Mergeable Functional Encryption

Vincenzo Iovino

University of Luxembourg, vincenzo.iovino@uni.lu

Karol Zebrowski

University of Warsaw, kz277580@students.mimuw.edu.pl

Abstract

In recent years, there has been great interest in Functional Encryption (FE), a generalization of traditional encryption where a token enables a user to learn a specific function of the encrypted data and nothing else. In this paper we put forward a new generalization of FE that we call *M*ergeable FE (mFE). In a mFE system, given a ciphertext c_1 encrypting m_1 and a ciphertext c_2 encrypting m_2, it is possible to produce in an oblivious way (i.e., given only the public-key and without knowledge of the messages, master secret-key or any other auxiliary information) a ciphertext encrypting the string $m_1||m_2$ under the security constraint that this new ciphertext does not leak more information about the original messages than what may be leaked from the new ciphertext using the tokens. For instance, suppose that the adversary is given the token for the function $f(\cdot)$ defined so that for strings $x \in \{0,1\}^n$, $f(x) = g(x)$ for some function $g : \{0,1\}^n \rightarrow \{0,1\}$ and for strings $y = (x_1||x_2) \in \{0,1\}^{2n}$, $f(x_1||x_2) = g(x_1) \vee g(x_2)$. Furthermore, suppose that the adversary gets a ciphertext c encrypting $(x_1||x_2)$ that is the result of "merging" some ciphertexts c_1 and c_2 encrypting respectively x_1 and x_2, and suppose that the token for f evaluates to 1 on c. Then, the security of mFE guarantees that the adversary only learns the output $f(x_1, x_2) = g(x_1) \vee g(x_2) = 1$ and nothing else (e.g., the adversary should not learn whether $g(x_1) = 1$ or $g(x_2) = 1$). This primitive is in some sense FE with the "best possible" homomorphic properties and, besides being interesting in itself, it offers wide applications. For instance, it has as special case multi-inputs FE and thus indistinguishability obfuscation (iO) and extends the latter to support more efficiently *homomorphic* and *re-randomizable* properties. We construct mFE schemes supporting a single merging operation, one from indistinguishability obfuscation for Turing machines and one for messages of unbounded length from *public-coin* differing-inputs obfuscation. Finally, we discuss a construction supporting unbounded merging operations from new assumptions.

Keywords: Functional Encryption, Obfuscation, Homomorphic cryptography.

1 Introduction

Functional Encryption (FE) [BSW11] is a sophisticated type of encryption that allows to finely control the amount of information that is revealed by a ciphertext. In a FE scheme, for any function f allowed by the system, the owner of the master secret key can compute a restricted key, called *token*, for f, that enables to compute $f(m)$ on a ciphertext encrypting m, and nothing else. In recent years, more expressive forms of FE were constructed in a series of works (see, e.g., [BDOP04,BW07,KSW08,LOS+10,OT12,Wat12]) culminating in the breakthrough

Short abstract submitted to Italian Conference on Theoretical Computer Science 2015 *28 May 2015*

of Garg *et al.* [GGH+13] who showed the first candidate construction of FE for all polynomial-size circuits from indistinguishability obfuscation. Another line of research investigated extensions and generalizations of FE such as multi-inputs FE [GGG+14,BLR+14], FE for randomized functionalities [GJKS13,KSY14], FE in the private-key model [SSW09,BS14] and in alternative models [BRS13a,BRS13b,BIP10]. While these works offer unique applications and pose new insights and challenges, we call for the need for a new and further generalization of FE not previously discussed in the literature.

1.1 Mergeable functional encryption.

We put forward the concept of mergeable FE (mFE) scheme. A mFE scheme is identical to a FE scheme but in addition it is endowed with a Merge algorithm that given two ciphertexts c_1 and c_2 encrypting respectively m_1 and m_2 can produce a ciphertext c encrypting $m_1||m_2$ where "'$||$ represents the concatenation symbol, i.e., can *merge* the original ciphertexts, in an oblivious way without knowledge of the underlying plaintexts. As special case, a mFE also allows to *update* a ciphertext in an oblivious way. That is, having a ciphertext encrypting an unknown plaintext m, the system allows to produce a ciphertext c encrypting $m||m_2$ where m_2 is any plaintext. Notice that a mFE system represents in some sense a FE scheme with the best possible *homomorphic* properties. Indeed, it is easy to see that a FE scheme can not be in general fully homomorphic: for instance, the token for the function $f(\cdot)$ such that $f(x) = f(y)$ for some messages x, y but $f(g(x)) \neq f(g(y))$ for some function $g(\cdot)$, allows to distinguish whether a ciphertext c encrypts x or y by homomorphically evaluating the function $g(\cdot)$ on c. Instead, the restricted form of homomorphism allowed by mFE preserves and does *not* contradict its functional properties. Apart from being interesting in itself, the applications of mFE and the settings where it can be applied are vast and we will illustrate few of them.

