Quantum attacks against iterated block ciphers

M. Kaplan*

Abstract

We study the amplification of security against quantum attacks provided by iteration of block ciphers. In the classical case, the Meet-in-the-middle attack reduces the time required to break double iterations to only twice the time it takes to attack a single block cipher. Here, we prove that for quantum adversaries, two iterated ideal block ciphers are more much difficult to attack than a single one. We quantize the Meet-in-the-middle attack and use tools from quantum complexity theory to prove that it is optimal. We then quantize a technique against 4-encryption called the dissection attack. Contrary to the classical cas, this quantum attack has a better time complexity than a quantum Meet-in-the-middle attack. It also shows that the resistance against quantum attacks decreases when the number of iteration grows.

1 Introduction

Quantum information processing has deeply changed the landscape of classical cryptography. In particular, cryptosystems based on integer factoring and discrete logarithm are known to be completely insecure against quantum computers. This opened the field of *post-quantum cryptography*, which tries to restore the security of classical cryptosystems against quantum attacks. Most research is devoted to public key cryptography and the common belief is that symmetric cryptography is not affected by quantum computing because security can be amplified by increasing key sizes.

This belief is based on the fact that symmetric cryptosystems are usually subject to generic attacks whose quantumization allow only polynomial speedups. However, attacking realistic, complex cryptosystems may require more effort than just applying basic quantum algorithms and understanding precisely the security against quantum attacks may require careful analysis [4]. Our work aims to show that the tools developed to understand quantum speedups can find fruitful applications in cryptographic settings. Specifically, we use quantum walk algorithms [3] to design quantum attacks and the generalized adversary method [8] to prove security results.

We focus here on one of the most fundamental situation in symmetric cryptography: block cipher encryption: a message m is cut into blocks of fixed size, and each block is encrypted using a permutation specified by a secret key. Block cipher encryption is widely used in practice, and is also an important building block for other cryptographic primitives.

We work at an abstract level and consider a block cipher as a collection of random permutations. The set of permutations is public, and anyone can efficiently compute $F_i(X)$ and $F_i^{-1}(X)$ where F_i is the permutation specified by the secret key i and X is a block. We consider an attacker that knows a few pairs of plaintext with corresponding ciphertexts, all encrypted with the same key. Its goal is to recover the secret key that was used for encryption.

Although increasing the key length is a neat theoretical answer, it may not be possible to implement starting from a specific block cipher with fixed parameters. The question of security amplification was raised when brute force attacks against the DES block cipher became realistic [6, 9]. A simple attempt to increase the key size is to compose permutations with independent keys. For double encryption, the size is doubled, but there is a clever attack against this construction. Suppose that an attacker knows a pair of plaintext-ciphertext (P, C). These satisfy $C = F_{k_2}(F_{k_1}(P))$, where (k_1, k_2) are the keys used for encryption. Since inverse permutations can be computed, an attacker can construct tables $F_k(P)$ and $F_{k'}^{-1}(C)$ for every possible keys k, k'. Finding a collision $F_{k_1}(P) = F_{k_2}^{-1}(C)$ reveals the keys used for encryption.

*LTCI, Télécom ParisTech, 23 avenue dItalie, 75214 Paris CEDEX 13, France

This attack, known as the Meet-in-the-middle attack, shows that it only takes twice more time to attack double iterations than it takes to attack a single one. A naive cryptographer would expect here a quadratic improvement. Of course, this is optimal up to a factor two. The Meet-in-the-middle attack shows that even a simple idea like security amplification by iteration should be carefully studied [2]. It also has practical consequences, and led to the standardization of triple-DES rather than the insecure double-DES.

2 Optimality of the quantum Meet-in-the-middle

We first study the problem of 2-encryption. The 2-Key Extraction ($KE_2^{P,C}$) problem with $P, C \in [M]$ takes input $\mathcal{F} = \{F_1, \ldots, F_N\}$, a collection of permutations of the block space $[M]$ with the promise that there exists a unique couple (k_1, k_2) such that $F_{k_2}(F_{k_1}(P)) = C$. The task is to output the pair (k_1, k_2). It is easy to prove that the complexity of the problem is independent of the pair (P, C), and we drop the exponent and write only KE_2 when it is unnecessary.

The problem is easily seen to reduce to the Element Distinctness problem for which an algorithm and a matching lower bound are known [1, 3]. The reduction immediately give a $O(N^{2/3})$ upper on time on both time and memory, where N is the size of the key space. Our main result is to prove that this attack is optimal to extract the keys.

Theorem 1. *A quantum algorithm that solves KE_2 needs $\Omega(N^{2/3})$ quantum queries to the input $\mathcal{F} = \{F_1, \ldots, F_N\}$, including queries to inverse permutations.*

A commonly used measure of efficiency of classical attacks is the time-space tradeoff. For example, a classical exhaustive search and a classical Meet-in-the-middle attack both have N^2 time-space tradeoff. A corollary of our result is that the most time-efficient quantum attack has a time space tradeoff $O(N^{4/3})$ whereas a quantum exhaustive search has a tradeoff $O(N)$.

