Removing the Outliersof Diverse Zero-Knowledge Proof Systems using Mahalanobis Distance

Sandeep Joshi[1], Jeril Kuriakose[2]
Manipal University, Jaipur
India
sandeep.joshi@jaipur.manipal.edu
jeril@muj.manipal.edu

ABSTRACT: *Cryptography and complexity theory have gained a lot of importance because of zero-knowledge proofs. The motive behind zero-knowledge proofs are to provide an obfuscation to the verifier, so that the verifier will not understand the information sent by the prover. Zero-knowledge proofs are normally used to verify a prover's theorem to a verifier, in such a way that the verifier will not be able to discover any supplementary evidence other than the proof given to him. An enigmatic conception was formalized, that lead to the formation zero-knowledge proof systems. In this paper, we have reviewed different zero-knowledge argument / proof techniques. We have also reviewed the proof system implications in the presence of malicious prover and malicious verifier. We have removed the outliers of the experiment by using Mahalanobis distance. Examples related to zero-knowledge argument systems are also given.*

Keywords: Zero-knowledge proofs, interactive zero-knowledge, non-interactive zero-knowledge, resettable zero-knowledge, concurrent zero-knowledge.

Received: 17 February 2016, Revised 19 March 2016, Accepted 26 March 2016

1. Introduction

Zero-knowledge proofs [1] are widely used in cryptography and complex systems because of their sophisticated techniques that limit the amount of classified information that is being conveyed from the prover to the verifier. In [2], the authors have proved that it is possible for the prover to prove any theorem to the verifier without giving the slenderest clue to the verifier about the information related to the theorem. An enigmatic conception was formalised, that lead to the formation zero-knowledge proof systems.

Cryptography and complexity theory have gained a lot of importance because of zero-knowledge proofs. In [3], the authors were

able to prove that few languages were not NP-complete with the help of zero-knowledge proofs. While in cryptography [4,5], the completeness of the encryption algorithm with larger integrity has been demonstrated with zero-knowledge proofs. The motive behind zero-knowledge proofs are to provide an obfuscation to the verifier, so that the verifier will not understand the informationsent by the prover. The significant factor that differs in the zero-knowledge proofs from the traditional ones are as follows [6]:

- *Interaction*: The prover and the verifier exchange information in a co-operative way.
- *Hidden Randomization*: The information exchanged between the prover and the verifier, are randomly selected by the verifier, and the prover need not show the production / source of his proofs to the verifier.
- *Computational Difficulty*: The proofs given by the prover would be computationally difficult for the verifier to dig up and obtain the full information.

In order to understand the above mentioned factors through which zero-knowledge can be achieved, the following scenarios were suggested:

Through an abstract consequence: Consider two chemist's A and B are playing the coin toss game. During the game both of them witnessed the same amount wins and loss. After sometime chemist A left for a world tour. During his trip he was solving complex chemical equations and discovers a new chemical equation proof. Now he likes to share his research proof with the chemist B, so he writes him a letter proving his findings in a zero-knowledge manner. Chemist A in his letter did not mention his address as he is travelling. This kind of process is generally called mono-directional or non-interactive, because the interaction is only from chemist A to chemist B, and not the vice versa.

Public uncertainty: If the two chemist's try to share a random string σ in order to make it interactive process, it would make the system a weaker one rather than an interactive process. This weakens the system because its needs to be an interactive process in order to share the random string among them. Finally if the chemist's decided to have the coin tosses (to provide hidden randomness) between them to share the proof as the prover was present during the coin toss game at the first, it again would be an uncertainty because it is not possible to predict the future events in a coin toss.

***Interactive Proof Systems and Arthur-Merlin Games*:** The random string exchanged will be weakening the system through public randomness and the coin tosses among the two chemist's would enhance hidden randomness but uncertainty arises because of future prediction. A wide research has already been done the field of public randomness and hidden randomness and has already been discoursed in the complexity theory. In [2, 7], the authors consider zero-knowledge proofs as an interactive game played between the prover and the verifier, where the prover and verifier can talk to each other and exchange their information. According to [2], the prover is not shown the outcome of the coin flips, and in [7] the coin flip outcomes are shown to both the prover and the verifier. Both these models are found to equally powerful because of the randomness involved in them [8].

The central measure of the zero-knowledge proofs are about the knowledge released during interaction among the prover and verifier. Generally, a verifier will not be able to ascertain the actual proof of the prover, from the proof procedures in a zero-knowledge proof system. A remarkable property of the zero-knowledge proofs are that they will be able to convince the verifier and yielding nothing from the prover other than the required procedures. Zero-knowledge proofs are used as a very influential tool in the field of cryptography. Convincing the server without giving any additional knowledge concerning the procedures and zero information related to the user.

