Breaking the quadratic barrier: Quantum cryptanalysis of Milenage, telecommunications' cryptographic backbone

Vincent Ulitzsch
Technische Universität Berlin
Berlin, Germany
vincent@sect.tu-berlin.de

Jean-Pierre Seifert
Technische Universität Berlin
Berlin, Germany
jean-pierre.seifert@tu-berlin.de

ABSTRACT

The potential advent of large-scale quantum computers in the near future poses a threat to contemporary cryptography. Without doubt, one of the most active and ubiquitous usage of cryptography is currently present in the very vibrant field of cellular networks, i.e., 3G, 4G, 5G and 6G, which is already in the planning phase. The entire cryptography of cellular networks is centered around seven secret-key algorithms $f_1, \ldots, f_5, f_1^*, f_5^*$, aggregated into an "authentication and key agreement" algorithm set. Still, these secret key algorithms have not yet been subject to quantum cryptanalysis. Instead, many quantum security considerations for telecommunication networks argue that the threat posed by quantum computers is restricted to public-key cryptography. On the other hand, the only threat to secret-key algorithms would stem from the famous Grover quantum search algorithm, which admits a general square root speedup of all oracle based search problems, thus resulting in an effectively halved key length of the above algorithms. However, various recent works have presented quantum attacks on secret key cryptography that result in more than a quadratic speedup. These attacks call for a re-evaluation of quantum security considerations for cellular networks, encompassing a quantum cryptanalysis of the secret-key primitives used in cellular security. In this paper, we conduct such a quantum cryptanalysis for the Milenage algorithm set, the prevalent instantiation of the seven secret-key algorithms that underpin cellular security. Building upon recent quantum cryptanalytic results, we show attacks that go beyond a quadratic speedup. Concretely, we provide for all Milenage algorithms various quantum attack scenarios, including exponential speedups distinguishable by different quantum attack models. The presented attacks include a polynomial time quantum existential forgery attack, assuming an attacker has access to a superposition oracle of Milenage and key recovery attacks that reduce the security margin beyond the quadratic speedup of Grover. Our results do not constitute an immediate quantum break of the Milenage algorithms, but they do provide strong evidence against choosing Milenage as the cryptographic primitive underpinning the security of quantum resistant telecommunication networks.

1 INTRODUCTION

Telecommunication operators are evidently expecting the advent of general purpose quantum computers, as indicated by their funding of various research projects investigating the new technologies' potential [27]. As part of these efforts, telecommunication standardization bodies also pay increasing attention to post-quantum security in telecommunication networks. As a result, the sixth generation of telecommunication networks (6G) is intended to be post-quantum secure, and proposals for extensions of the fifth generation (5G) already integrate quantum security considerations, cf. [13]. These security considerations are often based on the assumption that the threat posed by quantum computers is restricted to asymmetric cryptography. In contrast, symmetric cryptography would assume — up to a quadratic speed-up of exhaustive search due to Grover's algorithm — to be unaffected by quantum cryptananalysis. Hence, so the argument goes, increasing the key size of symmetric cryptography used in 6G to 256-bit would provide sufficient protection against quantum adversaries [24, 33].

In light of recent quantum cryptanalytic results however, this common belief can no longer be assumed to be trivially true. Indeed, it has been shown in various works that, depending on the assumed attacker capabilities, quantum computers can be used to either efficiently break certain symmetric-key cryptography schemes or significantly reduce the time needed to attack them [8, 21, 30]. The distinguishing feature in the attacker capabilities for quantum cryptanalysis is the kind of oracle access that is provided to the attacker. In the Q_1 model, the attacker has only classical access to an encryption oracle, but "offline" access to a quantum computer. In the Q_2 setting, also called the quantum known plaintext attack, the attacker can make superposition queries to an encryption oracle. This capability allows the attacker to leverage powerful quantum algorithms, the most prominent one being Simon's algorithm for quantum period finding, cf. [31]. For example, in the Q_2 setting, Simon's algorithm enables attackers to execute forgery attacks against an otherwise *classically* secure CBC-MACs in polynomial time [21].

Therefore, these results call for a careful re-evaluation of the truism that has guided quantum security considerations for 6G so far. Doubling the key-size might not be sufficient to ensure long-term security of telecommunication protocols. Instead, symmetric-key cryptographic schemes used in telecommunications protocols must be evaluated towards their resilience against quantum enabled adversaries as well.

Contributions. We conduct such a quantum cryptanalysis for the Milenage algorithm set, a set of symmetric-key cryptographic algorithms ubiquitously used for authentication and key derivation in the cellular world. The Milenage algorithms, consisting of five functions $f_1, \ldots, f_5$, are used to authenticate a subscriber towards a cellular network. The authentication procedure, a part of the Authentication and Key Agreement Protocol (AKA), leverages the Milenage algorithms in a challenge-response protocol. All Milenage algorithms make use of the network authentication key K, a secret key shared between the subscriber (stored in his network provider's SIM card) and the network. Breaking the Milenage algorithm set would therefore allow attackers to perform account takeover attacks. Thus, the security of Milenage algorithms is crucial for the security of pervasive cellular networks in general. As such, the algorithms underpin the security of the worldwide cellular networks and provide a great starting point for the required quantum cryptanalysis of symmetric ciphers.

In conducting the quantum cryptanalysis, we take a gentle approach that can be followed by researchers who are not familiar with the internals of quantum computing as well. First, in Section 3, we distill a *quantum toolbox* from the various works on quantum cryptanalysis and quantum algorithms, i.e., a minimum set of quantum algorithms and results about their complexity that have proven to be useful in quantum cryptanalysis. For each algorithm in the toolbox, we explain the requirements that an attacker needs to meet in order to use the respective algorithm. For example, whether a quantum algorithm requires superposition access or can also be executed with only classical oracle access to the encryption under attack. Once equipped with this quantum toolbox, no more detailed understanding of quantum computing is required. The attacker then only needs to construct a function that meets the respective requirements, after which the algorithms can be applied as a black-box.

Leveraging this minimum quantum toolbox, we develop multiple attacks on the Milenage algorithm set inspired by various prior works. The quantum cryptanalysis of Milenage is the main contribution of this paper and can be found in Section 4. We analyze the Milenage set from several dimensions. In the two different query models Q_1 and Q_2, considering different attacker goals such as full key recovery or existential forgery and considering more powerful attacker models such as the related key model. Our results show that the quantum toolbox can be utilized to provide speedups in all dimensions, and even leads to polynomial time attacks in the Q_2 model. As a helpful overview, Table 1 summarizes the breadth of our results. Our attacks imply that when considering the most powerful quantum adversaries, Milenage does not fulfill the security requirements that are specified in the corresponding design document [3]. Less powerful adversaries are still able to significantly speed up their attacks, albeit not to an extent that fully breaks the algorithm set in polynomial time.

Our results call into question whether quantum-resilient cellular networks can still rely on the Milenage algorithm set as their cryptographic foundation. The Milenage algorithms exhibit a structure that can be exploited by quantum computers to obtain attacks far more efficiently than Grover's search. This not only voids any chance to transfer proofs of Milenage's pseudorandomness to the quantum setting [15, 34]. It also entails that in the Q_2 model, Milenage must be even considered broken. Although the superposition access required in the Q_2 model is a very strong assumption and not trivially feasible, it would undermine security best practices to use Milenage in post-quantum secure networks. It is standard security best practice to pick cryptographic schemes which are secure against more powerful adversaries over schemes which are not [32]. This practice is motivated by mainly two reasons. First, attacks that can be executed by a powerful adversary could point to potential weaknesses against less powerful adversaries as well. Second, dismissing the powerful attacker model might be too optimistic; superposition queries could turn out to be feasible through either physical breakthroughs or unforeseen and overlooked use-cases of the analyzed cryptographic scheme. Indeed, visions for 6G networks demand that the cryptography in use "cannot be broken even by quantum computers of arbitrary complexity" ([13]).

A misjudgement and subsequent standardization of an insecure cryptographic foundation of 6G would be a post-quantum disaster. Our work however shows that quantum security considerations for cryptography used in telecommunication need to go beyond a trivial Grover attack. The threat of quantum attacks against Milenage might soon become reality. Compared to past surveys, experts in the field of quantum computing tend to view a quantum computer capable of breaking RSA-2048 as increasingly more likely [26]. The attacks presented here require a significantly lower amount of qubits than algorithms capable of breaking RSA-2048, and can thus be expected to be feasible much earlier. We discuss the implications of the present results as well as the potential for Milenage alternatives in Section 5.

In light of our results, two urgent tasks present themselves. First, further evaluation of symmetric cryptography in telecommunication networks needs to be guided by a more refined notion of post-quantum security. To this end, the desired security margins and the intended use-case for a cryptographic scheme in cellular protocols should determine the quantum attacker model. Second, more quantum cryptanalysis of the symmetric cryptography used in telecommunication networks is required. Once the quantum security requirements are established, standardization bodies can then draw on quantum cryptanalytical results such as the present one to decide whether a cryptographic scheme is suitable for usage in quantum resilient cellular networks.

2 PRELIMINARIES

2.1 Notation

Throughout this paper, we will make use of a block cipher encryption function E, which takes as input an m-bit message, an n-bit key and returns an m-bit output. We denote by $E_K[m]$ the encryption of bit-string m under block cipher E with secret k. Similarly, if a function f takes as input a secret key k and a message m, we denote by $f_k(m)$ the invocation of that function with k and message m. For a bit-string $x \in \{0,1\}^*$, we denote by $|x|$ the length of the bit-string. We write 0^n to denote the bit-string of n zeros.

Additionally, we define the function $rot_r(x)$ and $rot_r^{-1}(x)$ which are the results of cyclically rotating the 128-bit value x by r bit positions towards the most significant or least significant bit, respectively. If $x = x[0]||x[1]||\ldots x[127]$, and $y = rot_r(x)$, then $y = x[r]||x[r+1]||\ldots x[127]||x[0]||x[1]||x[r-1]$. Of course, it holds that $rot_r(rot_r^{-1}(x)) = x$ and $rot_r^{-1}(rot_r(x)) = x$.

To state complexities, we use the big-O notation, where we use $O(f(n))$ to hide constant factors and $O^*(f(n))$ to hide polynomial factors.

