Secure Outsourcing in Discrete-Logarithm-Based and Pairing-Based Cryptography (Invited Talk)

Damien Vergnaud[1,2](✉)

[1] CNRS, Laboratoire d'Informatique de Paris 6,
Sorbonne Université, LIP6, Paris, France
damien.vergnaud@lip6.fr
[2] Institut Universitaire de France, Paris, France

Abstract. Cryptographic operations are performed everywhere, from standard laptop to smart cards. Some devices computational resources can be very limited and it is natural to delegate costly operations to another device capable of carrying out cryptographic algorithms. In this setting, it is obviously important to ensure the limited device that the computation is carried out correctly and that the powerful device does not learn anything about what is actually computing (including the secret inputs and outputs). We briefly review the recent advances on secure outsourcing of group exponentiation (in groups of known prime order as well as in groups of unknown order) and pairing computation.

1 Introduction

Many widely used public-key cryptographic systems and protocols relies on the (supposed) computational hardness of the discrete-logarithm or the discrete-root problems. The core operation of these cryptosystems is group exponentiation in a finite Abelian group, i.e., computing u^a from a group element u and an exponent a. Besides, since their introduction in cryptography [4,15], *pairings* proved to be an amazingly flexible and useful tool for the construction of cryptosystems with unique features (*e.g.* efficient identity based cryptography [4]). In this setting, the core operation is the computation of pairings which is the most expensive operation in pairing-based cryptographic protocols.

We consider the problem of "outsourcing" group exponentiation and pairing computation from a weak computational device to a more powerful one. Indeed, some devices computational resources can be very limited and it is natural, as most of the devices are online or directly connected to a powerful device (like a SIM card in a smart phone) to securely delegate sensitive and costly operations to a device capable of carrying out cryptographic algorithms.Outsourcing cryptographic computations is a classical problem which was formalized in [13]

O. Blazy and C. Y. Yeun (Eds.): WISTP 2018, LNCS 11469, pp. 7–11, 2019.
https://doi.org/10.1007/978-3-030-20074-9_2

by Hohenberger and Lysyanskaya. In this scenario, the powerful device[1] can, potentially, be operated by a malicious adversary and it is obviously important to ensure the limited device that the computation is carried out correctly and that the powerful device does not learn anything about what is actually computing (including the secret inputs and outputs).

2 Group Exponentiation

In the last 30 years, the question of how a computationally limited device may outsource group exponentiation to another, potentially malicious, but much more computationally powerful device has been a very active research topic (*e.g.* [3, 6,7,17,18,26]). Many solutions have been proposed and then cryptanalyzed in follow-up papers (*e.g.* [7,14,21–24]). We briefly review the recent advances on secure outsourcing of group exponentiation.

Recently, Chevalier, Laguillaumie and Vergnaud [7] proposed a taxonomy of private exponentiation delegation protocols (to a single untrusted computational resource) in groups of *known prime* order. Their taxonomy covers all the practical situations: the group element u can be secret or public, variable or fixed, the exponent a can be secret or public, and the result of the exponentiation u^a can also be either public or secret. They provided simple constructions in all different settings and proved that these protocols cannot be significantly improved if one wants to use a single untrusted computational resource and to limit the computational cost of the delegating device to a small number of (generic) group operations. Aguilar-Melchor, Deneuville, Gaborit, Lepoint and Ricosset later showed [1] that using homomorphic encryption, it is sometimes possible to reduce the computational costs for privately delegating elliptic-curve operations (but at the cost of a very large communication complexity).

Another important use case is the setting of RSA exponentiation: a device wants to delegate the computation of a signature given a public key (N, e), a public message (or hash value of a message) m and the secret signing exponent d. By outsourcing some exponentiations to a powerful device, the delegation protocol outputs a (public) signature $\sigma = m^d \bmod N$. Most proposed protocols are variants of two protocols (named RSA-S1 and RSA-S2) that were proposed by Matsumoto, Kato and Imai in 1988 [18]. Both schemes use a random linear decomposition of the RSA private exponent d. Several attacks were proposed on the protocols RSA-S1 and its variants (*e.g.* [23]). Recently, Mefenza and Vergnaud [19] proposed an improved lattice-based attack on RSA-S1 and a simple variant of this protocol that provides better efficiency for the same security level. They also presented the first attacks on the protocol RSA-S2.

[1] Hohenberger and Lysyanskaya also considered delegation protocols to two devices that are physically separated (and do not communicate) that achieve security as long as one of them is honest. Since this separation of the two devices is a strong assumption hard to be met in practice, we consider only protocols to outsource cryptographic operations to a *single* untrusted server.

