ELSEVIER

Information Processing Letters 60 (1996) 277-282

Information Processing Letters

A new privacy homomorphism and applications

Josep Domingo i Ferrer[1]

Estadística i Investigació Operativa, Dep. Enginyeria Química, Universitat Rovira i Virgili,
Autovia de Salou s/n., E-43006 Tarragona, Catalonia, Spain

Received 2 October 1995; revised 30 January 1996
Communicated by L. Boasson

Abstract

An additive and multiplicative privacy homomorphism is an encryption function mapping addition and multiplication of cleartext data into two operations on encrypted data. One such privacy homomorphism is introduced which has the novel property of seeming secure against a known-cleartext attack. An application to multilevel statistical computation is presented, namely classified retrieval of exact statistics from unclassified computation on disclosure-protected (perturbed) data.

Keywords: Privacy homomorphisms; Statistical confidentiality; Safety/security in digital systems

1. Introduction

Privacy homomorphisms (PHs from now on) were formally introduced in [5] as a tool for processing encrypted data. Basically, they are encryption functions $E_k : T \rightarrow T'$ 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 a multilevel security environment: data can be encrypted at a classified level, be processed by an unclassified computing facility (external contractor), and the result be decrypted by the classified level. For illustration, consider the following example of PH, given in [5]

Example 1. Let p and q be two large secret primes. Consider the set of cleartext data $T = \mathbb{Z}_m$ and the set of cleartext operations $F = \{+_m, -_m, \times_m\}$ consisting, respectively, of the addition, subtraction and multiplication modulo m, with $m = pq$. Let the ciphertext data set be $T' = \mathbb{Z}_p \times \mathbb{Z}_q$. Ciphertext operations F' are the componentwise version of those in F. Define the encryption key $k = (p, q)$ and $E_k(a) = [a \bmod p, a \bmod q]$. Given $k = (p, q)$, $D_k([b, c])$ is computed using the Chinese remainder theorem.

Unfortunately, it is shown in [2] that this PH can be broken – i.e. p and q can be discovered – using a known-cleartext attack.

Next follow some well-known results about PHs. If a PH preserves order, then it is insecure against a ciphertext-only attack. If a PH has addition among its

[1] This work is partly supported by the Spanish CICYT under grant no. TIC95-0903-C02-02. Current address: Escola Tecnica Superior d'Enginyera, Universitat Rovira i Virgili, Carretera de Salou s/n., E-43006 Tarragona, Catalonia, Spain. Email: jdomingo@etse.urv.es.

PII S0020-0190(96)00170-6

ciphertext-domain operations, then it is insecure under chosen ciphertext attack [1]. With the exception of the RSA algorithm - which preserves only multiplication - all of the examples proposed in [5] were subsequently shown to be breakable by a ciphertext-only attack or, at most, a known-cleartext attack (see [2]); the authors of [2] 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, successful attempts at encrypted data processing have rather been based on ad-hoc procedures [1,7]. We present in this paper a new privacy homomorphism which preserves addition and multiplication, and has the remarkable property of seeming able to withstand a known-cleartext attack.

In Section 2 the new PH is motivated and specified, and its security is analyzed thereafter. Section 3 concludes by showing an application that allows recovery of exact statistics at a classified level from unclassified computation on disclosure-protected data (e.g. perturbed data): the way to subcontracting statistical computation is thus opened.

2. A new privacy homomorphism

We propose in this section a new privacy homomorphism which is similar to the one of Example 1 (it has the same sets T, T', F and F'), but entails two significant improvements

- Small values are nontrivially encrypted.
- The new PH is able to withstand a known-cleartext attack.

When p, q, $m = pq$ are very large integers, a small value a is very likely to have the same representation over $\mathbb{Z}_m$, $\mathbb{Z}_p$ and $\mathbb{Z}_q$, that is $a \bmod m = a \bmod p = a \bmod q$ if $a < \min(p,q)$. This is an undesirable feature, because the homomorphism of Example 1 leaves the cleartext unencrypted (trivial ciphertext). A possible solution is to multiply a by a pair of secret constants r_p and r_q such that $r_p < p$ and $r_q < q$ (the encryption key is now extended to $k = (p, q, r_p, r_q)$). A further improvement to deter ciphertext-only attacks based on frequency analysis is to secretly and randomly split a into $a_{.1}, \ldots, a_{.n}$, such that $a_{.j} \in \mathbb{Z}_m$ and $\sum_{j=1}^{n} a_{.j} \bmod m = a$. Thus the privacy homomorphism we propose is:

- *Public parameters.* n and m (actually m can be made secret, to increase security).
- *Secret key.* p and q large primes, such that $pq = m$. Also, $r_p \in \mathbb{Z}_p$, such that it generates a large multiplicative subgroup in $\mathbb{Z}_p - \{0\}$. Also, r_q with similar properties with respect to $\mathbb{Z}_q$.
- *Encryption.* Randomly split $a \in \mathbb{Z}_m$ into secret $a_{.1}, \cdots, a_{.n}$ such that $a = \sum_{j=1}^{n} a_{.j} \bmod m$ and $a_{.j} \in \mathbb{Z}_m$. Compute

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

- *Decryption.* Compute the scalar product of the jth $[\bmod p, \bmod q]$ 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 $(\mathbb{Z} \times \mathbb{Z})^n$ by the unclassified level, 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 $\bmod p$, respectively $\bmod q$, term is the exponent of the power of r_p, respectively r_q, contained in the term. This is necessary for the classified level to be able to multiply each term by r_p^{-1} (inverse of r_p over $\mathbb{Z}_p$) and r_q^{-1} (inverse of r_q over $\mathbb{Z}_q$) the right number of times, before adding all terms up, reducing the final result into $\mathbb{Z}_p \times \mathbb{Z}_q$, and decrypting into $\mathbb{Z}_m$.

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

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

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

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

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

- *Addition and subtraction.* In vector notation, they are done componentwise over $\mathbb{Z}$, 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 $\mathbb{Z}$, with a j_1th degree term by a j_2th 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

$$\frac{E_k(a)}{E_k(b)}.$$

Note 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

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

Note 2. If noninteger initial data are dealt with as fractions, then every result received from the unclassified level is a fraction; the numerator of the exact result must be decrypted and thereafter divided over the real numbers by the decrypted denominator (be it a power of 10 or not), in order to get the right number of decimal positions.

We next give one numerical example to illustrate computation with the proposed PH.

Example 2. This example is ridiculously small, but it illustrates the computation of a formula including two additions and one multiplication, i.e. $(x_1+x_2+x_3)x_4$. For brevity and clarity, take $n = 2$, that is, cleartexts are split into two parts during encryption.

Classified level. Let $p = 17$, $q = 13$, $r_p = 2$ and $r_q = 3$ be the secret key. Let $(x_1, x_2, x_3, x_4) = (-0.1, 0.3, 0.1, 2)$. In order to suppress decimal positions, multiply these data by 10, thus getting the fractions

$$x_1 = \frac{\tilde{x}_1}{10} = \frac{-1}{10}, \quad x_2 = \frac{\tilde{x}_2}{10} = \frac{3}{10},$$
$$x_3 = \frac{\tilde{x}_3}{10} = \frac{1}{10} \quad \text{and} \quad x_4 = \frac{\tilde{x}_4}{1} = \frac{2}{1}.$$

Numerators are secretly and randomly split and then transformed according to the proposed PH, thus obtaining first and second r-degree terms

$$E_k(\tilde{x}_1) = E_k(-1) = E_k(2,-3) = ([4,6],[5,12]),$$
$$E_k(\tilde{x}_2) = E_k(3) = E_k(2,1) = ([4,6],[4,9]),$$
$$E_k(\tilde{x}_3) = E_k(1) = E_k(4,-3) = ([8,12],[5,12]),$$
$$E_k(\tilde{x}_4) = E_k(2) = E_k(3,-1) = ([6,9],[13,4]).$$

The encrypted data are forwarded to the unclassified level, along with their denominators: $(1,1)$ for $E_k(\tilde{x}_4)$ and $(10,10)$ for the rest of data.

Unclassified level. First, compute the sum by directly adding the numerators of fractions, because the denominator is 10 in all data

$$\begin{aligned}\sum_{i=1}^{3} E_k(\tilde{x}_i) &= ([4+4+8, 6+6+12], \\ &\qquad [5+4+5, 12+9+12]) \\ &= ([16,24],[14,33]).\end{aligned}$$

The denominator of the sum is obviously $(10,10)$. Next, multiply by $E_k(\tilde{x}_4)$:

$$\begin{aligned}&(E_k(\tilde{x}_1) + E_k(\tilde{x}_2) + E_k(\tilde{x}_3))E_k(\tilde{x}_4) \\ &\quad = ([16,24],[14,33]) \times ([6,9],[13,4]) \\ &\quad = ([0,0],[16 \times 6, 24 \times 9], \\ &\qquad [16 \times 13 + 14 \times 6, 24 \times 4 + 33 \times 9], \\ &\qquad [14 \times 13, 33 \times 4]) \\ &\quad = ([0,0],[96,216],[292,393],[182,132]).\end{aligned}$$

Thus, the numerator of the result has terms up to the fourth r-degree. The denominator of the product is $10 \times 1 = 10$ for all terms. Return both numerator and denominator to the classified level.