2.2 The AKA Protocol and Milenage Algorithms

Cellular protocols base their security on seven secret-key cryptographic functions, referred to as a authentication and key generation algorithm set. Upon session establishment between the home

Attack	Model	Classical Queries	Quantum Queries	Complexity	OP Known?	Best Known Classical Attack	Description
Grover's attack for key recovery, OP known	Q_1	$O(1)$	0	$O\left(2^{\lvert K\rvert/2}\right)$	Yes	$O\left(2^{\lvert K\rvert}\right)$	Sec. 4.1
Grover's attack for key recovery, OP unknown	Q_1	$O(1)$	0	$O\left(2^{(\lvert K\rvert+\lvert OP_c\rvert)/2}\right)$	No	$O\left(2^{\lvert K\rvert+\lvert OP_c\rvert}\right)$	Sec. 4.1
Key Recovery f_2, OP unknown	Q_2	0	$O(\lvert M\rvert)$	$O\left(\lvert M\rvert^3 \cdot 2^{\lvert K\rvert/2}\right)$	No	$O\left(2^{\lvert K\rvert+\lvert OP_c\rvert}\right)$	Sec. 4.2
Offline Key Recovery f_2, OP unknown	Q_1	$O\left(2^{\lvert M\rvert}\right)$	0	$O^*\left(2^{\lvert M\rvert} + 2^{\lvert K\rvert/2}\right)$	No	$O\left(2^{\lvert K\rvert+\lvert OP_c\rvert}\right)$	Sec. 4.2
Existential Forgery f_1	Q_2	$O(1)$	$O(\lvert M\rvert)$	$O\left(\lvert M\rvert^3\right)$	No	$O\left(2^{\lvert M\rvert/2}\right)$	Sec. 4.3
Related Key Attack $f_1, \ldots, f_5$	Q_2	0	$O(\lvert K\rvert)$	$O\left(\lvert K\rvert^3\right)$	No	$O\left(2^{\frac{\lvert K\rvert+\lvert OP_c\rvert}{2}}\right)$	Sec. 4.4
Offline Related Key Attack $f_1, \ldots, f_5$	Q_1	$O\left(2^{\lvert K\rvert/3}\right)$	0	$O^*\left(2^{\lvert K\rvert/3}\right)$	No	$O\left(2^{\frac{\lvert K\rvert+\lvert OP_c\rvert}{2}}\right)$	Sec. 4.4

Table 1: Summary of the results. $|K|$ is the length of the message authentication key, $|OP_C|$ is the length of the OP_c bitstring and $|M|$ is the block length of the underlying block cipher. In the case of Milenage, $|K| = |OP_C| = |M| = 128$. For all complexity estimates, the big-O notation hides only a very small multiplicative constant.

network and the subscriber, these algorithms are used to authenticate the subscriber to the network and derive keys that are in turn used protect subsequent communication. To this end, telecommunication operators assign each subscriber a secret key, the network authentication key, denoted as K. The operator provisions each subscriber's SIM card with their individual network authentication key. To authenticate itself to the network, the subscriber then takes part in a challenge-response protocol, the so-called Authentication and Key Agreement (AKA) protocol. The use of the AKA protocol is mandated through standardization bodies — all cellular networks follow this protocol.

The AKA protocol is built around a set of cryptographic functions $f1, \ldots, f5$ and $f1^*, f5^*$, keyed with the network authentication key K. In summary, the subscriber sends the telecommunication operator an authentication request, containing the subscriber's identity. The operator then generates a random challenge $RAND$ and uses one of the provided cryptographic functions to calculate a corresponding response. The operator then sends the challenge $RAND$ to the subscriber's device, which derives the response using the same cryptographic function and sends the derived response to the network. If the derived response and the expected response match, the subscriber has successfully authenticated themselves to the operator. In addition, the cryptographic functions $f1, \ldots, f5^*$ are used to derive additional key material for encryption and integrity protection of subsequent messages and transferred user data. The exact details of the AKA protocol are not required to understand the present analysis — however, it is important to note that the results of the functions $f1$ and $f2$ are sent in cleartext over the network upon authentication. A more detailed protocol description is given in Appendix A.

Note that if an attacker obtains a subscriber's secret key K, the attacker can impersonate the respective subscriber towards the home network. This amounts to a complete account takeover. In addition, an attacker can derive all keys used for encryption and integrity protection and thus eavesdrop on all communication between the subscriber and the home network. Therefore, the security of cellular networks is completely contingent on the security of the cryptographic functions used in the AKA protocol.

The most commonly used set of functions for the AKA protocol is the Milenage authentication and key generation algorithm set. The Milenage algorithm set consists of five basis functions, $h1, \ldots, h5$,[1] whose outputs are mapped to the seven required outputs for the functions $f1, \ldots, f5^*$. Figure 1 describes the Milenage algorithm set, standardized through the 3rd Generation Partnership Project (3GPP) [2]. All five functions take as input the random 128-bit challenge $RAND$, generated by the operator upon registration of the subscriber's device towards the network. The second to fifth basis function, $h2, \ldots, h5$, take this random challenge as an input and output:

$$hi_{K,OP_c}(RAND) = E_K[c_i \oplus rot_{r_i}(OP_c \oplus E_K[RAND \oplus OP_c])] \oplus OP_c,$$

[1]The standard denotes the basis functions as OUT1, . . . , OUT5

where the function E_K, also referred to as the kernel, is a block cipher with block and key length of 128-bit.

The first basis function $h1$ takes as an additional input a 128 bit-string $IN1$, that is composed of the concatenation of a sequence number SQN and a fixed authentication management field AMF. The function $h1$ is then defined as:

$$h1_{K,OP_c}(RAND, IN1) = E_K[TEMP \oplus rot_{r_1}(IN1 \oplus OP_c) \oplus c_1] \oplus OP_c,$$

where $TEMP = E_K[RAND \oplus OP_C]$.

The output of the basis functions is mapped to the seven required outputs $f1, \ldots, f5^*$ as follows. The first 64 bits of the $h1$ output are mapped to represent the output of $f1$, the last 64 bits of $h1$'s output are used as the output of $f1^*$. The output of $h2$ is split in the same vein, to obtain the outputs for $f5$ and $f2$. The basis function $h3, h4, h5$ are used as-is for the output of $f3, f4, f5^*$. To highlight this almost one-to-one relation between the basis functions and their respective AKA counterparts and to support an intuitive understanding of the implications of our attacks, we will simply refer to the basis function $h1, \ldots, h5$ as the functions $f1, \ldots, f5$ for the remainder of this paper. This is also done to emphasize that vulnerabilities in the basis functions translate into immediate insecurities of their respective AKA counterparts.

All functions in the Milenage algorithm use AES as the underlying block cipher E_K. The cipher is keyed with the network authentication key K, a 128-bit-string shared between the operator and the subscriber. The bit-strings $c_1, \ldots, c_5$ and $r_1, \ldots, r_5$ are public constants which are defined in the standard. Notably, $r_2 = 0$ and $c_1 = 0$. As additional key material, the OP_c bit-string is derived from a (potentially secret) constant OP, defined by the operator. The operator provides the additional 128-bit string OP, which was intended to provide separation between different operators [2]. The per-subscriber secret OP_c is then derived as $OP_c = E_K[OP] \oplus OP$. Note that the OP-bit string is never used directly in the Milenage algorithm set, only the derived value OP_c. As such, it suffices to store the OP_c bit-string on a subscriber's SIM card, without ever revealing the operator constant OP.

There are no requirements on how the operators generate and manage the OP-bit string. It is conceivable that each operator uses the same OP bit-string for all handed-out SIM cards, but the operator could also rotate the OP for every batch of produced SIM cards. Although the Milenage algorithm set is designed to be secure even if the OP is public, in practice, operators do not reveal the value of OP. Instead of the OP, they store the OP_c bit-string on the SIM card. Arguably, this makes attacks only harder. In the present analysis, we will show attacks for both the case when the OP bit-string is known and when it is secret.

2.3 Classical Cryptanalysis of Milenage algorithms

The Milenage algorithm set was designed to fulfill the following security requirements, as specified in [3]:

(1) *Without knowledge of secret keys, the functions f1, f1*, f2, f3, f4, f5 and f5* should be practically indistinguishable from independent random functions of their inputs (RAND||SQN||AMF) and RAND. Examples: Knowledge of the values of one function on a fairly large number of given inputs should not enable its values to be predicted on other inputs. The outputs from any one function should not be predictable from the values of the other functions (on the same or other inputs).*

(2) *It should be infeasible to determine any part of the secret key K, or the operator variant configuration field, OP, by manipulation of the inputs and examination of the outputs to the algorithm.*

(3) *Events tending to violate criteria 1 and 2 should be regarded as insignificant if they occur with probability approximately* 2^{-128} *or less (or require approximately* 2^{128} *operations).*

(4) *Events tending to violate criteria 1 and 2 should be examined if they occur with probability approximately* 2^{-64} *(or require approximately* 2^{64} *operations) to ensure that they do not have serious consequences. Serious consequences would include recovery of a secret key, or ability to emulate the algorithm on a large number of future inputs.*

So far, no attack violating this criteria has been identified. Simplified versions (not using the constant OP_c) of the Milenage algorithm set have been proven to be pseudorandom under the assumption that the kernel function E_K is a random permutation. The proof gives rise to a lower bound of 2^{64} queries for attacks on the Milenage algorithms. This lower bound is tight, i.e., 2^{64} queries suffice to identify collisions between the functions $f1$ and $f2$ or in the function $f1$ itself. Once identified, a collision allows an attacker to perform existential forgery [3]. For a full key recovery however, no attacks that perform better than exhaustive search are known. The brute-force attacks amount to a complexity of $O\left(2^{|K|}\right)$ if the OP bit-string is known, and $O\left(2^{|K|+|OP_c|}\right)$ if OP is unknown.

2.4 Quantum Computation

For a thorough introduction to quantum computing in-depth, we refer to the accessible exposition of [28]. Briefly, quantum computation can be described as follows. Quantum computation is usually modelled in the quantum circuit model. A quantum circuit consists of a sequence of quantum gates, acting on logical qubits. A *qubit* is encoded in the state of a system, which is described by a vector in a 2-dimensional Hilbert space. This vector describes a complex linear superposition of two computational basis state vectors $|0\rangle$ and $|1\rangle$, i.e, $\alpha_0|0\rangle + \alpha_1|1\rangle$, where α_0, α_1 are called the complex amplitudes of the basis states and adhere to the normalization constraint $|\alpha_0|^2 + |\alpha_1|^2 = 1$. An n-qubit state $|\psi\rangle$ is described by the complex linear superposition over all 2^n computational basis states $|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x_1, \ldots, x_n\rangle$, where again it must hold that $\sum_x |\alpha_x|^2 = 1$. *Measuring* a state $|\psi\rangle$ will output the label x with probability $|\alpha_x|^2$ and leave the system in state $|x\rangle$. *Quantum gates* that act on n qubits are unitary operators U that transform a quantum state $|\psi\rangle$ into a quantum state $U|\psi\rangle$.

