International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

Security and Performance Enhanced Asynchronous Secure Multiparty Computation (ASMC)

Mr. S. Samson Dinakaran
Assistant Professor/Department of CS
VLB Janakiammal College of Arts & Science,
Coimbatore, Tamil Nadu, India.

Dr. M. Devapriya
Assistant Professor/Department of CS
Government Arts College,
Coimbatore, Tamil Nadu, India.

***Abstract*:- One of the most primary difficulties in security is the problem of Secure Multi-party Computation (SMC). The problem is simple to solve if consider the existence of a Trusted Third Party (TTP) gathers the inputs, computes the function and shares the results to everyone. However, the problem is very difficult if consider that there is no TTP available and parties may misbehave at all. To solve this challenging in SMC, a smart card-based security model such as TrustedPals was proposed and integrated with the failure detection and consensus in both synchronous and asynchronous network settings. However, this Asynchronous SMC (ASMC) model uses fixed-size of messages in fixed time periods. This can affect the tradeoff between security and performance to provide better security in worse performance. Therefore in this article, an adaptive model is proposed within the failure detector implementation for achieving an acceptable tradeoff between security and performance. This proposed model has the aim of fine-tuning the network settings and services of failure detector to balance the security in the cost of worse performance. The construction of this proposed tradeoff model consists of Performance Enhanced ASMC (PEASMC) and Security Enhanced ASMC (SEASMC) models that improve the set of metrics for measuring performance and security, respectively. In addition to these models, Performance and Security Enhanced ASMC (PSEASMC) model that improves the tradeoff objective function using both performance and security metrics together and the simultaneous parameter estimation via experiments. By using PSEASMC model, an acceptable tradeoff is computed by reducing the tradeoff objective function instead of automatically switching from one security configuration to another until the required tradeoff is achieved. Finally, the experimental results show the performance efficiency of the proposed PSEASMC model compared with the existing ASMC model in terms of differential privacy, latency, average accuracy and worst-case accuracy.**

Keywords*— *Secure multi-party computation, Smart cards, Failure detection, Fault-tolerance, Consensus, Tradeoff, Security performance.

I. INTRODUCTION

Secure Multi-party Computation (SMC) or Multi-Party Computation (MPC) is a subtype of cryptography with the aim of developing methods for parties to jointly compute the function over their inputs when maintaining those inputs. In contrast to the conventional cryptographic processes where cryptography ensures security and integrity of transmission or storage and the opponent is outside the system of participants, the cryptography in this model can protect the participant's privacy from each other. These SMC techniques have been rapidly developed in the previous decades with costs lowered by orders of magnitude. For instance, by using SMC technique, garbled circuit evaluation has been achieved at speeds of 1.15 billion gates/seconds and secret sharing supported privacy-preserving location services were available at Real World Crypto 2015.

Regardless of these significant gains, SMC cannot be considered adequately practical for utilization in a majority of applications where real-time performance is needed. This is specifically proper for techniques based on completely homomorphic encryption or secret sharing. Consider a set of parties who desire to properly compute few common functions of their local inputs when maintaining their local information as private as possible, however who do not trust each other, not the channels by which they communicate. This is the typical dilemma of SMC. SMC is a very common security problem i.e., it can be used for solving different real-time issues such as shared voting, private bidding and online auctions, distributing of signature or decryption functions, etc. Unfortunately, solving SMC without additional considerations is very expensive in terms of communication i.e., number of messages, resilience i.e., amount of redundancy and time i.e., number of synchronous rounds.

TrusedPals [1] is a smart card-based security framework to solve SMC which facilitates much more effective solutions to the problem. Theoretically, TrustedPals considers a distributed system in which processes are locally deployed with tamper-proof security modules. In conventional techniques, the processes are executed as a Java desktop application and security modules are recognized by using Java Card Technology enabled smart cards [2], tamper-proof Subscriber Identity Modules (SIM) [3] similar to those used in mobile phones or storage cards with integral tamper-proof processing devices. To solve SMC in the TrustedPals framework, the function F is coded as a Java function and is shared within the network in a primary stage. After that, processes hand their input value to their security module and the framework achieves the secure distribution of the input values. At last, all security modules compute the function and return the outcome to their process. The network of security modules sets up secret and legitimate channels between each other and operates as a secure overlay within the distribution phase.

