A Provably Secure Additive and Multiplicative Privacy Homomorphism*

Josep Domingo-Ferrer

Universitat Rovira i Virgili, Dept. of Computer Engineering and Maths, Av. Països Catalans 26, E-43007 Tarragona, Catalonia. Tel. +34-977559657, Fax +34-977559710,
jdomingo@etse.urv.es

Abstract. Privacy homomorphisms (PHs) are encryption transformations mapping a set of operations on cleartext to another set of operations on ciphertext. If addition is one of the ciphertext operations, then it has been shown that a PH is insecure against a chosen-cleartext attack. Thus, a PH allowing full arithmetic on encrypted data can be at best secure against known-cleartext attacks. We present one such PH (none was known so far) which can be proven secure against known-cleartext attacks, as long as the ciphertext space is much larger than the cleartext space. Some applications to delegation of sensitive computing and data and to e-gambling are briefly outlined.

Keywords: Privacy homomorphisms, Encrypted data processing, Cryptography, Delegation of computing and data, Multilevel security, E-gambling.

1 Introduction

The first general approach to encrypted data processing is due to the authors of [10], when they introduced the notion of privacy homomorphism (PH from now on). Basically, such homomorphisms are encryption functions $E_k : T' \longrightarrow T$ allowing a set F of operations on encrypted data without knowledge of the decryption function D_k. Knowledge of D_k allows the result of the corresponding set F' of cleartext operations to be retrieved. The availability of secure PHs is central to the development of multilevel secure computation underlying such applications as computing delegation, data delegation and e-gambling: the idea is to encrypt data at a classified level, to process them at an unclassified level and to decrypt the result at a classified level. By way of illustration, consider the following example of PH, given in [10]

* This work has been partly supported by the European Commission under project IST-2001-32012 "Co-Orthogonal Codes" and by the Spanish Ministry of Science and Technology and the European FEDER fund through project no. TIC2001-0633-C03-01 "STREAMOBILE".

A.H. Chan and V. Gligor (Eds.): ISC 2002, LNCS 2433, pp. 471–483, 2002.

Example 1. Let p and q be two large and secret primes (100 decimal digits each). Let $m = pq$ be public. Define the cleartext set as $T' = \mathbf{Z}_m$ and the set of cleartext operations as $F' = \{+_m, -_m, \times_m\}$ consisting, respectively, of addition, subtraction and multiplication modulo m. Let the ciphertext set be $T = \mathbf{Z}_p \times \mathbf{Z}_q$. Operations in the set F of ciphertext operations are the componentwise version of those in F'. Define the encryption key as $k = (p, q)$ and the encryption transformation as $E_k(a) = [a \bmod p, a \bmod q]$. Given $k = (p, q)$, the Chinese remainder theorem is used to compute $D_k([b, c])$. A technical detail is that when the unclassified level computes on encrypted data, it cannot reduce partial results to the secret moduli p and q; only reduction to the public modulus m is possible, so that in fact the unclassified level operates on $\mathbf{Z}_m \times \mathbf{Z}_m$; however, at decryption time, knowledge of the key allows the classified level to map encrypted results from $\mathbf{Z}_m \times \mathbf{Z}_m$ back to $\mathbf{Z}_p \times \mathbf{Z}_q$ prior to using the Chinese remainder theorem.

Unfortunately, it is shown in [3] that this PH can be broken —*i. e.*, p and q can be found— by a known-cleartext attack. •

Next follows a summary of the state of the art on PHs. If a PH preserves order, then it is insecure against a ciphertext-only attack. If addition is one of the ciphertext operations of a PH, then such a PH is insecure against a chosen-cleartext attack ([1]). With the exception of the RSA algorithm —which preserves only multiplication—, all examples given in [10] were broken by ciphertext-only attacks or, at most, known-cleartext attacks (see [3]); the authors of [3] invented R-additive PHs, which securely allow ciphertext addition at the cost of restricting the number of ciphertexts that can be added together. In [2], a partially homomorphic scheme for statistical computation on encrypted data was proposed that consists of a two-layer encryption: data records are first encrypted as sparse polynomials, and these are then encrypted as regular polynomials; while the first layer is homomorphic, the second is not (yet the second layer is needed because the PH in the first layer is insecure); therefore, encrypted data processing is not feasible without a trusted device able to decrypt the second layer. Lacking secure PHs that preserve more than one operation, subsequent attempts at encrypted data processing have relied on *ad-hoc* solutions ([1,15]). In [4], we presented a PH preserving addition and multiplication which was conjectured to be computationally resistant against known-cleartext attacks. In [8,12], it was wondered whether provably secure algebraic (*i.e.* additive and multiplicative) privacy homomorphisms exist; this paper is meant to answer that question.

