Final Report

Geng Lin, Bibhusa Rawal

Introduction

In this project, we conduct a survey on quantum attacks on symmetric key classical cryptographic systems. Motivated by the importance of cryptography and the works that break some classical cryptographic constructs with quantum computers, we survey the security of classical symmetric key systems against adversaries with quantum computing access.

Our survey can be broken into two parts: quantum key recovery of symmetric block ciphers, and quantum attacks on symmetric cryptosystems. In the first part, we discuss the quantum speedup on brute-force key finding with Grover's algorithm proposed by Grassl et al. [1] and Zhong et al. [2]. We then present the work of Martin et al. [3], which assumes some side channel knowledge, and uses Grover's to accelerate a key search with structural information. Lastly we introduce the related-key attack model and a quantum variant proposed by Roetteler and Steinwandt [4].

In the second part, we summarize the works of Kaplan et al. [5] on how quantum period finding can be applied to break CBC-MAC and other authentication modes of block cipher, as well as Alagic et al. [6]'s discussion on a potential defense against these attacks based on the hidden shift problem. We also discuss Zhangdry's work [7] on post-quantum security of random-oracle based functions, especially quantum indifferentiability of hash functions.

In the rest of this report, we describe our motivation, goals and findings in detail.

Motivation

Cryptography is one of the building blocks of the Internet that protects privacy and ensures security. Almost all online activities today, from instant messaging to online shopping, relies on various cryptographic systems to perform authentication, ensure data confidentiality and verify data integrity. These standard protocols and algorithms often rely on mathematic constructs that are computationally difficult for an adversary to break. Since most of them were developed when quantum computing was far from being practical, their proofs of difficulty assume adversaries in a classical context. However, in recent years, the developments in quantum computers and the researches to utilize the quantum computing model are imposing threats to classical security. Famous examples include Shor's algorithm [8] that breaks big number factoring, a difficult problem on classical computers, and Grover's algorithm [9] which accelerate brute-force searching by a square root factor. Therefore, it is natural to ask whether the well-established cryptographic systems are capable of resisting the attacks from adversaries with quantum computing power. Which of them are still safe, and what needs to be redesigned? Questions like these give rise to our survey project.

We focused on symmetric key systems for two reasons. One is that due to the limited time, it is not possible to cover every aspect of cryptography. The other is that, symmetric systems are significant because they are the work horses in most data encryption standard, such as TLS and OpenPGP. Symmetric key block ciphers are also widely used to create message authentication codes. Other parts of classical cryptography, including the asymmetric systems, are left for the future.

Goals

The goal of this project is to investigate the state-of-the-art quantum attacks on symmetric key classical cryptographic systems, as well as possible defenses. Based on the difficulty and efficiency of the proposed attacks, we will try to analyze and summarize the security of a variety of classical symmetric key systems against quantum adversaries, and try to answer these questions:

1. How efficient can quantum computers break block ciphers, especially AES?
2. Which cryptosystems are broken by quantum adversaries, and what are their underlying assumptions?
3. Is it possible to fix the broken systems, i.e. are new designs necessary?

At the same time, we will find out what problems remain open in these works.

Findings

Our survey focuses on works that proposes or analyzes quantum attacks on classical symmetric-key systems. Specifically, we considered two targets: the symmetric block ciphers, and the symmetric cryptosystems built upon them.

For the block ciphers, we learned about the practical quantum complexity to recover the keys of two popular standards, AES and 3-DES [1, 2], which shows that only AES-256 provides a good post-quantum guarantee and that both AES-128 and 3-DES are no longer secure against brute-force attacks in a quantum era. Considering the fact that brute-forcing is the slowest way to recover a key, we also studied [3] which accelerates the process with side channel advices. Since real world cryptosystem implementations are never perfect, their work shows a practical utilization of side channel knowledge with the structure-unaware Grover's algorithm. Lastly, we found a quantum version of the powerful related-key attack model proposed in [4]. Despite its impractical assumptions, it can be informative to cryptosystem designers of what should be avoided.

For symmetric cryptosystems, our findings are related to message authentication codes (MACs) and random oracles. We have learned how quantum period finding can be applied to break common constructs of message authentication codes that are based on symmetric key block ciphers, such as CBC-MAC and PMAC [5]. The authors showed how Simon's algorithm can solve real world problems. Based on their work, [6] has proposed modifications to the CBC-MAC and other similar systems. This work also explains how hidden subgroup and hidden shift problems can be connected to resistance to quantum attacks. We have also studied [7], which shows how it is possible to record quantum queries and build hash functions that are indifferentiable from random oracles.

