Homomorphic Signatures and Message Authentication Codes

Dario Catalano

Università di Catania, Italy
catalano@dmi.unict.it

Abstract. Homomorphic message authenticators allow to validate computation on previously signed data. The holder of a dataset $\{m_1, \ldots, m_\ell\}$ uses her secret key sk to produce corresponding tags $(\sigma_1, \ldots, \sigma_\ell)$ and stores the authenticated dataset on a remote server. Later the server can (publicly) compute $m = f(m_1, \ldots, m_\ell)$ together with a succinct tag σ certifying that m is the correct output of the computation f. A nice feature of homomorphic authenticators is that the validity of this tag can be verified *without* having to know the original dataset. This latter property makes the primitive attractive in a variety of context and applications, including, for instance, verifiable delegation of computation on outsourced data.

In this short survey, I will give an overview of the state of the art in the areas of homomorphic signatures and message authentication codes. I will (briefly) describe some of the most recent results and provide an overview of the main challenges that remain to address.

1 Introduction

Imagine that Alice wants to outsource large amounts of data to some external server (to the "cloud") so that she can later delegate the server to perform computation on this data. A natural requirement in such a situation is that the server performs the computation correctly. More precisely, the server should be able to perform the prescribed computation and also be able to convince Alice that the computation has been carried out as prescribed. What makes this task non trivial are the following additional requirements: (1) Alice does not want to keep a local copy of her data (2) the communication complexity of the protocol should not depend on the (total) size of the outsourced data. This latter restriction rules out, for instance, trivial solutions in which Alice authenticates each single message in the dataset and then receives back the same dataset to re-run the computation locally.

An elegant solution to this problem comes from the notion of homomorphic signatures (and message authentication codes, in cases were verification does not need to be publicly doable). In a preliminary phase Alice signs her dataset $\{m_i\}_{i=1,\ldots,\ell}$ and stores it on the cloud together with the corresponding signatures $\sigma_i = \mathsf{Sign}(\mathsf{sk}, m_i)$. Later the server can use a (publicly available) evaluation algorithm to compute $m = f(m_1, \ldots, m_\ell)$ together with a (succinct) valid signature

M. Abdalla and R. De Prisco (Eds.): SCN 2014, LNCS 8642, pp. 514–519, 2014.

σ on it; σ validates the computation, as homomorphic signatures are required to be unforgeable. Informally, this means that adversaries that can (adaptively) see the signatures corresponding to polynomially many messages of their own choice, cannot forge valid signature for $m^* \neq f(m_1, \ldots, m_\ell)$.

Beyond security, additional requirements of homomorphic signatures are succinctness and composability. Informally, succinctness states that both fresh and "derived" signatures should be short, meaning with this that transmitting them should require much less bandwidth than sending out the original dataset. Composability requires that derived signatures should be usable as inputs to authenticate new computations. Finally, a keynote feature of homomorphic signatures is that the validity of σ can be verified *without* needing to know the original messages $m_1, \ldots, m_\ell$.

Because of their flexibility homomorphic signatures have been investigated in several settings and flavors. Examples include homomorphic signatures for linear and polynomial functions [8,9], redactable signatures [27], transitive signatures and more [30,31].

Known Realizations. The problem of realizing homomorphic message authenticators in both the symmetric (i.e. MACs) and in the publicly verifiable setting (signatures), has been the focus of many previous works. The idea of homomorphic signature was first introduced by Desmedt [17] and later refined by Johnson *et al.* [27]. Linearly homomorphic signatures were introduced in 2009 by Boneh *et al.* [8] as a tool to prevent pollution attacks in linear network coding schemes. Following this work, many results further explored this notion both in the random oracle [22,10,9,13], and in the standard model [3,14,4,15,19,5]. In the symmetric setting constructions of (linearly) homomorphic message authentication codes have been proposed by [1]. Several more recent works consider the question of supporting larger classes of functionalities. Boneh and Freeman in [9] proposed an homomorphic signature scheme for constant degree polynomials. Gennaro and Wichs [23] gave a construction of homomorphic MACs supporting arbitrary computations. This construction relies on fully homomorphic encryption and it is proved secure in a weaker security model where the adversary cannot ask verification queries. Catalano and Fiore [11] revisited this result and put forward a construction that, while capturing a less general class of functionalities (i.e. arithmetic circuits of polynomial degree), is very efficient and explicitly allows for verification queries. This latter result was further generalized by Catalano *et al.* in [12]. Finally, Catalano, Fiore and Warinschi [16], proposed a construction of homomorphic signatures for polynomial functions that improves over the Boneh-Freeman solution in three main aspects. First the scheme is proven secure in the standard model. Second, security is proven in a stronger fully adaptive setting[1]. Finally, signature verification is more efficient (in a amortized sense) than recomputing the function from scratch. Let us