1.1 Our Contribution

In this paper, we propose the first PH preserving both addition and multiplication that can be proven secure against known-cleartext attacks, as long as the ciphertext space is much larger than the cleartext space. In Section 2 the homomorphism is specified. In Section 3 a numerical example is given. Security is proven in Section 4. Section 5 mentions some practical applications to delegation of computing and data and to e-gambling. Section 6 is a conclusion.

2 Specification of the New PH

The PH proposed in this paper can be described as follows:

- The public parameters are a positive integer $d > 2$ and a large integer m ($\approx 10^{200}$ or maybe larger). m should have many small divisors and at the same time there should be many integers less than m that can be inverted modulo m; the first condition can be satisfied by construction of m, and the second condition can be satisfied by iterating until an m is found such that $\phi(m)$ is close to $6m/(\pi^2)$, which is the expected value of $\phi(m)$ for a random m (see Lemma 5 below).
- The secret parameters are $r \in \mathbf{Z}_m$ such that $r^{-1} \bmod m$ exists and a small divisor $m' > 1$ of m such that $s := \log_{m'} m$ is a (secret) security parameter; the influence of the sizes of m' and m on security can be seen in Table 1 below. Thus, the secret key is $k = (r, m')$.

In this case the set of cleartext is $T' = \mathbf{Z}_{m'}$. The set of ciphertext is $T = (\mathbf{Z}_m)^d$. The set F' of cleartext operations is formed basically by addition, subtraction and multiplication in T'. The set F of ciphertext operations contains the corresponding componentwise operations in T. The PH transformations can be described as

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

$$E_k(a) = (a_{.1}r \bmod m, a_{.2}r^2 \bmod m, \cdots, a_{.d}r^d \bmod m) \tag{1}$$

Decryption. Compute the scalar product of the j-th coordinate by $r^{-j} \bmod m$ to retrieve $a_{.j} \bmod m$. Compute $\sum_{j=1}^{d} a_{.j} \bmod m'$ to get a.

As encrypted values are computed over $(\mathbf{Z}_m)^d$ at an unclassified level, the use of r requires that the terms of the encrypted value having different r-degree be handled separately —the r-degree of a term is the exponent of the power of r contained in the term—. This is necessary for the classified level to be able to multiply each term by r^{-1} the right number of times, before adding all terms up over $\mathbf{Z}_{m'}$.

The set F' of ciphertext operations consists of

Addition and subtraction. They are done componentwise, *i. e.* between terms with the same degree.

Multiplication. It works like in the case of polynomials: all terms are cross-multiplied in $\mathbf{Z}_m$, with a d_1-th degree term by a d_2-th degree term yielding a $d_1 + d_2$-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 and handle divisions in rational format by considering the field of rational functions: the encrypted version of a/b is $E_k(a)/E_k(b)$.

In addition to the operations in F', it is also possible to multiply all components of a ciphertext vector $E_k(a)$ by a cleartext constant c. If the resulting ciphertext vector is decrypted, $ac \bmod m'$ is obtained. Whenever possible, this operation should be preferred to multiplication of two ciphertexts, as ciphertext multiplication is the only operation in F that increases the r-degree of the result.

Note 2. Unlike for the PH in Example 1, in our PH the cleartext space is unknown to the unclassified level, because the parameter m' is secret. This will be useful to prove the security of our proposal. Notice that if the unclassified level is told which is the ciphertext space, then it needs no knowledge on the cleartext space to do encrypted computations. However, one of the troubles with Example 1 was that the unclassified level cannot be revealed which is the ciphertext space (giving away the ciphertext space $\mathbf{Z}_p \times \mathbf{Z}_q$ is equivalent to revealing the secret key (p, q)); therefore, knowledge of the size $m = pq$ of the cleartext space (or another common multiple of p and q) was needed to reduce partial encrypted results.

