A NEW SECURITY MODEL FOR SECURE THRESHOLDING

Nan Hu and Sen-ching S. Cheung

Center For Visualization and Virtual Environment
Department of Electrical and Computer Engineering
University of Kentucky, Lexington, KY 40507
nan.hu@uky.edu, cheung@engr.uky.edu

ABSTRACT

The goal of secure computation is for distrusted parties on a network to collaborate with each other without disclosing private information. In this paper, we focus on secure thresholding, or comparing two secret numbers, which is a key step in pattern recognition. Existing cryptographic protocols are too complex to be used in real-time signal processing. We propose a new security model based on noninvertible functions called Quasi-Information-Theoretic Security. Using this model, we develop a novel secure thresholding protocol that is both secure and computationally efficient. The proposed protocol hides information in a carefully-designed random polynomial and in a lower-rank subspace based on Chebyshev's polynomials.

Index Terms— Communication system security, Secure Multiparty Computation, Cryptography, Distributed Algorithms,

1. INTRODUCTION

The proliferation of signal capture devices, portable storage and wireless networks make sharing of digital data easier than ever. Such casual exchange of data, however, has increasingly raised questions on how sensitive information can be protected. Consider the scenario in which a user of a cellular-phone camera wants to send his/her pictures to an online photo-processing laboratory for image enhancement such as red-eye removal. The user would be concerned about the privacy of his/her pictures while the online store would need to protect the proprietary enhancement technologies against reverse-engineering. Consider another scenario that a law enforcement agency wants to search for possible suspects in a surveillance video owned by private company A, using a proprietary software from yet another private company B. The three parties involved (agency, company A, company B) all have information they do not want to share with each other (criminal biometric database from the agency, surveillance tape from company A and proprietary software from company B). To prevent these problems, we need to establish a joint computation and communication platform that can guarantee the secrecy of private data and algorithms, and at the same time achieve a well-defined objective that benefits all parties involved.

Such type of secure computation in a distributed environment is a well-known problem in cryptography, and is referred to as the Secure Multiparty Computation (SMC) problem. The goal of a SMC protocol is to allow multiple distrusted parties jointly compute a function without complete sharing of their own information [1]. Like many other cryptographic protocols, the security of SMC protocols can be guaranteed under two different security models — information-theoretical security and computational security. Information-theoretically secure protocols protects privacy in such a way that the information exchanged in the protocol provides no additional information, measured in entropy, about the private data. In computationally secure protocols, private information is first transformed before transmitting to other parties. The security is based on the huge computational burden of performing the inverse transformation. Although the information-theoretic security model provides the ideal level of security, it has been shown that many simple operations like inner product or thresholding cannot be securely computed between two distrusted parties [2]. As a result, most existing SMC protocols are built under the computational security model [1, 3, 4].

The main drawback of computationally-secure protocols is their high computational complexity. For example, the classical solution to the thresholding problem[1], or comparing two private numbers a and b, is to use Oblivious Transfer (OT) [4] – one party (Bob) creates a series of tables by bitwise comparing b with every possible value of a, encrypts the tables using a public-key cipher, and transfers them to Alice. Alice decrypts the entries in the tables that correspond to his own number a and deduces the result. Most public-key ciphers use modular exponentiations on very large finite field which is complex to compute. As a result, it is difficult to scale these protocols to signal processing applications that requires handling a large amount of data and satisfying the real-time constraint.

In this paper, we propose a new security model, called QUasi-Information-Theoretic (QUIT) security model to enable much more efficient SMC protocols to be developed. The QUIT model is a weaker form of information-theoretic security. Its security is provided by using non-invertible transformations on private data. Though not explicitly defined, various form of QUIT-secure protocols have already been developed for inner product computation [5] and linear filtering [6]. In this paper, we formally define the QUIT model and develop a QUIT-secure protocol for the thresholding problem which is a key step in building secure pattern recognition applications. We will show that, compared with existing protocols, our proposed protocol is more secure to one party (Alice) but not as secure to the other (Bob) – Alice can deduce Bob's number to be among N distinct numbers spread through the entire range of the input. N is a design parameter that can be changed based on the target level of security. Experimental results show that our protocol executes significantly faster than existing protocols.