2.4.1 Quantum Oracles and Quantum Complexity. When acting on a function $f : \{0,1\}^n \rightarrow \{0,1\}^n$, quantum computation requires

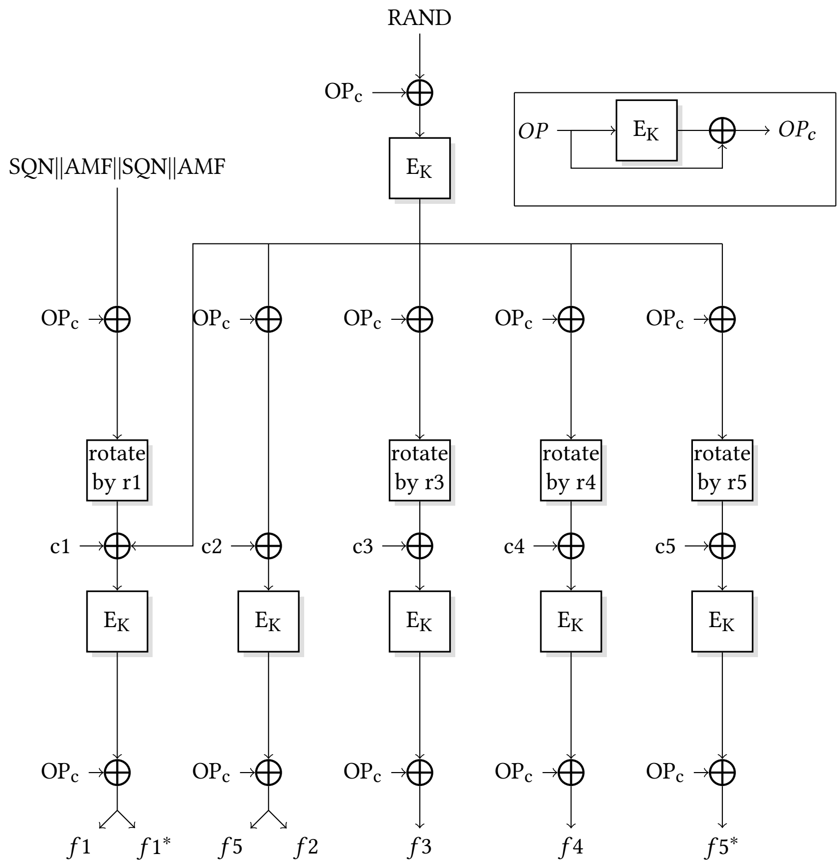

Figure 1: The Milenage algorithm set as standardized by 3GPP [2]. The outputs of the five Milenage basis functions are mapped almost one-to-one to the seven required outputs.

some kind of oracle access to this function. The oracle access is usually given through a unitary operator O_f, that performs the following calculation $O_f : |x\rangle \otimes |y\rangle \rightarrow |x\rangle \, |y \oplus f(x)\rangle$, where $x, y \in \{0,1\}^n$ and $|x\rangle, |y\rangle$ are the corresponding quantum states.

There are multiple ways to measure the complexity of quantum algorithms. We will focus here on two fundamental dimensions. The *query* complexity and the *time* complexity. Query complexity measures the number of accesses to the oracle O_f, while *time* complexity is measured by the depth of the respective quantum circuit consisting of elementary gate operators from a universal quantum gate set, cf. [28].

We note here that this model abstracts away constraints that arise when actually implementing physical systems for quantum computation. For example, instead of measuring just the depth of the circuit, it has been proposed to include also the number of qubits (the width of the circuit) [19], to account for the fact that ensuring coherence of idle qubits might be costly. Unless otherwise mentioned, our work will focus on the time and query complexity of the described attacks. Arguably, these metrics are fine-grained enough to justify the impact of the presented results. Accounting for other metrics would require to model the designed circuits in more detail, which we leave as future work.

2.5 Attacker Model

Almost all attacks described in this paper assume access to an encryption oracle which can be queried with arbitrary plaintexts. This follows the standard security model of a known plaintext attack. In quantum cryptanalysis, the attacker's capabilities are additionally determined by the kind of queries that are allowed to this oracle, namely whether only classical or also superposition queries are allowed.

In more detail, let $F = \{f_k : \{0,1\}^n \rightarrow \{0,1\}^n\}_{k \in \{0,1\}^n}$ be a family of functions indexed by k and assume that for any given $k, x \in \{0,1\}^n$, there exists a polynomial-time algorithm to compute $f_k(x)$. Intuitively, each function f_k defines encryption under key k. For a given function f_k sampled from F, the attacker is given oracle access to f_k, denoted by O_{f_k}. Following other quantum cryptanalytic works [20, 34], we will consider two quantum adversary models, distinguished by the capabilities of their oracle access.

In the *standard security* model, or Q_1 model, the attacker can only make classical queries to the function f_k. In this case, the oracle O_{f_k} is a classical function $O_{f_k} : \{0,1\}^n \mapsto \{0,1\}^n$.

In the *quantum security model*, or Q_2 model, the attacker is allowed to query the oracle in superposition. That is, the attacker can provide as input to the oracle O_{f_k} a superposition $\sum_{x,y} \lambda_{x,y} |x\rangle |y\rangle$ and the oracle will return the output $\sum_{x,y} \lambda_{x,y} |x\rangle |y \oplus f_k(x)\rangle$.

We stress that even in the Q_1 model, the attacker can still guess the key k and then construct (and access) a quantum circuit that, given any $k, x \in \{0,1\}^n$, efficiently evaluates $f_k(x)$. This quantum circuit can receive as input any superposition of k and x. We will make use of this *offline* computation later on.

Note that all Milenage functions $f1, \ldots, f5$ can be viewed as a function family F, where generating a random secret key k amounts to sampling a function from the family F. The attacker is given access to an oracle O_{f_k}, which evaluates a function f_k with a fixed key k, where k is not known by the attacker.

3 THE QUANTUM CRYPTANALYSIS TOOLBOX

In recent years, symmetric cryptography has received increasing scrutiny with respect to resilience against quantum attacks. This quantum cryptanalysis of symmetric cryptography has mostly uncovered new attacks in the Q_2 model, but also yielded speedups in the Q_1 model. Most of the cryptanalytic works present quantum algorithms that equip quantum attackers with powerful attack primitives that can be used as a black box. We follow this approach and present in this section a *quantum toolbox*. I.e., a set of algorithms that facilitate cryptanalytic attacks on symmetric key cryptography. To keep our work accessible to researchers outside of the quantum community, we will hereafter use these algorithms only as a black box.

The quantum cryptanalysis presented in this paper is based on three algorithms. Grover's algorithm to speed up exhaustive search, Simon's algorithm to identify a hidden period, and the offline version of Simon's algorithm, which combines the two former algorithms to speed up attacks in the Q_1 model. In this section, we will briefly describe the intuition of the relevant algorithms, the problems they solve, the requirements for their usage and their respective complexity. For the remainder of this work, we will then use these algorithms as a black box and focus our analysis on classical constructions that will then allow us to employ quantum algorithms in a simple fashion.

3.1 Grover's Algorithm: Fast unstructured search

In his seminal work, Grover [17] described an algorithm that achieves a quadratic speedup when performing an unstructured, brute-force search. We state the main result as relevant for this paper as follows, where we ignore small constants in Grover's time and query complexity and also the extremely high success probability for better readability.

Theorem 1 (Grover's Algorithm). *Consider a function $f : \{0,1\}^n \to \{0,1\}$, such that 2^t inputs map to 1 and the rest maps to 0. Given quantum oracle access to the function f, Grover's algorithm finds a preimage of 1, i.e., a $k \in \{0,1\}^n$ satisfying $f(k) = 1$, in $\sqrt{2^n/2^t}$ time and oracle queries. If there is exactly one preimage of 1, i.e. only one k such that $f(k) = 1$, then Grover's algorithm finds this k in in $\sqrt{2^n}$ time.*

Intuitively, Grover's algorithm "cooks" a solution k_0, such that $f(k_0) = 1$, by constructing an equal superposition over all inputs in the domain of f and repeating a sub-procedure that increases the amplitude of k_0 while decreasing all other amplitudes. For a detailed explanation, we refer the reader to the standard literature [17, 28]. Note that Grover's algorithm requires quantum oracle access to f.

In quantum cryptanalysis, Grover's algorithm is typically used to speed up the exhaustive search (bruteforce) of a key. To this end, an attacker can construct a quantum circuit for a given cipher, e.g., AES. This circuit will take as input a message and a key guess k^* and will return the encryption of the message under the key k^*. To then bruteforce the key for a fixed but unknown key k, the attacker first captures enough plaintext-ciphertext pairs so that the secret key is uniquely determined by those pairs. An attacker can then easily construct a quantum circuit for a function f that, on input of a key guess k^* returns 1 if k^* is equal to the correct k and zero otherwise. The construction works as follows. The quantum circuit encrypts the collected plaintexts under the key guess k^* and compares the resulting ciphertexts with the captured ciphertexts. If they match, f returns 1, otherwise f returns 0. Thus, an attacker can construct a quantum cirucit for f and then leverage Grover's algorithm to find the key k in time $2^{|k|/2}$.

3.2 Simon's Algorithm: Quantum Period Finding

Simon's algorithm can identify hidden period in a function f in polynomial time, given quantum oracle access to this function. This powerful primitive has been successfully used in various quantum attacks on symmetric cryptography [8, 21, 22]. Formally, Simon's algorithm solves the following problem:

Definition 1 (Simon's problem). *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function that is either injective, or there exists a single period $s \neq 0^n$ such that*

$$\forall x \neq x' : f(x) = f(x') \iff x' = x \oplus s;$$

determine s.

Given quantum oracle access to f through an oracle O_f, this problem can be solved with $O(n)$ quantum queries to f and $O(n^3)$ time using Simon's algorithm [31]. In summary, Simon's algorithm relies on a quantum subroutine which queries the function f with a superposition query and returns a random value y, s.t. $y \oplus s = 0$ or a random y if f is injective. After $c \cdot n$ invocations of Simon's quantum subroutine (for a small constant $c \geq 1$), we obtain n linear independent vectors $y_1, \ldots, y_n$, such that $y_i \oplus s = 0$. This gives rise to an equation system and allows us to recover s via Gaussian elimination.

