PRIVACY HOMOMORPHISMS FOR SUBCONTRACTING STATISTICAL COMPUTATION

Josep Domingo i Ferrer

Universitat Rovira i Virgili*†
Tarragona, Catalonia

Abstract

When publishing data containing official statistics, a need to preserve statistical confidentiality arises. Statistical disclosure of individuals' data must be prevented. There is a wide choice of techniques to achieve this anonymization: data perturbation, data suppression, etc. In this paper, we tackle the problem of using anonymized data to compute exact statistics; the goal is for a classified level (statistical institute) to be able to retrieve statistics computed by an unclassified level (external contractor) on disclosure-protected macrodata. Our approach is based on privacy homomorphisms, especially on a recent one.

1. INTRODUCTION

Statistical institutes that handle confidential data must guarantee statistical confidentiality, *i. e.* that published statistical data do not disclose individual information. Published data may be macrodata (*i. e.* contingency tables) or microdata (individual records). There is a wide choice of techniques to achieve anonymization of published data: cell suppression, cell perturbation, etc. (Schackis, 1993). *The underlying principle is that exact data are only available at a classified level (the institute); the unclassified level (rest of organizations or individuals) only see anonymized data or maybe just encrypted data.*

The above principle meets the requirements of statistical confidentiality, but seems to preclude subcontracting statistical computation. Notice the insecurity of the common practice by which any external contractor is labeled as "classified" on the only ground that he has signed a written agreement to preserve statistical confidentiality. If the institute cannot or does not wish to do all computation in its own facilities, it should be possible to have the job partly done by external contractors *without assuming that these are classified*, that is, without giving them exact sensitive data.

In this paper, we tackle the problem of subcontracting statistical computation in two scenarios

*Autovia de Salou, s/n, E-43006 Tarragona, Catalonia, Spain. E-mail jdomingo@etse.urv.es

†This work is partly supported by the Spanish CICYT under grant no. TIC95-0903-C02-02.

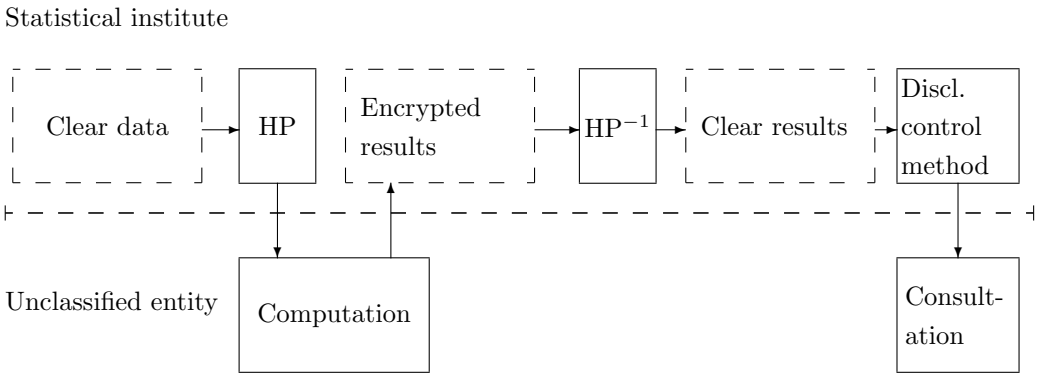

Figure 1: **Scenario A**

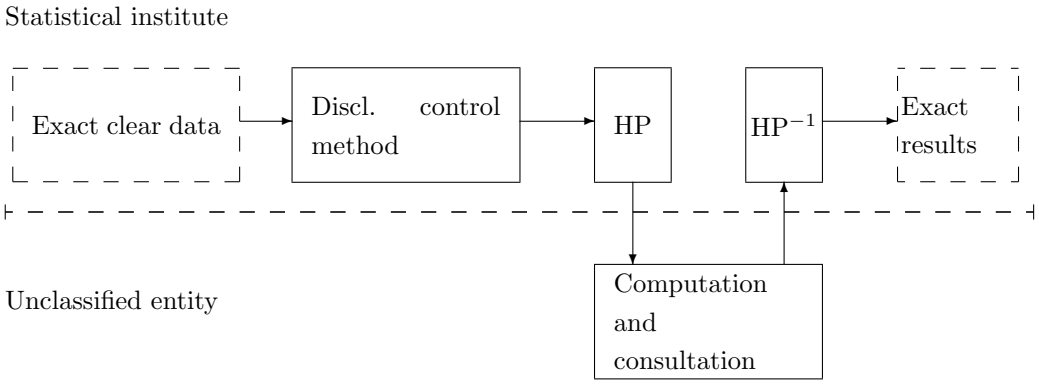

Figure 2: **Scenario B**

Scenario A: Sensitive data are encrypted by the statistical institute and are subsequently forwarded to the external contractor for computation of encrypted statistics or aggregate data. The results can later be decrypted by the statistical institute, who may decide to publish them after taking the proper disclosure control cautions.