3 Numerical Example

This example is unrealistically small but it illustrates the computation of a formula including two additions and one multiplication, namely $(x_1+x_2+x_3)x_4$. Although $d > 2$ is recommended for security reasons (see Remark 9 below), we will take $d = 2$ to keep computations brief and clear; thus, cleartexts will be split in two parts during encryption. The public modulus is chosen to be $m = 28$.

Classified level.

Let $r = 3$ and $m' = 7$ 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, initial data are multiplied by 10, which yields the fractions $x_1 = \tilde{x}_1/10 = -1/10$, $x_2 = \tilde{x}_2/10 = 3/10$, $x_3 = \tilde{x}_3/10 = 1/10$ and $x_4 = \tilde{x}_4/1 = 2/1$. Numerators are randomly and secretly split mod 7 and are transformed according to the proposed PH. In this way, first and second r-degree terms are obtained

$$E_k(\tilde{x}_1) = E_k(-1) = E_k(2, 4) = (6, 8)$$

$$E_k(\tilde{x}_2) = E_k(3) = E_k(2, 1) = (6, 9)$$

$$E_k(\tilde{x}_3) = E_k(1) = E_k(4, 4) = (12, 8)$$

$$E_k(\tilde{x}_4) = E_k(2) = E_k(3, 6) = (9, 26)$$

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, do the additions by directly adding the numerators in the fractions, since the denominator is 10 for all data

$$\sum_{i=1}^{3} E_k(\tilde{x}_i) = (6 + 6 + 12 \bmod 28, 8 + 9 + 8 \bmod 28) = (24, 25)$$

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

$$(E_k(\tilde{x}_1) + E_k(\tilde{x}_2) + E_k(\tilde{x}_3))E_k(\tilde{x}_4) = (24, 25) \times (9, 26)$$

$$= (0, 24 \times 9 \bmod 28, 24 \times 26 + 25 \times 9 \bmod 28, 25 \times 26 \bmod 28) = (0, 20, 9, 6)$$

In this way, 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

$$(0 \times r^{-1} \bmod m, 20 \times r^{-2} \bmod m, 9 \times r^{-3} \bmod m, 6 \times r^{-4} \bmod m)$$

$$= (0 \times 19 \bmod 28, 20 \times 19^2 \bmod 28, 9 \times 19^3 \bmod 28, 6 \times 19^4 \bmod 28)$$

$$= (0, 24, 19, 26)$$

Now add all terms in the last step over $\mathbf{Z}_{m'}$ to obtain 6 mod 7 = 6. Thus, we have $(\tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3)\tilde{x}_4 = 6 \bmod m' = 6$. Finally, divide 6 by the denominator 10 returned by the unclassified level, so that the final result is $(x_1 + x_2 + x_3)x_4 = 0.6$.

4 Security of the New Privacy Homomorphism

Definition 3. *A privacy homomorphism is said to be secure against a known-cleartext attack if, for any fixed number n of known cleartext-ciphertext pairs, the probability of successful decryption of a ciphertext for which the cleartext is unknown can be made arbitrarily small by properly choosing the security parameters of the homomorphism.*

We will show in this section that the PH whose encryption function is given by Expression (1) is secure. First, it will be shown that, for a fixed number n of known cleartext-ciphertext pairs, the probability of randomly guessing the right key can be made arbitrarily small. Second, it will be shown that there is only a small probability that a ciphertext decrypts to the same cleartext using two different keys. Combining both results, security will follow. We next recall three known preliminary results:

Lemma 4. *Assume that divisibility of an integer by different primes is independent and that divisibility of randomly chosen integers by the same prime is independent. Let B be a positive integer. If positive integers $c_1, \cdots, c_n$ are randomly drawn from the interval $(0, B)$, then*

$$\lim_{B \to \infty} Pr\{\gcd(c_1, c_2, \cdots, c_n) = 1\} \approx \frac{1}{\zeta(n)} \tag{2}$$

where $\zeta(n) = \sum_{t=1}^{\infty} t^{-n}$ is Riemann's zeta function.