Note that for cryptanalytic purposes, where f represents some sort of cryptographic construction, f does not necessarily fulfill the requirement of Simon's problem perfectly. Instead, there might be unwanted collisions in f. Kaplan et al. [21] showed that Simon's algorithm can still recover the period s efficiently, provided that the probability of an unwanted collision is bounded away from 1. They prove the following theorem.

Theorem 2 (Simon's algorithm with approximate promise). *Let $f : \{0,1\}^n \to X$ be a function with period s. Define the probability of an unwanted collision as*

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

If $\varepsilon(f, s) \leq p_0 < 1$, then with $c \cdot n$ calls of the quantum subroutine, Simon's algorithm returns s with probability at least

$$1 - \left(2 \cdot \left(\frac{1+p_0}{2}\right)^c\right)^n .$$

Note that the theorem also holds for cases where the codomain of the function is smaller than the domain, i.e., $|X| < 2^n$. It follows from Theorem 2 that as long as $c \geq 3/(1-p_0)$ the error probability decreases exponentially in n. Thus, given a constant bound on p_0 on the probability of unwanted collision for a function f, we can recover that function's period s with $O(n)$ quantum queries and polynomial time. Throughout this paper, we will make implicit use of a related theorem. For almost all functions with large enough outputs (in terms of bit length), the impact of unwanted collisions on the query cost is negligible, c.f. [7]. This allows us to ignore the issue of unwanted collisions for the remainder of this paper at all, since we will only deal with functions that have large enough outputs.

3.3 Offline Simon's algorithm: Attacks without superposition queries

In the Q_1 model, superposition queries to an oracle O_f are not possible. Instead, the attacker can only query O_f classically. Many quantum cryptanalytic attacks on symmetric ciphers thus are not applicable in the Q_1 setting, since the attacks require superposition queries to the attacked cipher. However, even in the Q_1 setting, quantum computers can speed up attacks. Indeed, Bonnetain et al. [8] introduced a new algorithm, called the "Offline Simon's Algorithm", which leverages structural properties of cryptographic schemes to execute quantum attacks which are ways faster than their known classical counterparts [8, 9]. The "Offline Simon's Algorithm" can be divided into two phases. An online phase, in which the attacker makes classical queries to the oracle. The results of the classical queries are then used to assemble a database of function inputs/outputs in superposition. Once this database is established, an offline phase follows. In the offline phase the attacker uses the database to run a quantum search and period finding algorithms. The key idea of the offline Simon's algorithm is that the database can be reused throughout the whole offline phase, without any further additional oracle queries. Thus, reusing the database yields speedups in the Q_1 model, as well as a reduced query complexity in the Q_2 model.

In more detail, the offline Simon's algorithm is applicable in the following situation. Consider a function $g : \{0,1\}^n \to \{0,1\}^l$ to which an attacker has only classical oracle access and a family of functions $F = \{f_i : \{0,1\}^n \to \{0,1\}^l, i \in \{0,1\}^m\}$. Assume that given any $(i, x) \in \{0,1\}^m \times \{0,1\}^n$, there exists a polynomial-time quantum circuit to compute $F(i, x) = f_i(x)$. For example, g might be an encryption oracle for an encryption under a fixed (and unknown) key k with a cipher E, while the function $F(i, x)$ is an encryption through the cipher E under a key i that is provided as input to the circuit. Further assume that there exists an $i_0 \in \{0,1\}^m$ such that $f_{i_0} \oplus g$ has a hidden period, i.e., $f_{i_0}(x) \oplus g(x) = f_{i_0}(x \oplus s) \oplus g(x \oplus s)$ for some $s \in \{0,1\}^n$.

The following result due to Bonnetain et al. [8] shows that in this setting, the strategy described above can be used to achieve a substantial speed up over classical algorithms when searching for the value i_0 and the period s.

Theorem 3 (Asymmetric Search of a Period). *Let $F = \{f_i : \{0,1\}^n \to \{0,1\}^l, i \in \{0,1\}^m\}$ be a family of functions, define $F(i, \cdot) = f_i(\cdot)$ and let g be a function $g : \{0,1\}^n \to \{0,1\}^l$. Assume that we are given quantum oracle access to F. Further, assume that there exists exactly one $i_0 \in \{0,1\}^m$ such that $f_{i_0} \oplus g$ has a hidden period, i.e., for all $x \in \{0,1\}^n$ it holds that $f_{i_0}(x) \oplus g(x) = f_{i_0}(x \oplus s) \oplus g(x \oplus s)$ for some s. Moreover, let the probability of unwanted collisions for all $f_i \oplus g$ be bounded from above by $1/2$, i.e.,*

$$\max_{\substack{i \in \{0,1\}^m \setminus \{i_0\} \\ t \in \{0,1\}^n \setminus \{0^n\}}} Pr_x[f_i(x) \oplus g(x) = f_i(x \oplus t) \oplus g(x)] \leq \frac{1}{2}.$$

Then, offline Simon's algorithm can identify i_0 with the following complexities:

1. *If we are given classical oracle access to g, then we can identify i_0 with extremely high success probability using $O(2^n)$ classical queries to g and additional computations with a time complexity of $O((n^3 + nT_F) \cdot 2^{m/2})$, where T_F is the time required to evaluate F once.*
2. *If we are given quantum oracle access to g, then we can identify i_0 with extremely high success probability, using $O(n)$ quantum queries to g and additional computations with time complexity $O((n^3 + nT_F) \cdot 2^{m/2})$.*

The offline version of Simon's algorithm leverages Grover's algorithm to search for the i_0 such that $f_{i_0} \oplus g$ has a period, and uses Simon's algorithm as a sub-procedure in that search to verify that a given guess i^* indeed results in a period for the function $f_{i^*} \oplus g$.

In the case where only classical access to g is provided, Bonnetain et al. [8] first build up a database of all $O(2^n)$ input-outputs pairs of g to obtain a superposition

$$|\phi_g\rangle = \bigotimes^{c \cdot n} \left(\sum_{x \in \{0,1\}^n} |x\rangle \, |g(x)\rangle \right),$$

where $\bigotimes$ is the usual tensor product, cf. [28]. This database can then be used to run the above-mentioned combination of Grover and Simon without any additional classical or quantum queries to g. In the case where quantum access to g is provided, this database can be built faster by querying g in superposition directly. Note that once that i_0 such that $f_{i_0} \oplus g$ has a period s is identified, we can recover the actual period s in polynomial time using Simon's algorithm — again reusing the g-database $|\phi_g\rangle$.

Throughout this paper, we will make use of the fact that the offline Simon's algorithm is also applicable in a more generalized setting, where the attacker combines the function g with a quantum circuit through means other than xoring the results [7, 8].

Theorem 4 (Generalized Offline Simon's Algorithm). *Consider a family of functions $F_i : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^l$, indexed by $i \in \{0,1\}^m$. Let g be a function $g : \{0,1\}^n \to \{0,1\}^l$ to which the attacker has classical or quantum oracle access and $p_i : \{0,1\}^n \to \{0,1\}^n$ be a permutation. Assume that for the index value i_0, the function $F_{i_0}(x, g(p_{i_0}(x)))$ has some period s. The Offline Simon's algorithm can identify i_0 with extremely high success probability, with the following complexities:*

(1) *If we are given classical oracle access to g, then we can identify* i_0 *using* $O(2^n)$ *classical queries to g and additional computations with time complexity* $O((n^3 + nT_F) \cdot 2^{m/2})$*, where* T_F *is the time required to evaluate F once.*
(2) *If we are given quantum oracle access to g, then we can identify* i_0 *using* $O(n)$ *quantum queries to g and additional computations with time complexity* $O((n^3 + nT_F) \cdot 2^{m/2})$.

In the same vein as Simon's algorithm, the offline Simon's algorithm can deal with unwanted collisions; again, for functions with large enough output the impact of unwanted collisions can be neglected [7].

4 QUANTUM CRYPTANALYSIS OF THE MILENAGE ALGORITHMS

The main idea of this paper is to leverage the above described quantum toolbox to perform a quantum cryptanalysis of the Milenage algorithm set. To this end, we extend existing attacks on symmetric ciphers to perform forgery attacks or recover the secret key K. Our results go beyond the trivial Grover attack and show the complexity of quantum security considerations. Contrary to common belief, a larger key size might not be sufficient to ensure quantum resilience of the Milenage algorithm set. We discuss the consequences of our work in more detail in Section 5.

To describe the complexities of the presented attacks, we will consider three parameters:

- the length of the secret key K,
- the length of the OP_c bit-string, and
- the block length of the underlying block-cipher E_K, which we denote by $|M|$.

Note that for the current Milenage configuration it holds that $|K| = 128, |OP_c| = 128$ and $|M| = 128$. With this we can summarize our four different attacks as follows.

(1) For reasons of (exposition) completeness, the trivial Grover attack that results in a quadratic reduction of the time complexity of exhaustive key search.
(2) A quantum slide attack against the $f2$ function, which reduces the complexity of recovering the secret key material in case the OP bit-string is not known. If quantum superposition access to $f2$ is granted, the attacker can acquire the OP_c and the key K with only $O(|M|)$ superposition queries and $O^*(2^{|OP_c|/2})$ time. If the attacker is given only classical access to $f2$, then we require $O(2^{|M|})$ online classical queries, and the attack has a time complexity of $O^*(2^{|M|} + 2^{|OP_c|/2})$, i.e., approximately 2^{64} operations in the Q_2 model and $2^{128} + 2^{64}$ operations in the Q_1 model. To the best of our knowledge, recovering the network authentication key K as well as the OP_c bit-string in a classical known-plaintext attack would require $O\left(2^{|K|+|OP_c|}\right)$, i.e. approximately 2^{256}, classical operations.
(3) A quantum polynomial time existential forgery attack on the MAC function $f1$, assuming quantum superposition access to $f1$. Classical attacks that achieve existential forgery on the $f1$ cipher require $O(2^{|M|/2})$ operations and queries, i.e., approximately 2^{64} queries and time in the current Milenage configuration.
(4) A quantum related key attack against Milenage, which can recover the secret key in polynomial time in the Q_2 model, and in $O\big(2^{(|K|+|OP_c|)/3}\big)$ time in the Q_1 model.

