BRIEF ANALYZIS OF TECHNIQUE FOR PRIVACY PRESERVING COMPUTATION[1]

Levon Aslanyan, Vardan Topchyan, Haykaz Danoyan

Abstract*: The privacy preserving computation research area is considered. The problem appear when one party have confidential data and need to do intense computations over that data, and computations will be done by the second party, which may be supposed being untrusted. So the content of the raw data should be kept private from the second party during the computations. Therefore these data are to be encrypted before sending them to the second party. Two possible solution scenarios are considered – one in physical and the second in theoretical levels. Physical level solution assumes some hardware integration and reorganizations. Theoretical level solution is based on cryptographic approach (homomorphic encryption).The main idea is to encrypt data in such a way that the owner, after getting the results of computations over the encrypted data, will be able to get the results on original data only by decrypting the received results. The paper brings description and analyzes of such known schemas. The final outcome is that practical cryptographic tools today are really not ready to be applied on privacy preserving computations, so that the way of solution is the use of heuristic data analyses models and algorithms that replace original data with synthesized data. Considering preparatory, this article is followed by the base research part where synthetic data generation is considered on base of CART algorithm and clustering type computational algorithms.*

Keywords*: Privacy preserving computations, homomorphic encryption, synthetic data generation.*

ACM Classification Keywords*: H.1 Information Systems – Models and principles, I.2.0 Artificial intelligence.*

1. Introduction

Privacy is one of the most important properties related to state or societal information and to information systems analyzing such data. When privacy restrictions applied by the user/owner, computation will guarantee that no leak of information happens during the computations [Ferrer, 96; Defays, 93].

Mission of statistical agencies and survey organizations is to disseminate summarized social or economical data. However, demand from an increasingly sophisticated and computationally capable research community for access to microdata - actual data records, as opposed to only summaries of the data - is high and growing. Dissemination of microdata vs. summaries greatly benefits society, as well as facilitates research and advances in economics, sociology, public health, and many other areas of knowledge. However, data disseminators cannot release microdata as collected, because doing so would reveal respondents' identities or values of sensitive attributes [Duncan, 91; Wallman, 04].

[1] Partially supported by grants № SCS 13RF-088, and № 3-1B340 of State Committee of Science of Ministry of education and science of Republic of Armenia

So, for public microdata releases, a special technique - statistical disclosure limitation/control (SDL) is used to alter the data in a way that maintains the utility but limits disclosure risk. Methods for SDL can be divided in two types:

- Perturbative methods [Dalenius, 82; Ferrer, 01a; Ferrer, 01b; Kim, 86; Moore, 96; Tendik, 94], which distort original values, so that the distorted values are publishable. For example, ages or incomes can be recorded in aggregated categories; or data values can be swapped for selected records, e.g. switch the sexes of some men and women in the data, in hopes of discouraging users from matching, since matches may be based on incorrect data. Or, they add noise to numerical data values to reduce the likelihood of exact matching on key variables or to distort the values of sensitive variables;
- Synthetical methods [Feinberg, 06; Sanz, 99; Reiter, 02; Reiter, 05], which lead to release of synthetical data - random samples drawn from the distribution representing original data. It is necessary to release multiple versions of synthetic data in order to guarantee the validity of statistical inferences. Other variant of synthetical approach is release of partially synthetical data, when only some of the values, which are considered sensitive, are synthesized, while others are left unchanged.

The encryption is one of the techniques that provide privacy of information. As a limitation to this technique it is enough to mention that an information system which works with encrypted data can at most store or retrieve the data for the user; and any more complicated operations seem to be requiring the data decryption before being operated on. Effective search over the encrypted data requires that the encryption scheme preserves the distances and similarities [Miyoung, 13]. Some particular encryption functions which permit encrypted data to be operated on without preliminary decryption of the operands are known for several sets of interesting operations. These special encryption functions are called “privacy homomorphisms” which form an interesting subset of arbitrary encryption schemes called “privacy transformations”. The idea of the discussion below is to learn if such partial schemes can be combined to an integrated application system for privacy preservation.

For example, let us consider a statistical organization which owns confidential data (financial data, credit card data, health data, etc.). And let there is a computational environment (datacenter, cluster, grid, etc.) to be used for information analysis. In simple scenario we will suppose that data is stored on a data bank in encrypted form, Figure 1. For such organization to provide not archiving but the necessary computations over the data, the following 2 ways – one physical level solution and the second - a theoretical level solution can be considered:

I. Allowing the computational system to store the data in decrypted form at the time of computations. This approach needs some additional hardware modifications to provide the security and will be considered in section 2.

II. The data in computational system stays always in encrypted form. In this case the encryption function must satisfy some additional properties such that the results of computation on original data coincide with the decrypted results of computations on encrypted data. More precisely this approach will be considered in section 3.

Since I. and II. provide limited opportunities for privacy preservation after the section 3 we will discusse the privacy preservation computation heuristics.

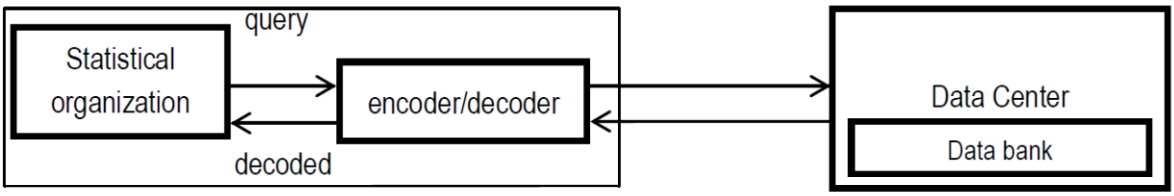

Figure 1. Encrypted data are stored in Data center

2. Physical Level Solution

Let us consider an example that shows how a computational system might be reorganized in a form to solve the problem of performing operations on decrypted data securely [Rivest, 78]. Those modifications are presented in Figure 2.

In this example, in addition to the standard register set and ALU, a secure register set and ALU is a added to the infrastructure. In this case, all communication of data between operation memory and the physically secure register set passes through an encoder-decoder E supplied with the user's key, so the unencrypted data can exist only within the physically secure register set. Moreover, it follows that all sensitive data in operating memory, data bank files, ordinary register set, and on the communications channel will be encrypted. During operation, a load/store instruction between operating memory and the secure register set will automatically cause the appropriate performed decryption/encryption operations.

An obvious problem is getting the encoder/decoder E which loaded with the user's key K without compromising the security of the user's key. In this case, one possible solution is to keep the user's key encrypted under control of a system key S. The encrypted form of key K, $E_S(K)$, can be transmitted over the insecure channel to the system, decrypted by the physically secure decoder F, and loaded into the encoder-decoder E.