A cryptographic delegation protocol that does not ensure verifiability may cause severe security problems (in particular if the computation occurs in the verification algorithm of some authentication protocol). Di Crescenzo, Khodjaeva, Kahrobaei and Shpilrain [10] proposed recently private and verifiable protocols in a large class of cyclic groups. In the presented protocols, the probability that a cheating server convinces the client of an incorrect computation result can be proved to be exponentially small (whereas previous best results could only achieve a constant probability). Their protocols need some pre-computation depending on the base u and cannot be used easily in practice if this group element is variable. The different proposals for verifiable group exponentiation where pre-computation does not depend on the base u are very inefficient and it is actually better in practice to directly perform the computation on the restricted device rather than using these solutions. A challenging problem is to study secure and verifiable outsourcing protocols for group exponentiation that covers all the practical situations as in [7].

3 Pairings

Pairings (or bilinear maps) were introduced in cryptography in 2000 by Joux [15] and Boneh-Franklin [4]. A pairing is a bilinear, non-degenerate and computable map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ where, in practice, $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups (of prime-order r) of the group of points of an elliptic curve defined over a finite field $\mathbb{F}_q$ and some finite field extension $\mathbb{F}_{q^k}$ (respectively) and the so-called *target* group $\mathbb{G}_T$ is the order r subgroup of $\mathbb{F}_{q^k}$. The pairing computation is more resource consuming compared to a scalar multiplication on the elliptic curve $E(\mathbb{F}_q)$.

In 2005, Girault and Lefranc [11] introduced the first secure pairing delegation protocol via the notion of *Server-Aided Verification*, which consists in speeding up the verification step of an authentication/signature scheme. Chevallier-Mames, Coron, McCullagh, Naccache and Scott [8,9] introduced the security notions of verifiable pairing delegation protocol and proposed the first verifiable pairing delegation protocol. Later in 2014, Canard, Devigne and Sanders [5] improved their construction and proposed a much more efficient verifiable delegation protocol. Canard, Devigne and Sanders showed that their construction is more efficient for the client than computing a pairing himself on the so-called KSS-18 curve [16]. Later, Guillevic and Vergnaud [12] showed that Canard, Devigne and Sanders protocol is actually less efficient than computing a pairing for the state-of-the-art optimal Ate pairing on a Barreto-Naehrig curve [2] and it remains open to propose an efficient verifiable delegation protocol for pairing computation on these curves.

Due to the inefficiency of the known protocols for delegation of a unique pairing, another approach is to propose efficient protocols when the client wants to compute several pairings at the same time. In 2007, Tsang, Chow and Smith [25] introduced the security notion of *batch* pairing delegation protocols and propose the first verifiable batch pairing delegation protocols when the client wants to compute several pairings $e(P_i, Q_i)$ where $P_i \in \mathbb{G}_1$ and $Q_i \in \mathbb{G}_2$ for $i \in \{1, \ldots, n\}$ and $n \geq 2$. In [20], Mefenza and Vergnaud recently proposed four new efficient batch pairing delegation protocols in different settings but it remains open to construct a generic verifiable batch pairing delegation protocol when both inputs of the pairing are variable and secret. Another interesting open problem is to provide lower bounds on the efficiency of verifiable pairing delegation protocols (as it was done in [7] for private delegation of group exponentiation).

Acknowledgments. The author would like to thank his co-authors on this active and interesting research area: Céline Chevalier, Aurore Guillevic, Fabien Laguillaumie and Thierry Mefenza. The author is supported in part by the French ANR ALAMBIC project (ANR16-CE39-0006) and the French ANR IDFIX project (ANR-16-CE39-0004).