The rest of the paper is organized as follows. In Section 2, we briefly review the existing secure models and introduce our new QUIT security model. A novel QUIT-secure threshold protocol is presented in Section 3. We prove the security of the new protocol in Section 4 and compare its performance with existing scheme in Section 5. We conclude the paper in Section 6.

[1] This problem is commonly referred to as the Secure Millionaire Problem in SMC literature.

1-4244-0728-1/07/$20.00 ©2007 IEEE

2. SECURITY MODELS

Following the convention used in cryptography, we refer the private information as *plaintext* and the information exchanged among distrusted parties as *ciphertext*. All existing cryptographic protocols are based on one of the two security models — information-theoretical security and computational security. *Information-theoretical security* means the *a posteriori* probability of the plaintext being x, given that the observed ciphertext y, is identical to the *a priori* probability of the plaintext being x, i.e. knowing y gives no information about x. On the other hand, *computational security* means given the ciphertext y, no polynomial-time algorithm can compute the correct plaintext x with a non-trivial probability.

2.1. Quasi-Information-Theoretic Security

The quasi-information-theoretic security is based on non-invertible mappings. Let us first define non-invertibility.

Definition 1 *Let $g : \mathcal{X} \to \mathcal{Y}$ be a mapping from a probability space*[2] *$\mathcal{X}$ to another probability space $\mathcal{Y}$. $\forall\, x \in \mathcal{X}$ with $P(x) > 0$, define $g^{-1} \circ g(x) = \{\, \alpha \mid \alpha \in \mathcal{X},\ g(\alpha) = g(x)$ and $P(\alpha) > 0\}$.*

1. *Given $\alpha,\ \beta \in \mathcal{X}$ with non-trivial probability, they are called QUIT-indistinguishable if $g(\alpha) = g(\beta)$.*
2. *Given $x \in \mathcal{X}$ with $P(x) > 0$, $g^{-1} \circ g(x)$ is called the QUIT indistinguishable set of x under g.*
3. *$g(x)$ is called noninvertible if the probability of finding a $x \in \mathcal{X}$ whose QUIT indistinguishable set has no element besides x is zero, i.e. $P\Big(\{\, \alpha \mid \alpha \in \mathcal{X},\ |g^{-1} \circ g(\alpha)| < 2\}\Big) = 0$. In particular, we call $g(x)$ N-noninvertible if the probability of finding a QUIT indistiinguisable set smaller than N is zero.*

Notice that given $\alpha \in g^{-1} \circ g(x)$, there is no relative increase in the knowledge about α and x based on $y = g(x)$. This can be easily shown by using the Bayes rule:

$$\frac{P(x|g(x) = y)}{P(\alpha|g(\alpha) = y)} = \frac{P(g(x) = y|x)P(x)/P(y)}{P(g(\alpha) = y|\alpha)P(\alpha)/P(y)} = \frac{P(x)}{P(\alpha)} \qquad (1)$$

Any cryptographic protocol A can be viewed as a mapping from the plaintext $\mathcal{P}$ to the ciphertext $\mathcal{C}$. As such, we introduce the following definitions:

Definition 2 *A cryptographic protocol A is called QUIT-secure if the underlying mapping A from plaintext to ciphertext is noninvertible. A is N-QUIT secure if the mapping is N-noninvertible.*

It is obvious that the QUIT security model is weaker than the information-theoretic security as g can be any noninvertible mapping which can certainly provide additional information about the plaintext $x \in \mathcal{P}$ given the ciphertext $y = g(x) \in \mathcal{C}$, i.e. $P(x|y) > P(x)$. On the other hand, based on equation (1), the QUIT model guarantees that the relative relationship between two plaintexts x and α that map to the same ciphertext y remains unchanged, though the individual conditional probability may increase.

