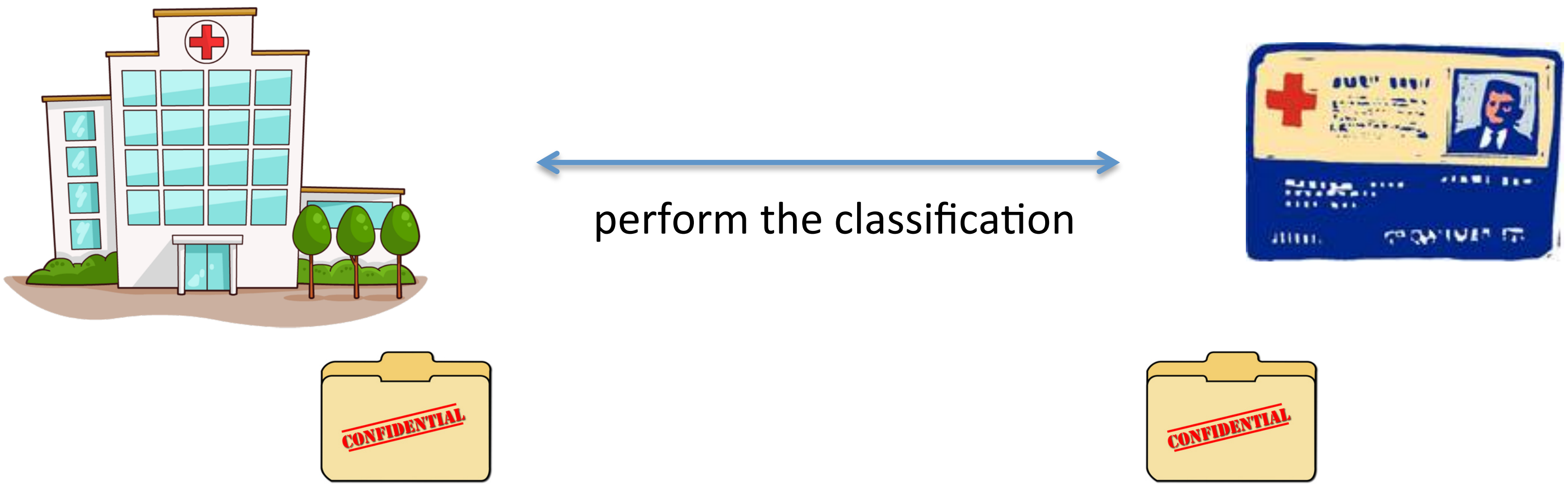
Efficient Unconditionally Secure Comparison and Privacy Preserving Machine Learning Classification Protocols

Bernardo David
Rafael Dowsley
Raj Katti
Anderson C. A. Nascimento

Problem



Both parties want to guarantee the privacy of their data.

Consider honest-but-curious adversaries.

Classifiers

Hyperplane decision classifier: model w consists of k vectors $w_1,\ldots, w_k$

$$C(w,v)=\text{argmax } \langle v,w_i \rangle$$

Naïve Bayes classifier: classification using maximum a posteriori decision rule and the model consists of the probability that each class happens and the probability that each input element happens in a certain class

$$C(w,v)=\text{argmax } (\log \Pr(C=c_i) + \Sigma \log \Pr(V_j=V_j|C=c_i))$$

Building blocks: argmax (comparison) and inner-product.

Building Blocks

Efficient and unconditionally secure solutions for the building blocks.

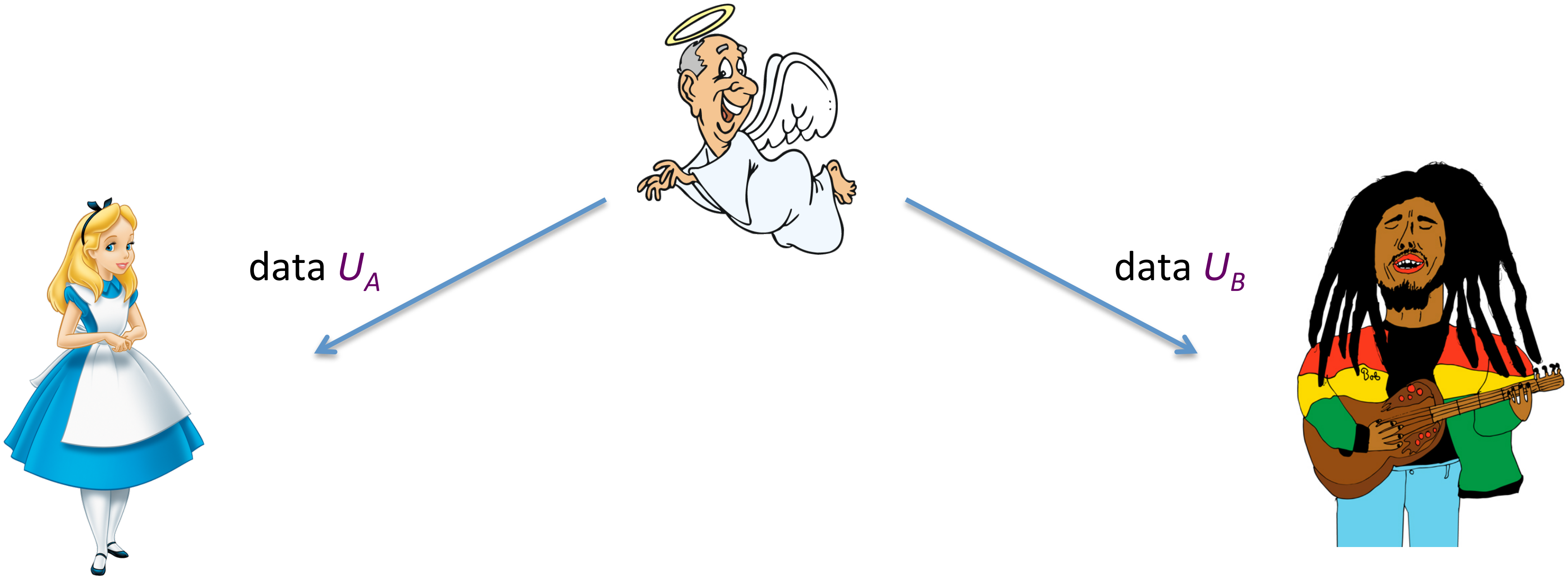
Consider the trusted initializer model.