Classified level. Compute

$$
\begin{aligned}
&([0 \times r_p^{-1} \bmod p, 0 \times r_q^{-1} \bmod q],\\
&\ [96 \times r_p^{-2} \bmod p, 216 \times r_q^{-2} \bmod q],\\
&\ [292 \times r_p^{-3} \bmod p, 393 \times r_q^{-3} \bmod q],\\
&\ [182 \times r_p^{-4} \bmod p, 132 \times r_q^{-4} \bmod q])\\
&\quad = ([0 \times 9 \bmod 17, 0 \times 9 \bmod 13],\\
&\qquad [96 \times 9^2 \bmod 17, 216 \times 9^2 \bmod 13],\\
&\qquad [292 \times 9^3 \bmod 17, 393 \times 9^3 \bmod 13],\\
&\qquad [182 \times 9^4 \bmod 17, 132 \times 9^4 \bmod 13])\\
&\quad = ([0,0],[7,11],[11,3],[5,5]) = ([6,6]),
\end{aligned}
$$

where, in the last step, all terms have been added up over $\mathbb{Z}_p \times \mathbb{Z}_q$. Now use the Chinese remainder theorem on the pair $[6,6]$ to recover the $(\tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3)\tilde{x}_4 = 6 \bmod pq = 6$. Finally, divide 6 by the denominator 10 returned by the unclassified level, so that the final result is $(x_1 + x_2 + x_3)x_4 = 0.6$.

2.1. Security analysis

If factoring is hard, we argue that the PH whose encryption function is specified by Eq. (1) cannot be broken in general by a known-cleartext attack in which the cryptanalyst knows some *random* cleartext-ciphertext pairs. We next justify this assertion, assuming that m is public (worst case).

Given cleartexts a_i, for $i = 1, \ldots, l$, ciphertexts may have several terms of different r-degrees. Let n be the maximum r-degree of ciphertexts in known pairs. Then, from the definition of the PH, the following equalities can be written:

$$
\begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{l1} & \cdots & b_{ln} \end{bmatrix} + \begin{bmatrix} n_{11} & \cdots & n_{1n} \\ \vdots & \ddots & \vdots \\ n_{l1} & \cdots & n_{ln} \end{bmatrix} p
= \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{l1} & \cdots & a_{ln} \end{bmatrix} \begin{bmatrix} r_p & 0 & \cdots & 0 \\ 0 & r_p^2 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & r_p^n \end{bmatrix}, \qquad (2)
$$

$$
\sum_{j=1}^{n} a_{ij} \equiv a_i \pmod m, \quad 1 \leqslant i \leqslant l. \qquad (3)
$$

An equality analogous to (2) can be written for the q, r_q and c_{ij} (the mod q ciphertexts). Remark that, from the cryptanalyst's point of view, the system formed by the $nl + l$ equations (2) and (3) has $2nl + 2$ unknowns, namely the a_{ij}, the n_{ij} and also p and r_p. Thus the system has $(n - 1)l + 2$ degrees of freedom, being completely undetermined. Adding the matrix equality for c_{ij}, q and r_q is of no use, since it means nl new equations and $nl + 2$ new unknowns, thus increasing the number of degrees of freedom by two.

Nevertheless, if the cryptanalyst factors m by finding p or q in some way, then the system 2 can be transformed into a system of congruences

$$
qb_{ij} \equiv qa_{ij}r_p^j \pmod m, \quad 1 \leqslant i \leqslant l, \ 1 \leqslant j \leqslant n. \qquad (4)
$$

The above equations, together with equations (3) yield a set of $nl + l$ congruences having $nl + 1$ unknowns (namely r_p and the a_{ij}). If $l \geqslant n$ then the powers of r_p can be dealt with as independent unknowns, so that the resulting system is linear and overdetermined ($nl + l$ equations and $nl + n$ unknowns). Thus n is, to some extent, a security parameter.