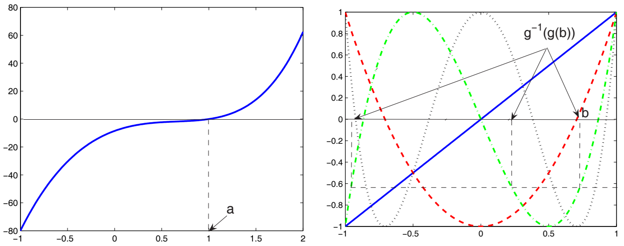
QUIT is also different from computational security. The computational security model depends solely on the computational hardness of computing the plaintext x given the ciphertext $g(x) = y$. However, for a given y, it is guaranteed that there is only one x that satisfies $g(x) = y$. In QUIT security, computing the QUIT indistinguishable set $g^{-1} \circ g(x)$ of x for a given mapping is often quite straightforward. However $g^{-1} \circ g(x)$ can be large and the true identify of x will remain hidden. It can also be seen from Equation (1) that, if $P(\alpha) = P(x)$, then $P(\alpha|g(x|\alpha)) = P(x|g(x))$, i.e. if the plaintext is uniformly distributed, the *a posteriori* probability is also uniform within the QUIT indistinguishable set of x. In this special case, there is no algorithm that can distinguish between α and x.

3. PROPOSED PROTOCOL

Assume we have two distrusted parties: Alice and Bob. Alice holds a secret scaler a, and Bob holds another secret scaler b. They want to find out who has a bigger number without disclosing their private data. Under our new notion of security, we propose to convert this problem into a special polynomial evaluation problem. Let n be an even number. Alice first randomly generates a $(n-1)^{\text{th}}$-degree polynomial $f(x)$ that has only one real root: Alices secret number a. In addition, we require that the derivative of $f(x)$ at a is non-negative. Alice can easily generate this polynomial by first randomly selecting $(n-2)/2$ complex conjugate numbers as the roots of the polynomial, and then multiplying the resulting polynomial by a negative random number if the derivative of f at a is negative or a positive random number otherwise. We will refine this procedure for better security in Section 4. The key property of $f(x)$ is that for any $b > a$, we have $f(b) > 0$ and for all $b < a$, we have $f(b) < 0$. An example of such a $f(x)$ is shown in Figure 1(a). Thus if Bob knows only the value of $f(b)$ without knowing the actual polynomial, he can easily solve the problem without any knowledge of a. Given $f(x) = a_{n-1}x^{n-1} + \cdots + a_1x + a_0$, we can evaluate $f(b)$ as an inner product between two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$:

$$f(b) \triangleq a_{n-1}b^{n-1} + \cdots + a_1b + a_0 \triangleq \mathbf{x}_1^T\mathbf{x}_2 \qquad (2)$$

where Alice has $\mathbf{x}_1 = [\ a_{n-1}\ \cdots\ a_1\ a_0\ ]^T$ and Bob has $\mathbf{x}_2 = [\ b^{n-1}\ \cdots\ b\ 1\ ]^T$.



(a) (b)

Fig. 1. *(a) Random polynomial with a single real root at $a = 1$. (b) Chebyshev's polynomials of degree one (blue solid), degree two (red dash), degree three (green dast-dot) and degree four (black dot).*

Thus, the evaluation of a polynomial becomes that of an inner product. Our secure inner product evaluation is based on [5]. The idea is to linearly map $\mathbf{x}_1$ and $\mathbf{x}_2$ into a lower-dimensional space such that given the transformed results, it is impossible to exactly recover a and b. We use an invertible matrix $M \in R^{n\times n}$, and vertically divide it into two parts $M_l \in R^{n\times k}$ and $M_r \in R^{n\times(n-k)}$. On the other hand, we horizontally divide M^{-1} into two parts $M_t \in R^{k\times n}$ and $M_b \in R^{(n-k)\times n}$. The design of M and its submatrices is critical to the security of the protocol and the details will be dis-

[2] We assume the probability space discrete. If it is continuous, then $\mathcal{X}$ and $\mathcal{Y}$ will be the collection of measurable sets.

cussed in Section 4. Given M and the submatrices, our protocol of secure thresholding is described in Algorithm 1 and 2.

