Achieving Communication-Efficient Privacy-Preserving Query for Fog-Enhanced IoT

Nafiseh Izadi Yekta (Master Student) and Rongxing Lu*

Contact Email: rlu1@unb.ca

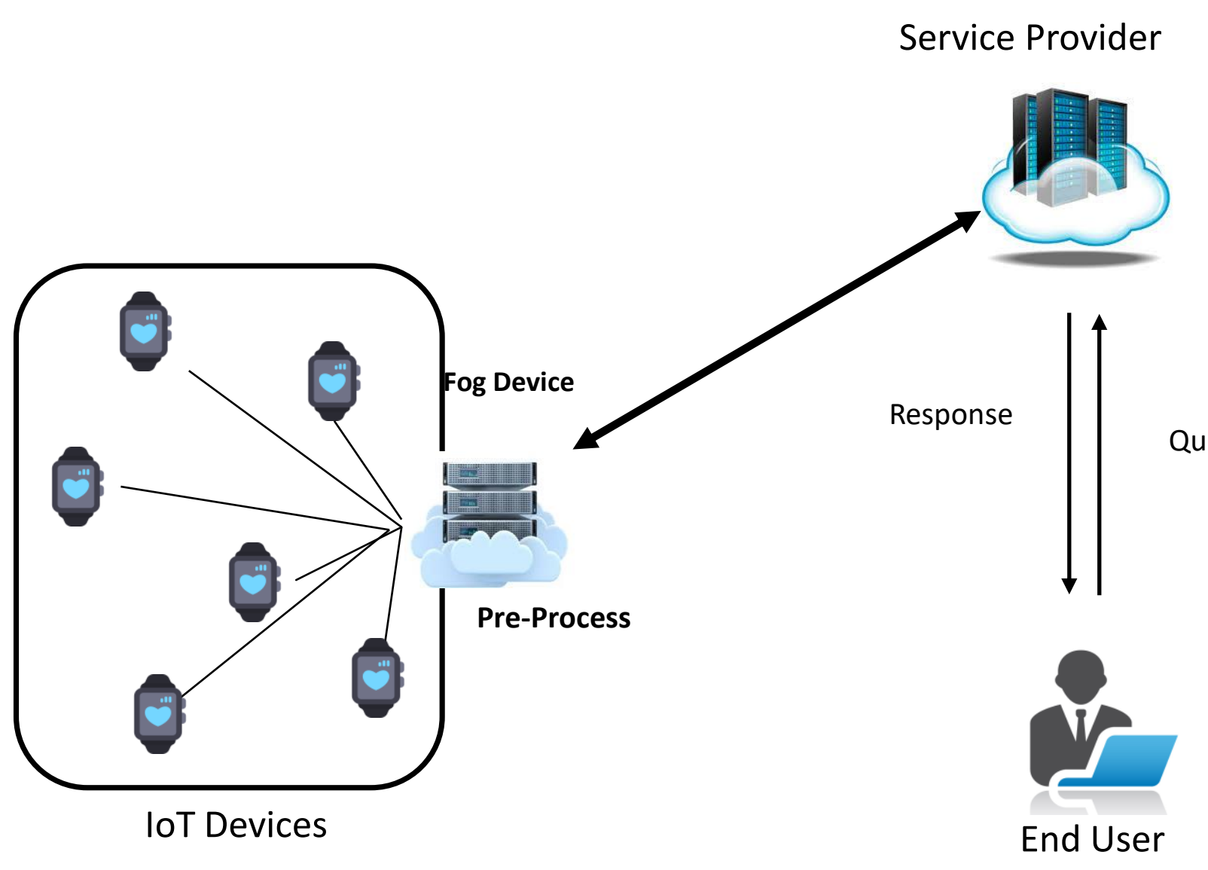
Canadian Institute for Cybersecurity (CIC), University of New Brunswick (UNB)

ABSTRACT

Internet of things (IoT) has attracted significant attention in recent years, and various IoT devices including industrial and utility components and other items embedded with electronics, sensors, and network connectivity have already provided rich services to the end users. Nevertheless, IoT still faces many security and privacy challenges. In this paper, we propose a new efficient privacy-preserving query scheme, called XRQuery, for fog computing-enhanced IoT. The proposed XRQuery scheme is characterized by employing a new communication-efficient private information retrieval technique, which can preserve the privacy for both the end user and the service provider in IoT query service. Detailed security analysis shows the XRQuery scheme really preserves the privacy. In addition, extensive performance evaluation also indicates XRQuery can vastly reduce the communication overheads between the fog device and the end user in fog computing-enhanced IoT.

System Model

- **IoT devices:** a set of IoT devices $D = \{D_1, D_2, \cdot\cdot\cdot, D_n\}$ are deployed at an area of interest.
- **Fog device:** is deployed at the network edge, which receives the data reported from IoT devices
- **Service provider:** is a server deployed at a cloud platform.
- **End user:** is an IoT service requester in our model.



Design Goals

- The proposed scheme should be privacy-preserving.
- The proposed scheme should be communication efficient.

