Secure Multi-party based Cloud Computing Framework for Statistical Data Analysis of Encrypted Data

G. P. Harsha Sandaruwan
P. S. Ranaweera
Vladimir A. Oleshchuk
Indika A. Balapuwaduge

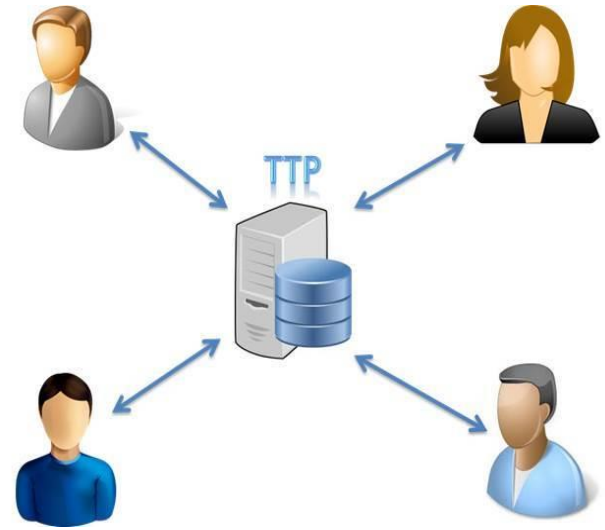
UNIVERSITETET I AGDER

Outline

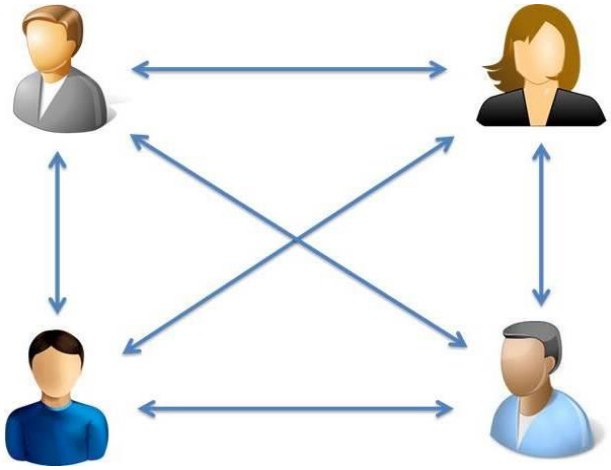
- Introduction
- Problem Statement
- Proposed Solution
- Results
- Conclusions

Introduction – Secure Multi-party Computations (SMC)

- SMC definition :
  - n parties having private inputs $x_1, x_2, \ldots, x_n$
  - Computing a Common Function – $f(x_1, x_2, \ldots, x_n)$
  - Private inputs are secret from each other

