# Achieving Privacy-Preserving Query with Communication Efficiency in Internet of Things

**Nafiseh Izadi Yekta and Rongxing Lu**

*Email: nizadi@unb.ca; rlu1@unb.ca*

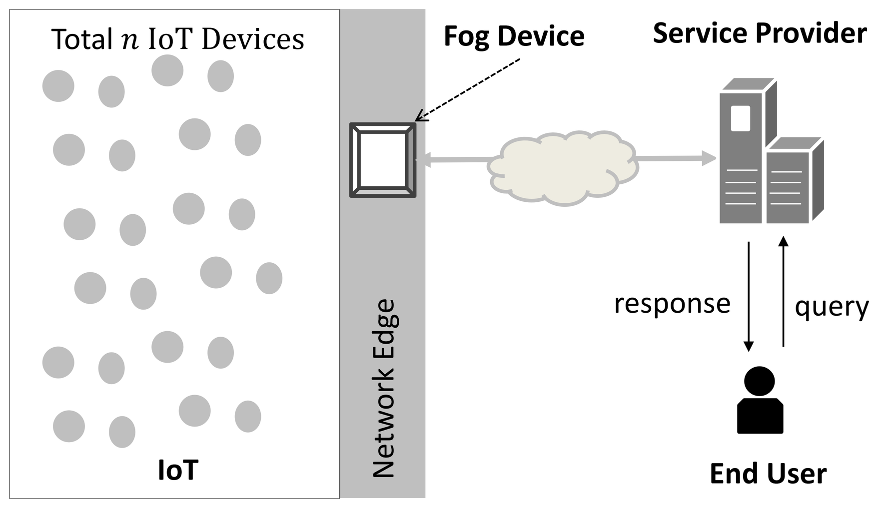
***Canadian Institute for Cybersecurity (CIC), University of new Brunswick (UNB)***

## ABSTRACT

Internet of Things (IoT), as it can provide many promising IoT services to end users, has received considerable attention in recent years. However, IoT's security and privacy are still challenging today. In this paper, we propose a privacy-preserving query scheme, called PQuery, for fog computing-enhanced IoT. The proposed PQuery scheme is characterized by employing two privacy enhancing techniques, i.e., private information retrieval and oblivious transfer, to preserve the privacy for both the end user and the service provider in IoT query service. Though the computational cost is still high at the fog device, the communication overheads in PQuery can be greatly reduced between the fog device and the end user.

## System Model

- **IoT devices:** a set of IoT devices $D = \{D_1, D_2, \cdots, D_n\}$ 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 $D = \{D_1, D_2, \cdots, D_n\}$ are affiliated with the service provider, it is reasonable to assume the service provider to bootstrap the whole system. In order to make the communication efficient, the service provider first arrange IoT devices $D = \{D_1, D_2, \cdots, D_n\}$ into a cube, where the length of each edge is m = $\sqrt[3]{n}$, and each IoT device then can be identified as $D_{ijk}$, where $i, j, k \in \{1, 2, \cdots, m\}$.

## BGN Homomorphic Encryption

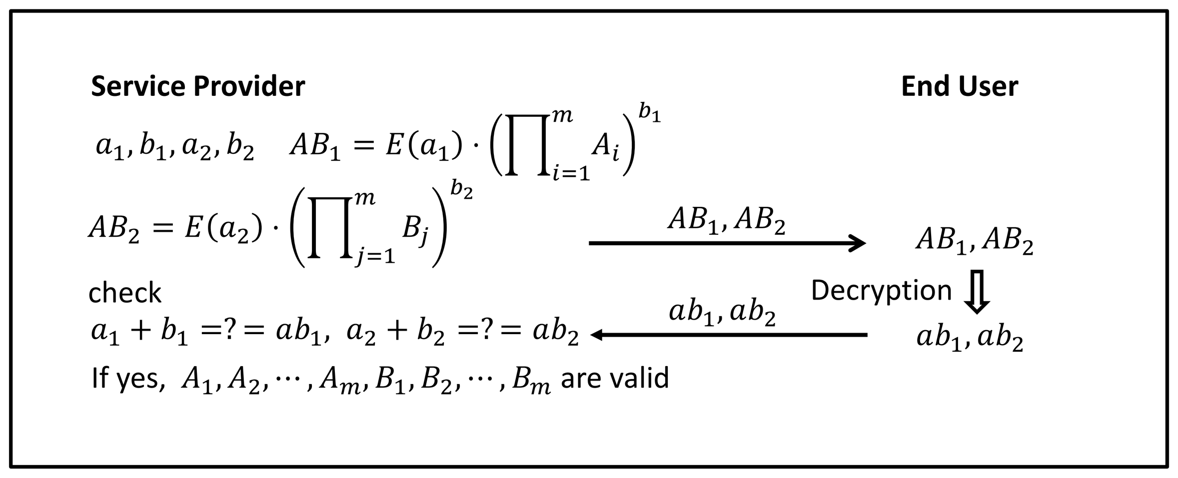
- Addition in $G$: Given $E(m_1; r_1) \in G$ and $E(m_2; r_2) \in G$, we have $E(m_1; r_1) \cdot E(m_2; r_2) = E(m_1 + m_2; r_1 + r_2) \in G$. For simplicity, we omit the random items, and we have $E(m_1) \cdot E(m_2) = E(m_1 + m_2)$.
- Multiplication in $G$: Given $E(m_1; r_1) \in G$ and $m_2 \in S$, we have $E(m_1; r_1)^{m2} = E(m_1 \cdot m_2; r1 \cdot m2) \in G$. For simplicity, we have $E(m_1)^{m2} = E(m_1 \cdot m_2)$.
- Multiplication from $G$ to $G_T$: Given $E(m_1); E(m_2) \in G$, we have $e(E(m_1); E(m_2)) = E_T(m_1 \cdot m_2) \in G_T$, where $E_T(\cdot)$ denotes a ciphertext in $G_T$.
- Addition in $G_T$: Given $E(m_1); E(m_2) \in G_T$, we have $E_T(m_1) \cdot E_T(m_2) = E_T(m_1 + m_2)$.
- Multiplication in $G_T$: Given $E_T(m_1) \in G_T$ and $m_2 \in S$, we have $E_T(m_1)^{m2} = E_T(m_1 \cdot m_2)$.