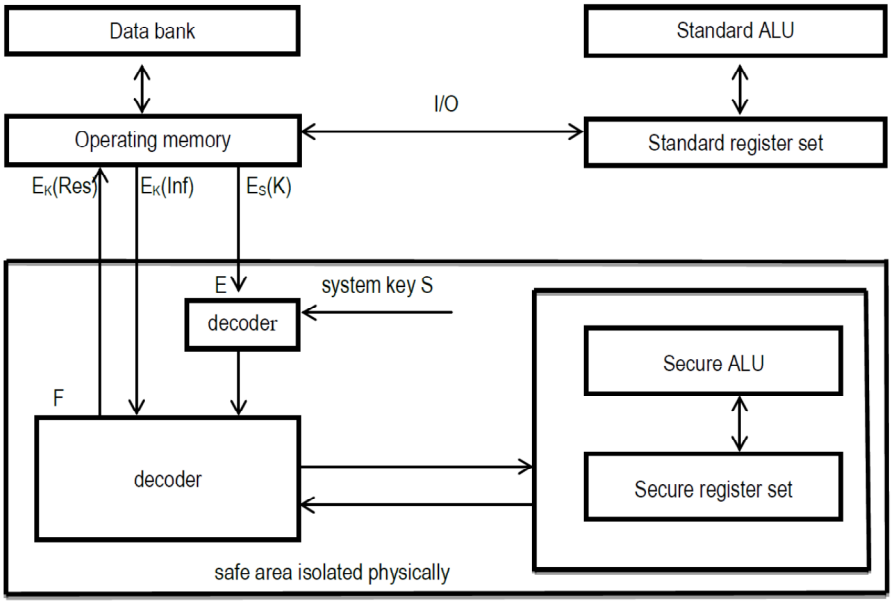

Figure 2. Computation in a secure environment

Remark: In addition to key management problems, in this scheme there appear speed degradation type questions, caused by invoking the encryption/decryption in each load and store.

3. Theoretical Level Solution

3.1 Basic Concepts of Encryption

Encryption (enciphering) is one of the main methods to protect data privacy. A process of enciphering is the transformation of one message (initial data), called a plaintext (irrespective to the data type), to another message, called ciphertext, using some specific transformation. As a rule in a cryptographic scheme this transformation is open, publicly known, and a secret key is incorporated to provide the security. The process of transformation of ciphertext into plaintext is called deciphering.

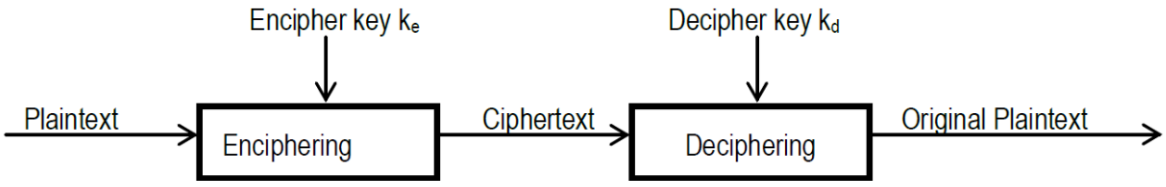

Figure 3. Enciphering/deciphering cycle

Let us denote a set of all possible keys by K and let each $\mathrm{k} \in \mathrm{K}$ can be represented as a tuple $\mathrm{k} = (\mathrm{k_e}, \mathrm{k_d})$, where $\mathrm{k_e}$ is an encipher key and the $\mathrm{k_d}$ is a decipher key. Let P be the set of all possible plaintexts and C be the set of all possible ciphertexts. The enciphering function is denoted by $\mathrm{E_k}: \mathrm{P} \to \mathrm{C}$ and the deciphering function is denoted by $\mathrm{D_k}: \mathrm{C} \to \mathrm{P}$.

So a cryptosystem is a five-tuple $(\mathrm{P}, \mathrm{C}, \mathrm{K}, \mathrm{E}, \mathrm{D})$, where the following condition is satisfied:

for each $\mathrm{k} \in \mathrm{K}$, there is an enciphering function $\mathrm{E_k} \in \mathrm{E}$ and corresponding deciphering function $\mathrm{D_k} \in \mathrm{D}$, each $\mathrm{E_k} : \mathrm{P} \to \mathrm{C}$ and $\mathrm{D_k} : \mathrm{C} \to \mathrm{P}$ are functions such that $\mathrm{D_k}(\mathrm{E_k}(\mathrm{p})) = \mathrm{p}$ for every plaintext $\mathrm{p} \in \mathrm{P}$.

Later, as mentioned, we will consider that the structure of $\mathrm{E_k}$ is known, or in other words - the safety of data do not depend on the secret structure of the encryption algorithm.

There are two main types of cryptosystems based on the keys: symmetric and asymmetric (public-key).

Symmetric cryptosystems are those where the encryption key can be calculated from the decryption key and vice versa. In most symmetric cryptosystems encryption and decryption keys are the same. These cryptosystems also called secret-key cryptosystems or cryptosystems with a single key require that the sender and recipient have agreed to use the key before secure messaging.

Asymmetric cryptosystems, also called public-key cryptosystems, are those when the encryption key is known but it is practically impossible to calculate the decryption key, even having some additional information (known plaintext attack, known ciphertext attack, etc.) So, in asymmetric cryptosystems $\mathrm{E_k}$ can be safely made public without allowing an adversary to decrypt messages.

3.2. Privacy Homomorphism

The idea of performing simple computations on encrypted data was first put forward by Rivest, Adleman, and Dertouzous [Rivest, 78] who referred to such computations as privacy homomorphism. The original motivation for privacy homomorphism was to allow for encrypted database to be stored by the untrusted second party, while still allowing the owner to perform simple updates and queries such that noting about the database content is revealed to the third party.

Let us consider two algebraic systems to represent plaintext and ciphertext systems. First system is the plaintext system $\mathcal{P}$, which consists from plaintext set P, and some operations $\mathrm{f}_1, \ldots, \mathrm{f}_\mathrm{n}$. And the second system is the ciphertext system $\mathcal{C}$, which consists of ciphertext set C, and some operations $\mathrm{g}_1, \ldots \mathrm{g}_\mathrm{n}$. For example, the system consisting of integers under the usual set of operations might be denote $< Z, +, \times >$; where Z is set of integers. Formally, the plaintext/ciphertext formalism looks as follows: $\mathcal{P} =< P, \mathrm{f}_1, \ldots, \mathrm{f}_\mathrm{n} >$, and $\mathcal{C} =< C, \mathrm{g}_1, \ldots, \mathrm{g}_\mathrm{n} >$. We must have also a set of encryption functions $\mathrm{E} = \{\mathrm{E}_\mathrm{k} : \mathrm{P} \to \mathrm{C} \,/\, \mathrm{k} \in \mathrm{K}\}$ and the set of decryption functions $\mathrm{D} = \{\mathrm{D}_\mathrm{k} : \mathrm{C} \to \mathrm{P} \,/\, \mathrm{k} \in \mathrm{K}\}$.