Proof. Let $B \to \infty$ and see [13], section 4.4. •

Lemma 5. *If $\phi(m)$ is Euler's totient function counting the number of integers less than m that are coprime with m, then*

$$\phi(1) + \cdots + \phi(m) = \frac{3m^2}{\pi^2} + O(m \log m) \tag{3}$$

In particular, the average order of $\phi(m)$ is $6m/\pi^2 \approx 0.608m$.

Proof. See [9], section 18.5. •

Lemma 6. *Let $d(n)$ be the number of divisors of a positive integer n, counting 1 and n. The average order of $d(n)$ is $\log n$.*

Proof. See [9], section 18.2. •

The first result concerning the security of the new privacy homomorphism against known-cleartext attacks regards the subset of keys consistent with the known cleartext-ciphertext pairs:

Theorem 7. *Consider a PH whose encryption function is given by Expression (1). Let n be the number of* random *cleartext-ciphertext pairs known by the cryptanalyst. If the r-degree of all ciphertexts is greater than 1, then the size of the subset of keys consistent with the known pairs grows exponentially with $s - n$ and has an expected value of at least $\max(6(m')^{s-n}/\pi^2, 1)$, where $s = \log_{m'} m$. Cleartext-ciphertext pairs derived from the n known pairs using the homomorphic properties do not compromise the security of the PH.*

Proof. Denote by d the maximal r-degree of ciphertexts in known message pairs. Let the n known random message pairs consist of cleartexts a_i and ciphertexts $(b_{i1}, \cdots, b_{id})$, for $i = 1, \cdots, n$. The following construction shows that if n is not too large, then there exist several keys $(\hat{r}, \hat{m}')$ consistent with the n known pairs

1. Randomly pick $\hat{r}$ such that $\hat{r}^{-1} \bmod m$ exists. Clearly, all numbers coprime to m are eligible; thus, there are $\phi(m)$ candidates.
2. For $i = 1, \cdots, n$ compute $\hat{a}_{i1}, \cdots, \hat{a}_{id}$ such that $\hat{a}_{ij} = b_{ij}\hat{r}^{-j} \bmod m$.
3. Find $\hat{m}'$ such that it divides m and verifies

$$\hat{a}_{i1} + \cdots + \hat{a}_{id} = a_i \bmod \hat{m}'$$

for $i = 1, \cdots, n$. A possibility (perhaps not unique) is to take

$$\hat{m}' = \gcd_{1 \le i \le n} \left(\sum_{j=1}^{d} \hat{a}_{ij} - a_i, m\right)$$

where, to keep the notation short, we have defined $\gcd_{1 \le i \le n}(c_i, m) := \gcd(c_1, c_2, \cdots, c_n, m)$. If $\hat{m}' \le \max_{1 \le i \le n}(a_i)$ is obtained, then go to Step 1. Otherwise a key $(\hat{r}, \hat{m}')$ consistent with the known pairs has been obtained and the procedure is finished.

The probability of coming up with a good $\hat{m}'$ at Step 3 of the above construction can be lower-bounded as

$$Pr(\gcd_{1\le i\le n}(\sum_{j=1}^{d}\hat{a}_{ij}-a_i,m)>\max_{1\le i\le n}(a_i))\ge Pr(\gcd_{1\le i\le n}(\sum_{j=1}^{d}\hat{a}_{ij}-a_i,m)\ge m')$$

$$\ge Pr(\gcd_{1\le i\le n}(\sum_{j=1}^{d}\hat{a}_{ij}-a_i,m)=m')$$

$$=Pr(A)Pr[\gcd_{1\le i\le n}(\frac{\sum_{j=1}^{d}\hat{a}_{ij}-a_i}{m'},m/m')=1|A]$$

$$\approx(\frac{1}{m'})^n\frac{1}{\zeta(n+1)}\approx\frac{1}{(m')^n} \tag{4}$$

where the last approximation is valid if n is not too small (say $\ge$ 10) and the ζ approximation is obtained from Lemma 4. A is the event "m' divides m and all $\sum_{j=1}^{d}\hat{a}_{ij}-a_i,\quad i=1,\cdots,n$"; clearly, by assumption m' divides m, and the probability that m' divides a random integer (such as $\sum_{j=1}^{d}\hat{a}_{ij}-a_i$) is $1/m'$; thus, $Pr(A)=(1/m')^n$. Let us check that Lemma 4 can be used:

1. In the last gcd computation, m/m' is random (because m is) and the rest of numbers can be viewed as being randomly drawn from the interval $(0,dm/m')$, since they depend on a random number $\hat{r}$.
2. dm/m' is large since m' is a small divisor of m.