Scenario B: Sensitive data are fed to a disclosure control procedure by the statistical institute, who also encrypts some output information generated by the method. The external contractor receives as inputs to its computation the anonymized data and related encrypted information. The contractor returns the output of its computation to the statistical institute. The statistical institute *alone* can extract (with little effort) *exact* results from the output received from the external contractor.

In both scenarios above just a "small core" of trusted resources (computers and manpower) is needed for classified tasks at a statistical office. The encryption tranformations used must have special homomorphic properties, *i. e.* it must be possible to perform some arithmetical and/or logical operations directly on encrypted data. In section 2, we give some background on such encryption algorithms, which are known as privacy homomorphisms (PHs for short). Once the concept of PH is understood, scenario A is quite straightforward and needs no further explanation: basically, the external contractor works on encrypted data all the time and this is possible thanks to the homomorphic properties of PHs. Of course, if the PH used preserves

a set F of operations then the contractor will be only able to use operations in F during his computations. On the other hand, scenario B makes use of PHs in more subtle ways which depend on the disclosure-control method being used. This scenario is more realistic because it assumes that the contractor (as any unclassified entity) has access to the anonymized data. Scenario B is discussed in combination with random perturbation (section 3) and data suppression methods (section 4). Section 5 is a conclusion.

2. BACKGROUND: PRIVACY HOMOMORPHISMS

Computation on encrypted data does not make sense unless the encryption transformation being used has some homomorphic properties. Privacy homomorphisms (PHs from now on) were formally introduced in Rivest *et al.* (1978b) as a tool for processing encrypted data. Basically, they are encryption functions $E_k : T \longrightarrow T'$ which allow to perform a set F' of operations on encrypted data without knowledge of the decryption function D_k. Knowledge of D_k allows to recover the outcome of the corresponding set F of operations on clear data. The security gain is especially apparent in multilevel security scenarios such as the ones described in section 1. Data can be encrypted by the classified level (institute), be processed by the unclassified level (contractor), and the result be decrypted by the classified level.

Next follow some well-known results about PHs. If a PH preserves order, then it is insecure against a ciphertext-only attack (can be decrypted by merely observing encrypted data). If a PH has addition among its ciphertext-domain operations, then it is insecure under chosen-ciphertext attack (in which the attacker can obtain the encrypted versions of whatever cleartexts she chooses, see Ahituv *et al.*, 1987). With the exception of the RSA algorithm —which preserves only multiplication—, all of the examples proposed in Rivest *et al.* (1978b) were subsequently shown to be breakable by a ciphertext-only attack or, at most, a known-cleartext attack (in which the attacker knows some random cleartext-ciphertext pairs, see Brickell and Yacobi, 1988). Brickell and Yacobi introduced R-additive PHs which remain secure at the cost of putting a restriction on the number of ciphertexts that can be added together. Lacking secure PHs that preserve more than one operation, most successful attempts at encrypted data processing have traditionally relied on *ad-hoc* procedures (Ahituv *et al.*, 1987, Trouessin, 1991). In Domingo (1996), we presented a new PH that preserves addition and multiplication and has the remarkable property of resisting known-cleartext attacks.

For illustration purposes, we give two examples of privacy homomorphisms, each having interesting properties in its own right

Example 1 An exponential cipher such as RSA (Rivest *et al.*, 1978a) is a PH. Let $m = pq$, where p and q are two large secret primes (about 100 decimal digits each). In this case,

$$T = T' = \mathbf{Z}_m$$

$$E_k(a) = a^e \bmod m$$

$$D_k(a') = (a')^d \bmod m$$

where $\mathbf{Z}_m$ is the set of integers modulo m, d is secret and $ed \bmod \phi(m) = 1$, with $\phi(m) = (p-1)(q-1)$ being Euler's totient function. Clearly,

$$D_k(E_k(a)) = a^{ed} \bmod m = a^{1+t\phi(m)} \bmod m = a$$

where Euler's theorem is used in the last step. Now, let $F = F' = \{\star\}$, where $\star$ denotes the modular multiplication over $\mathbf{Z}_m$. The following homomorphic property holds

$$E_k(a) \star E_k(b) = (a^e \bmod m)(b^e \bmod m) \bmod m = (a \star b)^e \bmod m = E_k(a \star b)$$