The encryption schema will be called a privacy homomorphism if the following takes place [Brickell, 87; Fontaine, 07]:

$$\forall \mathrm{i}\ \mathrm{D}_\mathrm{k}(\mathrm{g}_\mathrm{i}(\mathrm{E}_\mathrm{k}(\mathrm{a}), \ldots, \mathrm{E}_\mathrm{k}(\mathrm{b})) = \mathrm{f}(\mathrm{a}, \ldots, \mathrm{b}), \forall \mathrm{a}, \mathrm{b} \in \mathrm{P}.$$

In [Rivest, 78] is brought more general definition which requires that the decryption function will be a homomorphism.

For example, suppose we want to compute $\mathrm{f}_1(\mathrm{a}, \mathrm{b})$. We need only to ask the system to compute $\mathrm{g}_1(\,\mathrm{E}(\mathrm{a}), \mathrm{E}(\mathrm{b}))$. Since the schema is a privacy homomorphism $\mathrm{f}_1(\mathrm{a}, \mathrm{b}) = \mathrm{D}(\,\mathrm{g}_1(\,\mathrm{E}(\mathrm{a}), \mathrm{E}(\mathrm{b})\,)$, so we arrive at the encrypted form of the answer without having to decrypt the intermediate results.

Below the requirements on the choice of the algebraic system $\mathcal{C}$ and the functions E_k, D_k are provided:

1. Encryption and decryption functions, respectively E_k and D_k , should be easy to compute.
2. The operations g_i in ciphertexts system $\mathcal{C}$ should be efficiently computable.
3. An encrypted version of a plaintext d_i , $\mathrm{E}_\mathrm{k}(\mathrm{d}_\mathrm{i})$, should not require much more space to represent than a representation of d_i.
4. Knowledge of $\mathrm{E}_\mathrm{k}(\mathrm{d}_\mathrm{i})$ for many plaintexts d_i should not be sufficient to reveal E_k. (Ciphertext only attack).
5. Knowledge of d_i and $\mathrm{E}_\mathrm{k}(\mathrm{d}_\mathrm{i})$ for several values of d_i should not reveal E_k. (Chosen plaintext attack).
6. The operations and predicates in ciphertext system $\mathcal{C}$ should not be sufficient to provide efficient computation of decryption function D. (This applies primarily to use comparisons).

4. Examples of Privacy Homomorphisms

This section presents some examples of privacy homomorphisms to support the hypothesis that useful privacy homomorphisms may exist for many applications. Moreover, we present cryptanalysis of these examples [Rivest, 78; Brickell, 87]. Mention, that simply, criptosystems RSA and ElGamal are multiplicatively homomorphic [Fontaine, 07].

A. In this example the system of plaintext data is $\mathrm{P} =< \mathrm{Z}_{\mathrm{p}-1}, +_{\mathrm{p}-1}>$, system of integers modulo $(\mathrm{p}-1)$ with operation of addition by modulo $(\mathrm{p}-1)$, where p is a prime number such that [Brickell, 87]

$$p-1=\prod_{i=1}^{k} p_i^{d_i}, \text{ and for all } i, p_i \leq B, \text{ for some small } B. \quad (1)$$

And the corresponding system of ciphertext data is $\mathrm{C} =< \mathrm{Z}_{\mathrm{n}}, \times_{\mathrm{n}}>$, system of integers modulo n with operation of multiplication modulo n, with the product of p and large prime q. Encryption process is defined as follows: plaintext P is encrypted by computing $\mathrm{E_k(M)} \equiv \mathrm{g^M} \mathrm{mod\ n}$, where g is a generator of multiplicative group Z_p^* under multiplication by modulo p (i.e. $\forall \mathrm{a} \in \mathrm{Z_p^*}$ can be represented as $\mathrm{g^i}$ for some integer i) and k is an encryption key, $\mathrm{k} = (\mathrm{p}, \mathrm{q})$. And invers process, decryption, is $\mathrm{D_k(C)} \equiv \log_{\mathrm{g}} \mathrm{C}\ (\mathrm{mod\ p})$. The structure of p enables computation of the discrete logarithm by the method of Pohlig and Hellman [Pohling, 78] in time $\mathrm{O}(\mathrm{B}^{1/2})$.

To show that this schema is a privacy homomorphism let us prove that $\forall \mathrm{M_1}, \mathrm{M_2} \in \mathrm{P}$ $\log_{\mathrm{g}} \mathrm{g^{M_1}} \times_{\mathrm{n}} \mathrm{g^{M_2}}\ \mathrm{mod p} = \mathrm{M_1} +_{\mathrm{p}-1} \mathrm{M_2}$. We have that $\log_{\mathrm{g}}(\mathrm{g^{M_1}} \times_{\mathrm{n}} \mathrm{g^{M_2}}) = \log_{\mathrm{g}} \mathrm{g^{M_1+M_2}}\ \mathrm{mod p}$, where the sum $\mathrm{M_1} + \mathrm{M_2}$ is taken by its usual means (not by modulo p-1). Let us denote $\mathrm{x} = \log_{\mathrm{g}} \mathrm{g^{M_1+M_2}}\ \mathrm{mod p}$. By definition we have that $\mathrm{g^x} \equiv \mathrm{g^{M_1+M_2}} \mathrm{mod p}$. By definition of g we have that $\mathrm{g^{M_1+M_2}} = \mathrm{g^{M_1 +_{p-1} M_2}} \mathrm{mod p}$. Therefore $\mathrm{x} = \mathrm{M_1} +_{\mathrm{p}-1} \mathrm{M_2}$, which completes the proof.

We will say that the number n is B-powersmooth if each prime power dividing n is less than or equal to B [Pollard, 74].

