Modulo Reduction for Paillier Encryptions and Application to Secure Statistical Analysis (extended abstract)

Jorge Guajardo[1], Bart Mennink[⋆,2] and Berry Schoenmakers[⋆,3]

[1] Information and System Security Group
Philips Research, Eindhoven, The Netherlands
jorge.guajardo@philips.com
[2] Dept. Electrical Engineering, ESAT/COSIC and IBBT
Katholieke Universiteit Leuven, Belgium
bart.mennink@esat.kuleuven.be
[3] Dept. of Mathematics and Computer Science
Technische Universiteit Eindhoven, The Netherlands
berry@win.tue.nl

Abstract. For the homomorphic Paillier cryptosystem we construct a protocol for secure modulo reduction, that on input of an encryption $[\![x]\!]$ with x of bit length ℓ_x and a public 'modulus' a of bit length ℓ_a outputs an encryption $[\![x \bmod a]\!]$. As a result, a protocol for computing an encrypted integer division $[\![x \operatorname{div} a]\!]$ is obtained. Surprisingly, efficiency of the protocol is independent of ℓ_x: the broadcast complexity of the protocol varies between $O(nk\ell_a)$ and $O(n^2k\ell_a)$, for n parties and security parameter k, and it is very efficient in case of small ℓ_a (in practical cases ℓ_a often *is* much smaller than ℓ_x). Our protocol allows for efficient multiparty computation of statistics such as the mean, the variance and the median, and it is therefore very applicable to surveys for the benefit of statistical analysis.

1 Introduction

We consider the problem of integer division with remainder in the setting of secure multiparty computation. In its full generality, the problem is to evaluate securely the integer function $(x, y) \mapsto (x \operatorname{div} y, x \bmod y)$, where $x = (x \operatorname{div} y)y + x \bmod y$ and $0 \leq x \bmod y < y$. Whereas integer multiplication commonly allows for secure protocols for which the performance is independent of the bit length of the multiplicands, this is not true for known protocols for integer division. Typically, secure integer division protocols use the binary decomposition of the inputs x and/or y, and consequently these protocols are generally much more elaborate than secure multiplication protocols. To a certain extent, this is to be expected because integer comparison (which generally also requires bitwise-represented inputs) reduces to equality testing given integer division: $x < y$ if

⋆ Work done partly while visiting Philips Research Labs.

and only if $x = x \bmod y$.

In this paper we will focus on the computation of $x \bmod a$ for a public modulus a. We observe that in many applications, particularly in secure statistical analysis, division is used with a public modulus only. For example, for the mean $\bar{x} = (x_1 + \cdots + x_L) \operatorname{div} L$ of a set of L values, it suffices to divide by the publicly known L. A similar observation can be made for the computation of the variance. Hence, efficient protocols for the case of public a are clearly of interest, and we will show how to achieve efficient solutions. Although our results apply to a broad range of approaches in secure multiparty computation, we will present our protocols mostly for the framework based on threshold homomorphic cryptosystems (THCs) [11, 5, 22]. This framework allows n parties, $n \geq 2$, to securely and privately evaluate a given function f: given encrypted inputs $[\![x_1]\!], \ldots, [\![x_L]\!]$, it will be ensured that the output is an encryption $[\![f(x_1, \ldots, x_L)]\!]$, without leaking any further information on the values $x_1, \ldots, x_L$. In general, one may construct a Boolean or arithmetic circuit for f, consisting of basic gates such as NAND gates or addition/multiplication gates, and then evaluate this circuit securely. For performance reasons, however, specific protocols are needed to obtain more practical solutions.