[1] The solution from [9] is proven secure in a model where the adversary is required to ask all the signing queries for a given dataset at once. The scheme from [16], on the other hand, is more flexible as adversaries can ask one message at a time and even intersperse queries for messages belonging to different datasets

elaborate a bit more on this. Virtually all previous work in the area[2] proposed constructions where the cost of verifying the signature/MAC is proportional to the *description* of the function being evaluated. This means that, if one wants to check the validity of a derived signature σ for $m = f(m_1, \ldots, m_\ell)$, the cost of the verification procedure is proportional to the description of f. The solution from [16] enjoys efficient verification in the sense that verifying a signature against a function f can be done *faster* than computing f. More precisely, this holds in an amortized sense: once a first pre-computation of f is carried out locally, one can verify the evaluation of f on *any other* dataset efficiently. This feature opens the way to using homomorphic signatures for verifiable computation (e.g. [21]) and, in particular, it allows to realize simple verifiable computation schemes for outsourced data.

SNARKs. In principle one could construct (fully) homomorphic signatures using CS proofs [29] or, more in general, *succinct non interactive arguments of knowledge* (SNARKs) for NP [7]. For any given NP statement, one can use SNARKs to create a short[3] proof π that certifies knowledge of the corresponding witness. Slightly more in detail, one can create a short argument π that, given m proves knowledge of some input data set $\{m_1, \ldots, m_\ell\}$, together with corresponding signatures σ_i, such that $f(m_1, \ldots, m_\ell) = m$. The security of this construction comes from the fact that, being it an argument of knowledge, a forged signature for some function f allows to extract a (forged) signature for the underlying dataset. The main problem of (NP)-SNARKs is that they are known to require non standard assumptions [24]. In particular, known constructions either rely on random oracles [29] or on so-called "knowledge" assumptions (e.g. [25,7]).

Other related work. Recently Libert *et al.* [28] introduced and realized the notion of *Linearly Homomorphic Structure Preserving signatures* (LHSPS for short). Structure Preserving cryptography provides a simple and elegant methodology to compose algebraic tools within the framework of Groth-Sahai proof systems [26]. Informally LHSPS are like ordinary Structure Preserving Signatures but they come equipped with a linearly homomorphic property that makes them interesting even beyond their usage within the Groth Sahai framework. In particular Libert *et al.* showed that LHSPS can be used to enable simple verifiable computation mechanisms on encrypted data. More surprisingly, they observed that linearly homomorphic SPS (generically) yield efficient simulation sound trapdoor commitment schemes [20], which in turn imply non malleable trapdoor commitments [18] to group elements.

Other works considered the problem of modeling notions of privacy [9,2,4]for homomorphic signatures, so to be able to compute on authenticated data in a privacy preserving way.

[2] A nice exception is the work of Backes *et al.* [6] that introduced the notion of homomorphic MACs with efficient verification. Their scheme, while very efficient, can only support quadratic polynomials.

[3] Here by short we mean that the length of π does not depend on the size of the statement/witness.

Open Problems. Currently the main open problem in the area of homomorphic authenticators is to realize fully homomorphic signatures and message authentication codes[4]. Even more ambitious goals might be to realize fully homomorphic solutions with efficient verification, as this would allow to delegate arbitrary computations on outsourced data in an efficient, verifiable way.

Acknowledgements. I would like to thank Orazio Puglisi for his comments and all my co-authors in this exciting area of research: Dario Fiore, Bogdan Warinschi, Rosario Gennaro, Luca Nizzardo and Konstantinos Vamvourellis.