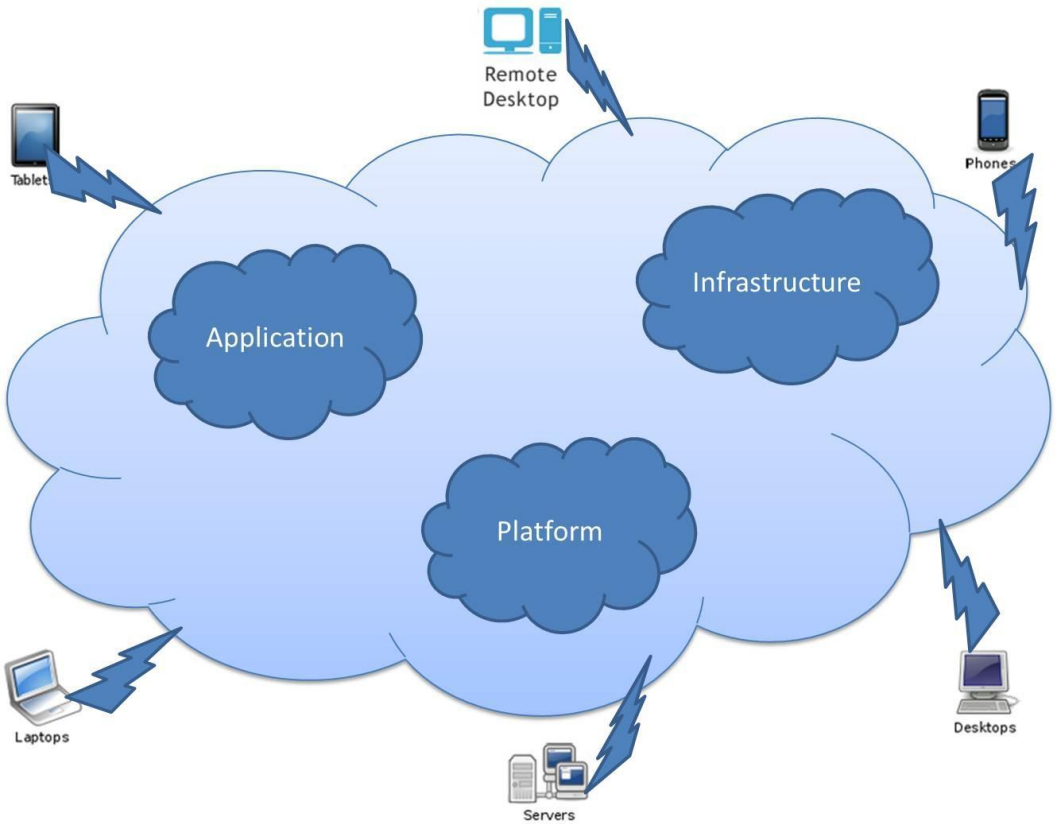


- Efficiency is a main concern



Introduction – Cloud Computing

- Cloud : Collection of Distributed Servers
- Cloud Computing provide On Demand Services for Users through CSPs
- Pros :
  - High Speed Infrastructure
  - High Availability
  - Maintenance free operation
  - Cost Effective
- Cons :
  - Reliability
  - Security….!!!



