Secure Function Extensions to Additively Homomorphic Cryptosystems

Mounika Pratapa and Aleksander Essex

Selected Areas in Cryptography - 2023

August 17, 2023

mpratapa@uwo.ca
aessex@uwo.ca

Agenda

- Methodology to extend the functionality of additively homomorphic encryption schemes by modifying secret key generation
- Steps for modified key generation
- Potential applications and results

Four Questions: What? Why? How? So what?

Secure Computation

- Increase in the availability of personal information
- Challenge: Make the best possible use of available data without giving away access to it
- Data Encryption- popular and secure
- Can we perform computations on this encrypted data, without decrypting it?

Secure Function Evaluation

In a two party setting:

- Alice and Bob with inputs x, y respectively
- They want to jointly evaluate a function $f(x, y)$, without sharing their inputs
- Upon SFE, Alice will learn $f(x, y)$ and nothing else. Bob learns nothing

Applications: Privacy-preserving machine learning, private information retrieval, similarity search in private databases such as genotype and other medical data, online voting, auctions and private credit checking.

Homomorphic Encryption

- Homomorphic encryption between two messages m_1, m_2:

$$Enc(m_1 \star m_2) = Enc(m_1) \diamond Enc(m_2)$$

- Decryption results match with operations on a plain-text message

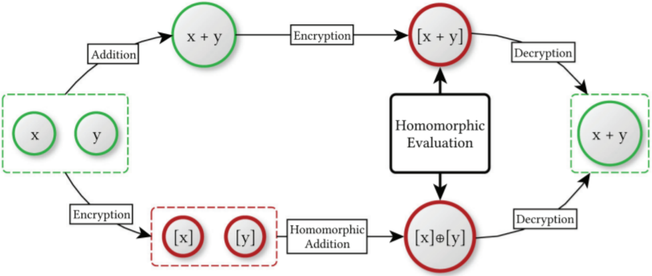

Figure: Additive Homomorphism Wood et al. [2020]

Categories

- Homomorphic Encryption
  - Partially HE
    - One type of gate
    - Add or Multiply
  - Somewhat HE
    - Two types of gates
    - Subset of circuits
  - Levelled HE
    - Arbitrary circuits
    - Multiple gates
    - Bounded depth
  - Fully HE
    - Arbitrary circuits
    - Multiple gates
    - Unbounded depth

Why PHE?

- Additive HE plays an important role in secure computations
- Examples: Medical applications, Internet-voting (Switzerland)
- Reasons:
  - Clear-cut parameterizations
  - More mature(well understood) hardness assumptions
  - Faster execution
  - Reduced communication overhead compared to Garbled circuits
- Can we do more than just addition?

Quadratic Residue Function: QR(x, p)

- Legendre symbol $L : \mathbb{Z} \times \mathbb{Z} \mapsto \{-1, 0, 1\}$:

$$\left(\frac{x}{p}\right) \equiv \begin{cases} 1 & \text{if x is quadratic residue mod } p \\ -1 & \text{if x is quadratic non-residue mod } p \\ 0 & \text{if } x \equiv 0 \text{ mod } p. \end{cases}$$

Quadratic Residue Function: $\mathrm{QR}(x, p)$

- Legendre symbol $L : \mathbb{Z} \times \mathbb{Z} \mapsto \{-1, 0, 1\}$:

$$\left(\frac{x}{p}\right) \equiv \begin{cases} 1 & \text{if x is quadratic residue mod } p \\ -1 & \text{if x is quadratic non-residue mod } p \\ 0 & \text{if } x \equiv 0 \bmod p. \end{cases}$$

- Let $\mathrm{QR} : \mathbb{Z} \times \mathbb{Z} \to \{0, 1\}$ be a function testing the quadratic residuosity of an integer $x \in \mathbb{Z}_p$, defined as

$$\mathrm{QR}(x, p) = \begin{cases} 1 & \text{if } x \text{ is a quadratic residue modulo } p. \\ 0 & \text{otherwise.} \end{cases}$$

Quadratic Residue Function: $\mathrm{QR}(x, p)$

- Legendre symbol $L : \mathbb{Z} \times \mathbb{Z} \mapsto \{-1, 0, 1\}$:

$$\left(\frac{x}{p}\right) \equiv \begin{cases} 1 & \text{if x is quadratic residue mod } p \\ -1 & \text{if x is quadratic non-residue mod } p \\ 0 & \text{if } x \equiv 0 \bmod p. \end{cases}$$