The advantage of our THC-based protocols is that the malicious case can be treated without efficiency loss (asymptotically) compared to the semi-honest case. Our protocols can also be translated to the framework based on verifiable secret sharing (cf. [6]). In this case, however, the malicious case will be (asymptotically) more expensive than the semi-honest case. The technical reason is that some of the particularly efficient zero-knowledge interval proofs used in the THC-based approach do not carry over to the VSS-based approach. Our protocols can be seen as a generalization of the bit decomposition protocols of [23]. As observed in [23], the problem of evaluating $x \mapsto x \bmod 2$ is already non-trivial as it cannot be solved when ElGamal is used as the underlying (additively) homomorphic cryptosystem: an efficient protocol for computing the least significant bit $[\![x \bmod 2]\!]$ for given $[\![x]\!]$ would imply efficient computation of a hard-core bit (of the one-way function $x \mapsto g^x$), contradicting the discrete log assumption. Therefore, we will use a sufficiently strong homomorphic cryptosystem for our protocols, concretely the Paillier cryptosystem. An immediate application of integer division is to securely access arbitrary bits of a given input $[\![x]\!]$ efficiently. For an ℓ_x-bit integer x, the work to access the i-th least significant bit will be proportional to i, as $x_i = (x \operatorname{div} 2^i) \bmod 2$. Our protocols actually simplify considerably for the case that a is a power of 2, such that the overall work is much less than one would need using the bit decomposition protocol of [23].

1.1 Our Contributions

For a set of n participants jointly sharing the decryption key of the Paillier cryptosystem, we construct a protocol which, on input $[\![x]\!]$ with $x < 2^{\ell_x}$ and a public 'modulus' a such that $2^{\ell_a - 1} < a \leq 2^{\ell_a}$, outputs an encryption of the

modulo reduction of x with respect to a, $[\![x \bmod a]\!]$. Consequently, this implies a protocol for computing an encrypted integer division $[\![x \operatorname{div} a]\!]$. The efficiency of the protocol relies on the fact that a is known and, in particular, its length is known. The protocol has a broadcast complexity varying between $O(nk\ell_a)$ and $O(n^2k\ell_a)$ (with corresponding round complexities $O(n)$ and $O(1)$), where k is a security parameter, and the variation depends on the building blocks used (e.g., for random bit generation several protocols are known, which differ in complexities). In [16, Sect. 5], the protocol is proven statistically secure in the framework of Cramer et al. [5].

As an interesting application, this protocol can be used for secure and efficient statistical analysis on encrypted data. In [16, Sect. 8], a protocol for the computation of the variance of L inputs is constructed in detail. Other statistics can be implemented similarly. The possibility to securely evaluate statistics allows for a broad range of applications, like (medical) surveys. In medical surveys, many users release medical data to some institute which analyzes the data and outputs some result (a diagnostic, a result of statistical analysis, etc.). However, medical data are privacy sensitive and users might be unwilling to reveal these data in plaintext. Using secure multiparty computation, the institute is represented by a set of multiparty computation servers and the users can input their medical data *in encrypted form*. The servers then use the modulo reduction protocol for secure statistical analysis. We end by noticing that the protocol can easily be carried over to a client/server setting [16, Sect. 7], and that it has many other practical applications, for instance in the area of secure face recognition [10], packing of encrypted values [2] and auctions [7, 12]. In particular, a modulo reduction protocol allows for easily obtaining packed encrypted values out of one encryption.

1.2 Related Work

The relation with the bit decomposition protocols of [23] has already been discussed. For the unconditional setting using verifiable secret sharing, Algesheimer et al. [1] constructed a modulo reduction protocol which works for encrypted modulus a. The protocol relies on approximating $1/a$ (for which also a protocol by Kiltz et al. [18] can be used). This protocol is only of theoretical interest[4]: instead of $x \bmod a$, the value $x \bmod a + ia$ is computed, with $|i| < (n+1)(5 + 2^{4+\ell_x-\ell_a-\ell'})$ for some additional security parameter ℓ' (the number of correctly approximated bits of $1/a$). The value $x \bmod a$ is then computed after $O(n2^{\ell_x-\ell_a-\ell'})$ executions of a comparison protocol, which makes the protocol inefficient. We note that our protocol does not rely on approximations. More comparable to ours is the VSS-based protocol by Damgård et al. [6], which opts for constant rounds. Unlike ours, their scheme does not make use of the form