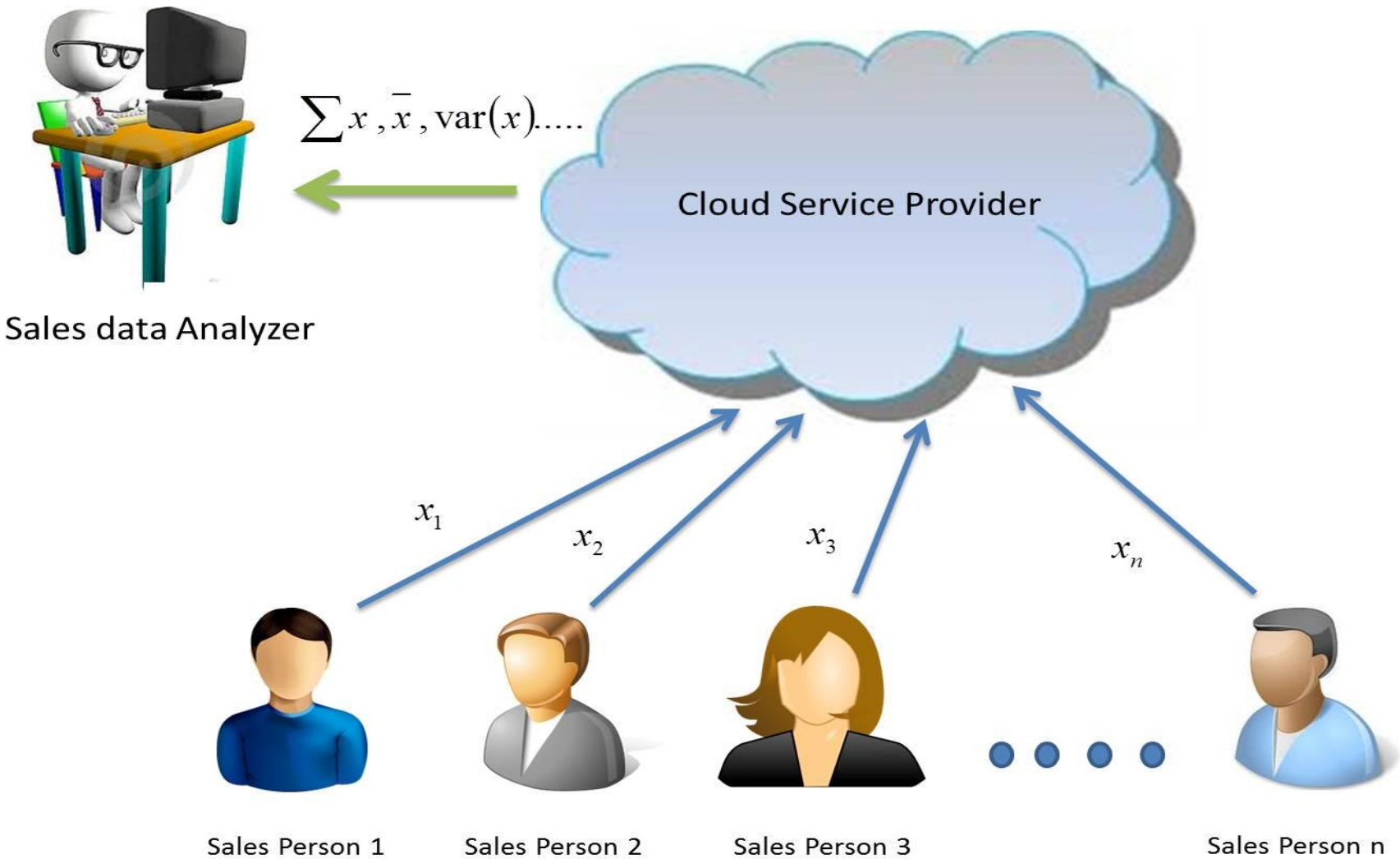
Problem Statement

- Our Intention : Improving the efficiency of SMC
- Our Approach : Deploying Cloud architecture in SMC solutions
- Problem :
  - Security concerns in cloud based solutions
    - Data Privacy
    - Data Security
    - Anonymity

  } CSPs are untrusted entities

- Goal :
  - Propose and Implement a Secure Multi-party based Cloud Computing Framework
    - Secure and Efficient