4.1 The Grover Key Recovery for $f1, \ldots, f5$

We first describe the most obvious attack on the Milenage algorithms, that gives an upper bound on the complexity of quantum attacks. Note that the Milenage algorithms only rely on AES encryption and the xor operation — both of these operations can be fully simulated by a quantum computer [35]. We can thus use Grover to execute the following attack:

(1) Using classical oracle access to one of the functions $f1, \ldots, f5$, obtain enough function input/outputs pairs $(c_1, m_1), \ldots, (c_r, m_r)$ to uniquely determine the network authentication key K and — if required — the bitstring OP_c.
(2) Given these plaintext/ciphertext pairs, we can construct a quantum circuit for the following function f: on input of a key guess K^*, OP_C^*, return 1 if $K^* = K, OP_c^* = OP_c$ and zero otherwise. This circuit can be constructed as described in Section 3.1.
(3) By this quantum circuit, we now have quantum oracle access to the function f. This allows us to apply Grover's algorithm to search for the key K and the bit-string OP_c.

With Theorem 1, the attack can recover the key with time and query complexity $O(2^{|K|+|OP_c|/2}))$ or $O(2^{|K|/2})$ if the bit-string OP is known. For the current Milenage configuration, this transfers to approximately 2^{128} and 2^{64} operations, respectively. Up until now, quantum security considerations [24, 33] took only this obvious and simple attack scenario into account. Clearly, the impact of the quadratic speedup that results from Grover's algorithm can be mitigated by simply doubling the key size. As we will showcase with the following attacks, other quantum attacks on the Milenage algorithm set will provide more than a quadratic speedup — up to exponential speedups.

4.2 Quantum Slide Attacks Against f_2

Bonnetain et al. [8] describe that the offline Simon algorithm can be used to execute a quantum slide attack against a 2-round self-similar cipher. A self-similar cipher builds upon a block cipher E to encrypt a message m, using two keys k_1, k_2 in the following way:

$$iFX(m) = E_{k_2}[E_{k_2}[m \oplus k_1] \oplus k_1] \oplus k_1.$$

The attack described by Bonnetain et al. [8] yields a speedup compared to classical attacks. This *quantum slide attack* can be adapted to work on the $f2$ function as well.

To this end, we first show how the $f2$ function be transformed into a 2-round self-similar cipher and then describe how the attack described by Bonnetain et al. [8] can be applied to our construction. This leads to an attack that reduces the additional security provided by the OP_c bit-string, a value which is unknown in practice.

In more detail, recall that function f_2 is defined as

$$f2(m) = E_K[rot_{r_2}(E_K[m \oplus OP_C] \oplus OP_C) \oplus c_2] \oplus OP_c.$$

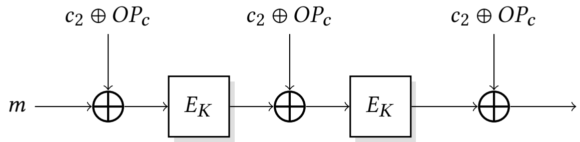

Figure 2: The $f2'$ function, which now resembles an iterated FX cipher.

Now, the standard defines r_2 as $r_2 = 0$, which simplifies $f2$ to

$$f2(m) = E_K[E_K[m \oplus OP_C] \oplus OP_C \oplus c_2] \oplus OP_c$$

To transform $f2$ into a self-similar cipher, we define the function f_2', which for each input m instead queries f_2 for $m \oplus c_2$ and then xor's the result with c_2. I.e.,

$$\begin{aligned} f2'(m) &\stackrel{\text{def}}{=} f2(m \oplus c_2) \oplus c_2 \\ &= E_K[E_K[m \oplus c_2 \oplus OP_C] \oplus OP_C \oplus c_2] \oplus OP_c \oplus c_2. \end{aligned}$$

Note that c_2 is public. As a result, if the attacker has (quantum) oracle access to $f2$, the attacker can easily construct a quantum circuit to also have (quantum) oracle access $f2'$. Clearly, $f2'$ follows the description of a self-similar cipher, as visualized in Figure 2.

This enables us to execute the attack presented in [8], which we now describe in the following. Define the functions p_i, F_i, g as follows:

$$F_i((b,x),y) \stackrel{\text{def}}{=} \begin{cases} y \oplus x & \text{if b=0} \\ E_i(y) \oplus x & \text{if b=1} \end{cases}$$

$$p_i((b,x)) \stackrel{\text{def}}{=} \begin{cases} E_i(x) & \text{if b=0} \\ x & \text{if b=1} \end{cases}$$

$$g(x) \stackrel{\text{def}}{=} f2'(x).$$

We combine now the above functions into a function F_i^*, indexed by i, which will have the desired hidden period,

$$F_i^*(b,x) \stackrel{\text{def}}{=} F_i((b,x), g(p_i(b,x))).$$

Note that for a given i, an attacker can easily construct an efficient quantum circuit for $F_i((b,x),y)$ and $F_i^*(b,x)$.

The function $F_k^*(b,x) = F_k((b,x), g(p_k(b,x)))$ has a hidden period $(1, OP_c \oplus c_2)$, as shown below. This is sufficient to apply the offline Simon's algorithm. Armed with Theorem 4 and the above definitions, we arrive at the following complexities.

- In the Q_2 setting, the attack requires $O(|M|)$ superposition queries to $f2$ and $O^*(2^{|K|/2})$ time. For the current Milenage configuration, this results in $c \cdot 128$ superposition queries and $c \cdot 2^{|K|/2}$ operations for a small constant c.
- In the Q_1 setting, the attack requires more time and queries to prepare the database of g's input-output pairs. To this end, the attacker needs to query $f2'(x)$ for all possible $2^{|M|}$ inputs. Once the database is prepared, the attacker can recover the key K as well as the OP_c bit-string via the offline Simon's algorithm. As such, the attack takes $O(2^{|M|})$ online classical queries, and has a time complexity of $O^*(2^{|M|}+2^{|K|/2})$. For the current Milenage configuration, this results in $c \cdot 2^{128}$ superposition queries and $c \cdot (2^{128}+2^{64})$ operations for a small constant c. Note that while this is no improvement over the trivial Grover attack, the advantage of the quantum slide attack shows when increasing the AES key length to 256 bit. Then, the quantum slide attack requires $c \cdot (2^{128}+2^{128})$ operations, while the Grover attack requires $c \cdot (2^{384/2}) = c \cdot 2^{192}$ operations, for a small constant c.

To the best of our knowledge, the best classical attack against the $f2$ construction — when both the OP bit-string as well the network authentication key K are unknown — has a complexity of approximately 2^{256}. Therefore, the presented quantum slide attack reduces the additional security provided by the OP_c bit-string significantly. In contrast, to the best of our knowledge, classical slide attacks against the $f2$ construction do not provide any advantage over a bruteforce attack [12].

It remains to be shown that $F_k^*(b,x) = F_k((b,x), g(p_k(b,x)))$ indeed has the hidden period $(1, OP_c \oplus c_2)$. To see why, first observe that

$$f2'(E_K(x \oplus OP_c^*)) \oplus (x \oplus OP_c^*) = E_K(f2'(x)) \oplus x, \quad (1)$$

where we write $OP_c^* = OP_c \oplus c_2$ for the sake of brevity. To see why Equation 1 holds, note that:

$$\begin{aligned} & f2'(E_K[x \oplus OP_c^*]) \oplus (x \oplus OP_c^*) \\ = \; & E_K[E_K[E_K[x \oplus OP_c^*] \oplus OP_c^*] \oplus OP_c^*] \oplus OP_c^* \oplus (x \oplus OP_c^*) \\ = \; & E_K[E_K[E_K[x \oplus OP_c^*] \oplus OP_c^*] \oplus OP_c^*] \oplus x \end{aligned}$$

and

$$\begin{aligned} & E_K(f2'(x)) \oplus x \\ = \; & E_K[E_K[E_K[x \oplus OP_c^*] \oplus OP_c^*] \oplus OP_c^*] \oplus x \\ = \; & f2'(E_K[x \oplus OP_c^*]) \oplus (x \oplus OP_c^*). \end{aligned}$$

Thus, it follows that $F_k^*(1,x) = F_k^*(0, x \oplus OP_c \oplus c_2)$ because

$$\begin{aligned} F_k^*(1,x) &= F_k((1,x), g(p_k(1,x))) \\ &= F_k((1,x), g(x)) \\ &= F_k((1,x), f_2'((x))) \\ &= E_k(f_2'(x)) \oplus x \end{aligned}$$

and

$$\begin{aligned} & F_k^*(0, x \oplus OP_c \oplus c_2) \\ = \; & F_k((0, x \oplus OP_c^*), g(p_k(0, x \oplus OP_c^*))) \\ = \; & F_k((0, x \oplus OP_c^*), g(E_k(x \oplus OP_c^*))) \\ = \; & f2'(E_k(x \oplus OP_c^*)) \oplus x \oplus OP_c^* \\ = \; & E_k(f2'(x)) \oplus x, \end{aligned}$$

where the last step follows from equation 1.

4.3 Existential forgery of $f1$

Our third attack is based on the seminal work of Kaplan et al. [21], who describe a polynomial time existential forgery attack against a CBC-MAC construction in the Q_2 model. As a result, if superposition queries against the CBC-MAC oracle are allowed, CBC-MACs must be considered insecure. The attack can be extended to an attack that allows for polynomial time existential forgery against the $f1$ function from the Milenage algorithm set. In the following, we provide the details of our novel quantum attack.

In summary, the attack assumes superposition access to an oracle $O_{f1_{K,OP_C}}(x,y) = f1_{K,OP_c}(x,y)$, invoking the function $f1$ on input (x,y) with a fixed network authentication key k and fixed value OP_c. Given this access, the attacker can efficiently construct $q+1$ outputs of the function $f1_{K,OP_c}$ after issuing a total of q quantum and classical queries to the function $f1_{K,OP_c}$.