- Let $\mathrm{QR} : \mathbb{Z} \times \mathbb{Z} \to \{0, 1\}$ be a function testing the quadratic residuosity of an integer $x \in \mathbb{Z}_p$, defined as

$$\mathrm{QR}(x, p) = \begin{cases} 1 & \text{if } x \text{ is a quadratic residue modulo } p. \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathrm{QR}(x, p) = \frac{\left(\frac{x}{p}\right) + 1}{2}$$

Quadratic Residue Symbol Sequences

For $p = 277$, the Residue symbols for first 10 positive integers:

x	1	2	3	4	5	6	7	8	9	10
$QR_{277}(x)$	1	1	0	1	1	0	0	1	0	1

Quadratic Residue Symbol Sequences

For $p = 277$, the Residue symbols for first 10 positive integers:

x	1	2	3	4	5	6	7	8	9	10
$QR_{277}(x)$	1	1	0	1	1	0	0	1	0	1

For $p = 277$ and an offset value of 178, the Legendre symbols of 10 elements from 178 are:

x	1	2	3	4	5	6	7	8	9	10
$QR_{277}(178 + x)$	0	0	0	0	0	0	1	1	1	1

Quadratic Residue Symbol Sequences

For $p = 277$, the Residue symbols for first 10 positive integers:

x	1	2	3	4	5	6	7	8	9	10
$QR_{277}(x)$	1	1	0	1	1	0	0	1	0	1

For $p = 277$ and an offset value of 178, the Legendre symbols of 10 elements from 178 are:

x	1	2	3	4	5	6	7	8	9	10
$QR_{277}(178 + x)$	0	0	0	0	0	0	1	1	1	1

Observe that the sequence is a consecutive occurrence of symbols- limited in scope

Linear Embeddings in Residue Symbol Sequences

Given $f(\cdot)$ and an integer sequence of the form $[\alpha x + \beta \mid 0 \leq x < t, \text{ and } \alpha, \beta > 0]$, our approach involves three components:

1. An efficient algorithm for finding a prime p for which

$$\mathrm{QR}(\alpha x + \beta, p) = f(x).$$

2. An additively homomorphic public-key cryptosystem embedding the required quadratic residue symbol sequence into the plaintext space, i.e., $\mathcal{M} \subset \mathbb{Z}_p$.
3. A public homomorphic operation that can blind the encryption of $\alpha x + \beta$ while preserving its quadratic residue symbol modulo p (and hence the output of the function $f(x)$).

Approach to secure computation

- $\text{CS} = \{\text{Gen}, \text{Enc}, \text{Dec}\}$
- Homomorphisms:

$$\text{Enc}(x_1) \cdot \text{Enc}(x_2) = \text{Enc}(x_1 + x_2 \bmod p)$$

$$\text{Enc}(x_1)^{x_2} = \text{Enc}(x_1 x_2 \bmod p).$$

- A mapping function $h : \mathbb{Z} \rightarrow \mathbb{Z}_p$, $h(x) = (\alpha x + \beta) \bmod p$

Approach to secure computation

- $\mathsf{CS} = \{\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}\}$
- Homomorphisms:

$$\mathsf{Enc}(x_1) \cdot \mathsf{Enc}(x_2) = \mathsf{Enc}(x_1 + x_2 \bmod p)$$

$$\mathsf{Enc}(x_1)^{x_2} = \mathsf{Enc}(x_1 x_2 \bmod p).$$

- A mapping function $h : \mathbb{Z} \to \mathbb{Z}_p$, $h(x) = (\alpha x + \beta) \bmod p$
- Given $\mathsf{Enc}(x)$ for $0 \leq x < t$, and an $\alpha, \beta > 0$, compute:

$$\mathsf{Enc}(h(x)) = \mathsf{Enc}(x)^{\alpha} \cdot \mathsf{Enc}(\beta) = \mathsf{Enc}(\alpha x + \beta \bmod (p)).$$

Approach to secure computation

- $CS = \{\text{Gen}, \text{Enc}, \text{Dec}\}$
- Homomorphisms:

$$\text{Enc}(x_1) \cdot \text{Enc}(x_2) = \text{Enc}(x_1 + x_2 \bmod p)$$

$$\text{Enc}(x_1)^{x_2} = \text{Enc}(x_1 x_2 \bmod p).$$

- A mapping function $h : \mathbb{Z} \to \mathbb{Z}_p$, $h(x) = (\alpha x + \beta) \bmod p$
- Given $\text{Enc}(x)$ for $0 \leq x < t$, and an $\alpha, \beta > 0$, compute:

$$\text{Enc}(h(x)) = \text{Enc}(x)^{\alpha} \cdot \text{Enc}(\beta) = \text{Enc}(\alpha x + \beta \bmod (p)).$$