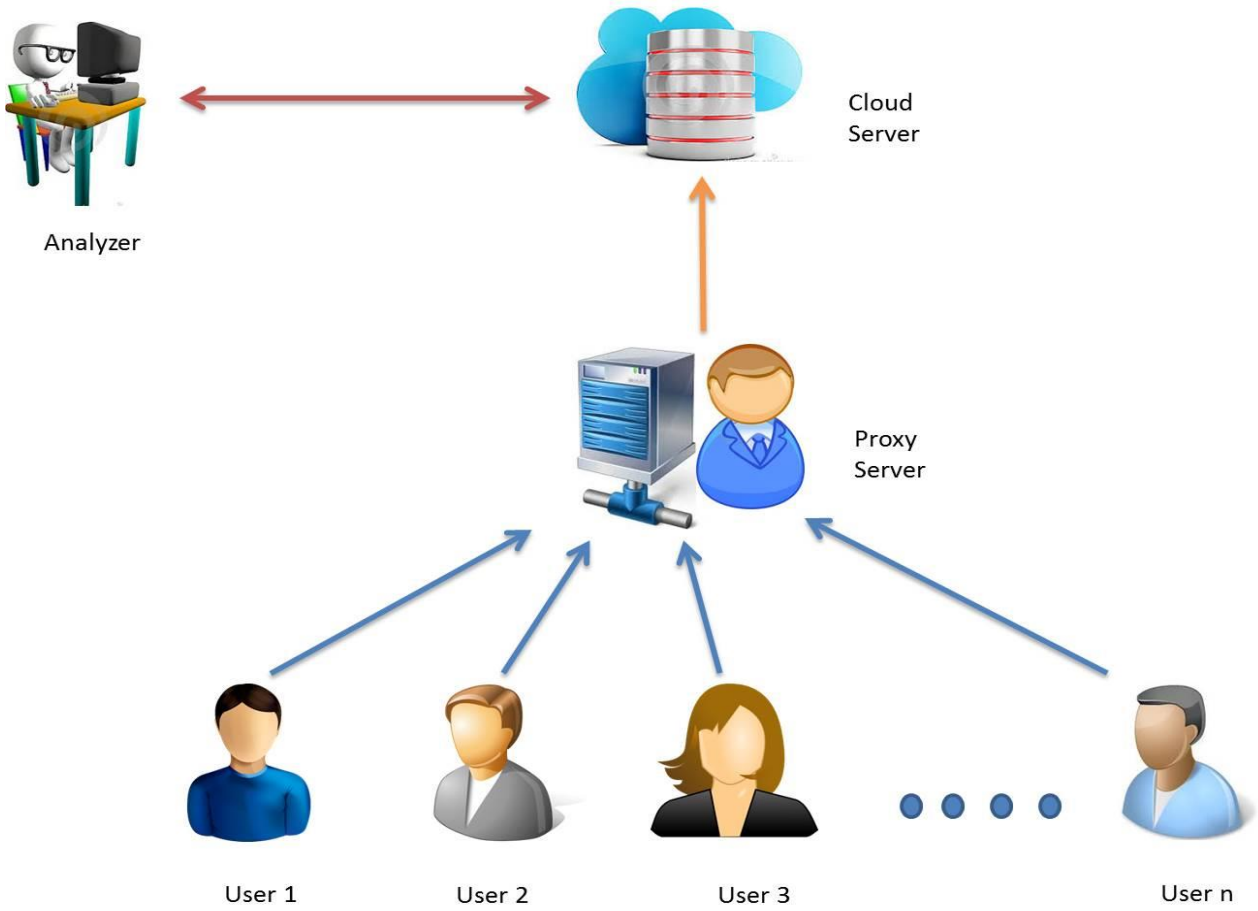
Problem Scenario - Case



Proposed Solution

- Proposed Secure Multi-party based Cloud Computing Framework

- Users :
  - Send encrypted data
- Proxy Server :
  - Anonymize user identities
- Cloud Server :
  - Computations on encrypted data
- Analyzer :
  - Retrieving statistical parameters
    - Mean
    - Variance
    - Standard Deviation
    - Skewness
    - Kurtosis



Proposed Solution – Phases of the Solution

- Key Generation
- Authentication and Key Exchange
- User Data Encryption
- Proxy Server Functionality
- Cloud Sever and Analyzer Functionality

Proposed Solution – Key Generation Phase

- RSA private and public key pairs :
  - Proxy Server
  - Cloud Server
  - Analyzer

  } Mutual Authentication

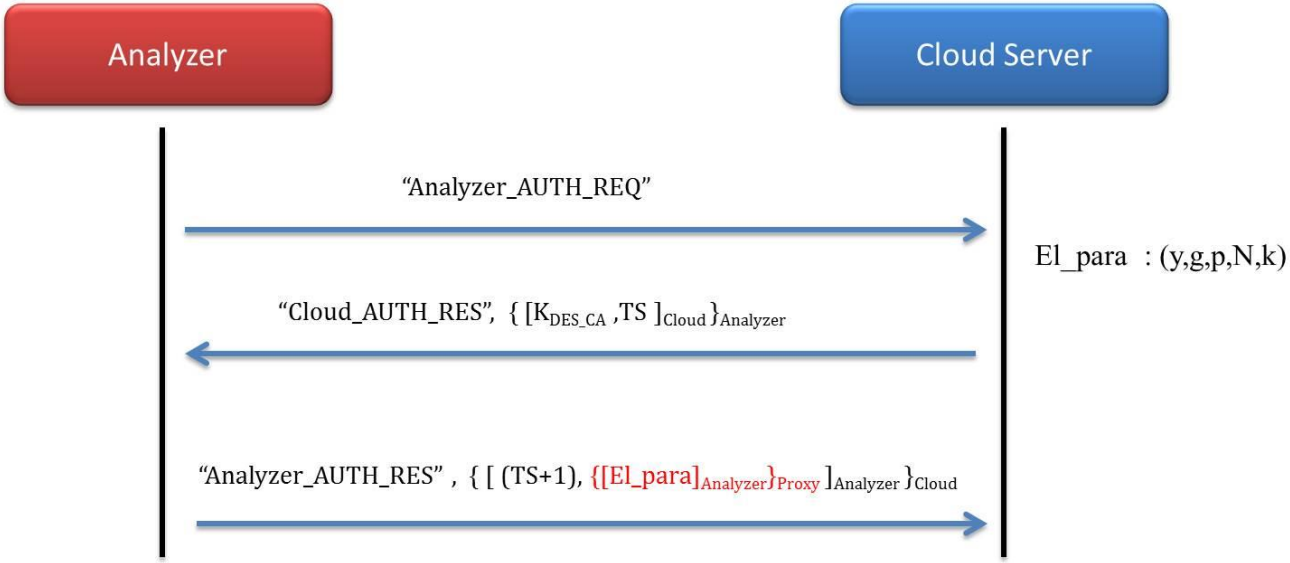
- Triple DES symmetric keys :
  - Proxy and Cloud - K_{DES_PC}
  - Cloud and Analyzer - K_{DES_CA}

  } Encrypting shared messages

- Extended ElGamal keys for analyzer → User data encryption

Proposed Solution – Authentication and Key Exchange Phase

- Analyzer and Cloud Server Mutual Authentication (MA)



- Proxy Server and Cloud Server MA – Similar approach
  - El_para is retrieved from Proxy Server
  - K_{DES_PC} is shared
- User authentication with proxy – user name & password