Hence, TrustedPals facilitates the security problem avoidance in SMC to a problem of fault-tolerant synchronization. Moreover, the reduction from security to fault-tolerance makes a new set of authentication requirements concerning the integration of a fault-tolerant algorithm into a secure system. The key definition of TrustedPals and its performance considered the synchronous

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

network configuration i.e., a configuration in which all significant timing parameters of the network are bounded and known. This provides TrustedPals susceptible to sudden deviations in the network delay and so not very appropriate for networks deployment. Cortinas et al. [4] explored how to make TrustedPals applicable in a network configuration with less synchrony. Also, they proved how to solve the Asynchronous version of SMC (ASMC) using asynchronous synchronization algorithms motivated by the current outcomes in fault-tolerant distributed computing. They used an asynchronous consensus algorithm and encapsulate timing hypotheses within a device known as a failure detector [5]. Instead of correct processes, well-connected processes were considered i.e., those processes which are able to compute and communicate without omissions with a majority of processes. However, it generates fixed size messages in fixed time intervals. Therefore, the size of the payload field was required to choose for finding an acceptable tradeoff between security and performance such that a message size offers better security in cost of worse performance.

Hence in this article, an adaptive model for the tradeoff between security and performance is proposed within the failure detector implementation. The major objective of this tradeoff model is adjusting the configurations and service operations of the failure detector in order to balance the QoS (Quality-of-Service) performance and security. In this model, the best tradeoff is computed by reducing the tradeoff objective function which incorporates service performance and security metrics together instead of automatically switching from one security configuration to another until the required tradeoff is achieved. The construction of this model engages the PEASMC and SEASMC for measuring the performance and security individually by improving the set of metrics. Also, PSEASMC model is developed by using a tradeoff objective function that includes both service performance and security metrics together simultaneously to estimate the tradeoff model's parameters. Thus, the payload size is selected and an acceptable tradeoff between security and performance is achieved efficiently.

The rest of the article is structured as follows: Section II presents the previous researches on SMC for different applications. Section III explains the methodology of the proposed model. Section IV illustrates the experimental results compared with the existing models and Section V concludes the entire discussion.

II. LITERATURE SURVEY

Chung et al. [6] designed a new security model with MPC for security. The main goal of this study was designing a security protocol for the requirements of the models by using the rational and universal composability models. In this study, the structures of secret sharing agreement, fair computation agreement, bit analysis agreement and the applications of these agreements on SMC were investigated. Also, the combination of a rational model and security MPC were further analyzed to propose a new rational secret sharing method with two rational participants. The particular permission secret sharing was proposed based on the new threshold value to construct the respondent rational secret sharing protocol. However, this protocol was computationally expensive.

Bogdanov et al. [7] proposed a secure system for jointly collecting and analyzing the financial data for a consortium of ICT companies. In this system, secret sharing and secure MPC techniques were used for ensuring each participant's privacy. However, more effort was required to make application deployment and administration easier. Thoma & Franchetti [8] proposed an SMC based privacy preserving protocol for smart meter based load management. By using this SMC and an appropriately designed electricity plan, the utility was able to perform the real-time demand management with individual users without knowing the actual value of each user's consumption data. By using homomorphic encryption, the billing was more secure and verifiable. However, this protocol was computationally expensive.

Abidin et al. [9] proposed a local electricity trading market based on SMC that permits the user to trade excess electricity among them in a decentralized and privacy-preserving manner. Users who have more electricity generated by their renewable energy sources than their requirement may trade this electricity to other users by using a bidding mechanism based on SMC. According to the bidding charges, the clearance cost was computed at which the electricity may be traded. Moreover, bid selection was performed including other market tasks. However, computational complexity was high and the information leakage was also high.

Mustafa et al. [10] proposed an MPC-based protocol that facilitates the suppliers and grid operators in order to collect the user's electricity metering data in a secure and privacy-preserving manner. This protocol was designed for a realistic scenario where the data required to be transmitted to different parties such as grid operators and suppliers and users can switch supplier at any point in time. As well, an appropriate computation of electricity transmission, distribution and grid balancing cost was performed in a privacy-preserving manner. However, this protocol has a high computational complexity.