1.2 Applications

To show the power of mFE we present some applications of it, but due to space constraints we defer others to the full version of this work [IZ15].

1.3 mFE implies Multi-inputs FE.

The works of [GGG+14,GKL+13,GGJS13] introduced the concept of multi-inputs FE (MI-FE), a generalization of FE where a token corresponds to a multi-variate function that takes multiple ciphertexts as input. In these works, several settings were defined. mFE implies MI-FE[1]: to evaluate a multi-variate token on multiple ciphertexts it is sufficient to merge the ciphertexts. Furthermore, mFE generalizes previous works on MI-FE to functions with *unbounded arity*, i.e., not putting any bound on the arity of the allowed functions. We stress that as special case mFE ssupporting a *single* merging operation implies 2-inputs MI-FE.

[1] This holds for the public-key setting where the adversary is given the public-key that allows to encrypt messages corresponding to any dimension

1.4 mFE implies CCA1-secure PKE with homomorphic properties.

It is well known that a PKE scheme can not be fully homomorphic (see also Prabhakaran and Rosulek [PR08] for alternative models). Notwithstanding it is reasonable to ask whether it can be CCA1-secure. Loft *et al.* [LMSV12] present a construction of somewhat homomorphic encryption from special knowledge assumptions. From mFE it is possible to derive CCA1-secure PKE with special homomorphic properties. Recall that FE (specifically, identity-based encryption) implies CCA1-secure public-key encryption (PKE) [CHK04,BCHK07]. If the underlying FE scheme used in the construction of CCA1-secure PKE is in particular a mFE scheme, the resulting PKE scheme would be a CCA1-secure PKE scheme supporting merging operations. Specifically, similarly to the transformation of Canetti *et al.* [CHK04], the CCA1 encryption of message m consists of a mFE ciphertext that encrypts (id, m) for a random identity id. mFE allows to merge such a ciphertext with others without the knowledge of the underlying message m.

1.5 Applications to searching over encrypted databases in a public-key setting.

One of the most notable applications of FE is to searching over encrypted databases in a cloud computing setting. In this scenario, Alice, the manager of a company, can distribute a token for a function f to any of her employees who can use such tokens to perform queries over encrypted databases located in a cloud server. For instance, one database D_1 is produced by Bob and sent to the cloud server in an encrypted form under the public-key of Alice. Similarly, Eve produced her database D_2 and sent it to the cloud server in the same encrypted form. The two databases could contain information about products sold, respectively, by the companies of Bob and Eve and of interest for the company of Alice. Moreover, we assume that the cloud servers own a lot of computational power and space but are not trusted by Bob and Eve, i.e., Bob and Eve wish to leak as few information as possible to the servers. For simplicity, suppose that the databases are implemented as lists of elements, let us say $D_1 = (x_1||\ldots||x_n)$ and $D_2 = (y_1||\ldots||y_n)$. An employee of the company of Alice could be interested in searching whether a specific product x is in *one* of the two databases but he does not care in which one it is in. Thus, the employee sends to both servers a token for the function $f_x(D) \overset{\triangle}{=} 1$ iff the list D contains x. The server evaluates the token on *both* encrypted databases one at time and then communicate to the employee whether the requested product x is or is not in the databases. Suppose that at some point there is a commercial agreement between the companies of Bob and Eve and as result of it they decide to merge their respective data without compromising the needs of the company of Alice and to keep on storing the merged encrypted data in the same cloud server. One solution could be to reveal to each other their data and re-encrypt anything under the public-key of Alice. However, this is not a valid solution as they wish to preserve the confidentiality of their own data. Another approach could be to store on the server just the concatenation of the previous ciphertexts. That is, if c_1 is the encryption of D_1 and c_2 is the encryption of c_2, then the new encrypted data could just consist of $c_1||c_2$. Anyway, recall that Bob and Eve wish to hide to the cloud server the contents of the new encrypted database. If the new encrypted database was just the concatenation of the encryptions of D_1 and D_2, then from a query for a product x, the server could figure out whether x was in the database of Alice or in the database of Bob by running the token separately on c_1 and c_2. Thus, this solution is not satisfactory. Another approach could to make Alice to create a *new* FE system and to ask the server to merge the two encrypted data under the *new* Alice's public-key. This solution incurs in a lot of problems as well. First of all, it requires a work from the Alice's side. Instead, Bob and Eve would like to merge their data without involving Alice's company (after all, recall that the employees of Alice are interested in searching whether a specific product is or is not in *one* of the encrypted databases and *not* also on which one it is in). That is, in the scenario we envision, Alice consents the companies with which she collaborates with (using their encrypted data) to merge their own encrypted data without even informing her, i.e., in a *non-interactive* way. Furthermore, the size of the public-keys and parameters would grow (as the new system is based on the old one) and the main drawback is that Alice would have to re-compute and re-distribute *new* tokens to each employee. Instead, mFE offers a valid solution to this problem: if the databases are encrypted with a mFE system, then Bob and Eve can merge their own encrypted databases in an oblivious way hiding any information on the original databases to the server.