Now the above construction can be run for $\phi(m)$ different values of $\hat{r}$. This means that the expected number of keys $(\hat{r},\hat{m}')$ consistent with the known pairs is at least

$$\max(\frac{\phi(m)}{(m')^n},1)\approx\max(\frac{6}{\pi^2}\frac{m}{(m')^n},1)=\max(\frac{6}{\pi^2}(m')^{s-n},1)$$

where the approximation is obtained from Lemma 5. Finally, to prove the last assertion of the theorem, imagine that two known pairs are added, subtracted or multiplied by the cryptanalyst to generate a new cleartext-ciphertext pair. If this new pair is input to the gcd computation at Step 3, it is easy to see that the gcd value remains unchanged. Thus only genuine randomly split cleartext-ciphertext known pairs are to be taken into account. •

Note 8. m' must be considered as a secret parameter for the above proof of Theorem 7 to be correct. Otherwise, consistent keys would be only those having the form $(\hat{r},m')$ and their number would be much smaller. Even if it is assumed that the enemy cryptanalyst can compute all divisors $\hat{m}'$ of m, she cannot decide which divisor is actually being used; the only clue is that $\hat{m}'>\max_{1\le i\le n}(a_i)$, as reflected in Derivation (4).

Note 9 (On the value of d). It is recommended that $d>2$. The shortcomings of $d=1$ and $d=2$ are examined below:

$d = 1$**:** Notice that random cleartext splitting is central to the proof of Theorem 7. Imagine that no splitting is done (*i. e.*, $d = 1$) and that one cleartext-ciphertext pair (a, b) is known. Then $b = ar \bmod m$ and r is revealed.

$d = 2$**:** It will be shown that, if $d = 2$, knowledge of m' allows to determine r, that is, knowledge of a part of the secret key k allows to determine the rest of k. Then for message 0, one has $E_k(0) = (ar, a'r^2)$, where $a = -a' \bmod m'$; now notice that $a'r^2/ar \bmod m'$ yields $r \bmod m'$. But how can the enemy cryptanalyst get $E_k(0)$? Assume that four cleartext-ciphertext pairs are known to her, namely $B_1 = E_k(a_1)$, $B_2 = E_k(a_2)$, $B_3 = E_k(a_3)$ and $B_4 = E_k(a_4)$. There is a high probability that two pairs of coprime numbers exist among the cleartexts; assume $\gcd(a_1, a_2) = 1$ and $\gcd(a_3, a_4) = 1$. Then u, v, u', v' exist such that $B_5 := E_k(1) = uB_1 + vB_2$ (equivalently $1 = ua_1 + va_2$) and $B_6 := E_k(1) = u'B_3 + v'B_4$ (equivalently $1 = u'a_3 + v'a_4$). Then $B_5 - B_6 = E_k(0)$. Notice that the above attack does not work without knowledge of m', so $d = 2$ is not necessarily insecure; still it is not recommended.

Note 10. Instead of using the average approximation from Lemma 5, one might think of using the Rosser-Schoenfeld lower bound on $\phi(m)/m$ ([11], p. 72) in the proof of Theorem 7. However, being based on a smooth function of m, this bound is often too conservative and can lead to a serious underestimation of the size of the subset of consistent keys. Due to this drawback and the probabilistic nature of the proof, it seems more realistic and natural to use the expected value of $\phi(m)$.