This system is insecure because (1) allow us to factor n (with high probability) by the Pollard $(\mathrm{p}-1)$ method [Brickell, 87; Cohen, 93]. The basic idea of this factoring algorithm is the following. We have number n and let p be a prime divisor of n. Let $\mathrm{a} > 1$ be an integer such that $\mathrm{GCD(a, n)} = 1$, otherwise consider the factor of n to be found. According to Fermat's little theorem, $\mathrm{a^{p-1}} \equiv 1 (\mathrm{mod p})$. Let $\mathrm{p}-1$ to be B-powersmooth for a small *B*. Then by definition $\mathrm{p}-1$ divides the least common multiple of the numbers from 1 to *B,* which we will denote by $\mathrm{lcm}\{1, .., \mathrm{B}\}$. Hence, $\mathrm{a^{lcm\{1,..,B\}}} \equiv 1 (\mathrm{mod p})$, which implies that $\gcd\left(\mathrm{a^{lcm\{1,..,B\}}} - 1, \mathrm{n}\right) > 1$.

Note that in this case the adversary does not know the plaintext data set, but in public-key cryptosystems there are considered that the adversary knows everything expect the private key.

B. Suppose that the system of plaintext data is the integers modulo p with operation multiplication and test for equality $< \mathrm{Z_p}, \times_{\mathrm{p}}>$, where p is prime number [Rivest, 78]. A corresponding system of ciphertext data is the integers modulo n with operation multiplication and test for equality $< \mathrm{Z_n}, \times_{\mathrm{n}}>$, as in the previos example, letting

$n = p \cdot q$, where q is large prime and supposing that n is difficult to factor. The encoding and decoding functions are the same as that used by Rivest, Shamir, and Adleman in their method of implementing public-key cryptosystems. Specifically, the encryption and decryption functions E_k and D_k are $E_k(M) \equiv M^e (\mathrm{mod}\ n)$, for a message M and $D_K(C) \equiv C^d (\mathrm{mod}\ p)$, for a ciphertext C, where e and d are integers, such that $GCD(e, \varphi(pq)) = 1$ and $ed \equiv 1 \mathrm{mod} \varphi(pq)$. Recall that by $\varphi(n)$ is denoted the Euler function. The encryption key is thus the pair of positive integer e and n, (e, n). Similarly, the decryption key is the pair of positive integer d and n, (d, n). Each user makes his encryption key public, and keeps the corresponding decryption key private. Since $(x^e)(y^e) = (xy)^e$, this is a homomorphism. The security of this system should be very good, even if the computer system is given the both e and n.

Note that in this case the adversary does not know the plaintext data set too because the number p is unknown.

C. In this example, the system of plaintext data is $< Z_n, +_n, \times_n >$, integers modulo n with operations of addition and multiplication by modulo n, where n is the product of two large primes p and q, $n = p \cdot q$. In turn, the system of ciphertext data we take a set $Z_p \times Z_q$ and operations are componentwise version of operations on plaintext data (i.e. operations on the first component is performed by modulo p, and over the second - by modulo q). It remains to describe the encryption and decryption functions (process). Encryption function defined as $E_k(a) = (a \bmod p, a \bmod q)$. The encryption key k, defined as a pair of numbers p and q, $k = (p, q)$. And decryption is: given key $k = (p, q)$, decryption function $D_k((b, c))$ is computed using the Chinese remainder theorem.

Now we will show that this method of encryption is Privacy Homomorphism. For simplicity, we prove the case with the addition operation. For multiplication, the proof is similar.

Suppose we have the following plaintexts P1, P2, P3 and their corresponding ciphertexts C1, C2 and C3, which have the form $E_k(P_i) = C_i = (c_i^p, c_i^q)$; for all i, $1 \le i \le 3$, where

$$c_i^p = P_i \, mod p \ ; \ 1 \le i \le 3 \quad (2)$$

$$c_i^q = P_i \, mod q \ ; \ 1 \le i \le 3 \quad (3)$$

More, let C_3 is a sum of C_1 and C_2, i.e. $c_3^p = (c_1^p + c_2^p) \mathrm{mod} p$ and $c_3^q = (c_1^q + c_2^q) \mathrm{mod} q$. In this case we need to show that $P_3 = P_1 + P_2 \mathrm{mod} n$.

The expressions (2) and (3), they can be written differently:

$$P_i = c_i^p \bmod p \ ; \ 1 \le i \le 3$$

$$P_i = c_i^q \bmod q \ ; \ 1 \le i \le 3$$

Based on the properties of modular arithmetic, we can find that

$$P_1 + P_2 = (c_1^p + c_2^p) \bmod p$$
$$P_1 + P_2 = (c_1^q + c_2^q) \bmod q$$

Finally, since n is a multiple of p and q, then we have the following expressions:

$$(P_1 + P_2) \bmod n = \left((c_1^p + c_2^p) \bmod p\right) \bmod n = (c_1^p + c_2^p) \bmod p$$
$$(P_1 + P_2) \bmod n = \left((c_1^q + c_2^q) \bmod q\right) \bmod n = (c_1^q + c_2^q) \bmod q$$

So we got that $P_3 = (P_1 + P_2) \bmod n$, that exactly what we wanted to prove.

Unfortunately, this Privacy Homomorphism can be broken, i.e. p and q can be discovered - using a known plaintext attack. Namely, assume that cryptanalyst has plaintext-ciphertext pairs: $P_i, (C_1^p, C_2^q), 1 \le i \le r$, where C_i^p and C_i^q computed according to (2) and (3), i.e.

$$C_i^p \equiv P_i \bmod p \quad \text{and} \quad C_i^q \equiv P_i \bmod q \;;\; 1 \le i \le r$$

According (3) follow that p divides $(C_i^p - P_i)$, for all $i = 1,2,3, \dots, r$. Suppose p' is the gcd of numbers $\{(C_i^p - P_i) \mid 1 \le i \le r\}$. Since p is a common divisor for the numbers $\{(C_i^p - P_i) \mid 1 \le i \le r\}$, then it will be a divisor and for p', i.e. p'/p. Similarly, if q' is GCD of numbers $\{(C_i^q - P_i) \mid 1 \le i \le r\}$, then we see that q'/q. And finally, If p'=p and q'=q, then cryptanalyst can decrypt any ciphertext. Even for small r, there is high probability that p'=p, q'=q. Even if it's not as, then if for any new ciphertext cryptanalyst is given the plaintext, he can improve his knowledge of p or q. In particular, given ciphertext (C^p, C^q), the cryptanalyst can find P' such that $P' \equiv C^p \bmod p'$ and $P' \equiv C^q \bmod q'$. If P' ≠ P, then follow that $P \ne C^p \bmod p'$ or $\ne C^q \bmod q'$. So if cryptanalyst given P, it can improve the value of p' and q' by replacing p' by $\mathrm{GCD}(P - C^p, p')$ and q' by $\mathrm{GCD}(P - C^q, q')$.

