proof of the tip to the IoT gateway of CAh. Unsubscribed IoT device can not produce a valid Merkle proof in the CAh Merkle tree. Also, for checking that the IoT subscription covers the amount of CO units required, the IoT gateway verifies that Hn(X) = T. In the redemption phase, CAh pays back the charges of n CO units. The CAh computes Ci = H(K||Si) and checks the payment token in the IoT hash chain. Assuming that K is known only to CAh, the IoT gateway has a negligible probability to find a new pre-image value X′ such that Hn(X′) = T.

4) Resilience to Replay Attack: The protocol counters replay attack by incurring the challenge r and the nonce N1 in the mutual authentication phase between the IoT device and IoT gateway and nonce N2 in the key establishment phase.

5) User Pseudo-anonymity: The IoT device registers with CAh using its real identity, and CAh uses the tree of secrets in order to encode their pseudo-identities Si. CAh shares with the IoT gateway the secret key S for constructing the tree of secrets. Thus, the other cloud admin CAc can build up the tree of secrets to decode the IoT temporary pseudo-identity P to the IoT device pseudo-identity Si without disclosing the real identity of the IoT device.

6) User Traceability: The IoT gateway can send the pseudo-identity of the misbehaved IoT device to CAh which can trace it back to its real identity.

VI. PERFORMANCE EVALUATION

To maintain a security level of 128 bits, we assume r, nonces, keys and all the secrets to be 128 bits. We also assume the length of n is 1 byte, the authenticated encryption tag to be 160 bits, and the message to be of an arbitrary length. The performance evaluations are summarized in Table I.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT storage</td>
<td>(1 + 2 × Log N) × 16 bytes</td>
</tr>
<tr>
<td>Gateway storage</td>
<td>2 × (N1 + 1) × 16 bytes</td>
</tr>
<tr>
<td>IoT computation</td>
<td>(L − n) hash, (1 + Log N) Enc, 1 Dec</td>
</tr>
<tr>
<td>Gateway computation</td>
<td>(n + 1 + Log N) hash, (1 + 2 × Log N) Enc, 3 Dec</td>
</tr>
<tr>
<td>Communication overhead</td>
<td>117 + 32 × Log N bytes</td>
</tr>
</tbody>
</table>

VII. CONCLUSION

We proposed a mutual authentication protocol between IoT device and multi-cloud providers. The storage requirement for the IoT gateway is 32 bytes / IoT device. The subscribed CA shares only a 128-bit key for authenticating the IoT temporary pseudo-identity as well as the Merkle root of the tips of the IoT hash chains. By using tree of secrets and Merkle tree, we presented a methodology through which the hosting CA can redeem the computation offloading charges based on the subscription of the IoT device and its accumulation in the Merkle tree.