Attack Beyond-Birthday-Bound MACs in Quantum Setting

Tingting Guo[1,2], Peng Wang[1,2(✉)], Lei Hu[1,2], and Dingfeng Ye[1,2]

[1] SKLOIS, Institute of Information Engineering, CAS
{guotingting, wpeng, hulei, yedingfeng}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences

Abstract. We systematically study the security of twelve Beyond-Birthday-Bound Message Authentication Codes (BBB MACs) in the Q2 model where attackers have quantum-query access to MACs. Assuming the block size of the underlying (tweakable) block cipher is n bits, the security proofs show that they are secure at least up to $\mathcal{O}(2^{2n/3})$ queries in the classical setting. The best classical attacks need $\mathcal{O}(2^{3n/4})$ queries. We consider secret state recovery against SUM-ECBC-like and PMAC_Plus-like MACs and key recovery against PMAC_Plus-like MACs. Both attacks lead to successful forgeries. The first attack costs $\mathcal{O}(2^{n/2}n)$ quantum queries by applying Grover-meet-Simon algorithm. The second attack costs $\mathcal{O}(2^{m/2})$ quantum queries by applying Grover's algorithm, assuming the key size of (tweakable) block cipher is m bits. As far as we know, these are the first quantum attacks against BBB MACs. It is remarkable that our attacks are suitable even for some optimally secure MACs, such as mPMAC+-f, mPMAC+-p1, and mPMAC+-p2.

Keywords: Beyond-Birthday-Bound · Message Authentication Codes · Quantum Attacks

1 Introduction

Quantum attacks against symmetric crypto primitives. Recent years we have seen amount of work to exploit the quantum security of symmetric crypto primitives, such as Feistel structure [16], Even-Mansour cipher [24], FX construction [25], message authentication codes (MACs) [20], authenticated encryption schemes [20], hash functions [17,13], etc, by using quantum algorithms including Simon's algorithm[32], Grover's algorithm [15], Grover-meet-Simon algorithm [25], GTH algorithm [7], etc. All the attacks are carried on in the Q2 model, where attackers can make superposition queries to a quantum oracle of $U_F : |x, y\rangle \mapsto |x, y \oplus F(x)\rangle$, where F is the classic primitive.

Simon's algorithm and birthday attacks. Common standard MACs such as CBC-MAC, CMAC, PMAC, GMAC, suffer from birthday attacks in the classic setting, and are broken by using Simon's algorithm in polynomial time [20]. The procedure of the attack using Simon's algorithm is as follows: first construct a periodic function $f(x)$ based on the scheme, where the period is a hidden

value s such that $f(x) = f(x \oplus s)$ for all x; second use Simon's algorithm to find the period s; third use the period s to carry out forgery etc. attacks. The period s also can be retrieved from a collision of $f(x) = f(y)$ for $x \neq y$, so that $s = x \oplus y$ with high probability. Therefore $\mathcal{O}(2^{n/2})$ classic queries is enough to find the period and break the scheme, where n is usually the block size of the underling (tweakable) block ciphers. Therefore the schemes broken by using Simon's algorithm are destined to suffer from birthday attacks.

Beyond-Birthday-Bound MACs. The crypto community made great efforts to enhance the security of MACs, by constructing beyond-birthday-bound (BBB) ones, which are secure for above $2^{n/2}$ queries, where n is the block size of the underlying (tweakable) block cipher. In 2010, Yasuda firstly proposed a provable BBB MAC: SUM-ECBC [35]. Later on, other BBB MACs, such as PMAC_Plus [36] , 3kf9 [37], LightMAC_Plus [29], 1k-PMAC_Plus [12], PloyMAC [8] and so on were proposed. In 2018, Datta et al. [10] reduced the number of keys and proposed BBB MACs: 2K-ECBC_Plus, 2K-PMAC_Plus, 2kf9, and so on, where 2kf9 was broken by a birthday bound attack by Shen et al. [31]. The primary proofs show that they are secure up to $2^{2n/3}$ queries (ignoring the maximum message length). All the above BBB MACs follow a generic design paradigm called Double-block Hash-then-Sum (in short DbHtS) [10], which generates double hash blocks on the message and then sum the two encrypted blocks as the output. So The computation of DbHtS consists of two chains, which were denoted as G and H, and $DbHtS(M) = G(M) \oplus H(M)$ for the message M.

In 2020, Cogliati et al. [9] proposed some variants of PMAC_Plus: mPMAC+-f, mPMAC+-p1, mPMAC+-p2, with optimal security, in other words, security up to 2^n queries. They follow a variant of DbHtS, we call it as Double-block Hash-then-Function (in short DbHtF). The computation of DbHtF also consists of two chains, denoted as G and H, but the chain results are processed by a more general function F from $2n$ bits to n bits: $DbHtF(M) = F(G(M), H(M))$. Therefore all PMAC_Plus-like MACs which follow DbHtS also follow DbHtF. The double blocks bring $2n$-bit internal state, making the classic birthday attack no longer applicable.

Classical attacks. Due to the $2n$-bit internal state in DbHtS MACs, the output collision of a pair of messages can not benefit forgery attacks. The best classical attacks against part of DbHtS MACs proposed by Leurent et al. [26] need $\mathcal{O}(2^{3n/4})$ queries. The crucial point is to find a quadruple of messages, which leads to successful forgeries. The search for such a quadruple is reduced to a 4-xor problem with $3n$-bit outputs based on DbHtS MACs. Recently, Kim et al. [23] further proved that some of them are secure up to $2^{3n/4}$ queries. So the attack is optimal in terms of the query number. In fact, Leurent's attack is suitable for 2K-ECBC_Plus, PloyMAC, and 2K-PMAC_Plus as well.

Direct quantum acceleration. The k-xor problem, a generalized birthday problem [33], is a hot topic related to quantum collision finding of hash functions [14,30,17,13]. The main idea comes from BHT algorithm [7]. To solve the core 4-xor problem with $3n$-bit outputs proposed in [26], the best algorithm needs

$\mathcal{O}(2^{3n/5})$ quantum queries, which is the lowest bound according to [2]. Note that $\mathcal{O}(2^{3n/5})$ is slightly better than the query complexity of classical attacks.

Motivations. Are there any better quantum attacks against BBB MACs? What about the security of BBB MACs in the quantum setting?

Grover-meet-Simon algorithm. For BBB MACs, Simon's algorithm is invalid. We need new techniques. In 2017, Leander and May [25] combined Grover's algorithm with Simon's algorithm to attack FX construction [21,22]. The main idea is to construct a function with two inputs based on FX, say $f(u, x)$. When the first input u equals to a special value k, the function has a hidden period s such that $f(k, x) = f(k, x \oplus s)$ for all x. Their combined algorithm use Grover's algorithm to search k, by running many independent Simon's algorithms to check whether the function is periodic or not, and recover both k and s in the end. The attack only costs $\mathcal{O}(2^{m/2}(m + n))$ quantum queries to FX, which is much less than the proved security up to $2^{\frac{m+n}{2}}$ queries [21], where m is the bit length of u, which is the key length of the underlying block cipher and n is the bit length of s, which is the block size. Their heuristic work provides a new tool to study the quantum security of symmetric schemes.

Attacking strategies. With strategies 1 and 2, we utilize Grover-meet-Simon algorithm to recover some secret states of BBB MACs, which lead to successful forgery attacks.

1) Strategy 1: For SUM-ECBC-like DbHtS MACs, G and H process the message in the same way but with different keys, and they are not secure under the quantum attack using Simon's algorithm. We can use the same method C, based on G (resp. H), to construct a periodic function denoted as $g(b, x) = C^G(b, x)$ (resp. $h(b, x) = C^H(b, x)$) where $b \in \{0, 1\}$ and $x \in \{0, 1\}^n$. The periods of g and h are denoted as $1\|s_1$ and $1\|s_2$ respectively. Then use the same method C on $DbHtS = G \oplus H$, we get $C^{DbHtS}(b, x) = C^G(b, x) \oplus C^H(b, x) = g(b, x) \oplus h(b, x)$. Unfortunately s_1 is equal to s_2 usually with negligible probability, so $C^{DbHtS}(b, x)$ is not a periodic function. We construct

$$\begin{aligned} f(u, x) =& C^{DbHtS}(0, x) \oplus C^{DbHtS}(1, x \oplus u) \\ =& g(0, x) \oplus h(0, x) \oplus g(1, x \oplus u) \oplus h(1, x \oplus u). \end{aligned}$$

We can verify that when $u = s_1$ or s_2, $f(u, x)$ is a periodic function: the period is $s_1 \oplus s_2$. Thus we can use Grove-meet-Simon algorithm to recover both s_1 and s_2.

2) Strategy 2: For PMAC_Plus-like DbHtF MACs, G and H process the message in different ways with the same keys, making Strategy 1 not applicable. But we can use the same method based on G (resp. H), to construct a function denoted as $g(u, b, x)$ (resp. $h(u, b, x)$). When u equals a special value say u^*, both $g(u^*, b, x)$ and $h(u^*, b, x)$ are periodic functions with the same period $1\|s$. If the method is applied to $DbHtF$, we get

$$f(u, b, x) = F(g(u, b, x), h(u, b, x)).$$

For $u = u^*$, $f(u^*, b, x)$ is a periodic function. So Grove-meet-Simon algorithm can be applied to recover the special value u^* and the period $1\|s$.

3) Strategy of key search: We notice that most BBB MACs have more than one key. For example, mPMAC+-f, mPMAC+-p1, and mPMAC+-p2, are respectively keyed by five independent m-bit keys. For a perfect crypto primitive, there should be no better way to recover the keys than the exhaustive search, whose complexity is $\mathcal{O}(2^{5m/2})$ for $5m$-bit keys by Grover's algorithm. We found it is sufficient to find one key in order to create forgeries for DbHstF MACs. Accelerating the search by Grover's search, the attack costs $\mathcal{O}(2^{m/2})$ quantum queries. Especially, we are able to recover all keys after recovering one key for some of DbHtS MACs.

Table 1. Summary of the main results. n is the message block size, m is the length of the key of underlying (tweakable) block cipher. The number of maximum blocks of a query is in number of constant length queries.

Scheme	Key space	Provable classical security query bound	Query complexity of classical attack	Query complexity of the quantum acceleration of classical attack	Quantum secret state recovery attack (our work)		Quantum key recovery attack (our work)	
					Queries	Qubits	Queries	Qubits
SUM-ECBC [35]	2^{4m}	$\Omega(2^{3n/4})$ [23]	$\mathcal{O}(2^{3n/4})$ [26]	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^m n)$	$\mathcal{O}(m+n^2)$
2K-ECBC_Plus [10]	2^{3m}	$\Omega(2^{2n/3})$ [10]	$\mathcal{O}(2^{3n/4})$	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^m n)$	$\mathcal{O}(m+n^2)$
PolyMAC [8]	2^{2m+2n}	$\Omega(2^{3n/4})$ [23]	$\mathcal{O}(2^{3n/4})$	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{(n+m)/2}n)$	$\mathcal{O}(m+n^2)$
GCM-SIV2 [19]	2^{4m+2n}	$\Omega(2^{2n/3})$ [19]	$\mathcal{O}(2^{3n/4})$ [26]	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{(n+m)/2}n)$	$\mathcal{O}(m+n^2)$
PMAC_Plus [36]	2^{3m}	$\Omega(2^{3n/4})$ [23]	$\mathcal{O}(2^{3n/4})$ [26]	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$
1k-PMAC_Plus [12]	2^m	$\Omega(2^{2n/3})$ [12]	$\mathcal{O}(2^{3n/4})$ [26]	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$
2K-PMAC_Plus [10]	2^{2m}	$\Omega(2^{2n/3})$ [10]	$\mathcal{O}(2^{3n/4})$	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$
3kf9 [37]	2^{3m}	$\Omega(2^{3n/4})$ [23]	$\mathcal{O}(\sqrt[4]{n}2^{3n/4})$ [26]	$\mathcal{O}(2^{3n/5})$	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(n)$	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$
mPMAC+-f [9]	2^{5m}	$\Omega(2^n)$ [9]	-	-	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$
mPMAC+-p1 [9]	2^{5m}	$\Omega(2^n)$ [9]	-	-	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$
mPMAC+-p2 [9]	2^{5m}	$\Omega(2^n)$ [9]	-	-	$\mathcal{O}(2^{n/2}n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$
PMAC_TBC3k [28]	2^{3m}	$\Omega(2^n)$ [28]	-	-	-	-	$\mathcal{O}(2^{m/2})$	$\mathcal{O}(m+n)$

Our contributions. Table 1 summarizes our main results and comparisons with provable security claims, best classical attack results, and its quantum acceleration results.

1) We reduce the query complexity from $\mathcal{O}(2^{3n/5})$ by direct quantum acceleration of classic method to $\mathcal{O}(2^{n/2}n)$ by our secret state recovery attacks in the Q2 model for both DbHtS and DbHtF MACs. Especially, our attacking strategies are even suitable to optimal secure MACs, including mPMAC+-f, mPMAC+-p1, mPMAC+-p2 and PMAC_TBC3k.
2) We reduce the key search complexity from $\mathcal{O}(2^{tm/2})$ (for perfect t-key MACs, $t = 2, 3$ or 5) to $\mathcal{O}(2^{m/2})$ by our key recovery attacks in the Q2 model for DbHtF MACs. Although one key recovery is enough to get successful forgery, we can further recover all keys of PMAC_Plus, 3kf9, and 2K-PMAC_Plus.

