Secure Computation on Hidden Markov Models
Fattaneh Bayatbabolghani
Department of Computer Science and Engineering, University of Notre Dame

Motivation
- HMM has several applications in pattern recognition such as speaker recognition.
- Personal data need to be protected.

Goals
- To develop privacy-preserving techniques for HMM.
- To develop techniques based on floating point arithmetic.
- To develop techniques for two-party setting based on threshold homomorphic encryption.

HMM
- HMM Consists of:
  - N states S1, ..., SN.
  - M possible outcomes m1, ..., mM.
  - A vector π = (π1, ..., πN) that contains the initial state probability distribution.
  - A matrix A of size $N \times N$ that contains state transition probabilities.
  - A matrix B of size $N \times M$ that contains output probabilities.

HMM Computation

1. Initialization step: for $i = 1$ to $N$
   - $\delta_1(i) = \pi_i \beta_{1i}$
   - $\psi_1(i) = 0$

2. Recursion step: for $k = 2$ to $T$ and $j = 1$ to $N$
   - $\delta_k(j) = \max_{1 \leq s \leq N} [\delta_{k-1}(s) a_{sj}] \beta_{jk}$
   - $\psi_k(j) = \max_{1 \leq s \leq N} [\delta_{k-1}(s) a_{sj}]$

3. Termination step:
   - $P_\star = \max_{1 \leq s \leq N} [\delta_T(s)]$
   - $q_T = \max_{1 \leq s \leq N} [\delta_T(s)]$
   - For $k = T - 1$ to $1$
     - $q_k^\star = \psi_{k+1}(q_{k+1}^\star)$
   - Return $(P_\star, q_\star)$

HMM algorithm

- $P_\star$ as in following equation:
  \[ \beta_{jk} = \sum_{i=1}^{N} a_{ij} e^{-\frac{1}{2} (x_k - \mu_i)\Sigma_i^{-1} (x_k - \mu_i)} \]

- 2- Set $\lambda = \{N, T, \pi, A, \omega, \mu, \Sigma, X\}$
- 3- Execute $(P_\star, q_\star) = Viterbi(\lambda)$
- 4- Return $(P_\star, q_\star)$.

Comparison point operations

**Comparison(LT) of two encrypted integer numbers $enc(x)$ and $enc(y)$**

**Server:**
1. Select $b_1 \in \{0, 1\}, b_2 \in \{0, 1\}, r_1 > r_1'$.
2. Compute $enc(x) = enc(x - y)$.
3. Compute $a_1 = enc(1 - b_1), a_2 = enc(b_1), a_3 = enc(-1)^{r_1} + (-1)^{r_1'}$, and send to Client.

**Client:**
4. Select $b_2 \in \{0, 1\}$,
5. $r_2, r_2' \in \{0, 1\}$, $r_2 > r_2'$.
6. Compute $dec(a_3)$. If it is negative, output is $a_2 - r_2$, otherwise, output is $a_1$.

**Multiplication(Mul) of two encrypted integer numbers $enc(x)$ and $enc(y)$**

**Server:**
1. Choose a random number $r$.
2. Compute $enc(x - r)$, and send to Client.

**Server & Client:**
3. Compute $dec(x - r)$.

**Client:**
4. Compute $enc(y(x - r))$, and send to Server.

**Server:**
4. Compute $enc(yr)$, and send to Client.

**Server & Client:**
5. Compute $enc(y)$.

Floating point operations

**Comparison(FLIT)**
- Each floating point operation consists of some integer operations that are computed by Server and Client.

**Floating point operations**

- Two floating point operations are used:
  - Comparison(FLIT)
  - Multiplication(FLIT)

Conclusion

- Privacy-preserving techniques are used for HMM computation in two-party setting.
- The overhead of communications and computations are minimized.