1. System Initialization

- As both the fog device and IoT devices $D = \{D_1, D_2, \cdot\cdot\cdot, D_n\}$ are affiliated with the service provider, it is reasonable to assume the service provider to bootstrap the whole system. For the ease of description, we consider n = 2^b so that the binary representation of n is b bits. b

Symmetric Homomorphic Encryption

- Key generation: Given the security parameter λ, parameters (s, p, q) will be generated by KeyGen(λ)

  (s, q, p) → KeyGen(λ)
- **Encryption**: E(SK, m, d) = s^d (rq + m) mod p
- **Decryption**: D(SK, c, d) = (c × s^{-d}mod p) mod q
- **Homomorphic Multiplication:**

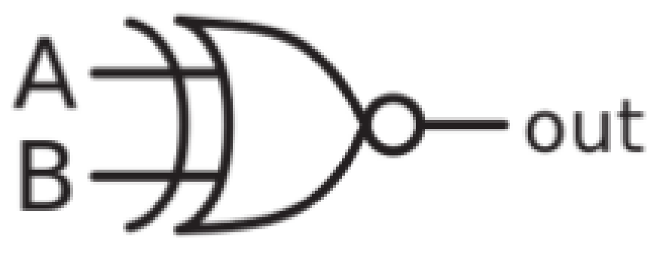
  $(c_1 \times c_2)\ mod\ p = s^{d1+d2}((r_1 r_2 q + r_1 m_2 + m_1 r_2)\ q + m_1 \times m_2)\ mod\ p$
- **Homomorphic Addition:**

  $(c_1 + c_2)\ mod\ p\ = s^{d1}((r_1 + r_2)q + m_1 + m_2)\ mod\ p$
- **Homomorphic Scalar Multiplication:**

  $(c_1 \times m_2)\ mod\ p\ = s^{d1}(r_1 .m_2 .q + m_1 \times m_2)\ mod\ p$

The XNOR Gate

The XNOR gate or Exclusive NOR gate is a digital logic gate with two or more inputs and one output that exerts logical equality.



(a) Distinctive symbol

Input		Output
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	1
Boolean Expression : $\overline{A \oplus B}$		

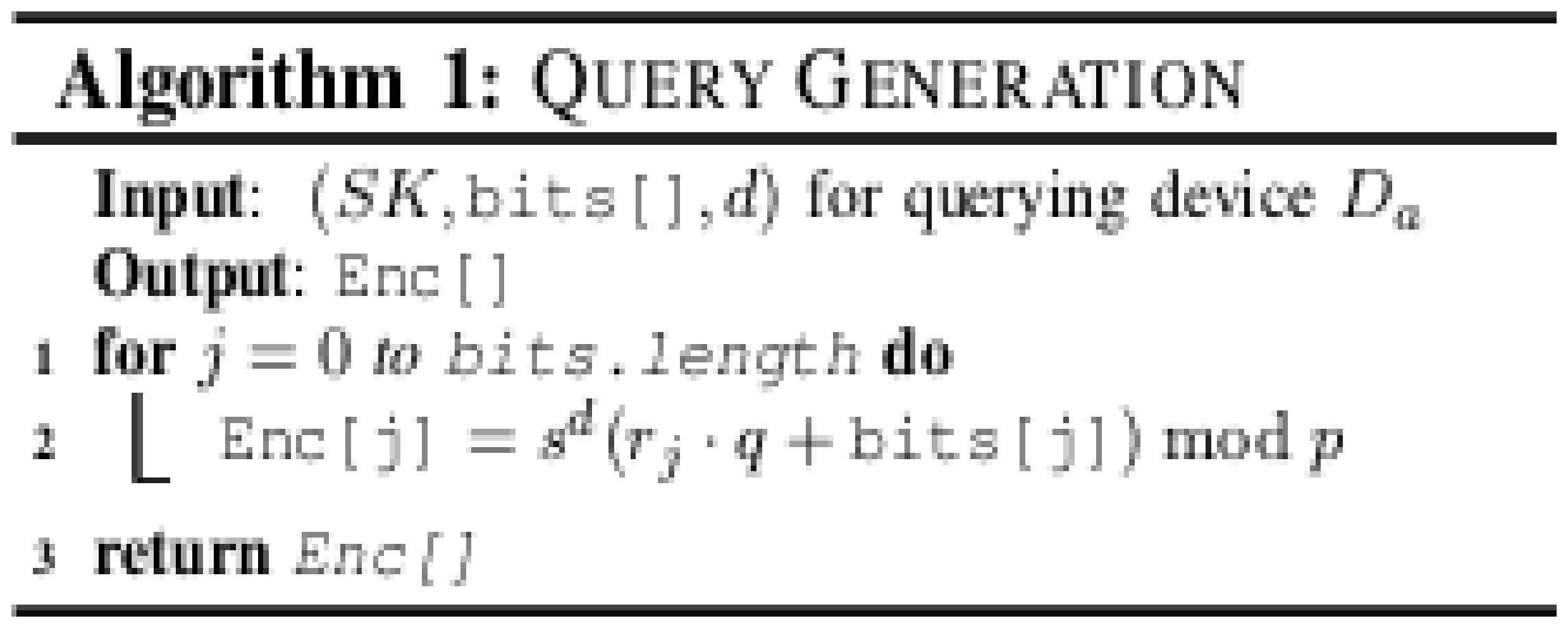
(b) Truth Table

2. End User Query

- End user's query (Enc(1), Enc[])
- **Enc[] = {Enc[0], Enc[1], · · · , Enc[m]}**

where m is equal to log(n)