Organization of the paper. Section 2 introduces quantum algorithms, the quantum security of MAC and previous attack for MAC by Simon's algorithm. Section 3 applies Strategy 1 and 2 to make secret state attacks for SUM-ECBC-like and PMAC_Plus-like MACs respectively. Section 4 applies the strategy of of key search to make key recovery attacks for PMAC_Plus-like MACs. Section 5 gives conclusions.

2 Preliminaries

For a positive integer m, let $\{0,1\}^m$ be the set of all m-bit binary string. For two bit strings x and y, the concatenation is $x\|y$, the *bitwise exclusive-or* is $x \oplus y$. Let $|\mathcal{U}|$ be the number of the elements in set $\mathcal{U}$.

2.1 Quantum Algorithms

In this section, we introduce some useful quantum techniques which will be involved in the following sections. We put quantum basis in Appendix A.

1) Grover's algorithm Grover's algorithm [15] can find a target value with high probability.

Grover problem. Let m be a positive integer, and $test : \{0,1\}^m \to \{0,1\}$ be a boolean function ($|\{u : test(u) = 1\}| = e$). Find a u such that $test(u) = 1$.

Classically, we can search an element who satisfies $test(u) = 1$ with $\mathcal{O}(\frac{2^m}{e})$ queries to $test(\cdot)$. However, the Grover's algorithm [15] can find such an element with only $\mathcal{O}(\sqrt{\frac{2^m}{e}})$ quantum queries [6]. Generally, the *test* function can't describe the target objects precisely. So we consider Grover problem with biased *test* function.

Grover problem with biased test function. Let m be a positive integers, $\mathcal{U}(|\mathcal{U}| = e)$ be a subset in $\{0,1\}^m$, $test : \{0,1\}^m \to \{0,1\}$ be a boolean function who satisfies

$$\begin{cases} \Pr[test(u) = 1] = 1, & u \in \mathcal{U}, \\ \Pr[test(u) = 1] \le p_1, & u \notin \mathcal{U}. \end{cases}$$

Find a $u \in \mathcal{U}$.

Grover's algorithm can solve the problem as well with some biases. In fact, Grover's algorithm do as follows: first there is an initial probability to get a u who satisfies $test(u) = 1$; second amplify the initial probability iteration by iteration; third measure the quantum state and get a u who satisfies $test(u) = 1$ with high probability. From definition 2.1, we obtain the initial probability to get a u who satisfy $test(u) = 1$ is between $[p_0, p_0 + p_1]$, where $p_0 = \frac{e}{2^m}$ and it is the initial probability to get a $u \in \mathcal{U}$. Bonnetain [4] has proved when the initial success probability to get a u where $test(u) = 1$ is between an interval $[p_0, p_0 + p_1]$, then after $t = \lceil \frac{\pi}{4 \arcsin \sqrt{p_0}} \rceil$ quantum queries to $test(\cdot)$, the final probability to get a u who satisfies $test(u) = 1$ is $[1 - (\frac{p_1}{p_0} + \sqrt{p_0 + p_1} + 2\sqrt{1 + \frac{p_1}{p_0}}^3 p_0)^2]$. Among all elements satisfying $test(u) = 1$, the proportion of $u \in \mathcal{U}$ is at least $\frac{p_0}{p_0 + p_1}$. Multiple them and we get the following theorem.

Theorem 1. (Adapted from [4]) *Let $m, e, \mathcal{U}, test$ be defined as in Grover problem with biased test function, and $p_0 := \frac{e}{2^m}$. Assume the quantum implementation of $test(\cdot)$ costs $\mathcal{O}(n)$ qubits. Then Grover's algorithm with $t = \lceil \frac{\pi}{4 \arcsin \sqrt{p_0}} \rceil$ quantum queries to $test(\cdot)$ and $\mathcal{O}(m + n)$ qubits will output a $u \in \mathcal{U}$ with probability at least $\frac{p_0}{p_0 + p_1}[1 - (\frac{p_1}{p_0} + \sqrt{p_0 + p_1} + 2\sqrt{1 + \frac{p_1}{p_0}}^3 p_0)^2]$.*

We put Grover's algorithm and the concrete proof of theorem 1 in Appendix B. When apply theorem 1 to concrete attacks of MACs, if $e = 1, p_1 \leq \frac{1}{2^{2m}}$, then for sufficient large m the Grover's algorithm with $\mathcal{O}(2^{m/2})$ quantum queries to $test(\cdot)$ and $\mathcal{O}(m+n)$ qubits will output a $u \in \mathcal{U}$ with probability almost 1 by theorem 1.

2) Simon's algorithm Simon's algorithm [32] finds the period of a periodic function in polynomial time.

Periodic/Aperiodic function. Let n, d be two positive integers, $f : \{0,1\}^n \to \{0,1\}^d$ be a boolean function. We call f as a periodic (resp. aperiodic) function if there is a unique (resp. no) $s \in \{0,1\}^n \backslash \{0^n\}$ such that $f(x) = f(x \oplus s)$ for all $x \in \{0,1\}^n$.

Simon problem. Let n, d be two positive integers, $f : \{0,1\}^n \to \{0,1\}^d$ be a periodic function with a period s. Find s.

Classically, if f is a periodic function, we can find out the period by searching a collision with $\mathcal{O}(2^{d/2})$ queries. However, if f is given as a quantum oracle, Simon's algorithm [32] can solve it with only $\mathcal{O}(n)$ quantum queries. Let

$$\varepsilon(f) := \max_{t \in \{0,1\}^n \backslash \{0^n, s\}} \Pr_x[f(x) = f(x \oplus t)]. \tag{1}$$

This parameter quantifies the disturbance of other partial periods, i.e., $f(x) = f(x \oplus t)$ where $t \in \{0,1\}^n \backslash \{0^n, s\}$. Kaplan et al. [20] have proved the following theorem.

Theorem 2. [20] *Let n, d, f, s be defined as in Simon problem. Let $\varepsilon(f)$ be defined as in equation (1), and c be a positive integer. Then Simon's algorithm with cn quantum queries to f and $\mathcal{O}(n+d)$ qubits will recover s with probability at least $1 - [2(\frac{1+\varepsilon(f)}{2})^c]^n$.*

We put Simon's algorithm and the proof of theorem 2 in Appendix C.

3) Grover-meet-Simon Algorithm In 2017 Leander and May [25] combined Grover's algorithm with Simon's algorithm to analyze FX construction. A general problem is described as follows:

Grover-meet-Simon problem. Let m, n, d be three positive integers, set $\mathcal{U} \subseteq \{0,1\}^m (|\mathcal{U}| = e)$ and $f : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^d$ be a function who satisfies

$$\begin{cases} f(u,\cdot) \text{ is a period function with period } s_u, & u \in \mathcal{U}, \\ f(u,\cdot) \text{ is an aperiodic function}, & u \notin \mathcal{U}. \end{cases}$$

Set $\mathcal{U}_s := \{(u, s_u) : u \in \mathcal{U}, s_u \text{ is the period of } f(u,\cdot)\}$. Find any tuple $(u, s_u) \in \mathcal{U}_s$.

The problem consists of the Grover problem as a whole and the Simon problem partially. The main idea is to search $u \in \mathcal{U}$ by Grover's algorithm and check whether or not $u \in \mathcal{U}$ by whether $f(u,\cdot)$ is periodic or not, which can

be implemented by Simon's algorithm. Bonnetain [4] has formalized the Grover-meet-Simon algorithm. He presented the success probability for $|\mathcal{U}| = 1$. Let

$$\varepsilon(f) := \max_{(u,t)\in\{0,1\}^m\times\{0,1\}^n\setminus(\mathcal{U}_s\cup\{0,1\}^m\times\{0^n\})} \Pr_x[f(u,x) = f(u,x\oplus t)] \quad (2)$$

to quantify the disturbance of $u \notin \mathcal{U}$ and other partial periods ts for $u \in \mathcal{U}$, i.e., $f(u,x) = f(u,x\oplus t)$ where $(u,t) \in \{0,1\}^m \times \{0,1\}^n\setminus(\mathcal{U}_s \cup \{0,1\}^m \times \{0^n\})$. We generalize the success probability of the algorithm for $|\mathcal{U}| \geq 1$ as follows.

Theorem 3. *Let $m, n, d, f, \mathcal{U}, \mathcal{U}_s, e$ be defined as in Grover-meet-Simon problem. Let $\varepsilon(f)$ be defined as in equation (2). Let c be a positive integer, $p_0 := \frac{e}{2^m}$ and $p_1 := [2\cdot(\frac{1+\varepsilon(f)}{2})^c]^n$. Then Grover-meet-Simon algorithm with $\lceil \frac{\pi}{4\arcsin\sqrt{p_0}}\rceil \cdot cn$ quantum queries to f and $\mathcal{O}(m+cn^2+cdn)$ qubits outputs a tuple $(u, s_u) \in \mathcal{U}_s$ with probability at least $\frac{(1-p_1)p_0}{p_0+p_1}[1-(\frac{p_1}{p_0}+\sqrt{p_0+p_1}+2\sqrt[3]{1+\frac{p_1}{p_0}}\,p_0)^2]$.*

We put Grover-meet-Simon algorithm and the proof of theorem 3 in Appendix D. When apply this algorithm to a concrete attack of MACs, if $\varepsilon(f) \leq 3/4, e \leq 2, d = m = n$ and n is sufficient large, then we let $c = 16$ and Grover-meet-Simon algorithm after $\mathcal{O}(2^{n/2}n)$ quantum queries to f using $\mathcal{O}(n^2)$ qubits will output a tuple $(u, s_u) \in \mathcal{U}_s$ with probability almost 1 by theorem 3.

2.2 Quantum Security of MACs

Message authentication code (MAC) generates a tag T for any message M with key K: $T = \mathrm{MAC}_K(M)$. Given the quantum oracle of $\mathrm{MAC}_K(\cdot)$, Boneh and Zhandry [3] defined the existential unforgeability against quantum chosen message attack (EUF-qCMA). One MAC is EUF-qCMA if no quantum attacker can output $q+1$ distinct massage-tag pairs with non-negligible probability after q quantum queries to MAC_K. Notice that we can regard any classical query as a special quantum query. So the q quantum queries contain q quantum and classical queries in all.

For all concrete MACs in this paper, we assume the bit length of message is integral multiples of n. Also, we assume the underlying (tweakable) block cipher of MACs is a (tweakable) random permutation.

2.3 Attacking ECBC-MAC

MACs of single-chain like ECBC-MAC are broken by using Simon's algorithm [20], with only $\mathcal{O}(n)$ quantum queries. We write the MAC as a function G. The attack in [20] is to construct a periodic function g based on G using a method C. We denote it as $g(b,x) = C^G(b,x)$ with a period $1\|s$.

In the following, we demonstrate how they give the construction g for the ECBC-MAC variant [1], the estimation of $\varepsilon(g)$ and the forgery attack after recovery of s. ECBC-MAC uses a block cipher keyed by two independent keys, denote as E_1, E_2.

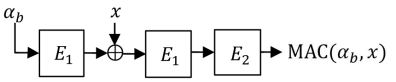


Fig. 1. ECBC-MAC with two-block message $M = (\alpha_b, x)$.

Construction of function g. Let $b \in \{0,1\}$, $x \in \{0,1\}^n$, and α_0, α_1 are two arbitrary fixed numbers in $\{0,1\}^n$. ECBC-MAC with message $M = (M[1], M[2]) = (\alpha_b, x)$ is shown in figure 1, which can be written as

$$\mathrm{MAC}(\alpha_b, x) = g(b, x),$$

where

$$g(b, x) = E_2(E_1(E_1(\alpha_b) \oplus x)).$$

Obviously, g has a period of $1\|s$ where $s = E_1(\alpha_0) \oplus E_1(\alpha_1)$:

$$\begin{aligned} g(b', x') = g(b, x) &\Leftrightarrow E_2(E_1(E_1(\alpha_{b'}) \oplus x')) = E_2(E_1(E_1(\alpha_b) \oplus x)) \\ &\Leftrightarrow \begin{cases} x' \oplus x = 0^n & \text{if } b' \oplus b = 0, \\ x' \oplus x = E_1(\alpha_0) \oplus E_1(\alpha_1) & \text{if } b' \oplus b = 1. \end{cases} \end{aligned}$$

Therefore, $\varepsilon(g) = 0$ and s can be recovered with $\mathcal{O}(n)$ quantum queries to g using $\mathcal{O}(n)$ qubits by theorem 2.

Forgery attack. After recovering s, by using the property of $g(b, x) = g(b, x \oplus s)$, they make a successful forgery after one classic queries as follows:

1) Query $M_1 = (\alpha_0, x)$ and get tag T;
2) Forge $M_2 = (\alpha_1, x \oplus s)$ and its tag T.