Proposed Solution - User Data Encryption

- Data Encryption : Extended ElGamal Encryption Scheme (EEES)
  - Public key – p, N, g, y
  - Encryption Equation - $E(M) = (A, B) = \left(g^k \bmod p,\ y^k \left((M + r \times p) \bmod \mathrm{N} \right) \bmod p \right)$

M - message

(A, B) - ciphertext

g - generator

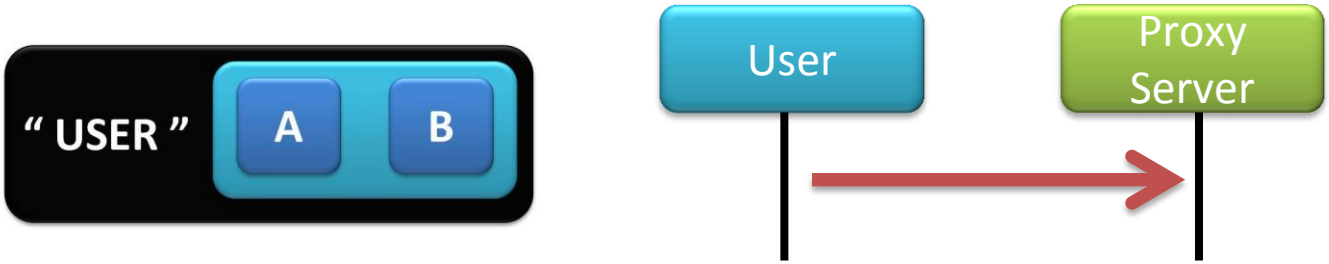
k - positive number common for all users

r - user selected positive number

$y = g^x \bmod p$; x is the private key of EEES

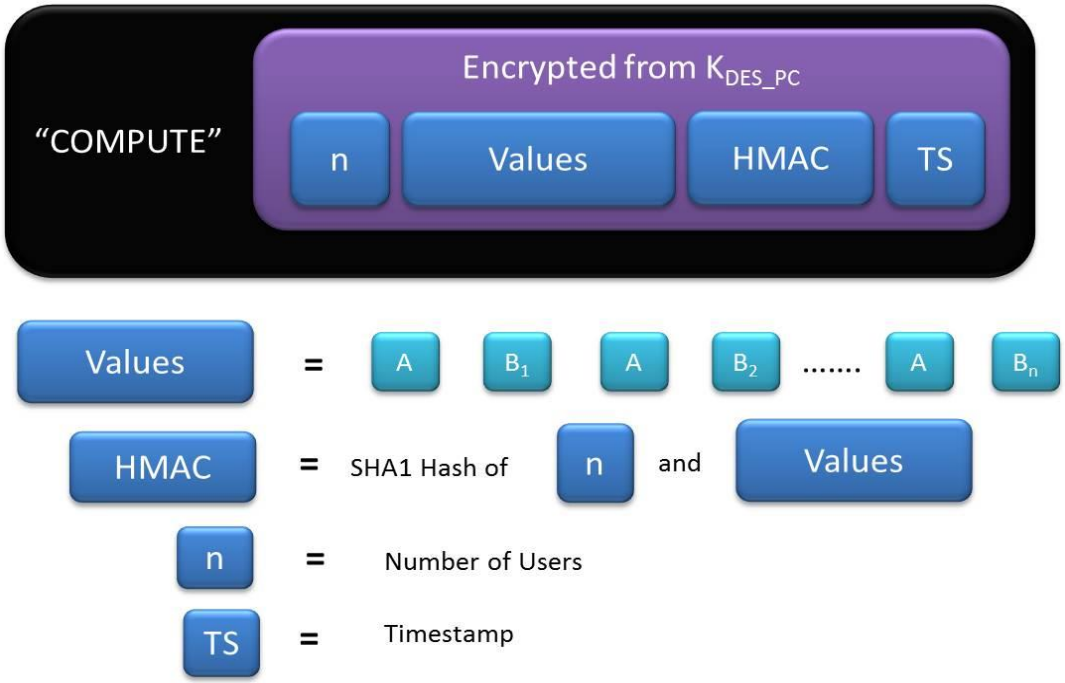
$\mathrm{N} = p \times q$; p, q are large secure prime numbers