Applying Grover search to AES and 3-DES

We start by discussing attacks to symmetric block ciphers with quantum computers. In common attack models, the adversary has some blocks of plaintext and their corresponding ciphertext. Their goal is to uncover the secret key used in encryption. A more powerful attacker is able to choose the plaintext to be encrypted, thus performing a chosen plaintext attack (CPA). However, for modern encryption, CPA does not give more advantage than knowing some pairs in terms of key recovery. Classically, there isn't any algorithm that breaks the advanced encryption standard (AES) much faster than brute-force search does. Quantumly, it is straightforward to accelerate the key searching problem with Grover's algorithm.

Theoretically, Grover's algorithm can be used to find the secret key of symmetric algorithms, provided the quantum oracles for encryption. It is then natural to ask whether these oracles are feasible to build. In

	depth	#qubits	Grover depth	Grover #qubits
AES-128	$2^{16.8}$	984	2^{81}	2953
AES-192	$2^{16.6}$	1112	2^{113}	4449
AES-256	$2^{17.0}$	1336	2^{145}	6681

Table 1: AES depth and qubit estimation

this survey, we learn the circuit complexity of the oracle for two ciphers, and also the precise complexity to break them with Grover's algorithm. Our choices of ciphers are AES and 3-DES for their wide application.

AES is the most common cipher choice for HTTPS. Its key size can be one of 128-bit, 192-bit and 256-bit. Classically, brute forcing an AES key of k bits would require about $O(2^k)$ operations[1]. Grassl, et al.'s work in 2015 [1] was the first to discuss the quantum implementation of AES in detail. They built a reversible circuit for AES of different key sizes with the T and Clifford gates. The circuit was constructed by implementing the four basic AES operations, and using them to achieve key expansion and encryption rounds. During the process, the authors favored optimizing the number of qubits than gate complexity. For example, they proposed two designs for `SubBytes`, and chose the 9-qubit, 9695 T-gates and 12631 Clifford gates version instead of the 40-qubit, 3584 T-gates and 4569 Clifford gates one. They then gave a detailed estimation of the number and depth of gates, as well as the number of qubits required. We summarize their results in Table 1. As the authors argued, to characterize the secret key uniquely, $r = 3$, 4 and 5 known plaintext-ciphertext pairs are required for 128, 192 and 256-bit AES respectively. Thus, the total number of required qubits is multiplied by r for Grover's algorithm. One extra qubit is added to indicate the comparison result. Since AES key length only affects the key expansion stage, the circuit depth for the three variants are similar. It worth noting that the depth of 192-bit AES is actually smaller than the 128-bit one because of the additional available space. The depth of running Grover's algorithm is simply the depth of one cipher computation multiplied by $\lfloor \frac{\pi}{4} 2^{\frac{\text{key length}}{2}} \rfloor$.

It can be seen from the results that with quantum computers, AES-128 barely meets the legacy 80-bit security strength requirement. However, it fails to achieve today's standard [10] which is 112 bits. AES-192, while providing 112-bit security at the first glance, may not be ideal if circuits with less depth and more qubits are designed. Fortunately, AES-256 still provide good security against brute-force attack in the post-quantum era.

3-DES is a common but retiring cipher. It is also part of the MSCHAPv2 protocol, which is used by the WPA2-Enterprise authentication process. 3-DES has three keys, each of which are 56 bits long, resulting in a 2^{168} key space. However, due to the meet-in-the-middle attack, the cipher only provides 112-bit security in a classical context. Zhang and Bao [2] presented an analysis of applying the meet-in-the-middle attack in a quantum setting. Their work showed that the keys of a three-key 3-DES cipher can be recovered with $O(2^{56})$ calls to an oracle. Although we have not been able to find a circuit construction for DES, we conjecture that the depth of such a circuit is smaller than that of AES, due to the reduction in complexity. As such, 3-DES provides no more than $(56 + 17) = 73$-bit security in a post-quantum setting, and should be considered insecure.