D. For this example as a system of plaintext data we take the set of integers under the usual operations of addition and multiplication, $P = < Z, +, \times >$. Encryption process is performed as follows: the user chooses an integer n and represents all of his data in radix-n notation, i.e. if $y = d_m n^m + d_{m-1} n^{m-1} + \cdots + d_1$ then the n-radix of y will be the vector $(d_m, \dots, d_1)$.. Corresponding operations on ciphertext data defined the same as in the case of algebraic polynomials by allowing individual coordinate positions to exceed n. For example, if n=15, we have

$$E_k(42) = (2, 12),\ E_k(23) = (1, 8),\ E_k(65) = (3, 20),\ E_k(966) = (2,28,96)$$

where k is encryption key, which defined by the integer n, k = (n). Easy to see that the computer system can operate on ciphertext data without knowledge of n, this means that the encryption is Privacy Homomorphism. Without loss of generality, we will show that in the case of the addition operation. To do this we need to show that, if we have two plaintexts P1 and P2, for which:

$$E_k(P_1) = (a_r, a_{r-1}, \dots, a_1) \quad (4)$$

$$E_k(P_2) = (b_s, b_{s-1}, \dots, b_1) \quad (5)$$

$$E_k(P_1) + E_k(P_2) = (c_m, c_{m-1}, \dots, c_1)$$

Then we need to show that the following equality holds: $(\mathrm{P}_1 + \mathrm{P}_2) = D_k((\mathrm{c_m}, \mathrm{c_{m-1}}, \dots, \mathrm{c_1}))$, where m = max(r, s) and c_i are the result of adding the values of the corresponding coordinates. For that let calculate the value of $(\mathrm{P}_1 + \mathrm{P}_2)$. From (4), (5) follow that P_1 and P_2 can be represented as follows:

$$P_1 = a_r \cdot n^{r-1} + a_{r-1} \cdot n^{r-2} + \cdots + a_2 \cdot n + a_1 \quad (6)$$

$$P_2 = b_s \cdot n^{s-1} + b_{s-1} \cdot n^{s-2} + \cdots + b_2 \cdot n + b_1 \quad (7)$$

According to (6) and (7) follows that $(\mathrm{P}_1 + \mathrm{P}_2)$ is equal to $\mathrm{P}_1 + \mathrm{P}_2 = \mathrm{c'_{m'}} \cdot \mathrm{n^{m-1}} + \mathrm{c'_{m'-1}} \cdot \mathrm{n^{m-2}} + \cdots + \mathrm{c'_2} \cdot \mathrm{n} + \mathrm{c'_1}$. From the last equality and the definitions of coordinates $\mathrm{c_i}$ and m it follows that $\mathrm{m'}$ coincide with m and coefficients $\mathrm{c'_i} = \mathrm{c_i}$, for all $\mathrm{i} = 1,2,$ .... And this in turn proves that $(\mathrm{P}_1 + \mathrm{P}_2) = D_k((\mathrm{c_m}, \mathrm{c_{m-1}}, \dots, \mathrm{c_1}))$.

As in the previous example, this system also insecure and can be broken by a known plaintext attack. Assume that the cryptanalyst has plaintext-ciphertext pairs $\mathrm{P_i}$, $(\mathrm{c^i_{k_i}}, \mathrm{c^i_{k_{i-1}}}, \dots, \mathrm{c^i_1})$ for all $\mathrm{i} = 1,2, \dots, \mathrm{s}$. Based on (6) and (7) follows that $(\mathrm{P_i} - \mathrm{c^i_1})/\mathrm{n}$. Let n' be the GCD of numbers $\{\mathrm{P_i} - \mathrm{c^i_1} \mid 1 \le \mathrm{i} \le \mathrm{s}\}$. Since n is common divisor for numbers $\{\mathrm{P_i} - \mathrm{c^i_1} \mid 1 \le \mathrm{i} \le \mathrm{s}\}$, then this means that n is divisor for n', $\mathrm{n'/n}$ .If $\mathrm{n} = \mathrm{n'}$, the cryptanalyst can decrypt any ciphertext. Even for small s, there is a high probability that $\mathrm{n} = \mathrm{n'}$. Even if it's not the case, then for any new ciphertext cryptanalyst is given the plaintext, he can improve his knowledge of n. Specifically, given ciphertext $(\mathrm{c_{k_1}}, \mathrm{c_{k_1-1}}, \dots, \mathrm{c_1})$, the cryptanalyst can find $\mathrm{P'}$ such that $\mathrm{P'} \equiv \mathrm{c_1 mod\ n'}$. If $\mathrm{P'} \neq \mathrm{P}$, then follow that $\mathrm{P} \neq \mathrm{c_1 \bmod n'}$. So if cryptanalyst given P, it can improve the value of n' by replacing it with $\mathrm{GCD(P - m_1, n')}$.

E. As in the previous example, as a system of plaintext data we take the set of integers under the usual operations of addition, and multiplication $\mathrm{P} = < Z, +, \times >$. More, let $\mathrm{a_0}, \mathrm{a_1}, \dots, \mathrm{a_{n-1}}$ by randomly chosen positive integers and A be the matrix:

$$A = \begin{pmatrix} 1 & \cdots & a_0^{n-1} \\ \vdots & \ddots & \vdots \\ 1 & \cdots & a_{n-1}^{n-1} \end{pmatrix}, \quad (8)$$

where n is chosen so that all intermediate results used in any calculation are less than 2^n. Encryption function defines as follows: given plaintext P, whose binary representation is $\overline{P} = (p_0, p_1, \dots, p_{n-1})_2$, the encryption of P is the column vector $\overline{C}$:

$$\overline{C} = E_k(P) = A \cdot \overline{P} = \begin{pmatrix} f_p(a_0) \\ \vdots \\ f_p(a_{k-1}) \end{pmatrix}, \quad (9)$$

where k is the encryption key, which constructed from the integers $a_0, a_1, \dots, a_{n-1}$, $k = (a_0, a_1, \dots, a_{n-1})$. And function $f_p(x)$ defined as follows: $f_P(x) = \sum_{i=0}^{k-1} p_i \cdot x^i$.