## 2. End User Query

**Algorithm 1:** QUERY GENERATION

**Input**: $d$ and $(m, a, b, c)$ for querying device $D_{abc}$
**Output**: $(A_1, A_2, \cdots, A_m); (B_1, B_2, \cdots, B_m); E_T(N-d)$

**for** $i = 1$ *to* $m$ **do**
  $A_i \leftarrow$ a BGN ciphertext $\begin{cases} E(1) \in \mathbb{G}, & \text{if } i = a; \\ E(0) \in \mathbb{G}, & \text{if } i \neq a. \end{cases}$
**for** $j = 1$ *to* $m$ **do**
  $B_j \leftarrow$ a BGN ciphertext $\begin{cases} E(1) \in \mathbb{G}, & \text{if } j = b; \\ E(0) \in \mathbb{G}, & \text{if } j \neq b. \end{cases}$
generate a BGN ciphertext $E_T(N-d)$;
**return** $(A_1, A_2, \cdots, A_m); (B_1, B_2, \cdots, B_m); E_T(N-d)$




## 3. Fog Device Response

**Algorithm 2:** RESPONSE GENERATION

**Input**: $(A_1, A_2, \cdots, A_m); (B_1, B_2, \cdots, B_m); E_T(N-d)$; the shared key $s$ between fog device and service provider
**Output**: $(C_1, C_2, \cdots, C_m); (F_1, F_2, \cdots, F_m)$

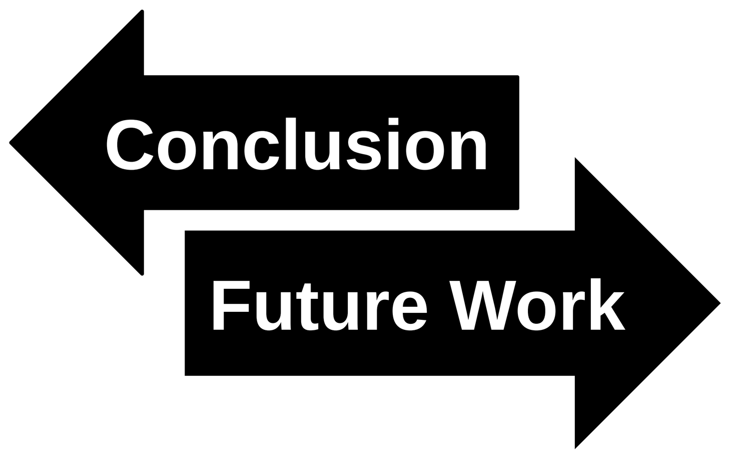
**for** $k = 1$ *to* $m$ **do**
  choose two random $r_k, r'_k \in \mathbb{Z}_N$, $C_k = e(g, g)^{r_k} \in \mathbb{G}_T$,
  $F_k = e(g, g)^{-s \cdot r_k} \cdot \left(\prod_{i,j \in \{1,2,\cdots,m\}} e(A_i, B_j)^{x_{ijkt}} \cdot E_T(N-d)\right)^{r'_k}$
**return** $(C_1, C_2, \cdots, C_m); (F_1, F_2, \cdots, F_m)$

## 4. End User Result Checking

$$e(g,g)^{-s \cdot r_k} \cdot \left(\prod_{i,j \in \{1,2,\cdots,m\}} e(A_i, B_j)^{x_{ijkt}} \cdot E_T(N-d)\right)^{r'_k} = E_T(r'_k(x_{abkt} - d) - s \cdot r_k)$$

$$\begin{aligned} F'_c &= e(g,g)^{r_c \cdot s} \cdot F_c \\ &= e(g,g)^{r_c \cdot s} \cdot E_T(r'_c(x_{abct} - d) - s \cdot r_c) \\ &= E_T(r'_c(x_{abct} - d)) \in \mathbb{G}_T \end{aligned}$$

- PQuery is characterized by combining the private information retrieval and 1-out-of-m oblivious transfer techniques to achieve privacy preservation for both the end user and the service provider in IoT query service.




- For future work, we will explore more functions of fog devices and balance the communication and computational costs in new privacy-preserving IoT query service designs.