A possible strategy for finding p and q is to use the estimators $\hat{p} = \max\{b_{ij}\}$ and $\hat{q} = \max\{c_{ij}\}$. However, $\hat{p}$ and $\hat{q}$ are not good enough to get polynomially close to p or q with polynomially many ciphertexts. On the other hand, the techniques described in [3] and [4] to break congruential generators with unknown parameters do not seem to apply here, since there is no recurrent relation between the known cleartexts and ciphertexts that can be exploited. A supplementary way to increase security is for the classified level to keep m secret, because in this case the cryptanalyst cannot even recognize whether the right p and q have been found; unfortunately, there is a tradeoff with cost-effectiveness since keeping m secret prevents the unclassified level from reducing modulo m during computation, which leads to handling very large numbers.

We will finally show that cleartext splitting is essential to the security of the PH, which means that one should take $n \geqslant 2$. The PH is insecure when the known cleartext-ciphertext pairs are initial data encrypted without cleartext splitting (that is, when $n = 1$ and ciphertexts consist of only a first r-degree term).

Table 1
Encrypted perturbations corresponding to elementary operations

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

Proposition 3. *Assume that $n = 1$ and that cleartext-ciphertext pairs $(a_i, [b_i, c_i])$, $i = 1, \ldots, l$ are known. Then the proposed privacy homomorphism can be broken by an extended Brickell-Yacobi gcd attack.*

Proof. The cryptanalyst can write $b_i + n_i p = a_i r_p$ and $b_j + n_j p = a_j r_p$. Now, solving for r_p in the first equation and substituting in the second one, she gets

$$a_i b_j - a_j b_i = (a_j n_i - a_i n_j) p.$$

The left-hand side is known to the cryptanalyst and is a multiple of p. The same procedure is repeated for $1 \leqslant i, j \leqslant l$ with $i \neq j$. Even if the number l of known pairs is small, there is a high probability that p is equal to the greatest common divisor of the multiples obtained in this way. Once p is known, any equation $a_i r_p \bmod p = b_i$ can be solved for r_p. Determining q and r_q is analogous. □

3. Multilevel computation on randomly perturbed data

We conclude by showing an application of the proposed PH to multilevel computation on sensitive data. Using a PH, a classified level can release disclosure protected data (e.g. perturbed data, see [6]) for, say, statistical processing at an unclassified level. Little effort at the classified level suffices to obtain exact results from computations performed by the unclassified level on this perturbed data: restoration of the exact result involves only decrypting the perturbation of the unclassified result. With this scheme, just a "small core" is needed for classified tasks at a statistical office. Moreover, external computing facilities can be subcontracted without compromising statistical secrecy – external service providers being special instances of unclassified levels.

Assume that, at a classified level, a statistical office uses a random perturbation method on sensitive data $x_1, x_2, \ldots, x_n$. This gives $x_i^* = x_i + \varepsilon_i$, for all $i = 1, \ldots, n$ 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 classified level releases the pairs $(x_i^*, E_k(-\varepsilon_i))$ to the unclassified level for further computation. This level 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 $E_k(-\varepsilon_i)$. As explained above, F and F' in the PH proposed here *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 it follows from Section 2 that it can be avoided if rational numbers are represented and handled as fractions.

When the classified level receives the result of a computation from the unclassified level, it decrypts the perturbation and adds the clear perturbation to the perturbed result to obtain the exact result. If perturbations of initial data were converted to integers as illustrated in Example 2, every perturbation received from the unclassified level is a fraction; the numerator of the perturbation must be decrypted and thereafter divided over the real numbers by the decrypted denominator (maybe a power of 10), in order to get the right number of decimal positions.

References

[1] N. Ahituv, Y. Lapid and S. Neumann, Processing encrypted data, *Comm. ACM* **20** (1987) 777–780.

[2] E. Brickell and Y. Yacobi, On privacy homomorphisms, in: D. Chaum and W. L. Price, eds., *Advances in Cryptology - Eurocrypt'87* (Springer, Berlin, 1988) 117-125.

[3] E.F. Brickell and A.M. Odlyzko, Cryptanalysis: A survey of recent results, in: G.J. Simmons, ed., *Contemporary Cryptology* (IEEE Press, New York, 1992) 501-540.

[4] A.M. Frieze, J. Hastad, R. Kannan, J.C. Lagarias and A. Shamir, Reconstructing truncated integer variables satisfying linear congruences, *SIAM J. Comput.* **17** (1988) 262-280.

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

[6] D. Schackis, *Manual on Disclosure Control Methods* (Eurostat, Luxembourg, 1993).

[7] G. Trouessin, Traitements fiables de données confidentielles par fragmentation-rédondance-dissémination, Ph.D. Thesis, Université Paul Sabatier, Toulouse, 1991.