The second result concerning the security of the proposed PH regards the behaviour of keys:

Theorem 11. *The expected probability that any two keys (r_1, m_1') and (r_2, m_2') decipher a random ciphertext to the same cleartext is $O((\log m)/m)$. Therefore, this probability can be made arbitrarily small by increasing m.*

Proof. Let the two keys be (r_1, m_1') and (r_2, m_2'). Let $(b_1, b_2, \cdots, b_d)$ be a randomly chosen ciphertext. Assume that both keys decrypt to the same cleartext a. Then from the specification of the PH (Section 2):

$$a = P(r_1^{-1}) \bmod m_1' = b_1 r_1^{-1} + b_2 r_1^{-2} + \cdots + b_d r_1^{-d} \bmod m_1'$$

$$= P(r_2^{-1}) \bmod m_2' = b_1 r_2^{-1} + b_2 r_2^{-2} + \cdots + b_d r_2^{-d} \bmod m_2' \qquad (5)$$

where the inverses are modulo m. Three cases must be considered:

- If $r_1 = r_2 = r$ then $m_1' \neq m_2'$ since both keys are assumed to be different. In that case, Equation (5) holds only if both m_1' and m_2' divide $a - P(r^{-1})$. Assuming that a is such that m_1' is a divisor, from Lemma 6 the expected probability that m_2' is also a divisor is $(\log m)/m$.

– If $m'_1 = m'_2 = m$ then $r_1 \neq r_2$ since both keys are assumed different. In that case, Equation (5) holds only if m divides $P(r_1^{-1}) - P(r_2^{-1})$. From Lemma 6, this happens with an expected probability

$$\frac{\log |P(r_1^{-1}) - P(r_2^{-1})|}{|P(r_1^{-1}) - P(r_2^{-1})|} = O((\log m)/m) \tag{6}$$

– If $m'_1 \neq m'_2$ and $r_1 \neq r_2$, Equation (5) implies that there exists an integer a such that m'_1 divides $P(r_1^{-1}) - a$ and m'_2 divides $P(r_2^{-1}) - a$. Using Lemma 6, the probability of such an event can be upper bounded by

$$\max_{0 \leq a \leq \min(m'_1, m'_2) - 1} \left(\frac{\log |P(r_1^{-1}) - a|}{|P(r_1^{-1}) - a|}, \frac{\log |P(r_2^{-1}) - a|}{|P(r_2^{-1}) - a|}\right) = O((\log m)/m) \tag{7}$$

Thus in all cases the expected probability of obtaining the same cleartext from decryption of the same ciphertext using two different keys is $O((\log m)/m)$ •

Note 12. It should be noted that the security provided by the proposed scheme is based on the fact that the subset of keys consistent with the known pairs is kept large and any two different keys yield different cleartexts from the same ciphertext with a high probability. Further, in the proof of Corollary 13 it is assumed that an infinitely powerful cryptanalyst can enumerate the subset of consistent keys, but no easy way to do this is obvious. In computational terms, even if there exists only one consistent key (probability of random key guessing equal to 1 for infinite computing power), this does not mean that such a key is easy to find.

Corollary 13. *If leakage of n cleartext-ciphertext pairs is to be tolerated, then the probability of success for a known-cleartext attack with unlimited computing power can be made arbitrarily small by a proper choice of the security parameter $s = \log_{m'} m$.*

Proof. Assume that the cryptanalyst has enough computing power to enumerate the subset of keys consistent with the known pairs. From Theorems 7 and 11 it is clear that the best attacking strategy is randomly guessing the key, which has a probability of success at most equal to

$$\begin{cases} \pi^2 (m')^{n-s}/6 & \text{if } s > n \\ 1 & \text{if } s \leq n \end{cases}$$

For any $\epsilon > 0$, there exists a value of s that makes this probability smaller than ϵ. •

Table 1 illustrates the dependency between parameters with several example choices. There are at least two scenarios for parameter design:

– If n, m' and the probability of random key guessing are specified as requirements, then suitable values for s and m must be determined.

– If m' and m are fixed in advance (*i.e.* the PH is fixed), then the probability of random key guessing can be computed for each number n of known cleartext-ciphertext pairs. If more and more random pairs are leaked over time, then a proactive key renewal scheme should be enforced by the classified level in order to keep the probability of random key guessing smaller than an alarm threshold set beforehand.