To break the notion of EUF-qCMA security, they produce $q + 1$ valid tags with only q queries to the quantum oracle of MAC. Let $q' = O(n)$ denote the number of of quantum queries made to find s. The attacker just repeats the above classic forgery step $q' + 1$ times. So that $2q' + 2$ messages with valid tags are produced, using a total of $2q' + 1$ classical and quantum queries. Therefore, ECBC-MAC is broken by a quantum existential forgery attack.

3 Secret State Recovery Attack for BBB MACs

3.1 Secret State Recovery Attack for SUM-ECBC-like MACs

We focus on DbHtS MAC [10]: $DbHtS(M) = G(M) \oplus H(M)$, which is the generic paradigm of SUM-ECBC-like MACs. Strategy 1 in section 1 constructs

$$f(u, x) = g(0, x) \oplus h(0, x) \oplus g(1, x \oplus u) \oplus h(1, x \oplus u),$$

where periodic function g (resp. h) based on G (resp. H) and $g(b, x)$ (resp. $h(b, x)$) with a period $1\|s_1$ (resp. $1\|s_2$). When $u = s_1$ or s_2, $f(u, x)$ has a period of $s_1 \oplus s_2$. If $\varepsilon(f) \leq 3/4$, then by theorem 3, Simon-meet-Grover algorithm can find both s_1 and s_2 with at most $\mathcal{O}(2^{n/2}n)$ quantum queries and $\mathcal{O}(n^2)$

qubits. In the following, for any concrete SUM-ECBC-like MAC, we only give the construction of function f, the estimation of $\varepsilon(f)$, and the forgery attack after recovery of s_1 and s_2. The method applies to SUM-ECBC [35], PolyMAC [23], the authentication part of GCM-SIV2 [19] and 2K-ECBC_Plus [10]. We only take SUM-ECBC and PolyMAC as examples.

1) Secret State Recovery Attack for SUM-ECBC. SUM-ECBC was designed by Yasuda in 2010 [35], which is the sum of two independent ECBC-MACs. The scheme uses a block cipher keyed by four independent keys, denoted as E_1, E_2, E_3, E_4.

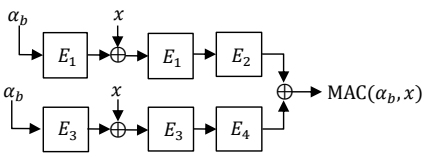


Fig. 2. SUM-ECBC with two-block message $M = (\alpha_b, x)$.

Construction of function f. Let $b \in \{0,1\}$, $x \in \{0,1\}^n$, and α_0, α_1 are two arbitrary different fixed numbers in $\{0,1\}^n$. SUM-ECBC with message $M = (M[1], M[2]) = (\alpha_b, x)$ is shown in figure 2, which can be written as

$$\mathrm{MAC}(\alpha_b, x) = g(b,x) \oplus h(b,x),$$

where

$$g(b,x) = E_2(E_1(E_1(\alpha_b) \oplus x)), h(b,x) = E_4(E_3(E_3(\alpha_b) \oplus x)).$$

Obviously, g (resp. h) has a period of $1\|s_1$ where $s_1 = E_1(\alpha_0) \oplus E_1(\alpha_1)$ (resp. $1\|s_2$ where $s_2 = E_3(\alpha_0) \oplus E_3(\alpha_1)$). Given that E_1, E_3 are two independent random permutations, the probability of $s_1 = s_2$ is at most $1 - 1/2^n$. So in the following we assume $s_1 \neq s_2$. Let

$$f(u,x) = \mathrm{MAC}(\alpha_0, x) \oplus \mathrm{MAC}(\alpha_1, x \oplus u).$$

Estimation of $\varepsilon(f)$. In this case, $\mathcal{U}_s = \{(s_1, s_1 \oplus s_2), (s_2, s_1 \oplus s_2)\}$,

$$\varepsilon(f) = \max_{(u,t) \in \{0,1\}^n \times \{0,1\}^n \setminus (\mathcal{U}_s \cup \{0,1\}^n \times \{0^n\})} \Pr_x[f(u,x) = f(u, x \oplus t)].$$

We consider $u = s_1$ as an example and the other situation is similar. In this case $f(u,x) = f(s_1,x) = E_4(E_3(x \oplus E_3(\alpha_0)) \oplus E_4(E_3(x \oplus s_1 \oplus E_3(\alpha_1))$. We will prove $\varepsilon(f(s_1,\cdot)) \leq \frac{1}{2}$ with overwhelming probability. Otherwise, there is $t \notin \{0^n, s_1 \oplus s_2\}$ such that $\Pr_x[f(s_1,x) = f(s_1, x \oplus t)] > 1/2$, i.e.,

$$\Pr_x \left[\begin{array}{c} E_4(E_3(x \oplus E_3(\alpha_0))) \oplus E_4(E_3(x \oplus s_1 \oplus E_3(\alpha_1))) \oplus \\ E_4(E_3(x \oplus t \oplus E_3(\alpha_0))) \oplus E_4(E_3(x \oplus t \oplus s_1 \oplus E_3(\alpha_1))) \end{array} \right] > 1/2. \quad (3)$$

When $t \notin \{0^n, s_1 \oplus s_2\}$ and $s_1 \neq s_2$, we know the four inputs of $E_4(E_3(\cdot))$ are different from each other. If E_4 is a random function and E_3 is a permutation, the equation (3) happens with negligible probability.

Forgery attack. After recovering s_1 and s_2, by using the property of $f(s_1, x) = f(s_1, x \oplus s_1 \oplus s_2)$, we can make a successful forgery after 3 classic queries as follows.

1) Query $M_1 = (\alpha_0, x)$ and get tag T_1;
2) Query $M_2 = (\alpha_1, x \oplus s_1)$ and get tag T_2;
3) Query $M_3 = (\alpha_0, x \oplus s_1 \oplus s_2)$ and get tag T_3;
4) Forge $M_4 = (\alpha_1, x \oplus s_2)$ and its tag $T_1 \oplus T_2 \oplus T_3$.

Now we try to break the notion of EUF-qCMA security. If $q' = \mathcal{O}(2^{n/2}n)$ denote the number of of quantum queries made to find s_1 and s_2. The attacker just repeats the above classic forgery step $q' + 1$ times. So that $4q' + 4$ messages with valid tags are produced, using a total of $4q' + 3$ classical and quantum queries. Therefore, SUM-ECBC is broken by a quantum existential forgery attack. Generally, the EUF-qCMA attack is straightforward after we find the hidden periods. So we omit it in the following examples.

2) Secret State Recovery Attack for PolyMAC. After replace the block cipher E_i in SUM-ECBC with multiplication functions $H_{k_i}(x) = k_i \cdot x$ for $i = 1, 3$, we get PolyMAC [23], where k_1, k_3 are two independent keys in $\{0,1\}^n$ and they are independent of the keys of E_2, E_4. The chain of MAC is actually PolyHash, whih is used in the authentication of associated data in GCM-SIV2 [19], GCM [27] and HCTR [34].

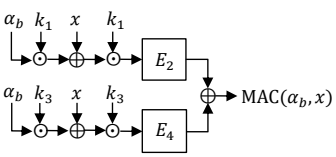


Fig. 3. PolyMAC with two-block message $M = (M[1], M[2]) = (\alpha_b, x)$.

Construction of function f. Let $b \in \{0,1\}$ and $x \in \{0,1\}^n$, α_0, α_1 are two arbitrary different fixed numbers in $\{0,1\}^n$. PolyMAC with message $M = (M[1], M[2]) = (\alpha_b, x)$ is shown in figure 3, which can be written as

$$\mathrm{MAC}(\alpha_b, x) = g(b, x) \oplus h(b, x),$$

where

$$g(b, x) = E_2(k_1^2 \alpha_b \oplus k_1 x), h(b, x) = E_4(k_3^2 \alpha_b \oplus k_3 x).$$

Obviously, g (resp. h) has a period of $1 \| s_1$ where $s_1 = k_1\alpha_0 \oplus k_1\alpha_1$ (resp. $1 \| s_2$ where $s_2 = k_3\alpha_0 \oplus k_3\alpha_1$). The probability of $s_1 \neq s_2$ is at most $1 - 1/2^n$ by the randomness of k_1, k_3. So in the following we assume $s_1 \neq s_2$. Let

$$f(u, x) = \mathrm{MAC}(\alpha_0, x) \oplus \mathrm{MAC}(\alpha_1, x \oplus u).$$

Similar as SUM-ECBC, we can prove $\varepsilon(f) \leq 3/4$.

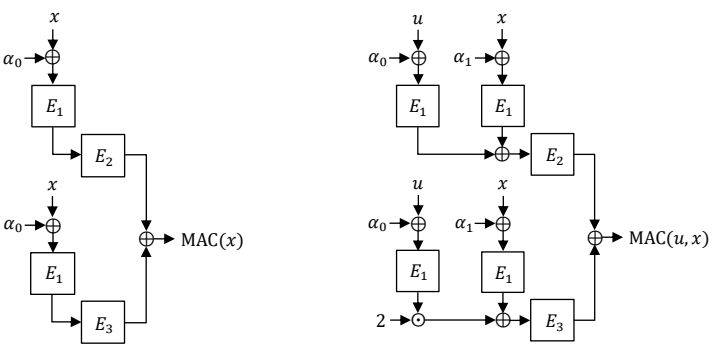
3.2 Secret State Recovery Attack for PMAC_Plus-like MACs

We focus on DbHtF MAC [10]: $DbHtF(M) = F(G(M), H(M))$, which is the generic paradigm of PMAC_Plus-like MACs. Stragety 2 in section 1 constructs

$$f(u, b, x) = F(g(u, b, x), h(u, b, x)),$$

where $g(u, b, x)$ (resp. $h(u, b, x)$) based on G (resp. H). When u equals a special value u^*, both $g(u^*, b, x)$ and $h(u^*, b, x)$ are periodic functions with the same period $1\|s$. Thus $f(u^*, b, x)$ is a periodic function with period $1\|s$. If $s \neq 0^n, \varepsilon(f) \leq 3/4$, we can apply Grover-meet-Simon algorithm (theorem 3) to recover u^*, s with at most $\mathcal{O}(2^{n/2}n)$ quantum queries and $\mathcal{O}(n^2)$ qubits. If $s = 0^n$, we can apply Grover algorithm (theorem 1) to recover u^* with at most $\mathcal{O}(2^{n/2})$ quantum queries and $\mathcal{O}(n)$ qubits. In the following, for any concrete PMAC_Plus-like MAC, we only give the construction of function f, the estimation of $\varepsilon(f)$. The method applies to PMAC_Plus [36], 1k-PMAC_Plus [11,12], 3kf9 [37] and 2K-PMAC_Plus [10], for which the function F is the sum of two cipher blocks. The method even applies to optimally secure MACs, including mPMAC+-f [9], mPMAC+-p1 [9] and mPMAC+-p2 [9], for which the function F is HtmB-f, HtmB-p1 and HtmB-p2 [9] respectively. We only take PMAC_Plus [36] and 3kf9 [37] as examples.

1) Secret State Recovery Attack for PMAC_Plus. PMAC_Plus was designed by Yasuda in 2011 [36]. The scheme uses a block cipher keyed by three independent keys, denoted as E_1, E_2, E_3.



(a) MAC(x). (b) MAC(u, x).

Fig. 4. PMAC_Plus with one-block message $M = (x)$ and two-block message $M = (u, x)$.

Construction of function f. Let $b \in \{0,1\}, u, x \in \{0,1\}^n$ and

$$\alpha_b := \begin{cases} 2E_1(0) \oplus 2^2 E_1(1), & \text{if } b = 0, \\ 2^2 E_1(0) \oplus 2^4 E_1(1), & \text{if } b = 1. \end{cases}$$

PMAC_Plus with message $M = (M[1]) = (x)$ and message $M = (M[1], M[2]) = (u, x)$ are shown as figure 4, which can be written as

$$\mathrm{MAC}(M) = \begin{cases} F(g(u,0,x), h(u,0,x)), \text{ if } M = (x), \\ F(g(u,1,x), h(u,1,x)), \text{ if } M = (u,x), \end{cases}$$

where

$$g(u,b,x) = \begin{cases} E_1(x \oplus \alpha_0), & \text{if } b = 0, \\ E_1(x \oplus \alpha_1) \oplus E_1(u \oplus \alpha_0), & \text{if } b = 1, \end{cases}$$

$$h(u,b,x) = \begin{cases} E_1(x \oplus \alpha_0), & \text{if } b = 0, \\ E_1(x \oplus \alpha_1) \oplus 2E_1(u \oplus \alpha_0), & \text{if } b = 1. \end{cases}$$

$$F(x', y') = E_2(x') \oplus E_3(y'),$$

where $x', y' \in \{0,1\}^n$. We define

$$f(u,b,x) = \begin{cases} \mathrm{MAC}(x), & \text{if } b = 0, \\ \mathrm{MAC}(u,x), & \text{if } b = 1. \end{cases}$$

Let $u^* \in \{0,1\}^n$ such that $E_1(u^* \oplus \alpha_0) = 0^n$. When $u = u^*$, $f(u,b,x)$ has a period $1 \| (\alpha_0 \oplus \alpha_1)$.