1.6 Applications to updating encrypted data in a private-key setting.

In a private-key setting, Alice delegates her encrypted data D to a powerful cloud server and at any point she can perform a search or any computation on the data sending to the server a token for the desired function. Moreover, Alice can compute this token from a device with a low computational power as a mobile phone in which the original data is *not* present. Suppose now that Alice needs to add a file x to her encrypted data. That is, she wishes to update her encrypted data so that it now should encrypts $D||x$. As before, Alice does not consider satisfactory a solution where a new ciphertext encrypting x is added in the server. For concreteness, suppose that Alice wishes to compute the following function: $f(D||x) = g(D) \vee g(x)$ on the encrypted data and suppose that we adopt the trivial approach in which the server stores the two ciphertexts, and suppose that Alice sends to the server the two tokens, one for $g(D)$ and one for $g(x)$. Then, the server would learn whether $g(D) = 1$ or $g(x) = 1$ whereas Alice wishes the server to only learn $f(D||x)$ and nothing else. mFE offers a valid solution to this problem: Alice could send the encryption of the new data to the server asking the server to merge it with the old encrypted data. One could object that the solutions is not satisfactory because the server could keep on storing *both* ciphertexts, the one encrypting D and the one encrypting x in addition to the one encrypting $D||x$, so to be able to leak the undesired information (e.g., $g(D)$ and $g(x)$). This problem is easily fixed in the following way. Alice can update on the server the encrypted data $\$x$, where \$ is a special symbol not present in D or x. The functions f's for which she will compute tokens afterwards will be defined so to check (1) whether the input contains the special symbol in the middle (i.e., the functions will check whether the input y has the form $D\$x$ for some strings D and x) and (2) for inputs not satisfying this condition, the function is defined to output an error $\perp$. In this way, the server can only use the new tokens on the updated encrypted data and not on the previous ones.

1.7 mFE: properties and security

1.7.1 The requirements of a mFE scheme.

In view of the above and further applications we desire mFE systems to satisfy the following properties:

- The operation of merging two ciphertexts c_1 and c_2 can be performed having just the public-key of the system, c_1 and c_2.
- The size of the merged ciphertext should be proportional to the sum of the lengths of the old ciphertexts plus an *additive* factor polynomial in the security parameter. That is, suppose that c_1 has size m_1 and c_2 has size m_2 and let k be the security parameter. Then, the the result of merging c_1 and c_2 must be a ciphertext of length $O(m_1+m_2+k)$. This is to rule out solutions where the ciphertext resulting from the merge has for instance length $k \cdot (|m_1| + \cdot|m_2|)$ that would bound the number of mergings to be logarithmic (or less) in the security parameter. We call this requirement *compactness*.
- The systems must be designed for the Turing Machine (TM) model of computation as the circuit model does not fit well with the mFE setting. In fact, circuits can compute over a fixed number of bits, and thus, even though the system allowed multiple merging operations, a token could be used only on ciphertexts resulting from a bounded number of merging operations.

Our main solution (see the full version of this work [IZ15]) only support a single merging operation. In the full version we also discuss a construction supporting unbounded messages from new assumptions. Anyway, we stress that also a mFE supporting single merge is already sufficient for most of our applications like the implications of MI-FE, iO, and all other applications when limited to a single operation. Furthermore, it is really trivial to generalize our constructions for single merging operation to support a *bounded* number q of merging operations with the drawback of having parameters growing as q. mFE schemes supporting only q merging operations are sufficient to build $(q-1)$-inputs MI-FE schemes and homomorphic obfuscators supporting $q-2$ merging operations.