Algorithm 1 `ThresholdingAlice`$(\mathbf{x}_1, M)$

```
Require: $\mathbf{x}_1 = [\ a_{n-1}\ \cdots\ a_1\ a_0\ ]^T \in \mathbb{R}^n$. $M = (M_l\ \ M_r)$
    is a $n \times n$ invertible matrix where $n \geq 2$; $M_l \in \mathbb{R}^{n \times k}$ and
    $M_r \in \mathbb{R}^{n \times (n-k)}$.
$\mathbf{x}_{11} \leftarrow \mathbf{x}_1^T M_l$
$\mathbf{x}_{12} \leftarrow \mathbf{x}_1^T M_r$
Transmit $\mathbf{x}_{12}$ to Bob.
Receive $\mathbf{x}_{21}$ from Bob.
Send $\mathbf{x}_{11}^T \mathbf{x}_{21}$ to Bob.
```

Algorithm 2 `ThresholdingBob`$(\mathbf{x}_2, M^{-1})$

```
Require: $\mathbf{c} = [\ b^{n-1}\ \cdots\ b\ 1\ ]^T \in \mathbb{R}^n$. $M^{-1} = \begin{pmatrix} M_t \\ M_b \end{pmatrix}$ is a
    $n \times n$ invertible matrix where $n \geq 2$; $M_t \in \mathbb{R}^{k \times n}$ and $M_b \in$
    $\mathbb{R}^{(n-k) \times n}$.
$\mathbf{x}_{21} \leftarrow M_t \mathbf{x}_2$
$\mathbf{x}_{22} \leftarrow M_b \mathbf{x}_2$
Transmit $\mathbf{x}_{21}$ to Alice.
Receive $\mathbf{x}_{12}$ from Alice.
Receive $\mathbf{x}_{11}^T \mathbf{x}_{21}$ from Alice.
Compute $f(b) = \mathbf{x}_{12}^T \mathbf{x}_{22} + \mathbf{x}_{11}^T \mathbf{x}_{21}$
Return $f(b) > 0$.
```

The correctness of this protocol can be easily verified.

$$\begin{aligned} f(b) &= \mathbf{x}_1^T \mathbf{x}_2 \\ &= \mathbf{x}_1^T M M^{-1} \mathbf{x}_2 \\ &= \mathbf{x}_1^T \begin{pmatrix} M_l & M_r \end{pmatrix} \begin{pmatrix} M_t \\ M_b \end{pmatrix} \mathbf{x}_2 \\ &= \mathbf{x}_{11}^T \mathbf{x}_{21} + \mathbf{x}_{12}^T \mathbf{x}_{22} \end{aligned}$$

4. SECURITY ANALYSIS OF THE PROTOCOL

In this section, we show that our proposed thresholding protocol is QUIT secure. First, let us consider the information Bob sent to Alice. Bob sends Alice $\mathbf{x}_{21} = M_t \mathbf{x}_2$. Since M_t is a $k \times n$ matrix and $\mathbf{x}_2 = [\ b^{n-1}\ \cdots\ b\ 1\ ]^T$, $M_t \mathbf{x}_2$ is equivalent to evaluating k different polynomials at b, whose coefficients are defined by the row vectors of M_t. The cryptosystem induced by M_t is m-QUIT secure if and only if there are at least m distinct values in the QUIT indistinguishable set of b. This is equivalent to saying that the $(n-1)^{th}$ degree polynomials with coefficients $[M_t(i,1)\ M_t(i,2)\ \ldots\ M_t(i,n-1)\ M_t(i,n) - \mathbf{x}(i)]$ for $i = 1, 2, \ldots, k$ share m distinct roots. To maximize the security, we would to have m as large as $n-1$ which is the degree of the polynomials. As shown below, this constraint impose a maximum value on k, the number of rows in M_t, one can use. To show this, let us start from the following lemma:

LEMMA 1 *Given two polynomials $g(x)$ and $h(x)$ of degree $n-1$ and a scalar b. If equations $g(x) = g(b)$ and $h(x) = h(b)$ share exactly the same roots, then $g(x) = k_1 h(x) + k_2$, where $k_1 \neq 0$ and k_2 are constants.*

Proof Since $g(x) = g(a)$ and $h(x) = h(a)$ share the same set of roots, we have $[g(x) - g(a)] = k_1[h(x) - h(a)]$ or $g(x) = k_1 h(x) + [g(a) - k_1 h(a)]$ for some $k_1 \neq 0$. Set $k_2 = g(a) - k_1 h(a)$ and results follow. Notice that as long as $g(x)$ is not a constant, the coefficient vector of $f(x)$ is linear independent of the coefficient vector of $g(x)$. Q.E.D.

THEOREM 1 *If the proposed thresholding protocol is $(n-1)$-QUIT secure with respect to Bob, then the number of rows k in M_t is at most two.*

Proof Since the full matrix M^{-1} invertible, the k row vectors of M_t must be linearly independent. k is at least two based on LEMMA 1. If k is larger than two, select any three row vectors and formulate the three corresponding polynomials $f_1(x)$, $f_2(x)$ and $f_3(x)$. Using LEMMA 1, we have $f_1(x) = k_0 f_3(x) + k_1$ and $f_2(x) = k_3 f_3(x) + k_4$. Thus, the coefficient vectors of both $f_1(x)$ and $f_2(x)$ lie in the subspace spanned by the coefficient vector of $f_3(x)$ and $[\ 0\ \cdots\ 0\ 1\ ]^T$ and we obtain a contradiction. Q.E.D.

Next, we come to the actual design of M_t. Even though Alice may not know the precise value of b, she can usually assume b to be within a certain range. Without loss of generality, assume that $b \in [-1, 1]$. Thus, we need to find a polynomial $g(x)$ such that for any $b \in [-1, 1]$, all the $n-1$ roots of $g(x) = g(b)$ are real and fall within the range $[-1, 1]$. An example of such function is the $(n-1)^{\text{th}}$ order Chebyshev's polynomial[3]: $T_{n-1}(x) = \cos[(n-1)\cos^{-1}(x)]$. Figure 1(b) shows the first four Chebyshev's polynomials. We state the following fact without proof about the Chebyshev's polynomials though it is quite obvious based on the figure.

FACT 1 *Except for at most $n+1$ distinct points within $[-1, 1]$, the n^{th} order Chebyshev's polynomial $T_n(x)$ is n-noninvertible on $[-1, 1]$*

The $n+1$ distinct points forms a measure-zero set in $[-1, 1]$. Thus, the mapping $M_t \mathbf{x}_2$ will be $(n-1)$-QUIT secure to Bob if we can set $M_t = \begin{pmatrix} C[T_{n-1}(x)] \\ C[k_0 T_{n-1}(x) + k_1] \end{pmatrix}$ where the operator $C[\cdot]$ denotes the coefficient vector of a polynomial. Given M_t, we can easily compute M_b by extending the two row vectors in M_t to a full set of basis in R^n.