Alexopoulos et al. [11] proposed an anonymous messaging system named MCMix that completely hides communication metadata and can scale in the order of hundreds of thousands of users. This approach was used for isolating two suitable functionalities such as dialing and conversation that while used in succession, realize anonymous messaging. Initially, SMC was applied and proceeded to realize them. After that, an implementation using Sharemind was presented to build an anonymous MPC system. But, this system has a high communication cost and running time.

III. PROPOSED METHODOLOGY

In this section, the proposed adaptive models for tradeoff computation are explained in brief. Consider the average delay of a payload is D, payload traffic is T and the security configuration vector is $SCV = \{F, A, l, p\}$ where F denotes the security function, A denotes the algorithm that consists of a set of deterministic automata, l denotes the payload size and p denotes the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

protection percentage. The delay D is the average time taken by the security mechanism to take in a payload, protect the payload with the parameters specified in SCV and return the payload back. The payload traffic T is defined as the number of payloads handled by the security mechanism per second. The performance and security allow quantitatively computing how much protection an SCV can provide and how much performance will be reduced by that SCV. Therefore, the tradeoff between performance and security is controlled by adjusting the parameters in the SCV. In this proposed model, three different tradeoff strategies are used for maximizing the performance, security and controlling the performance and security according to user's preferences, correspondingly.

3.1 Minimum Requirement Validation

Generally, the tradeoff between performance and security is executed via resource allocation. Initially, the considered trusted system has to allocate the specific number of resources for both performance and security in order to satisfy their minimum requirements. After that, if there are more resources, the trusted system can allocate the available resources for better performance or security. Therefore, to verify whether the tradeoff is possible, initially it is required to make ensure that the minimum performance and security requirements can be satisfied.

Consider the trusted system requires that the delay must be less than D_0, the payload traffic will be up to T_0 and the success probability of an attacker with capability c must be less than S_0. Based on the minimum security requirement,

$$S(l,p,c) \le S_0 \Rightarrow p \ge (1-S_0)/\big(1-cv(l)\big) \quad (1)$$

For constant payload traffic t, the delay metric is given as follows:

$$D(SCV,t) = \begin{cases} 0, & if\ t = 0 \\ a + a'p, & if\ 0 < t \le T_1 \\ a + a'p + be^t + b'e^{tp}, & T_1 < t < T_2 \end{cases} \quad (2)$$

Where a, b, T_1 and T_2 are four parameters related to SCV but independent from t. After that, to satisfy the minimum performance requirement from the metric of delay (2), the following condition is given:

$$\begin{cases} p \le (D_0 - a)/a', & if\ T_0 \le T_1 \\ p \le ln((D_0 - a - be^{T_0})/b')/T_0, & if\ T_1 < T_0 \le T_2 \end{cases} \quad (3)$$

Thus, to check whether the minimum performance and security can be satisfied, it is needed to verify whether there exist algorithms and payload sizes l satisfying the following condition:

$$\frac{1-S_0}{1-cv(l)} \le \begin{cases} (D_0 - a)/a', & if\ T_0 \le T_1 \\ ln(D_0 - a - be^{T_0})/T_0, & if\ T_1 < T_0 \le T_2 \end{cases} \quad (4)$$

All parameters including $a, a', b, b', v(l), T_1$ and T_2 are determined by the security algorithm and payload size l. Since any trusted system only supports a limited number of security algorithms and payload sizes, it can verify whether both the minimum performance and security requirements can be satisfied by enumerating all supported security algorithms and payload sizes to find if the above condition can be satisfied.

