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.

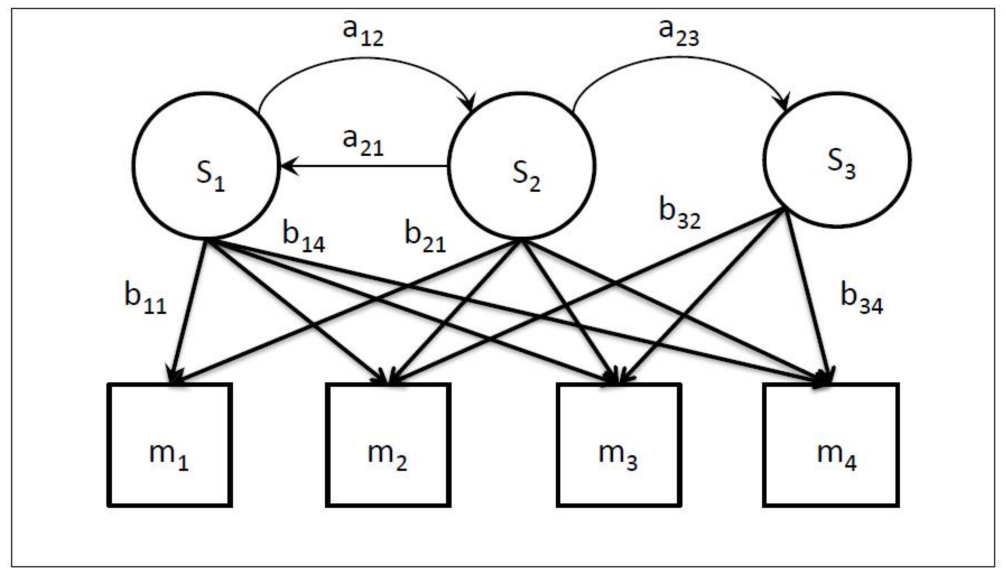
Goals

- ❑ 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 $S_1, \dots, S_n$.
- ❑ M possible outcomes $m_1, \dots, m_M$.
- ❑ A vector $\pi = \langle \pi_1, \dots, \pi_N \rangle$ 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 algorithm

$\langle P^*, q_* \rangle \leftarrow HMM(N, T, \pi, A, \alpha, \omega, \mu, \Sigma, X)$

1- For $j = 1$ to N, and $k = 1$ to T, compute

β_{jk} as in following equation:

$$\beta_{jk} = \sum_{i=1}^{\alpha} \omega_i \, e^{-\frac{1}{2}(X_k - \mu_i)^T \Sigma_i^{-1}(X_k - \mu_i)}$$

2- Set $\lambda = \langle N, T, \pi, A, \beta \rangle$;

3- Execute $\langle P^*, q^* \rangle = Viterbi(\lambda)$;

4- Return $\langle P^*, q^* \rangle$.

- ❑ P^* is the probability of the most likely path for a given sequences of observation.
- ❑ $q^* = \langle q_1{}^*, \dots, q_T{}^* \rangle$ denotes the most likely path.

And $Viterbi$ algorithm is:

$\langle P^*, q^* \rangle = Viterbi(\lambda)$

1- Initialization step: for $i = 1$ to N do

- $\delta_1(i) = \pi_i \beta_{i1}$
- $\psi_1(i) = 0$

2- Recursion step:

for $k = 2$ to T and $j = 1$ to N do

- $\delta_k(j) = \left(\max_{1 \le i \le N} [\delta_{k-1}(i) a_{ij}]\right) \beta_{jk}$
- $\psi_k(j) = \arg\max_{1 \le i \le N} [\delta_{k-1}(i) a_{ij}]$

3- Termination step:

- $P^* = \max_{1 \le i \le N} [\delta_T(i)]$
- $q_T{}^* = \arg\max_{1 \le i \le N} [\delta_T(i)]$
- For $k = T - 1$ to 1 do
  $q_k{}^* = \psi_{k+1}(q_{k+1}{}^*)$

4- Return $\langle P^*, q^* \rangle$

Floating point operations

Two floating point operations are used:

- ❑ Comparison(FLLT)
- ❑ Multiplication(FLMul)

Each floating point operation consists of some integer operations that are computed by Server and Client.