We now show that the proposed thresholding protocol is also QUIT-secure to Alice. Bob receives $\mathbf{x}_{12} = \mathbf{x}_1^T M_r$ from Alice. Bob also knows that $\mathbf{x}_1$ corresponds to the coefficient vector of a $(n-1)^{\text{th}}$ degree polynomial $f(x)$ with a single real root and non-negative derivative at that root. To show that the protocol is QUIT-secure to Alice, we need to find $\mathbf{x}_1'$ that corresponds to a polynomial with the same features and $\mathbf{x}_{12} = \mathbf{x}_1'^T M_r$. Given M_t is defined based on the Chebyshev's polynomials, we have the following theorem:

THEOREM 2 *Given that $\mathbf{x}_1$ is the coefficient vector of a polynomial $f(x)$ with only a single real root and non-negative derivative at that root and $M_l = \begin{pmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \end{pmatrix}$, there exists $\mathbf{x}_1' \neq \mathbf{x}_1$ such that $\mathbf{x}_1'^T M_r = \mathbf{x}_1^T M_r$ and $\mathbf{x}_1'$ corresponds to the coefficients of $f'(x)$ which also has a single real root with non-negative derivative at that root.*

Proof Recall that $M^{-1} = \begin{pmatrix} M_t \\ M_b \end{pmatrix}$ and $M = (M_l\ M_r)$. Thus, M_t and M_r relate to each other by the following relationship:

$$M_t \cdot M_r = 0$$

[3] Though stated in its general form, Chebyshev's polynomials can be easily computed as a true polynomial based on the recurrence relation $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$ with $T_{-1}(x) = 0$ and $T_0(x) = 1$.

As M_r and M_t are a part of an invertible matrix, the rank of M_t is 2 and the rank of M_r is $n-2$. Thus, if $\mathbf{v}^T M_r = 0$, $\mathbf{v}^T$ must be in the subspace S spanned by the row vectors of M_t. Note that $(\mathbf{x}_1 + \mathbf{v})^T M_r = \mathbf{x}_1^T M_r$. Our strategy is to find an appropriate $\mathbf{v}$ that can satisfy the conditions.

On the other hand, the row vectors of M_t denote the coefficients of the Chebyshev's polynomial $T_{n-1}(x)$ and $k_0 T_{n-1}(x) + k_1$ for arbitrary k_0 and $k_1 \neq 0$. It is obvious that the vector $[\ 0\ \cdots\ 0\ 1\ ]^T$ is in the subspace S. Define $\mathbf{v} = [\ 0\ \cdots\ 0\ \epsilon/2\ ]^T$ where $-\epsilon$ is the *largest local maximum* in $(-\infty, a]$ of Alice polynomial $f(x)$. If no such local maximum exists, ϵ can be chosen arbitrarily. The vector $\mathbf{x}_1' = \mathbf{x} + \mathbf{v}$ then corresponds to a polynomial $f'(x) = f(x) + \epsilon/2$. Note that this polynomial still has a single real root because the large local maximum on the left hand side of the root is still $\epsilon/2$ from zero. Furthermore, the derivative at the root must be non-negative otherwise a local maxima would have crossed the x-axis. Q.E.D.

In the unfortunate case when the largest local maximum left of a and the smallest local minimum right of a are both small, we can only shift $f(x)$ by a small amount before it starts to have more than one real root. In other words, it is possible for Bob to roughly estimate a despite the fact that the protocol is QUIT-secure. The security, however, can be significantly improved by imposing some constraints on the random complex roots of $f(x)$. Without loss of generality, we again assume that Alice's number $a \in [-1, 1]$. We have the following result:

THEOREM 3 *The thresholding protocol is INFORMATION THEORETICALLY secure to Alice if Alice first generates an auxiliary polynomial*

$$g(x) = (x-1) \prod_{i=1}^{(n-2)/2} (x - c_i)(x - \bar{c}_i) \qquad (3)$$

with random c_i under the constraint $\mathrm{Real}(c_i) > 1$ *for all i and then let $f(x) = g(x) - g(a)$.*

Proof For any real x, if we rewrite each term in Equation (3) in polar form, the complex exponential terms for the conjugate roots will cancel each other and $g(x)$ will become

$$g(x) = \mathrm{sign}(x-1) \cdot |x-1| \cdot \prod_{i=1}^{(n-2)/2} |x - c_i| \cdot |x - \bar{c}_i| \qquad (4)$$