Estimation of $\varepsilon(f)$. In this case, $\mathcal{U}_s = \{(u^*, 1\|\alpha_0 \oplus \alpha_1)\}$. Let $\mathcal{U}_t := \{0,1\}^n \times \{0,1\} \times \{0,1\}^n \backslash (\mathcal{U}_s \cup \{0,1\}^n \times \{0^{n+1}\})$, then

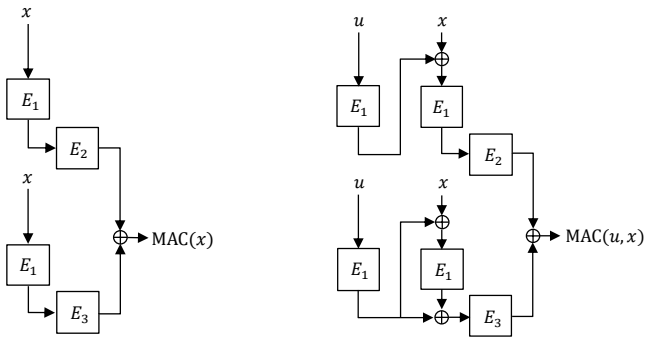
$$\varepsilon(f) = \max_{(u,t_1,t_2) \in \mathcal{U}_t} \Pr_{b,x}[f(u,b,x) = f(u, b \oplus t_1, x \oplus t_2)].$$

We consider $u = u^*$ as example and the other is similar. Firstly, we divide the scope $t_1 \| t_2 \in \{0,1\}^{n+1} \backslash \{0^{n+1}, 1\|\alpha_0 \oplus \alpha_1\}$ into two parts $t_1 = 0, t_2 \neq 0^n$ and $t_1 = 1, t_2 \neq \alpha_0 \oplus \alpha_1$. We take the former as example. In fact, when $u = u^*, t_1 = 0, t_2 \neq 0^n$, the equation $f(u,b,x) = f(u, b \oplus t_1, x \oplus t_2)$ equals

$$E_2(E_1(x \oplus \alpha_b)) \oplus E_2(E_1(x \oplus t_2 \oplus \alpha_b)) \oplus E_3(E_1(x \oplus \alpha_b)) \oplus E_3(E_1(x \oplus t_2 \oplus \alpha_b)) = 0^n. \tag{4}$$

When $t_2 \neq 0^n$ and E_1 is a random permutation, we obtain both the two inputs of E_2 and the two inputs of E_3 are different respectively. Therefore, by the randomness of E_2, E_3, the equation (4) holds with probability at most $1/2$ with overwhelming probability.

2) Secret State Recovery Attack for 3kf9. 3kf9 was designed by Zhang et.al. [37]. The scheme uses a block cipher keyed with three independent keys, denoted as E_1, E_2, E_3.



(a) MAC(x). (b) MAC(u, x).

Fig. 5. 3kf9 with one-block message $M = (x)$ and two-block message $M = (u, x)$.

Construction of function f. Let $b \in \{0,1\}, u, x \in \{0,1\}^n$. Then 3kf9 with message $M = (M[1]) = (x)$ and message $M = (M[1], M[2]) = (u, x)$ are shown in figure 5, which can be written as

$$\mathrm{MAC}(M) = \begin{cases} F(g(u,0,x), h(u,0,x)), \text{ if } M = (x), \\ F(g(u,1,x), h(u,1,x)), \text{ if } M = (u,x), \end{cases}$$

where

$$g(u,b,x) = \begin{cases} E_1(x), & \text{if } b = 0, \\ E_1(x \oplus E_1(u)), & \text{if } b = 1, \end{cases}$$

$$h(u,b,x) = \begin{cases} E_1(x), & \text{if } b = 0, \\ E_1(x \oplus E_1(u)) \oplus E_1(u), & \text{if } b = 1, \end{cases}$$

$$F(x,y) = E_2(x) \oplus E_3(y).$$

where $x', y' \in \{0,1\}^n$. We define

$$f(u,b,x) = \begin{cases} \mathrm{MAC}(x), & \text{if } b = 0, \\ \mathrm{MAC}(u,x), & \text{if } b = 1. \end{cases}$$

Let $u^* \in \{0,1\}^n$ such that $E_1(u^*) = 0^n$. It is easy to obtain that u^* is unique by permutation E_1. Then when $u = u^*$, $f(u^*, 0, x) = f(u^*, 1, x)$ holds for all $x \in \{0,1\}^n$. It means the period is $1\|0^n$, which is trivial. So we apply Grover algorithm to recover u^* directly. We define $test : \{0,1\}^n \to \{0,1\}$ as

$$test(u) = \begin{cases} 1, \text{ if } f(u,0,x_i) = f(u,1,x_i), i = 1, \ldots, q, \\ 0, \text{ otherwise}, \end{cases}$$

where $x_i \in \{0,1\}^n$ and x_i are different from each other.

Estimation of $\max_{u \in \{0,1\}^n \setminus \{u^*\}} \Pr[test(u) = 1] \leq 2^{-2n}$. The deviation

$$\max_{u \in \{0,1\}^n \setminus \{u^*\}} \Pr[test(u) = 1]$$
$$= \max_{u \in \{0,1\}^n \setminus \{u^*\}} \Pr[f(u,0,x_1) = f(u,1,x_1), \ldots, f(u,0,x_q) = f(u,1,x_q)].$$

Here, the equation system

$$f(u,0,x_i) = f(u,1,x_i), i = 1,2,\ldots,q,$$

equals

$$E_2(y_i^1) \oplus E_2(y_i^2) \oplus E_3(y_i^3) \oplus E_3(y_i^4) = 0^n, i = 1,2,\ldots,q,$$

where $y_i^1 = E_1(x_i), y_i^2 = E_1(x_i \oplus E_1(u)), y_i^3 = E_1(x_i), y_i^4 = E_1(x_i \oplus E_1(u)) \oplus E_1(u)$. To calculate the probability of these q equations, we consider sampling about E_2. If y_i^1, y_i^2, who are the inputs of E_2 in ith equation, have all appeared in the other $q-1$ equations, then we don't sample in the ith equation. In fact, if $x_i \oplus x_j = E_1(u)$ then $y_i^1 = y_j^2, y_i^2 = y_j^1$. Therefore, we have to sample E_2 in at least $\lfloor \frac{q+1}{2} \rfloor$ equations among q. For every equation, by the randomness of E_2, it holds with probability at most $\frac{1}{2^n - 2q}$. Therefore, for any $u \in \{0,1\}^n \backslash \{u^*\}$, we have $\Pr[test(u) = 1] \leq (\frac{1}{2^n - 2q})^{\frac{q-1}{2}}$. When $q = 7$, we have $\Pr[test(u) = 1] \leq 2^{-2n}$ for $n \geq 4$.

4 Key Recovery Attack for PMAC_Plus-like MACs

We observe that PMAC_Plus-like MACs such as PMAC_Plus [36], 3kf9 [37] etc., with message $M = (M[1], M[2], M[3])$ share a common structure as in figure 6.

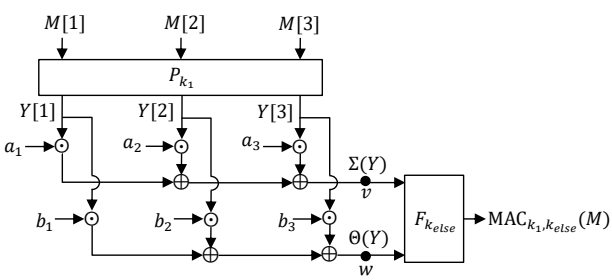


Fig. 6. PMAC_Plus-like MACs with three-block message $M = (M[1], M[2], M[3])$.

Let message $M = (M[1], M[2], M[3]) \in (\{0,1\}^n)^3$, the tag $\mathrm{MAC}_{k_1,k_{else}}(M) \in \{0,1\}^n$, the independent keys $k_1 \in \{0,1\}^m, k_{else} \in \{0,1\}^l$, P_{k_1} be a permutation from $3n$ bit to $3n$ bit keyed by k_1 and $F_{k_{else}}$ be a function from $2n$ bit to n bit keyed by k_{else}, $Y = (Y[1], Y[2], Y[3]) \in (\{0,1\}^n)^3$, public constants $A = (a_1, a_2, a_3) \in (\{0,1\}^n)^3, B = (b_1, b_2, b_3) \in (\{0,1\}^n)^3$. Then the procedure of $\mathrm{MAC}_{k_1,k_{else}}(M)$ is as follows.

1) Given message M, compute $Y = P_{k_1}(M)$;
2) Compute linear combination processes $\Sigma(Y) := a_1Y[1] \oplus a_2Y[2] \oplus a_3Y[3]$ and $\Theta(Y) := b_1Y[1] \oplus b_2Y[2] \oplus b_3Y[3]$;
3) Compute $F_{k_{else}}(\Sigma(Y), \Theta(Y))$ and output it.

4.1 Partial Key Recovery Attack for PMAC_Plus-like MACs

We notice that most BBB MACs have several keys. So we consider a partial key recovery attack, and find that knowing the key k_1 is enough to create forgeries. The recovery of k_1 is as follows. Firstly we fix arbitrary values at points v and w. Secondly, we reverse the linear combination process in step 2) to get two arbitrary different solutions $C_0, C_1 \in \{0,1\}^{3n}$. Thirdly, we guess k_1 and reverse step 1) to get two messages. Finally, input the two messages into the oracle of $\mathrm{MAC}_{k_1,k_{else}}(\cdot)$ to get two tags. If the guess is correct, then the two tags are same by colliding at both points v and w. Otherwise, the two tags may be different with overwhelming probability. That is to say, we check whether or not the guess is correct by whether or not the two tags are equal.

Accelerate the search of k_1 by applying Grover's search. Let set

$$\mathcal{C} := \left\{ (C_0, C_1) \;\middle|\; \begin{array}{l} \Sigma(C_0) = \Sigma(C_1), \Theta(C_0) = \Theta(C_1), \text{ where} \\ C_j = (C_j[1], C_j[2], C_j[3]) \in (\{0,1\}^n)^3, j = 0,1 \text{ and } C_0 \neq C_1 \end{array} \right\}$$

and function $f : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ as $f(k, C) = \mathrm{MAC}_{k_1,k_{else}}(P_k^{-1}(C))$. Then we define $test : \{0,1\}^m \to \{0,1\}$ as

$$test(k) = \begin{cases} 1, \text{ if } f(k, C_0^i) = f(k, C_1^i), i = 1, \ldots, q, \\ 0, \text{ otherwise,} \end{cases}$$

where $(C_0^i, C_1^i) \in \mathcal{C}$. We notice when $k = k_1$, $test(k) = 1$. Given quantum oracle of $\mathrm{MAC}_{k_1,k_{else}}(\cdot)$, if the deviation $\max_{k \in \{0,1\}^m \setminus \{k_1\}} \Pr[test(k) = 1] \leq 2^{-2m}$ for $q = \mathcal{O}(1)$, then we can recover k_1 by Grover's algorithm (theorem 1) with at most $\mathcal{O}(2^{m/2})$ quantum queries and $\mathcal{O}(m+n)$ qubits.

Forgery attack. After recovering k_1, we make a successful forgery after a classical query as follows.

1) Choose an arbitrarily pair $(C_0, C_1) \in \mathcal{C}$.
2) Compute $M_0 = (P_{k_1})^{-1}(C_0)$ and $M_1 = (P_{k_1})^{-1}(C_1)$;
3) Query M_0 to $\mathrm{MAC}_{k_1,k_{else}}(\cdot)$ and get T;
4) Forge message-tag pair (M_1, T).

The EUF-qCMA attack is straightforward. So we omit it.

The method apply to PMAC_Plus [36], PMAC_TBC3k [28], mPMAC+-f [9], mPMAC+-p1 [9], mPMAC+-p2 [9], 1k-PMAC_Plus [11,12], 3kf9 [37] and 2K-PMAC_Plus [10]. In section 4.1 and 4.1, we take PMAC_Plus [36] and 3kf9 [37] as examples and prove the deviation $\max_{k \in \{0,1\}^m \setminus \{k_1\}} \Pr[test(k) = 1] \leq 2^{-2m}$ for $q = \mathcal{O}(1)$ for both of them.

1) Deviation Estimation for PMAC_Plus We have introduced PMAC_Plus in section 3.2. Assume the three independent keys are $(k_1, k_2, k_3) \in (\{0,1\}^m)^3$. The construction with three-block message $M = (M[1], M[2], M[3])$ is shown in figure 7, where $t_{k_1}^j = 2^j E_{k_1}(0^n) \oplus 2^{2j} E_{k_1}(0^{n-1} \parallel 1), j = 1, 2, 3$.