Before we provide the details of the attack, recall that the function $f1$ is defined as

$$\begin{aligned} & f1_{K,OP_c}(RAND, IN1) \\ \stackrel{\text{def}}{=} \;& E_K[E_K[RAND \oplus OP_C] \oplus rot_{r_1}(IN1 \oplus OP_c) \oplus c_1] \oplus OP_c. \end{aligned}$$

Also, for the sake of brevity, we will set $x = RAND$, and $y = IN1$, where $x, y \in \{0,1\}^{|M|}$. Then, the function $f1$ can be a bit "shortened" to

$$f1_{K,OP_c}(x,y) = E_K[E_K[x \oplus OP_C] \oplus rot_{r_1}(y \oplus OP_c) \oplus c_1] \oplus OP_c.$$

To now perform an existential forgery attack, pick two arbitrary bit-strings $\alpha_0, \alpha_1 \in \{0,1\}^{|M|}$ with $\alpha_0 \neq \alpha_1$. We then define the following function $f' : \{0,1\} \times \{0,1\}^{|M|} \to \{0,1\}^{|M|}$ by

$$\begin{aligned} & f'(b,y) \\ \stackrel{\text{def}}{=} \;& f1_{K,OP_c}(\alpha_b, y) \\ = \;& E_K[E_K[\alpha_b \oplus OP_C] \oplus rot_{r1}(y) \oplus rot_{r1}(OP_c) \oplus c_1] \oplus OP_c. \end{aligned}$$

Clearly, if an attacker has access to a quantum oracle for $f1_{K,OP_C}$, the attacker can construct an efficient quantum circuit for f' as well. As we will show below, the function f' has the hidden period $(1, rot_{r1}^{-1}(\alpha_0^* \oplus \alpha_1^*))$, where $\alpha_b^* = E_k[\alpha_b \oplus OP_c]$. This hidden period can be recovered in polynomial time using Simon's algorithm. Once an attacker obtained the period $(1, rot_{r_1}^{-1}(\alpha_0^* \oplus \alpha_1^*))$, the attacker can easily perform an existential forgery. Assume the attacker knows the value $t = f1_{K,OP_c}(\alpha_0, x)$, where $x \in \{0,1\}^{|M|}$. Then he also knows the output of the function call $f1_{K,OP_c}(\alpha_1, x \oplus rot_r^{-1}(\alpha_0^* \oplus \alpha_1^*)) = f1_{K,OP_c}(\alpha_0, x) = t$. Since the $f1$ function is intended to be used as a MAC, this amounts to an existential forgery attack.

The attacks proceeds then as follows.

(1) Recover the hidden period $(1, rot_{r1}^{-1}(\alpha_0^* \oplus \alpha_1^*))$ using Simon's algorithm. Let q' denote the number of quantum queries made through running Simon's algorithm.
(2) Repeat the following steps $q'+1$ times:
   (a) Pick an arbitrary bit-string $y \in \{0,1\}^{|M|}$.
   (b) Query the function $f1_{K,OP_c}$ on input (α_0, y) to obtain $t = f1_{K,OP_c}(\alpha_0, y)$.
   (c) The same value t is also a value output/MAC tag for the input $(\alpha_1, y \oplus rot_r^{-1}(\alpha_0^* \oplus \alpha_1^*))$

This will produce a total of $2q'+2$ tags after issuing only $2q'+1$ queries. Overall the attack has a query complexity of $O(|M|)$ quantum queries to $f1_{K,OP_c}$ and $O(|M|^3)$ classical computation time. For the Milenage key lengths, this translates to $c \cdot 128$ quantum queries for a small constant c and a negligible amount of computation. This *quantum existential forgery* attack clearly violates the security requirements of the $f1$ function, as stated by 3GPP [3]:

> *"Without knowledge of secret keys, the functions f1, f1*, f2, f3, f4, f5 and f5* should be practically indistinguishable from independent random functions of their inputs (RAND||SQN||AMF) and RAND. Examples: Knowledge of the values of one function on a fairly large number of given inputs should not enable its values to be predicted on other inputs. The outputs from any one function should not be predictable from the values of the other functions (on the same or other inputs)."*

It remains to be shown that f' indeed has the hidden period $(1, rot_{r1}^{-1}(\alpha_0^* \oplus \alpha_1^*))$. To this end, we need to show that

$$f'(0,y) = f'(1, y \oplus rot_{r1}^{-1}(E_k[\alpha_0 \oplus OP_c] \oplus E_k[\alpha_1 \oplus OP_c])).$$

First, observe that by linearity of rotation it holds that

$$\begin{aligned} & f1_{K,OP_c}(x,y) \\ = \;& E_K[E_K[x \oplus OP_C] \oplus rot_{r_1}(y \oplus OP_c) \oplus c_1] \oplus OP_c \\ = \;& E_K[E_K[x \oplus OP_C] \oplus rot_{r_1}(y) \oplus rot_{r_1}(OP_c) \oplus c_1] \oplus OP_c. \end{aligned}$$

Thus, we have

$$f'(0,y) = E_K[\alpha_0^* \oplus rot_{r1}(y) \oplus rot_{r1}(OP_c) \oplus c_1] \oplus OP_c,$$

and

$$\begin{aligned} & f'(1, y \oplus rot_r^{-1}(\alpha_0^* \oplus \alpha_1^*)) \\ = \;& E_K[\alpha_1^* \oplus rot_{r_1}(y \oplus rot_{r_1}^{-1}(\alpha_0^* \oplus \alpha_1^*)) \oplus rot_{r_1}(OP_c) \oplus c_1] \oplus \\ & OP_c \\ = \;& E_K[\alpha_1^* \oplus rot_{r_1}(y) \oplus rot_{r_1}(rot_{r_1}^{-1}(\alpha_0^* \oplus \alpha_1^*)) \oplus \\ & rot_{r_1}(OP_c) \oplus c_1] \oplus OP_c. \end{aligned}$$

Now, using $rot_{r_1}(rot_{r_1}^{-1}(x)) = x$ we can continue as

$$\begin{aligned} = \;& E_K[\alpha_1^* \oplus rot_{r_1}(y) \oplus \alpha_0^* \oplus \alpha_1^* \oplus rot_{r_1}(OP_c) \oplus c_1] \oplus OP_c \\ = \;& E_K[rot_{r_1}(y) \oplus \alpha_0^* \oplus rot_{r_1}(OP_c) \oplus c_1] \oplus OP_c \\ = \;& f'(0,y), \end{aligned}$$

which indeed yields $f'(0,y) = f'(1, y \oplus rot_r^{-1}(\alpha_0^* \oplus \alpha_1^*))$.

4.4 Quantum Related Key Attacks against $f1, \ldots, f5$

Related key attacks, as introduced by Biham [6], consider attackers that can request encryption under multiple related keys. The exact values of the keys are unknown, but the way in which the keys are related is known to the attacker. The attacks can be modelled through a related key oracle, which provides the attacker access to encryption of a chosen-plaintext under related keys. Related key attacks are of interest because they have practical implications, for example when conducting fault-injection attacks. Recent works have shown that related key attacks on block ciphers can be sped up through quantum computers, both in the Q_2 as well as the Q_1 model. In the Q_2 model, with quantum superposition queries to the related key oracle, related key attacks can break any block cipher in polynomial time [30]. Using the offline Simon algorithm, the attack from [30] can be adapted to yield a speedup in the Q_1 model as well. Both attacks assume the following attacker model. For a given block-cipher E with a fixed secret K, the attacker has access to the following related key oracle:

- The oracle O_{E_K} takes as input a bitmask L and a bit string x and outputs $E_{K \oplus L}(x)$.

Considering this attacker model, classical related key attacks on an ideal block cipher require at least $2^{n/2}$ operations, where n is the key length and the bound is tight, cf. [32].

In this section, we will describe the attacks in detail and show how to apply these attacks to the Milenage algorithm set, yielding a polynomial time attack in the Q_2 model, and a speedup in the Q_1 model. The described attacks can be mounted on all Milenage functions $f1, \ldots, f5$, regardless of whether the *OP* bit string is known or unknown. To focus on an intuitive intuitive understanding, we will assume that the *OP* bitstring is public and thus the functions $f1, \ldots, f5$ take only the network authentication K as key material. The analysis for the case when *OP* is unknown follows then in an analogue fashion.

In the following, we denote by f the Milenage function under attack. Then, for a given function f_K, we assume that the attacker has access to an O_{f_K} that takes as input a bitmask $L \in \{0,1\}^n$ and a bit string $x \in \{0,1\}^n$ and outputs $f_{K\oplus L}(x)$, i.e., $O_{f_k}(L, x) = f_{K\oplus L}(x)$. In the Q_2 model, the attacker has superposition access to this oracle, while in the Q_1 model, the attacker only has classical access.

4.4.1 Quantum Related Key Attacks with Superposition Access. The quantum related key attacks described by Roetteler and Steinwandt [30] can be transferred in a one-to-one fashion to attack the Milenage algorithm set in the attacker model described above. Their attack works as follows.

Let $c = (c_1, \ldots, c_l)$ and $m = (m_1, \ldots, m_l)$ be a set of output-inputs pairs $c = (f_K(m_1), \ldots, f_K(m_l))$ such that (c, m) uniquely determines K. Assume an attacker has superposition access to a related key oracle for

$$O_{f_K}(s, m) = f_{K\oplus s}(m) = (f_{K\oplus s}(m_1), \ldots, f_{K\oplus s}(m_l)).$$

Then, define the following mapping

$$f'(s) \stackrel{\text{def}}{=} \{f_{K\oplus s}(m), f_s(m)\}.$$

Given quantum access to a related key oracle oracle $O_{f_K}(s, m)$ for f_K, one can construct an efficient quantum circuit for f'. To be efficiently encodable, f' outputs can be encoded as integers [30].

The mapping f' is two-to-one with period K, as shown below. Using Simon's algorithm, we can recover this period efficiently with only linear many queries to the related key oracle.

To see why f' is 2-to-1 with period K, let s, s' be two different bit-strings such that $f'(s) = f'(s')$ and assume $K \neq 0^n$. We consider two cases.

(1) Assume $f_s(m) = f_{s'}(m)$. As we choose the plaintexts $m = (m_1, \ldots, m_l)$ so that they uniquely determine the key, this would imply $s = s'$, which contradicts our assumption.
(2) Now let $f_s(m) \neq f_{s'}(m)$. Thus, if $f'(s) = f'(s')$, then $f_{K\oplus s}(m) = f_{s'}(m)$. The choice of plaintexts implies $K \oplus s = s'$.

4.4.2 Quantum Related Key Attacks without Superposition Access. In the Q_1 setting, the attacker only has classical access to the related key oracle $O_{f_K}(s, m)$. However, leveraging the offline Simon's algorithm, the attacker can still achieve a speedup over classical attacks [8]. We will show how to apply the offline Simon related key attack as stated by Bonnetain et al. [8] to the Milenage algorithm set.