Floating point operations

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

Server:

1- Select $b_1 \in \{0, 1\}, r_1, r'_1 \in \{0,1\}^*, r_1 > r'_1$.

2- compute $enc(c) = enc(x - y)$,

3- compute $a_1 = enc(1 - b_1),\ a_2 = enc\ (b_1), a_3 = enc(-1^{b_1} c r_1 + (-1)^{1-b_1} r'_1)$, and send to Client.

Client:

4- Select $b_2 \in \{0, 1\}$, $r_2, r'_2 \in \{0,1\}^*, r_2 > r'_2$.

5- compute $a'_1 = a_{1+b_2} enc(0), a'_2 = a_{2-b_2} enc\ (0), a'_3 = enc(-1^{b_2} a_3 r_2 + (-1)^{1-b_2} r'_2)$, and send to Server.

Server & Client:

6- Compute $dec(a'_3)$. If it is negative, output is a'_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 sent to Server.

Server:

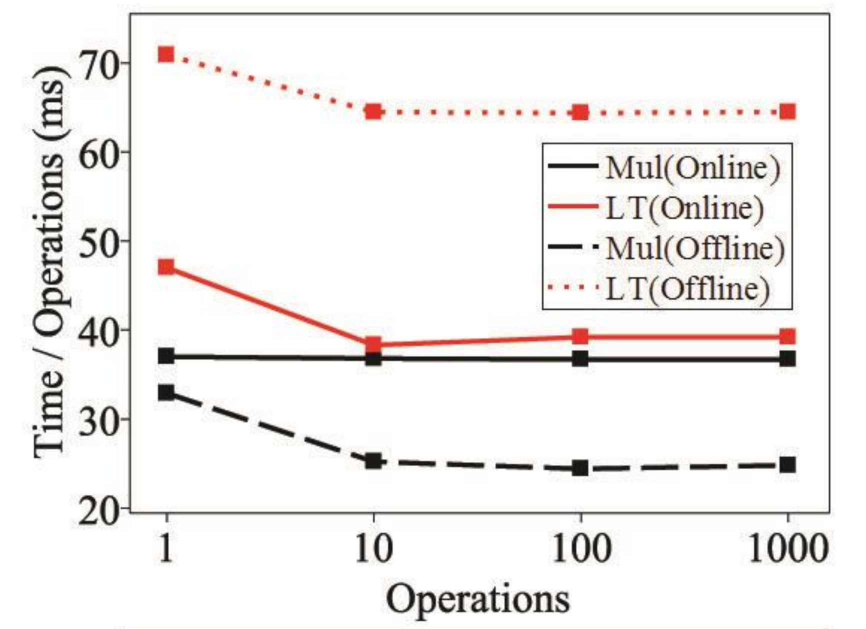
4- Compute $ecn(yr)$, and send to Client:

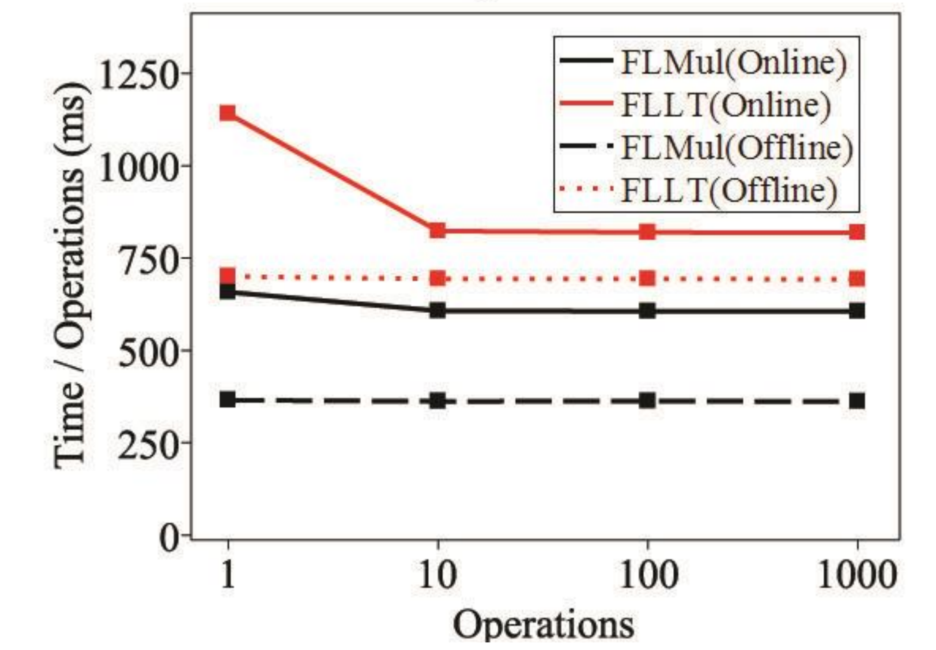
Server & Client:

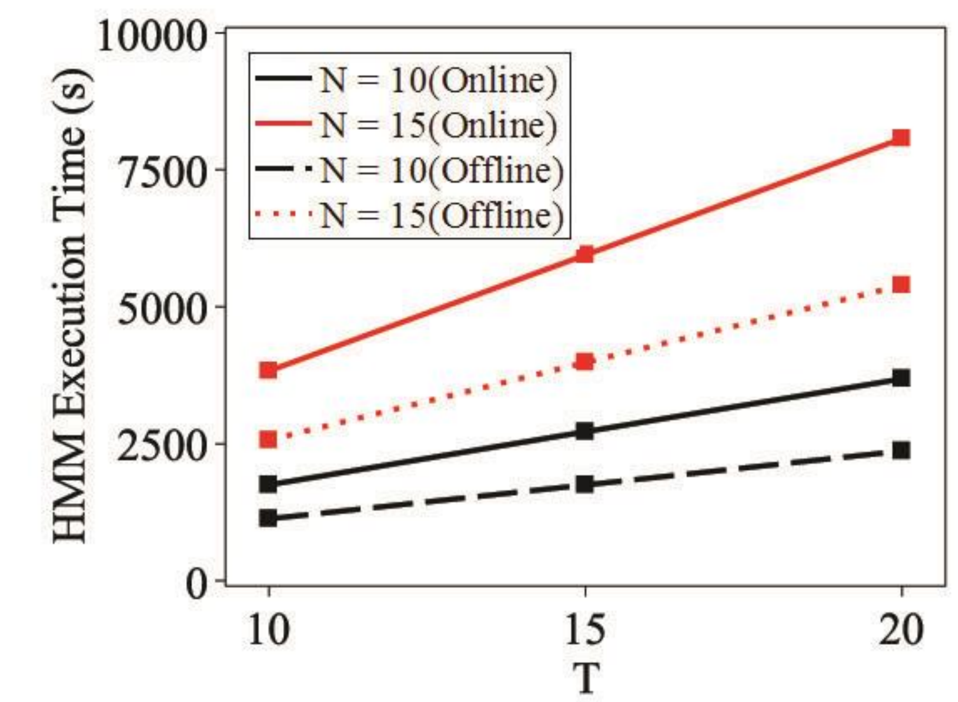
5- Compute $enc(xy)$.

Results

- ❑ Implementation of HMM for two-party setting.







Conclusion

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

Dr. Marina Blanton and Dr. Mehrdad Aliasgari mainly contributed to this research. This work was supported in part by grants CNS-1223699 from the National Science Foundation and FA9550-13-1-0066 from the Air Force Office of Scientific Research.