This homomorphism allows only one operation, but appears to be very secure. Finding D_k from E_k, *i. e.* finding d from e, seems to be equivalent to factoring a large modulus m — no polynomial-time algorithm for factoring has been published up-to-date—. An additional interesting property relates to the preservation of the equality predicate, because it holds that

$$E_k(a) = E_k(b) \text{ if and only if } a = b$$

To summarize, the RSA PH has the following features

- The only operation that can be carried out on encrypted data by the contractor is multiplication.
- The equality predicate is preserved, and thus comparisons for equality can be done by the contractor based on encrypted data.
- Breaking the security of the PH even by a chosen-plaintext attack seems hard.
- Cleartext and ciphertext lengths are about the same, so there is no storage penalty for keeping data in encrypted form.

□

Example 2 The PH in this example was recently proposed by this author in Domingo (1996). Let m be a large secret integer such that $m = pq$, where p and q are two secret primes. In this case the set of cleartext is $T = \mathbf{Z}_m$. The set of ciphertext is $T' = (\mathbf{Z}_p \times \mathbf{Z}_q)^n$, with n being a security parameter. The set F of cleartext operations is formed by addition, subtraction and multiplication in T. The set F' of ciphertext operations contains the corresponding componentwise operations in T'. The PH can be described as

Public parameter n

Secret key p, q large primes such that $pq = m$. The product m is kept secret. Also, $r_p \in \mathbf{Z}_p$, such that it generates a large multiplicative subgroup in $\mathbf{Z}_p - \{0\}$. Also, r_q with similar properties with respect to $\mathbf{Z}_q$.

Encryption Randomly split $a \in \mathbf{Z}_m$ into secret $a_{.1}, \cdots, a_{.n}$ such that $a = \sum_{j=1}^{n} a_{.j} \bmod m$ and $a_{.j} \in \mathbf{Z}_m$. Compute

$$E_k(a) = ([a_{.1}r_p \bmod p, a_{.1}r_q \bmod q], [a_{.2}r_p^2 \bmod p, a_{.2}r_q^2 \bmod q], \cdots \\ \cdots, [a_{.n}r_p^n \bmod p, a_{.n}r_q^n \bmod q]) \quad (1)$$

Decryption Compute the scalar product of the j-th [mod p,mod q] pair by $[r_p^{-j} \bmod p, r_q^{-j} \bmod q]$ to retrieve the $[a_{.j} \bmod p, a_{.j} \bmod q]$. Add up to get $[a \bmod p, a \bmod q]$. Use the Chinese remainder theorem to get $a \bmod m$.

As encrypted values are computed over $(\mathbf{Z} \times \mathbf{Z})^n$ by the contractor, the use of r_p and r_q requires that the terms of the encrypted value having different r-degree be handled separately —the r-degree of a mod p, resp. mod q, term is the exponent of the power of r_p, resp. r_q, contained in the term—. At the time of decryption, this is necessary for the institute to be able to multiply each term by r_p^{-1} (inverse of r_p over $\mathbf{Z}_p$) and r_q^{-1} (inverse of r_q over $\mathbf{Z}_q$) the right number of times, before adding all terms up, reducing the final result into $\mathbf{Z}_p \times \mathbf{Z}_q$, and decrypting into $\mathbf{Z}_m$.

The only operation that alters the r-degree is multiplication. If cleartext data x and y have been encrypted as $E_k(x)$ and $E_k(y)$, with r-degrees n_1 and n_2, then the product $E_k(z) = E_k(x)E_k(y)$ has r-degree $n = n_1 + n_2$. The result may have terms $E_{k,j}(z)$ with degrees ranging from $j = 1$ to n and can be represented in vector notation as

$$(E_{k,1}(z), E_{k,2}(z), \cdots, E_{k,n}(z))$$

If we set $\mathbf{r} = [r_p, r_q]$ then $E_k(z)$ can also be written as a polynomial

$$E_k(z)[\mathbf{r}] = t_1\mathbf{r} + \cdots + t_n\mathbf{r}^n$$

Although the coefficients t_j are in practice unknown to the contractor, the polynomial notation is useful to understand how algebraic operations should be carried out by the contractor using terms $E_{k,j}(z)$ rather than coefficients t_j

Addition and subtraction In vector notation, they are done componentwise over $\mathbf{Z}$, which in polynomial notation means adding terms with the same degree.