Operations on encrypted data are component wise version of addition and subtraction over the integers. Decryption is performed by multiplying ciphertext (column vector) $\overline{C}$ by matrix A^{-1}: $D_k(\overline{C}) = A^{-1} \cdot \overline{C}$. This encryption is Privacy Homomorphism. It is true, let we have two plaintexts P_1, P_2 and their corresponding ciphertexts are $\overline{C_1} = E_k(P_1) = A \cdot \overline{P_1}$ and $\overline{C_2} = E_k(P_2) = A \cdot \overline{P_2}$, where $\overline{P_1}$ and $\overline{P_2}$ are binary representations of P_1, P_2. We need to show that, if $\overline{C_1} \pm \overline{C_2} = \overline{C_3}$, then should take place the following equality: $P_1 \pm P_2 = D_k(\overline{C_3})$. To this end, consider the expression $(\overline{C_1} \pm \overline{C_2})$. According to (8) and (9), we find that $(\overline{C_1} \pm \overline{C_2}) = A \cdot \overline{P_1} \pm A \cdot \overline{P_2}$, but $\overline{C_1} \pm \overline{C_2} = \overline{C_3}$. Thus, it turns out that

$$\overline{C_3} = A \cdot \overline{P_1} \pm A \cdot \overline{P_2} = A \cdot (\overline{P_1} \pm \overline{P_2})$$

And this means that $(\overline{P_1} \pm \overline{P_2}) = A^{-1} \cdot \overline{C_3} = D_k(\overline{C_3})$. This equality proves that this encryption is Privacy Homomorphism.

This encryption is insecure and can be broken by ciphertext only attack. But first, let us consider one interesting fact. Suppose we are given an integer x and its binary representation $\overline{x} = (x_0, x_1, \dots, x_{n-1})_2$. Moreover, suppose j is the largest index in binary representation $\overline{x}$ such that $x_j = 1$, i.e. $x_{j+1} = x_{j+2} = \dots = x_{k-1} = 0$. Let z be any positive integer, then we have the following inequality: $z^j \le \sum_{i=0}^{j} x_i \cdot z^i < \sum_{i=0}^{j} z^i < (z+1)^j$. This means that $\left\lfloor (\sum_{i=0}^{j} x_i \cdot z^i)^{\frac{1}{j}} \right\rfloor = z$. From this fact follows that this encryption is weakens to ciphertext only attack. Suppose cryptanalyst has a cipher $\overline{C} = (c_0, \dots, c_{n-1})$ and he guess a value for j, the largest index such that $x_j = 1$. Then he can compute $\left\lfloor {c_0}^{1/j} \right\rfloor = b_0$ and write c_0 in base b_0 notation. If all coefficients are 0 and 1, then probably $b_0 = a_0$ and x is easily found. More, values $c_1, c_2, \dots, c_{k-1}$ can be used as an additional check. Otherwise, cryptanalyst can try a different choice for j.

There are known some security considerations for encryption systems [Fontaine, 07]. According [Fontaine, 07] the highest level of security which can be achieved by homomorphic encryption system is IND-CPA (Indistinguishability chosen plaintext attack).

In 2009 IBM (his employee Craig Gentry) presented the new Fully Homomorphic scheme - i.e. a scheme that allows one to evaluate circuits over encrypted data without being able to decrypt [Greaig, 09]. Gentry's scheme is completely impractical. It uses something called an ideal lattice as the basis for the encryption scheme, and both the size of the ciphertext and the complexity of the encryption and decryption operations grow enormously with the number of operations you need to perform on the ciphertext - and that number needs to be fixed in advance. And converting a computer program, even a simple one, into a Boolean circuit requires an enormous number of operations.

The other example of homomorphic encryption is brought in [Khan, 12], mathematical base of which is p-adic rings, but there are different opinions about the practical value of this system.

5. Complementary means of privacy computations

The need for new methods of disclosure limitation is further developed in the light of the modern advances in data mining, which can be used to undo the mask or to get narrow interval estimates for the original values. Data mining techniques aimed at the level of individual record pose a big threat to the respondents' privacy. [Rivest, 78] is one of the early referred papers the direction of privacy preserving data mining. These approaches however, focus on the secure multiparty computation. Existing literature on the protection of statistical databases against data mining attacks in the literature are very few, and can be characterized as the first steps made in this direction [Little, 93]. Current research projects aim to account the threats posed by data mining techniques by developing methodology of protection and using data mining to create masked data. Alternatively the project aims to apply data mining technique in synthetic data generation and in data distortion for general disclosure limitation.

Specifically, the research will result in the following:

- Methods of assessing attribute disclosure risk for different scenarios of data release. Intruders' actions will be simulated by the application of the relevant data mining technologies to the data sets, masked by different SDL methods;
- Methods, which reduce, attribute disclosure risk, based on the results of the previous step;
- Methods, which use local masking, approach to improve overall data utility and protect particular groups of individuals, e.g. outliers;
- Methods of masking for the data sets with special structural relationships between the variables, which should be preserved in the released, masked data.

These results will improve the quality of the released data and protect the confidentiality of individual information. Viewed more broadly, the results will facilitate better dissemination of different types of statistical data.

There is a variety of techniques in data mining, which can be used as classifiers for categorical variables and/or predictors for quantitative variables. These tools can be applied by the illegitimate data users to a particular record or a group of records with the goal of obtaining narrow interval estimates of the sensible variable. One of such tools is rule mining [Agrawal, 00; Aslan, 04].

Association Rule Mining (ARM) is the most popular data mining technique in the line of rule-based models, Incremental Reduced Error Pruning (IREP) and Frequent Fragments Mining. Assume we have a set $I =$

$\{I_1, I_2, \ldots, I_n\}$ of n different attributes and let $X \subseteq I$. Given a database D with records of type X, transactions. We say that $T \in D$ supports X, if $X \subseteq T$. Consider the standard concepts of support and confidence

$$supp(X) = |\{T \in D | X \subseteq T\}|/|D|$$
$$conf(X \to Y) = supp(X \cup Y)/supp(X)$$

An association rule is the expression of form $\{X \to Y$, confidence, support$\}$, where $X \cap Y = \emptyset$. Here support is simply the proportion of records in D that contain $X \cup Y$ and confidence is the proportion of records containing X for which the rule is correct. The problem of mining association rules is to generate all association rules that have support and confidence greater than the user-specified minimum support and minimum confidence.

The problem of discovering all association rules can be decomposed into two sub-problems:

1) Find all subsets from that have support above minimum support frequent subsets.
2) Use the found subsets to generate rules.