Table 1. Some example parameter choices for the proposed privacy homomorphism ($l(x) = \lceil log_{10} x \rceil$ is the length of x in decimal digits).

n	s	$l(m')$	$l(m)$	Prob. rand. key guessing
5	5	20	100	1
5	6	20	120	$\approx 1.64 \times 10^{-20}$
10	11	20	220	$\approx 1.64 \times 10^{-20}$
50	50	5	250	1
50	51	5	255	$\approx 1.64 \times 10^{-5}$
50	53	5	265	$\approx 1.64 \times 10^{-15}$

5 Applications to Delegation of Computing and Data and to E-gambling

Delegation of computing and data is a major field of application for PHs. When the computations to be performed on encrypted data are of an arithmetical nature, then the PH presented in this paper is especially useful. We next sketch some practical scenarios where delegation problems appear.

A computing delegation problem happens whenever a (small) company wants to use external computing facilities to do some calculations on corporate confidential data. A very common variant of this situation is a medical research team using a (insecure) university mainframe for processing confidential healthcare records. The reason for using external facilities may be the complexity of the calculations but also the huge volume of the data set. Clearly, computation on confidential data can be delegated to an unclassified facility (denoted by data handler) if data are kept in encrypted form during computations. The data owner is only left the task of decrypting the final result of computation.

Data delegation differs from computing delegation in that the party interested in the result of computations is not the data owner, but the data handler. Data delegation problems appear in the interaction between public administrations at several levels. For example, municipalities cooperate with national statistical offices in statistical data collection. In return, municipalities would like to be able to analyze the whole collected data set (pooled from all municipalities). But only national statistical offices are usually authorized to hold nation-wide individual

census data. A similar problem occurs in any federal-like structure (European Union, U.S.A., Germany, etc.). Member states cooperate with federal agencies in collecting data from individuals, companies, etc. In return, states would like to be able to analyze data at a federal level. A secure solution in both scenarios above is for the organization owning the whole data set to perform (probably for free) the analyses requested by the cooperating organizations. But then the data owning organization becomes a bottleneck and is forced to waste time and resources in uninteresting tasks. A better solution would be for the data owner to delegate data in a secure way and reduce its role in subsequent analyses to a minimum. Delegation of data can be performed by releasing them in encrypted form. The cooperating organization plays the role of the data handler in that it can compute on encrypted data and submit the results of computations to the data owner for decryption; in this way, the data owner is only left the (mechanical) job of decrypting and returning the final result. A prototype using the PH proposed here to implement such a scheme for delegation of sensitive statistical data has been sketched ([6]), completed and patented[7]; because of patent reasons, no public description of the proposed PH and its security properties had so far been offered.

In [5], availability of secure data delegation is assumed for increasing the multi-application capacity of smart cards. The basic idea is that if a very resource-demanding application is to be run on card-stored data then the card exports these data in encrypted form and the application is run on an external computing server.

Data delegation has stronger security requirements than computing delegation. In computing delegation the data handler only sees ciphertext. However, in data delegation the data owner decrypts the final result of computations and returns it as cleartext to the data handler. This means that in data delegation each time the data owner returns one decrypted result to the data handler, the data handler learns one new cleartext-ciphertext pair (or two if what is returned is an unreduced fraction). Therefore, the data owner should take care that the number n of returned decrypted results is not too large to become unsafe (given the security parameter s of the homomorphism, see Corollary 13). Once the safety limit has been reached, no more decrypted results should be returned by the data owner, unless he decides to change the key of the homomorphism and reencrypt all delegated data under the new key.

E-gambling, and more specifically electronic poker, is another recent application of the PH proposed in this paper. Cards chosen by players are homomorphically encrypted and they are manipulated in encrypted form. In e-poker, PH card encryption plays a role analogous to turning cards upside down in physical poker; computing on encrypted cards is analogous to manipulating reversed cards in the physical world. A protocol describing this application of the PH presented here is patent pending ([14]), so no further details on it can be disclosed here.