[4] We note that From and Jakobsen [13, Ch. 8] discuss the efficiency of the protocol of [1]. They conclude that the performance is generally low, particularly for a large number of participants. See also [24, Sect. 4.6].

of a. In particular, in the THC-setting their protocol has a broadcast complexity varying between $O(nk\ell_x(\log \ell_x + \ell_a))$ and $O(nk\ell_x(n + \log \ell_x + \ell_a))$, where $\ell_x \geq \ell_a$. In many practical applications the value ℓ_a is even *much smaller* than ℓ_x, as exemplified in Sect. 5. Using ideas of [1], their protocol can also be extended to secret a. We stress that for our purposes the protocol with public a suffices. In [4], Catrina and Dragulin independently introduce a modulo reduction protocol similar to ours. However, unlike ours, their protocol is constructed for secure computation based on secret sharing and considers modulo reduction by powers of two only. Our protocol works for general a, and in particular relies on efficient ways for generating random values from $[0, a)$ securely. Moreover, [4] provides security against semi-honest adversaries only, while our protocol covers the malicious case.

Although our main concern is the modulo reduction protocol, we also consider related work with respect to statistical multiparty computation, which is used as motivational example. Many works on privacy-preserving statistical analysis (e.g., [9, 18]) focus only on techniques other than THC-based secure multiparty computation. In [17], computation of moments is considered for Paillier encryptions and used to compute statistics like the mean and the variance. The authors circumvent the need for a modulo reduction protocol by applying division on the decrypted moments only. This protocol is not of practical interest: if for instance $\sum_{i=1}^{L} x_i$ is decrypted rather than $(\sum_{i=1}^{L} x_i) \operatorname{div} L$, the protocol unintentionally leaks information about the inputs, namely $(\sum_{i=1}^{L} x_i) \bmod L$.[5] Moreover, the protocol of [17] cannot be integrated as a sub-protocol with encrypted output, while this would be desirable in many applications like packing of encrypted values. In this sense the construction of the modulo protocol offers a new approach for privacy-preserving statistical computation.

2 Preliminaries

Throughout, we denote $[A, B) := \{A, A+1, \ldots, B-1\}$. By 'random' we implicitly mean 'uniformly randomly and independently distributed', and we denote by $x \in_R V$ the event that x is taken at random from V.

Paillier cryptosystem. Our protocol relies on the additively homomorphic cryptosystem by Paillier [20], but we consider its generalization and its threshold variant by Damgård and Jurik [8]. On input of a security parameter k, the public key consists of an RSA modulus $N = pq$ of length k, for $p = 2p' + 1$ and $q = 2q' + 1$ safe primes, and a positive integer s. We define $m := p'q'$. The secret key is a value d coprime to N^s satisfying $d = 0 \bmod m$. The message space is the ring $\mathbb{Z}_{N^s}$, and a message x is encrypted by taking an $r \in_R \mathbb{Z}^*_{N^{s+1}}$ and computing $c = (N+1)^x r^{N^s} \bmod N^{s+1}$. For the threshold decryption, d is polynomially shared among the n participants, each participant has a share d_i, and at least t participants are required to correctly decrypt a ciphertext. This

[5] Otherwise, if the value $x \bmod a$ is computed, the value $x \operatorname{div} a$ would leak.

decryption protocol operates in constant rounds and has broadcast complexity $O(nk)$. A more detailed specification of the cryptosystem can be found in [8]. Encryptions are denoted by $[\![x]\!]$.

PROOFS OF KNOWLEDGE. Our modulo reduction protocol involves zero-knowledge proofs of knowledge in order to achieve security against malicious adversaries. We use standard Σ-protocols, which can be made non-interactive using the Fiat-Shamir heuristic and are provably secure in the random oracle model. In particular, our protocol involves interval proofs in which a prover shows that a published encryption $[\![x]\!]$ encrypts a value $x \in [A, B)$. For this, one can use the protocol by Boudot [3] (refined in [19, 15]). This protocol operates in constant rounds and has broadcast complexity $O(k)$.