Intuitively, the attack works by dividing the key k and the bitmask l into two parts, i.e., $k = k_1||k_2$, $l = l_1||l_2$ where $l_1, k_1 \in \{0,1\}^{|M|/3}$. We then query the oracle O_{f_K} for each possible l_1 and construct a quantum circuit F so that $F_{k_2}(l) \oplus g(l)$ has period k_1, where g is a function derived from the related key oracle. This allows us to employ the offline Simon algorithm.

Let $l = l_1||l_2$, where $l_1 \in \{0,1\}^{|M|/3}$, $l_2 \in \{0,1\}^{|M|\cdot 2/3}$ and define the following function $g : \{0,1\}^{|M|/3} \to \{0,1\}^{l\cdot|M|}$ by

$$\begin{aligned} g(l_1) &\stackrel{\text{def}}{=} O(l_1||0^{n\frac{2}{3}}) \\ &= f_{(k_1||k_2)\oplus(l_1||0^{2/3\cdot|M|})}(m). \end{aligned}$$

Moreover let F be a family of functions indexed by h so that

$$F_h(j) = f_{j||h}(m).$$

Clearly F can be efficiently represented as a quantum circuit, while querying g requires oracle access. The function $F_{k_2}(l) \oplus g(l)$ has period k_1. Thus, we have a family of functions F such that there exists a k_2 so that $f_{k_2} \oplus g$ has a hidden period. This suffices to apply the offline Simon's algorithm to recover the key part k_2. Once we obtain the k_2, we can efficiently recover k_1 as well.

Applying now Theorem 3, the attack requires $O(2^{|K|/3})$ classical queries to the related key oracle and a has a time complexity of $O^*(2^{|K|/3})$. If the OP bit-string is known, this translates to approximately 2^{43} queries and operations. If the OP bit-string is not known, then the attack requires approximately $2^{85.3}$ queries and time.

To see why the function $F_{k_2}(l) \oplus g(l)$ has period k_1 note that

$$\begin{aligned} F_{k_2}(l \oplus k_1) \oplus g(l \oplus k_1) &= f_{l\oplus k_1||k_2}(m) \oplus f_{(k_1\oplus l\oplus k_1)||k_2}(m) \\ &= f_{l\oplus k_1||k_2}(m) \oplus f_{l||k_2}(m) \\ &= g(l) \oplus F_{k_2}(l). \end{aligned}$$

5 DISCUSSION AND CONSEQUENCES

Our results call into question whether Milenage can persist as the standard cryptography algorithm for authentication and key derivation in quantum-resistance cellular networks. They do so in two ways. First, because security margins are reduced in the Q_1 model, which puts the algorithms at a disadvantage compared to other alternatives. Second, because the Milenage algorithm set does not fulfill the desired security guarantees at all in the Q_2 model. Albeit being a very powerful attacker model, cryptography best practices suggest to move to other algorithms in the case of existence of such an insecurity.

The attacks that can be executed in the Q_1 model translate into immediate attacks against the Milenage ciphers once general purpose quantum computers come into existence. Although these attacks itself do not constitute a break of the Milenage algorithm set, they still improve on best-known classical attacks, as well as the trivial Grover, "quantum bruteforce" attack (depending on Milenage's configuration). The attacks showcase how structural properties exhibited by the Milenage algorithm set allow quantum adversaries to reduce security margins. Analyzing the security implications of the Q_2 attacks is more intricate. Due to the polynomial-time existential forgery attack, the Milenage algorithm set must be considered insecure in the Q_2 model. However it is currently not clear if quantum superposition attacks are at all feasible. Nevertheless, the Q_2 attacks still provide value beyond their mere theoretical merit.

It is a standard approach in cryptography to choose a cipher which is secure against the most powerful adversaries over a cipher which is not, even if that adversary is not conceivable in the use-case at hand at first sight. There are multiple reasons motivating this design guideline.

First, attacks feasible in scenarios that consider powerful adversaries could be an indicator that attacks exists which are feasible considering less powerful adversaries as well. Absence of vulnerabilities against even the most powerful attacker models is a great indicator for security, while the existence of such attacks cast doubt on the overall security of a scheme, even if they do not immediately undermine security immediately. For the Q_2 model, there are already multiple reductions that tie security against superposition attacks to security against other classical attacks, further underlining the Q_2 model's importance [10]. For example, attacks that utilize the offline Simon algorithm exploit the periodic structure of the attacked cryptographic schemes. This periodic structure is only present if the respective schemes are vulnerable to superposition attacks as well. In contrast, schemes that are not vulnerable to Q_2 attacks cannot be attacked with the offline Simon algorithm [8].

The second reason to consider the strongest attacker model is motivated by a defense-in-depth point argument. While there might not be realistic scenario for quantum superposition attacks right now, this situation might very well change in the future. Consider, for example, a scenario in which a frozen smart card scenario could enable superposition attacks, as described in [14]. The usages of Milenage in 5G and 6G already span more than just the AKA protocol and further use-cases or progress in physical research might enable attackers to execute superposition attacks, which would render the whole infrastructure insecure.

Consequently, our results serve as a great starting point for quantum security considerations and bring the following matter to attention. Before making any choices on the symmetric cryptography that will underpin quantum-resistant cellular networks, the research community and the telecommunication standardization bodies need to specify exactly what security requirements the cipher needs to fulfill and what kind of adversaries the cipher needs to resist. We recommend these requirements to be as conservative as possible, following standard best practices. This would entail replacing the Milenage algorithms with a post-quantum secure alternative. In light of recent breakthroughs in quantum computing [5, 11, 16, 18, 25, 29] and a growing tendency among experts to expect quantum computers in the upcoming decade [26], the process of finding a post-quantum secure instantiation of the functions $f1, \ldots, f5$ needs to be instigated now. Our attacks require only a small amount of qubits compared to algorithms breaking e.g. RSA, highlighting the imminent danger of the quantum threat towards Milenage.

The recommendation to replace Milenage is corroborated by the fact that moving away from Milenage to another cryptographic primitive that does not suffer from the presented vulnerabilities is indeed feasible. In fact, [4] show that certain block cipher modes of operations are secure against superposition queries as as long as the underlying cipher is secure against superposition queries. Moreover, with the TUAK algorithm set, an alternative to the Milenage algorithm set has already been standardized [1]. The TUAK algorithm set is based on the Keccak-f-permutation, which so far withstood quantum cryptanalysis and seemingly does not exhibit the structural properties that enabled the presented attacks. We thus conjecture it be secure against the "quantum period finding" attacks presented in this paper. In addition, the TUAK algorithm set was found to provide sufficient performance to be executed on a SIM card [23], and thus poses a (great) candidate to replace the Milenage algorithm set.

6 CONCLUSION

Given that experts increasingly view large-scale quantum computers as likely [26] and faced with the slow nature of standardization bodies, quantum security considerations for cellular networks and infrastructure need to start now. Our work shows that these quantum security considerations cannot simply stop at public-key cryptography, but instead need a paradigm shift. The security of symmetric key cryptography against quantum adversaries is not ensured by doubling the key size, contrary to popular belief. Bringing together research results from recent quantum cryptanalytic work and synthesizing their results into a quantum toolbox, we were able to develop various novel attacks against the Milenage algorithm set. Against the strongest quantum adversary, Milenage must be considered insecure. Our results do not translate into an immediate quantum break of the Milenage algorithms, but they do provide strong evidence against choosing Milenage as the cryptographic cipher underpinning the security of quantum resistant telecommunication networks. We see the following research directions as necessary to ensure the security of telecommunication networks against quantum adversaries. First, symmetric cryptography that is used in telecommunication networks needs to be subject to scrutiny, investigating the resistance against quantum-enabled attacks. With the synthesized quantum toolbox, we hope to make this work accessible to non-quantum experts in the research community as well. This scrutiny should also encompass the investigation whether the results of our attacks can be improved. Second, it is necessary to clarify what security guarantees suffice and what kind of quantum adversary models can be ignored in quantum security considerations for cellular networks. The answer to this question can then guide the choice for appropriate cryptographic algorithms. Third, the security community needs to look into efficient post-quantum secure alternatives to be employed in telecommunication protocols. We strongly encourage to investigate the possibility to replace the Milenage algorithm set with a more conservative choice, without suffering a performance loss. Standardizing an algorithm which later turns out to be vulnerable to quantum adversaries would be a disaster in a post-quantum world and should be prevented under any circumstances. To this end, this work should serve as a starting point to spark further investigations into the above-mentioned questions now, to ensure a smooth transition into quantum-resistant telecommunication networks into the future.

ACKNOWLEDGMENT

The work described in this paper has been supported by the Einstein Research Unit "Perspectives of a quantum digital transformation: Near-term quantum computational devices and quantum processors" of the Berlin University Alliance. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-RIC, project identification number: 16KISK030. We would like to thank Ryan Sweke for his valuable input which greatly improved the paper.

REFERENCES

[1] 3GPP. 2014. *ETSI TR 135 231.* Technical Report (TR) 35.231. 3rd Generation Partnership Project (3GPP). https://www.etsi.org/deliver/etsi_ts/135200_135299/135231/12.01.00_60/ts_135231v120100p.pdf Version 12.1.0.

[2] 3GPP. 2016. *ETSI TR 135 206.* Technical Report (TR) 35.206. 3rd Generation Partnership Project (3GPP). https://www.etsi.org/deliver/etsi_ts/135200_135299/135206/14.00.00_60/ts_135206v140000p.pdf Version 14.0.0.

[3] 3GPP. 2019. *ETSI TR 135 909.* Technical Report (TR) 35.909. 3rd Generation Partnership Project (3GPP). https://www.etsi.org/deliver/etsi_tr/135900_135999/135909/07.00.00_60/tr_135909v070000p.pdf Version 15.0.0.

[4] Mayuresh Vivekanand Anand, Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. 2016. Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In *Post-Quantum Cryptography*. Springer, 44–63.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[6] Eli Biham. 1994. New types of cryptanalytic attacks using related keys. *Journal of Cryptology* 7, 4 (1994), 229–246.

[7] Xavier Bonnetain. 2021. Tight bounds for Simon's algorithm. In *International Conference on Cryptology and Information Security in Latin America*. Springer, 3–23.

[8] Xavier Bonnetain, Akinori Hosoyamada, María Naya-Plasencia, Yu Sasaki, and André Schrottenloher. 2019. Quantum attacks without superposition queries: the offline Simon's algorithm. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 552–583.