References

1. Aguilar Melchor, C., Deneuville, J.-C., Gaborit, P., Lepoint, T., Ricosset, T.: Delegating elliptic-curve operations with homomorphic encryption. In: 2018 IEEE Conference on Communications and Network Security, CNS 2018, pp. 1–9. IEEE (2018)
2. Barreto, P.S.L.M., Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006). https://doi.org/10.1007/11693383_22
3. Béguin, P., Quisquater, J.-J.: Fast server-aided RSA signatures secure against active attacks. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 57–69. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-44750-4_5
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
5. Canard, S., Devigne, J., Sanders, O.: Delegating a pairing can be both secure and efficient. In: Boureanu, I., Owesarski, P., Vaudenay, S. (eds.) ACNS 2014. LNCS, vol. 8479, pp. 549–565. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07536-5_32
6. Cavallo, B., Di Crescenzo, G., Kahrobaei, D., Shpilrain, V.: Efficient and secure delegation of group exponentiation to a single server. In: Mangard, S., Schaumont, P. (eds.) RFIDSec 2015. LNCS, vol. 9440, pp. 156–173. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24837-0_10
7. Chevalier, C., Laguillaumie, F., Vergnaud, D.: Privately outsourcing exponentiation to a single server: cryptanalysis and optimal constructions. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016. LNCS, vol. 9878, pp. 261–278. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45744-4_13
8. Chevallier-Mames, B., Coron, J.-S., McCullagh, N., Naccache, D., Scott, M.: Secure delegation of elliptic-curve pairing. Cryptology ePrint Archive, Report 2005/150 (2005). http://eprint.iacr.org/2005/150
9. Chevallier-Mames, B., Coron, J.-S., McCullagh, N., Naccache, D., Scott, M.: Secure delegation of elliptic-curve pairing. In: Gollmann, D., Lanet, J.-L., Iguchi-Cartigny, J. (eds.) CARDIS 2010. LNCS, vol. 6035, pp. 24–35. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12510-2_3
10. Di Crescenzo, G., Khodjaeva, M., Kahrobaei, D., Shpilrain, V.: Practical and secure outsourcing of discrete log group exponentiation to a single malicious server. In: Thuraisingham, B.M., Karame, G., Stavrou, A. (eds.) CCSW@CCS 2017, Dallas, TX, USA, 3 November 2017, pp. 17–28. ACM (2017)

11. Girault, M., Lefranc, D.: Server-aided verification: theory and practice. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 605–623. Springer, Heidelberg (2005). https://doi.org/10.1007/11593447_33
12. Guillevic, A., Vergnaud, D.: Algorithms for outsourcing pairing computation. In: Joye, M., Moradi, A. (eds.) CARDIS 2014. LNCS, vol. 8968, pp. 193–211. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16763-3_12
13. Hohenberger, S., Lysyanskaya, A.: How to securely outsource cryptographic computations. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 264–282. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_15
14. Jakobsson, M., Wetzel, S.: Secure server-aided signature generation. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 383–401. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44586-2_28
15. Joux, A.: A one round protocol for tripartite Diffie–Hellman. In: Bosma, W. (ed.) ANTS 2000. LNCS, vol. 1838, pp. 385–393. Springer, Heidelberg (2000). https://doi.org/10.1007/10722028_23
16. Kachisa, E.J., Schaefer, E.F., Scott, M.: Constructing brezing-weng pairing-friendly elliptic curves using elements in the cyclotomic field. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 126–135. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85538-5_9
17. Laih, C.-S., Yen, S.-M., Harn, L.: Two efficient server-aided secret computation protocols based on the addition sequence. In: Imai, H., Rivest, R.L., Matsumoto, T. (eds.) ASIACRYPT 1991. LNCS, vol. 739, pp. 450–459. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-57332-1_38
18. Matsumoto, T., Kato, K., Imai, H.: Speeding up secret computations with insecure auxiliary devices. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 497–506. Springer, New York (1990). https://doi.org/10.1007/0-387-34799-2_35
19. Mefenza, T., Vergnaud, D.: Cryptanalysis of server-aided RSA protocols with private-key splitting, in submission (2017)
20. Mefenza, T., Vergnaud, D.: Verifiable outsourcing of pairing computations, in submission (2018)
21. Merkle, J.: Multi-round passive attacks on server-aided RSA protocols. In: Jajodia, S., Samarati, P., (eds.) ACM CCS 2000, pp. 102–107. ACM Press, November 2000
22. Merkle, J., Werchner, R.: On the security of server-aided RSA protocols. In: Imai, H., Zheng, Y. (eds.) PKC 1998. LNCS, vol. 1431, pp. 99–116. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054018
23. Nguyen, P.Q., Shparlinski, I.E.: On the insecurity of a server-aided RSA protocol. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 21–35. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_2
24. Pfitzmann, B., Waidner, M.: Attacks on protocols for server-aided RSA computation. In: Rueppel, R.A. (ed.) EUROCRYPT 1992. LNCS, vol. 658, pp. 153–162. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-47555-9_13
25. Tsang, P.P., Chow, S.S.M., Smith, S.W.: Batch pairing delegation. In: Miyaji, A., Kikuchi, H., Rannenberg, K. (eds.) IWSEC 2007. LNCS, vol. 4752, pp. 74–90. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75651-4_6
26. Wang, Y., et al.: Securely Outsourcing exponentiations with single untrusted program for cloud storage. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 326–343. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11203-9_19