2.1 Multiparty Computation Gates

The proposed protocol requires several efficient gates, which will be introduced in this section. Using efficient Σ-protocols, these gates handle the malicious case efficiently. We recall that the Paillier cryptosystem is additively homomorphic, which means that given encryptions $[\![x]\!], [\![y]\!]$ and a public a, the encryptions $[\![x+y]\!] = [\![x]\!][\![y]\!]$ and $[\![ax]\!] = [\![x]\!]^a$ can be computed non-interactively.

MULTIPLICATION. Cramer et al. [5] constructed a constant round protocol for n participants to securely compute $[\![xy]\!]$ given $[\![x]\!], [\![y]\!]$. This protocol has broadcast complexity $O(nk)$.
RANDOM BIT GENERATION. Several multiparty protocols for generating random bits are known, varying between an $O(n^2k)$ broadcast complexity protocol in constant rounds [5], and an $O(nk)$ broadcast complexity protocol in $O(n)$ rounds [23].
COMPARISON GATE. On input of two encrypted bit representations $([\![x_0]\!], \ldots, [\![x_{\ell-1}]\!])$ and $([\![y_0]\!], \ldots, [\![y_{\ell-1}]\!])$, a comparison gate outputs an encrypted bit $[\![[x < y]]\!]$. An $O(\lg \ell)$ round complexity protocol [14], as well as a constant round protocol [6] are known, but the latter has a considerably higher hidden constant. Both protocols have broadcast complexity $O(nk\ell)$.

3 Random Bitwise Value Generation

The modulo reduction protocol introduced in Sect. 4 requires a sub-protocol to generate a value $r \in_R [0, a)$ in a bitwise manner. We refer to this gate as the *random bitwise value generation* protocol and we discuss such a protocol in this section. Other protocols for securely generating random values from a restricted domain are known as well [21].

Protocol 1 (Random bitwise value generation). Given a publicly known value a such that $2^{\ell_a - 1} < a \leq 2^{\ell_a}$, the following protocol generates an encrypted bit representation $([\![r_0]\!], \ldots, [\![r_{\ell_a - 1}]\!])$ of r such that $r \in_R [0, a)$. The participants $\mathcal{P}_i$ $(i = 1, \ldots, n)$ perform the following steps:

1. For $j = 0, \ldots, \ell_a - 1$, the participants jointly generate random bit encryptions $[\![r_j]\!]$ for $r_j \in_R \{0,1\}$;
2. Using a comparison gate, $[\![[r < a]]\!]$ is computed and jointly decrypted. If $[r < a] = 0$, the protocol is restarted.

Notice that in case $a \neq 2^{\ell_a}$, the number of restarts of the protocol is $2^{\ell_a}/a < 2$ on average. Using this observation, we conclude that Prot. 1 has broadcast complexity varying between $O(nk\ell_a)$ (with round complexity $O(n)$) and $O(n^2 k\ell_a)$ (in constant rounds). Correctness and security are proven in [16, Propositions 1 and 3].

4 Multiparty Modulo Reduction

We consider input $[\![x]\!]$ with $x \in \{0,1\}^{\ell_x}$ and a public value a such that $2^{\ell_a - 1} < a \leq 2^{\ell_a}$ for some ℓ_a, and construct a protocol for the computation of $[\![x \bmod a]\!]$. Without loss of generality, we assume that $\ell_a \leq \ell_x$: clearly, if $\ell_a > \ell_x$ then certainly $a > x$, in which case $x \bmod a = x$. The protocol relies on the fact that it is unnecessary to compute the ℓ_x bits of x if the modulus $a \leq 2^{\ell_a}$ is known for some $\ell_a \leq \ell_x$. As in many cases ℓ_a is relatively small compared to ℓ_x (cf. Sect. 5), this reduces the costs. We recall that we have n participants (t, n)-threshold sharing the secret key for Paillier decryption, and that the public key for the cryptosystem is (N, s). We introduce a security parameter ℓ_s, which we require to satisfy $an2^{\ell_x + \ell_s} < N^s$.