- Using QR Function:

$$\text{QR}(\text{Dec}(\text{Enc}(\alpha(x) + \beta)), p) = \text{QR}(h(x), p) = f(x).$$

Theorem (1)

Consider a list of k distinct primes $\{a_1, \ldots, a_k\}$ and a list of residue symbols $\{\ell_1, \ldots, \ell_k\}$ where $\ell_i \in \{-1, 1\}$. For all $1 \leq i \leq k$, there exists a prime p such that

$$\left(\frac{p}{a_i}\right) = \ell_i.$$

Theorem (1)

Consider a list of k distinct primes $\{a_1, \ldots, a_k\}$ and a list of residue symbols $\{\ell_1, \ldots, \ell_k\}$ where $\ell_i \in \{-1, 1\}$. For all $1 \leq i \leq k$, there exists a prime p such that

$$\left(\frac{p}{a_i}\right) = \ell_i.$$

Theorem (2)

For all $t \in \mathbb{Z}^+$ and all functions $f : \mathbb{Z}_t \to \{0, 1\}$ $\exists$ a prime p and two integers $0 < \alpha, \beta < p$ such that for all $0 \leq x < t$ $\dfrac{\left(\frac{\alpha x + \beta}{p}\right) + 1}{2} = f(x)$

Components

$\mathrm{CS} = \{\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec}, \mathrm{Add}, \mathrm{Smul}, \mathrm{Eval}\}$

- $\mathrm{Gen}(1^\rho, \alpha, \beta, f)$: Secret keys $\mathcal{SK} = \{p, q\}$ and public key $\mathcal{PK} = \{n\}$ where $n = p^2 q$
- $\mathrm{Enc}(\mathcal{PK}, m)$: $c = [\![m]\!]$
- $\mathrm{Dec}(\mathcal{SK}, c)$: m
- $\mathrm{Add}(c_1, c_2)$: $c' = [\![(m_1 + m_2) \bmod p]\!]$
- $\mathrm{Smul}(s, c)$: $c' = [\![(m_1 m_2) \bmod p]\!]$
- $\mathrm{Eval}(\mathcal{PK}, \alpha, \beta, c)$:
  - Choose $r_c \leftarrow [1, 2^\lambda]$

  $\mathrm{Smul}(r_c^2, \mathrm{Add}(\mathrm{Smul}([\![m]\!], \alpha), \mathrm{Enc}(\beta)) = [\![r_c^2 \cdot (\alpha m + \beta) \bmod p]\!]$

Finding p

- $S = \{s_m \mid s_m = \alpha m + \beta, 0 \leq m < t\}$ - an odd sequence for some $\alpha, \beta \in \mathbb{Z}^+$

Finding p

- $S = \{s_m \mid s_m = \alpha m + \beta, 0 \leq m < t\}$ - an odd sequence for some $\alpha, \beta \in \mathbb{Z}^+$
- Factorize $s_m \in S$ to $s_{m,0}^{(e_{m,0})}, \ldots, s_{m,\rho_m}^{(e_{m,\rho_m})}$ and form the following set of equations:

$$\left(\frac{s_m}{p}\right) = \left(\frac{s_{m,0}^{(e_{m,0})} \cdot \ldots \cdot s_{m,\rho_m}^{(e_{m,\rho_m})}}{p}\right) = \left(\frac{s_{m,0}}{p}\right) \cdot \ldots \cdot \left(\frac{s_{m,\rho_m}}{p}\right) = 1 - 2 \cdot f(m)$$

Here $e_{m,j}$ is an odd power

Finding p

- $S = \{s_m \mid s_m = \alpha m + \beta, 0 \leq m < t\}$ - an odd sequence for some $\alpha, \beta \in \mathbb{Z}^+$
- Factorize $s_m \in S$ to $s_{m,0}^{(e_{m,0})}, \ldots, s_{m,\rho_m}^{(e_{m,\rho_m})}$ and form the following set of equations:

$$\left(\frac{s_m}{p}\right) = \left(\frac{s_{m,0}^{(e_{m,0})} \cdot \ldots \cdot s_{m,\rho_m}^{(e_{m,\rho_m})}}{p}\right) = \left(\frac{s_{m,0}}{p}\right) \cdot \ldots \cdot \left(\frac{s_{m,\rho_m}}{p}\right) = 1 - 2 \cdot f(m)$$

Here $e_{m,j}$ is an odd power

$$\mathrm{QR}(s_m, p) = \mathrm{QR}(s_{m,0}, p) + \ldots + \mathrm{QR}(s_{m,\rho_m}, p) \equiv f(m) \bmod 2.$$