The deviation $\max_{k \in \{0,1\}^m \setminus \{k_1\}} \Pr[test(k) = 1]$ is equals to

$$\max_{k \in \{0,1\}^m \setminus \{k_1\}} \Pr[f(k, C_0^1) = f(k, C_1^1), \ldots, f(k, C_0^q) = f(k, C_1^q)].$$

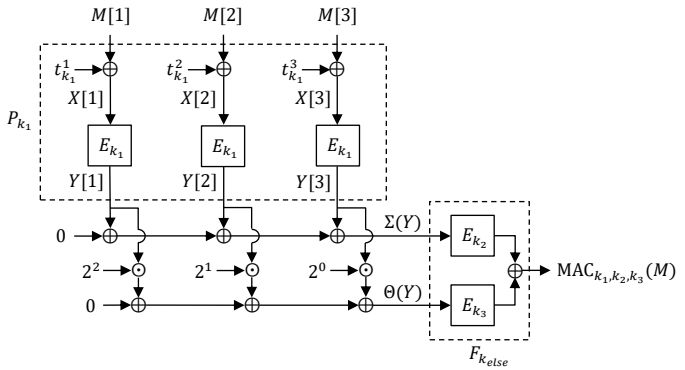


Fig. 7. PMAC_Plus with three-block message $M = (M[1], M[2], M[3])$.

Here, the equation system

$$f(k, C_0^i) = f(k, C_1^i), i = 1, 2, \ldots, q, \tag{5}$$

equals

$$E_{k_2}(\Sigma(Y_0^i)) \oplus E_{k_3}(\Theta(Y_0^i)) = E_{k_2}(\Sigma(Y_1^i)) \oplus E_{k_3}(\Theta(Y_1^i)), i = 1, 2, \ldots, q,$$

where

$$\begin{aligned}\Sigma(Y_b^i) =& E_{k_1}(X_b^i[1]) \oplus E_{k_1}(X_b^i[2]) \oplus E_{k_1}(X_b^i[3]), b = 0, 1,\\ \Theta(Y_b^i) =& 2^2 E_{k_1}(X_b^i[1]) \oplus 2E_{k_1}(X_b^i[2]) \oplus E_{k_1}(X_b^i[3]), b = 0, 1,\end{aligned}$$

and

$$\begin{aligned}X_b^i[1] &= E_k^{-1}(C_b^i[1]) \oplus t_k^1 \oplus t_{k_1}^1,\\ X_b^i[2] &= E_k^{-1}(C_b^i[2]) \oplus t_k^2 \oplus t_{k_1}^2,\\ X_b^i[3] &= E_k^{-1}(C_b^i[3]) \oplus t_k^3 \oplus t_{k_1}^3.\end{aligned}$$

We assume all $C_b^i[a], i = 1, \ldots, q, b = 0, 1, a = 1, 2, 3$ are different. This can be realized easily. Then all $X_b^i[1], i = 1, \ldots, q, b = 0, 1$ are different, all $X_b^i[2], i = 1, \ldots, q, b = 0, 1$ are different and all $X_b^i[3], i = 1, \ldots, q, b = 0, 1$ are different as well.

In the following, we only consider the equations which have new sample of E_{k_1} among the q equations in (5). If $X_b^i[a], b = 0, 1, j = 1, 2, 3$, who are the inputs of E_{k_1} in ith equation, have all appeared in the other $q-1$ equations, then we don't sample in the ith equation. In fact, there may be $X_b^i[a_1] = X_{b'}^{i'}[a_2] = X_{b''}^{i''}[a_3]$, where a_1, a_2, a_3 are three different values belong to $\{1, 2, 3\}, b, b', b'' \in \{0, 1\}, i', i'' \in \{1, \ldots, q\}$. Take $X_0^i[1]$ as example, there may be $b', b'' \in \{0, 1\}, i', i'' \in \{1, \ldots, q\}$ such that $X_0^i[1] = X_{b'}^{i''}[2] = X_{b''}^{i''}[3]$. Therefore, it is easily to obtain that we have to sample E_{k_1} in at least $\lfloor \frac{q+2}{3} \rfloor$ equations among q. Then we consider the probability of the ith equation $f(k, C_0^i) = f(k, C_1^i)$ where we have new sample of E_{k_1}.

1) If
$$\Sigma(Y_0^i) = \Sigma(Y_1^i), \Theta(Y_0^i) = \Theta(Y_1^i), \tag{6}$$
then the ith equation holds. We want to know the upper bound of the probability of this case. So we only consider $\Sigma(Y_0^i) = \Sigma(Y_1^i)$. It means
$$E_{k_1}(X_0^i[1]) \oplus E_{k_1}(X_0^i[2]) \oplus E_{k_1}(X_0^i[3]) = E_{k_1}(X_1^i[1]) \oplus E_{k_1}(X_1^i[2]) \oplus E_{k_1}(X_1^i[3]).$$
By the randomness of E_{k_1}, the probability to make the above equation holds by sampling E_{k_1} is at most $\frac{1}{2^n-6q}$.
2) When the equation set (6) doesn't holds but
$$E_{k_2}(\Sigma(Y_0^i)) \oplus E_{k_3}(\Theta(Y_0^i)) = E_{k_2}(\Sigma(Y_1^i)) \oplus E_{k_3}(\Theta(Y_1^i)), \tag{7}$$
then the ith equation holds as well. Firstly, we exclude the case that $\Sigma(Y_0^i)$, $\Theta(Y_0^i)$,$\Sigma(Y_1^i)$,$\Theta(Y_1^i)$ in ith equation have all appeared in other $q-1$ equations, whose probability is at most $(\frac{2q}{2^n-6q})^4$. Then we assume that in ith equation that at least $\Sigma(Y_0^i)$ hasn't been appeared in other $q-1$ equations, which means $E_{k_2}(\Sigma(Y_0^i))$ is a new sample. Thus the ith equation holds with probability at most $\frac{1}{2^n-2q}$. Overall, this case happens with probability at most $(\frac{2q}{2^n-6q})^4 + \frac{1}{2^n-2q}$.

Sum of case 1) and 2), the ith equation holds with probability at most $\frac{1}{2^n-6q} + (\frac{2q}{2^n-6q})^4 + \frac{1}{2^n-2q} \leq \frac{q}{2^{n-3}}$ assuming $6q \leq 2^{n-1}$. Therefore, the q equations happens with probability at most $(\frac{q}{2^{n-3}})^{\frac{q-1}{3}}$. For PMAC_Plus, the key length $m \leq 2n$. Then when $q = 16$, we have $\Pr[test(k) = 1] \leq 2^{-2m}$ for $m \geq 42$ and any $k \in \{0,1\}^m \backslash \{k_1\}$.

2) Deviation Estimation for 3kf9 We have introduced 3kf9 in section 3.2. Assume the three keys are $(k_1, k_2, k_3) \in (\{0,1\}^m)^3$. The construction with massage $M = (M[1], M[2], M[3])$ is defined as in figure 8.

The deviation $\max_{k \in \{0,1\}^m \backslash \{k_1\}} \Pr[test(k) = 1]$ is equals to
$$\max_{k \in \{0,1\}^m \backslash \{k_1\}} \Pr[f(k, C_0^1) = f(k, C_1^1), \ldots, f(k, C_0^q) = f(k, C_1^q)].$$
Here, the equation system
$$f(k, C_0^i) = f(k, C_1^i), i = 1, 2, \ldots, q, \tag{8}$$
equals
$$E_{k_2}(\Sigma(Y_0^i)) \oplus E_{k_3}(\Theta(Y_0^i)) = E_{k_2}(\Sigma(Y_1^i)) \oplus E_{k_3}(\Theta(Y_1^i)), i = 1, 2, \ldots, q,$$
where
$$\begin{aligned}\Sigma(Y_b^i) =& E_{k_1}(X_b^i[3]), b = 0, 1,\\ \Theta(Y_b^i) =& E_{k_1}(X_b^i[1]) \oplus E_{k_1}(X_b^i[2]) \oplus E_{k_1}(X_b^i[3]), b = 0, 1,\end{aligned}$$

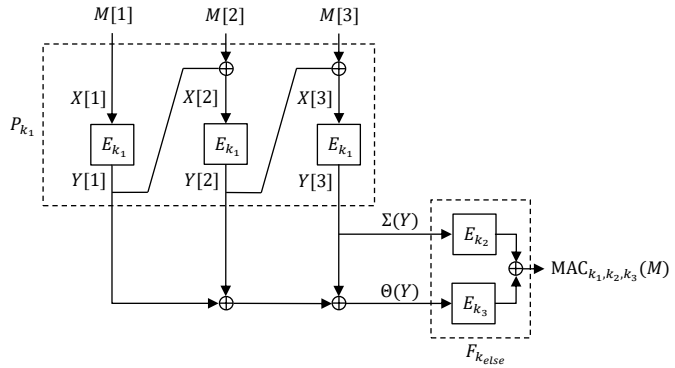


Fig. 8. 3kf9 with three-block message $M = (M[1], M[2], M[3])$.

and

$$\begin{aligned} X_b^i[1] &= E_k^{-1}(C_b^i[1]), \\ X_b^i[2] &= E_{k_1}(X_b^i[1]) \oplus C_b^i[1] \oplus E_k^{-1}(C_b^i[2]), \\ X_b^i[3] &= E_{k_1}(X_b^i[2]) \oplus C_b^i[2] \oplus E_k^{-1}(C_b^i[3]). \end{aligned}$$

We assume all $C_b^i[1], i = 1, \ldots, q, b = 0, 1$ are different. This can be realized easily. Then all $X_b^i[1], i = 1, \ldots, q, b = 0, 1$ are different from each other, which means we have to sample for $E_{k_1}(X_0^i[1])$ in every equation in (8). Similar as the PMAC_Plus in appendix 4.1, every equation $f(k, C_0^i) = f(k, C_1^i)$ holds with probability at most $\frac{q}{2^{n-3}}$. Therefore, the q equations happens with probability at most $(\frac{q}{2^{n-3}})^q$. For 3kf9, the key length $m \leq 2n$. Then when $q = 5$, we have $\Pr[test(k) = 1] \leq 2^{-2m}$ for $m \geq 24$ and any $k \in \{0,1\}^m \backslash \{k_1\}$.

4.2 Full Key Recovery Attack for PMAC_Plus-like MACs

Although one key recovery is enough to get successful forgery, we can further recover all keys of PMAC_Plus, 3kf9, and 2K-PMAC_Plus after knowing k_1. Their finalization functions all can be represented as the sum of two keyed permutations. That is to say, $k_{else} = (k_2, k_3)$ and for all $x, y \in \{0,1\}^n$ we have $F_{k_{else}}(x, y) = F_{k_2}(x) \oplus F'_{k_3}(y)$ where F, F' are two keyed permutations on n bits and k_2, k_3 are two m-bit keys. Our goal is to recover k_2, k_3. In fact, after recovering k_1, we are able to evaluate the inputs of $F_{k_{else}}(\cdot, \cdot)$ and get the output which is tag. That is to say, we are able to construct the quantum oracle of $F_{k_{else}}(\cdot, \cdot)$ by given the quantum oracle of $P_{k_1}^{-1}(\cdot)$ and $\mathrm{MAC}_{k_1, k_{else}}(\cdot)$. Let function $f : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ as $f(k, x_1, x_2) = F_k(x_1) \oplus F_k(x_2)$ for $k \in \{0,1\}^m, x_1, x_2 \in \{0,1\}^n$. Then we are able to know whether $k = k_2$ or not by whether $f(k, x_1, x_2) = F_{k_{else}}(x_1, y) \oplus F_{k_{else}}(x_2, y)$ or not. Applying Grover's algorithm we can recover k_2. Then the last unknown key k_3 can be recovered easily by Grover's algorithm as well. The whole attack costs $\mathcal{O}(2^{m/2})$ quantum queries and $\mathcal{O}(m + n)$ qubits.

5 Conclusions

In this paper, we introduce secret state recovery and key recovery for a series of BBB MACs in the Q2 model, leading to forgery attacks. Notice that PMAC_TBC3k handles message blocks with different tweakable block ciphers but not the same block cipher as other PMAC_Plus-like MACs in section 3.2. So we are not able to construct a period function and the secret state recovery attack is not suitable for it. Another notice is that SUM-ECBC-like MACs handle the message with two different hash block chains and have no linear combination processes. So we can't apply key recovery attack in section 4 to them. However, there is another key recovery attack. Take SUM-ECBC as an example. The complexity of the attack is $\mathcal{O}(2^m n)$ quantum queries assuming the size of message block is n bits and the size of all keys is $4m$ bits. We describe it in appendix E. The further question is if there is provable security in the quantum setting to show the tightness of the bound. We leave it as an open problem.