Protocol 2 (Modulo reduction). Given $[\![x]\!]$ for $x \in \{0,1\}^{\ell_x}$ and a publicly known value a, the following protocol outputs an encryption $[\![x \bmod a]\!]$. The participants $\mathcal{P}_i$ $(i = 1, \ldots, n)$ perform the following steps:

1. The participants jointly generate a random encrypted bit representation $([\![r_0]\!], \ldots, [\![r_{\ell_a - 1}]\!])$ of r such that $r \in_R [0, a)$, using Prot. 1. In parallel, each participant takes $s_i \in_R \{0,1\}^{\ell_x + \ell_s}$ and publishes $S_i = [\![s_i]\!]$ together with an interval proof of knowledge for relation $\{(S_i; s_i) \mid S_i = [\![s_i]\!] \wedge s_i \in [0, 2^{\ell_x + \ell_s})\}$;
2. Each participant individually computes

$$[\![\tilde{x}]\!] = [\![x]\!][\![r]\!]^{-1} \left(\prod_{i=1}^{n} [\![s_i]\!] \right)^a = \left[\!\!\left[x - r + a \sum_{i=1}^{n} s_i \right]\!\!\right];$$

3. Using threshold decryption the participants obtain $\tilde{x}$, and compute $\bar{x} = \tilde{x} \bmod a$;
4. Using a comparison gate the participants compute $[\![c]\!] = [\![[a - 1 - \bar{x} < r]]\!]$;
5. Each participant individually computes

$$\bmod([\![x]\!], a) = [\![\bar{x}]\!][\![r]\!][\![c]\!]^{-a} = [\![\bar{x} + r - ca]\!].$$

Notice that in phase 4 the comparison gates of Sect. 2.1 can be used, as $a - 1 - \bar{x}$ is known in plaintext, and the participants know the encrypted bit representation of r. Correctness and security are proven in [16, Propositions 2 and 4].

5 Efficiency Analysis

The modulo reduction protocol has average broadcast complexity varying between $O(nk\ell_a)$ (in $O(n)$ rounds) and $O(n^2k\ell_a)$ (in constant rounds). The absolute number of rounds highly depends on the gates used. In [6], Damgård et al. also construct a modulo reduction gate, although for verifiable secret sharing. The THC-analogue of this gate is less efficient than the one proposed here. Their protocol has a broadcast complexity varying between $O(nk\ell_x(\log \ell_x + \ell_a))$ and $O(nk\ell_x(n + \log \ell_x + \ell_a))$ (with round complexities $O(n + \ell_x)$ and $O(1)$, respectively). More importantly, in many practical applications the value ℓ_a is rather small compared to ℓ_x: consider for example a scenario where 100 millionaires want to securely compute an encryption of their average fortune. In this case $\ell_a = 7$, while $\ell_x = 37$ but needs to be extended to 47 to cover billionaires as well[6]. Note that Damgård et al.'s construction needs to compute the complete bit representation of x, while the idea of the proposed scheme relies on knowledge of the form of a.

Acknowledgments

This work has been funded in part by the European Community's Sixth Framework Programme under grant number 034238, SPEED project - Signal Processing in the Encrypted Domain, in part by the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II. The work reported reflects only the authors views; the European Community is not liable for any use that may be made of the information contained herein.

References

[1] J. Algesheimer, J. Camenisch, and V. Shoup. Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In *Advances in Cryptology - CRYPTO '02*, volume 2442 of *LNCS*, pages 417–432, Berlin, 2002. Springer-Verlag.

[2] T. Bianchi, A. Piva, and M. Barni. Efficient pointwise and blockwise encrypted operations. In *MM&Sec '08*, pages 85–90, New York, 2008. ACM.

[3] F. Boudot. Efficient proofs that a committed number lies in an interval. In *Advances in Cryptology - EUROCRYPT '00*, volume 1807 of *LNCS*, pages 431–444, Berlin, 2000. Springer-Verlag.