1.7.2 Our security notion.

We adopt as security definition an indistinguishability-based (IND) one. Recall that in the standard IND-Security game for FE an efficient adversary can output two challenge messages m_0 and m_1 and can ask any token for a function f that does not allow to distinguish the two messages, i.e., such that $f(m_0) = f(m_1)$. This is to avoid trivial attacks. In mFE this constraint is not sufficient. In fact, it could be that $f(m_0) = f(m_1)$ but $f(m||m_0) \neq f(m||m_1)$ allowing the adversary to distinguish by merging the challenge ciphertext with a ciphertext encrypting m. Notice that this situation is similar to the case of MI-FE. Therefore, we change the definition in the obvious way, by generalizing the above constraint to take in account any sequence of messages that "extends" the challenge messages (in poor words, to any message that has the challenge message as substring). More formally we also allow the challenge to be a pair of sequences (s_0, s_1) of merging operations with the same length and structure. Furthermore, we relax the security to the the *selective* model where the adversary has to declare its challenge at beginning of the game, i.e., before receiving the public-key. This is because, as standard in many works in the area, the selective model simplifies the security reductions. However, we remark that the selective model is sufficient to imply iO, selectively IND-Secure MI-FE, and the kind of homomorphic CCA1-secure PKE described before. Details on our security notion are given in the full version.

1.8 Constructions

Due to space constraints, we defer to [IZ15] for the full version containing the constructions of a mFE scheme and further applications.

References

[BCHK07] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

[BDOP04] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.

[BIP10] Carlo Blundo, Vincenzo Iovino, and Giuseppe Persiano. Predicate encryption with partial public keys. In Swee-Huay Heng, Rebecca N. Wright, and Bok-Min Goi, editors, *CANS 10: 9th International Conference on Cryptology and Network Security*, volume 6467 of *Lecture Notes in Computer Science*, pages 298–313, Kuala Lumpur, Malaysia, December 12–14, 2010. Springer, Berlin, Germany.

[BLR$^+$14] Dan Boneh, Kevin Lewi, Mariana Raykova, Amit Sahai, Mark Zhandry, and Joe Zimmerman. Semantically secure order-revealing encryption: Multi-input functional encryption without obfuscation. Cryptology ePrint Archive, Report 2014/834, 2014. http://eprint.iacr.org/.

[BRS13a] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 461–478. Springer, 2013.

[BRS13b] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private subspace-membership encryption and its applications. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 255–275. Springer, 2013.

[BS14] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. Cryptology ePrint Archive, Report 2014/550, 2014. http://eprint.iacr.org/.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273, Providence, RI, USA, March 28–30, 2011. Springer, Berlin, Germany.

[BW07] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Berlin, Germany.

[CHK04] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.

[GGG$^+$14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, volume 8441 of *Lecture Notes in Computer Science*, pages 578–602. Springer, 2014.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 40–49. IEEE Computer Society, 2013.

[GGJS13] Shafi Goldwasser, Vipul Goyal, Abhishek Jain, and Amit Sahai. Multi-input functional encryption. Cryptology ePrint Archive, Report 2013/727, 2013. http://eprint.iacr.org/.

[GJKS13] Vipul Goyal, Abhishek Jain, Venkata Koppula, and Amit Sahai. Functional encryption for randomized functionalities. Cryptology ePrint Archive, Report 2013/729, 2013. http://eprint.iacr.org/.

[GKL$^+$13] S. Dov Gordon, Jonathan Katz, Feng-Hao Liu, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. *IACR Cryptology ePrint Archive*, 2013:774, 2013.

[IZ15] Vincenzo Iovino and Karol Zebrowski. Mergeable functional encryption. Cryptology ePrint Archive, Report 2015/103, 2015. http://eprint.iacr.org/.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.

[KSY14] Ilan Komargodski, Gil Segev, and Eylon Yogev. Functional encryption for randomized functionalities in the private-key setting from minimal assumptions. Cryptology ePrint Archive, Report 2014/868, 2014. http://eprint.iacr.org/.

[LMSV12] Loftus, May, Smart, and Vercauteren. On cca-secure somewhat homomorphic encryption. In Ali Miri and Serge Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 55–72. Springer Berlin Heidelberg, 2012.

[LOS$^+$10] Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 62–91, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[OT12] Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 591–608, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany.

[PR08] Manoj Prabhakaran and Mike Rosulek. Homomorphic encryption with CCA security. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008: 35th International Colloquium on Automata, Languages and Programming, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 667–678, Reykjavik, Iceland, July 7–11, 2008. Springer, Berlin, Germany.

[SSW09] Emily Shen, Elaine Shi, and Brent Waters. Predicate privacy in encryption systems. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 457–473. Springer, Berlin, Germany, March 15–17, 2009.

[Wat12] Brent Waters. Functional encryption for regular languages. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 218–235, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany.