Private Membership Test Protocol with Low Communication Complexity

**Introduction:** Private membership test (PMT) protocols enable clients to query for a certain item in server’s database without revealing to the server what the item is. Clients are also prevented from learning anything else about the server’s database.

**Protocol:** Our protocol is based on a real world scenario. A server possesses $2^{21}$ malware samples and stores their hash values in 16 different subsets, based on the first 4 bits of the hash value. Each subset has $2^{17}$ items. Client Wants to privately search for an item with hash value equal to $h$.

**Server**
- Divides each subset into 64 segments based on the next 6 bits of the hash value (bits 5,...,10). Each segment has $2^{11}$ items.
- Inserts each segment into a Bloom filter with 10 hash functions. Each filter has $2^{15}$ bits.
- Divides each filter into 16 parts and arranges sixteen $2^3 \times 2^3$ matrices $M$, with parts of the filters as their elements, in such a way that the concatenation of all 16 elements in a particular position of all matrices, results to one of the Bloom filters.

For each of the matrices $M$:
- Computes $\sigma_i = \prod_{j=0}^7 (\beta_j)^{M(i,j)} \mod N^2 = E(M(i,j^*))$.
- Represents $\sigma_i = u_i N + v_i; i \in \{0,1,...,7\}$.
- Computes $u = \prod_{i=0}^7 (\alpha_i)^{u_i} \mod N^2 = E(u_i^*)$.
- Computes $v = \prod_{i=0}^7 (\alpha_i)^{v_i} \mod N^2 = E(v_i^*)$.

![Diagram](image)

- Client spends 1.8s to generate $\sigma_T$ and $\beta_T$.
- Client sends 8 KB data to the Server.

- (Hash functions of the filter) and $(u, v)$ for each matrix

- Server spends 0.9s to generate $u_j$ and $v_j$.
- Server sends 16 KB data to the Client.
- Bloom filter has a false positive rate of 0.001.

**Conclusion:** Homomorphic encryption allows server to search in the matrix without knowledge of client’s keys. The proposed protocol has significantly smaller communication complexity than prior art.

- In order to reduce the false positive rate to 1 out of a million, client can repeat the protocol for the next 6 bits of $h$ (bits 11th-16th).