3.2 Tradeoff Objective Function

While both minimum performance and security requirements are satisfied, the trusted system can utilize the available resources for achieving better performance or security. To achieve better security, the trusted system can utilize either a stronger algorithm with a longer payload size or a larger protection percentage as follows:

$$S(l,p,c) = (1-p) + pcv(l) \quad (5)$$

Conversely, the performance metric (2) shows that a stronger algorithm with a longer payload size will generate larger parameters a, a', b, b' and smaller T_1, T_2 and a larger protection percentage will provide the delay increase faster with the payload traffic. Both of these will minimize performance. Therefore, to control the tradeoff between performance and security, the performance metric and security metric are combined together as a tradeoff objective function. The tradeoff objective function between the performance metric (2) and the security metric (5) can be defined as:

$$G(SCV,t) = \alpha D(SCV,t) + \beta S(l,p,c) \quad (6)$$

Where α and β are two weighting factors representing the trusted system client's preferences on performance and security, respectively. To normalize the function G, consider that $\alpha + \beta = 1$ and $0 \le \alpha, \beta \le 1$. After that, the optimized tradeoff between D and S is computed by minimizing the value of G with constraints $D(SCV, T_0) \le D_0$ and $S(l,p,c) \le S_0$.

3.2.1 Performance Enhanced ASMC (PEASMC)

The performance-biased objective function is a tradeoff objective function that tries to maximize the performance without infringing the minimum security requirement. For the tradeoff objective function (6), the performance-biased objective function sets the weighting factor β of the security to 0. In this case, the tradeoff objective function is equivalent to minimizing the delay D. While the TrustedPals with failure detector and consensus algorithm and payload size are predetermined, the performance biased tradeoff must always utilize the minimum protection percentage $\frac{1-S_0}{1-cv(l)}$ computed in (4).

3.2.2 Security Enhanced ASMC (SEASMC)

The security biased tradeoff function is a tradeoff objective function that tries to maximize security without infringing the minimum performance requirements. For the tradeoff objective function (6), the security biased tradeoff function sets the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

weighting factor α of the performance to 0. In this case, the tradeoff objective function is equivalent to minimizing the attacker's success probability S. While the TrustedPals with failure detector and consensus algorithm and payload size are predetermined, the upper limit for the protection percentage is computed from the minimum performance requirements as given in (3).

3.2.3 Performance and Security Enhanced ASMC (PSEASMC)

If the trusted system client's preferences on both performance and security i.e., the weighting factors α and β do not modify with the real-time performance and security conditions, such as tradeoff objective function is called as a linear tradeoff objective function similar to the PEASMC and the SEASMC. In contrast, if the weighting factors α and β are associated with the current performance and security, then such a tradeoff objective function is known as a non-linear tradeoff objective function. To define this tradeoff model, different ways are available; however, all definitions must have the following properties:

- The weighting factor of performance or security increases while the performance/security achieves the minimum performance/security requirement.
- The minimum performance or security requirements are essential for trusted systems. Thus, while the minimum performance/security requirements are not satisfied, the weighting factor of performance/security becomes infinite.

Hence, a possible non-linear tradeoff objective function for the performance metric (2) and the security metric (5) can be defined as follows:

$$G'(SCV,t) = e^{a'/(D_0 - D(SCV,t))} D(SCV,t) + e^{b'/(S_0 - S(SCV,t))} S(l,p,c) \qquad (7)$$

The weighting factors of delay and security can increase faster with larger delay and security and will become infinite while the minimum delay and security requirements are not satisfied. Thus, an acceptable tradeoff is computed to balance the security and performance such that the payload size offers better security in worse performance.

IV. RESULTS AND DISCUSSIONS

In this section, the performance effectiveness of proposed models PEASMC, SEASMC and PSEASMC is evaluated and compared with the existing model ASMC by using Java. The comparison is made in terms of differential privacy, latency and accuracy measure. Those performance metrics are described below:

- **Differential Privacy:** Privacy is measured by differential privacy. The required privacy level of i^{th} party is denoted as ε_i. An algorithm A is ε_i-differentially confidential for the i^{th} party if for $i \in [k]$ and all $x_i, x_i' \in \{0,1\}, x_{-i} \in \{0,1\}^{k-1}$ and $\tau \in \mathcal{T}$ as:

  $$\mathbb{P}(\tau | x_i, x_{-i}) \leq e^{\varepsilon_i} \mathbb{P}(\tau | x_i', x_{-i}) \qquad (8)$$

  This condition guarantees no adversary can infer the confidentiality data x_i with high sufficient confidence. If the algorithm is ε_i-differentially confidentiality for all $i \in [k]$, then it is said that the protocol is $\{\varepsilon_i\}$-differentially confidentiality for all parties.