Classically, the security of symmetric key ciphers rely on obscurity rather than computationally difficult mathematical constructs. Unlike many asymmetric constructs such as RSA, there is no rigid mathematical proof of their security. They are considered safe because despite decades of work, there has been no practical and efficient attacks that recover the keys of AES and 3-DES better than brute-force search. Unfortunately, the situation has not been changed by the availability of quantum computation model. To our best knowledge, no work has been proposed to attack a specific cipher with quantum computers by leveraging some properties

[1]Although more efficient attacks exists, they only improve k by a constant less than 2, which does not make a big difference for our analysis. Existing related-key attacks are impractical and thus not considered here.

Value	k_1	k_2
00	0	1
01	1	1
10	2	0
11	1	2

Table 2: Side channel advice example, $m = 2, l = 2$

of its construct. It appears that brute-force searching, albeit being possible to accelerate as we will discuss in the next section, is still the only known attack up to today. It remains unknown whether an efficient quantum attack exists, or whether it is not possible for quantum computers to do better than classical ones in this scenario.

Quantum key search with side channel advice

As the previous section shows, Grover's algorithm undermines the security of AES-128 with brute-force key search. In reality, it is often the case that due to implementation limitations, some information about the key can be obtained by an adversary by, for example, measuring the time and power it takes to encrypt or decrypt the data. These attacks, that do not directly target the algorithms but their implementations, are referred to as side channel attacks. The information about the key is therefore called side channel advice. Martin et al. [3] proposed their key search algorithm with this knowledge which is accelerated with Grover's.

Mathematically, a side channel advice can be formulated as a probability distribution of the keys. This is structural information about the key space. In order to utilize this information with Grover's algorithm, which is unstructured search, [3] breaks down the key search problem in smaller key spaces. The high level idea is to search first in a subspace of most likely keys, and move to less likely ones if no key is found.

[3] models the side channel information and key search in the following way. First, break the secret key into m chunks $k = (k_1, ..., k_m)$, each of length l. The goal is to find one key $t = (t_1, ..., t_m)$ that is used in encryption. The side channel advice can be interpreted as the probability values $p[k_i = x], x \in \{0, 1\}^l$. A likelihood number (weight) can then be assigned for each chunk/value pair, which forms an 2^l by m matrix $w = (w_1, ..., w_m)$, where each column w_i contains the weights of the i-th chunk taking every possible value. The higher the probability $p[k_i = x]$, the smaller its corresponding weight in w_i. An example is illustrated in 2, where we consider a 4-bit key broke into $m = 2$ chunks. The most likely key in this case is 0010 with weight 0, followed by 0000 with weight 1. The weight of a key is simply the sum of all of its chunks' weights.

They proposed a graph-based key enumeration algorithm which generates keys whose weights are within a given range $[W_1, W_2]$. The key searching algorithm takes in the matrix w, the number of keys to search e, a size parameter a, and a testing function T. It starts by finding an W_e such that there are approximately e keys in the range $[0, W_e]$. The minimum and maximum weights of these keys are denoted W_{min} and W_{max}. Then, it initializes $W_1 \leftarrow W_{min}, W_2 \leftarrow W_1 + 1$ and begin to iteratively search in the subspace of keys within $[W_1, W_2]$. At the end of the i-th iteration, if no key has been found, it finds W such that there are approximately a^i keys in the range $[W_2, W]$ and updates $W_1 \leftarrow W_2$ and $W_2 \leftarrow W$.

Classically, given the collection of keys K to test in an iteration, it takes $O(|K|)$ calls to the testing function T. This is where Grover's search can be used to accelerate the algorithm and reduce the number of calls to $O(\sqrt{|K|})$. The analysis in [3] shows that the overall time complexity to perform a key search is $O(m^2 \cdot n \cdot \log n(W_{max} + \sqrt{e} + W_e \cdot \log e))$. Since m and n are small for a given algorithm, it is essentially $O(W_{max} + \sqrt{e} + W_e \log e)$. It is clear that the better the side channel advice is, the smaller e will be and the faster this algorithm runs. However, while it is practical to obtain side channel information, it remains unclear how good these information can be. Since it relies on specific implementations, this question may be difficult to answer in theoretical researches.

It worth mentioning that [3] is largely a classical algorithm; its main contribution is better key enumeration based on key ranking. Yet, it is this construct that allows it to create key subspaces and be accelerated by Grover search. We find this work inspiring as it shows the potential of applying Grover's to structured search by partitioning the set in clever ways.