Unconditionally secure comparisons protocols (and so argmax) can be designed using unconditionally secure multiplication as a building block.

Optimize use of the multiplication protocol.

Efficient inner-product protocol already known [DGMN11].

Trusted Initializer Model



Trusted initializer pre-distributes correlated randomness to the parties.

Trusted initializer does not learn the inputs and does not participate anymore.

Advantage: unconditional security can be achieved with very efficient protocols.

Computing Using Secret Shares

Use additively secret sharing (over some finite field) for performing secure computations.

For a value x, Alice receives a share x_A and Bob a share x_B such that $x=x_A+x_B$. Let $[x]$ denote the secret sharing of x.

Given shares $[x]$, $[y]$ it is easy to compute shares corresponding to $z=x+y$, $z=x-y$, or to add a/multiply by a constant.

Not so easy to compute shares for z=xy without revealing additional information.

Multiplication Triples

random [s], [t], [u] such that u=st

compute [m]=[x]-[s] and open m

compute [n]=[y]-[t] and open n

[z]=n[s]+m[t]+[u]+mn

Due to the blinding factors, no information about x, y or z is leaked.

Secure Comparison

For inputs of l-bits, our protocol only uses l instances of the secure multiplication.

The inputs are given as bit-wise secret sharings $[x_i]$ and $[y_i]$ in Z_q with $q>2^{l+2}$.

The output is either [0] if $y>x$ or $[w]$ for a random w in Z_q^* if $y<=x$.

This modified form of output is good enough for our applications.

Secure Comparison

$[z]$ for random non-zero z and
l multiplication triples

locally compute $[d_i]=[x_i]-[y_i]$

locally compute $[c_i]=[d_i]+1+\Sigma^{l}_{j=i+1}\,[d_j]2^{l-j+2}$

compute $[w]=[z]\ \Pi^{l}_{i=1}[c_i]$

Secure Comparison

$[z]$ for random non-zero z and
l multiplication triples

locally compute $[d_i]=[x_i]-[y_i]$ $d_i=-1$ and $d_j=0$

locally compute $[c_i]=[d_i]+1+\Sigma^{l}_{j=i+1}\,[d_j]2^{l-j+2}$ $c_i=0$

compute $[w]=[z]\,\Pi^{l}_{i=1}[c_i]$

Correctness: $y>x$ if and only if there is an i such that $y_i>x_i$ and $y_j=x_j$ for $j=i+1,\ldots,l$.

Secure Argmin

Input: bit-wise secret sharings of vectors $v_1, \ldots, v_k$

correlated data necessary for the underlying building blocks

compare all ordered pairs v_j and v_i to get $[w_{i,j}]$

compute $[p_i] = \Pi^k_{j=1, j\neq i}[w_{i,j}]$

open p_i to Alice. If $p_i \neq 0$, she adds i to the output

Naïve Bayes Classifier

$$C(w,v)=\text{argmax}\,(\log \Pr(C=c_i) + \Sigma \log \Pr(V_j=V_j \mid C=c_i))$$

log of the probabilities are converted to field elements

obliviously compute the converted log $\Pr(V_j=V_j \mid C=c_i)$

use secure argmax protocol

Hyperplane Decision Classifier

$$C(w,v)=\text{argmax} <v,w_i>$$

securely compute the inner-products [DGMN11]

convert to bit-wise secret sharings [Veu15]

use secure argmax protocol

Recap

- Possible to obtain privacy-preserving schemes for important machine learning classifiers using as building blocks comparison, argmax and inner products.

- Optimized secure comparison protocol that fits our applications.

- Possible to eliminate the trusted initializer at the cost of having some pre-computation between the parties and losing the unconditional security.

Thank You!