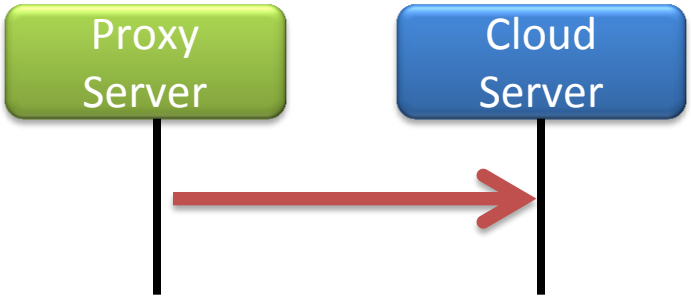
- Data frame :



Proposed Solution – Proxy Server Functionality

- Waits till all the user encrypted data is received
- Data Frame :



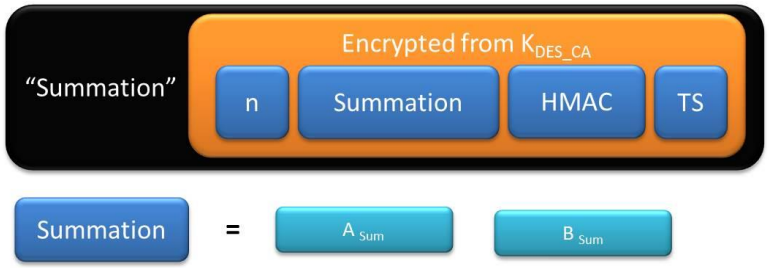


Proposed Solution – Cloud Server and Analyzer Functionality

- Encrypted summation (A_{Sum} , B_{Sum}) computing :

$$A_{Sum} = A$$

$$B_{Sum} = B_1 + B_2 + \ldots + B_n$$

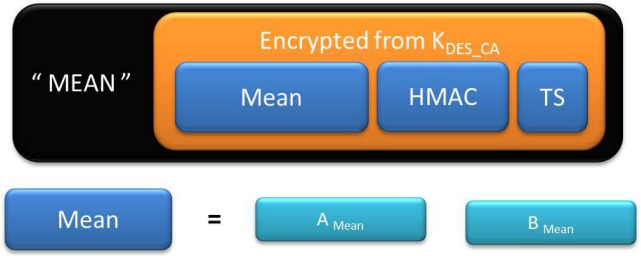


- Analyzer - Decrypts (A_{Sum} , B_{Sum}) and compute Mean :

$$M_1 = B_{Sum} \times \left(A_{Sum}^{x}\right)^{-1} \bmod p$$

$$Mean = \frac{M_1}{n}$$

- Analyzer – encrypting Mean and sending to cloud



Proposed Solution – Cloud Server and Analyzer Functionality

- Calculating Encrypted Mean Deviation :

$$A_{Deviation,i} = A_{Mean} \text{ and } B_{Deviation,i} = B_i - B_{Mean}$$

- Calculating j^{th} power of Deviation :

$$\text{Let}; D_i = \left(A_{Deviation,i}, B_{Deviation,i}\right)$$

$$D_i^j = \left(A_{Deviation,i}^j, B_{Deviation,i}^j\right)$$

- Calculating Encrypted V_1, S_1 and K_1 :

$$V_1 = \sum_{i=1}^{n} D_i^2 \quad , \quad S_1 = \sum_{i=1}^{n} D_i^3 \quad , \quad K_1 = \sum_{i=1}^{n} D_i^4$$

- Sending Encrypted statistical parameters to Analyzer with TS and HMAC encrypted with K_{DES_CA}

- Decrypt V_1,S_1,K_1 using private key x

$$\text{Variance} = \frac{V}{n} \qquad \text{Standard Deviation, } \sigma = \sqrt{\frac{V}{n}}$$

$$\text{Skewness} = \frac{S}{\sigma^3} \qquad \text{Kurtosis} = \frac{K}{\sigma^4}$$

Results

- Performance Parameters :
  - EEES Encryption Time (T_e)
  - Entity Process Time (T_{EP})
  - Total Process Time (T_{TP})
  - Transmission Delay (T_{TD})
  - Total Time (T_T)