Quantum related-key attacks

Related-key attack is a powerful tool in classical cryptoanalysis. However, these attacks often base on very strong, if not impractical, assumptions. For example, in [11], to recover an 256-bit AES key in $O(2^{99.5})$ time, the adversary need to be able to persuade the target to encrypt not only with their secret key k, but also three other keys $f(k)$ such that f is deterministic and known by both parties. Nonetheless, related-key attacks are sometimes argued to have practical meanings and have some popularity in the cryptography community. Therefore, we tried to find out what related-key attack is capable of in a quantum setting.

Roetteler and Steinwandt presented a related-key attack on block ciphers in 2015 [4]. Their attack leverages Simon's algorithm by reducing the key search problem to a function that can be computed in polynomial time and satisfies the formation of Simon's problem. To recover a k-bit secret key K for an encryption E_K, their method requires an oracle that:

1. Given $L \in \{0,1\}^k, m \in \{0,1\}^n$, computes $E_{K \oplus L}(m)$.

2. Can be queried in superposition.

Then, for a fixed secret key s and some known messages m that can distinguish secret keys[2], they defined the function

$$f_s(x) = \{E_x(m), E_{s \oplus x}(m)\}$$

which can be shown to meet the conditions of Simon's problem: given two key strings $x \neq x'$ such that $f_s(x) = f_s(x')$, it is impossible that $E_x(m) = E_{x'}(m)$ because of the choice of messages. Then, it must be the case that $E_x(m) = E_{s \oplus x}(m)$, that is, $x = s \oplus x'$. Therefore, $f_s(x) = f_s(x') \rightarrow x = s \oplus x'$. Applying Simon's algorithm, s can be found with high probability using k queries to the oracle. Their paper includes a circuit design for the function f_s given the oracle.

This attack is very powerful that it recovers the secret key of a k-bit cipher with $O(k)$ queries, which is much faster than the classical related-key attack mentioned above. Although, as the authors admitted, it is very unlikely that the scenario can happen in a real world setting, it provides an interesting insight of improvement one can gain when a type of attack is considered in post-quantum settings. It is also possible that, due to poor designs or implementations, some related-key properties will appear in certain systems[3]. Related studies will be helpful in informing cryptographic system designers of what to avoid.

To our best knowledge, up to today [4] is the only work addressing quantum related-key attacks on block ciphers. It remains unknown whether more practical setups are possible to impose a threat to symmetric key systems.

Attack on symmetric cryptosystems using Simons algorithm

We now turn our interest to symmetric cryptosystems. Surprisingly, several classical attacks based on finding collisions can be dramatically speedup in a quantum setting using Simons algorithm in polynomial

[2]For simplicity, we use m to denote several n-bit message blocks $(m_1, ..., m_r)$, and $E(m)$ to denote the encryption result $(E(m_1), ..., E(m_r))$ of these blocks. Blocks m distinguish secret keys means that given two different keys K_1, K_2, $E_{K_1}(m) \neq E_{K_2}(m)$ holds with $(1 - 2^{-rn})$ probability. For a block cipher with k-bit keys and n-bit blocks, it can be shown that $r = \lceil 2k/n \rceil$ messages are needed. For AES-128, $r = 3$.

[3]A famous example is the WEP security protocol.

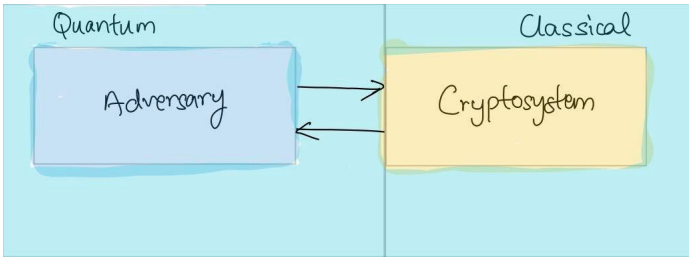

Figure 1: Interface model of classical and quantum setting

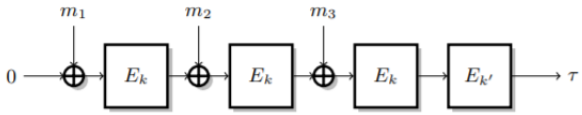

Figure 2: Encrypt-last-block CBC-MAC [5]

time $O(n)$. While breaking symmetric cryptosystems, quantum attack based on Simons algorithm depends on security model of the Simons algorithm, i.e. the quantum adversary has a quantum access to our classical cryptosystems, as shown in 1.