6 Conclusion

The features of the proposed homomorphism can be summarized as follows

- Addition, subtraction, multiplication and division can be carried out on encrypted data at an unclassified level.
- The proposed homomorphism is the first one to allow full arithmetic while being secure against known-cleartext attacks. Security against known-cleartext attacks can be proven provided that cleartext splitting is always used when encrypting (recommended splitting factor $d > 2$) and the ciphertext space is much larger than the cleartext space. Pairs that are derived from random pairs using the homomorphic properties do not compromise the security of the PH.
- Encryption and decryption transformations can be implemented efficiently, because they only require modular multiplications. Note that no exponentiation is needed, because the powers of r can be precomputed. Unlike exponential ciphers, the proposed PH can be fast even if the ciphertext space is very large.
- A ciphertext with r-degree d is about

$$d\frac{\log m}{\log m'} = d\log_{m'} m = ds$$

  times longer than the corresponding cleartext. Although this is a storage penalty, a choice of $d = 3$ at the time of encryption should be affordable while remaining secure. For a given r-degree, the ciphertext expansion grows linearly with the security parameter s.
- Multiplication of two ciphertexts is the only operation that increases the r-degree of the resulting ciphertext. A good strategy is to use multiplication by cleartext constants instead of multiplication of two ciphertexts whenever possible.
- The equality predicate is not preserved, and thus comparisons for equality cannot be done at an unclassified level based on encrypted data. A given cleartext can have many ciphertext versions for two reasons: A) random splitting during encryption; B) the unclassified level computes over $(\mathbf{Z}_m)^d$ and only the classified level can perform a reduction to $\mathbf{Z}_{m'}$ during decryption.

Acknowledgments. Thanks go to Stefan Brands and Josep Rifà for useful talks and comments.

References

1. N. Ahituv, Y. Lapid and S. Neumann, "Processing encrypted data", *Communications of the ACM*, vol. 20, no. 9, pp. 777–780, Sept. 1987.

2. G. R. Blakley and C. Meadows, "A database encryption scheme which allows the computation of statistics using encrypted data", in *Proceedings of the IEEE Symposium on Research in Security and Privacy*. New York: IEEE CS Press, 1985, pp. 116–122.
3. E. F. Brickell and Y. Yacobi, "On privacy homomorphisms", in *Advances in Cryptology-Eurocrypt'87*, D. Chaum and W. L. Price, Eds. Berlin: Springer-Verlag, 1988, pp. 117–125.
4. J. Domingo-Ferrer, "A new privacy homomorphism and applications", *Information Processing Letters*, vol. 60, no. 5, pp. 277–282, Dec. 1996.
5. J. Domingo-Ferrer, "Multi-application smart cards and encrypted data processing", *Future Generation Computer Systems*, vol. 13, pp. 65–74, Jun. 1997.
6. J. Domingo-Ferrer and R. X. Sánchez del Castillo, "An implementable scheme for secure delegation of statistical data", in *Information Security-ICICS'97*, Lecture Notes in Computer Science 1334, Y. Han, T. Okamoto and S. Qing, Eds. Berlin: Springer-Verlag, 1997, pp. 445–451.
7. J. Domingo-Ferrer and Ricardo X. Sánchez del Castillo, "Method for secure delegation of statistical data", Spanish patent no. P9800608, granted Dec. 2000.
8. J. Feigenbaum and M. Merritt, "Open questions, talk abstracts, and summary of discussions", *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 2, pp. 1–45, 1991.
9. G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, 5th ed. Oxford: Clarendon, 1993.
10. R. L. Rivest, L. Adleman and M. L. Dertouzos, "On data banks and privacy homomorphisms", in *Foundations of Secure Computation*, R. A. DeMillo *et al.*, Eds. New-York: Academic Press, 1978, pp. 169-179.
11. J. B. Rosser and L. Schoenfeld, "Approximate formulas for some functions of prime numbers", *Illinois Journal of Mathematics*, vol. 6, no. 1, pp. 64–94, Jan. 1962.
12. T. Sander and C. F. Tschudin, "Protecting mobile agents against malicious hosts", in *Mobile Agent Security*, Lecture Notes in Computer Science 1419. Berlin: Springer-Verlag, 1998, pp. 44–60.
13. M. R. Schroeder, *Number Theory in Science and Communication*, 2nd ed. Berlin: Springer-Verlag, 1986.
14. SCYTL Online World Security. `http://www.scytl.com`
15. G. Trouessin, *Traitements Fiables des Données Confidentielles par Fragmentation-Rédondance-Dissémination*. Ph. D. Thesis, Univ. Paul Sabatier, Toulouse, France, 1991.