Multiplication It works like in the case of polynomials: all terms are cross-multiplied in $\mathbf{Z}$, with an j_1-th degree term by a j_2-th degree term yielding a $j_1 + j_2$-th degree term; finally, terms having the same degree are added up.

Division Cannot be carried out in general because the polynomials are a ring, but not a field. A good solution is to leave divisions in rational format by considering the field of rational functions, *i. e.* fractions whose numerator and denominator are polynomials. In this way, if a and b are two integers, we encrypt a/b as $\frac{E_k(a)}{E_k(b)}$.

Two remarks about fraction handling

1. When addition or subtraction are performed on fractions with different denominators, numerators cannot be added or subtracted directly. The rules for ordinary fractions should be followed

$$\frac{E_k(a)}{E_k(b)} \pm \frac{E_k(c)}{E_k(d)} = \frac{E_k(a)E_k(d) \pm E_k(b)E_k(c)}{E_k(b)E_k(d)}$$

2. If noninteger initial data are dealt with as fractions, then every result received from the contractor is a fraction; the numerator of the exact result must be decrypted and thereafter divided over the real numbers by the decrypted denominator, in order to get the right number of decimal positions.

To summarize, this PH has the following features

- Addition, subtraction, multiplication and division can be carried out on encrypted data by the contractor.

- The equality predicate is not preserved, and thus comparisons for equality cannot be done by the contractor based on encrypted data. Remark that a given cleartext can have many ciphertext versions for two reasons: A) random splitting during encryption; B) the contractor computes over $(\mathbf{Z} \times \mathbf{Z})^n$ and only the institute can perform a reduction to $\mathbf{Z}_p \times \mathbf{Z}_q$ during decryption.
- Breaking the security of the PH by a known-plaintext attack seems hard, provided that $n > 1$ is chosen (see Domingo, 1996).
- A ciphertext is about n times longer than the corresponding cleartext. Even if this is a storage penalty, a choice of $n = 2$ should be affordable while remaining secure.

□

3. SCENARIO B WITH PERTURBED DATA

In this section, we show how to implement scenario B when the disclosure-control method being used is based on data perturbation (see Adam and Wortmann, 1989 or Schackis, 1993 for an overview of such methods). The contractor works on perturbed data and the institute can obtain with little effort exact results from computations performed by the contractor: restoration of the exact result involves only decrypting the perturbation of the unclassified result.

Assume that, at a classified level, the statistical institute uses a random perturbation method on sensitive data $x_1, x_2, \cdots, x_n$. This gives $x_i^* = x_i + \varepsilon_i$, $\forall i = 1, \cdots, n$ where ε_i is a random value. Let E_k be the encrypting transformation of a PH with a set F of cleartext operations and a corresponding set F' of ciphertext operations. Now, the institute releases the pairs $(x_i^*, E_k(-\varepsilon_i))$ to the contractor for further computation. The contractor is able to perform on the x_i^* the operations in F and compute the encrypted perturbation of the result by using the operations in F' on the $E_k(-\varepsilon_i)$.

For instance, if we focus on the PH of example 2 then F and F' *include the elementary arithmetical operations, thus being suited for statistical computation.* Table 1 summarizes the computations on encrypted perturbations generated by elementary operations. Given that $x = x^* - \varepsilon_x$ and $y = y^* - \varepsilon_y$, deriving the perturbations for addition, subtraction and multiplication is straightforward. Division is not explicitly considered, because in the PH of example 2 it is avoided by representing and handling rational numbers as fractions. When the institute receives the result of a computation from the contractor, the institute decrypts the perturbation and adds the clear perturbation to the perturbed result to obtain the exact result. If perturbations of initial data were fractions then every perturbation received from the contractor is a fraction; the numerator of the perturbation must be decrypted and thereafter divided over the real numbers by the decrypted denominator, in order to get the right number of decimal positions.

4. SCENARIO B WITH PARTIALLY SUPPRESSED MACRODATA

The following disclosure protected table is taken from Cox (1993). Disclosure —sensitive— cells have been suppressed (primary suppressions); other cells have been secondarily suppressed to prevent inferences. A D-cell stands for a suppressed cell.