- **Accuracy Measure:** For the i^{th} party, let an accuracy measure $w_i : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ such that $w_i\left(f_i(x), \hat{f}_i(\tau, x_i)\right)$ measures the accuracy when the function to be computed is $f_i(x)$ and the approximation is $\hat{f}_i(\tau, x_i)$. Then, the average accuracy for this i^{th} party is defined as follows:

  $$Acc_{avg}(P, w_i, f_i, \hat{f}_i) = \frac{1}{2^k} \sum_{x \in \{0,1\}^k} \mathbb{E}_{\hat{f}_i, P_{x,\tau}} \left[w_i\left(f_i(x), \hat{f}_i(\tau, x_i)\right) \right] \qquad (9)$$

  Where $\mathbb{E}$ denotes the expectation which is taken over the random transcript τ distribution as P and also any randomness in the decision function $\hat{f}_i$. Similarly, the worst-case accuracy is defined as follows:

  $$Acc_{wc}(P, w_i, f_i, \hat{f}_i) = \min_{x \in \{0,1\}^k} \mathbb{E}_{\hat{f}_i, P_{x,\tau}} \left[w_i\left(f_i(x), \hat{f}_i(\tau, x_i)\right) \right] \qquad (10)$$

- **Latency:** Latency is defined as the number of rounds required to connect all honest parties with high probability after at most $\lceil log\lceil (t+1)/32 \rceil \rceil + 1$ iterations of the while loop.

The following Table 1 shows the performance comparison of PEASMC, SEASMC and PSEASMC compared with existing ASMC in terms of considered metrics such as differential privacy, average accuracy, worst-case accuracy and latency.

Table.1 Comparison of Proposed and Existing TrustedPals Models

	No. of Corrupt Parties	ASMC	PEASMC	SEASMC	PSEASMC
Differential Privacy	25	0.48	0.52	0.56	0.60
	50	0.52	0.56	0.60	0.64
	75	0.55	0.59	0.63	0.67
	100	0.58	0.62	0.65	0.69
	125	0.61	0.65	0.68	0.71
	150	0.64	0.69	0.72	0.73
	No. of Corrupt Parties	**ASMC**	**PEASMC**	**SEASMC**	**PSEASMC**
Average Accuracy (%)	25	71.9	73.2	74.6	75.4
	50	73.5	74.8	75.9	76.7
	75	75.1	76.4	77.5	78.4
	100	76.3	77.1	78.3	79.5
	125	77.2	78.6	79.7	80.9

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

	150	78.4	79.5	80.8	82.0
Worst-case Accuracy (%)	**No. of Corrupt Parties**	**ASMC**	**PEASMC**	**SEASMC**	**PSEASMC**
	25	52.3	53.6	54.9	56.0
	50	53.2	54.5	55.7	57.0
	75	54.8	55.9	57.1	58.2
	100	56.1	57.2	58.5	59.4
	125	57.4	58.8	59.6	60.7
	150	58.9	60.1	61.3	62.5
Latency (Number of Rounds)	**No. of Corrupt Parties**	**ASMC**	**PEASMC**	**SEASMC**	**PSEASMC**
	25	19	16	14	12
	50	21	18	16	14
	75	23	20	18	16
	100	25	22	20	18
	125	27	24	22	20
	150	29	26	24	22