Breaking message authentication codes

In the work of Kuwakado et al [12], it has been shown that Even-Mansour Cipher (block cipher) can be broken using superposition queries. The quantum queries are sent to the quantum oracle to get a superposition of encrypted outputs. The function of the oracle is:

$$\sum_x \psi_x |x\rangle \mapsto \sum_x \psi_x |E_x\rangle$$

Although this model was very strong, it was not a concept that could be used in real life as there was no practical implementation.

Kaplan et al. [5] have given a real-world quantum attack against Even-Mansour Cipher using the weak form of Simon's algorithm or Quantum Period Finding. The weak form of Simon's algorithm says that given $f : \{0,1\}^n \mapsto \{0,1\}^n$, there exists $s \in \{0,1\}^n$ with $f(x) = f(x \oplus s)$ where $x \oplus x' \in \{0^n, s\}$. That is, we want to find s while there can be some random unstructured collisions and not just s. This concept is used to build quantization of classical attack against Even-Mansour cipher. The quantum attack builds the same function as a classical attack:

$$f : \{0,1\}^n \mapsto \{0,1\}^n$$

$$x \mapsto E_{k_1,k_2} \oplus P(x) = P(x \oplus k_1) \oplus P(x) \oplus k_2$$

When $f(x) = f(x \oplus k)$ is achieved, we can recover the key k_1.

[5] showed how using quantization of several classical attacks, several block cipher modes of operation such as CBC-MAC, PMAC, GMAC etc can also be broken.

For quantum attack on a CBC-MAC, illustrated in 2, the adversary sends a superposition of messages and receives a corresponding superposition of $k+1$ MAC values or tags:

$$\sum_x \psi_x |x\rangle |0\rangle \mapsto \sum_x \psi_x |x\rangle |MAC(x)\rangle$$

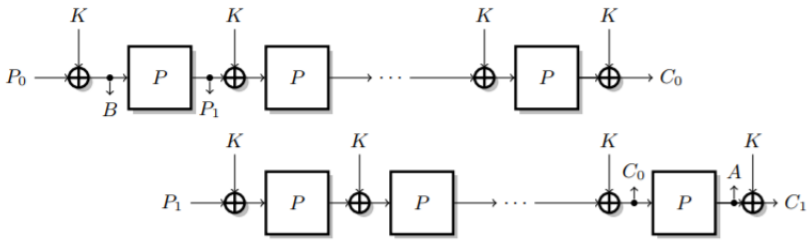

Figure 3: Representation of a slid-pair used in a slide attack [5]

In the paper, they have given a function for CBC-MAC:

$$\begin{aligned} f : \{0,1\}^n x \{0,1\}^n \mapsto \{0,1\}^n \\ b, x \mapsto MAC(b||x) = E_{k'}(E_k(x \oplus E_k(b))) \\ f(0,x) = E_{k'}(E_k(x \oplus E_k(b))) \\ f(1,x) = E_{k'}(E_k(x \oplus E_k(1))) \end{aligned}$$

so $f(b,x) = f(b \oplus 1, x \oplus \delta)$ where $\delta = E_k(0) \oplus E_k(1)$.

The hidden period here is that the bit b can be changed like the above equation to get a collision. Thus the value $1||\delta$ can be recovered. This value can be used to make forgeries as:

$$MAC(0||m||m') = MAC(1||m \oplus \delta||m')$$

So, with a quantum attack, a key can be recovered in CBC-MAC. In a similar way, using Simon's algorithm a quantum attack was implemented and a key k was recovered from PMAC, GMAC and Authenticated Encryption with Associated Data cryptosystems using superposition queries. The authors also broke eight Caesar candidates: AEZ, CLOC, COPA, Minalpher, OCB,OMD, OTR, POET. The paper also described a way to quantize a slide attack, shown in 3, in linear time.

Therefore, by executing the classical functions in a quantum computer using Simon's algorithm in linear time $O(n)$, [5] managed to break several symmetric cryptosystems.

In [6], the authors proposed a defense to [5,12]. It is based on a pair of shifted functions: $f(x) = g(x \oplus k)$. The shifted function is the target in the hidden shift problem. In the Hidden Shift problem, two functions are given with the promise that one is the hidden shift. The objective of the hidden shift problem is to identify this shift.