Finding p

- $A = \{a_0, \ldots, a_{u-1}\} \rightarrow$ set of u unique prime factors from the complete set of factors of each element

Finding p

- $A = \{a_0, \dots, a_{u-1}\} \rightarrow$ set of u unique prime factors from the complete set of factors of each element
- Define a function:

$$d(a_j, s_m) = \begin{cases} 1 & \text{if } a_j \mid s_m \\ 0 & \text{otherwise.} \end{cases}$$

Finding p

- $A = \{a_0, \ldots, a_{u-1}\} \rightarrow$ set of u unique prime factors from the complete set of factors of each element
- Define a function:

$$d(a_j, s_m) = \begin{cases} 1 & \text{if } a_j \mid s_m \\ 0 & \text{otherwise.} \end{cases}$$

- Construct a matrix based on the factor list in each element

Finding p

$$M = \begin{matrix} \\ s_0 \\ s_1 \\ \vdots \\ s_{t-1} \end{matrix} \begin{matrix} a_0 & a_1 & \dots & a_{u-1} & \\ \left(\begin{matrix} d(a_0, s_0) \\ d(a_0, s_1) \\ \vdots \\ d(a_0, s_{t-1}) \end{matrix}\right. & \begin{matrix} d(a_1, s_0) \\ d(a_1, s_1) \\ \vdots \\ d(a_1, s_{t-1}) \end{matrix} & \begin{matrix} \dots \\ \dots \\ \\ \dots \end{matrix} & \begin{matrix} d(a_{u-1}, s_0) \\ d(a_{u-1}, s_1) \\ \vdots \\ d(a_{u-1}, s_{t-1}) \end{matrix} & \left|\begin{matrix} f(0) \\ f(1) \\ \vdots \\ f(t-1) \end{matrix}\right) \end{matrix}$$

- Compute $M' \leftarrow \text{RREF}(M)$
- If the system of equations is consistent and exactly determined, each $a_j \in A$ implies a residue value $\ell_j \in \{0, 1\}$
- Satisfies $\text{QR}(s_m) = f(m)$ for $0 \leq m < t$.

Finding p

- For each $a_j \in A$ and each residue value $\ell_j \in \{0, 1\}$, select $b_j \leftarrow [0, a_j)$ such that $\mathrm{QR}(b_j, a_j) = \ell_j$.
- For each pair a_j, b_j:

$$\begin{aligned} p' &\equiv b_0 \bmod a_0 \\ p' &\equiv b_1 \bmod a_1 \\ &\vdots \\ p' &\equiv b_{u-1} \bmod a_{u-1}. \end{aligned}$$

- Compute $p \leftarrow k\left(\prod_{j=0}^{u-1} a_j\right) + p'$ for $k \xleftarrow{R} [k_{min}, k_{max}]$ such that $|p| = \lambda$.
- If $p \equiv 1 \bmod 4$ and p is prime, output p, else find new b_j

Okamoto-Uchiyama Cryptosystem

Encryption:

- $g \in \mathbb{Z}_n^* \mid g^{p-1} \not\equiv 1 \pmod{p^2}$
- $h \equiv g^n \pmod{n}$
- $c \leftarrow g^m h^r \pmod{n} \mid n = p^2 q$

Decryption:

- $a = \dfrac{(c^{p-1} \pmod{p^2}) - 1}{p}$
- $b = \dfrac{(g^{p-1} \pmod{p^2}) - 1}{p}$
- $m = ab^{-1} \pmod{p}$

(*Eval*) Correctness

$$
\begin{aligned}
c' = \mathrm{Eval}(\mathcal{PK}, \alpha, \beta, c) &= (c^{\alpha} \cdot [\![\beta]\!])^{r_c^2} \bmod n \\
&= ([\![m]\!]^{\alpha} \cdot [\![\beta]\!])^{r_c^2} \\
&= [\![(\alpha m + \beta) \cdot r_c^2]\!].
\end{aligned}
$$

$$
\mathrm{Dec}(c') = \alpha m + \beta \cdot r_c^2
$$

Apply QR-function

$$
\begin{aligned}
\mathrm{QR}((\alpha m + \beta) \cdot r_c^2, p) &= \mathrm{QR}(\alpha m + \beta, p) \cdot \mathrm{QR}(r_c^2, p) \\
&= f(m) \cdot 1 \\
&= f(m).
\end{aligned}
$$

Semantic Security

Two decision problems:

- p-th residue decisional problem (PRDP): Given $a \in \mathbb{Z}_n^*$ and $n = p^2q$ for unknown p, q, deciding if $\exists$ a b where $a \equiv b^p \mod n$
- Quadratic residuosity mod p decisional problem (QRDP): Given $\text{Enc}(m)$ and an unknown p, computing $\text{QR}(m, p)$

QRDP is reducible to PRDP $\implies$ modified CS is semantically secure

Public: $\mathcal{PK}, \{\alpha, \beta\}, f : \mathbb{Z}_t \mapsto \{0,1\}$

Alice		*Bob*
$X \leftarrow \{x_1, \ldots, x_a\}$		$Y = \{y_1, \ldots, y_b\}$
$\mathcal{SK} = \{p, q\}$		
	$\llbracket X \rrbracket$ $\longrightarrow$	
		$\llbracket m \rrbracket \leftarrow \pi_{sub}(X, Y)$
		$c' \leftarrow \text{Eval}(\mathcal{PK}, \alpha, \beta, \llbracket m \rrbracket)$
	c' $\longleftarrow$	
$m' \leftarrow \text{Dec}(\mathcal{SK}.c')$		
$m' = (\alpha \cdot m + \beta) \cdot r_c^2$		
$\text{QR}(m', p) = f(m)$		

Protocol Security

This protocol is secure under Honest-But-Curious model- proved by taking the security of Alice and Bob separately

Protocol Security

This protocol is secure under Honest-But-Curious model- proved by taking the security of Alice and Bob separately:

- Alice's privacy is dependent on the cryptosystem itself as Alice shares only encrypted output. Since the *CS* is semantically secure, Alice's data is secure

Protocol Security

This protocol is secure under Honest-But-Curious model- proved by taking the security of Alice and Bob separately:

- Alice's privacy is dependent on the cryptosystem itself as Alice shares only encrypted output. Since the CS is semantically secure, Alice's data is secure
- The privacy of Bob's output, relies on the hider $r_c \xleftarrow{R} \mathbb{Z}_k$ where k is the bit-size of the prime p. Bob's decrypted output has the same distribution as that of Quadratic Residues and Non-Residues, making the security hardness equivalent to Quadratic Residuosity Problem

Key Generation Implementation

Function size (domain cardinality t)	512	256	128	50
Gaussian Elimination	0.236	0.078	0.015	0.004
Test for consistency	0.016	0.009	0.002	0.001
Finding the right b_x	87.30	21.00	3.900	0.560
CRT	25.24	3.6	0.142	0.062

Table: Run time for various steps in the the key generation in seconds

Results Analysis

Performance Indicator	Abspoel et al. [2019]	Yu [2011]	Essex [2019]	Our Protocol
Domain cardinality (t)	623	$\Omega(log(p))$	26	512
Residue symbol sequence type	$\{1\}^t$	$\{1\}^t$	$[0]^t \mid\mid [1]^t$	$\{0,1\}^t$
Secure function evaluation type	Specific (sign functions)	Specific (sign functions)	Specific (thresholds)	General (Boolean)

Table: Comparison between SFE protocols that rely on the runs of quadratic residues.

Practical use

- Private record linkage, information retrieval and machine learning inference
- Display of intermediate computations leads to potential database reconstruction attacks
- Hiding intermediate computations requires increase in communication rounds or reliance on some trusted third parties
- Our approach achieves single round communication while displaying only the end result

Summary

- Explored the properties of quadratic residue sequences and combined it with public key cryptography to expand the functionality of existing additive homomorphic encryption schemes
- Implemented a modified key-generation algorithm that produces primes based on arbitrary residue symbol sequences
- Designed protocol for SFE domains which is secure in honest-but curious setting
- Future work could optimize methods to find such α and β to generate primes with smaller bit-size

Thank You!

Questions?

References

M. Abspoel, N. J. Bouman, B. Schoenmakers, and N. de Vreede. Fast secure comparison for medium-sized integers and its application in binarized neural networks. In *Topics in Cryptology–CT-RSA 2019: The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4–8, 2019, Proceedings*, pages 453–472. Springer, 2019.

A. Essex. Secure approximate string matching for privacy-preserving record linkage. *IEEE Transactions on Information Forensics and Security*, 14 (10):2623–2632, 2019.

A. Wood, K. Najarian, and D. Kahrobaei. Homomorphic encryption for machine learning in medicine and bioinformatics. *ACM Computing Surveys (CSUR)*, 53(4):1–35, 2020.

C.-H. Yu. Sign modules in secure arithmetic circuits. *Cryptology ePrint Archive*, 2011.