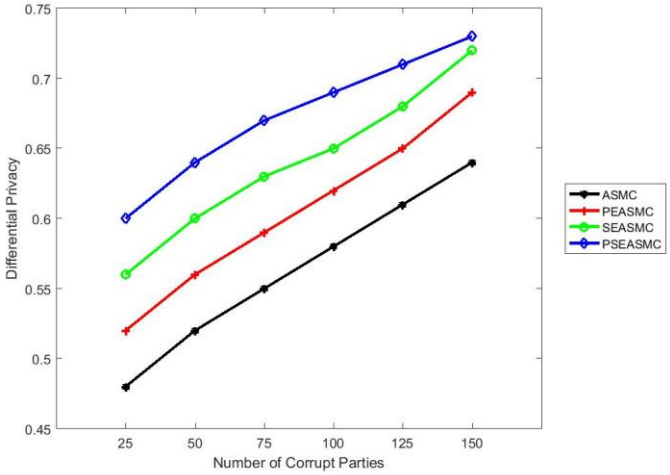

Figure.1 Comparison of Differential Privacy

Figure 1 shows the comparison of proposed models with existing ASMC in terms of differential privacy. For example, consider the number of corrupt users is 150. Then, the differential privacy for PSEASMC is 14.06% higher than ASMC, 5.8% higher than PEASMC and 1.39% higher than SEASMC. From this analysis, it is proved that the proposed PSEASMC model increases the confidentiality for all parties efficiently.

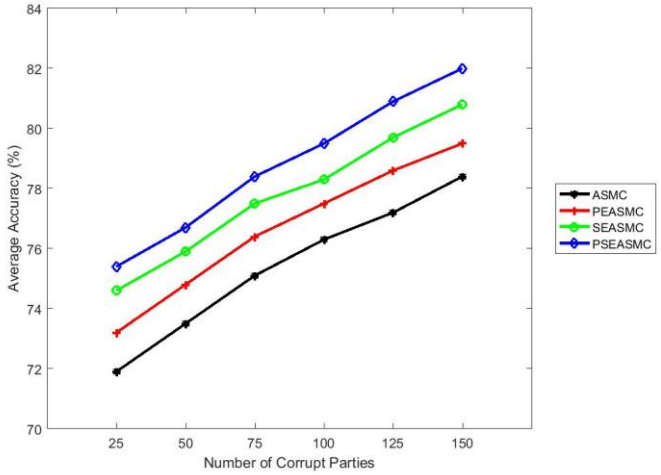

Figure.2 Comparison of Average Accuracy

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

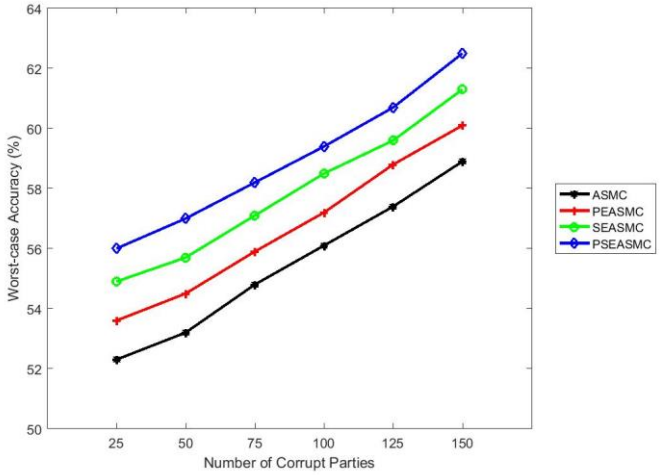

Figure.3 Comparison of Worst-case Accuracy

Figure 2 & 3 illustrates the comparison of average and worst-case accuracy for proposed and existing models, respectively. If the number of corrupt parties is 150, then the average accuracy of proposed PSEASMC model is 4.59% increases than the ASMC model, 3.14% higher than PEASMC and 1.49% higher than SEASMC model. Similarly, the worst-case accuracy of PSEASMC is 6.11% higher than ASMC model, 3.99% higher than PEASMC and 1.96% higher than SEASMC model. Therefore, it is concluded that the proposed PSEASMC model achieves higher accuracy in both average and worst-case for computing the function at each party.