References

1. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. J. Comput. Syst. Sci. **61**(3), 362–399 (2000), https://doi.org/10.1006/jcss.1999.1694 7
2. Belovs, A., Spalek, R.: Adversary lower bound for the k-sum problem. In: Proceedings of the 4th conference on Innovations in Theoretical Computer Science. pp. 323–328 (2013) 3
3. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) Advances in Cryptology - EUROCRYPT 2013. Lecture Notes in Computer Science, vol. 7881, pp. 592–608. Springer (2013), https://doi.org/10.1007/978-3-642-38348-9_35 7
4. Bonnetain, X.: Tight bounds for Simon's algorithm. IACR Cryptol. ePrint Arch. **2020**, 919 (2020), https://eprint.iacr.org/2020/919 5, 7, 26, 29
5. Bonnetain, X., Jaques, S.: Quantum period finding against symmetric primitives in practice. CoRR **abs/2011.07022** (2020), https://arxiv.org/abs/2011.07022 30
6. Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics **305**, 53–74 (2002) 5
7. Brassard, G., Hoyer, P., Tapp, A.: Quantum algorithm for the collision problem. arXiv preprint quant-ph/9705002 (1997) 1, 2
8. Cogliati, B., Jha, A., Nandi, M.: How to build optimally secure prfs using block ciphers. IACR Cryptol. ePrint Arch. **2020**, 1097 (2020), https://eprint.iacr.org/2020/1097 2, 4
9. Cogliati, B., Jha, A., Nandi, M.: How to build optimally secure prfs using block ciphers. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 754–784. Springer (2020), https://doi.org/10.1007/978-3-030-64837-4_25 2, 4, 11, 15
10. Datta, N., Dutta, A., Nandi, M., Paul, G.: Double-block hash-then-sum: A paradigm for constructing BBB secure PRF. IACR Trans. Symmetric Cryptol.

2018(3), 36–92 (2018), https://doi.org/10.13154/tosc.v2018.i3.36-92 2, 4, 8, 9, 11, 15

11. Datta, N., Dutta, A., Nandi, M., Paul, G., Zhang, L.: Building single-key beyond birthday bound message authentication code. Tech. rep., Cryptology ePrint Archive, Report 2015/958, 2015. http://eprint. iacr. org ... (2015) 11, 15
12. Datta, N., Dutta, A., Nandi, M., Paul, G., Zhang, L.: Single key variant of PMAC_plus. IACR Trans. Symmetric Cryptol. **2017**(4), 268–305 (2017), https://doi.org/10.13154/tosc.v2017.i4.268-305 2, 4, 11, 15
13. Dong, X., Sun, S., Shi, D., Gao, F., Wang, X., Hu, L.: Quantum collision attacks on AES-like hashing with low quantum random access memories. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12492, pp. 727–757. Springer (2020). https://doi.org/10.1007/978-3-030-64834-3_25, https://doi.org/10.1007/978-3-030-64834-3_25 1, 2
14. Grassi, L., Naya-Plasencia, M., Schrottenloher, A.: Quantum algorithms for the k -xor problem. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018, Part I. Lecture Notes in Computer Science, vol. 11272, pp. 527–559. Springer (2018), https://doi.org/10.1007/978-3-030-03326-2_18 2
15. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, 1996. pp. 212–219 (1996), https://doi.org/10.1145/237814.237866 1, 5
16. Hodzic, S., Knudsen, L.R., Kidmose, A.B.: On quantum distinguishers for Type-3 generalized feistel network based on separability. In: Ding, J., Tillich, J. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020. Lecture Notes in Computer Science, vol. 12100, pp. 461–480. Springer (2020). https://doi.org/10.1007/978-3-030-44223-1_25, https://doi.org/10.1007/978-3-030-44223-1_25 1
17. Hosoyamada, A., Sasaki, Y.: Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 249–279. Springer (2020). https://doi.org/10.1007/978-3-030-45724-2_9, https://doi.org/10.1007/978-3-030-45724-2_9 1, 2
18. Ito, G., Hosoyamada, A., Matsumoto, R., Sasaki, Y., Iwata, T.: Quantum chosen-ciphertext attacks against Feistel ciphers. In: Matsui, M. (ed.) Topics in Cryptology - CT-RSA 2019 - The Cryptographers' Track at the RSA Conference 2019. Lecture Notes in Computer Science, vol. 11405, pp. 391–411. Springer (2019), https://doi.org/10.1007/978-3-030-12612-4_20 29
19. Iwata, T., Minematsu, K.: Stronger security variants of GCM-SIV. IACR Trans. Symmetric Cryptol. **2016**(1), 134–157 (2016), https://doi.org/10.13154/tosc.v2016.i1.134-157 4, 9, 10
20. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Advances in Cryptology - CRYPTO 2016, Proceedings, Part II. pp. 207–237 (2016), https://doi.org/10.1007/978-3-662-53008-5_8 1, 6, 7, 28
21. Kilian, J., Rogaway, P.: How to protect DES against exhaustive key search. In: Advances in Cryptology - CRYPTO '96. pp. 252–267 (1996), https://doi.org/10.1007/3-540-68697-5_20 3

22. Kilian, J., Rogaway, P.: How to protect DES against exhaustive key search (an analysis of DESX). J. Cryptology **14**(1), 17–35 (2001), https://doi.org/10.1007/s001450010015 3
23. Kim, S., Lee, B., Lee, J.: Tight security bounds for Double-Block Hash-then-Sum MACs. In: Advances in Cryptology - EUROCRYPT 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 435–465. Springer (2020), https://doi.org/10.1007/978-3-030-45721-1_16 2, 4, 9, 10
24. Kuwakado, H., Morii, M.: Security on the quantum-type Even-Mansour cipher. In: Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012. pp. 312–316 (2012), http://ieeexplore.ieee.org/document/6400943/ 1
25. Leander, G., May, A.: Grover meets simon - quantumly attacking the FX-construction. In: Advances in Cryptology - ASIACRYPT 2017, Proceedings, Part II. pp. 161–178 (2017), https://doi.org/10.1007/978-3-319-70697-9_6 1, 3, 6, 29
26. Leurent, G., Nandi, M., Sibleyras, F.: Generic attacks against beyond-birthday-bound MACs. In: Advances in Cryptology - CRYPTO 2018, Proceedings, Part I. pp. 306–336 (2018), https://doi.org/10.1007/978-3-319-96884-1_11 2, 4
27. McGrew, D.A., Viega, J.: The security and performance of the galois/counter mode (GCM) of operation. In: Progress in Cryptology - INDOCRYPT 2004. pp. 343–355 (2004), https://doi.org/10.1007/978-3-540-30556-9_27 10
28. Naito, Y.: Full PRF-secure message authentication code based on tweakable block cipher. In: Provable Security - 9th International Conference, ProvSec 2015. pp. 167–182 (2015), https://doi.org/10.1007/978-3-319-26059-4_9 4, 15
29. Naito, Y.: Blockcipher-based MACs: Beyond the birthday bound without message length. In: Advances in Cryptology - ASIACRYPT 2017, Proceedings, Part III. pp. 446–470 (2017), https://doi.org/10.1007/978-3-319-70700-6_16 2
30. Naya-Plasencia, M., Schrottenloher, A.: Optimal merging in quantum k-xor and k-xor-sum algorithms. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology - EUROCRYPT 2020, Part II. Lecture Notes in Computer Science, vol. 12106, pp. 311–340. Springer (2020), https://doi.org/10.1007/978-3-030-45724-2_11 2
31. Shen, Y., Wang, L., WengS, J.: Revisiting the security of DbHtS MACs: Beyond-birthday-bound in the multi-user setting. IACR Cryptol. ePrint Arch. **2020**, 1523 (2020), https://eprint.iacr.org/2020/1523 2
32. Simon, D.R.: On the power of quantum computation. SIAM J. Comput. **26**(5), 1474–1483 (1997), https://doi.org/10.1137/S0097539796298637 1, 6
33. Wagner, D.A.: A generalized birthday problem. In: Yung, M. (ed.) Advances in Cryptology - CRYPTO 2002. Lecture Notes in Computer Science, vol. 2442, pp. 288–303. Springer (2002). https://doi.org/10.1007/3-540-45708-9_19, https://doi.org/10.1007/3-540-45708-9_19 2
34. Wang, P., Feng, D., Wu, W.: HCTR: A variable-input-length enciphering mode. In: Information Security and Cryptology, First SKLOIS Conference, CISC 2005. Lecture Notes in Computer Science, vol. 3822, pp. 175–188. Springer (2005), https://doi.org/10.1007/11599548_15 10
35. Yasuda, K.: The sum of CBC macs is a secure PRF. In: Topics in Cryptology - CT-RSA 2010. pp. 366–381 (2010), https://doi.org/10.1007/978-3-642-11925-5_25 2, 4, 9
36. Yasuda, K.: A new variant of PMAC: beyond the birthday bound. In: Advances in Cryptology - CRYPTO 2011. pp. 596–609 (2011), https://doi.org/10.1007/978-3-642-22792-9_34 2, 4, 11, 14, 15

37. Zhang, L., Wu, W., Sui, H., Wang, P.: 3kf9: Enhancing 3gpp-mac beyond the birthday bound. In: Advances in Cryptology - ASIACRYPT 2012. Lecture Notes in Computer Science, vol. 7658, pp. 296–312. Springer (2012), https://doi.org/10.1007/978-3-642-34961-4_19 2, 4, 11, 12, 14, 15

A Quantum Basics

For two n-bit strings $x = x_1x_2 \dots x_n$ and $y = y_1y_2 \dots y_n$ where $x_i, y_i \in \{0,1\}$, the *inner product* of them is $x \cdot y = x_1y_1 \oplus \dots \oplus x_ny_n$.

Qubits. We call quantum bits as *qubits* . Let notation *ket* "$|\cdot\rangle$" represent a column vector. The n-qubit system is associated with the 2^n-dimension Hilbert space in complex field. Let the unit orthogonal basis of the Hilbert space be $\{|x\rangle\}$ where $x \in \{0,1\}^n$, which also is the basis of the n-qubit system. If we let $|x\rangle$ be an unit column vector whose x-th component is 1 and other components are 0. Then any n-qubit state can be represented as the linear combination of the basis:

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle = \begin{bmatrix} \alpha_{00\dots0} \\ \alpha_{00\dots1} \\ \dots \\ \alpha_{11\dots1} \end{bmatrix}.$$

where $\alpha_x \in \mathbb{C}$ and $\sum_{x\in\{0,1\}^n} |\alpha_x|^2 = 1$. It means any n-qubit state $|\psi\rangle$ is a unit length complex vector in the Hilbert space. If we measure $|\psi\rangle$, the superposition state will collapse into a basis state $|x\rangle$ with probability $|\alpha_x|^2$. Let notation *bra* "$\langle\cdot|$" represent a row vector. Then $|\psi\rangle = (|\psi\rangle)^\dagger = [\alpha^\dagger_{00\dots0}, \dots, \alpha^\dagger_{11\dots1}]$. We call $\langle\psi_1|\psi_2\rangle$ as *inner product* and $|\psi_1\rangle\langle\psi_2|$ as *outer product*. The *orthogonal basis* means the inner product of any two different vectors in the basis is equal to 0. For two independent quantum system $|\psi_1\rangle = \sum_{x_1\in\{0,1\}^n} \alpha_{x_1}|x_1\rangle$ and $|\psi_2\rangle = \sum_{x_2\in\{0,1\}^m} \alpha_{x_2}|x_2\rangle$, the joint state can be represented by *tensor product*: $|\psi_1\rangle \otimes |\psi_2\rangle = \sum_{x_1\in\{0,1\}^n} \sum_{x_2\in\{0,1\}^m} \alpha_{x_1}\alpha_{x_2}(|x_1\rangle \otimes |x_2\rangle)$,where $|x_1\rangle \otimes |x_2\rangle$ can be represented as $|x_1x_2\rangle$ as well.

Quantum Operations. Unitary operation (unitary matrix, unitary gate) U can transform a quantum state $|\psi_1\rangle$ to another quantum state $|\psi_2\rangle = U|\psi_1\rangle$. For a joint system of two independent quantum system $|\psi_1\rangle$ and $|\psi_2\rangle$, a joint quantum unit operation on the system can be represented as by tensor product $U \otimes V$, where $U \otimes V(|\psi_1\rangle \otimes |\psi_2\rangle) = (U|\psi_1\rangle) \otimes (V|\psi_2\rangle)$. There are some useful unitary operations. The first is Hadamard transform

$$H = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

on a single qubit. For example, when we apply it to state $|1\rangle$, we get $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ by

$$H|1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

If we apply H on $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ again, it is easy to know we will get $|1\rangle$ again. So H is the inverse of itself. Let $H^{\otimes n}$ be the operation that apply H to every qubit of n-qubit quantum state. Then for n-qubit basis state $|x\rangle$, we get $H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x\cdot y}|y\rangle$. Assume $|\psi_1\rangle, |\psi_2\rangle$ are n qubits state. The second unitary operation is $D_{|\psi_1\rangle} = 2|\psi_1\rangle\langle\psi_1| - I_{2^n}$. The transform $D_{|\psi_1\rangle}|\psi_2\rangle$ implements filpping the vector $|\psi_2\rangle$ with $|\psi_1\rangle$ as the symmetry axis, which is the core operation of Grover's algorithm.