The first step can be accomplished using the following iterative algorithm. In the first round the algorithm counts the support of individual attributes and determines which of them have minimum support. In each subsequent pass, the algorithm starts with a set of attributes found in the previous round and uses this set and single item additions for generating new potentially frequent subsets, called candidates, and counts the actual support for these subsets. At the end of the pass, it is determined which of the candidate subsets are actually frequent, and they are used for the next pass. The process continues until no new frequent subsets are found. There is a well-known algorithm in this domain, APRIORI, which is factually the de facto standard for ARM. Algorithm is recursive. The basic intuition is that any subset of a frequent subset must be frequent which monotonicity type property is. Therefore, the candidate subset having k elements can be generated by joining frequent subsets having $k-1$ elements, and deleting those that contain any subset that is not frequent. This procedure results in generation of a much smaller number of candidate subsets.

For solving the second sub-problem the following algorithm can be used. For every frequent subset X, find all non-empty subsets $a \subseteq X$ and output a rule of the form $a \to (X - a)$ if the ratio of support of X to support of a (the confidence of the rule) is greater than minimum confidence.

Finding all frequent item sets in a database is difficult since it involves searching almost all possible item sets and uses several passes over these data. The set of possible item sets is the power set of I and this algorithmic problem is evidently NP hard. Even the system of maximal possible frequent itemsets, which is a Sperner system, is large. There are several critical issues related to ARM.

Privacy-preserving ARM starts with information, which may contain intentionally wrong survey information items, as it is the case of masked data. The reconstructed support values cannot coincide exactly with the actual supports, and errors positive or negative may occur, having more pernicious effect than just a wrong cipher, enlarging or narrowing the set of rules mined. Often the rule sets produced are large, but most of them are uninteresting. Control of number of associations produced by change of support may reduce them to a

manageable number, but this may lead to the loss of interesting rules. Support threshold alone is not enough to find interesting structures. Symmetric combinations of attributes may lead to a large set of similar or unnecessary rules mined. At this point ARM recovers many associations between the attributes, which do not have cause-and-effect relationship. There are a number of proposals, but they lack the systematic approach. It turns out that the same mathematical tool can address such behavior effectively.

The model, which will be investigated, is based on a mechanism called chain split and computations [Aslan, 09; Tonoyan, 76]. This approach allows generating the frequent subsets using the minimal possible amount of memory. Chain split is a special partition of n-cube vertices set into the growing chains. The special property provides that the 3-chain fragments with complementary vertex α may be determined by α and this is the key of chain computations. At the moment algorithms are computing and saving the maximal frequent item sets there is an inside potential to consider extended chain computation, which uses lager chain fragments, which leads to approximate results.

Moreover, thinking from point of view of monotone Boolean functions, it is important to construct the absorbing function of limited complexity, which may play the role of best approximation scaling the frequent sets density and uniting several sets of upper, zero points into the imputed values. Specifically, in chain split data mining for privacy preserving use, it may be given general limitations for frequent subsets in term of their sizes. This requires extending the chain split to these conditions, which allows finding more effective chain computation algorithms. Chain computation algorithms will be extended to serve such constructions and likely an approximate threshold will be introduced. Relative compliments on chains will be modified from distance 2 to higher distances. Expected results of this part of the project are monotone Boolean approximations. These represent a new approach with high value due to massive applications of monotone Boolean functions in different areas. Such mechanism seems to be necessary to generate frequent sets effectively when a set of input datasets are to be designed and analyzed from the point of view of a hacker actions. The approximate computation is hard computationally and heuristic in its nature.

Bootstrap and boosting approaches. To increase an accuracy of the prediction and classification the intruder can use bootstrap aggregation or bagging. In particular, for iteration i $(i = 1, 2, \dots, k)$, a training set D_i, of d records is sampled with replacement from the original set of records, D. A classifier model M_i, is learned for each training set, D_i. To classify an unknown record, X, each classifier, M_i, returns its class prediction, which counts as one vote. The bagged classifier, M_*, counts the votes and assigns the class with the most votes to X. If the intruder's goal is the prediction of the continuous variable, bagging can be applied by taking the average value of each prediction for a given record. The bagged classifier often has significantly greater accuracy than a single classifier derived from D, the original training data. Bagged classifier is also quite robust to the effects of noisy data. The increased accuracy occurs because the composite model reduces the variance of the individual classifiers [Aslan, 13].

For prediction, it was theoretically proven that a bagged predictor will always have improved accuracy over a single predictor derived from D.

Another approach to increase accuracy of prediction is boosting. Classical example here is Adaboost algorithm. This algorithm assigns weights to each training record. Usually on the initial stage these weights are equal to $1/d$

for all the records in the training set D with d records. Then in the k rounds, set D is sampled with replacement according to the records' weight to form a training set D_i and this set is used to derive a classifier model M_i. After a classifier M_i is learned, its error is computed and the weights for the misclassified records are increased, so that the subsequent classifier M_{i+1}, "pay more attention" to these misclassified records. The final boosted classifier M_*, combines votes of each individual classifier. Each classifier has got a weight: the lower a classifier's error rate, the more accurate it is, and therefore, the higher it's weight for voting. For each class c, the weights of each classifier that assigned the record to a class c are summed. The class with the highest sum is the "winner" and is returned as the class prediction for this record.

Application of data mining to the original data by data protector (agency) opens the door to many possibilities in Statistical Disclosure Limitation. Data mining algorithms, in particular association rule mining can be applied to discover those relationships, which are not obvious. These relationships are not necessarily strict, in the sense that they may be satisfied for most of the records, but not necessarily for all the records. Algorithms for frequent set finding can be used to discover such relationships. Our research will focus on the application of these algorithms for discovery of such relationships, but most importantly for the creation of masked data sets.

So, to be summarized, the research agenda of the second part of the project is focused on the developments of the following algorithms:

1) Algorithms of the identification of the zones (groups of records) which should be masked with different SDL methods.
2) SDL Technique which can be applied to positive variables and also to the variables than can take positive as well as negative values.
3) Combinations of SDL methods which would offer adequate protection for the records in different zones, including methods for outlier's protection.

Acknowledgment

Authors would like to thank Gurgen Khachatryan and Anna Oganyan for many important remarks and valuable discussions.

Bibliography

[Agrawal, 00] Agrawal, R. and Srikant, R. "Privacy-Preserving Data Mining," Proc. ACM SIGMOD Conf. Management of Data, ACM Press, 2000, pp. 439–450; http://doi.acm.org/ 10.1145/342009.335438.

[Aslan, 04] L. Aslanyan, V. Topchyan, Hierarchical Cluster Analysis For Partially Synthetic Data Generation, Transactions of IIAP of NAS of RA, Mathematical Problems of Computer Science, submitted, 2013.

[Aslan, 09] Aslanyan, L. and Sahakyan, H. "Chain Split and Computations in Practical Rule Mining", International Book Series, Information Science and Computing, Book 8, Classification, forecasting, Data Mining, pp. 132-135, 2009.