In this paper, we have compared the different zero-knowledge proofs. We have also shown how for few NP statements zero-knowledge proofs can be constructed and how the cryptographic protocols are affected because of the zero-knowledge proofs. Earlier analysis paper [9], studied only one particular type zero-knowledge proof and omitted the other techniques, and in our paper we have analysed most of the widely used zero-knowledge proofs.

In this paper, we have mentioned the notation '$|\,.\,|$,' which can be taken in three different ways. At some places it is used to denote the cardinality of a set, in few places it is used to denote the length of a string, and finally it is used to denote the absolute value.

We hope that the readers would understand it by the context. For a finite set *A*, *Sym(A)* is used to represent the symmetric group of *A*. *Prc* $\in$ *A [P(c)]* indicates that the probability *P(c)* holds when the uniform probability distribution of $c \in C$ taken over all the values.

***Organization of paper*:** Section 2 covers the preliminaries related to zero-knowledge proofs, section 3 analyses different zero-knowledge proof systems, and section 4 studies a couple of composition schemes used in zero-knowledge proofs. Correlation with Mahalanobis distance is given in section 5. Discussion and future events are covered in section 6, and section V concludes 7 the paper.

2. Preliminaries

2.1 Properties of zero-knowledge proof

The following are the properties that are to be satisfied by a zero-knowledge proof system:

1. Correctness: For a statement $s \in L$ (where L is a language in NP) with given W as witness, it would be possible to convince the verifier with the proofs. In simpler terms, the verifier will always 'accept' the proof, if the statement $s \in L$ is a truthful statement.

2. Soundness: A fraudulent prover will not be able to prove a false statement as true, even though the fraudulent prover has infinite computational power. In simpler terms, the verifier will continually cast-off / reject if the statement is false.

3. Zero-knowledge: A fraudulent verifier will not learn anything about the statement other than the truthfulness of the statement. The proof π given by the prover will not sufficient by the verifier to get the witness w. In simpler terms, the verifier cannot obtain any other information other than the prover's proof procedures.

In order to reduce the time complexity involved in proving, an additional property named *succinctness* has been introduced in [10]. The motive was to make the prover's proof to be more concise and error-free. The succinctness property would be quite desirable and also can play a vital role in several security applications. The succinctness property can also help in reducing the scalability problem faced in the earlier systems, due to their space complexity [11]. The authors in [12, 13], tried to reduce the scalability with the help of recursive composing proofs (i.e., proofs related to the acceptance probability of the testing verifier or verifier of the proof system).

Schemes: The set of natural numbers are represented by N. The exponent is related to concatenation (i.e., $1N$ is equivalent to 1 concatenated N times). An integer x's corresponding binary representation is indicated using σ. The length of the integer is indicated using $|x|$, and the binary representation of x is an integer if x is a string. The binary representation of x will be a real number if x is an integer. And the binary representation of x will be a cardinality of x if x is a set. The concatenation of the two binary strings σ and τ is represented as $\sigma \circ \tau$ or $\sigma\tau$.

A language L in nondeterministic polynomial time is a subset of $\{0, 1\}$. For the language L and $v > 0$, we can set $L\ v = \{x \in L, |x| \leq v\}$. A string can referred to a 'theorem' if the string falls within the set of language and if a string is outside the set of language, then it's said to be a 'false theorem.'

***Computational models*:** Consider Turing machine as an algorithm and an algorithm is said to be efficient if expected runtime of the Turing machine is in polynomial time. In order to emphasise the algorithm, whenever a Turing machine receives a single input, it is written as "*A (.),*" and when another input is received it is written as 'A *(., .).*' The '.' Is used to indicate that an input has been received.

Consider there exist a non-negative constant c, and if the size of a program is $\leq nc$ where $n \in N$ then it is expected for the Turing machine $\{Tn\}$ to halt for at least every nc steps. Then the Tn is said to be an efficient non-uniform algorithm. Non-uniformity refers to the non-uniformity in getting inputs, and this enhances the use of Turing machine to gain power, in using the algorithms.

A special oracle generally a random oracle, called random selector consisting a pair of strings *(s, S),* where S is used to conceal

a finite set of elements. The work of the oracle is to answer a query by randomly an element from S. The oracle will be returning the same element from the set, even if the same query is asked again and again. For different queries different, different elements are returned. The oracle would return two independent and randomly selected elements for two different queries s1 and *s*2 (i.e., (*s*1, *S*) and (*s*2, *S*)). In simpler terms, the special oracle can be generally considered as a hash function, which can be used for security proofs.

A Turing machine can become random selecting algorithm if given access to the random selector (special oracle). The random selecting algorithm will be the more powerful than the random selector or the unbiased coin flip, because of its uniform probability. Consider *RS* as a random selector used in coin flip, to certify the uncertainty in the selection Ei, where i is the ith coin flip, then the randomness in the selection is *Ei = RS(x* ° i, *{0, 1}*) or *RS(xi, {0, 1})*. The description of the algorithm can be minimized by using random selectors. Generally, the prover's are made 'memoryless' (i.e., the prover need not keep track his previously proved proofs), whereas for zero-knowledge it is better to have previous knowledge. The reason would be elaborated in the later section. A random selector will be a conceptual tool if used without previous history.