The authors established the security of symmetric cryptographic functions using the difficulty assumption of Hidden-Shift problem. This problem is random self-reducible and can amplify the hardness of symmetric key constructions. The enhancement is done by replacing bitwise XOR operation with operations over alternative finite groups (such as $Z/2^n$). This small adaptation results in retaining the same level of security the symmetric cryptosystems had as in the classical case.

For a hidden shift CBC-MAC, a bitwise XOR operation in the original construction is replaced by composition in some exponentially-large family of finite-groups G. Each message was then identified with an element of G, and the pseudorandom permutaion E_k was taken as a permutation of the group elements of G such that:

$$\text{CBC-MAC}_{k,k'}{}^G : G* \to G(m_1, \ldots, m_l) \to E_{k'}(E_k(m_l \cdot E_K(\ldots E_k(m_2 \cdot E_k(m_1)) \ldots)))$$

where $\cdot$ denotes the group operation in G

The authors also showed that the decision and search version of the hidden shift problem are equivalent. With this proof, the authors have shown that it is more difficult for quantum attackers to break the hidden

shift version of the block cipher algorithms. The adversary will need a subroutine that solved the hidden shift problem first. So, with the assumption that there doesn't exist a polynomial-time quantum algorithm for the Hidden Shift problem, the authors have shown that CBC-MAC can be modified to remain secure against quantum attacks.

Post-quantum security of random-oracle based functions

In a classical setting, classical random oracle model (ROM) proofs ensure security of cryptosystems by recording information about the queries made by an adversary. However, in a quantum setting, this was not possible owing to the quantum no-cloning theorem and the fact that doing any kind of measurement on the adversary query state would be easily detected by an adversary. As any measurement will disturb the quantum system, such measurement may be detectable to the adversary. When an adversary would detect such recording of its queries, it could then refuse to continue. [7] gave an insight on post-quantum cryptography with the help of a quantum random oracle model (QROM). This paper solved the problem of recording information about the adversarys queries in a quantum setting with the adversary detecting. This observation is done by viewing both the adversarys query and the oracle itself in the Fourier domain.

Hash functions (for ROM or QROM), in general, are built from smaller building blocks called compression functions. For these compression functions to be secure like monolithic object, they should be indifferentiable to an adversary. To illustrate what this means, consider an adversary who has access to two oracles h and H. In case (1), h is a random function and H is built from h according to the hash function construction. That is, $H(x) = B(h(x))$ where B is the hash function. In Case (2), H is a random function, and h is simulated so as to be consistent with H. A hash function B is indifferentiable from a random oracle if no efficient adversary can distinguish the two cases. For such indifferentiability to hold, recording of the quantum adversary queries need to be possible.

A way to do this is explained in paper [7] using compressed oracle technique, which allows for recording the adversarys queries in a way that the adversary can never detect. It is based on the fact that when an adversary interacts with a random oracle, it will be entangled with a uniform superposition of oracles. Owing to the symmetric property of entanglement, if the adversary every has any information about the oracle, the oracle must also have information about the adversary. So, in this way, a simulator may get away with recording the information about the adversary.

The compressed oracle technique allows simulation of a compressed Fourier oracle with independent applications. Such a compressed oracle gives a way to record information from queries that the adversary makes. This information is whether a particular value has been queried by the adversary and what the value of the query at that point is. Compressed oracle also have the power of forgetting such that it can also forget some of the oracle points simulated previously. Implementing a compressed oracle, the adversary can never detect that it is interacting with a simulated oracle.

In a compressed Fourier oracle, a state vector D is set to 0^{n2^m}. After q queries, D will be the sum of q point functions such that the output will be zero in all but q locations, i.e. after a query q, D will be set to: $0^n 0^n \dots q \dots 0^n 0^n$. Therefore, we can actually compress D in an unordered list of at most q pairs (x, z) with distinct x such that $z \neq 0$.

We can instead describe how the map behaves directly on the compressed encoding. This gives us the compressed Fourier oracle. It starts with an empty database, and on each query it performs the map:

$$|x, z\rangle \otimes |D\rangle \mapsto |x, z\rangle \otimes |D \oplus (x, z)\rangle$$

where $D \oplus (x, z)$ is the procedure that does the following:

1. If $z = 0$ it outputs D.

2. If there is a pair $(x, z) \in D$ for some z , it does the following:

- If $z = z$, it removes (x, z) from D, and outputs the new D
- if $z \neq z$, it replaces (x, z) with $(x, z \oplus z)$ and outputs the new D

3. Finally, if there is no such pair $(x, z) \in D$, it adds the pair (x, z) to D and outputs the new D

Using a compressed oracle in this way, a simulator can record information on the exact location of a particular value that was queried by the adversary and what the value of the query at that point is. For example,

$$(H \otimes CNOT \otimes H) \otimes |x\rangle \otimes |y\rangle \otimes 0^{n2^m} \mapsto |x\rangle \otimes |y\rangle \otimes |0 \ldots 0Y0 \ldots\rangle$$

The index of y is given by x.