Equation (4) shows that a) $g(x)$ is negative for $x < 1$ and positive for $x > 1$ and b) $g(x)$ is strictly increasing or $\frac{dg}{dx} > 0$ for $x \leq 1$. This is because as the real parts of all the complex roots are larger than one, every modulus term in Equation (4) decreases as x approaches 1 from $-\infty$. As $\mathrm{sign}(x-1)$ is negative, $g(x)$ is strictly increasing. Clearly $f(x) = g(x) - g(a)$ for $a \in [-1, 1]$ satisfies our requirements of having a single real root and non-negative derivative at a. Recall that the coefficient vector of $f'(x) = f(x) + c$ for any constant c is in the null space of M_r. By choosing $c \in [g(a), g(a) - g(-1)]$, $f'(x)$ can have its single real root anywhere in $[-1, 1]$. Thus, based on the information sent by Alice, Bob has no information about a and the protocol is information theoretically secure to Alice. Q.E.D.

In closing, we have developed a thresholding protocol that achieves perfect security for Alice but leaks some information about Bob's secret number. Compared with existing computationally secure protocols, our protocol is more secure to one party but less secure to the other one.

5. EXPERIMENTAL RESULTS

To compare the computational performance of the proposed protocol with existing schemes, we use the cryptographic secure millionaire protocol described in [4]. We have implemented both protocols in Matlab 7.0.1 on a Pentium 4 Dual Core 3.4GHz machine with 1GB memory. To ensure the validity of the protocols, the protocols for Bob and Alice are run separately in two processes and the two protocols exchange information using TCP/IP.

For the cryptographic protocol, Bob creates a series of tables by bitwise comparing his secret number b with every possible value of Alice's secret number a, encrypts the tables using a public-key cipher, and then transfers them to Alice. Alice decrypts the only entry of the table that is corresponding to his own number a and extracts the results. We have implemented our own 512-bit RSA public-key cipher using the long-integer operations provided by the Maple kernel within Matlab. We have run a series of comparison between random pairs of 64-bit floating point numbers. The average computation time per pair on Bob's side is 84.70 seconds. Excluding the time spent on network operations, this number reduces to 83.73 seconds. The computation times per pair for Alice are 10.72 seconds with networking and 10.43 without. Alice is faster because she does not need to generate large tables. We have pre-generated a set of random public keys used in the protocol and have excluded the time for key generation in the measurement.

On the other hand, our proposed technique runs *significantly* faster. On average, Alice takes 35.40 milliseconds with network and 1.31 milliseconds without for each comparison. Bob takes 35.41 milliseconds with network and 0.23 milliseconds without. Alice takes longer as she needs to generate a 19$^{\text{th}}$ order random polynomial. Compared with the cryptographic protocols, this is a factor of 10^4 improvement in computation time.

6. CONCLUSION

In this paper, we propose a novel security model called Quasi Information Theoretical model which enables us to design a secure thresholding protocol that is significantly faster than its cryptographic counterpart. Currently, we are refining the implementation of our cryptographic protocols based on commercial optimized public-key ciphers in order to produce more realistic comparisons.

7. REFERENCES

[1] O. Goldreich, *Foundations of Cryptography: Volume II Basic Applications*, Cambridge, 2004.

[2] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, 1996.

[3] A. C. Yao, "How to generate and exchange secrets," *27th FOCS*, pp. 162–167, 1986.

[4] S. Avidan and M. Butman, "Blind vision," in *Computer Vision – ECCV 2006*, Aleš Leonardis, Horst Bischof, and Axel Pinz, Eds. 2006, vol. 3953 of *LNCS*, pp. 1–13, Springer.

[5] W. Du, Y. Han, and S. Chen, "Privacy-preserving multivariate statistical analysis: Linear regression and classification," *proc. of the 4th SIAM Int'l Conf. on Data Mining*, pp. 222–233, 2004.

[6] N. Hu, S. Cheung, and T. Nguyen, "Secure image filtering," *proc. of the 13th Int'l Conf. Image Processing*, 2006.