References

1. Agrawal, S., Boneh, D.: Homomorphic mACs: MAC-based integrity for network coding. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 292–305. Springer, Heidelberg (2009)
2. Ahn, J.H., Boneh, D., Camenisch, J., Hohenberger, S., Shelat, A., Waters, B.: Computing on authenticated data. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 1–20. Springer, Heidelberg (2012)
3. Attrapadung, N., Libert, B.: Homomorphic network coding signatures in the standard model. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 17–34. Springer, Heidelberg (2011)
4. Attrapadung, N., Libert, B., Peters, T.: Computing on authenticated data: New privacy definitions and constructions. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 367–385. Springer, Heidelberg (2012)
5. Attrapadung, N., Libert, B., Peters, T.: Efficient completely context-hiding quotable and linearly homomorphic signatures. In: Kurosawa, K., Hanaoka, G. (eds.) PKC 2013. LNCS, vol. 7778, pp. 386–404. Springer, Heidelberg (2013)
6. Backes, M., Fiore, D., Reischuk, R.M.: Verifiable delegation of computation on outsourced data. In: Sadeghi, A.-R., Gligor, V.D., Yung, M. (eds.) ACM CCS 2013, pp. 863–874. ACM Press (November 2013)
7. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Goldwasser, S. (ed.) ITCS 2012, pp. 326–349. ACM (January 2012)
8. Boneh, D., Freeman, D., Katz, J., Waters, B.: Signing a linear subspace: Signature schemes for network coding. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 68–87. Springer, Heidelberg (2009)
9. Boneh, D., Freeman, D.M.: Homomorphic signatures for polynomial functions. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 149–168. Springer, Heidelberg (2011)
10. Boneh, D., Freeman, D.M.: Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 1–16. Springer, Heidelberg (2011)

[4] In this sense the fully homomorphic solution by Gennaro and Wichs [23] cannot be considered fully satisfactory as it does not support verification queries

11. Catalano, D., Fiore, D.: Practical homomorphic mACs for arithmetic circuits. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 336–352. Springer, Heidelberg (2013)
12. Catalano, D., Fiore, D., Gennaro, R., Nizzardo, L.: Generalizing homomorphic mACs for arithmetic circuits. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 538–555. Springer, Heidelberg (2014)
13. Catalano, D., Fiore, D., Gennaro, R., Vamvourellis, K.: Algebraic (trapdoor) one-way functions and their applications. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 680–699. Springer, Heidelberg (2013)
14. Catalano, D., Fiore, D., Warinschi, B.: Adaptive pseudo-free groups and applications. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 207–223. Springer, Heidelberg (2011)
15. Catalano, D., Fiore, D., Warinschi, B.: Efficient network coding signatures in the standard model. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 680–696. Springer, Heidelberg (2012)
16. Catalano, D., Fiore, D., Warinschi, B.: Homomorphic signatures with efficient verification for polynomial functions. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 371–389. Springer, Heidelberg (2014)
17. Desmedt, Y.: Computer security by redefining what a computer is. In: NSPW (1993)
18. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: 23rd ACM STOC, pp. 542–552. ACM Press (May 1991)
19. Freeman, D.M.: Improved security for linearly homomorphic signatures: A generic framework. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 697–714. Springer, Heidelberg (2012)
20. Garay, J.A., MacKenzie, P.D., Yang, K.: Strengthening zero-knowledge protocols using signatures. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 177–194. Springer, Heidelberg (2003)
21. Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 465–482. Springer, Heidelberg (2010)
22. Gennaro, R., Katz, J., Krawczyk, H., Rabin, T.: Secure network coding over the integers. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 142–160. Springer, Heidelberg (2010)
23. Gennaro, R., Wichs, D.: Fully homomorphic message authenticators. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 301–320. Springer, Heidelberg (2013)
24. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC, pp. 99–108. ACM Press (June 2011)
25. Groth, J.: Short pairing-based non-interactive zero-knowledge arguments. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340. Springer, Heidelberg (2010)
26. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)

27. Johnson, R., Molnar, D., Song, D.X., Wagner, D.: Homomorphic signature schemes. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 244–262. Springer, Heidelberg (2002)
28. Libert, B., Peters, T., Joye, M., Yung, M.: Linearly homomorphic structure-preserving signatures and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 289–307. Springer, Heidelberg (2013)
29. Micali, S.: CS proofs (extended abstracts). In: 35th FOCS, pp. 436–453. IEEE Computer Society Press (November 1994)
30. Micali, S., Rivest, R.L.: Transitive signature schemes. In: Preneel, B. (ed.) CT-RSA 2002. LNCS, vol. 2271, pp. 236–243. Springer, Heidelberg (2002)
31. Yi, X.: Directed transitive signature scheme. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 129–144. Springer, Heidelberg (2006)