

Secure Biometric Computation and Outsourcing

Marina Blanton

**Department of Computer Science and Engineering
University of Notre Dame**

**University of California, Irvine
September 2011**

What is This Talk All About?

- Everyone agrees that **biometric data are sensitive** in nature
- **Proper protection is required** any time such data is used in untrusted environments
 - data is distributed across different entities
 - external computing resources are used
 - only in-house resources are used, but additional protection against break-ins, malware, etc. is desirable
- Computing on biometric data is non-trivial due to **noisy nature of each reading**
- This talk treats the problem of **secure biometric processing and outsourcing** for the above scenarios

Secure Computing on Biometric Data

- **Where do we need to securely process biometric data?**
 - **data is located at different agencies**
 - **data access by third parties can be prohibited by law or other provisions**
 - **can they compute if there are related biometrics that appear in both databases?**
 - **biometric computation can be so large in scale that using external computing resources is necessary**
 - **researchers test a new biometric algorithm**
 - **pairwise distance between each pair of biometrics in the database needs to be computed**

Secure Biometric Computation

- Let's use a specific setup of **two-party iris identification**
 - A has a database D of iris images
 - B has a biometric image and would like to know if the biometric appears in A 's database
 - B extracts features from the image and obtains an iris code X represented as a binary string
 - A and B compare X to each $Y \in D$ in such a way that
 - A doesn't learn anything about X
 - B learns nothing about each Y other than the result of the comparison (a bit)

Iris Code Matching

- An **iris** is represented as an m -bit binary string X
- There is an additional m -bit string $M(X)$, called **mask**, for each X
 - mask indicates what bits in X are unreliable and shouldn't be used to make a decision about proximity of two iris codes
- To compute a **distance** between iris codes X and Y , we compute what fraction of reliable bits are the same in X and Y
 - let X_i denote i th bit of X

$$\begin{aligned}\text{dist}(X, Y) &= \frac{\|(X \oplus Y) \cap M(X) \cap M(Y)\|}{\|M(X) \cap M(Y)\|} \\ &= \frac{\sum_{i=1}^m (X_i \oplus Y_i) M(X_i) M(Y_i)}{\sum_{i=1}^m M(X_i) M(Y_i)}\end{aligned}$$

Iris Code Matching

- Iris codes X and Y are considered to be a **match** if their distance is below a certain threshold T : $\text{dist}(X, Y) < T$
- The matching process also needs to address **iris rotation** due to head tilt
 - during matching one code needs to be rotated to find the optimal alignment
 - let $LS^i(Y)$ and $RS^j(Y)$ denote circular rotation of Y i positions left and j positions right, respectively
 - now the matching process becomes

$$\min(\text{dist}(X, LS^c(Y)), \dots, \text{dist}(X, LS^1(Y)), \text{dist}(X, Y), \text{dist}(X, RS^1(Y)), \dots, \text{dist}(X, RS^c(Y))) < T$$

Secure Iris Code Matching

- **How do A and B perform the computation without revealing their data?**
 - the parties can compute on encrypted data or on shares of the data
 - in **additively homomorphic encryption**:
 - $\text{Enc}(m_1) \cdot \text{Enc}(m_2) = \text{Enc}(m_1 + m_2)$ and
 $\text{Enc}(m)^a = \text{Enc}(a \cdot m)$
- **Questions that we need to answer**
 - all encrypted values must be integers, what does it mean for us?
 - division is difficult to perform
 - can we substitute it with something else?
 - how about computation of the minimum now?
 - intermediate distances are incomparable

Secure Iris Code Matching

- **Let** $D(X, Y) = \|(X \oplus Y) \cap M(X) \cap M(Y)\|$ **and**
 $M(X, Y) = \|M(X) \cap M(Y)\|$

- **We now obtain**

$$\begin{aligned} & (D(X, LS^c(Y)) < T \cdot M(X, LS^c(Y))) \vee \dots \vee \\ & \vee (D(X, Y) < T \cdot M(X, Y)) \vee \dots \vee \\ & \vee (D(X, RS^c(Y)) < T \cdot M(X, RS^c(Y))) \end{aligned}$$