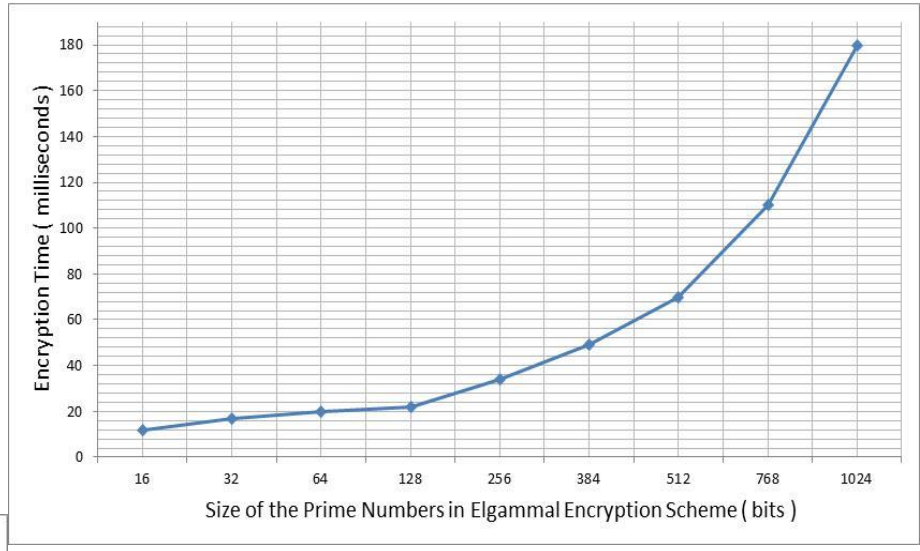
$$T_{TP} = T_{EP} + T_E \times n$$

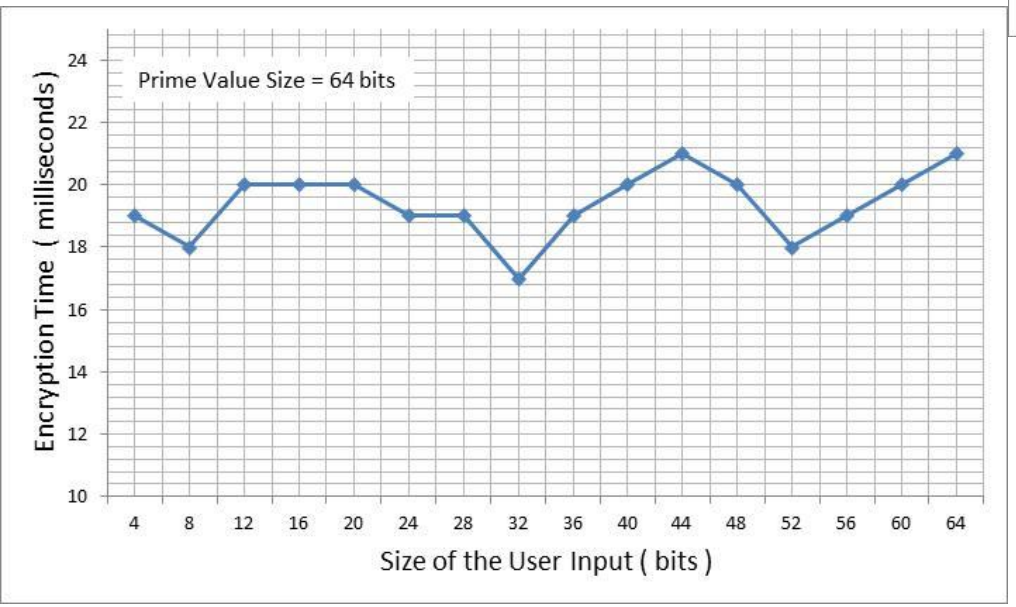
$$T_T = T_{TP} + T_{TD}$$

- Test Cases :
  - Variation of T_e with prime size and input message size (l)
  - Variation of T_{EP} and T_{TP} with n and l