Post-quantum random oracle model is an important tool that gives efficient quantum resistant cryptosystems. Using compressed Fourier Oracle to show indifferentiability, the QROM records information regarding the queries of the adversary in the Fourier domain, which can be used to find exactly what the adversary was looking for.

Conclusion

In our survey, we have studied quantum key recovery of symmetric block ciphers. We have found no attack that fundamentally breaks the standard AES and 3-DES. However, a straightforward brute-force search accelerated by Grover's is able to put AES-128, AES-192 and 3-DES below today's security recommendations. It is also possible to further increase key recovery if some side channel advice is collected by the adversary. With some impractical assumptions, classical related-key attacks can achieve quadratic speed up, whereas its quantum version dramatically reduces the required time. However, few has been done to analyze whether the inner design of any specific cipher, like the AES, can be abused with quantum computation.

While block ciphers remain their obscurity-based security, message authentication codes have less luck. In a quantum-query setting, many MACs can be broken with Simon's algorithm. Luckily, it is possible to make MACs secure again by incorporating hidden shift constructs. Lastly, we have also explored the possibility to build quantum random oracles with hash functions. The open problem that interests us the most is whether the hidden shift construct can be extended to other systems beyond MACs.

References

[1] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying Grover's Algorithm to AES: Quantum Resource Estimates," in *Post-Quantum Cryptography*, T. Takagi, Ed. Cham: Springer International Publishing, 2016, pp. 29–43. [Online]. Available: https://arxiv.org/abs/1512.04965

[2] P. Zhong and W. Bao, "Quantum mechanical meet-in-the-middle search algorithm for Triple-DES," *Chinese Science Bulletin*, vol. 55, no. 3, pp. 321–325, Jan 2010. [Online]. Available: https://doi.org/10.1007/s11434-009-0532-5

[3] D. P. Martin, A. Montanaro, E. Oswald, and D. Shepherd, "Quantum Key Search with Side Channel Advice," in *Selected Areas in Cryptography – SAC 2017*, C. Adams and J. Camenisch, Eds. Cham: Springer International Publishing, 2018, pp. 407–422.

[4] M. Roetteler and R. Steinwandt, "A note on quantum related-key attacks," *Information Processing Letters*, vol. 115, no. 1, pp. 40 – 44, 2015. [Online]. Available: https://arxiv.org/abs/1306.2301

[5] M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia, "Breaking Symmetric Cryptosystems Using Quantum Period Finding," in *Advances in Cryptology – CRYPTO 2016*, M. Robshaw and J. Katz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 207–237.

[6] G. Alagic and A. Russell, "Quantum-Secure Symmetric-Key Cryptography Based on Hidden Shifts," in *Advances in Cryptology – EUROCRYPT 2017*, Paris, France, 2017, pp. 65–93. [Online]. Available: https://arxiv.org/abs/1610.01187

[7] M. Zhandry, "How to Record Quantum Queries, and Applications to Quantum Indifferentiability," Cryptology ePrint Archive, Report 2018/276, 2018, https://eprint.iacr.org/2018/276.

[8] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997. [Online]. Available: https://arxiv.org/abs/quant-ph/9508027

[9] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: ACM, 1996, pp. 212–219. [Online]. Available: https://arxiv.org/abs/quant-ph/9605043

[10] National Institute of Standards and Technology, "Key Management - Publications." [Online]. Available: https://csrc.nist.gov/Projects/Key-Management/publications

[11] A. Biryukov and D. Khovratovich, "Related-key Cryptanalysis of the Full AES-192 and AES-256," Cryptology ePrint Archive, Report 2009/317, 2009. [Online]. Available: https://eprint.iacr.org/2009/317

[12] H. Kuwakado and M. Morii, "Security on the quantum-type Even-Mansour cipher," in *2012 International Symposium on Information Theory and its Applications*, Oct 2012, pp. 312–316.