[4] O. Catrina and C. Dragulin. Multiparty computation of fixed-point multiplication and reciprocal. In *DEXA '09*, pages 107–111. IEEE Computer Society, 2009.

[5] R. Cramer, I. Damgård, and J. Nielsen. Multiparty computation from threshold homomorphic encryption. In *Advances in Cryptology - EUROCRYPT '01*, volume 2045 of *LNCS*, pages 280–300, Berlin, 2001. Springer-Verlag.

[6] For simplicity we assume that the fortune of a millionaire is upper bounded by one billion.

[6] I. Damgård, M. Fitzi, E. Kiltz, J. Nielsen, and T. Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In *TCC '06*, volume 3876 of *LNCS*, pages 285–304, Berlin, 2006. Springer-Verlag.

[7] I. Damgård, M. Geisler, and M. Krøigaard. Efficient and secure comparison for on-line auctions. In *ACISP '07*, volume 4586 of *LNCS*, pages 416–430, Berlin, 2007. Springer-Verlag.

[8] I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *PKC '01*, volume 1992 of *LNCS*, pages 119–136, Berlin, 2001. Springer-Verlag.

[9] W. Du and M. Atallah. Privacy-preserving cooperative statistical analysis. In *ACSAC '01*, pages 102–112. IEEE Computer Society, 2001.

[10] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *PETS '09*, volume 5672 of *LNCS*, pages 235–253, Berlin, 2009. Springer-Verlag.

[11] M. Franklin and S. Haber. Joint encryption and message-efficient secure computation. *Journal of Cryptology*, 9(4):217–232, January 1996.

[12] M. Franklin and M. Reiter. The design and implementation of a secure auction service. *IEEE Transactions on Software Engineering*, 22(5):302–312, 1996.

[13] S. From and T. Jakobsen. Secure multi-party computation on integers. Master's thesis, University of Århus, Århus, 2006. `http://www.cs.au.dk/~tpj/uni/thesis/report.pdf`.

[14] J. Garay, B. Schoenmakers, and J. Villegas. Practical and secure solutions for integer comparison. In *PKC '07*, volume 4450 of *LNCS*, pages 330–342, Berlin, 2007. Springer-Verlag.

[15] J. Groth. Non-interactive zero-knowledge arguments for voting. In *ACNS '05*, volume 3531 of *LNCS*, pages 467–482, Berlin, 2005. Springer-Verlag.

[16] J. Guajardo, B. Mennink, and B. Schoenmakers. Modulo reduction for Paillier encryptions and application to secure statistical analysis. Full version of this paper, available from the authors, 2009.

[17] A. Kiayias, B. Yener, and M. Yung. Privacy-preserving information markets for computing statistical data. In *FC '09*, volume 5628 of *LNCS*, pages 32–50, Berlin, 2009. Springer-Verlag.

[18] E. Kiltz, G. Leander, and J. Malone-Lee. Secure computation of the mean and related statistics. In *TCC '05*, pages 283–302, Berlin, 2005. Springer-Verlag.

[19] H. Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *Advances in Cryptology - ASIACRYPT '03*, volume 2894 of *LNCS*, pages 398–415, Berlin, 2003. Springer-Verlag.

[20] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238, Berlin, 1999. Springer-Verlag.

[21] B. Schoenmakers and A. Sidorenko. Distributed generation of uniformly random bounded integers. Unpublished manuscript, October 1, 2007.

[22] B. Schoenmakers and P. Tuyls. Practical two-party computation based on the conditional gate. In *Advances in Cryptology - ASIACRYPT '04*, volume 3329 of *LNCS*, pages 119–136, Berlin, 2004. Springer-Verlag.

[23] B. Schoenmakers and P. Tuyls. Efficient binary conversion for Paillier encrypted values. In *Advances in Cryptology - EUROCRYPT '06*, volume 4004 of *LNCS*, pages 522–537, Berlin, 2006. Springer-Verlag.

[24] SecureSCM. Secure computation models and frameworks. Technical Report D9.1, SecureSCM, July 2008. `http://www.securescm.org`.