***Probabilistic algorithm*:** The notion *A(x)* for an input *x*, denotes the probability space which allocates the binary string σ for the probability *A*, that allocates the output σ, based on the input *x*. Given *S* as the probability space, then the algorithm '$x \xleftarrow{R} S^n$' allots a random element from S to *x*. For a finite set *F*, the algorithm '$x \xleftarrow{R} F$' allots a random element to x from the finite set F representing a uniform probability distribution on the finite set's sample points. In a predicate p(., ., …), the representation $\Pr(x \xleftarrow{R} S, x_1 \xleftarrow{R} T, ...,: p(x, x1, \ldots))$ will always return the probability as true after sequentially executing the algorithms $x \xleftarrow{R} S, x_1 \xleftarrow{R} T, \ldots$ The representation $\{x \xleftarrow{R} S, x_1 \xleftarrow{R} T, ...,:(x, x1, \ldots)\}$ the sequential execution of $x \xleftarrow{R} S, x_1 \xleftarrow{R} T, \cdots$ generates the proability space over $\{(x, x1, \ldots)\}$.

A motivating example to demonstrate the above properties is as follows:

Consider a commanding prover is proving his proofs to a verifier who verifies the proofs in polynomial time. The prover is proving his theorem '*Th*' by giving a small amount of information *x*, as input. If the input *x*Î*Th*, then the prover would be able to convince the verifier with a high probability. If the input *x* !Î*Th*, then no matter how hard the prover tries he won't be able to convince the verifier that the false proof is true. The former condition is called the correctness property, and the latter condition is called the soundness property.

The following are the conditions that are to be satisfied for correctness and soundness:

Correctness: If input $x \in Th$, then

$$\Pr\left[(p,v)(x)=1\right] \geq 1 - negl(|x|)$$

where:

p is the prover,

v is the verifier,

x is the input,

Th is the theorem that the prover is proving,

negl is the negligible function

Soundness: If input $x \notin Th$, then

$$\Pr[(p^x, v)(x) = 1] \leq negl(|x|)$$

where,

p^x , the imposter prover is using an interactive Turing machine.

3. Analysing Zero-KnowledgeProofs

There are several zero-knowledge proof systems, such as interactive zero-knowledge [14, 15], non-interactive zero-knowledge [6, 10], constant round zero-knowledge [16], concurrent Zero-knowledge [17, 18], resettable zero-knowledge [19, 20], leakage-resilient zero-knowledge [21], multiple non-interactive zero-knowledge [22]. In this paper, we will be discussing only the first four types, as it adds more importance to zero-knowledge proofs.

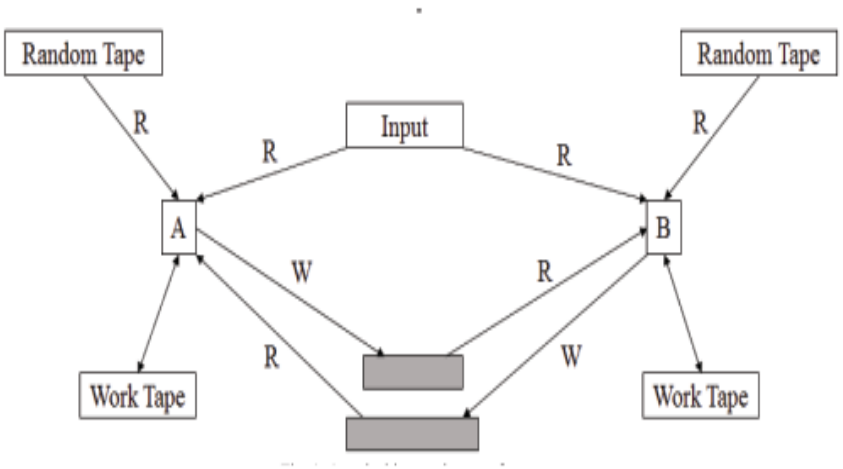

Figure 1. A typical interactive proof system

3.1 Interactive zero-knowledge

An interactive proof is a two-party protocol tries to prove his proof to the verifier for the language *L*, and the verifier tries to verify the proof within the probabilistic polynomial time. The completeness and soundness property states that a prover can prove his proof of the input *x* with a higher probability if the input $x \in L$, and if the input $x! \in L$ then the prover cannot prove his proofs to the verifier. Fig. 1. Shows a typical interactive system. The properties for interactive zero-knowledge proofs that uses Turing machine are as follows:

Correctness: If input $x \in Th$, then,

$$\Pr[(p, v)(x) = 1] \geq 1 - negl(|x|)$$

where:

p is the prover,

v is the verifier,

x is the input,

Th is the theorem that the prover is proving,
negl() is the negligible function

Soundness: If input $x \notin Th$, then

$$\Pr[(p^x, v)(x) = 1] \leq negl(|x|)$$

where,

p^x, is a cheating prover.