- **The above can be implemented using arithmetic operations and comparisons**

$$- X_i \oplus Y_i = X_i + Y_i - 2X_iY_i = X_i(1 - Y_i) + (1 - X_i)Y_i$$

Secure Iris Code Matching

- **Initial secure iris matching protocol**

- B creates a public-key encryption pair (pk, sk) for homomorphic encryption and gives pk to A
- B sends encrypted bits of X and $M(X)$ to A
 - necessary computation will be possible only if the transmitted information is in the correct form
 - for each $i = 1, \dots, m$, B transmits
$$\langle a_{i1}, a_{i2} \rangle = \langle \text{Enc}(X_i M(X_i)), \text{Enc}((1 - X_i)M(X_i)) \rangle$$
 - this will allow A to compute encrypted $(X_i \oplus Y_i)M(X_i)M(Y_i)$ and $M(X_i)M(Y_i)$ and therefore $D(X, Y)$ and $M(X, Y)$
 - A first sets $a_{i3} = a_{i1} \cdot a_{i2} = \text{Enc}(M(X_i))$ for $i = 1, \dots, m$

Secure Iris Code Matching

- **Initial secure iris matching protocol**

- **A will perform the same operations for each $Y \in D$ and each rotation Y^j of Y for $j = -c, \dots, c$**
- **to get $D(X_i, Y_i^j) = (X_i(1 - Y_i^j) + (1 - X_i)Y_i^j)M(X_i)M(Y_i^j)$ in encrypted form, A performs $b_i^j = a_{i1}^{(1-Y_i^j)M(Y_i^j)} \cdot a_{i2}^{Y_i^j M(Y_i^j)} = \text{Enc}(X_i M(X_i)(1 - Y_i^j)M(Y_i^j) + (1 - X_i)M(X_i)Y_i^j M(Y_i^j))$**
- **A then multiplies all b_i^j to obtain $\text{Enc}(D(X, Y^j))$**
- **to obtain $\text{Enc}(T(\|M(X) \cap M(Y^j)\|))$, A performs $d_i^j = a_{i3}^{M(Y_i^j)} = \text{Enc}(M(X_i)M(Y_i^j))$ and then computes $d^j = (\prod_{i=1}^m d_i^j)^T = \text{Enc}(T(\sum_{i=1}^m M(X_i)M(Y_i^j)))$**
- **A now computed all $D(X, Y^j)$ and $T \cdot M(X, Y^j)$**

Secure Iris Code Matching

- **Initial secure iris matching protocol**
 - what remains is secure computation of $2c + 1$ comparisons and then OR of the resulting bits
 - the most efficient way to achieve this is to use called garbled circuits
 - it is a Boolean circuit for comparison that is evaluated obliviously without revealing any information about the data
 - before comparisons can be done, A and B need to split the encrypted values into random shares and then decrypt
 - A adds a random value r to computed distances
 - B obtains $\text{Enc}(D(X, Y^j) + r)$ and decrypts without being able to learn $D(X, Y^j)$

Secure Iris Code Matching

- **Efficiency** of this solution can be improved in many ways
 - proper choice of **encryption scheme** can make a tremendous difference
 - many encrypted values can be **computed in advance** before the data are known
 - A can precompute encryptions of all random values
 - B can precompute (and even send) encryptions of bits
 - computation of $D(X_i, Y_i^j)$ and $M(X_i, Y_i^j)$ can be **optimized** based on A 's data
 - e.g., if $M(Y_i^j) = 0$, computation of both b_i^j and d_i^j can be skipped
 - additional **one-time computation** is possible that makes processing of each Y faster

Secure Iris Code Matching

- **Implementation of secure iris matching**
 - the solution was implemented using a recent encryption scheme of Damgard, Geisler, and Kroigard
 - the parameters were set to $m = 2048$ and $c = 5$
 - **the performance is notable for a fully secure solution**
 - two biometrics X and Y can be compared in less than 0.2 second
 - this involves on the order of $(2c + 1)m$ cryptographic operations
 - shorter biometric representations or simpler computations can be processed faster

Secure Iris Outsourcing

- **Now suppose Alice has biometric images and would like to outsource the same computation to a cloud service**
- **Consider two options:**
 - **computation is performed by a single server**
 - **computation is outsourced to a number of servers and takes the form of secure multi-party computation**
- **Techniques used for these options and their capabilities greatly differ**

Secure Iris Outsourcing

- **Securely outsourcing computation to a single server**
 - the main tool used for this purpose is **predicate encryption**
 - a ciphertext has a number of attributes I associated with it
 - a token corresponding to a predicate f is issued
 - a token can be used to decrypt a ciphertext iff its predicate evaluates to true on the ciphertext's attribute, i.e., $f(I) = 1$
 - both ciphertext and predicate privacy are essential
 - Alice stores encryptions of iris codes from her database at a server
 - to perform identification of a new biometric, she issues a token for it
 - the server applies the token to the ciphertexts, and all entries that decrypt correspond to related biometrics