We don't use a reduction from Element Distinctness to prove a lower bound on the complexity of KE_2. These two problems have important structural differences. Firstly, it is possible to query a permutation F_i on any input $X \in [M]$, leading to a lot more inputs. Secondly, it is possible to query inverse permutations F_i^{-1}. These are strong objections against the possibility of constructing a query-preserving reduction.

Instead, we use the generalized adversary matrix, a method leading to an algebraic characterization of quantum query complexity. Starting from a matrix Γ_{ED} for Element Distinctness, we build an adversary matrix for KE_2 that has two important properties. Firstly, it has a tensor product structure, which allows to express its norm as a function of $\| \Gamma_{ED} \|$. Secondly, we show that, up to a permutations of rows and columns, the set of queries that optimally distinguish the inputs to KE_2 are those that projects onto a query to an Element Distinctness input. Roughly speaking, these are queries of the form $F_i(P)$ and $F^{-1}(C)$ only.

We measure the gain achieved by quantum algorithms with quantities of the form $\log C / \log Q$, where C is a classical complexity measure and Q is its quantum counterpart. We compare gains for exhaustive search and Meet-in-the-middle attacks. The gain for exhaustive search is 2, whereas the gain with the quantum Meet-in-the-middle is 1.5. Since the quantum Meet-in-the-middle has lower gain than a brute-force attack, one may conclude that security amplification by iteration resists better to quantum attacks than to classical ones.

We can also compare with the amplitude amplification based algorithm for Element Distinctness of [5]. Compared to the classical Meet-in-the-middle, this algorithms leads to a gain of 4/3 if time and 8/5 is time-space. An attacker can pay with time in order to get a better time-space tradeoff.

3 Quantum attack against 4-encryption

We then study the case of four successive encryptions. The 4-Key Extraction ($KE_4^{P,C}$) problem with $P, C \in [M]$ takes input $\mathcal{F} = \{F_1, \ldots, F_N\}$, a collection of permutations of $[M]$. The goal of the problem is to output (k_1, k_2, k_3, k_4) such that $F_{k_4}(F_{k_3}(F_{k_2}(F_{k_1}(P)))) = C$.

We give an explicit attack against 4-encryption. This algorithm is a quantized version of the $Dissect_2(4,1)$ algorithm of Dinur, Dunkleman, Keller and Shamir [7]. In the classical setting, this algorithm achieves the best known time-space tradeoff for 4-encryption. The basic idea of the dissection attack is to make an exhaustive search of the intermediate value after two encryptions. The candidate values are then checked using a Meet-in-the-middle procedure. The next result assumes that M and N are of comparable sizes and is stated as a function of N.

Theorem 2. *There exists a quantum algorithms that solves KE_4 in time $O(N^{7/6})$ and using memory $O(N^{2/3})$. The time-space tradeoff for this attack is $O(N^{11/6})$.*

This quantization leads to a gain of $12/7$ in time and $18/11$ in time-space. Although we have no indication that this attack is optimal, it is much more efficient than applying a Meet-in-the-middle attack treating each pair of key as a single one. Applying the quantized dissection attack gives both better performances and gain.

4 Conclusion

We have studied security amplification by iterative block ciphers. In the case of double encryption, we have found that the gain in time of the best quantum algorithm is lower than the quantization of an exhaustive search. This indicates that double iteration resists better to quantum attacks than to classical ones. Moreover, the most time-efficient attack has a worse time-space tradeoff than an exhaustive search.

We have then studied the case of 4-encryption, for which we have given a quantized version of the dissection attack. Although we don't know if this attack is optimal, it leads to better performances and gains than a Meet-in-the-middle attack. This indicates that increasing the number of iterations may in fact decrease the resistance against quantum attacks.

Although the general question of security amplification by iteration remains opened, our work shows that the tools from quantum computing, such as quantum walks and the generalized adversary method, are well suited to tackle cryptographic questions.

The full version of the paper is available at the address `http://arxiv.org/abs/1410.1434`.

References

[1] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595 – 605, 2004.

[2] W. Aiello, M. Bellare, G. Di Crescenzo, and R. Venkatesan. Security amplification by composition: The case of doubly-iterated, ideal ciphers. In *Advances in Cryptology — CRYPTO*, 1998.

[3] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37:210–239, 2007.

[4] D. Bernstein. Grover vs. McEliece. In *Proc. of the Third international conference on Post-Quantum Cryptography (PQCrypto'10)*, pages 73–80, 2010.

[5] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34, 2005.

[6] W. Diffie and M.E. Hellman. Exhaustive cryptanalysis of the NBS data encryption standard. *Computer*, 10(6):74–84, 1977.

[7] I. Dinur, O. Dunkelman, N. Keller, and A. Shamir. Efficient dissection of composite problems, with applications to cryptanalysis, knapsacks, and combinatorial search problems. In *Advances in Cryptology – CRYPTO*, 2012.

[8] P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proc. of 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 526–535, 2007.

[9] R. Merkle and M. Hellman. On the security of multiple encryption. *Communication of the ACM*, 1981.