The above mentioned soundness property is valid only for probabilistic polynomial time Turing machine.

***Zero-knowledge*:** An interactive proof is said to have the zero-knowledge if the verifier is not able to ascertain the theorem for the proofs. Considering a cheating verifier *V**, who is represented by *viewv*(x, z)*, replicates the interaction with the prover p and starts simulating using a probabilistic polynomial time machine *S*. Generally, *S* is referred to as simulator. The additional information obtained from the input x of V* and S are denoted by $z \in \{0, 1\}$. It must be computationally indistinguishable for every $x \in L$, $z \in \{0, 1\}$, *viewv*(x, z)*, and *S(x, z)* by the cheating verifier *V** to get any other slightest information apart from the information given in the proof. This makes the proof zero-knowledge. In simpler terms, the cheating verifier *V** must not be able to ascertain any auxiliary input from any simulations.

***Polynomial time*:** Consider a non-deterministic Turing machine *Tn*, then *Tn(x, r)* represents the Turing machine *n*, with input *x* and r denoting the total bit sequence. If the bit sequence *r* is finite then *Tn(x, r)* is finite. The expectation of a polynomial *P* for all $x \in \{0, 1\}^*$, taken over the infinite / total bit sequence *r*, is confined over by *P(|x|)*. In simpler terms we can say as the following:

$$E_r(T_n(x, r)) \leq Q(|x|) for\ all\ x$$

where:
Er is the expectation of the polynomial P taken over the infinite sequence r.

A motivating example to demonstrate interactive zero-knowledge with the help of private communication done with the help of interactive Turing machine, and is as follows:

An interactive Turing machine has a deterministic Turing machine consisting if six tape. It also has an input tape, random tape, communication tape, all with read-only ability, and output tape and communication tape with write-only ability, and a work tape with read and write abilities. The input generally appears on the input tape and the random tape can be used for the outcomes of an infinite coin flips. The output generally appears on the output tape when the halts, the write-only communication is when the contents / messages from the machine is transferred to another device, and the read-only communication is when the messages / contents are received by the machine.

Complexity: Generally, the input / the contents of the input tape is used to measure the complexity of a Turing machine. Consider an example, where a polynomial *P* containing *M* steps that's need to be performed on x input, where the maximum steps for the polynomial *P* in the Turing machine can be *P(|x|)*. For measuring the complexity the contents from are the random are considered, rather than the steps.

***Turing machine pairs*:** Whenever the communication of tapes of two Turing machines are shared then they are considered to be interactive Turing machine pairs. The read-only communication of one machine corresponds with the write-only communictionof the paired machine, and vice versa. The computing phases taken by both the machines are done in accordancewith the alternating sequence. At a given time, no two machines can have the read-only communication or write-only communication are once. With the help of a special idle state, the pair of Turing machines change the alternatingsequence machines change the alternating sequence.

The following notations can be formed from the above statements:

1. Consider an interactive Turing machine A, which has *A(x, r; a1, …, an)*, where *x* is the input for the interactive Turing machine *A*, *r* is the contents of the random tape, and *a1, …, an* represents the messages that has been received.

2. For an interacting pair of Turing machines *A* and *B*, *[(B(y), A(x))]* denotes that machine A has the input x and acts in the active mode, and machine *B* has the input y and acts in the idle mode (i.e., receives messages from machine *A*).

The following properties hold true for the above notations:

Correctness: For input $x \in L$ and constant c > 0,

$$\Pr\left[\left(p(x), v(x)\right) = 1\right] \geq 1 - |x|^{-c}$$

Soundness: If input $x ! \in L$ and constant c > 0,

$$\Pr\left[\left(p^{*}(x), v(x)\right) = 0\right] \geq 1 - |x|^{-c}$$

3.2 Non-interactive zero-knowledge

Non-interactive zero knowledge is carried out with the help of special oracle, in order to provide security to the proofs. For the pair *(x, w)*, where *x* is the input and *w* is the witness, let *R* be the efficiently computable relation by the oracle that will be holding the pair *(x, w)*. *R* contains the statements from the language *L*, and *K* is a setup algorithm for the language *L* with a prover p and a verifier v. A common reference binary string σ is generated by the setup algorithm *K*. The input taken by the prover p is *(*σ*, x, w)* and checks whether *(x, w)*$\in$R, in order to satisfy the above condition. If *(x, w)* $\in$ *R*, then a proof string *p* is generated which is given to the verifier *v*. If a proof string is not generated then the output fails. Given an algorithm tuple *(b, p, v)*, can be called as non-interactive zero-knowledge by the following:

Correctness: For any cheating *Q*,

$$\Pr\left[\sigma \leftarrow b\left(1^{b}\right); (x, w) \leftarrow Q(\sigma); \pi \leftarrow p(\sigma, x, w) : v(\sigma, x, \pi) = 1 if (x, w) \in R\right] \geq 1 - negl(b)$$

Soundness: For any cheating *Q*,

$$\Pr\left[\sigma \leftarrow b\left(1^{b}\right); (x, \pi) \leftarrow Q(\sigma) : v(\sigma, x, \pi) = 1 if x ! \in L\right] \leq negl(b)$$

Generally, the algorithm *(b, p, v)* is said to be non-interactive when the soundness condition embraces against the probabilistic polynomial time cheater.

***Zero-knowledge*:** For all cheating *Q*, if it has a simulator *S = (S1, S2)*, then the particular algorithm *(b, p, v)* is said to have zero-knowledge.

$$\Pr\left[\sigma \leftarrow b\left(1^{k}\right) : Q^{p(\sigma, \ldots)}(\sigma) = 1\right] \overset{c}{\equiv} \Pr\Pr\left[(\sigma, \tau) \leftarrow S_1\left(1^{k}\right) : Q^{S'(\sigma, \tau, \ldots)}(\sigma) = 1\right]$$

where*S'(*σ*,* τ*, x, w) = S2(*σ*,* τ*, x)* if *(x, w)* $\in$ *R* and output fails otherwise.

Honest prover state construction: In [21], the authors have made an extension to the above zero-knowledge condition and named it as honest prover state construction.

For all cheating Q, if it has a simulator $S = (S1, S2, S3)$, then the particular algorithm is said to have honest prover state condition for relation R.

$$\Pr\Big[\sigma \leftarrow b(1^b) : Q^{pR(\sigma,\ldots)}(\sigma) = 1\Big] \underset{\equiv}{C} \Pr\big[(\sigma,\tau) \leftarrow S_1(1^k) : Q^{SR(\sigma,\tau,\ldots)}(\sigma) = 1\big]$$

Initial: A large prime P is chosen by the prover and the verifier, P is selected such a way that $P - 1$ is factorable. A confidential parameter k, and a generator Z_p^* are also agreed upon by the prover and the receiver. The initial step (current step) is not considered for the counting the total rounds.	
v_1	**p_1**
v_1 selects $R \in Z_p^*$ and $\vec{y} \in \Sigma^k$, $(\vec{e},\vec{r}) = \beta(\vec{y})$, $R, \vec{e}\rightarrow$	$\forall i,\ 1 \le i \le k, \vec{c}_i = \sigma(\Psi)$, and $(\vec{x}_i,\vec{z}_i) = \beta(\vec{c}_i)$. $\forall x_i[j]$, p chooses $\vec{b}_{ij} \in \{0,1\}^k$, computes $(\vec{u}_{ij},\vec{v}_{ij}) = \beta(\vec{b}_{ij})$, $\vec{x}_i \& \vec{u}_{ij}\rightarrow$
v_2	**p_2**
$\forall i, j$ verifier chooses $\vec{h}_{ij} \in \{0,1\}^k$ randomly, $\vec{h}_{ij}\rightarrow$	$\forall i, j, m$ if $h_{ij}[m] = 0$, prover changes $w_{ij}[m] = v_{ij}[m]$ & $t_{ij}[m] = b_{ij}[m]$, if $h_{ij}[m] = 1$, prover changes $w_{ij}[m] = v_{ij}[m] - z_i[m]$ & $t_{ij}[m] = 0$ (Iff $c_i[j] = b_{ij}[m]$), $\vec{w}_{ij} \& \vec{t}_{ij}\rightarrow$
v_3	**p_3**
if $h_{ij}[m] = 0$, checks $\phi(w_{ij}[m], v_{ij}[m], t_{ij}[m])$, if $h_{ij}[m] = 1$, checks $u_{ij}[m] = \alpha^{w(ij)[m]} x_i[j]$, 0 or $1\rightarrow$	checks $\phi(\vec{e},\vec{r},\vec{y})$, if 0 prover quits, else if $y[i] = 0$, $\vec{z}_i \& \vec{c}_i\rightarrow$, of $y[i] = 1$, computes $\vec{d}_i = \gamma(\Psi, \vec{c}_i, \vec{a}_i)$, $\vec{z}_i \& \vec{d}_i\rightarrow$
v_4	
if $y[i] = 0$, checks $\vec{c}_i \in \xi_\psi$ and $\varphi(\vec{x}_i,\vec{c}_i,\vec{z}_i)$, if $y[i] = 1$, checks $\vec{d}_i \in \xi_\psi$ and $\varphi(\vec{x}_i,\vec{z}_i,\vec{d}_i)$.	

Figure 2. Working of constant round zero-knowledge.

where

$pR(\sigma, x, w)$ computes $r \leftarrow \{0, 1\}lp(b)$,
$\pi \leftarrow p(\sigma, x, w; r)$ returns (σ, w, r),
$SR(\sigma, \tau, x, w)$ computes $\rho \leftarrow \{0, 1\}lS(k)$,
$\pi \leftarrow S2(\sigma, \tau, x; \rho)$,
$r \leftarrow S3(\sigma, \tau, x, w, \rho)$ and returns (π, w, r),
the output of both oracles fail if $(x, w)\ !\epsilon\ R$.

4. Co-Relating With Mahalanobis Distance

Posterior probability is being used in Mahalanobisdistance to identify the outliers. The range measurebetween two beacon nodes can identified by usingMahalanobis distance when the beacon nodes areassociated with two or more locationcoordinates.

This reduces the error obtained during localizationby excluding the unwanted location coordinates. Theredundant location coordinates are identified by usinga centroid value and corroborating it with the locationcoordinates. The trilateration was used as the centroidvalue in our instance. The identification of range measureusing Mahalanobis distance is as follows:

$$d(mahalanobis) = \left\{\left[(x_j, y_j) - (x_i, y_i)\right]^T * C^{-1} * \left[(x_j, y_j) - (x_i, y_i)\right]\right\}^{1/2}$$

where:

d(mahalanobis) is the distance between two anchor nodes, (x_i, y_i)&(x_j, y_j) are the location coordinates of the two anchor nodes,

C is the sample covariance matrix

The variance-covariance matrix *C* is constructed in order to gauge Mahalanobis distance,

$$C = \frac{1}{(n-1)}[(x, y)]^T[(x, y)]$$

where:
(x, y) is the matrix containing the location coordinates, is the number of nodes.

The variance-covariance matrix in the presence ofmultiple location coordinates will be converted asfollows:

$$C = \begin{bmatrix} \sigma_1^2(x_i, y_i) & \rho_{12}\sigma_1(x_i, y_i)\sigma_2(x_j, y_j) \\ \rho_{12}\sigma_1(x_i, y_i)\sigma_2(x_j, y_j) & \sigma_2^2(x_j, y_j) \end{bmatrix}$$

where:
σ_1^2&σ_2^2 are the variances of the multiple location references, $\rho_{12}\sigma_1(x_i, y_i)\sigma_2(x_j, y_j)$ is the covariance between the multiple location references.

The value of C^{-1} is computed as follows:

$$C^{-1} = \begin{bmatrix} \frac{\sigma_2^2(x_j, y_i)}{|C|} & \frac{-\rho_{12}\sigma_1(x_i, y_i)\sigma_2(x_j, y_j)}{|C|} \\ \frac{-\rho_{12}\sigma_1(x_i, y_i)\sigma_2(x_j, y_j)}{|C|} & \frac{\sigma_1^2(x_j, y_j)}{|C|} \end{bmatrix}$$

where:
$|C|$ is the variance covariance matrix's determinant and is equal to $\sigma_1^2\sigma_2^2(1-\rho_{12}^2)$

We modified the Mahalanobis distance function toidentify the range between multiple location coordinatesto the trilateration point, as follows:

$$d(\delta_i) = \{[(x_i, y_i) - (x_c, y_c)]^T * C^{-1} * [(x_i, y_i) - (x_c, y_c)]\}^{1/2} \text{for} i = 1, 2, \ldots, n$$

where:
$d(\delta_i)$ is the distance between centroid and i^{th} anchor node,

(x_i, y_i) is the location coordinate of the i^{th} anchor node,
(x_c, y_c) is the location coordinate of the centroid.

The new distances obtained using Mahalanobis distance, are compared using posterior probability; leading to the confirmation of the anchor nodes adversity. The distances obtained using this method is marginally accurate than the previous method.

We have also compared our work with trilateration based location technique. Fig. 3, and fig. 4 shows the average error obtained while using trilateration technique and Mahalanobis distance, respectively.

5. Discussions and Future Events

Using a Zero-knowledge proof protocols would increase the system's cost and constrain on time remains based on the proof that is required to convince the verifier. Hypothetically zero-knowledge proof protocols will be efficient, because of the polynomial time and cost. Zero-knowledge proof protocols are used in many cryptographic areas such as passwordauthenticated key agreement [23], Secure Remote Password protocol [24], and Feige–Fiat–Shamir identification scheme [1]. Multi-prover interactive proof systems [25] can allow the verifier to 'debrief' all the multiple prover's instead of one prover, thus avoiding the chances of being misled.

Authentication systems has highly motivated the Zeroknowledge proofs, where one party (client / prover) wants to prove its individuality to the second party (server / verifier) with the help of some confidential information such as login authentication details, and the client / prover do not want the server / verifier to learn or understand anything about the confidential information, thus making it 'zero-knowledge proof of knowledge'.

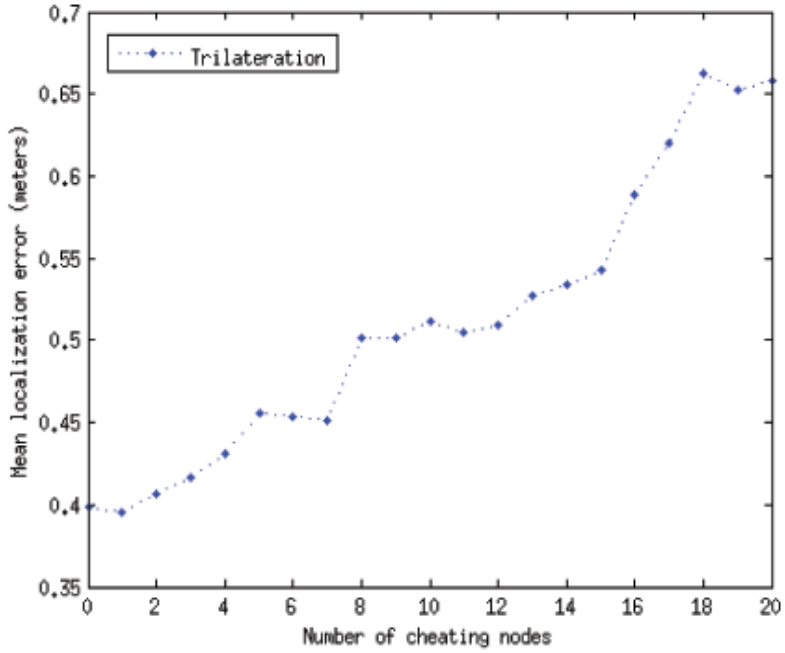

Figure 3. Mean error obtained while using trilateration technique.

It is not possible to have a perfect / flawless zeroknowledge proof system, unless all interactive proofs have co- NP (which is a complexity class) language. In order to identify the power of the interactive proof systems, the requisite for error probability is not essential [26, 27]. Completeness allows error probability in the interactive proof system.

The amount of computational resources used are not considered when proving a theorem. The probabilities of completeness and soundness can be increased up and down by looping the proving process several times. The non-interactive proof systems are generally immune towards cheating verifiers. The following ensures the strong and perfect completeness:

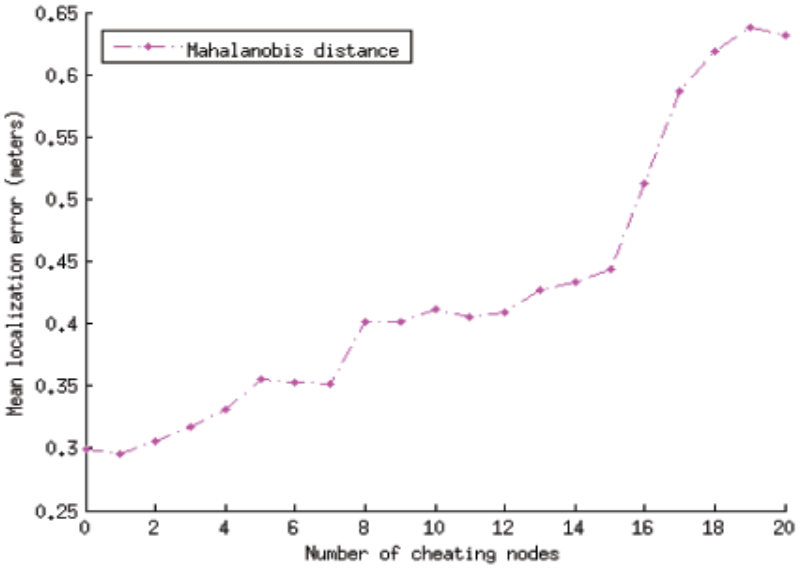

Figure 4. Mean error obtained while using Mahalanobis distance.

Strong completeness:

$$\Pr\left[\sigma \xleftarrow{R} \{0,1\}^{n^c}; x \xleftarrow{R} choose\ L(\sigma); proof \xleftarrow{R} p(\sigma, x) : v(\sigma, x, proof) = 1\right] > 1 - 2^{-n}$$

Perfect completeness:

$$\Pr\left[\sigma \xleftarrow{R} \{0,1\}^{n^c}; proof \xleftarrow{R} p(\sigma, x) : v(\sigma, x, proof) = 1\right] = 1$$

6. Conclusion

In this paper, we have analysed different zero-knowledge argument / proof techniques. We have also analysed the proof system implications in the presence of malicious prover and malicious verifier. From our analysis we have identified that it is not possible to have a perfect / flawless zero-knowledge proof system, unless all interactive proofs have co-NP (which is a complexity class) language.Using a Zero-knowledge proof protocols would increase the system's cost and constrain on time remains based on the proof that is required to convince the verifier, and finding a solution for the time and cost constrain can be considered for future work.

References

[1] Feige, Uriel., Fiat, Amos ., Shamir, Adi (1988). Zeroknowledge proofs of identity. *Journal of Cryptology* 1 (2) 77-94.

[2] Goldwasser, Shafi., Micali, Silvio., Rackoff, Charles (1985). The knowledge complexity of interactive proofsystems, *In*: Proceedings of the Seventeenth Annual ACM symposium on Theory of computing. ACM, 1985.

[3] Boppana, Ravi B., Hastad, Johan ., Zachos, Stathis (1987). Does co-NP have short interactive proofs?. *Information Processing Letters* 25 (2) 127-132.

[4] Benhamouda, Fabrice, et al. (2014). Better zero-knowledge proofs for lattice encryption and their application to group signatures. Advances in Cryptology–ASIACRYPT 2014. Springer Berlin Heidelberg. 551-572.

[5] Gentry, Craig, et al. (2014). Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology* 1-24.

[6] Blum, Manuel, Paul Feldman, and Silvio Micali. "Noninteractive zero-knowledge and its applications. *In*: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. ACM, 1988.

[7] Kuriakose, Jeril, et al. (2014) A Review on Mobile Sensor Localization. Security in Computing and Communications. Springer Berlin Heidelberg, 2014. 30- 44.

[8] Goldwasser, Shafi., Sipser, Michael (1986). Private coins versus public coins in interactive proof systems. *In:* Proceedings of the Eighteenth Annual ACM symposium on Theory of computing. ACM.

[9] Wu, Huixin., Wang,.Feng (2014). A Survey of Noninteractive Zero Knowledge Proof System and Its Applications. *The Scientific World Journal* 2014

[10] Ben-Sasson, Eli, et al. (2014). Succinct non-interactive zero knowledge for a von Neumann architecture. USENIX Security.

[11] Ben-Sasson, Eli, et al. (2014). Scalable zero knowledge via cycles of elliptic curves. Advances in Cryptology– CRYPTO 2014. Springer Berlin Heidelberg. 276- 294.

[12] Bitansky, Nir, et al.(2013). Recursive composition and bootstrapping for snarks and proof-carrying data. *In*: Proceedings of the forty-fifth annual ACM symposium on Theory of computing. ACM.

[13] Kuriakose, Jeril, et al. (2014). A review on localization in wireless sensor networks. Advances in signal processing and intelligent recognition systems. Springer International Publishing, 2014. 599-610.

[14] Goldreich, Oded, Micali, Silvio., Wigderson, Avi (1991). Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM (JACM)* 38 (3) 690-728.

[15] Goldreich, Oded., Micali, Silvio., Wigderson, Avi. (1986). Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. *In*: 27th Annual Symposium on Foundations of Computer Science. IEEE, 1986.

[16] Goldreich, Oded, and Ariel Kahan. (1996). How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology* 9 (3) 167-189.

[17] Dwork, Cynthia., Naor, Moni Sahai, Amit.(1998). Concurrent zero-knowledge. *In*: Proceedings of the Thirtieth Annual ACM symposium on Theory of Computing. ACM.

[18] Pandey, Omkant., Prabhakaran, Manoj., Sahai, Amit (2015). Obfuscation-based non-black-box simulation and four message concurrent zero knowledge for np. Theory of Cryptography. Springer Berlin Heidelberg, 2015. 638- 667.

[19] Vikram Raju, R., (2014). A review on host vs. Network Mobility (NEMO) handoff techniques in heterogeneous network. Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2014 3rd International Conference on. IEEE, 2014.

[20] Chung, Kai-Min, et al. (2014). 4-round resettably-sound zero knowledge. Theory of Cryptography. Springer Berlin Heidelberg. 192-216.

[21] Garg, Sanjam., Jain, Abhishek., Sahai, Amit (2011). Leakageresilient zero knowledge. Advances in Cryptology–CRYPTO 2011. Springer Berlin Heidelberg, 297- 315.

[22] Feige, Uriel, Lapidot, Dror., Shamir, Adi (1990). Multiple non-interactive zero knowledge proofs based on a single random string. Foundations of Computer Science, 1990, *In*: Proceedings, 31st Annual Symposium on. IEEE.

[23] Zhang, Liping., Tang, Shanyu., Cai, Zhihua. (2014). Efficient and flexible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card. *International Journal of Communication Systems* 27(11) 2691-2702.

[24] Wu, Thomas D. (1998). The Secure Remote Password Protocol. NDSS. Vol. 98.\.

[25] Ito, Tsuyoshi. (2014). Parallelization of entanglement-resistant multi-prover interactive proofs. *Information Processing Letters* 114 (10) 579-583.

[26] Goldreich, Oded., Mansour, Yishay., Sipser, Michael (1987). Interactive proof systems: Provers that never fail and random selection. Foundations of Computer Science, 1987. 28th Annual Symposium on. IEEE, 1987.

[27] Kuriakose, Jeril., Amruth, V., Vikram Raju. R. (2015). Secure Multipoint Relay Node Selection in Mobile Ad Hoc Networks. Security in Computing and Communications. Springer International Publishing. 402-411