- Encrypts the value 1 as **Enc(1)**

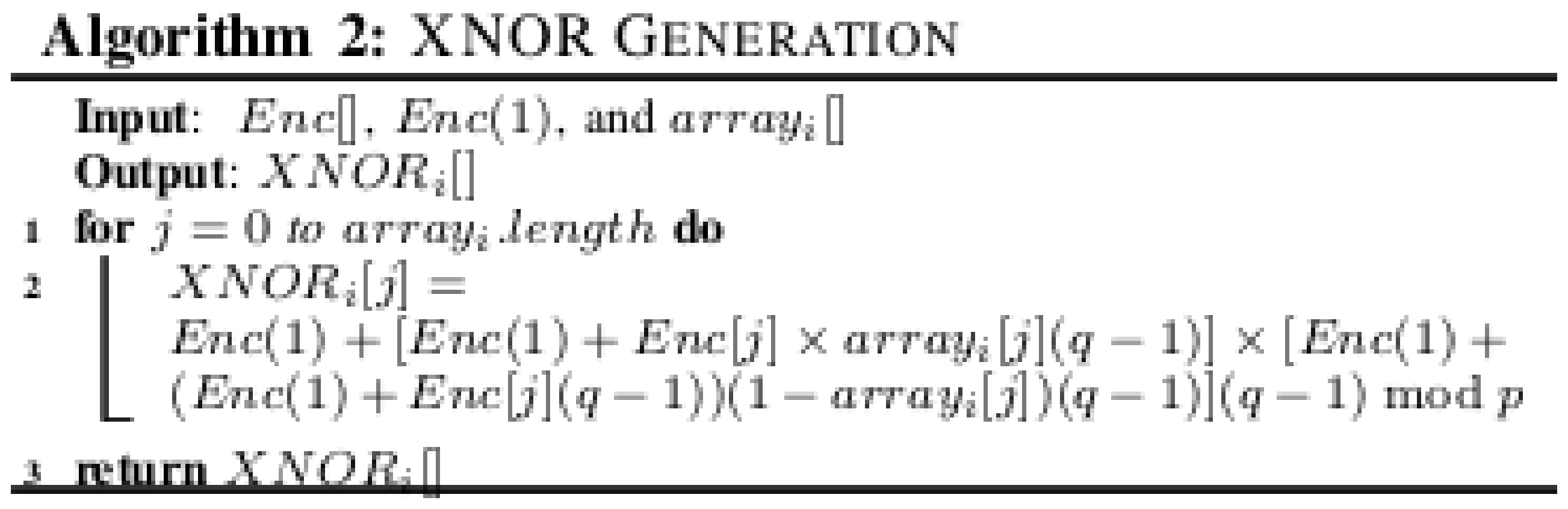


Algorithm 1: QUERY GENERATION

```
Input: $(SK, bits[], d)$ for querying device $D_a$
Output: Enc[]
for $j = 0$ to bits.length do
    $Enc[j] = s^d(r_j \cdot q + bits[j])$ mod $p$
return Enc[]
```

3. Fog Device Response

- Step 1 : For each IoT device D_i, the fog device compute the XNOR with homomorphic properties.



Algorithm 2: XNOR GENERATION

```
Input: $Enc[]$, $Enc(1)$, and $array_i[]$
Output: $XNOR_i[]$
for $j = 0$ to $array_i.length$ do
    $XNOR_i[j] = Enc(1) + [Enc(1) + Enc[j] \times array_i[j](q-1)] \times [Enc(1) + (Enc(1) + Enc[j](q-1))(1 - array_i[j])(q-1)](q-1)$ mod $p$
return $XNOR_i[]$
```

- Step 2 :For each IoT device, Multiply all XNOR $_i$ [j] to each other to achieves the AND operation.
- Step 3: Multiply each IoT device's value x $_{it}$ in time slot t to the output of the AND operation.
- Step 4: Add all V $_i$ together and sends it to the service provider.

4. End User Result Checking

The end user with underlying decryption algorithm can decrypt the received value. Since only one f_i for i = a will be 1 and other values of f_i for i ≠ a will be 0 the decrypted value is the IoT device's value that the end user desires.

Conclusion

- XRQuery is inspired by XNOR gates in logical circuits to achieve privacy preservation for both service provider and user in an IoT query service.
- XRQuery is super efficient in term of communication cost i.e., achieving O(log n) between the end user and the fog device.
- The extensive performance evaluations show it is very efficient in terms of computational cost.

Future Work

For future work, I want to investigate other areas of private information retrieval to assess the possibility of using the same technique to improve the state of the art of those areas.