Perturbed operation	Clear perturbation	Encrypted perturbation
$x^* + y^*$	$-(\varepsilon_x + \varepsilon_y)$	$E_k(-\varepsilon_x) + E_k(-\varepsilon_y)$
$x^* - y^*$	$-(\varepsilon_x - \varepsilon_y)$	$E_k(-\varepsilon_x) - E_k(-\varepsilon_y)$
$x^* y^*$	$-x^*\varepsilon_y - y^*\varepsilon_x + \varepsilon_x\varepsilon_y$	$x^* E_k(-\varepsilon_y) + y^* E_k(-\varepsilon_x) + E_k(-\varepsilon_x)E_k(-\varepsilon_y)$

Table 1: **Encrypted perturbations corresponding to elementary operations**

D	10	D	D	20	80
D	10	D	5	15	60
40	10	D	D	10	90
5	5	D	D	5	40
75	35	65	45	50	270

Now, an alternative approach to cell suppression is to use a PH to encrypt the D-cell values

$E_k(10)$	10	$E_k(25)$	$E_k(15)$	20	80
$E_k(20)$	10	$E_k(10)$	5	15	60
40	10	$E_k(20)$	$E_k(10)$	10	90
5	5	$E_k(10)$	$E_k(15)$	5	40
75	35	65	45	50	270

The contractor views the encrypted cells as if they had been suppressed, but it can perform on them the operations in F'. The institute can take advantage of this work by decrypting the result. Using a PH breakable by a known-cleartext attack (such as most PHs proposed in Rivest *et al.*, 1978b) is not practical, because the contractor cannot be given the encrypted versions of the non-suppressed cells. This means that the contractor should operate separately on encrypted suppressed cells (operations in F') and on clear non-suppressed cells (operations in F); the difficult task of merging the results of both computational streams would be left to the institute. The use of a PH resistant to a known-cleartext attack (such as the ones in examples 1 and 2) allows the encrypted version of the whole table to be safely distributed by the institute, along with the non-suppressed cells as cleartext; thus the unclassified level can operate on the whole encrypted table using the operations in F', so that the only job left to the classified level is decryption of the final result.

5. CONCLUSION

We have presented two solutions to the problem of subcontracting statistical computation on sensitive data. Both solutions rely on homomorphic encryption and do not need to assume that the external contractor is classified. Data handled by the contractor are disclosure-protected or encrypted. A "small core" of classified resources suffices to recover exact results from the

outcome of the computations delivered by the contractor. The approach has some inherent limitations (*e. g.* the nature of the operations that can be carried out by the contractor is restricted by the privacy homomorphism used), but is a step ahead in the process of *enforcing* legal confidentiality requirements by technical means.

REFERENCES

Adam, N. R., and Wortmann, J. C. (1989) "Security-control methods for statistical databases: a comparative study", *ACM Computing Surveys* **21**, pp. 515-556.

Ahituv, N., Lapid, Y., and Neumann, S. (1987) "Processing encrypted data", *Communications of the ACM* **20**, pp. 777-780.

Brickell, E., and Yacobi, Y. (1988) "On privacy homomorphisms", in W. L. Price and D. Chaum (Eds.), *Advances in Cryptology-Eurocrypt'87*, Berlin: Springer-Verlag, pp. 117-125.

Cox, L. H. (1993) "Solving confidentiality protection problems: a model for cell suppresion in U.S. economic censuses", in D. Lievesley (Ed.), *Proceedings of the First International Seminar on Statistical Confidentiality*, Luxembourg: Eurostat, pp. 229-245.

Domingo, J. (1996) "A new privacy homomorphism and applications", *Information Processing Letters* (to appear).

Rivest, R. L., Shamir, A., and Adleman, L. (1978a) "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM* **21**, pp. 120-126.

Rivest, R. L., Adleman, L., and Dertouzos, M. L. (1978b) "On data banks and privacy homomorphisms", in R. A. DeMillo *et al.* (Eds.), *Foundations of Secure Computation*, New York: Academic Press, pp. 169-179.

Schackis, D. (1993) *Manual on Disclosure Control Methods*, Luxembourg: Eurostat.

Trouessin, G. (1991) *Traitements Fiables de Données Confidentielles par Fragmentation-Rédondance-Dissémination*, Toulouse: Univ. Paul Sabatier (Ph. D. Thesis).