IPA problem solution

Łukasz Chmielewski

February 15, 2006

1 Introduction

In this article I will briefly present solution of problem presented on IPA fall days.

2 Problem definition

- Two participants: Client and Server.
- Important is privacy of Client, privacy of Server is not important.
- In this solution is used semantically secure homomorphic encryption scheme, that provides following operations:
  1. Given the encryptions of a and b, $E_{pk}(a)$ and $E_{pk}(b)$, we can efficiently compute the encryption of $a+b$, denoted $E_{pk}(a+b) := E_{pk}(a) +_h E_{pk}(b)$
  2. Given a constant c and the encryption of a, $E_{pk}(a)$, we can efficiently compute the encryption of $c \cdot a$, denoted $E_{pk}(ca) := c \cdot_h E_{pk}(a)$

  Example of such a cryptosystem is Paillier cryptosystem.
- Inputs (P is some domain - group):
  - Input of Client: $x \in P$;
  - Input of Server: $a = \texttt{array}\ [1..n]\ \texttt{of}\ P$;
- Outputs:
  - Output of Client: at least (he may know more about Server's data) knowledge if $x \in a$;
  - Output of Server: nothing;

3 Solution

In sake of simplicity I assume that $\sqrt{n}$ is integer value.

1. Server sorts his array.
2. Server sends to Client $\left\{\forall_{i\in[1..\sqrt{n}]}\ (a[(i-1)\cdot\sqrt{n}+1])\right\}$.

$$\overbrace{\underbrace{a[1]}_{\downarrow}, a[2], \cdots, a[\sqrt{n}], \underbrace{a[\sqrt{n}+1]}_{\downarrow}, a[\sqrt{n}+2], \cdots, a[2\cdot\sqrt{n}], \cdots, \underbrace{a[2\cdot\sqrt{n}+1]}_{\downarrow}, ...a[n]}^{\texttt{array}\ a}$$

3. Client finds in which interval x is. He looks for such a k that:

$$k \in [1..\sqrt{n}-1] : a[(k-1)\cdot\sqrt{n}+1] \leq x < a[k\cdot\sqrt{n}+1] \text{ otherwise } k = \sqrt{n}$$

4. Client sends to Server vector v of length where all of the elements are $E_{pk}(0)$. Only at position k there is $E_{pk}(1)$.

$v =$

1	2	$\cdots$	$k-1$	k	$k+1$	$\cdots$	n
$E_{pk}(0)$	$E_{pk}(0)$	$\cdots$	$E_{pk}(0)$	$E_{pk}(1)$	$E_{pk}(0)$	$\cdots$	$E_{pk}(0)$

5. Server for each $i \in [1..\sqrt{n}]$:
   for each $j \in [1..\sqrt{n}]$ calculates:

$$r[i][j] = a[(i-1) \cdot \sqrt{n} + j] \cdot_h v[i]$$

   Which means that for $i \neq k$:

$$r[i][j] = E_{pk}(0)$$

   and otherwise (for $i = k$):

$$r[i][j] = E_{pk}(a[(k-1) \cdot \sqrt{n} + j])$$

6. Server for each $j \in [1..\sqrt{n}]$:

$$v[j] = 0$$

   for each $i \in [1..\sqrt{n}]$ calculates:

$$v[j] = v[j] +_h r[i][j]$$

7. Server sends vector v to Client.
   Visualization of steps 5, 6 and 7:

	$j = 1$	$j = 2$	$\cdots$	$j = \sqrt{n}$
$i = 1$	$r[1][1] = E_{pk}(0 \cdot a[1])$	$r[1][2] = E_{pk}(0 \cdot a[2])$	$\cdots$	$r[1][\sqrt{n}] = E_{pk}(0 \cdot a[\sqrt{n}])$
$i = 2$	$r[2][1] = E_{pk}(0 \cdot a[\sqrt{n}+1])$	$r[2][2] = E_{pk}(0)$	$\cdots$	$r[2][\sqrt{n}] = E_{pk}(0)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$i = k-1$	$r[k-1][1] = E_{pk}(0)$	$r[k-1][2] = E_{pk}(0)$	$\cdots$	$r[k-1][\sqrt{n}] = E_{pk}(0)$
$i = k$	$r[k][1] =$ $E_{pk}(1 \cdot a[(k-1) \cdot \sqrt{n} + 1])$	$r[k][2] =$ $E_{pk}(1 \cdot a[(k-1) \cdot \sqrt{n} + 2])$	$\cdots$	$r[k][\sqrt{n}] = E_{pk}(a[k \cdot \sqrt{n}])$
$i = k+1$	$r[k+1][1] = E_{pk}(0)$	$r[k+1][2] = E_{pk}(0)$	$\cdots$	$r[k+1][\sqrt{n}] = E_{pk}(0)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$i = \sqrt{n}$	$r[\sqrt{n}][1] = E_{pk}(0)$	$r[\sqrt{n}][2] = E_{pk}(0)$	$\cdots$	$r[\sqrt{n}][\sqrt{n}] = E_{pk}(0)$

$+_h$

$$\underbrace{v}_{\downarrow} = \quad [\quad [E_{pk}(a[(k-1) \cdot \sqrt{n} + 1]), \quad E_{pk}(a[(k-1) \cdot \sqrt{n} + 2]), \quad \cdots \quad E_{pk}(a[k \cdot \sqrt{n}]) \quad]$$

8. Client decrypts all of the elements from v and checks if $x \in \mathit{decrypted}(v)$ (and this is his output).

4 Comments

In this protocol there are send $O(\sqrt{n})$ number of messages. Each message is encryption of element from domain P.

Privacy of Client is protected because all of the messages received by Server are encrypted.

Correctness of protocol is shown on the picture in step 7 of protocol.

5 Improvement

It is possible to use cipher presented in paper "Evaluating 2-DNF Formulas on Ciphertexts" written by Dan Boneh, Eu-Jin Goh and Kobbi Nisim. This cipher provides additively homomorphic property and possibility of performing one homomorphic multiplication of ciphertexts. Then is is possible to speed up algorithm to: $\sqrt[3]{n}$ number of messages. Idea is to divide array to $(\sqrt[3]{n})^2$ blocks of size $\sqrt[3]{n}$ and then looking two times for desired interval (for send interval there is used multiplication property).