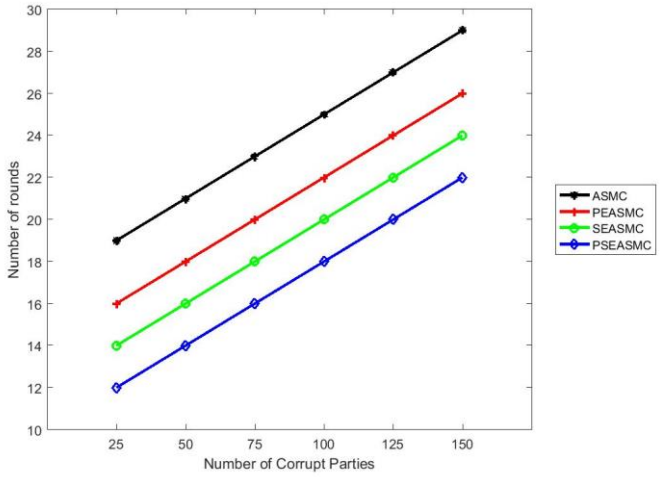

Figure.4 Comparison of Latency

Figure 4 shows the comparison of PSEASMC with existing ASMC in terms of latency. For instance, consider the number of corrupt parties is 150. In this case, the latency (number of rounds) of proposed PSEASMC model is 24.14% minimized than the ASMC model, 15.38% less than PEASMC and 8.33% less than SEASMC model. Thus, it is found that the proposed PSEASMC model significantly reduces the number of rounds to link all honest parties with the highest confidentiality.

V. CONCLUSION

In this article, three adaptive models, namely PEASMC, SEASMC and PSEASMC are proposed within the fault-tolerant computing algorithm to find an acceptable tradeoff between security and performance. The PEASMC and SEASMC models involve the development of a set of metrics to determine the performance and security, respectively. As well, the PSEASMC model involves the development of a tradeoff objective function together with the service performance and security metrics as well as the simultaneous parameter estimation by means of experiments. Accordingly, an acceptable tradeoff is computed by choosing payload size and minimizing the tradeoff objective function efficiently. Additionally, the network configurations and services of failure detector are fine-tuned to balance the security in worse performance. Finally, the experimental results proved that the proposed PSEASMC model achieves higher performance than the other models in terms of maximizing the differential privacy, average and worst-case accuracy and minimizing the latency efficiently.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 06, June-2019

REFERENCES

[1] Fort, M., Freiling, F., Penso, L. D., Benenson, Z., & Kesdogan, D. (2006, September). TrustedPals: Secure multiparty computation implemented with smart cards. In *European Symposium on Research in Computer Security* (pp. 34-48). Springer, Berlin, Heidelberg.
[2] Keersebilck, P. (2004). Smart card technology based on java. In *7th International Conference on Development and Application Systems* (pp. 398-402).
[3] Leavitt, N. (2005). Will proposed standard make mobile phones more secure?. *Computer*, *38*(12), 20-22.
[4] Cortinas, R., Freiling, F. C., Ghajar-Azadanlou, M., Lafuente, A., Larrea, M., Penso, L. D., & Soraluze, I. (2012). Secure failure detection and consensus in trustedpals. *IEEE Transactions on Dependable and Secure Computing*, *9*(4), 610-625.
[5] Atif, M. (2011). Formal modeling and verification of distributed failure detectors. *Faculty of Mathematics and Computer Science, TU/e*, *10*.
[6] Chung, Y. F., Chen, T. L., Chen, C. S., & Chen, T. S. (2012). The study on general secure multi-party computation. *International Journal of Innovative Computing, Information and Control*, *8*(1), 895-910.
[7] Bogdanov, D., Talviste, R., & Willemson, J. (2012). Deploying secure multi-party computation for financial data analysis. In *International Conference on Financial Cryptography and Data Security* (pp. 57-64). Springer, Berlin, Heidelberg.
[8] Thoma, C., Cui, T., & Franchetti, F. (2012). Secure multiparty computation based privacy preserving smart metering system. In *2012 North American power symposium (NAPS)* (pp. 1-6). IEEE.
[9] Abidin, A., Aly, A., Cleemput, S., & Mustafa, M. A. (2016). Towards a local electricity trading market based on secure multiparty computation. COSIC internal report, KU Leuven, imec-COSIC.
[10] Mustafa, M. A., Cleemput, S., Aly, A., & Abidin, A. (2017). An MPC-based protocol for secure and privacy-preserving smart metering. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)* (pp. 1-6). IEEE.
[11] Alexopoulos, N., Kiayias, A., Talviste, R., & Zacharias, T. (2017). MCMix: Anonymous messaging via secure multiparty computation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)* (pp. 1217-1234).