PMQ: Achieving Privacy-Preserving Multi Dot-Product Query in Fog Computing-Enhanced IoT

Hassan Mahdikhani and Rongxing Lu

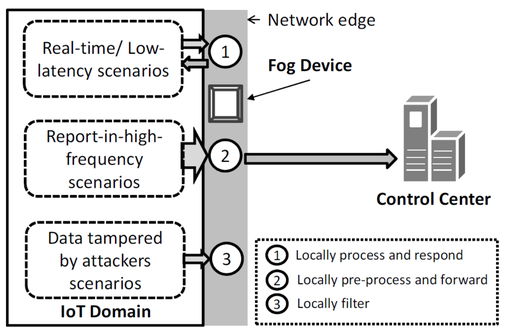
Canadian Institute for Cybersecurity (CIC), University of new Brunswick (UNB)

ABSTRACT

Fog computing-enhanced IoT (Internet of Things), as it can provide better IoT services at the network edge, has received considerable attention in recent years. In this paper, for this new paradigm, we present a new privacy-preserving multi dot-product query scheme, called PMQ, which enables the control center to gain k dot-product results simultaneously in one query. Specifically, in the proposed PMQ scheme, the BGN homomorphic encryption is employed for encrypting query request and response, and a fog device is deployed at the network edge to assist the privacy-preserving k dot-product query. Detailed security analysis shows that the proposed PMQ can achieve better privacy preservation, i.e., no information in query request and response will be disclosed. In addition, extensive simulations are conducted, and the results demonstrate that the proposed PMQ scheme can achieve acceptable efficiency in terms of communication overheads and computational costs.

System Model

- **IoT devices (IoT) in IoT Domain:** a set of IoT devices $D = \{D_1, D_2, \dots, D_n\}$ are deployed at an area of interest with sensing and communication capabilities and report a vector of data $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{im}\}$ periodically reported by $D_i \in D$.
- **Fog device (FD):** is deployed at the network edge, i) locally processes data and respond to the IoTs and ii) pre-processes data and forwards to CC.
- **Control Center (CC):** holds a vector $\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$, queries a subset of IoTs $D^* = \{D_{1*}, D_{2*}, \dots, D_{k*}\} \in D$, and periodically computes dot products $\vec{\alpha}.\vec{\beta}$ in a privacy preserving way.



Design Goals

- **Secure and Privacy-Preserving communication**:
  - Level-1: $\beta, all\ \alpha_i, and\ all\ \beta.\alpha_i$.
  - Level-2: $subset\ D^*$.
- **Efficient communication:** The proposed scheme should be concerned about communication overhead and computation costs.

A. System Initialization

- CC will bootstrap the whole system.
- Given k_1, k_2 security parameters with $|N|=k_1$ and hash function $H: \{0,1\}^* \rightarrow \{0,1\}^{k_2}$.
- CC will generate public key [pk = (N,G,G_T,e,g,h)] and private key (sk = p) of BGN homomorphic encryption.
- CC chooses below random numbers as shared keys for secure communication:
  - K_{fc} between CC & FD.
  - K_{fi} between FD & each IoT device.
  - K_{fb} for secure broadcasting between FD & all IoT Devices.

BGN Homomorphic Encryption

- **Key Generation:** Public Key(pk)=(N, G, G_T, e, g, h), Private Key(sk)=p.
  - Where N=pq, p,q are two k-bit prime numbers, k is a security parameter and composite bilinear parameters (N, g, G, G_T, e) are generated by CGen(k), g is a generator of order n and $h = g^p$ is a random generator of the subgroup of G of order p.
- **Encryption:** $c = E(m,r) = g^m h^r \in G$. / **Decryption:** $c^p = (g^m h^r)^p = (g^p)^m$, To recover m, it is sufficient to compute the discrete log of c^p base g^p.
- **Homomorphic properties:**
  - **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 $E_T(m_1)^{m2} = E_T(m_1 \cdot m_2)$.

B. Control Center Query

Algorithm 1: QUERY GENERATION

Input: β, $|\beta| = m$, σ, $|\mathbb{D}| = n$, $\mathbb{D}^* = \{D_{1*}, D_{2*}, \cdots, D_{k*}\}$
Output: $(A_1, A_2, \cdots, A_n); (B_1, B_2, \cdots, B_m)$

for $i = 1$ *to* n **do**
  if $i == j^*$ *with* $D_{j*} \in \mathbb{D}^* = \{D_{1*}, D_{2*}, \cdots, D_{k*}\}$ **then**
    $A_i \leftarrow$ a BGN ciphertext $E(1) \in \mathbb{G}$
  else
    $A_i \leftarrow$ a BGN ciphertext $E(0) \in \mathbb{G}$
for $j = 1$ *to* m **do**
  $B_j \leftarrow$ a BGN ciphertext $E(\beta_j) \in \mathbb{G}$
return $(A_1, A_2, \cdots, A_n); (B_1, B_2, \cdots, B_m)$

C/I. IoT Devices Response

Algorithm 2: IOT DEVICE RESPONSE

Input: $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{im}\}$ and ReqB, k_{fi}
Output: c_i, mac_{if}, ts

choose a random number r_i and gain the current time stamp ts
compute $c_i = \left(\prod_{j=1}^{m} B_j^{\alpha_{ij}}\right) \cdot h^{r_i}$, $mac_{if} = H(c_i||ts||k_{fi})$
return $c_i||mac_{if}||ts$

C/II. Fog Device Response

Algorithm 3: FOG DEVICE RESPONSE

Input: $(c_1, c_2, \cdots, c_n)$, ReqA $= (A_1, A_2, \cdots, A_n)$, $f()$, k_{fc}
Output: $(C_1, C_2, \cdots, C_k)$, mac_{fc}, ts

set $C_j = 1_{\mathbb{G}_T}$ for all $j = 1, 2, \cdots, k$
for $i = 1$ *to* n **do**
  if $f(i) == j$ *with* $j \in \{1, 2, \cdots, k\}$ **then**
    $C_j = C_j \cdot e(A_i, c_i) \in \mathbb{G}_T$
gain the current time stamp ts
compute $mac_{if} = H((C_1, C_2, \cdots, C_k)||ts||k_{fc})$
return $(C_1, C_2, \cdots, C_k)||mac_{fc}||ts$

D. Control Center Result Reading

After receiving (C1, C2, …, Ck) from fog device

- First of all, the CC will check the validity of received data from fog device, then
- The CC can use the private key (sk=p) to recover each $\alpha_{i*}.\beta$, as each $C_i = e(g,g)^{\alpha_{i*}.\beta}.e(g,h)^{R_i}$, and
- Finally the CC can compute each logistic function $F(\alpha_{i*}) = \frac{1}{1+e^{-(\beta_0+\alpha_{i*}.\beta)}}$.

Conclusion

PMQ is characterized by employing a fog device deployed at the network edge and the BGN homomorphic encryption to achieve privacy-preserving k dot-product query from IoT domain, i.e., the details of the control center's query will not be disclosed to the fog device and IoT devices, and each IoT device's query response is also not seen by the fog device. Through the extensive simulation evaluations, the efficiency of PMQ in term of computational costs is acceptable.

Future work

In our future work, we will examine the PMQ performance in real IoT scenarios, and also put efforts in reducing the communication overhead of the query request.