[Aslan, 13] L. H. Aslanyan, H. E. Danoyan, "On the optimality of a hash-coding type search algorithm", Proceedings of the 9th conference CSIT, Yerevan, Armenia, pp. 55-57, 2013

[Brickell, 87] E. F. Brickell and Y. Yacobi, On Privacy Homomorphisms (Extended Abstract), Advances in Cryptology - EUROCRYPT '87, Workshop on the Theory and Application of Cryptographic Techniques, Amsterdam, The Netherlands, April 13-15, 1987, pp. 117-125.

[Cohen, 93] H. Cohen, A Course in Computational Algebraic Number Theory, Springer, 1993, 580 p.

[Dalenius, 82] Dalenius, T. and Reiss, S. P. "Data-swapping: A technique for disclosure control", Journal of Statistical Planning and Inference, volume 6, pages 7385, 1982.

[Defays, 93] Defays, D. and Nanopoulos, P. "Panels of enterprises and confidentiality: the small aggregates method", in Proceedings of the 92 Symposium on Design and Analysis of Longitudinal Surveys, pages 195204, Ottawa. Statistics Canada, 1993

[Duncan, 91] Duncan, G. T. and Pearson, R. W. Enhancing access to microdata while protecting confidentiality: Prospects for the future. Statistical Science, 6:219239, 1991

[Feinberg, 06] Fienberg, S. "Privacy and Confidentiality in an e-Commerce World: Data Mining, Data Warehousing, Matching and Disclosure Limitation", Statistical Science, 21, 143-154, 2006

[Ferrer, 01a] Domingo-Ferrer, J., Mateo-Sanz, J. M., and Torra, V. "Comparing SDC methods for microdata on the basis of information loss and disclosure risk", In Proc ETKNTTS 2001, pages 807 - 825, Luxembouorg. Eurostat, 2001

[Ferrer, 01b] Domingo-Ferrer, J. and Torra, V. "A quantitative comparison of disclosure control methods for microdata." In Doyle, P., Lane, J., Theeuwes, J., and Zayatz, L., editors, Confidentiality, Disclosure and Data Access, pages 111133. NorthHolland, Amsterdam, 2001.

[Ferrer, 96] J. D. Ferrer, “A new privacy homomorphism and applications,” Information Processing Letters, vol. 60, no. 5, pp. 277-282, 1996

[Fontaine, 07] C. Fontaine and F. Galand, A Survey of Homomorphic Encryption for Nonspecialists, EURASIP Journal on Information Security 2007, pp. 1-10.

[Greaig, 09] Greaig Gentry, A Fully Homomorphic Encryption Scheme, 2009

[Khan, 12] Z. Khan, Quasi-Linear Time Fully Homomorphic Public Key Encryption Algorithm (ZK111), Journal of Theoretical Physics & Cryptography, vol. 1, November 2012, pp. 14-17.

[Kim, 86] Kim, J. J., A method for limiting disclosure in microdata based on random noise and transformation. In Proceedings of the ASA Section on Survey Research Methodology, pages 303-308. Alexandria V.A. American Statistical Association, 1986

[Little, 93] Little, R. J. A. "Statistical analysis of masked data", in Journal of Ofcial Statistics, 9:407426, 1993.

[Loureiro, 04] Loureiro, A., Torgo, L. and Soares, C., "Outlier Detection Using Clustering Methods: a data cleaning application" , Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector, 2004

[Miyoung, 13] Miyoung Jang, Min Yoon, and Jae-Woo Chang A k-Nearest Neighbor Search Algorithm for Privacy Preservation in Outsourced Spatial Databases, ISA 2013, ASTL Vol. 21, pp. 223 - 226, 2013.

[Moore, 96] Moore, R. "Controlled data swapping techniques for masking public use microdata sets." U. S. Census Bureau, 1996

[Pohling, 78] Stephen C. Pohling and Martin E. Hellman, An Improved algorithm for computing Logarithms over GF (p) and its cryptographic significance, IEEE trans. on Inf. th. Vol. IT-24, No. 1 Jan. 1978. pp. 160-110.

[Pollard, 74] J. M. Pollard, ‘‘Theorems on factorization and primality testing”, Roc. Cambridge Philos. SOC. vol. 76, 1974. pp. 521-528.

[Reiter, 02] Reiter, J. P., "Satisfying disclosure restrictions with synthetic data sets," Journal of Official Statistics 18 (2002) 531-544.

[Reiter, 05] Reiter, J. P., "Using CART to generate partially synthetic public use microdata.", in Journal of Official Statistics, 21, 441 - 462, 2005.

[Rivest, 78] Ronald L. Rivest , Len Adleman, Michael L. Dertouzous, On data Banks And Privacy Homomorphisms, in : R.A. DeMillo et al., eds, Foundations of Secure Computation(Academic Press, New York, 1978) 169-179.

[Sanz, 99] Mateo-Sanz J.M. and Domingo-Ferrer J, "A method for data-oriented multivariate microaggregation", in Proceedings of Statistical Data Protection'98, Luxembourg: Office for Official Publications of the European Communities, pp. 89-99, 1999.

[Tendik, 94] Tendik, P. and Matlof, N. "A modified random perturbation method for database security", ACM Transactions on Database Systems, 19(1):4763, 1994.

[Tonoyan, 76] Tonoyan, G. "Chain decomposition of n dimensional unit cube and reconstruction of monotone Boolean functions", JVM and F, v. 19, No. 6, 1532-1542, 1976

[Wallman, 04] Wallman, K. K. and Harris-Kojetin, B. A. "Implementing the confidential information protection and statistical efficiency act of 2002", Chance, 17(3):2125, 2004.

Authors' Information

Levon Aslanyan *– ITHEA ISS, Sofia, Bulgaria; Institute for informatics and automation problems of NAS RA, 1, P. Sevak street, Yerevan 0014, Armenia, e-mail: lasl@sci.am*

Major Fields of Scientific Research: Discrete optimization, Artificial intelligence, NLP, WSN, Privacy preserved computation

Vardan Topchyan *– Institute for informatics and automation problems of NAS RA, 1, P. Sevak street, Yerevan 0014, Armenia, e-mail:vardan.topchyan@gmail.com*

Major Fields of Scientific Research: Decision models, Homomorphic encryption, Privacy preserved computation

Haykaz Danoyan *– Institute for informatics and automation problems of NAS RA, 1, P. Sevak street, Yerevan 0014, Armenia, e-mail: hed@ipia.sci.am*

Major Fields of Scientific Research: NN Search, Discrete optimization, Coding theory, Homomorphic encryption