Results – Variation of T_e

- Exponential variation
- Optimized prime size = 64 bits

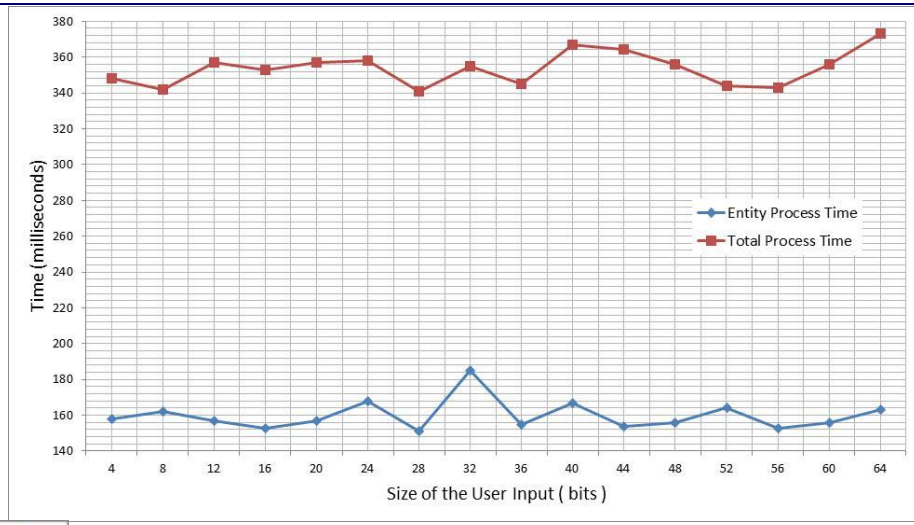


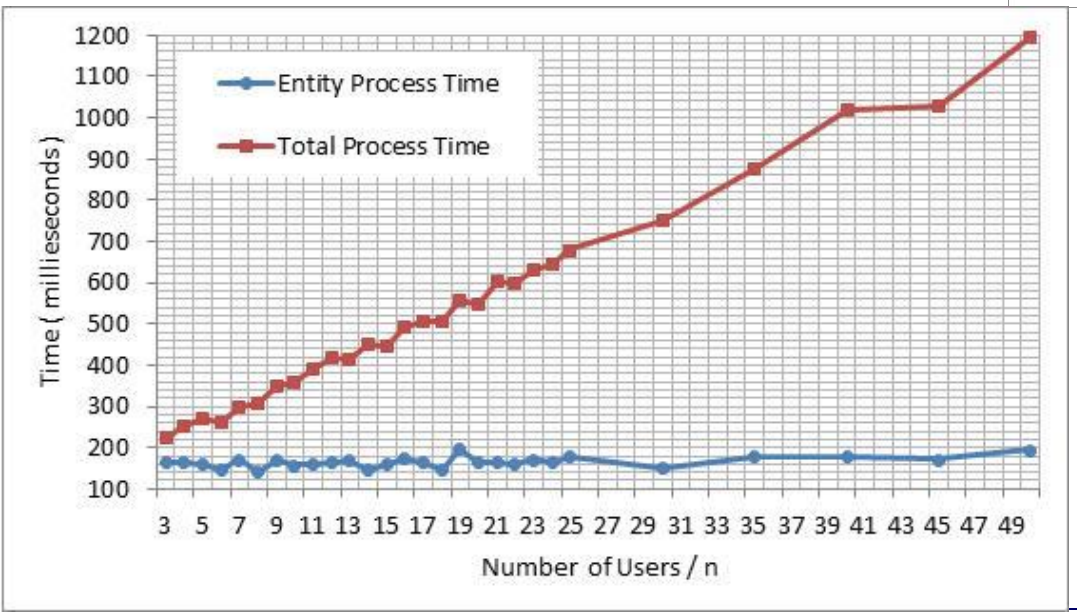


- T_e is independent of l
- Average T_e = 20ms

Results – Variation of T_{EP} and T_{TP}

- T_{EP} and T_{TP} are independent of l
- T_{TP} is elevated approximately about 200ms – Total T_e for 10 users





- T_{EP} is independent of n
- T_{TP} varies linearly with n

$$\begin{pmatrix} Average \\ T_{TP} \end{pmatrix} = \begin{pmatrix} Average \\ T_{EP} \end{pmatrix} + \left(\begin{pmatrix} Average \\ T_e \end{pmatrix} \times n \right)$$

$$\begin{pmatrix} Average \\ T_{TP} \end{pmatrix} = (162.53 + 20 \times n)\, ms$$

Conclusions

- Security Analysis :
  - Data privacy guaranteed – framework operates on encrypted data
  - Identity Anonymization – Proxy Server
  - Data Security – TS, HMAC, Symmetric key encryption, MA
- Numerical Results :
  - Provides evidence for the efficiency of the implemented framework

$$\begin{pmatrix} Average \\ T_{TP} \end{pmatrix} = (162.53 + 20 \times n)\, ms$$

  - Due to parallel processing of computing parties, average T_{TP} will be less…

- *Considering security analysis and numerical results, we can conclude that cloud environments can be successfully deployed to improve the efficiency of multi-party computations while ensuring the security requirements of user parties*

Thank you!

#ICIN2016

UNIVERSITETET I AGDER