[9] Xavier Bonnetain, André Schrottenloher, and Ferdinand Sibleyras. 2021. Beyond quadratic speedups in quantum attacks on symmetric schemes. *arXiv preprint arXiv:2110.02836* (2021).

[10] Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. 2013. Superposition attacks on cryptographic protocols. In *International Conference on Information Theoretic Security*. Springer, 142–161.

[11] Oliver Dial, Jerry Chow, and Jay Gambetta. 2021. IBM quantum breaks the 100-qubit processor barrier. https://research.ibm.com/blog/127-qubit-quantum-processor-eagle

[12] Orr Dunkelman, Nathan Keller, Noam Lasry, and Adi Shamir. 2020. New slide attacks on almost self-similar ciphers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 250–279.

[13] Gerhard P. Fettweis and Holger Boche. 2022. On 6G and trustworthiness. *Commun. ACM* 65, 4 (April 2022), 48–49.

[14] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. 2016. Semantic security and indistinguishability in the quantum world. In *Annual international cryptology conference*. Springer, 60–89.

[15] Henri Gilbert. 2003. The Security of" One-Block-to-Many" Modes of Operation. In *International Workshop on Fast Software Encryption*. Springer, 376–395.

[16] Roger A Grimes. 2019. *Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto.* John Wiley & Sons.

[17] Lov K Grover. 1996. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 212–219.

[18] Intel Corporation. 2019. Intel introduces 'horse ridge' to enable commercially viable quantum computers. https://newsroom.intel.com/news/intel-introduces-horse-ridge-enable-commercially-viable-quantum-computers/#gs.ngaylt

[19] Samuel Jaques and André Schrottenloher. 2020. Low-gate quantum golden collision finding. In *International Conference on Selected Areas in Cryptography*. Springer, 329–359.

[20] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. 2015. Quantum differential and linear cryptanalysis. *arXiv preprint arXiv:1510.05836* (2015).

[21] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia. 2016. Breaking symmetric cryptosystems using quantum period finding. In *Annual international cryptology conference*. Springer, 207–237.

[22] Gregor Leander and Alexander May. 2017. Grover meets Simon–quantumly attacking the FX-construction. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 161–178.

[23] Keith Mayes, Steve Babbage, and Alexander Maximov. 2016. Performance evaluation of the new Tuak mobile authentication algorithm. *Proc. ICONS/EMBEDDED* (2016), 38–44.

[24] Chris J Mitchell. 2020. The impact of quantum computing on real-world security: A 5G case study. *Computers & Security* 93 (2020), 101825.

[25] Michele Mosca. 2018. Cybersecurity in an Era with Quantum Computers: Will We Be Ready? *IEEE Security & Privacy* 16 (09 2018), 38–41. https://doi.org/10.1109/MSP.2018.3761723

[26] Marco Piani and Michele Mosca. 2021. Quantum Threat Timeline Report 2021. , 87 pages.

[27] PlankQK. 2022. PlankQK: Konsortium. https://planqk.stoneone.de/partner/

[28] Eleanor G Rieffel and Wolfgang H Polak. 2011. *Quantum computing: A gentle introduction.* MIT Press.

[29] Rigetti Computing. 2021. Rigetti computing announces next-generation 40Q and 80Q Quantum Systems. https://www.globenewswire.com/news-release/2021/12/15/2352647/0/en/Rigetti-Computing-Announces-Next-Generation-40Q-and-80Q-Quantum-Systems.html

[30] Martin Roetteler and Rainer Steinwandt. 2015. A note on quantum related-key attacks. *Inform. Process. Lett.* 115, 1 (2015), 40–44.

[31] Daniel R Simon. 1997. On the power of quantum computation. *SIAM journal on computing* 26, 5 (1997), 1474–1483.

[32] Robert Winternitz and Martin Hellman. 1987. Chosen-key attacks on a block cipher. *Cryptologia* 11, 1 (1987), 16–20.

[33] Jing Yang and Thomas Johansson. 2020. An overview of cryptographic primitives for possible use in 5G and beyond. *Science China Information Sciences* 63, 12 (2020), 1–22.

[34] Mark Zhandry. 2012. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*. IEEE, 679–687.

[35] Jian Zou, Zihao Wei, Siwei Sun, Ximeng Liu, and Wenling Wu. 2020. Quantum Circuit Implementations of AES with Fewer Qubits. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 697–726.

A THE AKA PROTOCOL

The Milenage algorithm set's main usage is the AKA protocol, used for authentication and session establishment in cellular networks as well as other cellular related applications, e.g., as a variant of the Extensible Authentication Protocol (EAP), the EAP-AKA. Figure 3 describes the authentication towards the network as implemented in the 4th generation of cellular networks (LTE), using the AKA protocol and the functions $f1, \ldots, f5$. Table 2 provides an overview over all abbreviations.

In summary, the LTE-AKA protocol is a challenge-response protocol that allows the subscriber to authenticate themselves to the network. The AKA protocol also derives a session key K_{ASME} that is used for encryption and integrity protection of communication at later points. The functions $f1, \ldots, f5$ from the Milenage algorithm set serve to derive a MAC, an expected response to a challenge, and the confidentiality and integrity keys (commonly denoted as CK and IK), which are in turn used to derive session keys. The function $f5$ is used to derive an Anomity Key (AK). The AK serves to mask the Sequence Number (SQN), where the purpose of the SQN itself is to prevent replay attacks.

The authentication procedure in the fifth generation (5G) of cellular networks networks add various security and privacy enhancements to the LTE-AKA protocol, but uses the functions $f1, \ldots, f5$ in the same way. Given that the functions provide authentication and serve as a basis for later encryption and integrity protection, the security of cellular networks is completely contigent on the security of the functions $f1, \ldots, f5$.

B LIST OF ABBREVIATIONS

3GPP	Third Generation Partnership Project	3
AK	Anomity Key	14
AKA	Authentication and Key Agreement	3
SQN	Sequence Number	14
HN	Home Network	15
MME	Mobility Management Entity	15
BS	Base Station	15
MS	Mobile Station	15
LTE	Long-Term Evolution	15
EAP	Extensible Authentication Protocol	14
3GPP	3rd Generation Partnership Project	3

Table 2: Summary of Acronyms

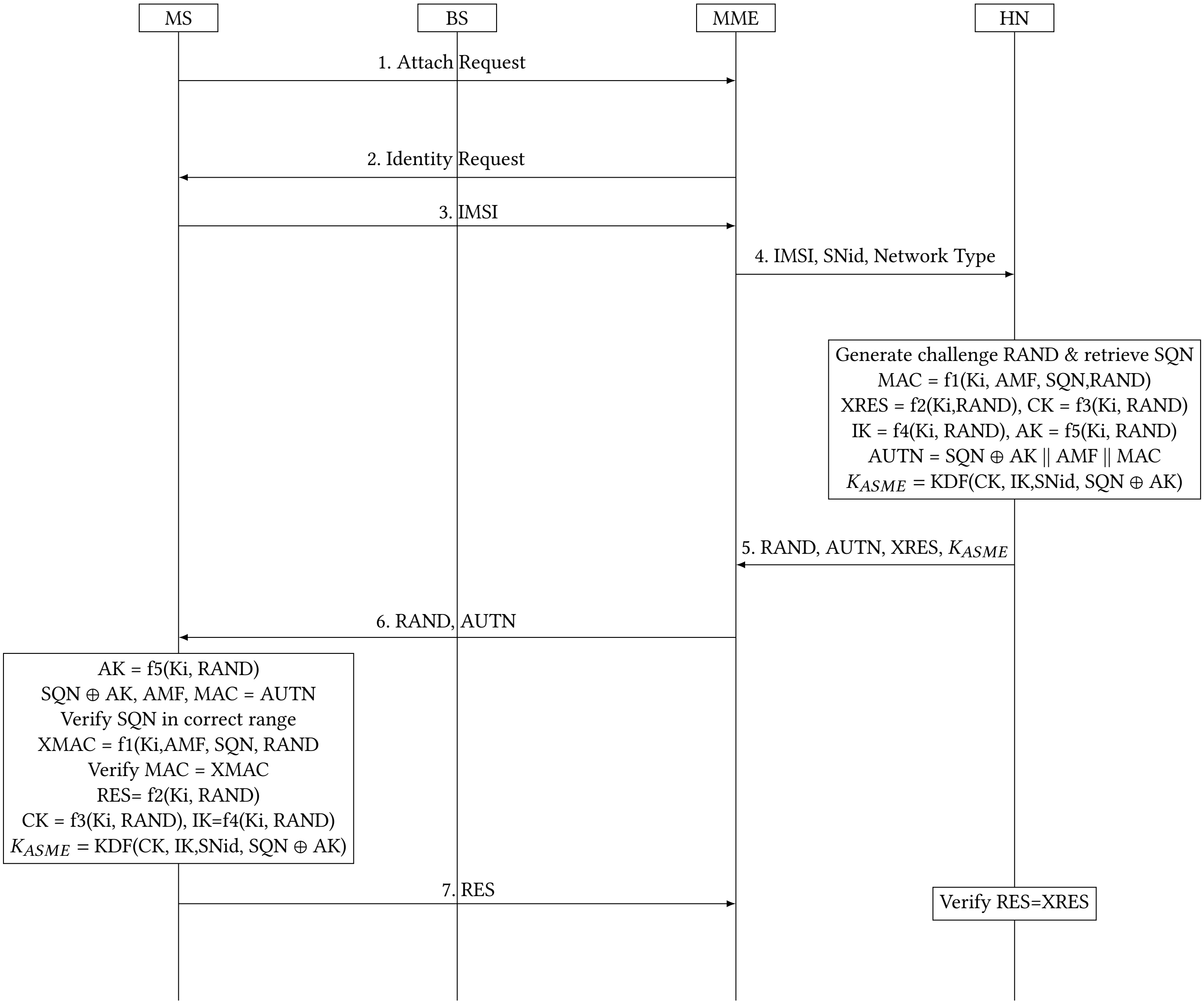

Figure 3: The AKA protocol as used in Long-Term Evolution (LTE). The user's device, referred to as Mobile Station (MS), communicates with the Base Station (BS) to authenticate towards the network. The BS forwards the request to the Mobility Management Entity (MME), which in turn forwards it to the Home Network (HN). The home network uses the function $f1, \ldots, f5$ to calculate session information and secret key material and forwards the necessary information back to the MME.