Quantum Queries. Let $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$ be a quantum oracle for implementing function $f : \{0,1\}^n \to \{0,1\}^m$, where $|y\rangle$ is ancilla m qubits and $|x\rangle, |y\rangle$ are basis states. For $f : \{0,1\}^n \to \{0,1\}$, there is another available quan-

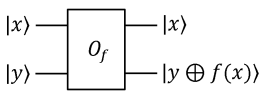


Fig. 9. The oracle $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$.

tum oracle O'_f, which is constructed from O_f by the following quantum circuit. The input of the quantum circuit is $|x\rangle|1\rangle$ and the output is $(-1)^{f(x)}|x\rangle|1\rangle$. If we

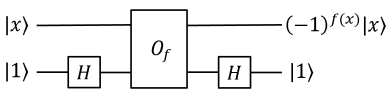


Fig. 10. The quantum circuit to construct oracle O'_f from oracle O_f.

neglect the last qubit $|1\rangle$, then we get the quantum oracle $O'_f : |x\rangle \to (-1)^{f(x)}|x\rangle$. In fact, for state

$$\begin{aligned}
|\psi\rangle &= \sum_{x\in\{0,1\}^n} \alpha_x|x\rangle \\
&= \sum_{f(x)=0} \alpha_x|x\rangle + \sum_{f(x)=1} \alpha_x|x\rangle \\
&= \sqrt{\sum_{f(x)=0} \alpha_x^2} \sum_{f(x)=0} \frac{\alpha_x}{\sqrt{\sum_{f(x)=0} \alpha_x^2}}|x\rangle + \sqrt{\sum_{f(x)=1} \alpha_x^2} \sum_{f(x)=1} \frac{\alpha_x}{\sqrt{\sum_{f(x)=1} \alpha_x^2}}|x\rangle \\
&= \cos\theta|\psi_0\rangle + \sin\theta|\psi_1\rangle,
\end{aligned}$$

$O'_f|\psi\rangle = O'_f(\cos\theta|\psi_0\rangle + \sin\theta|\psi_1\rangle) = \cos\theta|\psi_0\rangle - \sin\theta|\psi_1\rangle$. That is to say, O'_f flip the vector $|\psi\rangle$ with $|\psi_0\rangle$ as its symmetry axis.

In fact, in EUF-qCPA the quantum adversary maintains its state as follows. Let $|0^n\rangle$ be the initial state of adversary. Let O_i be the i-th quantum oracle query for adversary of MAC function and let $U_0, U_1, \ldots, U_q$ be the unitary operations applied by adversary between queries. Then after q quantum queries, the final state of adversary will be $U_q O_q \ldots U_1 O_1 U_0 |0^n\rangle$. Finally, the adversary applies the final state to get some useful information and make forgeries.

Quantum Complexity. There are three dimensions to measure the complexities of a quantum algorithm: *query complexity*, *time complexity*, *memory complexity*. The *query complexity* counts the number of the superposition oracle queries O_f used for function f. Notice that the classical queries are specific cases of superposition queries. So we add the number of classical queries to query complexity in the quantum algorithm. The *time complexity* is the number of quantum operations (gates, unitaries). The *memory complexity* is the number of qubits in a quantum circuit. In our work, the time complexity of the quantum algorithm is close to query complexity, so we only consider query complexity and memory complexity.

B Grover's Algorithm and Proof of Theorem 1

B.1 Grover's Algorithm

The Grover's algorithm consists of a series of Grover's routines. Before all iterations, when we measure $|\psi\rangle$ the initial probability to get a u who satisfies $test(u) = 1$ is small. However, every routine of the algorithm will amplify the amplitude of such elements. When the amplitude of such elements is large enough, then when we measure the state we will get a u who satisfies $test(u) = 1$. In our paper, we will apply the Grover algorithm to find some hidden useful information, such as the correct secret key. The quantum circuit of Grover's algorithm is showed figure 11 and the algorithm is showed in algorithm 1.

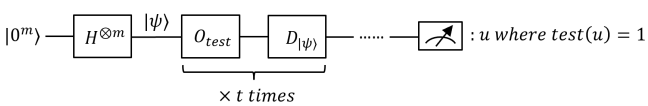


Fig. 11. The quantum circuit of Grover's algorithm.

Firstly, we divide the initial superposition state $|\psi\rangle = H^{\otimes m}|0^m\rangle$ $= 2^{-\frac{m}{2}} \sum_{u\in\{0,1\}^m} |u\rangle$ as two parts by whether $test(u) = 0$ or not. Let $\theta = \arcsin\sqrt{\frac{e}{2^m}}$, $|\psi_0\rangle = \sum_{test(u)=0} \frac{1}{\sqrt{2^m - e}}|u\rangle$, $|\psi_1\rangle = \sum_{test(u)=1} \frac{1}{\sqrt{e}}|u\rangle$. Then the initail state $|\psi\rangle = \cos\theta|\psi_0\rangle + \sin\theta|\psi_1\rangle$. It is easy to know that $|\psi_1\rangle$ and $|\psi_0\rangle$ are two orthogonal unit vectors. Then we can establish a coordinate system with $|\psi_0\rangle$

Algorithm 1 Grover's Algorithm

Input: $m, t, test : \{0,1\}^m \to \{0,1\}$
Output: u who satisfies $test(u) = 1$
Let $O_{test}|u\rangle = (-1)^{test(x)}|u\rangle, D_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - I_{2^m}$.
1: Initialize m qubits registers $|0^m\rangle$;
2: Apply $H^{\otimes m}$ to obtain $|\psi\rangle = H^{\otimes m}|0^m\rangle$;
3: Repeat Grover's routines with t times to get $|\phi\rangle = (D_{|\psi\rangle}O_{test})^t|\psi\rangle$;
4: Measure $|\phi\rangle$ in order to get u who satisfies $test(u) = 1$;
5: **return** u;

and $|\psi_1\rangle$ as its orthogonal coordinate axis. In this coordinate system, the vector $|\psi\rangle$ is a unit-length vector with angle θ. In the first Grover routine, the query O_{test} flip the state $|\psi\rangle$ with $|\psi_0\rangle$ as the symmetry axis to a unit-length vector whose angle is $-\theta$, i.e., $\cos\theta|\psi_0\rangle - \sin\theta|\psi_1\rangle$. Then the flip operation $D_{|\psi\rangle}$ will flip vector $O_{test}|\psi\rangle$ with $|\psi\rangle$ as its symmetry axis to get a unit-length vector whose angle is 3θ, i.e., $\cos 3\theta|\psi_0\rangle + \sin 3\theta|\psi_1\rangle$. We show the above process in figure 12. It is easy to know that every iteration will add an angle 2θ. After t Grover iterations, we will get a quantum state $|\phi\rangle = \cos((2t+1)\theta)|\psi_0\rangle + \sin((2t+1)\theta)|\psi_1\rangle$. For $t = \lceil \frac{\pi}{4\theta} \rceil$, the final state $|\phi\rangle$ will be close to $|\psi_1\rangle$ and we will get a good elements with probability almost 1.

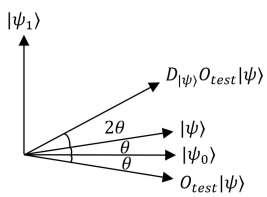


Fig. 12. The effect of the first Grover's routine.

In the above Grover algorithm, we amplify some amplitudes of a uniform superposition state $|\psi\rangle = 2^{-m/2}\sum_{u\in\{0,1\}^m}|u\rangle$, which in produced by $H^{\otimes m}$ on $|0^n\rangle$. In the following, we will introduce a more general Grover algorithm: amplitude amplification algorithm (algorithm 2). It can amplify some amplitudes of any quantum state $|\psi\rangle = \sum_{u\in\{0,1\}^m}\alpha_u|u\rangle$ as long as we can produce $|\psi\rangle$ by a unitary operation U on $|0^m\rangle$.

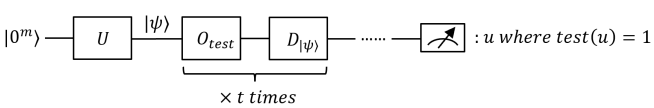


Fig. 13. The quantum circuit of amplitude amplification algorithm.

Algorithm 2 Amplitude Amplification Algorithm

Input: $m, t, test : \{0,1\}^m \to \{0,1\}$, unitary operation U
Output: u who satisfies $g(u) = 1$
Let $O_{test}|u\rangle = (-1)^{test(u)}|u\rangle$, $D_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - I_{2^m}$.
1: Initialize m qubits registers $|0^m\rangle$;
2: Apply U to obtain $|\psi\rangle = U|0^m\rangle$;
3: Repeat Grover's routines with t times to get $|\phi\rangle = (D_{|\psi\rangle}O_{test})^t|\psi\rangle$;
4: Measure $|\phi\rangle$ in order to get a u who satisfies $test(u) = 1$;
5: **return** u;

B.2 Proof of Theorem 1

Firstly, we prove the following lemma.

Lemma 1. (Adapted from [4]) *Let* $test : \{0,1\}^m \to \{0,1\}$, $\mathcal{U}$ *as defined in definition 2.1. Assume in algorithm 2 the initial probability to get a* $u \in \mathcal{U}$ *after measuring* $|\psi\rangle$ *is* p_0 *and the quantum implement of* $test(\cdot)$ *costs* j *qubits. Then amplitude amplification algorithm with* $t = \lceil \frac{\pi}{4 \arcsin \sqrt{p_0}} \rceil$ *quantum queries to* $test(\cdot)$ *and* $\mathcal{O}(m + j)$ *qubits will output a* $u \in \mathcal{U}$ *with probability at least* $\frac{p_0}{p_0+p_1}[1 - (\frac{p_1}{p_0} + \sqrt{p_0 + p_1} + 2\sqrt{1 + \frac{p_1}{p_0}}^3 p_0)^2]$.

Proof. From lemma 1 we get the initial probability of measuring $|\psi\rangle$ to get a u who satisfies $test(u) = 1$ is between $[p_0, p_0 + p_1]$. Paper [4] has proofed when the initial probability is between $[p_0, p_0 + p_1]$, then after $t = \lceil \frac{\pi}{4 \arcsin \sqrt{p_0}} \rceil$ Grover's routines the probability to get a u who satisfies $test(u) = 1$ is at least $1 - (\frac{p_1}{p_0} + \sqrt{p_0 + p_1} + 2\sqrt{1 + \frac{p_1}{p_0}}^3 p_0)^2$. Among all u who satisfies $test(u) = 1$, the proportion of $u \in \mathcal{U}$ is at least $\frac{p_0}{p_0+p_1}$. Multiple them and then we can get the lower bound of the probability of getting a $u \in \mathcal{U}$.

By setting $U = H^{\otimes m}$ in amplitude amplification algorithm (algorithm 2), We obtain Grover's algorithm (algorithm 1) and $p_0 = \frac{e}{2^m}$. By lemma 1, we prove theorem 1.

C Simon's Algorithm and Proof of theorem 2

C.1 Simon's Algorithm

Simon's algorithm consist of many Simon's routines. The quantum circuit and the quantum algorithm of of a Simon's routine is showed in figure 15 and algorithm 3. If f is a periodic function with period s, Simon's routine outputs $v_i \in \{0,1\}^n$ who is perpendicular to the period s. Assume l Simon's routines output $v_1, v_2, \ldots, v_l$. If $v_1, v_2, \ldots, v_l$ span the whole space $\{0^n, s\}^\perp$, then we can get the nontrivial period s by solving the equation system $s \cdot v_i = 0, i = 1, \ldots, l$.

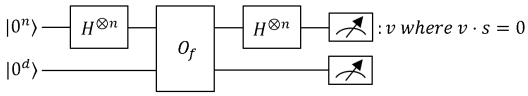


Fig. 14. The quantum circuit of Simon's routine.

Algorithm 3 Simon's routine

Input: $n, d, f : \{0,1\}^n \to \{0,1\}^d$ who has a hiddden period s
Output: v who satisfies $v \cdot s = 0$
Let $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$.
1: Initialize $n + d$ qubits registers $|0^n\rangle|0^d\rangle$;
2: Apply $U = (H^{\otimes n} \times I_{2^d})O_f(H^{\otimes n} \times I_{2^d})$ on $|0^n\rangle|0^d\rangle$ to get

$$|\psi\rangle = 2^{-n} \sum_{v \in \{0,1\}^n} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot v} |v\rangle|f(x)\rangle;$$

3: Measure $|\psi\rangle$ and get the first n-bit v;
4: **return** v;

Algorithm 4 Simon's algorithm

Input: $n, d, l, f : \{0,1\}^n \to \{0,1\}^d$ who has a hiddden period s
Output: the period s
Let $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$.
1: Initialize $n + d$ qubits registers $|0^n\rangle|0^d\rangle$;
2: **For** $i = 1$ to l **do**
3: Apply $U = (H^{\otimes n} \times I_{2^d})O_f(H^{\otimes n} \times I_{2^d})$ on $|0^n\rangle|0^d\rangle$ to get

$$|\psi_i\rangle = 2^{-n} \sum_{v_i \in \{0,1\}^n, x_i \in \{0,1\}^n} (-1)^{x_i \cdot v_i} |v_i\rangle|f(x_i)\rangle;$$

4: Measure $|\psi_i\rangle$ to get the first n-bits values v_i;
5: **end for**
6: Compute the period s by solving the equation system $s \cdot v_i = 0, i = 1, 2, \ldots, l$;
7: **return** s;

The whole Simon's algorithm is in algorithm 4.

In fact, we can parallel Simon's routines to construct Simon's algorithm as in algorithm 5. The quantum circuit of algorithm 5 is in figure 15.