Secure Iris Outsourcing

- There are limitations to what functions can be evaluated in this manner
- In the most powerful type of **predicate encryption**
 - attributes I and predicates f correspond to vectors
 - $f(I) = 1$ iff the inner product of f and I is 0
 - this supports checking whether a polynomial evaluates to a specific value, OR and AND of polynomials, but not the overall computation

- What can be computed now is

$$(D(X, LS^c(Y)) < T) \vee \dots \vee (D(X, Y) < T) \vee \dots \vee (D(X, RS^c(Y)) < T)$$

- with a special robust representation of iris codes $M(X, Y)$ is always high and changing the function doesn't introduce a large error

Secure Iris Outsourcing

- **Using predicate encryption for iris outsourcing**

- to evaluate polynomial $p(x) = a_t x^t + \dots + a_1 x + a_0$ on point x_0 , set $I = \langle a_t, \dots, a_1, a_0 \rangle$ and $f = \langle x_0^t, \dots, x_0, 1 \rangle$
- to test whether $p(x_0) = d$, change the above to $I = \langle a_t, \dots, a_1, a_0, -d \rangle$ and $f = \langle x_0^t, \dots, x_0, 1, 1 \rangle$
- to compute $f_1(I_1) \vee f_2(I_2)$, use polynomial $p_3 = p_1 \cdot p_2$
 - p_3 evaluates to 0 when at least one of p_1 and p_2 evaluates to 0
- we can now compute the Hamming distance

$$\begin{aligned} H(X, Y) &= \sum_{i=1}^m ((X_i \oplus Y_i) \wedge M(X_i) \wedge M(Y_i)) \\ &= \sum_{i=1}^m (X_i + Y_i - 2X_i Y_i) M(X_i) M(Y_i) \\ &= \sum_{i=1}^m ((1 - 2X_i) Y_i + X_i) M(X_i) M(Y_i) \end{aligned}$$

using polynomial representation

Secure Iris Outsourcing

- **Using predicate encryption for iris outsourcing**
 - **set** $I = \langle M(Y_1)Y_1, M(Y_1), \dots, M(Y_m)Y_m, M(Y_m) \rangle$ **and**
 $f = \langle M(X_1)(1 - 2X_1), M(X_1)X_1, \dots, M(X_m)(1 - 2X_m), M(X_m)X_m \rangle$
 - **this corresponds to a polynomial p evaluated on points $X_i, M(X_i)$**
 - **to test whether the distance lies in the interval $[0, T - 1]$, construct new polynomial $q = p(p - 1) \cdots (p - (T - 1))$**
 - **finally, form polynomials q_{-c}, \dots, q_c , where q_i uses Y shifted i times to the right, and compute their OR by taking their product**
- **The server will apply the token for f to all ciphertexts in the database and return the indices of records that matched**

Secure Iris Outsourcing

- **When computation is outsourced to several servers, precise and significantly more efficient computation can be achieved**
 - use secure computation techniques based on linear secret sharing
 - each Hamming distance – i.e., $D(X, Y)$ and $M(X, Y)$ – is computed locally
 - communication is no longer linear in m
 - interactive work for comparing two biometrics is linear in the length of the distance representation $\log m$
 - security against active adversaries can be achieved

Adding Robustness to Computation

- **Another important topic for computation outsourcing is ensuring that the returned result is correct**
 - we consider **“lazy” adversaries** that might attempt to skip the computation, but don't try to intentionally corrupt it
 - cheating should be detected with desired probability when the server skips a noticeable portion of computation
 - e.g., if the server performs **95%** of computation or less, probability of detection should be at least **99%**

Adding Robustness to Biometric Computation

- **Outsourcing of biometric identification is not suited for efficient computation verification**
 - we don't want the client to do work proportional to the database size
 - verification of the correctness of returned indices is not possible with sub-linear work
- **Other types of computing on biometric data can be verified at work sublinear in the size of computation**
 - testing a new biometric algorithm involves computing distances between a large number of biometrics
 - performing “all-pairs” computation or producing statistics about the distribution involves $O(n^2)$ work for a database of size n
 - computation is inevitably placed on a cloud or grid

Adding Robustness to Biometric Computation

- **Cheating detection in All-Pairs and statistics computation**
 - to detect cheating, we use simple ideas
 - add a number of fake biometrics at random locations
 - add fake elements at random locations to each biometric
 - ensure that checked values are unpredictable
 - the analysis is complex and determines the values of security parameters
 - need to ensure that probability of guessing the values that haven't been computing and are being checked is sufficiently low

Conclusions

- **Secure processing of biometric data moves closer to practicality**
- **Many interesting research directions remain**
 - a number of biometric types are still unexplored
 - efficient robustness techniques deserve more attention