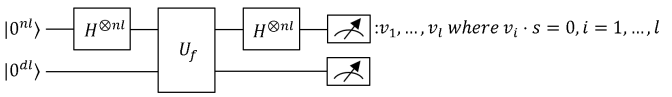


Fig. 15. The quantum circuit of Simon's algorithm.

Algorithm 5 Simon's algorithm

Input: $n, d, l, f : \{0,1\}^n \to \{0,1\}^d$ who has a hiddden period s
Output: the period s
Let $O_f : |x\rangle|y\rangle \to |x\rangle|y \oplus f(x)\rangle$, $U_f|x_1\rangle \ldots |x_l\rangle|y_1\rangle \ldots |y_l\rangle \to |x_1\rangle \ldots |x_l\rangle|y_1 \oplus f(x_1)\rangle \ldots |y_l \oplus f(x_l)\rangle$ with l calls to O_f.
1: Initialize $nl + dl$ qubits registers $|0^{nl}\rangle|0^{dl}\rangle$;
2: Apply $U = (H^{\otimes nl} \times I_{2^{dl}})U_f(H^{\otimes nl} \times I_{2^{dl}})$ on $|0^{nl}\rangle|0^{dl}\rangle$ to get

$$|\psi\rangle = 2^{-nl} \sum_{\substack{v_1,\ldots,v_l \in \{0,1\}^n, \\ x_1,\ldots,x_l \in \{0,1\}^n}} (-1)^{x_1 \cdot v_1}|v_1\rangle \ldots (-1)^{x_l \cdot v_l}|v_l\rangle|f(x_1)\rangle \ldots |f(x_l)\rangle;$$

3: Measure $|\psi\rangle$ to get the first nl-bits values $v_1, v_2, \ldots, v_l$;
4: Compute the period s by solving the equation system $s \cdot v_i = 0, i = 1, 2, \ldots, l$;
5: **return** s;

C.2 Proof of Theorem 2

Proof. Firstly, let us focus on Simon's routine. Kaplan et al (Appendix A in [20]) have proved for $t \in \{0,1\}^n \setminus \{0^n\}$, therefore is a relationship between the probability of get a v who satisfies $v \cdot t = 0$ after measuring $|\psi\rangle$ and the proportion of x who satisfies $f(x) = f(x \oplus t)$ among $\{0,1\}^n$. It is

$$\Pr_v[v \cdot t = 0] = \frac{1}{2}(1 + \Pr_x[f(x) = f(x \oplus t)]). \tag{9}$$

If $t = s$, we have $\Pr_x[f(x) = f(x \oplus s)] = 1$, which leads to $\Pr_v[v \cdot s = 0] = 1$ by equation (9). That is to say, for function f with period s, we can always get a v who satisfies $v \cdot s = 0$ after Simon's routine. By

$$\varepsilon(f) = \max_{t \in \{0,1\}^n \setminus \{0^n, s\}} \Pr_x[f(x) = f(x \oplus t)],$$

we have $\Pr_v[v \cdot t = 0] \leq \frac{1}{2}(1 + \varepsilon(f))$ for $t \in \{0,1\}^n \backslash \{0^n, s\}$. That is to say, the probability of getting a v who satisfies $v \cdot t = 0$ for $t \in \{0,1\}^n \backslash \{0^n, s\}$ is at most $\frac{1}{2}(1 + \varepsilon(f))$.

Now, let us focus on Simon's algorithm. The line 1 to 3 are l parallel Simon routines. Then $v_1, \ldots, v_l$ are all satisfy $v_i \cdot s = 0, i \in \{1, \ldots, l\}$. Therefore, the space spanned by $v_1, \ldots, v_l$ is the subspace of $\{0^n, s\}^\perp$. If the space spanned by $v_1, \ldots, v_l$ is equal to $\{0^n, s\}^\perp$, then we can get s by solving the equation system $v_i \cdot s = 0, i = 1, \ldots, l$. However, Simon's algorithm may fail when there is at least one $t \in \{0,1\}^n \backslash \{0^n, s\}$ such that $v_i \cdot t = 0, i = 1, \ldots, l$. The probability of this bad case is at most $2^n \cdot (\frac{1+\varepsilon(f)}{2})^l$. Let $l = cn$ then we get theorem 2.

D Grover-meet-Simon Algorithm and Proof of Theorem 3

D.1 Grover-meet-Simon Algorithm

For Grover-meet-Simon problem in definition 2.1, Leader and May [25] firstly propose Grover-meet-Simon algorithm to solve it. The main idea is to search $u \in \mathcal{U}$ by Grover's algorithm and in every Grover's routine check whether or not each $u \in \mathcal{U}$ by whether $f(u, \cdot)$ is periodic or not, which can be implemented by Simon's algorithm. Assume the l parallel Simon routines in Simon's algorithm output $v_1, \ldots, v_l$. For simplicity, we only check whether or not the rank of $v_1, \ldots, v_l$ is at most $n-1$ instead of whether $f(u, \cdot)$ is periodic or not, this the first proposed in [18] and then combined with Grover's algorithm in [4]. The replacement is available for the following reason. For $u \in \mathcal{U}$, $f(u, x)$ is a periodic function. Thus the space spanned by $v_1, \ldots, v_l$ is the subspace of $\{0^n, s\}^\perp$. So the rank of such space is no more than $n-1$. However, for $u \notin \mathcal{U}$, $f(u, x)$ is an aperiodic function. Thus the space spanned by $v_1, \ldots, v_l$ is the subspace of $\{0,1\}^n$. For sufficient large l, the $v_1, \ldots, v_l$ can span the whole space $\{0,1\}^n$. We let the the output of the test function be 1 when the rank of $\{v_1, \ldots, v_l\}$ is at most $n-1$. Otherwise, it is 0. Therefore, Grover's routine will amplify the amplitude of $u \in \mathcal{U}$. At last, we can get a $u \in \mathcal{U}$ and its corresponding $v_1, \ldots, v_l$. Like the Simon's algorithm, we can get s_u by solving the equation system $v_i \cdot s = 0, i = 1, \ldots, l$ in the end. The whole Grover-meet-Simon is in algorithm 6 and the quantum circuit is in figure 16. More accurately, the whole algorithm is an amplitude amplification algorithm (algorithm 2) with Hadamard transform and the parallel Simon's routines without measurement as the unitary operation U in algorithm 2.

D.2 Proof of Theorem 3

Proof. When $u \in \mathcal{U}$, the classifier function *test* will output 1. If we measure $|\psi\rangle$, it is easy to know the probability to get a $u \in \mathcal{U}$ is $\frac{e}{2^m}$. For $u \notin \mathcal{U}$, if there is at least one $t \in \{0,1\}^n \backslash \{0^n\}$ who satisfies $t \cdot v_i = 0, i = 1, \ldots, l$, then *test* output

1 as well. By

$$\varepsilon(f) = \max_{(u,t)\in\{0,1\}^m\times\{0,1\}^n\setminus(\mathcal{U}_s\cup\{0,1\}^m\times\{0^n\})} \Pr_x[f(u,x) = f(u,x\oplus t)],$$

this case happens with probability at most $2^n\cdot(\frac{1+\varepsilon(f)}{2})^l$. By lemma 1, we will get the lower bound of the probability of get a $u\in\mathcal{U}$ after measuring $|\phi\rangle$. For $u\in\mathcal{U}$, Simon's algorithm with function $f(u,\cdot)$ output the period s_u with probability at least $1-2^n\cdot(\frac{1+\varepsilon(f)}{2})^l$. Multiple them and then we can get lower bound of the probability of get a tuple $(u,s_u)\in\mathcal{U}_s$. Let $l=cn$. By paper [5], we get the qubits of these algorithm is $\mathcal{O}(m+cn^2+cdn)$. Now, we have proved the theorem 3.

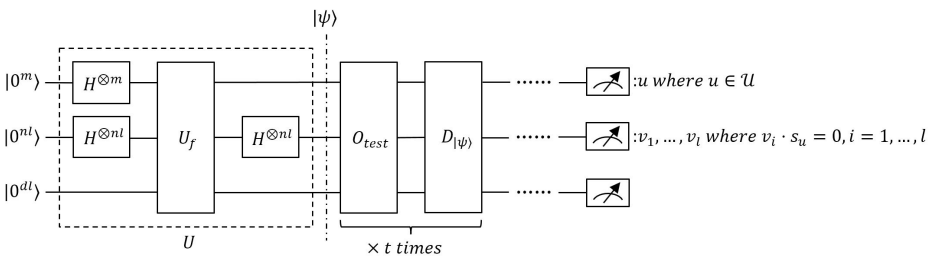


Fig. 16. The quantum circuit of Grover-meet-Simon algorithm.

E Key Recovery Attack for SUM-ECBC

Let $b\in\{0,1\}, x\in\{0,1\}^n$. Similar as introduction (section 1, strategy 1), we construct $C^{\mathrm{MAC}_{k_1,k_2,k_3,k_4}}(b,x) = g_{k_1,k_2}(b,x)\oplus h_{k_3,k_4}(b,x)$ from SUM-ECBC through method C, where $g_{k_1,k_2}(b,x)$ and $h_{k_3,k_4}(b,x)$ have periods $1\|s_1$ and $1\|s_2$ respectively and k_1,k_2,k_3,k_4 are keys. Then we construct a function $f: \{0,1\}^m\times\{0,1\}^m\times\{0,1\}\times\{0,1\}^n\to\{0,1\}^n$ as

$$\begin{aligned} f_{k_1,k_2,k_3,k_4}(k_3',k_4',b,x) =& C^{\mathrm{MAC}_{k_1,k_2,k_3,k_4}}(b,x)\oplus h_{k_3',k_4'}(b,x)\\ =& g_{k_1,k_2}(b,x)\oplus h_{k_3,k_4}(b,x)\oplus h_{k_3',k_4'}(b,x). \end{aligned}$$

When $(k_3',k_4') = (k_3,k_4)$, f equals $g_{k_1,k_2}(b,x)$ and have a period $1\|s_1$. By applying Grover-meet-Simon algorithm, we can recover k_3,k_4,s_1, which leads to a forgery attack. After recover k_3,k_4, it is easily to recover k_1,k_2 by Grover's search. Either the forgery attack or full key recover attack costs $\mathcal{O}(2^m n)$ quantum queries with $\mathcal{O}(m+n^2)$ qubits by theorem 3 and theorem 1.

Algorithm 6 Grover-meet-Simon Algorithm

Input: $m, n, r, l, t, f : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^d$, for $u \in \mathcal{U} \subseteq \{0,1\}^m$ that $f(u,\cdot)$ is a periodic function, otherwise it is an aperiodic function.

Output: a good element x

Let $O_f|u\rangle|x\rangle|y\rangle = |u\rangle|x\rangle|y \oplus f(u,x)\rangle, U_f|u\rangle|x_1\rangle \dots |x_l\rangle|y_1\rangle \dots |y_l\rangle = |u\rangle|x_1\rangle \dots |x_l\rangle|y_1 \oplus f(u,x_1)\rangle \dots |y_l \oplus f(u,x_l)\rangle$ with l calls to $O_f, test : \{0,1\}^{m+nl+dl} \to \{0,1\}$ with

$$test(u, v_1, \dots, v_l, y_1, \dots, y_l) = \begin{cases} 1, \dim\{v_1, \dots, v_l\} \leq n-1 \\ 0, \dim\{v_1, \dots, v_l\} = n \end{cases},$$

$O_{test}|u, v_1, \dots, v_l, y_1, \dots, y_l\rangle = (-1)^{test(u, v_1, \dots, v_l, y_1, \dots, y_l)}|u, v_1, \dots, v_l, y_1, \dots, y_l\rangle, D_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - I_{2^m}$.

1: Initialize $m + nl + dl$ qubits registers $|0^m\rangle|0^{nl}\rangle|0^{dl}\rangle$;

2: Apply $U = (I_{2^m} \otimes H^{\otimes nl} \otimes I_{2^{dl}})U_f(H^{\otimes m} \otimes H^{\otimes nl} \otimes I_{2^{dl}})$ to $|0^m\rangle|0^{nl}\rangle|0^{dl}\rangle$ to obtain

$$|\psi\rangle = 2^{-(\frac{m}{2}+nl)} \sum_{\substack{u \in \{0,1\}^m, \\ v_1, \dots, v_l \in \{0,1\}^n, \\ x_1, \dots, x_l \in \{0,1\}^n}} |u\rangle(-1)^{x_1 \cdot v_1}|v_1\rangle \dots (-1)^{x_l \cdot v_l}|v_l\rangle|f(u,x_1)\rangle \dots |f(u,x_l)\rangle;$$

3: Repeat Grover's routines with t times to get $|\phi\rangle = (D_{|\psi\rangle}O_{test})^t|\psi\rangle$;

4: Measure $|\phi\rangle$ to in order to get the first $(m + nl)$-bit values $u \in \mathcal{U}$ and $v_1, \dots, v_l$ who satisfy $s_u \cdot v_i = 0, i = 1, \dots, q$;

5: Compute the period s_u by solving the equation system $s_u \cdot v_i = 0, i = 1, 2, \dots, l$;

6: **return** u, s_u;